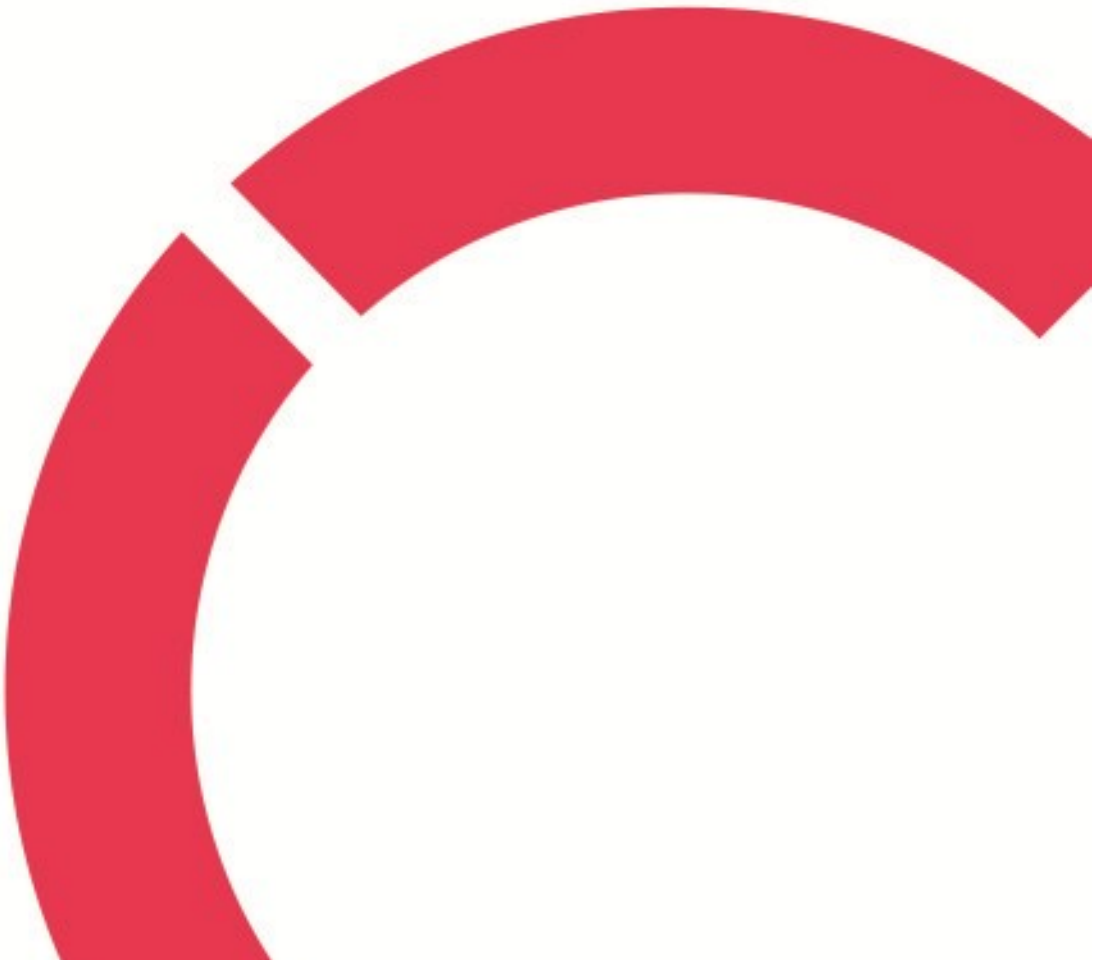
Anu Kaarlela

KUVADATAN ANNOTOINTIPROSESSI YOLO-MALLIN KOULUTUKSESSA

Opinnäytetyö
CENTRIA-AMMATTIKORKEAKOULU
Insinööri, tieto- ja viestintätekniikka
Toukokuu 2025



TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Centria-ammattikorkeakoulu	**Aika** Toukokuu 2025	**Tekijä/tekijät** Anu Kaarlela
Koulutus Tieto- ja viestintätekniikan insinööri		☒ AMK ☐ YAMK
Työn nimi KUVADATAN ANNOTOINTIPROSESSI YOLO-MALLIN KOULUTUKSESSA		
Työn ohjaaja Henry Paananen		**Sivumäärä** 36
Työelämäohjaaja Elisa Saarela		

Opinnäytetyö liittyy Centria TKI:n Optiwood-projektiin. Projektissa kerättiin dataa puuteollisuuden pintakäsitellyistä materiaaleista ja niiden mahdollisista tuotantovirheistä, suoritettiin annotointiprosessi kuvissa esiintyville virheille sekä valmisteltiin annotoiduista kuvista tietoaineisto YOLO-mallin kouluttamista varten.

Opinnäytetyön tavoitteena oli analysoida annotointiprosessin keskeisiä vaiheita, tunnistaa mahdolliset haasteet ja arvioida käytettyjen työkalujen sekä menetelmien soveltuvuutta. Lisäksi työssä tarkasteltiin, kuinka annotoinnin laatu vaikutti syväoppimismallin suorituskykyyn ja mitä parhaita käytäntöjä voidaan hyödyntää vastaavissa projekteissa. Tavoiteltuna lopputuloksena syntyi dokumentoitu annotointiprosessi, joka sisältää valitun annotointistrategian, työkalujen ja tiedostomuotojen perustelut sekä suositukset jatkokehitystä varten.

Opinnäytetyön konkreettisena tuotoksena syntyi laadukkaasti annotoitu tietoaineisto, joka on optimoitu YOLO-mallin koulutukseen. Tämä tietoaineisto tukee Optiwood-projektin tavoitteita kehittää luotettava virheentunnistusjärjestelmä puuteollisuuden pintakäsittelyn tarpeisiin.

Asiasanat
annotointi, koneoppiminen, kuvadata, pintakäsittely, puuteollisuus, YOLO-algoritmi

ABSTRACT

Centria University of Applied Sciences	Date May 2025	Author Anu Kaarlela
Degree programme Bachelor of Engineering, Information and Communication Technology		
Name of thesis THE IMAGE DATA ANNOTATION PROCESS IN YOLO MODEL TRAINING		
Centria supervisor Henry Paananen		**Pages** 36
Instructor representing commissioning institution or company Elisa Saarela		

This thesis is related to Centria RDI's Optiwood project. As part of the project tasks, image data of surface-treated wood materials and their possible production defects was collected. The annotation process was carried out for the defects in image data, and a dataset of annotated images was prepared for training a YOLO model.

The aim of the thesis was to analyse the stages of the annotation process, identify potential challenges and evaluate the suitability of the tools and methods used. In addition, the study examined how the quality of the annotation affected the performance of the deep learning model and identified best practises that could be applied in similar projects. The desired result was a documented annotation process that includes the chosen annotation strategy, justifications for the selected tools and file formats, and recommendations for further development.

The concrete result of the thesis was a high-quality annotated dataset that has been optimized for the training of YOLO model. This dataset supports the goals of the Optiwood project to develop a reliable defect detection system for the needs of wood industry's surface treatment.

Key words
annotation, image data, machine learning, surface-treatment, wood industry, YOLO-algorithm

KÄSITTEIDEN MÄÄRITTELY

ANNOTOINTI

Aineiston kuvaaminen, luokittelu ja jäsentely systemaattisella tavalla.

NEUROVERKKO

Ihmisaivojen toimintaa jäljittelevä laskennallinen malli.

SYVÄOPPIMINEN

Neuroverkkoja oppimisessa hyödyntävä tekoälymenetelmien joukko.

TIETOAINEISTO

Annotoinnin lopputuloksena syntynyt, luokiteltu aineisto (*engl. dataset*).

YOLO

Nopea ja tehokas yksivaiheinen algoritmi reaaliaikaiseen objektintunnistukseen (*YOLO = You Only Look Once).*

TAULUKOT

1 JOHDANTO

Tämä opinnäytetyö on toteutettu yhteistyössä Centria TKI:n Optiwood-hankkeen kanssa. Optiwood-hankkeen avulla pyritään tarjoamaan Pohjois-Pohjanmaan alueen puutuoteteollisuuden toimijoille mahdollisuutta optimoida prosesseja tekoälyn, data-analyysin ja älykkäiden mittausteknologioiden avulla. Tässä opinnäytetyössä tarkastellaan kuvatiedon annotointiprosessia, joka liittyy tekoälyn ja koneoppimisen hyödyntämiseen pintakäsittelyprosessissa. Pintakäsittelyn yhteydessä tuotteiden pintoihin voi syntyä virheitä, joita on vaikea havaita ihmisen näköaistilla. Näiden virheiden tunnistamiseen konenäkö ja koneoppiminen, erityisesti sen alalaji syväoppiminen, tarjoavat tehokkaan ratkaisun. Kuvissa esiintyvät virheet merkitään annotointiprosessissa ja luokitellaan ennalta määrättyihin virheluokkiin. Näistä annotoiduista kuvista muodostetaan tietoaineisto, jonka avulla koulutetaan objektintunnistukseen kehitettyä Yolov8-mallia tunnistamaan virheet automaattisesti kuvista. Syväoppimisen ja tekoälyn avulla voidaan tehostaa pintakäsittelyprosessin laatutarkastusta ja vähentää tuotantohävikkiä.

Laadukas annotointiprosessi on edellytys tarkan, luotettavan ja johdonmukaisen tietoaineiston luomiselle. Tässä opinnäytetyössä kuvataan annotointiprosessin vaiheet sekä Optiwood-projektissa käytetty annotointityökalu ja sen valintaperusteet. Työkalun valintaa on ohjannut perehtyminen aihetta käsittelevään tutkimuskirjallisuuteen, käytännön annotointimenetelmiin sekä syväoppimismallien kehittäjien suosituksiin eri työkalujen yhteensopivuudesta algoritmien kanssa.

Opinnäytetyön tavoitteena on tuottaa pintakäsittelyalan tarpeisiin soveltuva laadukkaasti annotoitu tietoaineisto, sillä vastaavaa aineistoa ei ole toistaiseksi saatavilla avoimista tietokannoista. Luotu aineisto on yksi osa Optiwood-hankkeessa kehitettävää laajempaa tietokokonaisuutta, joka koostuu useista eri lähteistä. Työssä tarkastellaan myös YOLO-algoritmin toimintaperiaatteita, jotta voidaan perustella annotointiprosessissa tehdyt ratkaisut YOLO-mallin suorituskyvyn näkökulmasta. Lopuksi valmista tietoaineistoa käytetään YOLOv8-mallin kouluttamiseen, jotta voidaan osoittaa aineiston täyttävän laadukkaan annotoinnin vaatimukset.

2 CENTRIA TKI

Centria TKI toteuttaa tutkimus-, kehitys- ja innovaatiotoimintaa yhteistyössä alueen yritysten, julkisten organisaatioiden ja kolmannen sektorin toimijoiden kanssa. Centrian TKI-toiminnan tavoitteena on parantaa alueen toimijoiden kilpailukykyä ja lisätä ihmisten hyvinvointia. Toiminnan perustana ovat aluetta palvelevat tutkimusryhmät ja tutkimusympäristöt. Lisäksi toiminnassa hyödynnetään kansallisia ja kansainvälisiä verkostoja, joiden avulla alueen yrityksiä voidaan tukea erikoistumaan omalla toimialallaan. Centria TKI tukee yrityksiä tutkimustulosten kaupallistamisessa ja liiketoiminnan kehittämisessä. (Centria-ammattikorkeakoulun tutkimus ja kehitys strategia 2021–2024.)

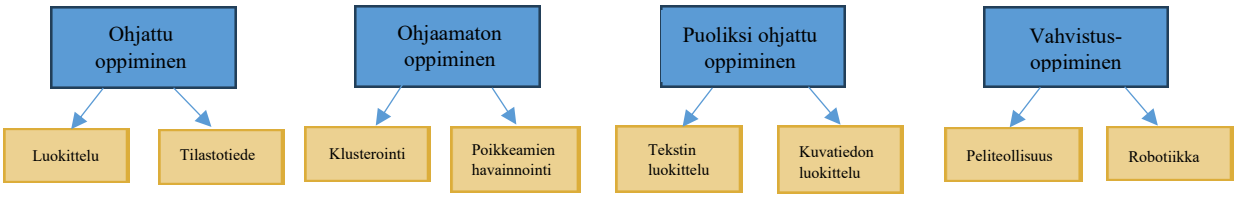
Centrian TKI-toiminnan painopisteet ovat digitalisaatio, tuotantoteknologia, kemia ja biotalous, yrittäjyys ja hyvinvointi sekä koulutus- ja kehittämispalvelut. Näiden alle on muodostettu kahdeksan tiimiä, joissa TKI-toimintaa toteutetaan. (Tutkimus, kehitys ja palvelut.) Tämä opinnäytetyö on tehty yhteistyössä tuotantoteknologian tiimiin kuuluvan Älykäs puurakentaminen ja pintakäsittely -tutkimusryhmän kanssa, ja sen aihe pohjautuu Optiwood-hankkeen tavoitteisiin. Tavoitteisiin kuuluvat aiemmin johdannossa mainitut tekoälyn, digitaalisten ratkaisujen ja älykkäiden mittausteknologioiden käyttö puutuoteteollisuuden prosessien optimoinnissa.

3 KONEOPPIMINEN

Valmistavassa teollisuudessa on havahduttu koneoppimisen mahdollisuuksiin prosessien tehostamisessa ja hävikin vähentämisessä. Koneoppimisessa ohjelmalle syötetään tietojoukko, ja ohjelmalla on kyky oppia lisää yhdistämällä tähän tietojoukkoon prosessissa opetusvaiheessa keräämäänsä aineistoa. Tämän opetusvaiheessa oppimansa tiedon perusteella ohjelma luo algoritmin, jonka avulla se pystyy ennustamaan ja tekemään parempia päätöksiä jatkossa kehittyen samalla koko ajan tehokkaammaksi työvälineeksi prosessin hallinnassa. (Alpaydin 2021, 40–41.) Koneoppimisen ja tekoälyn yhdistäminen osaksi teollisuuden prosesseja on neljänneksi teolliseksi vallankumoukseksi kutsutun aikakauden oleellinen osaalue, joka on jatkunut vahvana viime vuosikymmenen ajan (Hänninen 2022, 17).

3.1 Koneoppimisen oppimistyylit

Perinteiseen ohjelmointiin verrattuna koneoppimisen etuna on kyky oppia. Koneoppiminen voidaan jakaa neljään erilaiseen oppimistyyliin (KUVIO 1). Ohjatussa oppimisessa tietoaineisto annotoidaan eli luokitellaan ja merkitään algoritmin opettamiseksi, ja sitä käytetään esimerkiksi luokittelussa, tilastotieteessä sekä neuroverkoissa. Ohjaamaton oppiminen taas toimii päinvastoin kuin ohjattu, eikä tietojoukkoa merkitä ollenkaan. Ohjaamattomassa oppimisessa algoritmi tunnistaa tiedon mallit sekä suhteet itsenäisesti, tätä oppimistyyliä käytetään esimerkiksi klusteroinnissa ja poikkeaman havaitsemisessa. Puoliksi ohjatussa oppimisessa tietoaineisto on sekoitus jäsenneltyä ja jäsentelemätöntä tietoa, jonka avulla algoritmi oppii luokittelemaan myös merkitsemätöntä, ilman luokittelua olevaa tietoa. Puoliksi ohjatun oppimisen käyttökohde on esimerkiksi lääketieteellisen kuvatiedon luokittelu. Vahvistusoppiminen tarjoaa algoritmille mahdollisuuden oppia itse yrityksen ja erehdyksen kautta, tätä oppimistyyliä käytetään esimerkiksi peliteollisuudessa. (Choi, Coyner, Kalpathy-Cramer, Chiang & Campbell 2020; What is machine learning.)



KUVIO 1. Esimerkkejä eri oppimistyylien käytöstä koneoppimisessa.

Kuvadatan annotointiprosessissa ohjattu oppiminen on oleellista. Algoritmia koulutetaan tietoaineistolla, johon on merkitty tietyt kohteet eli esimerkiksi virheelliset kohdat. Tätä kutsutaan objektin tunnistamiseksi, jossa kuvaan merkittyjen kohteiden ympärille piirretään rajauslaatikko (*engl. bounding box*) ja luokitellaan kohteet tyypin mukaan omiin luokkiinsa. Algoritmi oppii opetusvaiheessa tunnistamaan nämä kohteet, luokittelemaan ne oikeisiin luokkiin ja edelleen paikallistamaan ne uusista kuvista piirtämällä rajauslaatikon kohteiden ympärille. (Zhao, Zheng, Xu, & Wu X 2019, 1–2; Ravpreet & Sarbjeet 2023.) Ohjatussa oppimisessa voidaan käyttää myös muita, objektin tunnistusta läheisesti muistuttavia tekniikoita. Yleisimmät näistä ovat objektien luokittelu sanallisesti, kuvan pikseleihin perustuva semanttinen segmentointi sekä objektien yksilöllinen erottelu kuvasta. (Liu, Ouyang, Xiaogang, Fieguth, Chen, Liu & Pietikäinen 2020.)

3.2 Syväoppiminen ja neuroverkkomallit

Syväoppiminen on koneoppimisen alalaji, joka hyödyntää oppimisprosessissa neuroverkkoja (KUVIO 2). Syväoppiminen erottuu muista koneoppimisen lajeista erityisesti piirteiden oppimisella suoraan datasta ilman erillistä piirrevalintaa, joka on yleensä ollut koneoppimisprojekteissa paljon resursseja vaativa tehtävä. Kyseinen taito tekee syväoppimisesta erityisen hyödyllisen suuria tietomääriä ja aineistoja käsiteltäessä, sillä riittävän laajalla aineistolla syväoppivat mallit pystyvät tarkentamaan oppimistaan ja tuottamaan täsmällisiä tuloksia. Tämän ansiosta syväoppimista hyödynnetään monimutkaisilla aloilla, kuten puheentunnistuksessa tai videonkäsittelyssä. (Kelleher 2020, 40.)

KUVIO 2. Syväoppimisen suhde tekoälyyn ja koneoppimiseen (Kelleher 2020, 15)

Kuten aiemmin todettiin, syväoppimista hyödynnetään yleisesti suurten ja monimutkaisten tietoaineistojen käsittelyssä, sillä sen avulla voidaan luoda tarkkoja ja konkreettiseen päätöksentekoon soveltuvia neuroverkkomalleja. Neuroverkkomalli käyttää päätöksenteon tukena sille syötettyä tietoaineistoa, josta se tunnistaa ja erittelee piirteitä. Näiden piirteiden avulla se pystyy luomaan monimutkaista syötteistä luotettavia päätöksiä. Neuroverkkojen kehityksessä

on pyritty jäljittelemään ihmisaivojen rakennetta laskennallisen mallin muodossa. Neuroverkot koostuvat ihmisaivojen tapaan useista toisiinsa kytkeytyneistä neuroneista. Neuronit vastaanottavat tietoa, käsittelevät sitä ja lähettävät sen eteenpäin seuraaville neuroneille, jotka ovat järjestäytyneet eri kerroksiin neuroverkossa. Neuroni laskee painotetun summan ja käsittelee tuloksen aktivaatiofunktiolla. Tämän perusteella neuroni tekee päätöksen, mitä tietoa välitetään eteenpäin seuraaville neuroneille. Kun tieto on käsitelty koko neuroverkossa, verkko antaa lopullisen ennusteen tai luokittelun. Jos vastaus on virheellinen, neuroverkko säätää painojaan ja parantaa vastaustaan ajan myötä. (Kellher 2020, 11–15, 64–78.)

Konvoluutioneuroverkkomallit (CNN) ovat erityisesti kuvien ja visuaalisten tietojen käsittelyyn suunniteltuja neuroverkkomalleja. Konvoluutioverkkomalleja kokeiltiin ensimmäistä kertaa objektintunnistuksessa jo 1990-luvun alussa, ja 2000-luvun alussa ensimmäisiä objektintunnistukseen liittyviä tehtäviä sovellettiin eri teollisuudenaloilla. Objektintunnistuksen yleistymistä vauhditti myös runsaasti saatavilla ollut tietoaineisto esimerkiksi liikennemerkeistä, biologisista ja lääketieteellisistä kuvista sekä tekstistä kuvissa. Kasvojentunnistuksessa otettiin harppaus 2010-luvun puolivälissä, ja nykyään kasvojentunnistusta käytetään jo yleisesti esimerkiksi henkilövalvonnassa. (Lecun, Bengio & Hinton 2015.) Lääketieteen alalla konvoluutioneuroverkkoja on viime vuosina hyödynnetty esimerkiksi RNA-transkription tutkimuksessa, minkä toivotaan auttavan geneettisten mutaatioiden tunnistamisessa ja edelleen lääkehoitojen kehittämisessä (Nolan 2024).

4 YOLO-ALGORITMI

YOLO-algoritmi esiteltiin uudenlaisena objektintunnistusmallina vuonna 2015. Sen ensimmäisen version kehittivät Washingtonin yliopistossa Joseph Redmon, Santosh Divvala, Ross Girschick ja Ali Farhadi. YOLO-algoritmi tarjosi aiempiin objektintunnistukseen käytettyihin algoritmeihin verrattuna poikkeavan lähestymistavan, jossa objektintunnistusta käsitetiin regressio-ongelmana. YOLO on lyhenne englannin kielen sanoista ”You Only Look Once”, joka viittaa yksivaiheisesti suoritettavaan objektintunnistukseen. Tässä menetelmässä kohteiden sijainnit ja luokat ennustetaan suoraan tiheän ankkuriverkon avulla. (Redmon, Divvala, Girshick & Farhadi 2016.) Yksivaiheinen tunnistus on nopeutensa ansiosta parhaimmillaan reaaliaikaisissa objektintunnistuksissa, joissa on rajallinen laskentateho ja resurssit (Kundu 2023).

Toinen objektintunnistukseen käytettävien algoritmien kategoria on kaksivaiheinen, siinä algoritmi määrittää ensimmäisessä vaiheessa mahdollisia kohteita sisältävät kiinnostavat alueet (engl. Region of Interest, RoI). Kaksivaiheisen objektintunnistuksen toisessa vaiheessa konvoluutioneuroverkko määrittää kiinnostavien alueiden sisällä olevat kohteet sekä säätää rajauslaatikot tarkemmiksi. Yksivaiheisten algoritmien, kuten YOLOn, etuna on nopeus, kun taas kaksivaiheiset algoritmit ovat tarkempia varsinkin monimutkaisissa, pieniä kohteita sisältävissä kuvissa. (Du, Zhang & Wang 2020.)

4.1 YOLO-algoritmin historia

YOLO-algoritmi on noussut nopeasti suosituksi objektintunnistuksen alalla, ja siitä on kehitetty useita paranneltuja versioita. Ensimmäiset kolme versiota (YOLOv1, YOLOv2 ja YOLOv3) kehittivät alkuperäisen idean luoneet Redmon ja Farhad. YOLO-algoritmia päivitettiin edelleen nopeammaksi ja tarkemmaksi käyttämällä v2-versiossa Darknet19-neuroverkkoa ja v3-versiossa vastaavasti Darknet53-neuroverkkoa. Parannusten ansiosta kohteet pystyttiin rajaamaan entistä paremmin ja pienet kohteet tunnistamaan tarkemmin. (Terven & Cordova-Esparza 2023.)

Vuonna 2020 täysin uusi kehittäjätiimi julkaisi YOLOv4-mallin, joka poikkesi edeltäjistään monella tavalla. Lähes ainoana yhteisenä tekijänä aiempiin malleihin verrattuna säilyi Darknet-

arkkitehtuuri, mutta uusi versio säilytti kuitenkin YOLOn filosofian yksivaiheisena ja reaaliaikaisena, avoimen lähteen algoritmina. YOLOv4 käytti CSPDarknet53-neuroverkkoa, jossa laskenta oli jaettu kahteen osaan. Tämän ansiosta laskentateho väheni, mutta tarkkuus säilyi edelleen samana. YOLOv4-versioon oli myös lisätty edistyksellisiä optimointitekniikoita, kuten mosaiikkisekoitus (*engl. mosaic augmentation).* Mosaiikkisekoituksen avulla neljä kuvaa yhdistettiin yhdeksi kuvaksi, mikä paransi mallin kykyä havaita kohteita myös epätavallisissa ympäristöissä. Kohteiden tunnistusta eri kokoluokissa taas parannettiin Spatial Pyramid Pooling (SPP)- ja Path Aggregation Network (PANet) -tekniikoiden avulla. (Bochkovskiy, Chien-Yao & Hong-Yuan 2020; Terven & Cordova-Esparza 2023.)

Ultralytics-yhtiössä oli myös kehitetty omaa YOLO-mallia, ja vain pari kuukautta YOLOv4-mallin julkistamisen jälkeen vuonna 2020 yrityksen työntekijä Glenn Jocher julkaisi YOLOv5-mallin. Tässä mallissa oli hyvin paljon samoja ominaisuuksia kuin YOLOv4-mallissa, mutta YOLOv5-mallin kehityksessä oli käytetty PyTorchia Darknetin sijaan. PyTorchin käyttö helpotti mallin käyttöönottoa ja koulutusta. Muistinhallinta ja tehokkaampi GPU-käyttö tekivät YOLOv5-mallista nopeamman kuin YOLOv4, ja näiden ominaisuuksien ansiosta YOLOv5 saavutti johtoaseman mallien välillä. (Terven & Cordova-Esparza 2023.) Vuonna 2022 julkaistut YOLOv6 ja YOLOv7 eivät tuoneet suuria uudistuksia malleihin, mutta niiden avulla YOLO saatiin optimoitua entistä kevyemmäksi, nopeammaksi, tehokkaammaksi ja tarkemmaksi. YOLOv6 on optimoitu teollisuuden sovellusten käyttöön, ja se soveltuu erityisesti nopeaan ja kevyempään objektitunnistukseen esimerkiksi mobiililaitteissa ja reunalaskennassa (Li, Li, Jiang, Weng, Geng, Li, Ke, Li, Cheng, Nie, Li, Zhang, Liang, Zhou, Xu, Chu, Wei & Wei 2022). YOLOv7 julkaistiin nimestään huolimatta hieman ennen YOLOv6-mallia, ja sen kehittäjinä toimivat samat tekijät kuin YOLOv4-mallissa. YOLOv7 sisälsi Extended Efficient Layer Aggregation Networks (E-ELAN) -rakenteen, joka helpotti pienikokoisten kohteiden havaitsemista. Tämä malli soveltuu hyvin esimerkiksi reaaliaikaiseen videoseurantaan. (Terven & Cordova-Esparza 2023.)

Vuonna 2023 Ultralytics julkaisi YOLOv8-mallin, joka oli rakennettu yhtiön aiemmin kehittämän YOLOv5-mallin päälle. Kehitys oli toteutettu vain vähäisillä muutoksilla, kuten lisäämällä parempia virhefunktioita parantamaan pienempien objektien tunnistusta ja käyttämällä ankkurivapaata mekanismia yksinkertaistamaan mallin koulutusta. Ankkurivapaa mekanismi vapautti mallin ennalta määriteltyjen ankkurilaatikoiden käyttämiseltä, koska YOLOv8 ennustaa kohteiden sijainnin kohdepisteiden avulla. Lisäksi optimointitekniikoita oli paranneltu ja malli

käyttää mosaiikkisekoitusta myös harjoitusvaiheen aikana lukuun ottamatta kymmentä viimeistä koulutusjaksoa (*engl. epoch*). YOLOv8-mallin annotaatiot tallennetaan tekstitiedostoon, joka sisältää jokaisen annotaation rajauslaatikon koordinaatit kuvassa. Lisäksi tietoaineistossa on YAML-tiedosto, jossa on määritelty mallin arkkitehtuuri ja kohteiden luokat. YOLOv8-mallista on tehty useita vertailuanalyyseja, ja sen tarkkuus sekä tehokkuus olivat objektintunnistukseen käytettävien mallien parhaimmistoa julkaisuvuonna 2023. (Yaseen 2024; Terven & Cordova-Esparza 2023.)

Vaikka YOLOv8-mallin suorituskyky objektintunnistuksessa oli huipputasoa, mallien kehitys on jatkunut edelleen sen julkaisun jälkeen. Vuonna 2024 esiteltiin YOLOv9-, YOLOv10- ja YOLOv11-mallit ja tuorein julkaisu kuluvalta vuodelta 2025 on YOLOv12-malli. YOLOv9-malli käyttää uusia tekniikoita Programmable Gradient Information (PGI) ja Generalize Efficient Layer Aggregation Network (GELAN) vähentämään tietohävikkiä ja varmistamaan paremmat painopäivitykset mallille syötetylle tiedolle. Malli pystyy hyödyntämään parametreja edeltäjiään huomattavasti tehokkaammin ja säilyttämään tärkeitä tietoja objektientunnistuksessa. Tämä tekee YOLOv9-mallista kevyemmän ja nopeamman ilman tarkkuuden heikentymistä. (Wang, Yeh & Liao 2024; YOLOv9: A Leap Forward in Object Detection Technology.)

Tsinghuan yliopiston tutkijat kehittivät YOLOv10-mallin Ultralyticsin Python-paketille. Tässä mallissa suurin uudistus oli Non-Maximum Suppression (NMS) -ominaisuuden poistaminen. NMS-ominaisuutta käytettiin kaikissa aiemmissa YOLO-malleissa, ja sen avulla oli suodatettu päällekkäisiä ennusteita. NMS-ominaisuuden poistaminen pienensi laskennallista viivettä, ja sen sijaan YOLOv10-malli hyödyntää kaksoisluokituspäätä (*engl. dual label assignment*) ja johdonmukaista vastaavuusmittaria (*engl. Consistent Matching Metric*), joiden avulla varmistetaan tiheä opetus ja yksiselitteiset, yhtenäiset ennustukset sekä parannetaan tarkkuutta ja vähennetään viivettä. Tehokkuutta ja tarkkuutta parannettiin myös eri komponenttien optimoinnilla ja parannelluilla ominaisuuksilla, kuten suurydinkonvoluutiot ja sisäiset tarkkailumoduulit. (Wang, Chen, Liu, Chen, Lin, Han & Ding 2024; YOLOv10: Real-Time End-to-End Object Detection.)

YOLOv11-mallin kehittäjänä toimi jälleen Ultralytics, joka oli tehnyt merkittäviä parannuksia edellisiin YOLO-malleihin verrattuna. YOLOv11-malli on suunniteltu käytettäväksi laaja-alaisesti tietokonenäköä vaativissa tehtävissä, kuten luokittelussa, objektintunnistuksessa, asennon arvioimisessa, instanssisegmentoinnissa ja suuntautuneessa objektintunnistuksessa

(OBB, Oriented Object Detection). Mallin runko- ja kaula-arkkitehtuuria on paranneltu kohteiden havaitsemisen tarkentamiseksi, ja siihen on lisätty myös optimoituja koulutusputkia (*engl. training pipelines*) nopeamman käsittelynopeuden saavuttamiseksi. YOLOv11 voidaan myös ottaa käyttöön monissa erilaisissa ympäristöissä, kuten pilvialustoilla ja NVIDIA:n näytönohjaimia tukevissa järjestelmissä. (Khanam & Hussain 2024.)

Tällä hetkellä viimeisin malli YOLO-sarjassa on YOLOv12, joka julkaistiin helmikuussa 2025. Mallin keskeisimpiä parannuksia ovat aluehuomiointi, joka käsittelee tehokkaasti laajoja havaintokenttiä ja vähentää laskentakustannuksia, sekä Residual Efficient Layer Aggregration Networks (R-ELAN), joka parantaa piirteiden yhdistämistä ja vakauttaa mallin koulutusta. Arkkitehtuuria on tehostettu vähentämällä parametreja sekä hyödyntämällä FlashAttention-tekniikkaa muistinkäsittelyn tehostamisessa. YOLOv12 on vertailussa osoittautunut tähän asti julkaistuista YOLO-malleista tarkimmaksi ja nopeimmaksi. (Tian, Ye & Doermann 2025.) YOLO-mallien kehitys tulee todennäköisesti jatkumaan myös lähitulevaisuudessa objektintunnistuksen ollessa kasvava ala esimerkiksi robotiikassa.

4.2 YOLO-algoritmin sovellukset ja käyttökohteet

YOLO-algoritmia hyödynnetään laajasti eri teollisuudenaloilla. Ensimmäisenä esimerkkinä voidaan nostaa esiin teollisuuden tuotantolinjat, joissa YOLO-algoritmia käytetään muun muassa virheentunnistukseen, laadunvalvontaan, kappaleiden tunnistukseen ja paikannukseen sekä teollisuusympäristön valvontaan, kuten tulipalojen havaitsemiseen (Kang, Hu, Liu, Zhang & Cao 2025). YOLO-algoritmin nopea prosessointikyky ja tehokas objektintunnistus tekevät siitä ihanteellisen työkalun tuotantolinjoille, joissa tarvitaan nopeita ja tarkkoja ratkaisuja. Mukautuminen erilaisiin tehtäviin mahdollistaa algoritmin laaja-alaisen käytön elintarviketeollisuudesta rakennusteollisuuteen. Tuotantolinjojen lisäksi myös varasto- ja logistiikkapalvelut tuotantolaitoksissa ovat automatisoineet tehtäviään, esimerkiksi inventaariota ja laitteiston valvontaa, YOLO-algoritmin avulla. (Ali & Zhang 2024.)

Lääketieteessä YOLO-algoritmia käytetään erityisesti lääketieteellisten kuvien analysoinnissa. Analysointiin käytetään monia eri tekniikoita, kuten objektintunnistusta, luokittelua, segmentaatiota ja paikannusta. (Ali & Zhang 2024.) Esimerkkinä voidaan mainita vuoden 2023 tutkimus mammografiakuvien analysoinnista, jossa YOLOv5-mallin avulla tunnistettiin kuvista rintasyövän esiasteita 80–90 prosentin tarkkuudella (Coskun, Karaboga, Basturk,

Akay, Nalbantoglu, Dogan, Pacal & Karagöz 2023). Objektintunnistusta käytettiin myös COVID-19-epidemian aikana kasvosuojien ja muiden suojavarusteiden käytön valvonnassa sairaaloissa ja hoitolaitoksissa (Vibhuti, Jindal, Singh & Rana 2022). YOLOv8-mallia käytetään osana segmentointia useilla lääketieteen aloilla, kuten röntgenkuvauksissa, ultraäänitutkimuksissa ja magneettikuvauksissa (Ali & Zhang 2024).

Automatisoidut autot, eli robottiautot, ovat yksi autoalan nopeimmin kehittyvistä tutkimus- ja kehitysalueista. Itseohjautuva ajaminen edellyttää tarkkaa objektien tunnistusta, luokittelua ja kykyä tehdä nopeita ratkaisuja reaaliaikaisesti. YOLO-algoritmin erinomainen suorituskyky, erityisesti nopeuden ja tehokkuuden osalta, tekee siitä ihanteellisen tällaisiin sovelluksiin. Lisäksi mallien kehitys entistä pienempien objektien tunnistamisessa edistää ajoneuvojen turvallisuutta ja luotettavuutta. (Ali & Zhan 2024.) Myös perinteisissä, kuljettajan hallinnassa olevissa autoissa YOLO-algoritmilla on useita erityisesti turvallisuuteen liittyviä käyttökohteita. Esimerkiksi liikennemerkkien tunnistus, kaistavahti, hätäjarrutusjärjestelmät, mukautuvat vakionopeudensäätimet ja virevahdit ovat automatisoituja järjestelmiä, joiden reaaliaikaiseen valvontaan YOLO-algoritmin ominaisuudet soveltuvat erinomaisesti. (Gheorghe, Duguleana, Boboc & Postelnicu 2024.)

YOLO-algoritmia sovelletaan myös maataloudessa, erityisesti viljelykasvien terveyden seurannassa. Reaaliaikainen objektintunnistus mahdollistaa tuholaisten ja tautien havaitsemisen sekä rikkakasvien erottamisen viljelykasvien joukosta. YOLO-algoritmin avulla voidaan siis lisätä tuottavuutta ja vähentää torjunta-aineiden tarvetta viljelyssä. Lisäksi algoritmia käytetään karjanhoidossa ja ympäristön tarkkailussa, mikä auttaa hallitsemaan resursseja ja tehostamaan maatilojen toimintaa. YOLO-algoritmin mahdollisuudet eivät kuitenkaan rajoitu vain maatalouteen ja muihin edellä mainittuihin käyttökohteisiin, vaan sitä hyödynnetään laajasti eri teollisuudenaloilla. Esimerkiksi turvallisuusalalla sen objektintunnistusta käytetään valvontakameroissa (Ali & Zhan 2024). YOLO-algoritmin sovellusalueet ovat siis erittäin monipuoliset, ja mallien kehittyessä uusia käyttökohteita syntyy varmasti lisää.

5 KUVA-AINEISTON ANNOTOINTIPROSESSI

Kuva-aineiston annotointiprosessissa ensimmäinen vaihe on kuvamateriaalin kerääminen. Syväoppimismallin tehokas harjoittaminen edellyttää laadukasta ja täsmällistä aineistoa, joten tämä vaihe on kriittinen koko prosessin onnistumisen kannalta. Annotointiprosessin muita vaiheita ovat annotointi, jälkikäsittely, laadun arviointi ja tietoaineiston tallennus. Kuvamateriaalina voidaan hyödyntää jo olemassa olevia, valmiiksi merkittyjä tietoaineistoja, tai muodostaa oma tietoaineisto kerätystä materiaalista. Jos itse kerätty aineisto ei ole tasapainoinen tai muuten riittävä, sitä voidaan täydentää valmiiksi merkityillä tietoaineistoilla tai muokatuilla kuvilla. Syväoppimismallin tarkkuuden kannalta luokkien riittävä määrä annotoinnissa on erittäin tärkeää, ja kuva-aineistoa koostettaessa tulee varmistua kaikkien luokkien tasapuolisesta esiintymisestä. (Sager, Janiesch & Zschech 2021.)

5.1 Annotointimenetelmät

Kuvien annotointi, eli huomautuksien lisääminen kuviin ja niiden luokittelu, on prosessin työläin ja haastavin osuus. Annotointi voidaan tehdä täysin käsityönä, puoliautomaattisesti tai kokonaan automaattisesti. Jos annotointi tehdään kokonaan käsin, toinen käyttäjä voi tarkistaa annotoidut kuvat korkean laadun varmistamiseksi. Ennen annotoinnin aloitusta laaditaan prosessiin ohjeet, joissa määritellään luokat ja annetaan tarpeen vaatiessa yksityiskohtaista tietoa annotoinnista. (Sager ym. 2021.) Annotoinnin laadun arvioimisessa voidaan käyttää esimerkiksi ACC-menetelmää, jossa laatu mitataan neljän päämittarin avulla. Nämä mittarit ovat 1. tarkkuus, eli miten hyvin annotoinnit vastaavat oikeaa tasoa; 2. luotettavuus, eli kuinka todennäköisesti annotointi on oikea; 3. pitkäaikainen yhdenmukaisuus, eli pysyvätkö annotoinnit samoina ajan myötä, ja 4. välitön yhdenmukaisuus, eli miten yhdenmukaiset näkemykset usealla annotoijalla on saman objektin annotoinnista. (Lavitas, Redfield, Lee, Fletcher, Eck & Janardhanan 2021.)

Puoliautomaattisessa annotoinnissa osa prosessista automatisoidaan, mutta ihminen säilyttää oikeuden vahvistaa ja tarvittaessa korjata automaation ehdotukset. Tässä menetelmässä konenäköalgoritmi tunnistaa mahdolliset objektit ja niiden luokat ja välittää nämä annotaatiot ihmisen tarkistettavaksi. Tällä tavalla annotoinnin tarkkuus pysyy korkeana, mutta prosessia

voidaan nopeuttaa huomattavasti. (Sager ym. 2021.) Puoliautomaattista annotointia käytetään erityisesti lääketieteellisessä kuvantamisessa, jossa suurten kuvamäärien merkitseminen edellyttää asiantuntijatietoa sairauksista ja ihmiskehon rakenteista. Sen avulla voidaan vähentää työhön tarvittavien asiantuntijoiden määrää, kohdentaa resursseja tehokkaammin ja parantaa annotoinnin objektiivisuutta. (Hailiang, Bin, Yu, Weiwei, Yijun, Jiacheng & Linfeng 2020.)

Täysin automaattinen annotointi perustuu valmiiksi määriteltyyn luokkamalliin, ja siinä hyödynnetään yleisesti klusterointia eli ryhmittelyä. Tämä tarkoittaa sitä, että algoritmi ryhmittelee samankaltaiset kuvat tai niissä näkyvät kohteet ennen annotointia. Vaihtoehtoisesti voidaan käyttää esimerkkejä annotoinnista pienellä aineistolla. (Sager ym. 2021.) Automaattisen annotoinnin apuna voidaan käyttää esimerkiksi Segment Anything Model (SAM) -tekoälymallia, joka tunnistaa ja segmentoi kohteet kuvista. SAM suoriutuu kohtuullisen hyvin tunnistamisesta selkeissä kuvissa, mutta se saattaa ohittaa monimutkaisia tai epäselviä kohteita ja rajata kohteita epätarkasti. (Kirillov, Mintun, Ravi, Mao, Rolland, Gustafson, Xiao, Whitehead, Berg, Lo, Dollár & Girshick 2023.) SAM-malli on toimintaperiaatteeltaan erilainen kuin YOLO-malli, ja se suoriutuu paremmin joustavissa, hieman raskaammissa segmentointitehtävissä, kun taas YOLO-malli on ihanteellinen nopeaa, kevyttä ja laskennallisesti tehokasta segmentointia vaativissa tehtävissä (Segment Anything Model (SAM)). YOLO- ja SAM-malleja voidaan käyttää rinnakkain automaattisessa annotoinnissa, sillä objektintunnistus voidaan suorittaa YOLO:n avulla ja kohteiden tarkempi segmentointi SAM-mallilla (Zhu, Yang & Xu 2024).

5.2 Kuva-aineiston annotointitekniikat

Kuva-aineistoa voidaan annotoida useilla eri tekniikoilla, ja oikean tekniikan valinta riippuu aineistosta, tavoitellusta lopputuloksesta sekä halutusta tarkkuudesta. Myös annotointiin käytettävä työkalu vaikuttaa valintaan, sillä työkalut ovat usein erikoistuneet tiettyihin tekniikoihin.

Rajauslaatikko on kuva-aineiston annotoinnissa käytettävä tehokas ja yksinkertainen perustekniikka. Kuten jo aiemmin mainittiin, suorakaiteen muotoinen rajauslaatikko piirretään kohteen ympärille kuvassa. Rajauslaatikolle määritetään koordinaatit sijainnin mukaan, ja usein käytetään ylävasemman kulman sekä alaoikean kulman koordinaatteja muodossa (x1, y1,

x2, xy2). Rajauslaatikon koordinaatit voidaan myös määrittää ylävasemman kulman koordinaattien ja laatikon leveyden sekä korkeuden mukaan muodossa (x1, y1, leveys, korkeus). Rajauslaatikoiden etuna on annotoinnin helppous, koska laatikoita on helppo piirtää myös manuaalisesti. Lisäksi niiden sijainti pystytään määrittämään tarkasti koordinaattien avulla. Varsinkin epäsäännöllisen muotoisia kohteita merkittäessä rajauslaatikoiden sisään voi kuitenkin kuulua myös suuri määrä kuvapikseleitä, jotka eivät liity kohteeseen. Tämä lisää rajauksen epätarkkuutta. Myös epäsäännöllisissä asennoissa olevien kohteiden rajaus on hankalaa suorakaiteen muotoisilla laatikoilla. (Kumari 2023.)

Monikulmio (*engl. polygon*) on rajauslaatikkoa tarkempi tekniikka kuva-aineiston kohteiden annotoinnissa. Monikulmion voi piirtää tarkasti kohteen ympärille niin, että sen sisään jää mahdollisimman vähän ylimääräisiä kuvapikseleitä. Tätä tekniikkaa käytetään silloin, kun tarvitaan kohteen yksityiskohtaista erottelua, ja se soveltuu myös loistavasti epäsäännöllisen muotoisten kohteiden rajaamiseen. Nämä ominaisuudet tekevät monikulmioista tehokkaan tekniikan annotoinnissa. (Kumari 2023.) Monikulmioiden piirtäminen manuaalisesti on hidasta varsinkin suuria aineistoja annotoitaessa, joten tätä tekniikkaa voi soveltaa esimerkiksi osana puoliautomaatista annotointia (Fang, Zhu, Zhou, Liu & Yao 2021).

Pikselitason segmentoinnissa (*engl. semantic segmentation*) jokaiselle kuvapikselille annetaan oma merkintä sekä luokitus. Käytännössä kuvasta luodaan matriisitaulukko, joka vastaa alkuperäistä kuvaa korkeudeltaan ja leveydeltään. Matriisitaulukon jokainen solu sisältää tiedon siitä, mihin luokkaan kyseinen pikseli kuuluu. (Jain 2020.) Segmentoinnin tuloksena syntyy esimerkiksi PNG-muotoinen maskikuva, jossa luokat on esitetty eri väreillä. Pikselitason segmentoinnin tarkkuus on todella korkea, mutta laskentateholtaan tämä tekniikka on raskas. Tätä annotointitekniikkaa käytetään suurta tarkkuutta ja yksityiskohtaisuutta vaativissa sovelluksissa, kuten automatisoiduissa ajoneuvoissa tai lääketieteellisessä kuvantamisissa. (Kumari 2023; Jain 2020.)

Muita annotointitekniikoita ovat esimerkiksi instanssisegmentointi, avainpisteannotointi ja 3D-laatikkoannotointi. Instanssisegmentointi yhdistää objektintunnistuksen ja pikselitason segmentoinnin, ja sen avulla voidaan erottaa segmenttiluokkien erilliset esiintymät. Tämän annotointitekniikan viimeisimpiä käyttökohteita ovat esimerkiksi satelliittikuvaus ja vähittäiskaupan kuva-aineiston annotointi. (Bandyopadhyay 2022.) Avainpisteannotoinnissa kuvaan merkitään kuvan keskeiset pisteet, esimerkiksi kasvonpiirteet, nivelkohdat tai esineiden kulmat.

Tätä tekniikkaa käytetään haastavimmissa annotointitehtävissä, kuten kasvontunnistuksessa ja liikeseurannassa, joissa objektit täytyy hahmottaa yksityiskohtaisesti. (Rajnerowicz 2023.) 3D-laatikkoannotointia käytetään esimerkiksi robotiikassa ja lisätyn todellisuuden sekä virtuaalitodellisuuden sovelluksissa. Tässä tekniikassa kuvan kohteet merkitään kolmiulotteisilla laatikoilla, jolloin on mahdollista ottaa huomioon myös syvyys. (Kumari 2023.)

5.3 Kuva-aineiston annotointityökalut

Kuva-aineiston annotointiin on tarjolla useita erilaisia työkaluja. Työkalun valintaan vaikuttavia tekijöitä ovat esimerkiksi käytettävä annotointitekniikka, kuva-aineiston koko ja laatu sekä tuetut tiedostomuodot. Alla on esitelty muutama vaihtoehto kuva-aineiston annotointiominaisuuksia painottaen:

CVAT on avoimen lähdekoodin selainpohjainen sovellus, jota voidaan käyttää monipuolisesti kuva- ja videoaineiston annotointiin, kuten objektintunnistukseen ja 3D-laatikointiin. Ohjelmasta on tarjolla rajoitettu ilmainen versio, joka tukee puoliksi automaattisia annotointityökaluja. Maksullisessa versiossa on saatavilla enemmän tallennustilaa, mahdollisuus tehdä useita projekteja, usean käyttäjän tuki ja paremmat turvallisuusominaisuudet. (CVAT.)

Label Studio on avoimen lähdekoodin annotointisovellus, joka asennetaan käyttäjän tietokoneelle. Ilmainen versio on toiminnoiltaan rajattu, eikä siinä ole esimerkiksi automaattista annotointia. Maksullinen versio sisältää esimerkiksi automaattisen annotoinnin sekä usean käyttäjän tuen. Label Studio -sovellusta voidaan käyttää kuva-aineiston luokitteluun, objektintunnistukseen ja pikselitason segmentointiin. (Label Studio.)

Roboflow on selainpohjainen annotointityökalu, jossa dataa hallitaan pilvipalvelussa. Annotointia voi automatisoida esimerkiksi Label AI- ja Box Prompting -ominaisuuksien avulla. Roboflow soveltuu objektintunnistukseen, kuva-aineiston luokitteluun ja segmentointiin. Sovellus tukee tiimityöskentelyä myös ilmaisessa versiossa. Maksullisissa versioissa tarjotaan pääsyy laajempaan mallin esikoulutukseen ja suorituskyvyn arviointiin. Ilmaisversiossa mallit ja aineisto ovat julkisia, kun taas maksullisessa versiossa ne voidaan pitää yksityisinä. (RoboFlow.)

SuperAnnotate on monipuolinen annotointialusta, joka soveltuu myös kuva- ja videoaineiston annotointiin. Sovellusta voi käyttää kuva-aineiston objektintunnistukseen, segmentaatioon ja luokitteluun. Ilmainen versio ei tue automaattista annotointia, mutta maksullisessa versiossa automaatiotyökalut, kuten SAM, ovat saatavilla. SuperAnnotate soveltuu hyvin ryhmätyöskentelyyn, sillä myös ilmaisessa versiossa on usean käyttäjän tuki. (SuperAnnotate.)

Kuten lyhyestä annotointityökalujen esittelystä käy ilmi, eri vaihtoehdot tarjoavat melko samankaltaisia perustoimintoja. Suurimmat erot liittyvät ilmaisten ja maksullisten versioiden ominaisuuksiin, kuten automaattiseen annotointiin, yhteistyömahdollisuuksiin ja tietoturvaan, Kaikki tarkastellut työkalut tukevat yleisimpiä annotointiformaatteja, joita ovat esimerkiksi COCO, Pascal VOC ja YOLO. (CVAT; LabelStudio; Roboflow; SuperAnnotate.)

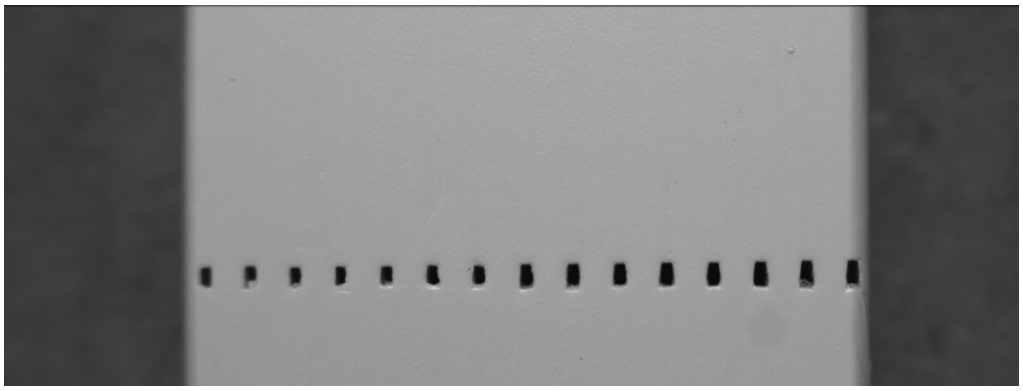
6 ANNOTOINTIPROSESSIN TOTEUTUS

Annotointiprosessin toteutuksen kuvauksessa käytän esimerkkinä Centria TKI:n Optiwood-projektissa mukana olevaa, puuteollisuudessa toimivaa yritys A:ta ja yrityksen materiaaleista saatua dataa. Annotointiprosessin tarkoituksena oli kerätä tietoaineisto pintakäsiteltyjen puumateriaalien virheistä ja auttaa yritystä kehittämään sen avulla virheentunnistusjärjestelmä tuotantolinjalle. Yritys A:n puumateriaalit pintakäsitellään maalilla, ja toisinaan maalipintaan syntyy ihmissilmällä hankalasti havaittavia virheitä. Optiwood-tiimi pyysi yritystä keräämään ja toimittamaan virheitä sisältäviä, pintakäsiteltyjä materiaaleja kuvattavaksi annotointiprosessia varten.

6.1 Virheluokkien määritys

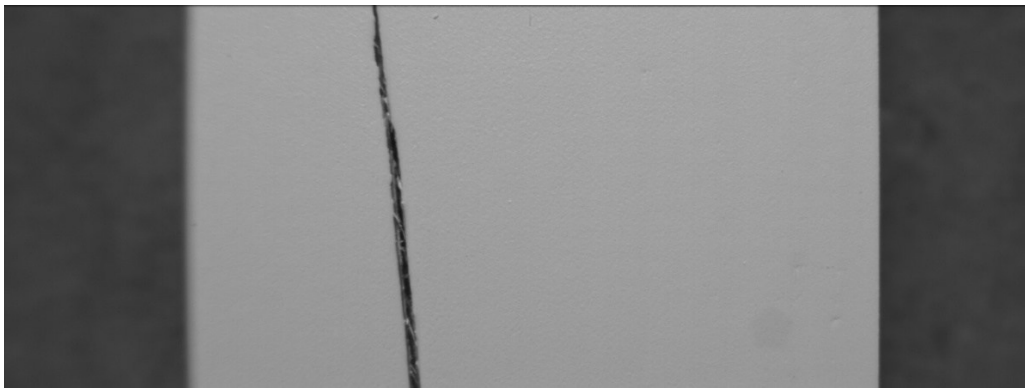
Ensimmäinen vaihe ennen datan fyysistä keräämistä oli virheluokkien määritteleminen. Yritys A:lta saatujen tietojen ja materiaalien tutkimisen jälkeen päädyimme tiimin kanssa muodostamaan neljä luokkaa: sormijatkos (*engl. fingerjoint*), halkeama (*engl. crack*), höylän repimä pinta (*engl. planing torn*) ja naarmu (*engl. scratch*).

Sormijatkos syntyy, kun kaksi puukappaletta liitetään yhteen (KUVA 1). Näissä kohdissa maali saattaa levittyä epätasaisesti muodostaen virheen pintamateriaaliin.



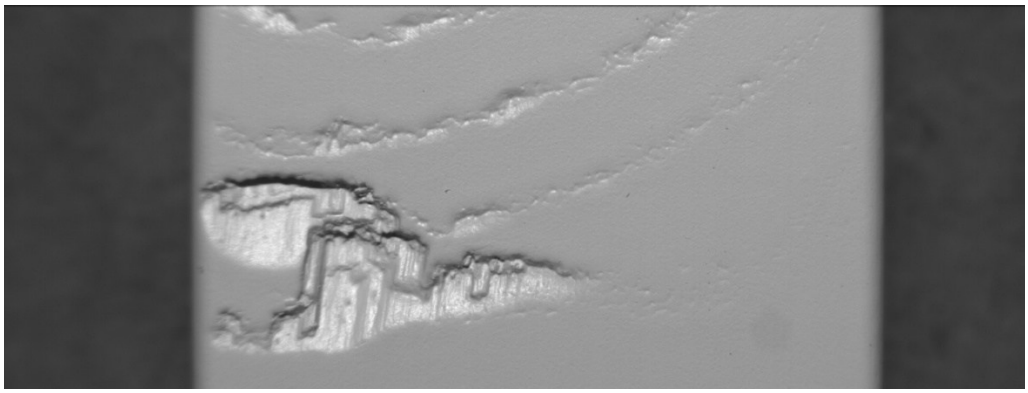
KUVA 1. Sormijatkos maalipinnassa

Halkeama on puun kuivumisvaiheessa ilmaantuva virhe, joka voi myös johtaa maalin epätasaiseen levittymiseen. (KUVA 2.)



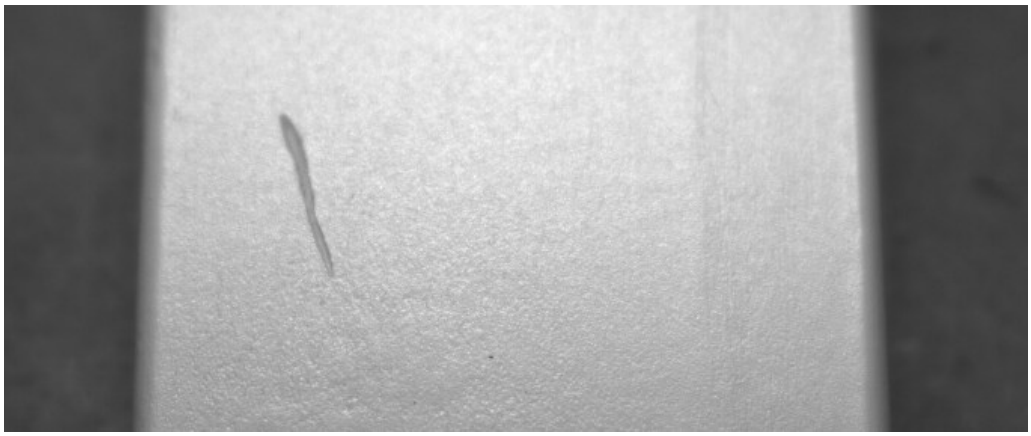
KUVA 2. Halkeama maalipinnassa

Toisinaan höylä osuu puun pintaan ennen maalausta, tämä aiheuttaa maalipintaan repimäjäljen. (KUVA 3.)



KUVA 3. Höylän repimä maalipinta

Naarmu puolestaan on pienempi, pinnallinen virhe maalipinnassa. (KUVA 4.)



KUVA 4. Naarmu maalipinnassa

Kuvien järjestämisen helpottamiseksi kuvaamisvaiheessa päätimme yhdessä tiimin kanssa keskittyä yhteen virheluokkaan kerrallaan. Kuvat tallennettiin virheluokan mukaan nimettyihin kansioihin, mikä helpotti virheiden tunnistamista annotointivaiheessa. Yksittäisessä kuvassa saattoi kuitenkin esiintyä myös muiden luokkien virheitä, jotka huomioin myöhemmin annotoinnin yhteydessä.

6.2 Kuvadatan kerääminen

Seuraava vaihe kuvadatan annotointiprosessissa oli datan kerääminen, jonka suoritimme yhdessä tiimin kanssa. Keräämisessä käytettiin Sick Ranger 2D -erikoiskameraa ja tehokasta valaistusta, joiden avulla virheet saatiin näkyviin selkeästi kuvissa. Materiaalien virheet olivat melko pieniä, joten kuvausalue merkittiin tarkasti oikean kohdistuksen varmistamiseksi. Asetimme materiaalit kuvausalueelle, ja suoritimme kuvauksen manuaalisesti kameran laukaisupainikkeella. Kuvaus suoritettiin kameran etäisyyttä kohteesta vaihdellen, jotta pystyimme simuloimaan realistista tilannetta tuotantolinjalla, ja samalla onnistuimme saavuttamaan parhaan mahdollisen kuvanlaadun. Kameran rajoitusten vuoksi otimme kuvat harmaasävyisinä, sillä värikuvien tarkkuus oli huomattavasti heikompi. Tallensimme kuvat korkean resoluution varmistamiseksi PNG-muodossa.

Otimme virheitä sisältäviä kuvia yhteensä 1379 kappaletta ja virheettömiä kuvia saman verran. Tarkoituksemme oli ottaa jokaisesta luokasta tasapuolisesi kuvia, mutta sormijatkos ja halkeama olivat yleisimpiä virheitä, ja niiden osuus kokonaismäärästä on sen vuoksi suurempi. Koska yhdessä kuvassa saattoi olla useita virheitä, oli yksittäisten virheiden lukumäärän laskeminen vielä tässä vaiheessa haastavaa.

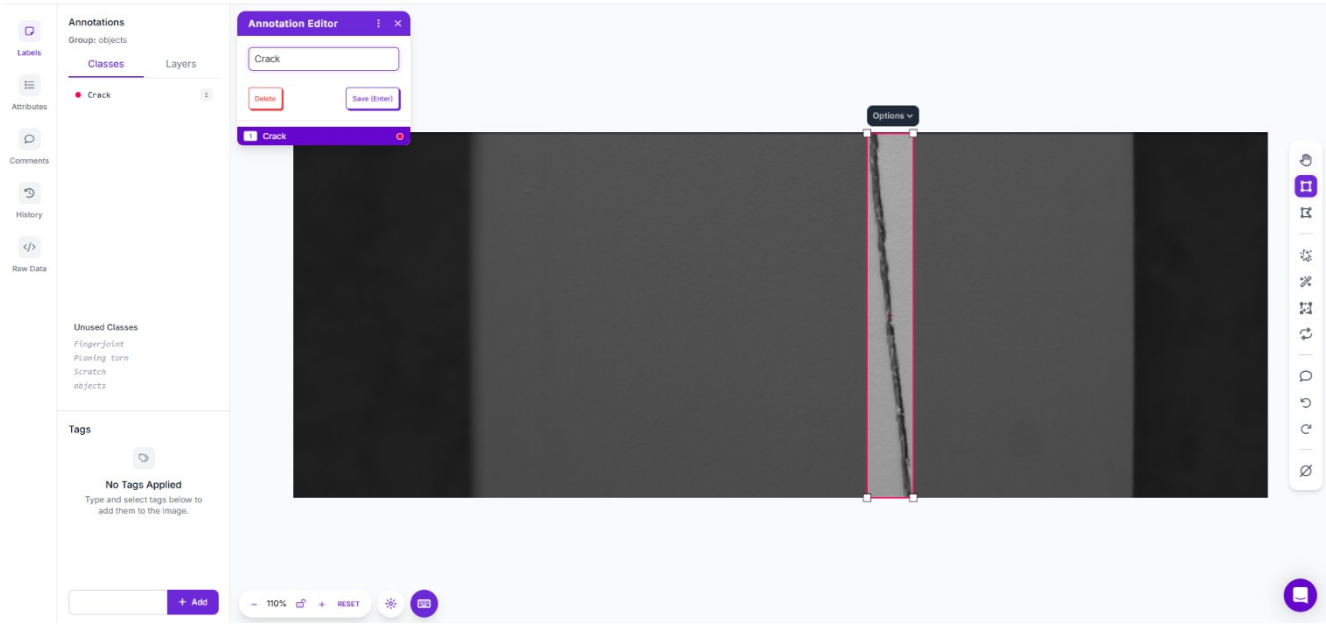
6.3 Annotointityökalun valinta

Annotointityökalun valinnan suoritin itsenäisesti, ja painotin valinnassa erityisesti yhteensopivuutta YOLO-algoritmin kanssa, sillä kyseistä algoritmia oli tarkoitus kouluttaa myöhemmin virheiden tunnistamiseen kuvista. Työkalun tuli olla helppokäyttöinen ja soveltua useiden projektien sekä käyttäjien samanaikaiseen käyttöön. Tallennustilaa vaadittiin riittävästi tuhansien PNG-tiedostojen lisäämiseen, ja annotoinnissa piti käyttää YOLO-algoritmin mukaista rajauslaatikkoa virheiden merkitsemiseen. Näitä kriteerejä arvioidessani ja vaihtoehtoja kartoittaes-

sani valintani kohdistui Roboflow-annotointityökaluun. Roboflow sisältää kaikki vaaditut ominaisuudet, ja lisäksi YOLOv8- ja YOLO11-algoritmien kehittäjä Ultralytics tekee tiivistä yhteistyötä Roboflow'n kanssa (Ladi 2024). Tämän annotointityökalun muina etuina voidaan pitää puoliautomaattista ja automaattista annotointia sekä mahdollisuutta esikouluttaa tietoaineistoa Roboflow Train -ominaisuuden avulla.

6.4 Kuvien annotointi

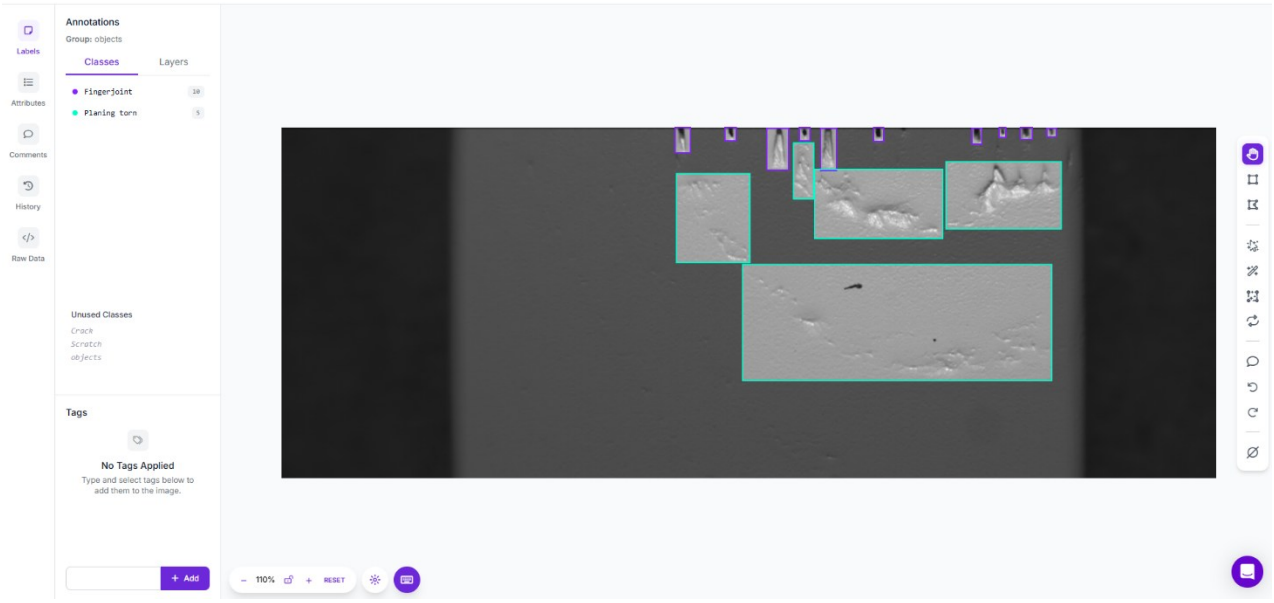
Datan keräämisen, luokkien määrittelyn ja työkalun valinnan jälkeen oli ajankohtaista aloittaa kuvien annotointi. Tämä prosessi oli minun vastuullani ja suoritin sen itsenäisesti. Latasin kuvat sisältävät kansiot Roboflow-alustalle. Annotoinnin aloitus oli hyvin yksinkertaista ja suoraviivaista, sillä Roboflow'n käyttöliittymästä oli helppo siirtyä virheiden merkitsemiseen kuvista. Merkitsin virheet kuviin piirtämällä rajauslaatikon virheen ympärille. (KUVA 5.) Rajauslaatikon piirtämisen yhteydessä nimesin virheelle halutun luokan tai käytin valmista luokkaa, ja Roboflow merkitsi nimetyt luokat eri väreillä piirrettyinä laatikkoina.



KUVA 5. Virheen merkitseminen rajauslaatikolla Roboflow-työkalussa

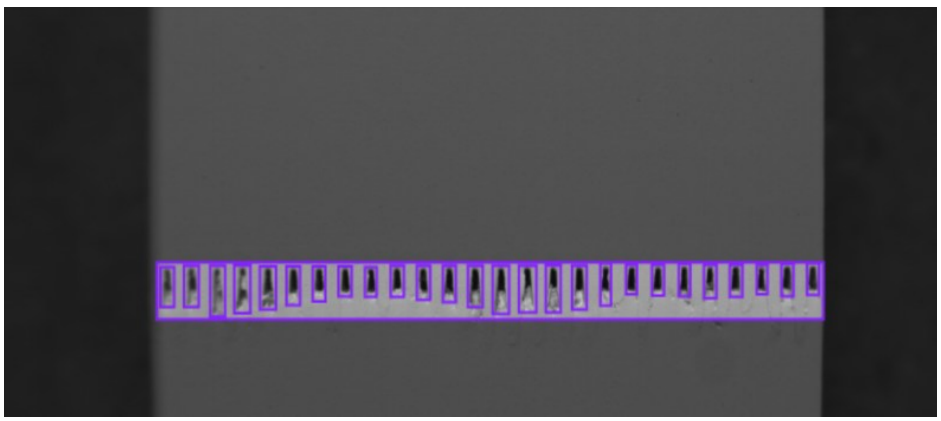
Jos kuvissa esiintyi useita saman luokan virheitä tai eri luokkiin kuuluvia virheitä, merkitsin ne kaikki rajauslaatikoilla (KUVA 6). Rajauslaatikoiden oli sallittua sijoittua limittäin, mutta yritin välttää tätä limittymistä YOLO-mallin suorituskyvyn optimoimiseksi. Koska kuvat oli otettu

erittäin tehokkaassa valaistuksessa, joka ei vastaa esimerkiksi tuotantolinjan valaistusta, jätin himmeästi näkyvät virheet merkitsemättä. Jos virhe oli epäselvä tai sen luokittelu oli hankalaa, poistin kuvan kokonaan aineistosta. Samoin menettelin, jos kuva oli epätarkka.

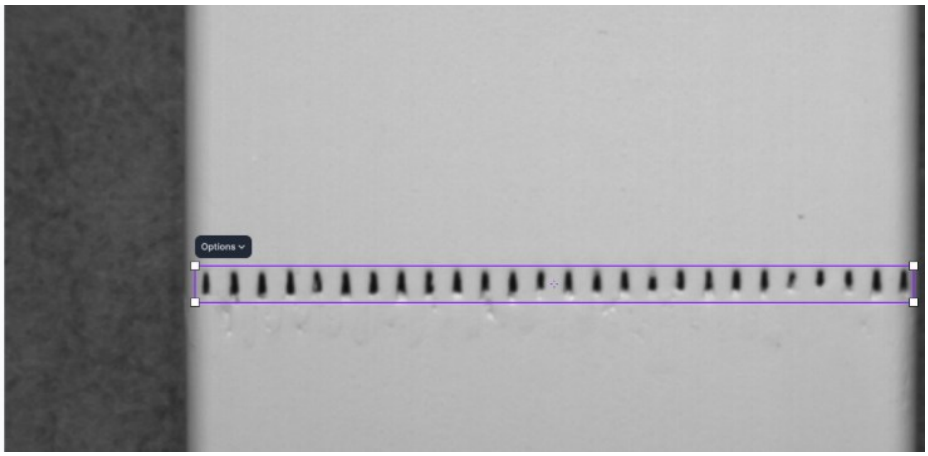


KUVA 6. Kahteen eri luokkaan kuuluvia virheitä merkittynä rajauslaatikoilla

Sormijatkos-virheen kohdalla minulla oli kaksi vaihtoehtoista tapaa merkitä virhe. Koska sormijatkos esiintyi yleensä yhtenäisenä rivinä, pystyin joko merkitsemään koko rivin virheeksi tai piirtämään laatikot jokaisen yksittäisen jäljen ympärille. Käytin molempia tapoja, jotta YOLO-malli oppisi tunnistamaan tarvittaessa myös yksittäisiä sormijatkoksen osia. (KUVA 7 ja KUVA 8.)



KUVA 7. Yksittäiset sormijatkosvirheet merkittynä rajauslaatikoilla



KUVA 8. Kokonainen sormijatkosvirhe merkittynä yhdellä rajauslaatikolla

Käytin samaa vaihtoehtoisten merkintätapojen logiikkaa myös esimerkiksi höylän repimien pintojen kohdalla, sillä nämä virheet olivat toisinaan hankalia hahmottaa ja sisälsivät yksittäisiä pienempiä virhealueita laajemman alueen sisällä. Kuten aiemmin totesin, YOLO-mallin on tärkeää saada mahdollisimman kattava koulutus erilaisista virheiden esiintymismuodoista.

Kuvien manuaalinen annotointi on hyvin hidasta, joten kokeilin käyttää Roboflow'n tarjoamaa Box Prompting -työkalua annotoinnin apuna. Tämä tekoälyyn pohjautuva työkalu tekee merkinnät kuvaan käyttäjän antamilla, kuvaan merkitsemillä esimerkeillä, jotka kuuluvat johonkin määriteltyyn virheluokkaan. Box Prompting oppii käyttäjän hyväksymistä ja hylkäämistä merkinnöistä ja parantaa luotettavuuttaan annotointiprosessin edetessä. (Box Prompting.) Huomasin, että työkalu tunnisti hyvin selkeät, tarkkarajaiset ja kooltaan isommat virheet. Sen sijaan pienempien tai epäsäännöllisen muotoisten virheiden tunnistuksessa esiintyi usein virheellisiä rajauksia tai virheet jäivät kokonaan huomiotta. Väärien tunnistusten poistaminen yksitellen hidasti prosessia erityisesti kuvissa, joissa oli useita virheitä. Box Prompting -työkalussa voi säätää *confidence*-arvoa, jolla hallitaan ennustusten luotettavuutta, sekä *overlap*-arvoa, jolla voidaan rajata limittäin sijaitsevien objektien määrää. Toinen annotoinnin apuna käytettävä tekoälyavusteinen työkalu, Label Assist, perustuu olemassa olevan mallin pohjalta annettaviin ehdotuksiin, joten tätä vaihtoehtoa en päässyt kokeilemaan aiemman mallin puuttuessa.

6.5 Tietoaineiston muodostaminen ja esikoulutus

Kun olin merkinnyt kaikki virheet kuviin, tarkistin merkinnät läpi vielä kerran ja korjasin ne tarvittaessa. Toisen annotoijan käyttäminen apuna tarkistusvaiheessa on suositeltavaa, mutta tässä prosessissa vastasin sekä annotoinnista että tarkistuksista itse. Kuvissa esiintyi jonkin verran virhetulkintoja ja puuttuvia merkintöjä, minkä vuoksi tarkistusvaihe oli tarpeellinen. Tarkistuksen jälkeen siirsin annotoidut kuvat Roboflow-alustalla *dataset*-osioon, johon lisäsin myös vastaavan määrän virheettömiä kuvia. *Dataset*-osiossa olevista kuvista muodostetaan tietoaineisto, jota käytetään myöhemmin mallin koulutuksessa. Osiossa oli nyt yhteensä 2758 kuvaa, jotka sisälsivät yhteensä 5473 merkintää neljästä eri virheluokasta. Taulukossa 1 on esitelty virheluokkien jakautuminen koko tietoaineistossa. Esitettyjen tietojen perusteella sormijatkos on yleisin virheluokka, kun taas naarmu on selvästi harvinaisempi.

Virheluokka	Virheitä sisältävien kuvien määrä (kpl)	Virheiden määrä (kpl)	Virheiden osuus (%)
Sormijatkos	542	2570	47,0
Halkeama	626	823	15,0
Höylän repimä pinta	417	2010	36,7
Naarmu	52	70	1,3

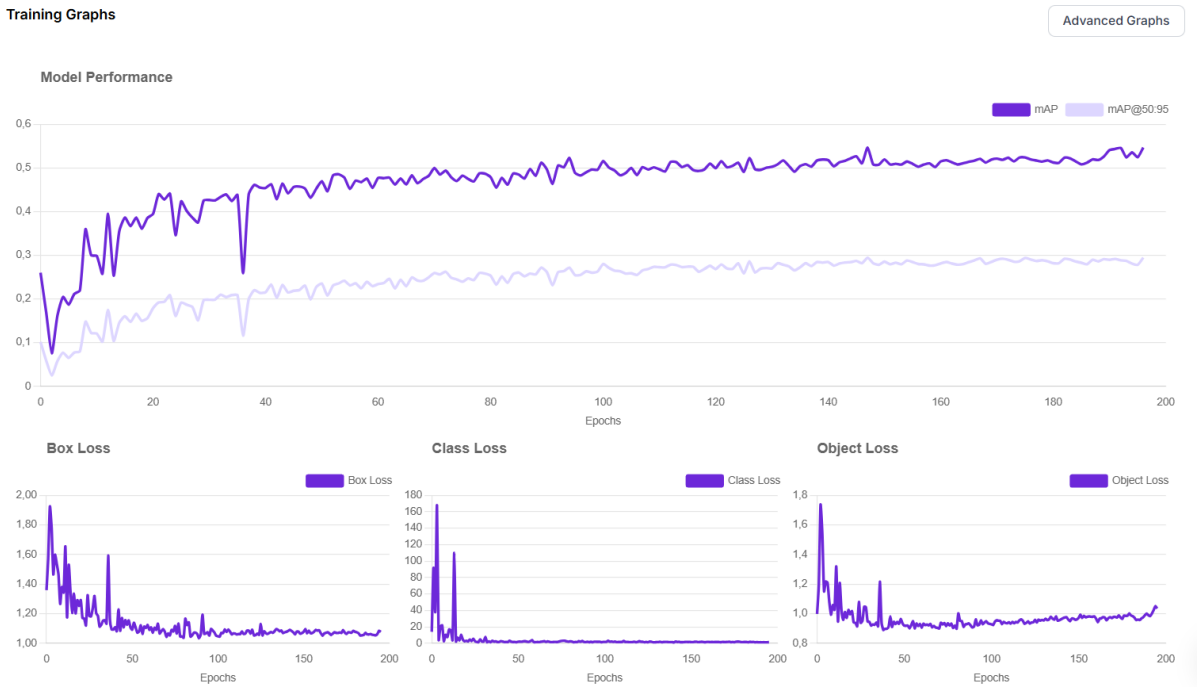
TAULUKKO 1. Virhejakauma luokittain

Tietoaineiston kaikki kuvat olivat nyt valmiina, joten aloitin tietoaineiston ensimmäisen version luomisen. Mallin koulutusta varten tietoaineisto jaettiin koulutus-, validointi- ja testausosuuksiin. Tässä prosessissa päätin käyttää jakoa, jossa 70 % kuvista kuuluu koulutusosuuteen, 20 % validointiosuuteen ja 10 % testausosuuteen. Tälle jaolle ei ole olemassa vakiintunutta standardia, ja sitä voidaan tarvittaessa muokata myöhemmissä versioissa mallin suoriutumisen perusteella. Roboflow-alusta suorittaa kuvien jaon automaattisesti käyttäjän määrittelemien osuuksien mukaisesti. Seuraavassa vaiheessa on mahdollista määritellä kuvien esikäsittely, ja Roboflow tarjoaa tähän useita vaihtoehtoja, kuten kuvien rajauksen, kuvakoon muuttamisen ja kontrastin säätämisen. Päätin pitää esikäsittelyn mahdollisimman vähäisenä ensimmäisessä versiossa ja käytin ainoastaan Roboflow'n suosittelemaa *Auto Orient* -säätöä. Kameran sensorit voivat tallentaa pikselitiedon eri tavalla, joten kuvat saattavat näkyä

väärinpäin. *Auto Orient*-säädön avulla varmistetaan, että kuvat näytetään oikeinpäin metatiedon (EXIF) mukaisesti. (Dwyer 2020.)

Esikäsittelyn lisäksi kuvia voidaan Roboflow-alustalla muokata käyttämällä augmentaatiota. Tällä tarkoitetaan kuvien käsittelyä esimerkiksi säätämällä kirkkautta, saturaatiota ja värikylläisyyttä, lisäämällä sumennusta tai kohinaa sekä kääntämällä tai peilaamalla kuvia. Augmentaation tavoitteena on simuloida aitoja tilanteita, joissa valaistus tai muut olosuhteet voivat vaikuttaa kuvan tarkkuuteen ja värimaailmaan. Koska kyseessä on vasta tietoaineiston ensimmäinen versio, en ottanut augmentaatiota käyttöön tässä vaiheessa. Myöhemmissä versioissa sen avulla on myös mahdollista kasvattaa aineiston kuvamäärää ilman, että palveluun tarvitsee ladata uusia kuvia.

Roboflow loi ensimmäisen version tietoaineistosta, kun olin määritellyt kaikki tarvittavat tiedot. Valmis aineisto on ladattavissa useissa eri tiedostomuodoissa, kuten YOLOv8, YOLOv5, COCO JSON tai Tensorflow TFRecord, ja sitä voi käyttää suoraan mallin koulutuksessa. Lisäksi Roboflow tarjoaa mahdollisuuden esikouluttaa aineistoa omalla *Roboflow Train*-työkalullaan. Esikoulutuksen avulla voidaan arvioida tietoaineiston rakenteellista laatua, sillä tulokset antavat viitteitä siitä, kuinka hyvin annotoinnit ja luokkajako tukevat mallin oppimista. Esikoulutettua mallia ei voi ladata jatkokoulutukseen, vaan sitä voi käyttää aineiston arviointiin sekä uusien kuvien annotoinnissa ja virheiden automaattisessa tunnistuksessa *Label Assi*-työkalun aineistona. Tässä työssä käytin esikoulutusta tietoaineiston laadun arviointiin, ja tämän esikoulutuksen tulokset on esitetty kuvassa 9.



KUVA 9. Tietoaineiston esikoulutuksen tulokset

Tulosten perusteella nähdään, että mAP-arvo (ylin kuvaaja) vaihtelee huomattavasti ja pysyy melko alhaisena. Tämä tarkoittaa, että malli ei vielä tunnista virheitä kovin tarkasti. *Box Loss*-arvo (kuvaaja vasemmalla alhaalla) laskee alas, mutta sen arvo pitäisi olla vielä matalampi, jotta mallin ennustamat rajauslaatikot vastaisivat paremmin todellisia kohteita. *Class Loss*-arvo (kuvaaja alhaalla keskellä) laskee myös alaspäin, mutta senkin tulisi olla pienentyä vielä lisää oikean virheluokituksen takaamiseksi. *Object Loss*-arvo (alhaalla oikealla) vaihtelee paljon, mikä voi johtua joko virheiden puuttumisesta osassa kuvista tai mallin heikosta kyvystä tunnistaa virheitä. Esikoulutuksen tulosten perusteella päätimme käyttää tietoaineistoa YOLOv8-mallin koulutuksessa, sillä arvot olivat kehittymässä oikeaan suuntaan ja tehokkaammalla koulutuksella mallin luotettavuutta voitaisiin parantaa.

6.6 YOLO-mallin koulutus muodostetulla tietoaineistolla

Latasin tietokoneelleni Roboflow-alustalla muodostetun tietoaineiston YOLOv8-tiedostomuodossa, joka sisältää kansiot *train, valid* ja *test.* Jokaisessa näistä kansioista on alikansiot *images*, joka sisältää kuvatiedostot JPG-muodossa, ja *labels,* joka sisältää merkintöjen luokat ja koordinaatit TXT-muodossa. Jokaiselle kuvatiedostolle on olemassa vastaavasti nimetty *labels*-tiedosto. Tietoaineistossa on myös *data.yaml*-tiedosto, jossa kerrotaan tietoaineiston

luokkien määrä ja nimet sekä Roboflow-alustan tiedot projektista, tietoaineiston versiosta ja URL-osoite tietoaineistoon. *README.roboflow.txt*-tiedostossa kuvataan tarkemmin Roboflow-alustaa ja tietoaineiston sisältöä, kuten kuvien määrä, annotointiformaatti sekä mahdolliset esikäsittely- ja augmentaatiotoimenpiteet.

Koska YOLO-mallin koulutus vaatii tehokasta prosessoria (CPU) ja näytönohjainta (GPU), päätin käyttää mallin koulutuksessa Google Colab -palvelua. Colab on pilvipohjainen ympäristö, joka tarjoaa Jupyter Notebook -käyttöliittymän, ja se on optimoitu koneoppimisen, datatieteen ja opetuksen tarpeisiin (Colaboratory). Colab-ympäristössä voi ajaa Python-koodia suoraan selaimessa, mikä mahdollistaa mallin kouluttamisen ilman omaa tehokasta laitteistoa. Latasin tietoaineiston ympäristöön ja asensin YOLO- ja Ultralytics-kirjastot koulutusta varten. Mallin koulutuksessa Python-koodiin määritellään käytettävä malli (*model*), koulutuskierrosten määrä (*epochs*), eräkoko (*batch size*), kuvan koko (*image size*) sekä koulutuksessa käytettävä laitteisto (CPU tai GPU). Kuvassa 10 näkyvät näille muuttujille määrittelemäni arvot.

```
from ultralytics import YOLO

model = YOLO("yolov8n.pt")  # käytettävän YOLO-mallin versio

data_yaml = '/content/dataset/Dataset_All_Class_v1/data.yaml' # käytettävän datan polku

model.train(data=data_yaml, epochs=100, batch=16, imgsz=640) # valitut arvot koulutukselle
```

KUVA 10. Ensimmäinen Python-koodiesimerkki YOLOv8 Nano -mallin koulutusta varten

Valitsin malliksi YOLOv8 Nano -mallin, joka on nopein ja kevyin, sekä tarkkuudeltaan matalin YOLOv8-malli. Muita mallivaihtoehtoja ovat Small, Medium, Large ja Extra-large, joista kaksi viimeistä ovat todella tarkkoja, mutta sen vuoksi myös melko hitaita. Koulutuskierrosten määräksi valitsin 100 kierrosta, eräkooksi 16 ja kuvakooksi 640. YOLOv8 Nano-mallin koulutukseen kului noin 60 minuuttia. Koulutuksen jälkeen suoritettiin mallin validointi parhaalla tallennetulla mallipainotiedostolla (*best.pt*), joka sisältää koulutuksen aikana optimoidut painot (KUVA 11).

```
100 epochs completed in 1.020 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

Validating runs/detect/train/weights/best.pt...
Ultralytics 8.3.109 🚀 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,006,428 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all        552       1111      0.693      0.495      0.542      0.289
                 Crack        127        169      0.795      0.734      0.799      0.486
           Fingerjoint        106        502      0.839      0.624      0.766      0.367
          Planing torn         87        430      0.531      0.521      0.502      0.276
               Scratch          8         10      0.606        0.1      0.101     0.0267
```

KUVA 11. YOLOv8 Nano -mallin tulokset 100 koulutuskierroksen jälkeen

Tulokset osoittavat mallin suorituskyvyn eri virheluokissa. Ultralyticsin tarjoamien metriikkaselitysten mukaan (Perfomance Metrics Deep Dive) voidaan tulkita, että keskimääräinen tarkkuus oli mAP50 = 0,542 ja keskimääräisen tarkkuuden keskiarvo oli mAP50-95 = 0,289. Tämä tarkoittaa, että malli kykeni havaitsemaan virheitä kohtalaisella tarkkuudella. Tarkkuus (*Box (P))* osoitti, kuinka tarkasti malli osaa piirtää rajauslaatikot oikeaan kohtaan. Tämä arvo oli tuloksissa myös kohtalainen kaikkien luokkien osalta. Parhaiten malli onnistui rajauslaatikoiden piirtämisessä sormijatkoksen (*Fingerjoint*) kohdalla ja heikoimmin höylän repimän pinnan (*Planing torn*) kohdalla. Eroa selittää se, että sormijatkoksen muoto on säännöllinen ja helposti tunnistettava, kun taas höylän repimä pinta on epäsäännöllinen ja muodoltaan vaihteleva. Kyky tunnistaa virheet (*Recall, R*) vaihteli luokkien välillä huomattavasti, ja kaikkien luokkien keskiarvo jäi melko matalaksi. Recall-arvoa laski erityisesti naarmu (Scratch) -luokan heikko tunnistettavuus, joka todennäköisesti johtui luokan pienestä kuvamäärästä testiosiossa (vain 8 kuvaa ja 10 virhemerkintää).

Tulokset viittaavat siihen, että tietoaineisto on koulutuskelpoinen, mutta mallin oppimista voitaisiin parantaa esimerkiksi lisäämällä aineistoa pieniin luokkiin kuten naarmuluokkaan tai käyttämällä toista malliversiota. Vertailun vuoksi ajoin saman koodin samoilla arvoilla myös YOLOv8 Medium -mallilla, jotta voisin vertailla mallien nopeutta ja tarkkuutta käytännössä tietoaineistollamme. Tulokset olivat hieman heikommat kuin YOLOv8 Nano -mallilla, mutta erot olivat käytännössä pieniä (KUVA 12). Tämä oli yllättävää, sillä YOLOv8 Medium -mallissa oli tulosten mukaan käytössä noin 25 miljoonaa parametria, kun taas YOLOv8 Nano -mallissa niitä oli vain noin 3 miljoonaa. Laskentateho oli samoin selvästi suurempi YOLOv8 Medium -mallissa (78,7 GFLOPs) verrattuna YOLOv8 Nano -malliin (8,1 GFLOPs).

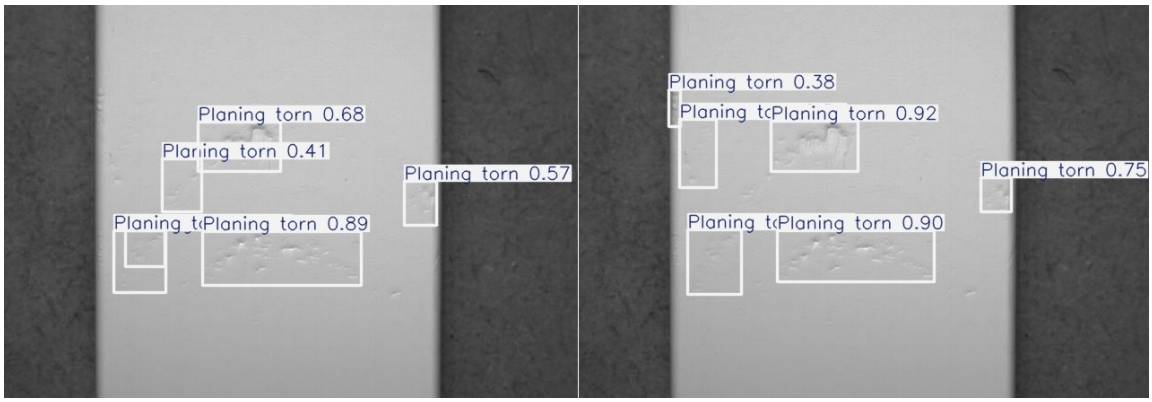
```
100 epochs completed in 2.015 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 52.0MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 52.0MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics 8.3.109 🚀 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 92 layers, 25,842,076 parameters, 0 gradients, 78.7 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all        552       1111      0.521      0.485      0.514      0.288
                 Crack        127        169      0.752      0.746      0.776      0.486
           Fingerjoint        106        502       0.83      0.659       0.78      0.385
          Planing torn         87        430      0.503      0.535      0.483      0.276
               Scratch          8         10          0          0     0.0198    0.00623
```

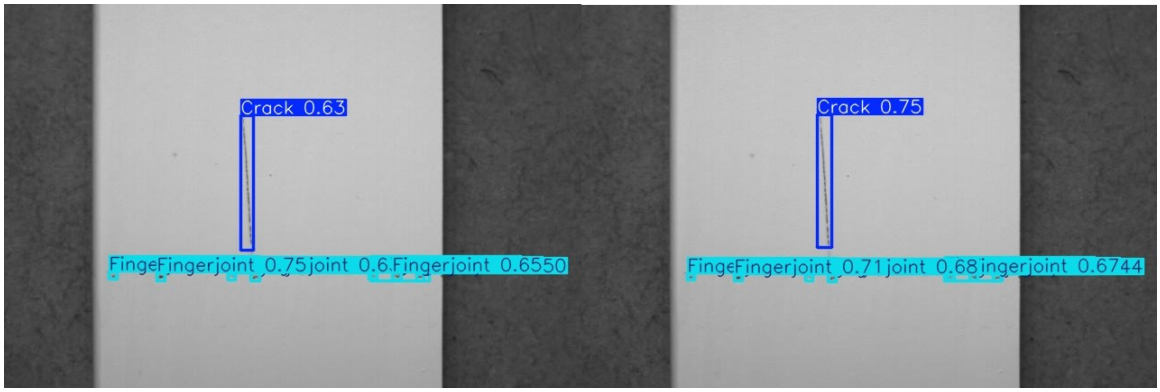
KUVA 12. YOLOv8 Medium -mallin tulokset 100 koulutuskierroksen jälkeen

Kokonaisuudessaan YOLOv8 Medium -mallin suorituskyky oli hyvin samankaltainen YOLOv8 Nano -mallin kanssa. Keskimääräinen tarkkuus (mAP50) oli hieman alempi (Medium 0,514 ja Nano 0,543), mutta mAP50-95 oli lähes sama (Medium 0,288 ja Nano 0,289), mikä viittaa siihen, ettei virheiden tunnistuksessa ollut merkittävää parannusta koko tietoaineistossa. Tarkkuudessa (*Box (P)*) YOLOv8 Medium tunnisti hyvin sormijatkokset (*fingerjoint*) ja kohtalaisesti halkeamat (*Crack*), mutta höylän repimien pintojen (*planing torn*) sekä naarmujen (*scratch*) tunnistus jäivät edelleen heikoiksi. Virheiden tunnistuskyky (*Recall R*) oli lähes sama YOLOv8 Nano -mallin kanssa, ja kaikkien luokkien keskiarvo jäi alle 0,5, mikä kertoo suhteellisen heikosta tunnistuskyvystä. Keskiarvoa laski jälleen naarmu-luokan heikko tunnistettavuus, joka johtui pienestä määrästä havaintoaineistoa.

Lopuksi testasin vielä molempien YOLO-mallien kykyä tunnistaa virheet testiaineistosta. Molemmat mallit suoriutuivat testistä lähes samantasoisesti ja tunnistivat virheitä kuvista kohtalaisella tarkkuudella. Kuvassa 13 näkyvät höylän repimän pinnan (*planing torn*) virheet YOLOv8 Nano- ja YOLOv8 Medium -mallien tunnistamana, ja kuvassa 14 vastaavasti sormijatkos- ja halkeamavirheet molempien mallien tunnistamana. Kuvissa esillä olevat tarkkuusprosentit vastaavat koulutustuloksia, eli mallit ovat hyvin tasavertaisia tunnistuksessa ja löytävät samankaltaisia virheitä kuvista.



KUVA 13. Höylän repimä pinta (planing torn) -virheluokka YOLOv8 Nano- (vasemmalla) ja YOLOv8 Medium (oikealla) -mallien tunnistamana



KUVA 14. Sormijatkos- (fingerjoint) ja halkeama (crack) -virheluokat YOLOv8 Nano (vasemmalla) ja Medium (oikealla) -mallien tunnistamana

Näiden tulosten pohjalta ja tällä tietoaineistolla tarkasteltuna YOLOv8 Nano vaikuttaisi olevan parempi vaihtoehto jatkokoulutukseen. Vaikka YOLOv8 Medium -malli sisälsi huomattavasti enemmän parametreja ja sen laskentateho oli suurempi, ei sen suorituskyky ollut merkittävästi parempi YOLOv8 Nano -malliin verrattuna. Lisäksi Medium-mallin koulutus kesti yli kaksi kertaa kauemmin. Tietoaineiston laatua parantamalla, esimerkiksi lisäämällä naarmuluokan kuvia tai jopa jättämällä kyseisen luokan kokonaan pois aineistosta, voitaisiin koulutuksen tuloksia parantaa huomattavasti.

7 YHTEENVETO JA POHDINTA

Tämän opinnäytetyön tavoitteena oli esitellä kuvadatan annotointiprosessi, siihen käytettävät menetelmät ja työkalut sekä arvioida annotointiprosessin tuotoksena syntyneen tietoaineiston toimivuutta pintakäsiteltyjen puukappaleiden virheentunnistuksessa. Prosessissa muodostettua tietoaineistoa käytettiin YOLO-mallin koulutuksessa, ja koulutustulosten perusteella arvioitiin tietoaineiston laatua. Annotointiprosessissa käytettäväksi työkaluksi valittiin Roboflow, jonka käyttöliittymä oli yksinkertainen ja selkeä. Työkalussa oli mahdollista toteuttaa objektintunnistusta YOLO-mallin koulutuksessa käytettävien rajauslaatikoiden avulla sekä tarkastella muodostetun tietoaineiston laatua esikoulutuksen ja sen tuottaman analyytikan perusteella.

Annotointiprosessissa yllätti sen hitaus ja suuri työmäärä, sillä kuvien manuaalinen merkintä vaati erityistä tarkkuutta sekä perehtymistä pintakäsiteltyjen puupintojen virheiden tunnistamiseen. Erityisen tärkeäksi osoittautuivat virheluokkien tarkat määrittelyt ja selkeät ohjeet niiden tunnistamiseen kuvista. Tässä käytin apuna tiimin asiantuntijoita, joilla on pitkä kokemus puualalta. Automaattinen objektintunnistus nopeutti prosessia jonkin verran, ja jatkossa esikoulutetun mallin käyttö parantaa automaattisen tunnistuksen tarkkuutta uudessa kuvamateriaalissa huomattavasti, mikä voi tehostaa koko annotointiprosessia.

Tietoaineiston laadun testaaminen YOLO-mallin koulutuksessa oli keskeinen vaihe, sillä koulutuksen tulosten voidaan arvioida aineiston soveltuvuutta syväoppimisen kehittämiseen ja tehdä tarvittavia parannuksia. Testasin tässä työssä kahta eri versiota YOLOv8-mallista (Nano ja Medium), mutta jatkossa voisi olla hyödyllistä kokeilla esimerkiksi uudempaa YOLO11-mallia samalla aineistolla ja vertailla eri malliversioiden suorituskykyä keskenään. YOLOv8 Nano- ja Medium -mallien koulutustulokset osoittavat, että tietoaineistossa harvoin esiintyvät virheluokat heikensivät mallin tarkkuutta. Tämän perusteella voidaan suositella, että jatkokehityksessä näitä luokkia joko täydennetään riittävällä määrällä virheitä sisältäviä kuvia tai tarvittaessa poistetaan tietoaineistosta kokonaan. Liian vähän virheitä sisältävien luokkien mukanaolo voi vääristää mallin oppimista ja heikentää sen suorituskykyä. Tässä työssä erityisesti *naarmu*-luokka osoittautui selkeästi haasteelliseksi koulutuksen kannalta sen vähäisen esiintymismäärän vuoksi.

Tämän opinnäytetyön konkreettisena tuotoksena muodostettiin laadukkaasti annotoitu tietoaineisto, jossa on noudatettu lähdekirjallisuudessa ja tutkimuksissa esitettyjä suosituksia ja periaatteita. Tietoaineisto on ensimmäinen versio, jota tullaan jatkokehittämään laajentamalla sen kokoa ja tarkastelemalla virheluokkien määrää.

Olen itse tyytyväinen tuottamani aineiston laatuun, sillä YOLOv8-mallin koulutustulokset osoittivat sen soveltuvan hyvin syväoppimismallin opettamiseen. Työskentely vaati huolellista ja järjestelmällistä työtapaa, jotta kaikki virheet saatiin tunnistettua ja merkittyä johdonmukaisesti. Annotointiprosessin kuvaus vaihe vaiheelta auttoi hahmottamaan sen kulkua, ja se toimii myös käytännön tarkastuslistana annotointia tehdessä. Perehtyminen YOLO-algoritmia käsitteleviin tutkimuksiin ja lähdekirjallisuuteen kannatti, sillä se syvensi ymmärrystäni annotoinnin merkityksestä sekä työkalujen ja merkintätapojen vaikutuksesta mallin koulutuksen onnistumiseen. Oli myös mielenkiintoista kouluttaa mallia käytännössä ja vertailla kahden eri malliversion tuloksia, sekä osoittaa kouluttamani mallin virheentunnistuksen toimivuus testikuvien avulla.

Halusin pitää opinnäytetyöni lähestymistavan tieteellisenä, mutta kuitenkin helposti ymmärrettävänä, ja uskon onnistuneeni siinä kohtuullisen hyvin. Eräänä haasteena koin englanninkielisen termistön runsaan määrän, ja yritin mahdollisuuksien mukaan etsiä niille suomenkielisiä vastineita. Toivon, että työstäni on hyötyä kuvadatan annotointiprosessien ja syväoppimismallien parissa työskenteleville henkilöille, sekä yrityksille, jotka harkitsevat tekoälyn käyttöönottoa virheentunnistuksen tehostamiseksi.

LÄHTEET

Agamirov, L., Agamirov, V., Los, V., Nosikov, M. & Toutova, N. 2024. Dataset Annotation Converter for Training Neural Networks to Detect Defects on Various Surfaces. *Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED),* Moscow, Russian Federation, 1–4. Saatavissa: https://doi.org/10.1109/TIR-VED63561.2024.10769786. Viitattu 24.3.2025.

Ali, M. L., & Zhang, Z. 2024. The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection. *Computers*, *13*(12), 336. https://doi.org/10.3390/computers13120336. Viitattu 24.3.2025.

Alpaydin, E. 2021. *Koneoppiminen*. Helsinki: Terra Cognita Oy. Viitattu 28.2.2025.

Bandyopadhyay, H. 2022. *The Definitive Guide to Instance Segmentation [+V7 Tutorial].* V7. Saatavissa: https://www.v7labs.com/blog/instance-segmentation-guide. Viitattu 31.3.2025.

Bochkovskiy, A., Chien-Yao, W. & Hong-Yuan ML. 2020. *YOLOv4: Optimal Speed and Accuracy of Object Detection.* Saatavissa: https://doi.org/10.48550/arXiv.2004.10934. Viitattu 22.3.2025.

Centria-ammattikorkeakoulun tutkimus ja kehitys strategia 2021–2024. Centria. Saatavissa: https://net.centria.fi/wp-content/uploads/2022/06/STRATEGIA_tutkimus-ja-kehitys_2021-2024.pdf. Viitattu 18.4.2025.

Choi RY, Coyner AS, Kalpathy-Cramer J, Chiang MF & Campbell JP. 2020. Introduction to Machine Learning, Neural Networks, and Deep Learning. *Transl Vis Sci Technol.* 9(2), 14. Saatavissa: https://pubmed.ncbi.nlm.nih.gov/32704420/. Viitattu 7.3.2025.

Coskun, D., Karaboga, D., Basturk, A., Akay, B., Nalbantoglu, ÖU., Dogan, S., Pacal, I. & Karagöz, M. 2023. A comparative study of YOLO models and a transformer-based YOLOv5 model for mass detection in mammograms*. Turkish Journal of Electrical Engineering and Computer Sciences*, 31(7), artikkeli 10. Saatavissa: https://doi.org/10.55730/1300-0632.4048. Viitattu 25.3.2025.

CVAT. *Leading Data Annotation Platform.* Saatavissa: https://www.cvat.ai/. Viitattu 2.4.2025.

Du, L., Zhang, R. & Wang, X. 2020. *Overview of two-stage object detection algorithms.* Journal of Physics: Conference Series. 1544: 012033. Saatavissa: http://doi.org/10.1088/1742-6596/1544/1/012033. Viitattu 21.3.2025.

Dwyer, B. 2020. *When should I Auto-Orient My Images?*. Roboflow Blog. Saatavissa: https://blog.roboflow.com/exif-auto-orientation/. Viitattu 12.4.2025.

Fang, Y., Zhu, D., Zhou, N., Liu, L. & Yao, J. 2021. *PiPo-Net: A Semi-automatic and Polygon-based Annotation Method for Pathological Images*. IEEE. Saatavissa: https://doi.org/10.1109/IROS51168.2021.9636146. Viitattu 31.3.2025.

Gheorghe, G., Duguleana, M., Boboc, RG. & Postelnicu, CC. 2024. Analyzing Real-Time Object Detection with YOLO Algorithm in Automotive Applications: A Review. *CMES - Computer Modeling in Engineering and Sciences*. 141(3), 1939–1981. Saatavissa: https://doi.org/10.32604/cmes.2024.054735. Viitattu 25.3.2025.

Google. *Colaboratory.* Saatavissa: https://research.google.com/colaboratory/faq.html?authuser=0. Viitattu 15.4.2025.

Hailiang, L., Bin, Z., Yu, Z., Weiwei, L., Yijun, M., Jiacheng, H. & Linfeng, W. 2020
A semi-automated annotation algorithm based on weakly supervised learning for medical images. *Biocybernetics and Biomedical Engineering*, 40(2), sivut 787–802. Saatavissa: https://doi.org/10.1016/j.bbe.2020.03.005. Viitattu 28.3.2025.

Hänninen, P. 2022 *Robotiikka ja tekoäly.* Tammertekniikka / Amk-Kustannus Oy. Viitattu 10.3.2025.

Jain, V. 2020. *Introduction to Semantic Image Segmentation.* Medium, Analytics Vidhya. Saatavissa: https://medium.com/analytics-vidhya/introduction-to-semantic-image-segmentation-856cda5e5de8. Viitattu 31.3.2025.

Kang, S., Hu, Z., Liu, L., Zhang, K. & Cao, Z. 2025. Object Detection YOLO Algorithms and Their Industrial Applications: Overview and Comparative Analysis. *Electronics*, *14*(6), 1104. Saatavissa: https://doi.org/10.3390/electronics14061104. Viitattu 24.3.2025.

Khanam, R. & Hussain, M. 2024. *YOLOv11: An Overview of the Key Architectural Enchancements.* Saatavissa: https://doi.org/10.48550/arXiv.2410.17725. Viitattu 23.3.2025.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, AC., Lo, W-Y., Dollár, P. & Girshick, R. 2023. *Segment Anything*. Saatavissa: https://doi.org/10.48550/arXiv.2304.02643. Viitattu 30.3.2025.

Kellher, JD. 2020. *Syväoppiminen. Kuinka tekoäly toimii.* Helsinki: Terra Congnita Oy. Viitattu 13.3.2025.

Kumari, P. 2023. *The Ultimate Guide to Image Annotation: Techniques, Tools and Best Practices.* Labellerr. Saatavissa: https://www.labellerr.com/blog/guide-to-image-annotation-techniques-tools-and-best-practices/. Viitattu 31.3.2025.

Kundu, R. 2023. *YOLO: Algorithm for Object Detection Explained [+Examples]*. Saatavissa: https://www.v7labs.com/blog/yolo-object-detection. Viitattu 21.3.2025.

Label Studio. *Open Source Data Labeling Platform.* Saatavissa: https://labelstud.io/. Viitattu 2.4.2025.

Ladi, N. 2024. *Roboflow on building with open-source and Ultralytics YOLOv8*. Ultralytics. Saatavissa: https://www.ultralytics.com/blog/roboflow-on-building-with-open-source-and-ultralytics-yolov8. Viitattu 7.4.2025.

Lavitas, L., Redfield, O., Lee, A., Fletcher, D., Eck, M. & Janardhanan, S. 2021. *Annotation Quality Framework – Accuracy, Credibility, and Consistency.* Saatavissa: https://datacentricai.org/neurips21/papers/49_CameraReady_DCAI2021_tex(8).pdf. Viitattu 24.3.2025.

Lecun, Y., Bengio, Y. & Hinton, G. 2015. Deep learning. *Nature*, 521 (7553), 436–444. Saatavissa: https://doi.org/10.1038/nature14539. Viitattu 21.3.2025.

Liu, L., Ouyang, W., Xiaogang, W., Fieguth, P., Chen, J., Liu, X. & Pietikäinen, M. 2020. Deep Learning for Generic Object Detection: A Survey. *Int J Comput Vis,* 128, 261–318. Saatavissa: https://doi.org/10.1007/s11263-019-01247-4. Viitattu 12.3.2025.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X. & Wei, X. 2022. *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications.* Saatavissa: https://arxiv.org/pdf/2209.02976. Viitattu 22.3.2025.

Nolan M. 2024. *Deep Learning Picks Apart Data-Copying Puzzles.* IEEE Spectrum. Saatavissa: https://spectrum.ieee.org/ai-genetics-rna-transcription. Viitattu 21.3.2025.

Perfomance Metrics Deep Dive. Ultralytics. Saatavissa: https://docs.ultralytics.com/guides/yolo-performance-metrics/. Viitattu 17.4.2025.

Rajnerowicz, C. 2023. *Keypoint Annotation: Labeling Data With Keypoints & Skeletons.* V7. Saatavissa: https://www.v7labs.com/blog/keypoint-annotation-guide. Viitattu 31.3.2025.

Ravpreet K. & Sarbjeet S. 2023. A comprehensive review of object detection with deep learning. *Digital Signal Processing,* 132. https://doi.org/10.1016/j.dsp.2022.103812. Viitattu 12.3.2025.

Redmond, J., Divvala, S., Girshick, R & Farhadi, A. 2016. *You Only Look Once: Unified, Real-Time Object Detection*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA. Saatavissa: https://10.1109/CVPR.2016.91. Viitattu 21.3.2025.

Roboflow. *Everything you need to build and deploy computer vision applications.* Saatavissa: https://roboflow.com/. Viitattu 5.4.2025.

Roboflow Docs. *Box Prompting.* Saatavissa: https://docs.roboflow.com/annotate/use-roboflow-annotate/box-prompting-ai-labeling. Viitattu 10.4.2025.

Sager, C., Janiesch, C., & Zschech, P. 2021. A survey of image labelling for computer vision applications. *Journal of Business Analytics*, *4*(2), 91–110. Saatavissa: https://doi.org/10.1080/2573234X.2021.1908861. Viitattu 26.3.2025.

SuperAnnotate. *Annotate faster with one-shot image annotation.* Saatavissa: https://www.superannotate.com/image-annotation-tool. Viitattu 5.4.2025.

Supervised learning. Ultralytics. Saatavissa: https://www.ultralytics.com/glossary/supervised-learning. Viitattu 8.3.2025.

Tian, Y., Ye, Q. & Doermann, D. 2025. *YOLOv12: Attention-Centric Real-Time Object Detectors*. Saatavissa: https://doi.org/10.48550/arXiv.2502.12524. Viitattu 23.3.2025.

Terven, J. & Cordova-Esparza, DM. 2023. *A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and Beyond.* Saatavissa: https://doi.org/10.48550/arXiv.2304.00501. Viitattu 21.3.2025.

Tutkimus, kehitys ja palvelut. Centria. Saatavissa: https://net.centria.fi/tki/. Viitattu 20.4.2025.

Segment Anything Model (SAM). Ultralytics. Saatavissa: https://docs.ultralytics.com/models/sam/. Viitattu 30.3.2025.

Vibhuti, Jindal, N., Singh, H. & Rana, PS. 2022. Face mask detection in COVID-19: a strategic review. *Multimed Tools Appl* 81, 40013–40042. Saatavissa: https://doi.org/10.1007/s11042-022-12999-6. Viitattu 25.3.2025.

Wang, C-Y., Yeh, I-H. & Liao H-YM. 2024. *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information.* Saatavissa: https://doi.org/10.48550/arXiv.2402.13616. Viitattu 23.3.2025.

Wang A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J. & Ding, G. 2024. *YOLOv10: Real-Time End-to-End Object Detection.* Saatavissa: https://doi.org/10.48550/arXiv.2405.14458. Viitattu 23.3.2025.

What is machine learning?. IBM. Saatavissa: https://www.ibm.com/think/topics/machine-learning. Viitattu 7.3.2025.

Yaseen, M. 2024. *What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next Generation Object Detector.* Saatavissa: https://doi.org/10.48550/arXiv.2408.15857. Viitattu 22.3.2025.

YOLOv9: A Leap Forward in Object Detection Technology. Ultralytics. Saatavissa: https://docs.ultralytics.com/models/yolov9/. Viitattu 23.3.2025.

YOLOv10: Real-Time End-to-End Object Detection. Ultralytics. Saatavissa: https://docs.ultralytics.com/models/yolov10/. Viitattu 23.3.2025.

Zhao, Z-Q., Zheng, P., Xu, S. & Wu, X. 2019. *Object Detection with Deep Learning: A Review.* Saatavissa: https://doi.org/10.48550/arXiv.1807.05511. Viitattu 12.3.2025.

Zhu, Y., Yang, Q. & Xu, L. 2024. *Active Learning Enabled Low-cost Cell Image Segmentation Using Bounding Box Annotation.* Saatavissa: https://doi.org/10.48550/arXiv.2405.01701. Viitattu 30.3.2025.