

# Kartongin leikkausjäljen tutkiminen konenäön avulla

## Tiivistelmä

Tekijä(t) Niemi, Aki	Julkaisun laji Opinnäytetyö, AMK	Valmistumisaika 2024
	Sivumäärä 26	
Työn nimi <b>Kartongin leikkausjäljen tutkiminen konenäön avulla</b>		
Tutkinto Insinööri AMK		
Toimeksiantajan nimi, titteli ja organisaatio Heikki Kokkomäki, Kunnossapitopäällikkö, Stora Enso Ingerois Oy		
Tiivistelmä <p>Tässä opinnäytetyössä oli tavoitteena tutkia kartongin leikkausjälkeä konenäön avulla. Toimeksiantajana toimi Stora Enso Ingerois Oy. Konenäkömalli oli tarkoituksenaan opettaa tunnistamaan huonosta leikkausjäljestä syntyvät epätäydellisyydet.</p> <p>Pallettien suurien kokovaihteluiden takia konenäkömallin kameralle tehtiin automaattinen tarkennuksen säätö Tia-portaalin ja Pythonin avulla. Tarkennuksen säädölle tehtiin oma käyttöliittymä, jossa oli parametrit erikokoisille palleteille. Kuvat luokiteltiin ja niistä merkittiin huonon leikkausjäljen luomat epäkohdat konenäkömallille. Konenäkömalli opetettiin ja testattiin kuvien avulla.</p> <p>Konenäkömalli saatiin opetettua tunnistamaan opetusdatan kuvat oikein ja halutulla tarkkuudella. Uusien kuvien kohdalla malli teki virheitä epäkohtien tunnistuksessa. Tämä johtui pallettien eroavaisuuksista ja liian pienestä määrästä opetusdataa. Malli saatiin opetettua hyvin, mutta sen yleistäminen oli haluttua heikompaa.</p>		
Asiasanat Konenäkö, Neuroverkko, Kartonkipalletti		

## Abstract

Author(s) Niemi, Aki	Type of Publication Thesis, UAS	Published 2024
	Number of Pages 26	
Title of Publication <b>Studying cardboard cutting quality using computer vision</b>		
Name of Degree Engineer (UAS)		
Name, title and organization of the client Heikki Kokkomäki, Maintenance Manager, Stora Enso Ingerois Oy		
Abstract <p>In this thesis the objective was to investigate the cutting quality of cardboard using computer vision. The client was Stora Enso Ingerois Oy. The computer vision model was intended to be trained to recognize defects resulting from bad cutting quality.</p> <p>Due to significant variations in pallet sizes, an automatic focus adjustment for the computer vision model camera was implemented using Tia-portal and Python. A dedicated user interface was created for the focus adjustment, incorporating parameters for different pallet types. Images were classified, and the defects caused by poor cutting quality were annotated for the computer vision model. The model was trained and tested using these images.</p> <p>The computer vision model was successfully trained to correctly identify the images in the training data with the desired accuracy. However, when presented with new images, the model made errors in defect detection. This was attributed to variations in pallets and the insufficient amount of training data. While the model was well-trained, its generalization performance was weaker than desired.</p>		
Keywords Computer vision, Neural networks, Cardboard pallets		

## Sisällys

1	Johdanto.....	1
2	Teoria .....	2
2.1	Konenäkö .....	2
2.2	Oppimisen vaikuttajat .....	2
2.3	Neuroverkot.....	3
3	Työympäristö .....	5
3.1	Stora Enso .....	5
3.2	Jälkikäsittely .....	5
3.3	Kartonkilajit.....	8
4	Taustatietoa ja aiheen esittelemistä .....	10
4.1	Huono leikkausjälki.....	10
4.2	Kameran paikan päättäminen.....	11
4.3	Kameran valitseminen .....	11
4.4	Axis Q1659 kamera .....	13
4.5	Kameran laukaisu.....	16
4.6	Kameran tarkennus .....	16
5	Datan kerääminen ja luokittelu.....	20
5.1	Kuvien kerääminen.....	20
5.2	Stora Enso CVP .....	20
6	Yhteenveto .....	23
6.1	Konenäköjärjestelmästä saadut tulokset.....	23
6.2	Kehitysideat.....	24
	Lähteet .....	25

## 1 Johdanto

Opinnäytetyön tarkoituksena on tutkia kartongin leikkausjälkeä konenäköä hyödyntäen. Tällä hetkellä Stora Enso Ingerois -tehtaalla leikkausjälkeä tarkistaa arkkileikkureiden operaattorit. Leikkausjäljen laadunarviointi suoritetaan silmämääräisesti. Leikkausjäljelle ei tällä hetkellä ole selkeää pistettä, milloin se on hyvää tai huonoa, vaan se riippuu operaattorin mielipiteestä.

Tavoitteena on luoda konenäkö ja opettaa sitä tunnistamaan leikkausjäljen laatu. Kun leikkausjälki alkaa huonontua, alkaa kartonkiarkkien sivuille muodostua kuituja, jotka saavat palleit näyttämään karvaisilta. Kartonki tulee yhtenä rullana arkkileikkurin perälle, josta rulla nostetaan koneeseen ja kartonki leikataan pituus- ja leveysuunnassa haluttuihin mittoihin. Leikkausjäljen huonontuessa joko kiristetään teriä tai terät vaihdetaan, jos kiristys ei ratkaise ongelmaa.

Kun luodaan ja opetetaan konenäköä, vaaditaan tarpeeksi tarkka kamera. Kuvan pitää olla tarpeeksi tarkka, jotta siitä voidaan erottaa repsottavat kuidut. Jos kuvan laatu ei ole tarpeeksi hyvä, on siitä vaikea erottaa yksityiskohtia ja täten on myös vaikeampi opettaa ja luoda toimivaa ohjelmaa.

## 2 Teoria

### 2.1 Konenäkö

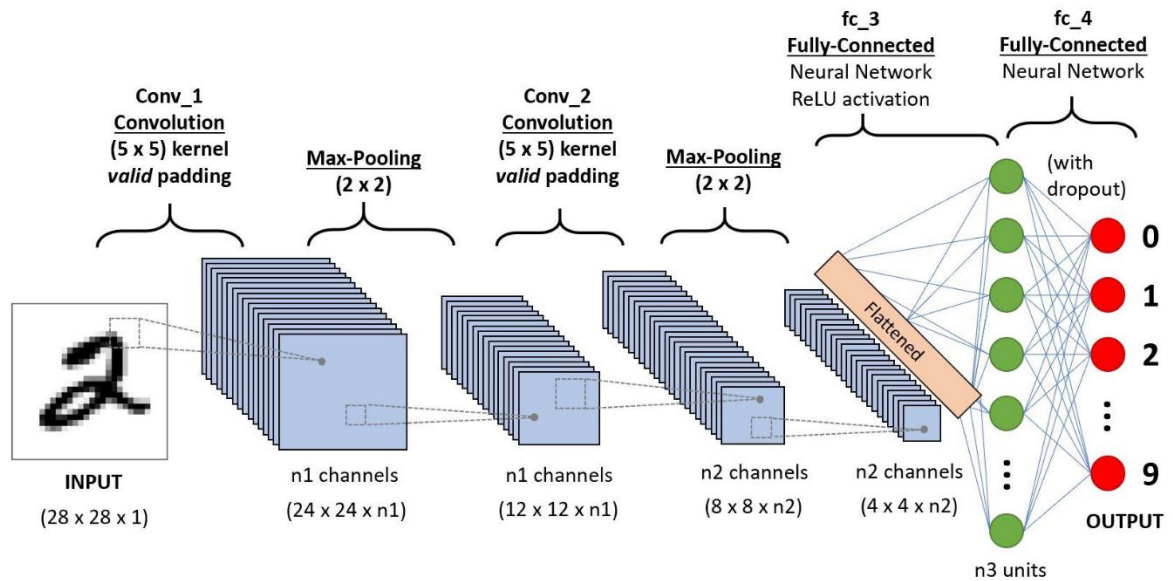
Konenäkö (Machine vision) on tietokoneen kyky havaita ympäristöä (Boesch 2023). Konenäkö pyrkii jäljittelemään ihmisen näköä. Tämä ei kumminkaan ole helppoa sillä maailmamme on kolmiulotteinen, kun taas kamerat ottavat kuvan kaksiulotteisena, jossa tapahtuu suuri määrä tiedon häviämistä. (Sonka et al. 1993, 1.) Konenäkö automaattisesti tulkitsee ja havainnoi kuvainformaatiota. Projektissa käytetään algoritmeihin perustuvaa yksityiskohtaisempaa kuvatiedon tulkintaa kuvantunnistusmallien avulla.

Kvanttunnistusmallien tarkoituksena on huomata kuvista pikseli-intensiteetin vaihtelua ja spatiaalista assosiaatiota, joka yhdistetään johonkin kuvan ominaisuuteen. Projektissa mallin tulisi tunnistaa huonosta leikkausjärjelmästä syntyvät epäkohdat. Konenäkömallin tarkkuuteen ja onnistumiseen vaikuttaa sekä opetettavissa olevan datan määrä ja laatu, että myös koneoppimismallin rakenne eli neuroverkon topologia. Mallin optimointi on vaikeaa ja datan kerääminen ja luokittelu on työlästä. Mallille pitää kertoa mitä kuvista etsitään. (Ruokolainen 2020.)

### 2.2 Oppimisen vaikuttajat

Neuroverkkojen oppimiseen suurin vaikuttaja on datan määrä ja laatu. Mitä enemmän monipuolista ja hyvälaatuista dataa on sitä parempi. Mikäli haluttaisiin esimerkiksi tunnistaa kuvista autoja konenäön avulla, voidaan konenäölle opettaa autojen olevan kuvattu sivusta, ja niiden olevan vihreitä. Tällöin konenäkö tunnistaa kuvista kaikki autot, jotka ovat vihreitä ja kuvattu sivusta. Jos samalla konenäöllä yritetään tunnistaa kuvasta musta auto, joka on kuvattu edestä, tekee konenäkö todennäköisesti virheen kyseisen auton luokittelussa, koska sitä ei ole opetettu havaitsemaan autoa sellaisena. Tämän vuoksi on tärkeää, että konenäkö on opetettu käyttötarkoitukseensa riittävän monipuolisesti. (Kallio 2018.)

Neuroverkkojen rakennetta, joka nähdään kuvassa 1., voidaan muokata oppimisen parantamiseksi. Siihen voidaan lisätä kerroksia niin monta kuin halutaan. Neuroverkolla on mahdollista luokitella hyvällä tarkkuudella sellaisiakin kuvia, joiden kanssa ihminen voi tehdä herkästi virheitä. (Kallio 2018.)



Kuva 1. Neuroverkon rakenne (Ratan 2023)

## 2.3 Neuroverkot

Neuroverkot ovat laskennallisia prosessointisysteemejä, jotka on suunniteltu jäljittelemään ihmisten aivojen toimintaa ja kykyä oppia. Neuroverkot ovat tekoälyn tärkeimpien komponenttien joukossa. Neuroverkko rakentuu useista tietojenkäsittelyelementeistä, jotka ovat kytkettyinä toisiinsa joko vahvistavasti tai heikentävästi. Opetusdatan avulla neuroverkko opetetaan antamaan haluttu vastaus. Jos neuroverkko antaa väärän vastauksen, säädetään yhteyksiä niin, että saavutetaan haluttu vastaus. (Winter 2023.)

Kuvantunnistukseen neuroverkot soveltuvat hyvin. Neuroverkot pystyvät tunnistamaan automaattisesti erilaisia piirteitä kuvasta, ilman ihmisen tekemää määrittystä. Neuroverkko pyrkii etsimään kuvista piirteitä, jotka ovat yhteisiä kuvissa esiintyvillä kohteilla. Opetuksen jälkeen neuroverkko pystyy tunnistamaan uusista kuvista piirteitä, joita oli opetusdatassa. Neuroverkot mahdollistavat monimutkaisten ja suurien kuvamäärien nopean tunnistamisen. (Winter 2023.)

Konvoluutioinen neuroverkko eli CNN on yleinen valinta, kun käsiteltävä data on kuvadataa (Nvidia 2023). Nimensä tämä neuroverkko on saanut matemaattisen konvoluution soveltamisesta laskennassaan. CNN-neuroverkon opettamistapoja on kolme: ohjattu oppiminen, ohjaamaton oppiminen ja semiohjattu oppiminen. Ohjatussa oppimisessä neuroverkko tutkii ihmisen luokittelemia kuvia pikselitasolla ja kehittää niistä laskennalliset parametrit. Neuroverkko oppii matkimaan ihmisen tapaa luokitella kuvia ja lopulta se pystyy luokittelemaan uusia kuvia parametriensa avulla. Ohjattu oppiminen on yleisin tapa opettaa CNN

neuroverkkoa. Ohjaamattomassa oppimisessa opetusdataa ei ole luokiteltu ollenkaan vaan neuroverkko etsii kuvista samankaltaisuuksia ja luokittelee kuvat niiden mukaan. Semiohjatusoppimisessa hyödynnetään molempia tapoja, eli vain osa opetusdatasta on luokiteltu. (Kallio 2018.)



## 3 Työympäristö

### 3.1 Stora Enso

Stora Enso syntyi vuonna 1998, kun suomalainen Enso Oyj ja ruotsalainen Stora Kopparbergs Bergslags Aktie Bolag yhdistyivät. Stora oli kaivos-, raudanjalostus- ja puuliiketoimintaan keskittyvä yritys vuoteen 1970 asti, kunnes kaivos-, ja metallitoiminta myytiin ja keskittyttiin metsätalouteen. Enso oli yritysostojen ja fuusioiden myötä 1990-luvulla Suomen suurin metsäyhtiö. (Stora Enso 2023a.)

Nykyään Stora Enso on uusiutuvien pakkaus-, biomateriaali- ja puuteollisuuden maailmanlaajuinen tuotteiden toimittaja. Stora Enso edistää muutosta kohti biopohjaista kiertotaloutta vähähiilisillä ja kierrätettävillä puukuitupohjaisilla tuotteilla. Stora Enso työllistää noin 21 000 ihmistä ympäri maailmaa. (Stora Enso 2023b.)

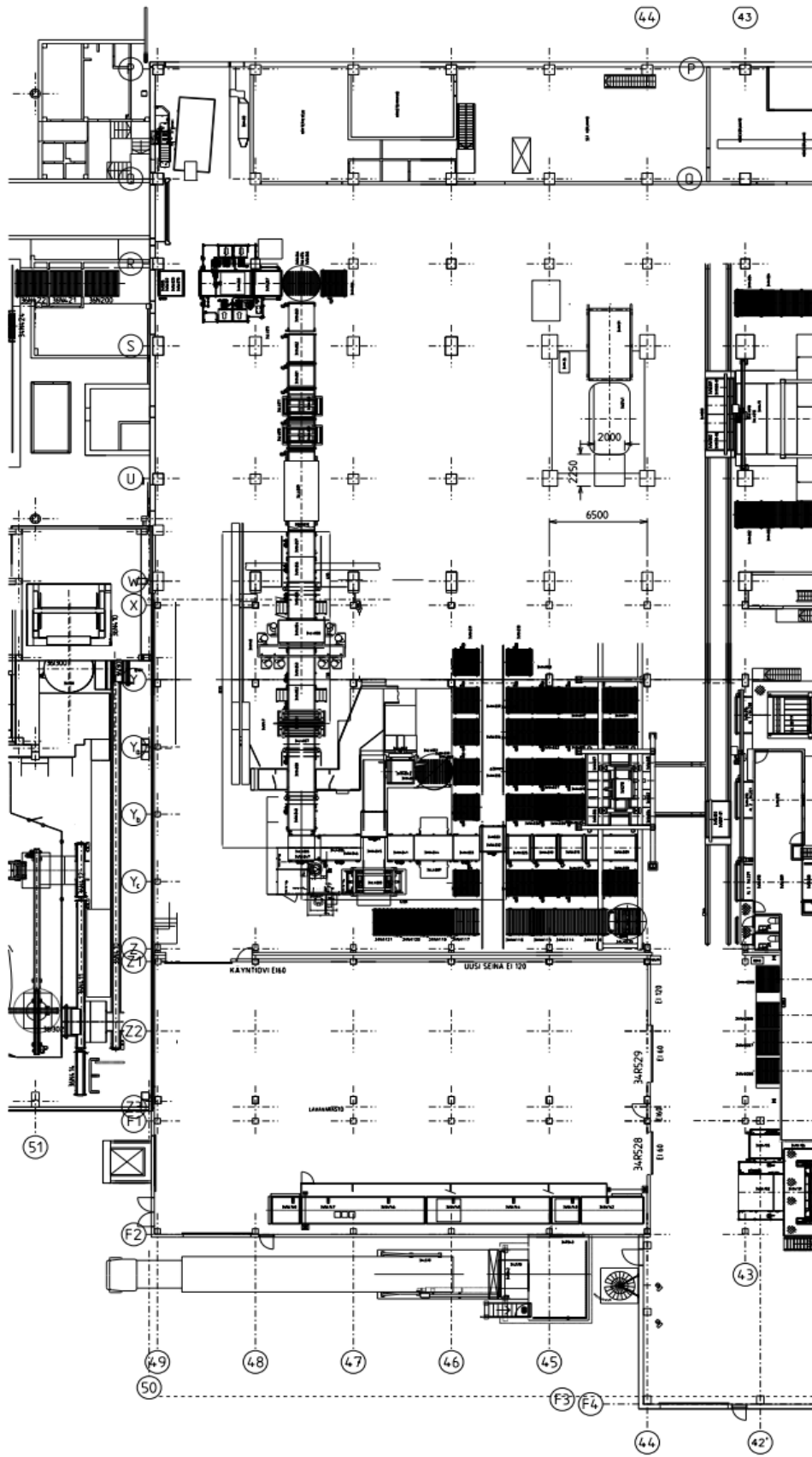
#### **Stora Enso Ingerois**

Stora Enson Inkeröisten kartonkitehtaalla valmistetaan taivekartonkia kuluttajapakkausteollisuuden tarpeisiin kuten ruoka-, lääke- ja makeispakkaukset. Kartonkitehdas työllistää 200 henkilöä ja päämarkkina-alueena on Eurooppa. Tehtaan arkituskapasiteetti on 204 000 tonnia vuodessa ja kartonkikoneen kapasiteetti on 310 000 tonnia vuodessa. Varastokapasiteettia tehtaalla on 4000 tonnia. (Stora Enso 2022.)

### 3.2 Jälkikäsittely

Opinnäytetyön pääympäristönä toimi kartonkitehtaan jälkikäsittely, jonka pohjapiirros nähdään kuvassa 2. Jälkikäsittelyyn kuuluu:

- Välrullavarasto
- 5 kpl arkkileikkureita
- Lavavarasto
- Lajittelualue
- Arkkipakkaamo
- Tuotevarasto



Kuva 2. Jätkikäsittelyn pohjapiirros

Välirullavarastossa varastoidaan arkkileikkureille arkitukseen tarkoitetut kartonkirullat, jotka on leikattu pituusleikkurilla pienemmiksi rulliksi kartonkikoneen konerullasta. Valmistelu- asema, jossa valmistellaan arkkileikkureille arkitukseen tulevat rullat, sijaitsee välirullava- raston välittömässä läheisyydessä. Valmisteluasemalle rullat siirretään vastapainotrukeilla.

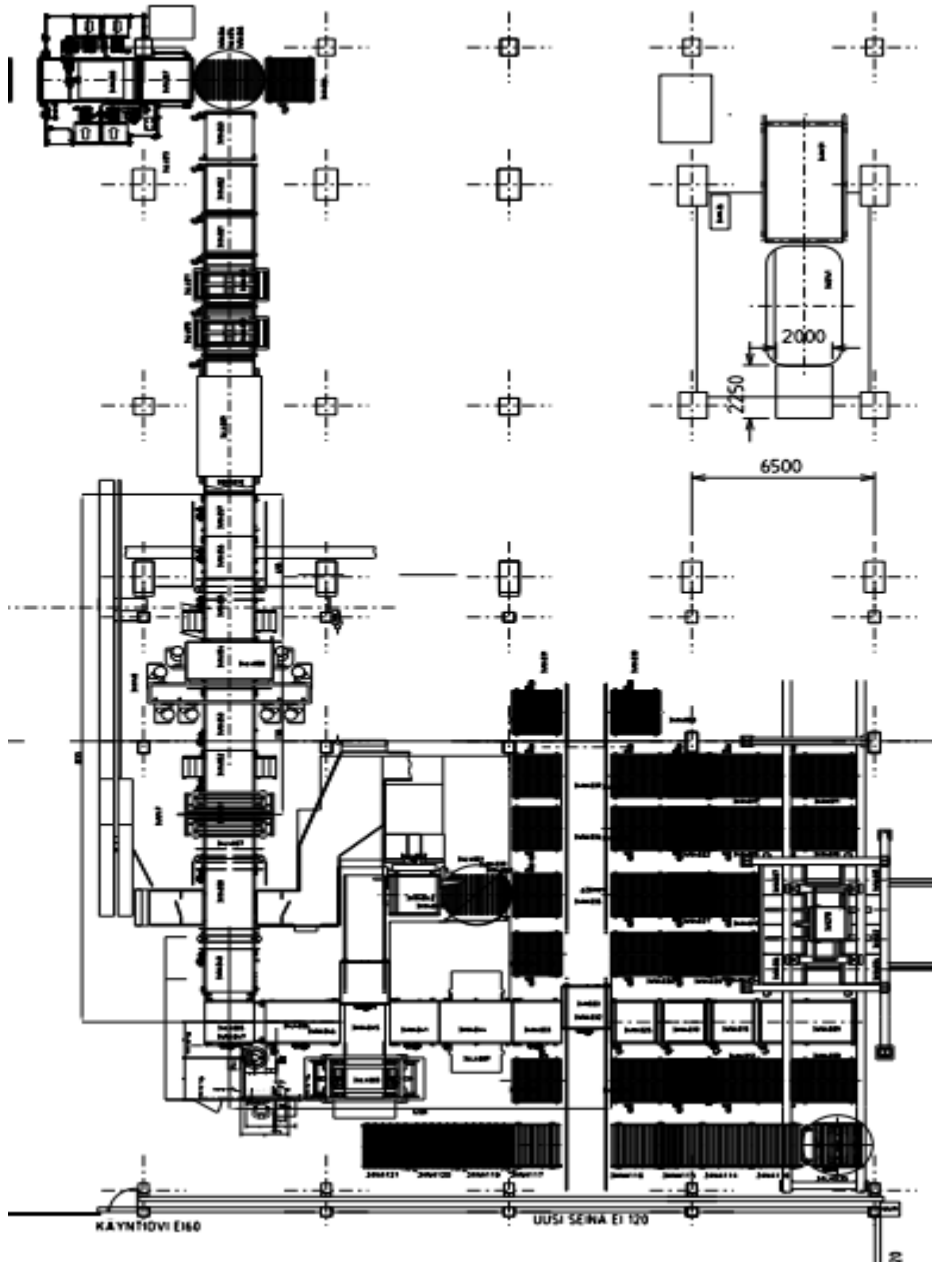
Valmisteluasemalla valmistellut rullat siirretään vihivaunuilla ensin odottamaan arkitusta ja sen jälkeen arkkileikkurin aukirullauspukkiin arkitettavaksi. Aukirullauspukeilla on kaksi rul- lapaikkaa, jotka mahdollistavat ”lentävät rullanvaihdot”, jolloin leikkuria ei tarvitse vaih- doissa pysäyttää. Arkkileikkureita on yhteensä viisi kappaletta, hyötyleveydet ovat 2100 mm ja 4500 mm välillä. Leikkureilla leikataan kartonkirata arkeiksi asiakkaan haluamien arkki- mittojen mukaan, ensin halkileikkuuasemalla ja sen jälkeen poikkileikkuuasemalla. Leik- kauksen jälkeen arkit limitetään ja pinotaan leikkurin vastaanottopäähän arkkilavojen päälle. Näitä valmiita pinoja kutsutaan palleteiksi.

Valmiit palletit siirtyvät siirtovaunujen sekä kuljettimien avulla lajittelualueelle ja sieltä kah- den oikaisijan kautta arkkipakkaukseen. Oikaisijoilla oikaistaan kartonkipinot vielä suoriksi sekä kohdistetaan arkkilava kohdalleen. Lajittelualueella on päälinjan lisäksi useampi kul- jetinlinja, jotka toimivat varastointipaikkana pakkaukseen meneville palleteille arkkipakkaa- mon muovirullien vaihtojen sekä muiden pakkaamon häiriöiden aikana.

Lajittelualueen ja pakkaamon välissä sijaitsee myös kaksi kääntölaitetta, joissa on mahdol- lista poistaa palleteista ja arkeista viat, joita ei voi asiakkaalle lähettää. Näitä vikoja ovat esim. roskat arkkien välissä, taipuneet arkit, pintaviat kartongissa, huono leikkaus ym. La- jittelualueelta asiakkaalle jostain syystä kelpaamattomat palletit voidaan ohjata myös pois linjalta. Nämä hylkypalletit voidaan pulpperoimisen jälkeen hyötykäyttää uudelleen karton- kimassana.

Arkkipakkaamo toimii automaattisesti. Kuljettimilla liikkuvien pallettien päälle laitetaan ensin ns. hattumuovi ja sen jälkeen pallettien ympärille tulee verhokalvo, joka kuumasaumataan tiiviiksi ”hupuksi”. Tämän jälkeen palletit siirtyvät kutisteuuniin 220 asteen lämpötilaan noin puoleksi minuutiksi. Tämän jälkeen ovat vielä jäähdytyspuristimet, jotka puristavat palletit kasaan kutistemuovin kanssa, tässä kohtaa pakkausmuovi kutistuu vielä lähelle loppuki- reyttään. Pakkauslinjan pohjapiirros esitellään kuvassa 3.

Pakkauslinjan lopussa on vielä etikettirobotit, jotka laittavat palletin tiedot sisältävät etiketit pallettien jokaiselle sivulle. Pallettien tietoja ovat muun muassa: kartongin laatu, tilausnu- mero, palletin id-numero, paino, arkkimitat ja viivakoodi. Tämän jälkeen palletit siirtyvät tuo- tevarastoon. Tuotevarastosta palletit lastataan joko junanvaunuihin tai rekka-autoihin. Las- tausta tapahtuu ympäri vuorokauden kaikkina viikonpäivinä.



Kuva 3. Pakkauslinjan layout

### 3.3 Kartonkilajit

Arkkileikkaamossa leikataan erilaisia kartonkilajeja ja niissä on eroavaisuuksia, jotka vaikuttavat leikkausjälkeen. Suurimmat vaikuttajat leikkausjälkeen ovat leikattavan kartongin laatu ja grammapaino. Kartongit voidaan jakaa kahteen eri osaan: Inkeröiden omat kartonkilajit Tambrite ja Tamfold sekä palveluarkituksena Imatran tehtaiden kartonki.

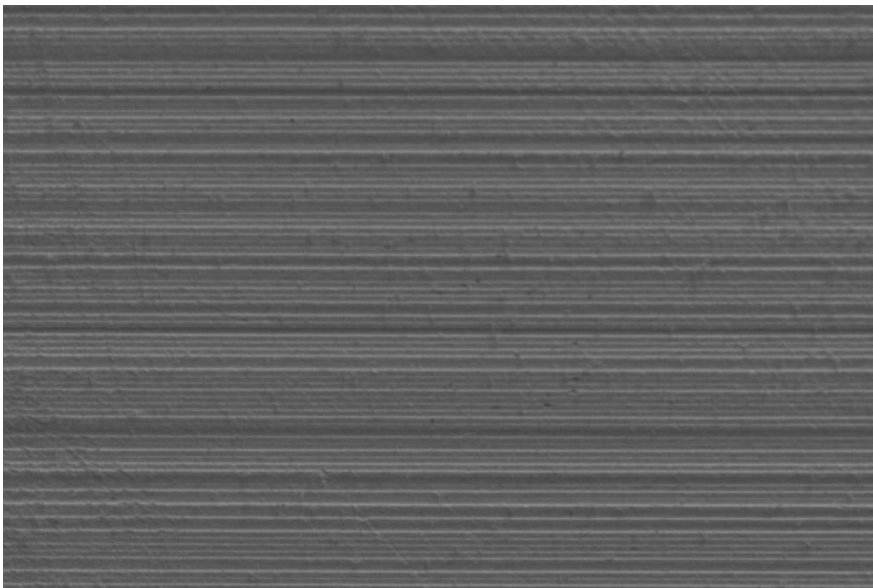
Inkeröiden omalla kartongilla grammapainoa tai paksuutta nostattaessa kasvaa pölyämisen riski. Grammapainon kasvaessa kasvaa kartongin runkokerros samalla, kun pinta-, selkä- ja päällyskerrokset pysyvät samankokoisena. Mitä paksumpaa kartonki on, sitä helpommin runkokerros pölyyntyä. Imatran kartonki on sellukartonki, eli siinä ei ole hiokekerrosta pölyntymässä, joten leikkausjäljen saa helpommin hyväksi.

## 4 Taustatietoa ja aiheen esittelemistä

### 4.1 Huono leikkausjälki

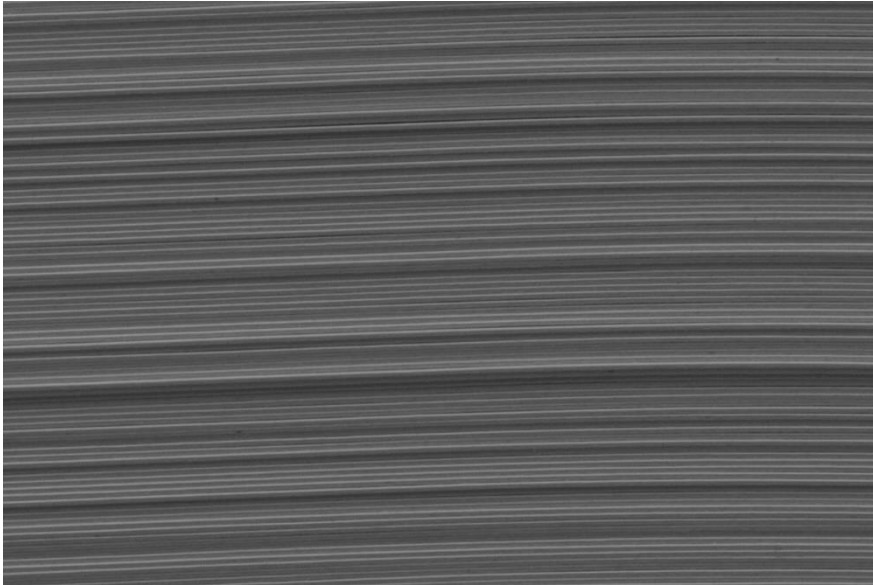
Palletit leikataan halki- ja poikkileikkaussuunnassa, joten huonoa leikkausjälkeä voi tulla kumpaan suuntaan tahansa, mutta yleensä se on poikkileikkaussuunnassa. Tämä johtuu siitä kun arkki leikataan poikki, terä vetää kartonkia vähän eteenpäin, jolloin poikkileikkausjälki ei ole niin hyvää, jos terä on tylsä. Tämä saadaan korjattua vetoteliin painetta nostamalla. Yleisin syy huonoon leikkausjälkeen on terien tylsistyminen tai löystyminen. Ensin terää kiristetään ja katsotaan, paraneeko leikkausjälki. Jos leikkausjälki ei parantunut, vaihdetaan terä.

Suurimman osan ajasta leikkausjälki oli hyvää, mikä taas vaikeutti projektia. Ensimmäinen vaikeus tuli, kun konenäön opetusdataa ei ollut saatavilla paljoa. Toinen vaikea tapaus tuli kameran säätämisen suhteen, kun ei ollut varmaa tietoa oliko kameran tarkkuus tarpeeksi hyvä. Kuvassa 4 nähdään huonoa leikkausjälkeä.



Kuva 4. Huonoa leikkausjälkeä

Huonon leikkausjäljen erottaa palletista, kun se näyttää "pörröiseltä". Pörröiseltä näyttävät epätäydellisyydet ovat kartongin kuituja, jotka eivät ole leikkaantuneet täydellisesti. Hyvässä leikkausjäljessä ei ole nähtävissä repsottavia kuituja ja jälki näyttää tasaisemmalta. Hyvän leikkausjälki nähdään kuvassa 5.



Kuva 5. Hyvää leikkausjälkeä.

## 4.2 Kameran paikan päättäminen

Projektin alussa suunniteltiin, mihin kamera voitaisiin asentaa siten, että saadaan mahdollisimman paljon tietoa ja parhaiten kuvattua palleja. Ensimmäisenä ideana oli asentaa kamerat jokaisen leikkurin vastaanottopäättyyn. Tämä ei kuitenkaan olisi ollut järkevää Proof of concept eli Poc-vaiheessa, kun tarkoituksena oli vain todistaa, että idea toimii. Tämän jälkeen mietittiin kameran laittamista oikaisijoille.

Ongelmana tässä huomattiin, että oikaisijat tärisivät ja liikkuvat todella paljon, joten kameraa ei olisi voinut kiinnittää niihin. Oikaisijoiden ympäristöllä ei myöskään ollut tilaa laittaa kameraa mihinkään. Tästä seuraavana lähdettiin ajattelemaan kameran laittamista kääntöpöydän lähelle. Kääntöpöydällä sai kuvattua molemmat leikkuujäljet ja siellä oli tarpeeksi tilaa kameralle. Lopuksi kamera laitettiin kääntöpöydän alueelle juuri näiden syiden takia.

## 4.3 Kameran valitseminen

Kameran valitsemisessa otimme yhteyttä laitetoimittajaan, joka on toimittanut Inkeröisten tehtaalle useita kameroita eri käyttötarkoituksiin. Päädyimme vastaavanlaiseen kameraan, jota oli käytetty toisessa konenäköprojektissa, jossa se oli toiminut hyvin. Jouduimme odottamaan kameraa noin 6 viikkoa suuren komponenttipulan takia. Kamera asennettiin ja kohdistettiin valon kanssa, joka voidaan nähdä kuvassa 6.



Kuva 6. Kamera ja valo asennettuna

Kameran kuvanlaadussa ilmeni ongelmia. Kamerasta tulevat kuvat olivat muuten tarkkoja, mutta se pakkasi kuvia liikaa, jolloin huono leikkausjälki jäi näkemättä. Yritimme monin keinoin saada kameran kuvia vähemmän pakatuiksi ja saimmekin kuvia yksityiskohtaisemmiksi. Pakkausta ei kuitenkaan saatu tarpeeksi alhaiseksi, jolloin kuvista jäi liian paljon yksityiskohtia pois. Liian pakatuista kuvista ei huonoa leikkausjälkeä pystynyt erottamaan, joka voidaan nähdä kuvasta 7.





Kuva 7. Ensimmäisen kameran kuva.

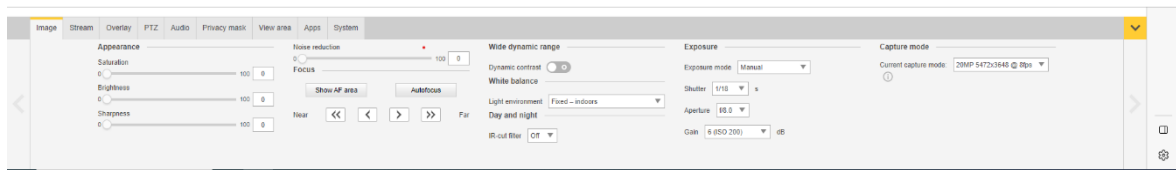
#### 4.4 Axis Q1659 kamera

Aikaisemman kameran heikon kuvanlaadun takia aloitettiin etsimään uutta korkeampilaatua kameraa. Kamera, johon päädyimme, oli Axis Q1659 20 Mp kamera. Syynä tämän kameran valintaan oli suuri megapikselimäärä. 20 megapikselin kamera riittäisi tarpeeksi yksityiskohtaisen kuvan ottamiseen. Laitetoimittajalla oli myös kamera, jota voitiin lainata ja kokeilla toimisiko se projektin käyttötarkoitukseen. Uusi kamera täytti vaatimukset suuren pikselimäärän ja pienen pakkauksen takia.

Kameralla on oma käyttöliittymä, jossa pystyy manuaalisesti säätämään kameran eri asetuksia kuten:

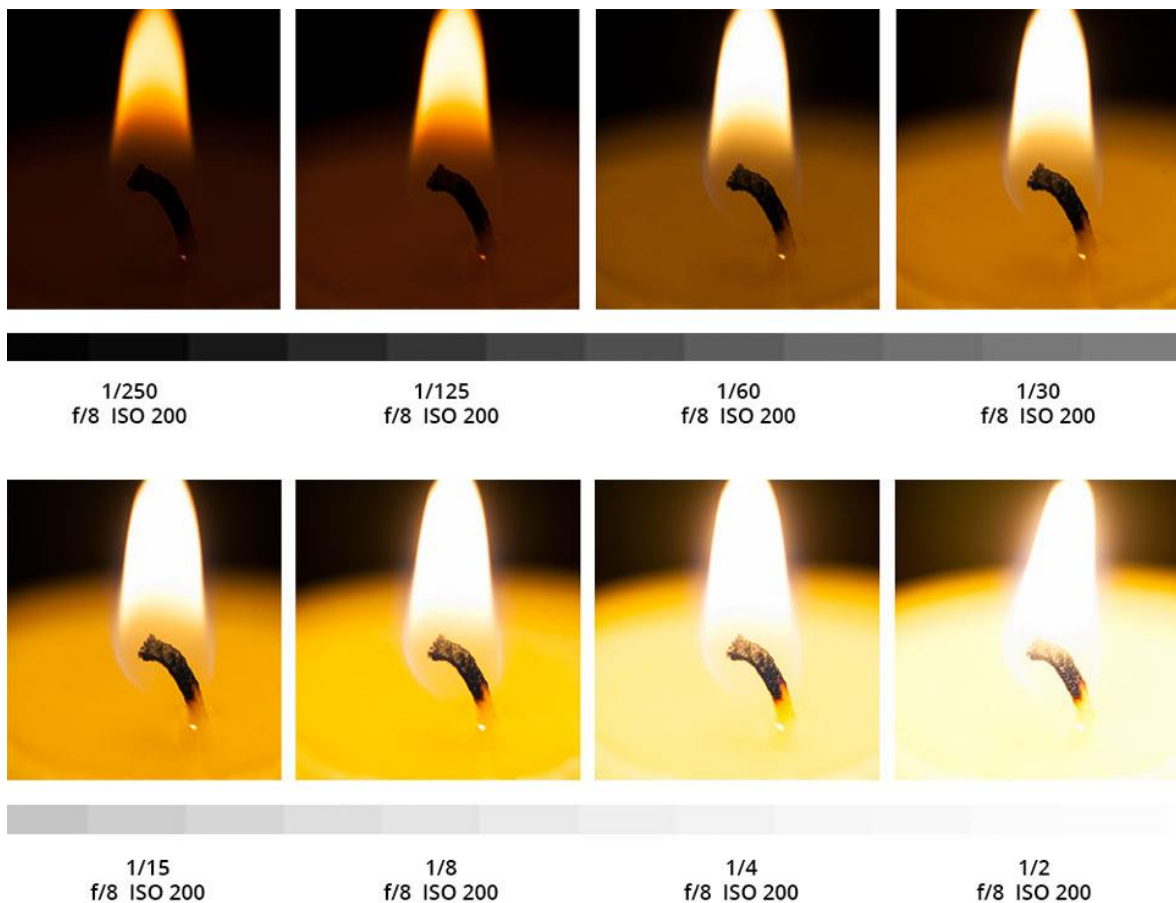
- Kuvan ulkomuodon muokkaaminen
- Kameran tarkennus
- Dynaaminen kontrasti
- Valkotasapaino
- Valotus
- Tallennustapa

Kuvan kylläisyyttä, kirkkautta ja tarkkuutta pystyy säätämään keinotekoisesti käyttöliittymästä, mutta näitä liikaa muuttamalla voi menettää kuvan yksityiskohtia. Tästä syystä keskityimme pelkästään valotuksessa oleviin asetuksiin ja niiden luomiin eroihin kuvan laadussa. Kameran asetukset ovat esitettynä kuvassa 8.



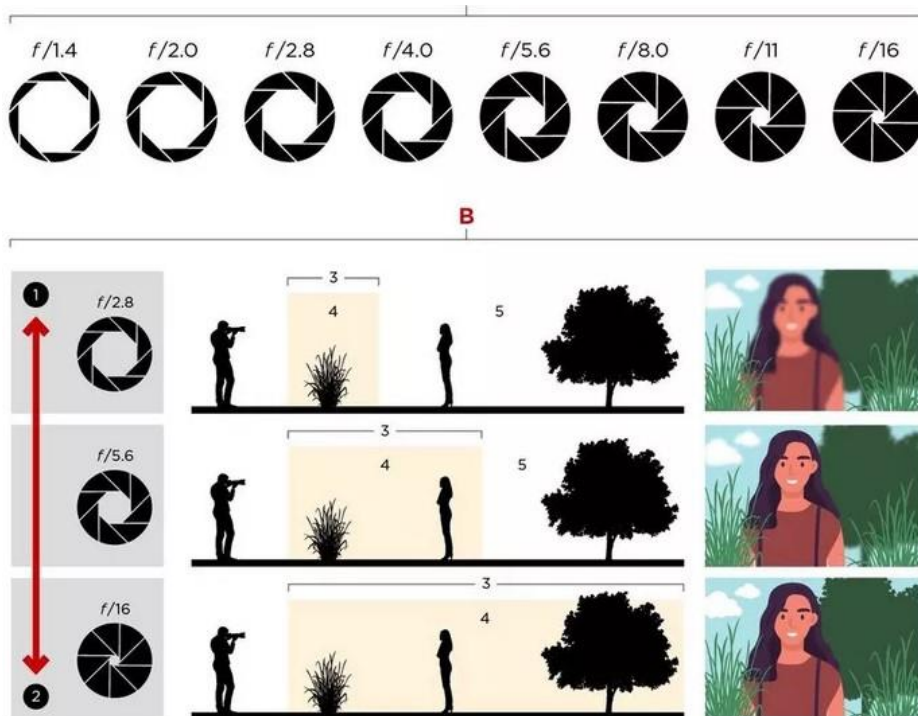
Kuva 8. Kameran asetukset

Ensimmäisenä valotuksen asetuksena on suljinaika. Suljinaika määrää ajan, kuinka kauan suljin on auki ja asettaa kennon alttiiksi valolle. Lyhyt suljinaika saa kuvan alivalottumaan ja kuvasta tulee liian tumma, kun taas pitkä suljinaika saa kuvan ylivalottumaan tehden kuvasta kirkkaan. Suljinajan vaikutus kuvan valottumiseen esitetään kuvassa 9. Lyhyempi suljinaika sopii nopeasti liikkuvan kohteen kuvaamiseen, kun taas pitkä suljinaika sopii paikallaan olevan tai hitaasti liikkuvan kohteen kuvaamiseen. Suljinaika mitataan sekunneissa ja sekuntien murto-osissa. (Scandinavianphoto 2023.)



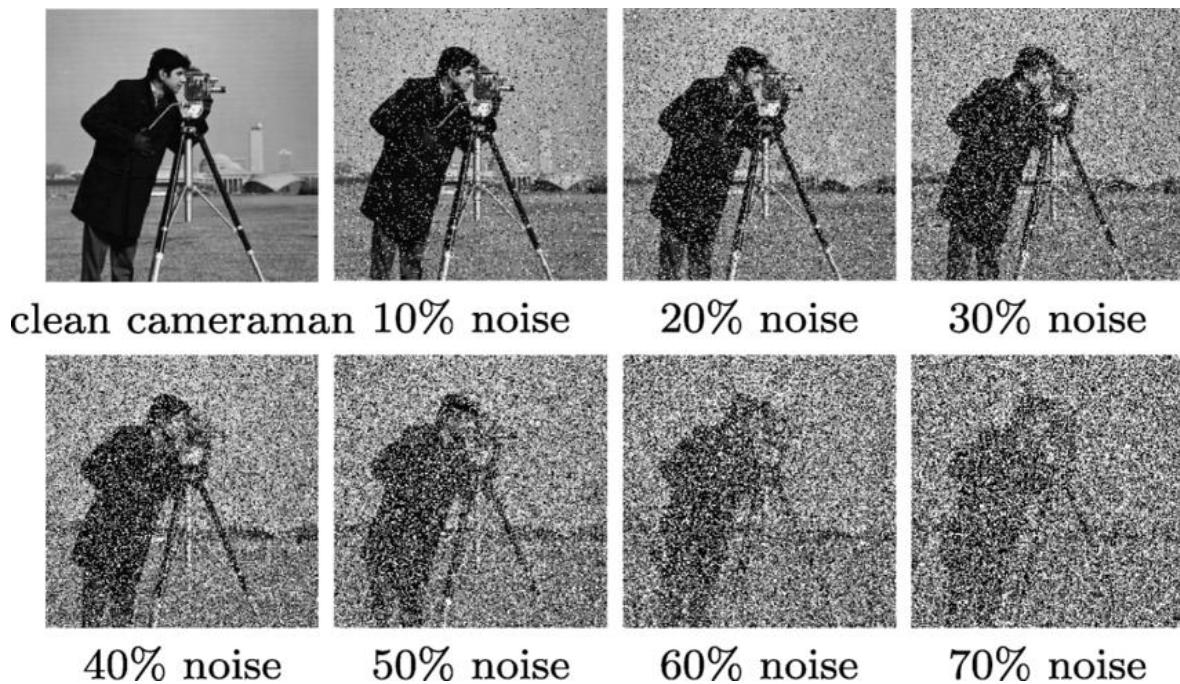
Kuva 9. Suljinajan vaikutus (Vorenkamp 2018)

Toisena asetuksena on objektiivin aukon koko, joka määrittelee paljonko valoa pääsee objektiivin läpi kameran sensoriin. Objektiivin aukkoa voidaan verrata ihmisen silmän pupilliin, joka suurenee ja pienenee ollessa joko valoisassa tai pimeässä tilassa. (Mansurov 2023.) Aukon koko ilmoitetaan f-stoppeina, kuten  $f/2.0$ , jossa  $f$  on polttoväli ja jakamalla se annetulla numerolla saadaan aukon ympäröimä (Meyer & Summersby 2023). Aukon koko siis kontrolloi syvyysterävyysaluetta. Suuri aukon koko soveltuu läheltä otettaviin kuviin, jossa halutaan lähellä oleva objekti kuvaan tarkimpana. Pieni aukon koko soveltuu kauempaa otettuihin kuviin, joissa halutaan kaukana olevien asioiden olevan tarkkoja, kuten esimerkiksi maisemakuvat. (Mansurov 2023.) Aukon koon muutos esitetään kuvassa 10.



Kuva 10. Aukon muutos (Canon-Europe 2023)

Viimeisenä muokattavana arvona kamerallamme on ISO-arvo. ISO-arvo säätelee kameran kuvakennon signaalin herkkyyttä. Mitä suurempi arvo sitä vahvempi signaali on. ISO-arvolla säädetään kameran lopullinen valotus. ISO-arvo on erittäin hyödyllinen, jos halutaan käyttää jotakin tiettyjä suljinajan ja aukon asetuksia. Pientä ISO-arvoa kannattaa käyttää, kun valoa on runsaasti ja suurempaa arvoa kun kyseessä on tummempi kuvauspaikka. Iso-arvoa suurentaessa syntyy kuvaan kumminkin kohinaa, joten oikean suuruuden löytäminen on tärkeää. Kohinan määrä riippuu kameran kuvakennosta, sillä suuremmalla kuvakennolla pikselit ovat suurempia ja ne vastaanottavat enemmän valoa. (Siersbaek 2019.) Kuvassa 11 näemme miten kohina vaikuttaa kuvaan.



Kuva 11. Kohinan vaikutus kuvaan (Ju et al. 2021)

#### 4.5 Kameran laukaisu

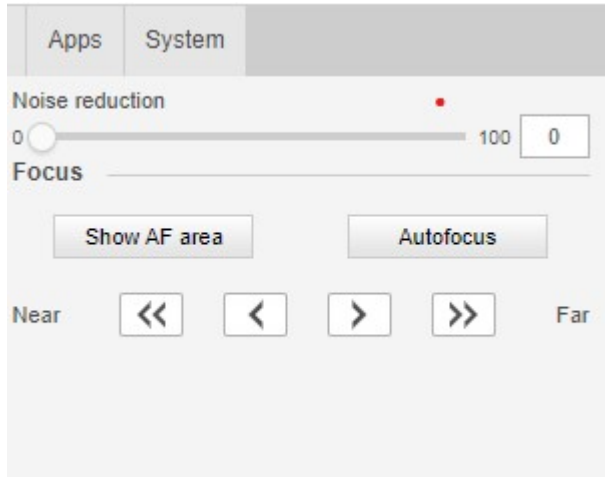
Kuvien ottamista varten kameralle tarvitsi saada lähetettyä käsky, joka ilmoittaa kameralle, milloin ottaa kuvat. Ensimmäisenä suunnitelmassa oli Adam I/O:n asennus kameralle ja paikalla olevan tietokoneen väliin. Moduuli olisi toiminut kytkimenä kameralle laukaisulle. Tämä ei kuitenkaan onnistunut pitkien toimitusaikojen takia.

Kameran laukaisu toteutettiin linjanohjauslogiikan avulla, joka ohjaa kuljettimia, jotka kuljettavat palleita pakkalinjan läpi. Palletti ajetaan kääntöpöydälle, kun se on vastaanottoasennossa. Ohjelma antaa yhden sekunnin mittaisen signaalin asettamalla digitaalilähdön päälle, kun palletti on keskitetty kuvausasentoon kääntöpöydällä. Tämä logiikan lähtö on johdettu välireleen kautta kameralle, joka on konfiguroitu laukaisemaan kuvanotto. Tämän jälkeen kääntöpöytä ohjataan logiikan avulla luovutusasentoon ja pysäytetään siihen, jolloin aikaisemmin mainittua digitaalilähtöä käytetään uudelleen laukaisemaan pallein toisen sivun kuva.

#### 4.6 Kameran tarkennus

Suurena ongelmana huomattiin palleiden suuret kokovaihtelut. Vaihteluväli oli jopa yli 60 cm palleiden välillä. Tämä aiheutti ongelmia tarkennuksen kanssa, kun yhdellä tarkkuudella ei pysty kuvaamaan tarkasti niin suurta vaihteluväliä. Testailujen aikana noin 10 cm koon vaihtumisen jälkeen kuvasta alkoi hävitä tarkkuutta ja kuva ei ollut enää tarpeeksi tarkka. Myöskään kameralle omassa käyttöliittymässä ei ollut työkaluja tähän. Kameran tarkennusta

pystyi kuitenkin muuttamaan html-komennoilla. Komennot, joita käytimme, olivat +bigstep, +minstep, -bigstep ja -minstep. Plus-merkkiset komennot kasvattavat fokusta ja miinus-merkkiset pienentävät fokusta. Komennot vastaavat käyttöliittymän near- ja far- painikkeita, jotka voidaan nähdä kuvassa 12.



Kuva 12. Kameran Near- ja Far- painikkeet.

Automaatiovastaava teki Tia-portaalissa ja Wincc:ssä ohjelmakoodin, joka lähetti kameralle komennon tarkentaa lähemmäs tai kauemmas. Käyttöliittymästä tai kameran valmistajalta ei saanut numeerisia arvoja siihen, kuinka paljon jokainen bigstep tai minstep olivat, joten tarkennus päätettiin nollata aina uuden pallelin kohdalla, jotta olisi aina sama piste mistä aloittaa tarkennus. Automaatiovastaava teki tarkennukselle oman käyttöliittymän, jossa oli lajiteltu pallettikoot 400 mm – 1400 mm kymmenessä kategoriassa 100 mm vaihtelulla. Testauksien jälkeen huomattiin, että 100 mm vaihteluväli oli liian vähän, joten vaihdettiin 50 mm vaihteluväliin. Tia-portaalissa kulki valmiiksi pallelin mittatiedot, joten niitä hyödynnettiin tarkennuksen määrittämisessä. Pallelin mittatiedot ilmoitetaan pituus- ja leveysmittoina. Tarkennus tehdään joko leveyden tai pituuden mukaan, riippuen siitä saapuuko palletti leveys- vai pituussuunta edellä kääntöpöydälle.

Ensimmäisessä kameran tarkennusohjelmassa ilmeni ongelmana askelten epäsäännöllisyys. Ohjelma tarkensi kameraa oikein noin 60 % ajasta ja loput 40 % ajasta ohjelma jätti tarkennuksen nollaamisesta askelia pois, mikä johti heikosti tarkennettuihin kuviin. Askelten jääminen välistä vaikeutti tarkennusta myös uusien pallettikokojen kanssa. Ei saatu täyttä varmuutta, oliko ohjelma jättänyt nollauksen tekemättä, vai oliko käyttöliittymään asetetut tarkennusparametrit väärät. Ohjelmalta ei myöskään saatu tietoa ulos, oliko käsky lähetetty ja vastaanotettu kameran päädyssä. Tästä syystä Pythonilla kehitettiin uusi ohjelma, jossa pystyi tarkistamaan, oliko käsky mennyt perille. Koodi nähdään kuvassa 13.

```

import requests, sys
import time
from requests.auth import HTTPDigestAuth

def main():
    ip_address = sys.argv[1]
    bigsteps = sys.argv[2]
    minsteps = sys.argv[3]

    i = 1
    while i <= 20:
        r = requests.get(f'http://{ip_address}/axis-cgi/opticssetup.cgi?rfocus=-bigstep', auth=HTTPDigestAuth('root','root'))
        print( 'i=' + str(i) )
        print( r.status_code )
        if r.status_code == 200: #200 = OK
            print('success')
            #Mitä tapahtuu kun palvelin käsittelee käskyn hyväksytysti
        else:
            print('failure')
            break
            #Mitä tapahtuu kun palvelin ei hyväksy käskyä
        i+=1
        time.sleep(0.075)

    i = 1
    while i <= int(bigsteps):
        r = requests.get(f'http://{ip_address}/axis-cgi/opticssetup.cgi?rfocus=+bigstep', auth=HTTPDigestAuth('root','root'))
        print( 'i=' + str(i) )
        print( r.status_code )
        if r.status_code == 200: #200 = OK
            print('success')
            #Mitä tapahtuu kun palvelin käsittelee käskyn hyväksytysti
        else:
            print('failure')
            break
            #Mitä tapahtuu kun palvelin ei hyväksy käskyä
        i+=1
        time.sleep(0.065)

    i = 1
    while i <= int(minsteps):
        r = requests.get(f'http://{ip_address}/axis-cgi/opticssetup.cgi?rfocus=+minstep', auth=HTTPDigestAuth('root','root'))
        print( 'i=' + str(i) )
        print( r.status_code )
        if r.status_code == 200: #200 = OK
            print('success')
            #Mitä tapahtuu kun palvelin käsittelee käskyn hyväksytysti
        else:
            print('failure')
            break
            #Mitä tapahtuu kun palvelin ei hyväksy käskyä
        i+=1
        time.sleep(0.04)

if __name__ == '__main__':
    main()

```

Kuva 13. Kameran tarkennuksen Python-koodi.

Python-koodi on yksinkertainen. Se lähettää kameralle yhden neljästä html-komennoista, jotka ovat big- ja minstep sisään ja ulos. Komennon lähetyksen jälkeen ohjelma saa palvelimelta kuittauksen, että komento on lähetetty. Komennon onnistuessa ohjelma tulostaa "success" ja epäonnistuessa "failure". Ohjelma alkaa aina lähettämällä 20 -bigstep-komentoa. Tämä toimii kameran fokuksen nollauksena. Nollauksen jälkeen ohjelma lähettää +bigstep- ja +minstep -komennot käyttöliittymän parametrien mukaan. Ohjelma myös lähettää komentoja niin kauan kunnes ne ovat onnistuneesti menneet läpi. Python ohjelma käynnistetään Tia-portaalin kautta.

## Käyttöliittymä

Käyttöliittymä suunniteltiin niin, että se on helposti ymmärrettävissä ja muokattavissa. Vasemmassa reunassa nähdään Fokusointi 1, 2 ja kameran liipaisu (kuva 14). Tekstien alla on merkkivalot, jotka syttyvät, kun jokin askel on käynnissä. Fokusointi 1 on ensimmäisen sivun tarkennus ja fokusointi 2 on toisen sivun tarkennus. Näiden alla on testinappulat, joista voi kokeilla, että ohjelma toimii oikein ja näiden alla näkyy mikä scripti on käynnissä. Fokusointi-asetuksien tekstin yläpuolella näkyy viimeisimmän kuvatun palletin mitat ja käytetyt big- ja minstepit. Fokusointiasetuksien tekstin alapuolella nähdään fokusointiparametrit, jotka ovat täysin muokattavissa. Oikealla puolella on kaikki tarvittava tieto palletista, joka on viimeksi kuvattu. Tämän alapuolella on seuraavien pallettien mittatiedot ja millä kuljettimella ne ovat.

**CUTTING EDGE QUALITY ANALYSIS**  
Versio  
2023-02

**Fokusointi 1. kuva**  
Fokusointi 2. kuva  
Kameran liipaisu

Leveys/Pituus: 8 714 6 1 16

**Fokusointi asetukset**

	Leveys/Pituus -kuokka	BigSteps+	MinSteps+
1	0 - 400	8	9
2	400 - 450	6	8
3	450 - 500	6	6
4	500 - 550	6	4
5	550 - 600	6	2
6	600 - 650	6	2
7	650 - 700	6	2
8	700 - 750	6	1
9	750 - 800	5	8
10	800 - 850	5	6
11	850 - 900	5	3
12	900 - 950	5	1
13	950 - 1000	4	11
14	1000 - 1050	4	11
15	1050 - 1100	4	9
16	1100 - 1150	4	8
17	1150 - 1200	4	6
18	1200 - 1250	4	4
19	1250 - 1350	4	2
20	1350 - 9999	3	0

**Pallettidata**

Kollinumero: 1009913980  
Leikkuri: AL3  
Leveys: 714 mm  
Pituus: 946 mm  
Korkeus: 1377 mm  
Paino: 445 kg  
Arkkiluku: 0 kpl  
Laji:   
Tilaus:   
Osoite: Pakkaus  
Syykoodi:   
Jalkapalikka: Matala  
Lavatyyppi: Normaali  
Jalkalauta: Lyhyen sivun suuntaan  
Saatetus: Kyllä  
Liputusväli: 1  
Pakkaustapahtuma: Ei tehty

Leveys Pituus  
34N4041 985 mm 1159 mm  
34N504 600 mm 720 mm  
34N4046 0 mm 0 mm  
√ 34N4047 714 mm 946 mm

1

Ilro Aika Päivämäärä Tila Teksti Käyttää ryhmä

Kuva 14. Fokusointi käyttöliittymä



## 5 Datan kerääminen ja luokittelu

### 5.1 Kuvien kerääminen

Palleteja kulkee pakkauksen läpi noin 900 kappaletta vuorokaudessa. Eli vuorokaudessa tulee 1800 kuvaa. Kuvien suuren määrän takia pyrittiin tekemään ohjelma, joka olisi tunnistanut huonon leikkausjäljen. Ohjelmaa ei saatu tämän opinnäytetyön aikana toimimaan, minkä seurauksena kuvat piti tarkistaa manuaalisesti. Kuvien manuaalinen läpikäyminen ja luokittelu vei erittäin paljon aikaa.

Kuvat kerättiin paikan päällä olevan tietokoneen kansioon, joka oli jaettuna projektissa olevan henkilöstön kanssa helpon pääsyn vuoksi. Aluksi kansioon tallennettiin kuvat, jotka olivat nimetty vuoden, kuukauden, päivän ja kellonajan mukaan. Kuvien lisäksi kansioon tallentui tekstitiedosto, jossa oli palletti id. Palletti id:stä saa selville palletin tiedot, kuten miltä leikkurilta se on tullut. Jotta kansiota saatiin selkeytettyä, poistettiin tekstitiedosto myöhemmin kokonaan ja lisättiin kuvan nimeen suoraan palletti id:stä saatavat tiedot ja leikkaussuunta. Nimetyt kuvat siirrettiin uuteen kansioon.

Kuvien tallentamista varten tarvittiin muutama scripti, jotka hoitivat kuvien nimeämisen, tekstitiedostojen luomisen ja liittämisen oikeaan kuvaan, kuvien siirtämisen uuteen kansioon sekä vanhojen kuvien poistamisen tietokoneelta.

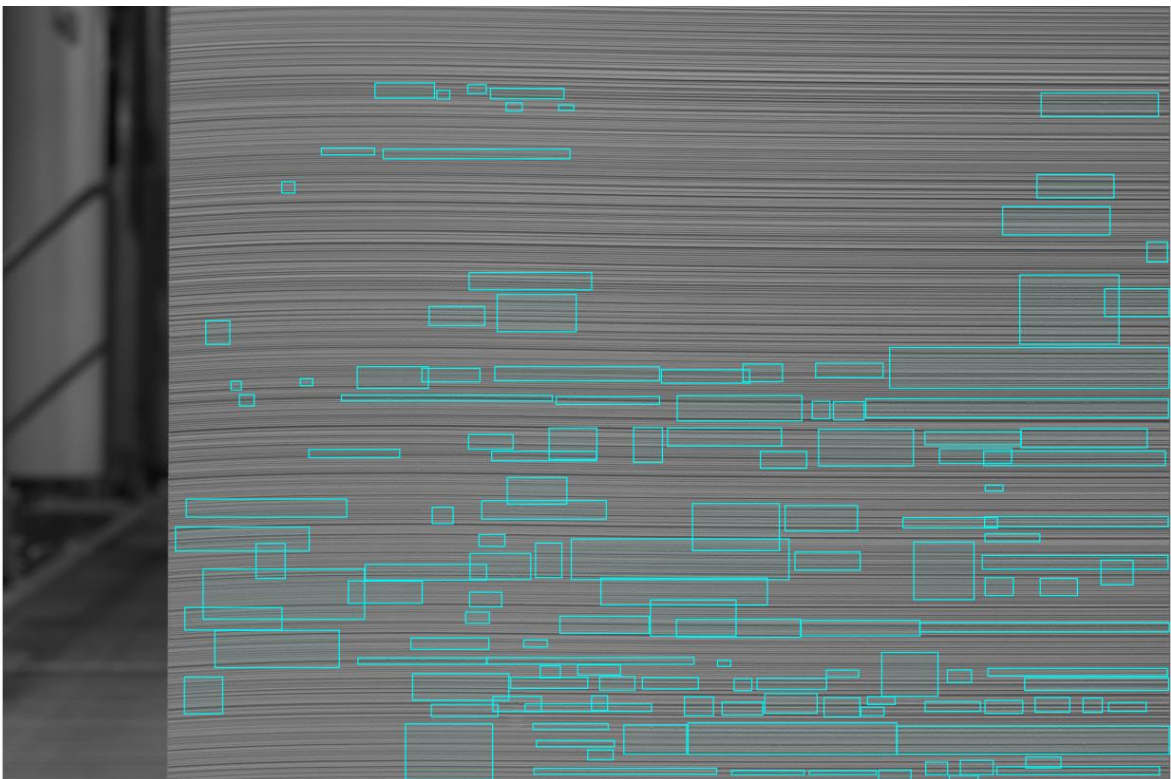
### 5.2 Stora Enso CVP

Stora Enson paperidivisioona ja ryhmät ovat luoneet alustapohjaisen lähestymistavan jakaa valmiiksi kehitettyjä konenäköalgoritmeja ja -malleja tehtaiden välillä. Alustalle pystyy myös lataamaan ja kehittämään omia konenäkömalleja. Alusta koostuu eri osista kuten kirjastosta, jonne tallennetaan valmiit konenäkömallit. New solution antaa mahdollisuuden kehittää uusia konenäkömalleja. Alustalla käytetään Visual Object Tagging Toolia (VOTT), jolla pystyy merkitsemään kuvasta tai videosta halutut kohdat, joka auttaa luomaan oman data-setin konenäköprojekteille. (Stora Enso 2023c.)

Kuvankäsittelymalleilla on kolme päämenetelmää analysoida kuvia. Tehtävät ovat objektien havaitseminen, segmentointi ja luokittelu. Objektin havaitseminen tunnistaa ja luokittelee mielenkiinnon kohteita kuvasta tai videosta täsmällisten kehikkojen kanssa. Segmentointi tunnistaa mielenkiinnon kohteita pikselitasolla ja värikoodaa kohteet omiin luokkiinsa. Luokittelu lokeroi mielenkiinnon kohteet kuvasta tai tietyn kuvan alueesta ennalta määrättyihin luokkiin. (Picsellia 2023.)

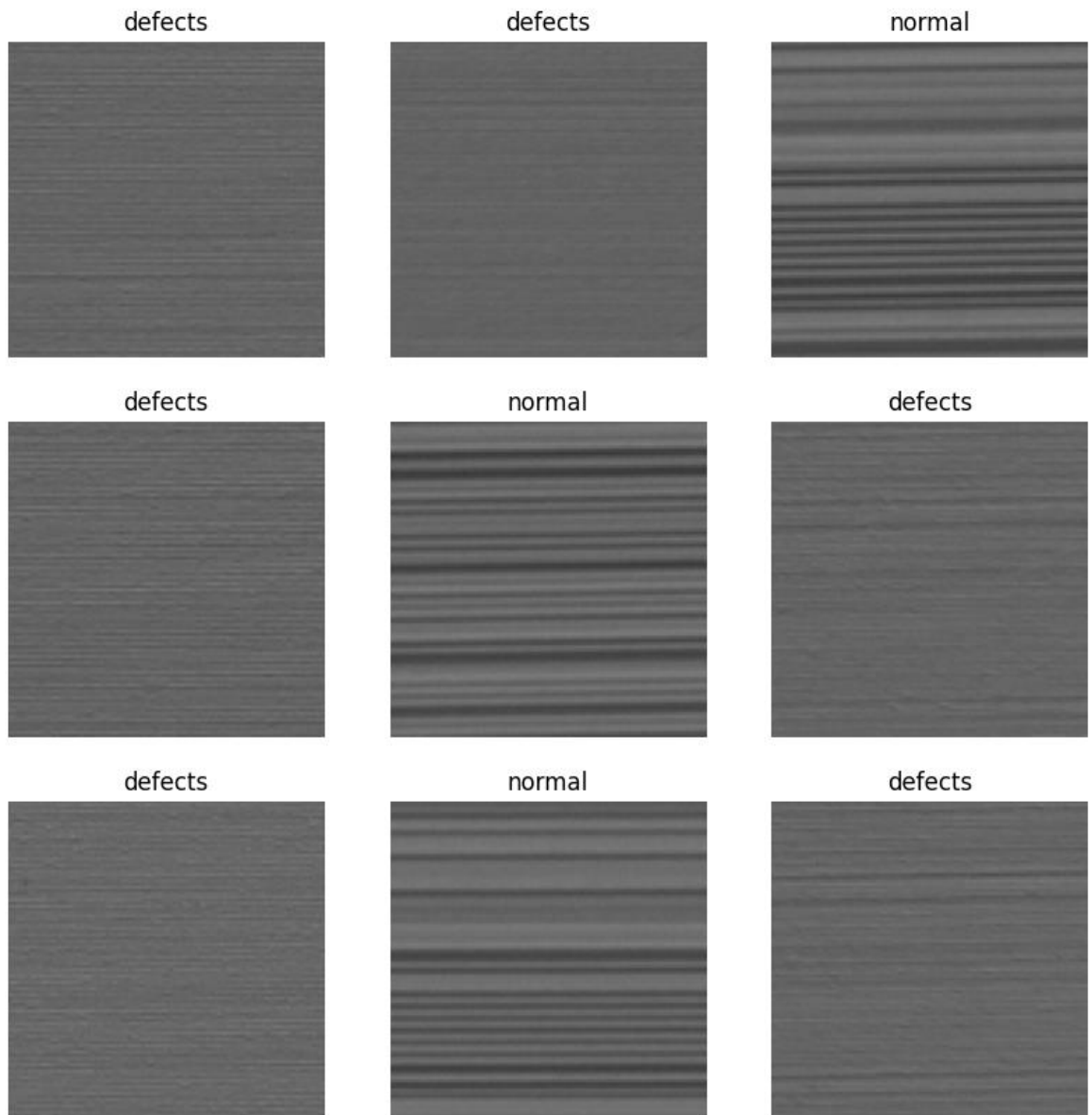


Kuvien keräämisen ja luokittelun jälkeen kuvista merkittiin epäkohdat konenäkömallille Stora Enson CVP:ssä, joka voidaan nähdä kuvassa 15. Kuvien kaikkien epäkohtien oikein merkitseminen on erittäin tärkeää. Jos epäkohdat on merkitty kuvan oikeassa reunassa, mutta vasemmassa reunassa epäkohtia on jäänyt merkkäämättä, yrittää malli ymmärtää, mitä eroa näillä kahdella epäkohdalla on. Tämä voi johtaa virheisiin opetuksessa tai tulevaisuudessa. Epäkohtien helpompaa havaitsemista varten myös kuvanlaadun oli tärkeää olla mahdollisimman tarkka, jotta pienemmätkin epäkohdat erottuisivat mahdollisimman helposti.



Kuva 15. Leikkausjäljen epäkohdat merkitty.

Mallin opettaminen epäkohdat merkitsemällä ei tuottanut haluttua tulosta. Kuvat kokeiltiin merkitä useaan otteeseen uudelleen, mutta mallin oppiminen jäi alle halutun tason. Mallin opettamista varten kokeiltiin uutta lähestymistapaa, jossa noin 5500 pikseliä leveä ja 3700 pikseliä korkea kuva jaettiin 300 pikseliä leveisiin ja 300 pikseliä korkeisiin kuviin. Kuvista ei merkitty jokaista epäkohtaa vaan kuvat luokiteltiin hyvään tai huonoon leikkausjälkeen. Tämä nopeutti konenäkömallin testausta ja päästiin haluttuihin tuloksiin. Kuvien luokittelu nähdään kuvassa 16.



Kuva 16. Kuvien luokittelu.

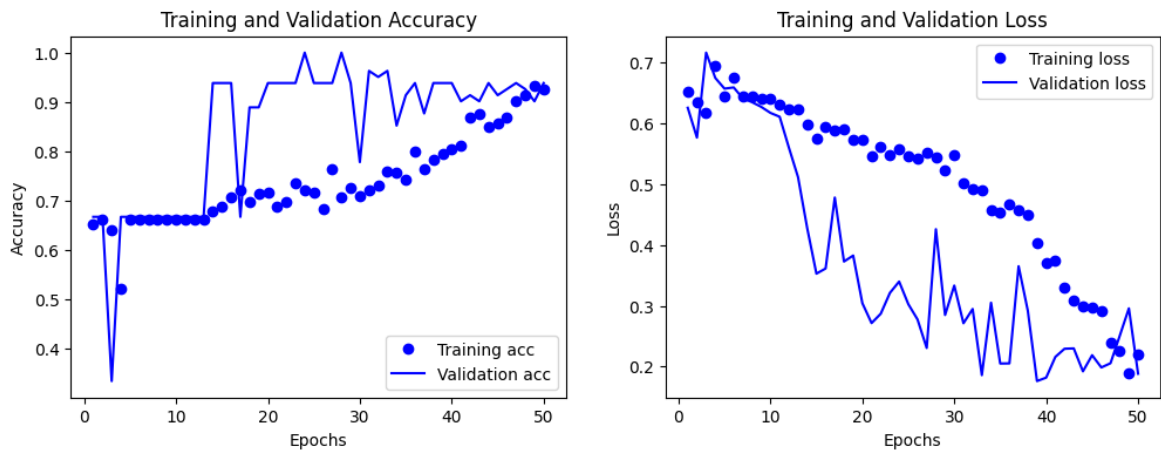
## 6 Yhteenveto

### 6.1 Konenäköjärjestelmästä saadut tulokset

Konenäkömallilta saatiin vaihtelevia tuloksia. Työn aikana malli saatiin tunnistamaan huono leikkausjälki 99 %:n tarkkuudella tietyillä kuvilla, mutta uudella kuvasarjalla oikeintunnistus laski jopa 0 %:iin. Yksikään erilaisista lähestymistavoista ei tuottanut tasaista hyvää tulosta. Syynä tälle on huonon leikkausjäljen tekemät epäkohdat, jotka ovat suhteellisen pieniä verrattuna koko kuvaan. Malli ei myöskään osannut erottaa huonoa leikkausjälkeä ja huonosta limityksestä syntyneitä tummempia viivoja. Mallille pystyi manuaalisesti opettamaan ja kertomaan, mikä on huonoa leikkausjälkeä ja näyttämään kaikki kohdat missä sitä on. Lähestymistavan ongelmana tuli epäkohtien suuri määrä. Jos mallille opetetusta kuvasta jäi näyttämättä jotakin epäkohtia, ei se silloin osannut poimia niitä myöskään muista kuvista. Malli myös ihmetteli, miksi tietty epäkohta on merkitty, mutta toinen samanlainen ei.

Kameran tarkennus loi haasteita kuvien osalta, kun se ei aina toiminut vaaditulla tasolla. Välillä kamera tarkensi liian paljon, jolloin tarkennuksen palautus ei ollut toiminut, ja välillä kameralta jäi askelia välistä. Syytä tähän ei löydetty. Kun tarkennus toimi 90 % ajasta ajateltiin, että se olisi tarpeeksi Poc-vaiheeseen. Tämä kuitenkin tarkoitti, että noin 10 % kuvista olisivat käyttökelvottomia huonon tarkennuksen vuoksi.

Konenäkömallin opetusta varten luotiin kolme datapakettia, jotka olivat opetus, testaus ja vahvistaminen. Malli saatiin tunnistamaan ja luokittelemaan opetusdata halutulla tavalla. Uusien kuvien kohdalla malli teki virheitä. Tämä johtuu pienestä opetusdatan määrästä ja siitä, että jokainen palletti on erilainen. Kuvia olisi tarvinnut tuhansia, mutta niitä oli vain satoja. Pallettien limitys, sävy, huonon leikkausjäljen sijainti, arkkikoko ja paksuus vaihtelevat pallettien välillä, mikä johtaa pallettien erilaisuuksiin, jotka heikentävät oikeintunnistusta. Mallin opetus kesti noin tunnin ja testaus oli nopeaa. Mallin oppimisen tulokset nähdään kuvassa 17. Kuvan vasemman puolen diagrammista voidaan havaita, että mallin tarkkuus on kasvanut ja malli on tunnistanut oikein enemmän kuvia. Oikean puolen diagrammista voidaan havaita, että opetusdatan virhe on myös vähentynyt.



Kuva 17. Mallin opetus- ja vahvistamistarkkuus.

## 6.2 Kehitysideat

Kameraa zoomaamalla saisi epäkohdista suuremmat, mutta kuvattu alue pienenisi, jolloin huono leikkausjälki voisi jäädä huomaamatta. Zoomaamalla kameraa joutuisi myös tarkennuksen parametrit tehdä uusiksi. Parametrien helpottamiseksi tai jopa automatisoinniksi olisi tärkeää saada tarkennuksessa tapahtuva muutos numeerisena arvona. Numeerisista arvoista voisi laskea, paljonko tarkennusta pitäisi tapahtua esimerkiksi 80 millimetrin koon vaihdoksessa.

Tarkennuksen voisi toteuttaa tekemällä kameralle automatisoidun alustan, minkä päällä kamera liikkuisi aina pallelin koon vaihdoksen mukaan, kameran tarkennuksen pysyessä vakiona. Kokeilemalla muita konenäköratkaisuja voisi löytyä tapa, joka toimisi tässä käyttötarkoituksessa. Suurella määrällä opetusdataa voisi saada konenäkömallin tunnistamaan uudet kuvat paremmin, mutta kuvia tarvittaisiin tuhansia. Tämä vaatisi suuren määrän manuaalista työtä ja aikaa.

## Lähteet

Boesch, A. Machine Vision – What you need to know (Overview). Viso.ai. Viitattu 2.11.2023. Saatavissa: <https://viso.ai/computer-vision/machine-vision/>

Ju, F., Sun, Y., Li, M. 2021. Non-parametric Bayesian dictionary learning based on Laplace noise. Multimedia Tools and Applications. 80. 1-15.

Kallio, J. 2018. Neuroverkot analytiikan edistäjinä. Insta. Viitattu 25.12.2023 Saatavissa: <https://www.insta.fi/ajankohtaista/neuroverkot-analytiikan-edistaajina>

Meyer & Summersby. 2023. All about aperture. Canon-Europe. Viitattu 27.12.2023. Saatavissa: <https://www.canon-europe.com/pro/infobank/aperture/>

Nasim, M. 2023. Understanding Aperture in Photography. photographylife. Viitattu 21.12.2023. Saatavissa: <https://photographylife.com/what-is-aperture-in-photography>

Nvidia. 2023. Convolutional Neural Network (CNN). Viitattu 27.12.2023. Saatavissa: <https://developer.nvidia.com/discover/convolutional-neural-network>

Picsellia. 2023. Segmentation vs Detection vs Classification in Computer Vision: A Comparative Analysis. Viitattu 3.1.2023. Saatavissa: <https://www.picsellia.com/post/segmentation-vs-detection-vs-classification-in-computer-vision-a-comparative-analysis>

Ratan, P. 2023. What is the Convolutional Neural Network Architecture? Viitattu 27.12.2023. Saatavissa: <https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/>

Ruokolainen, L. 2020. Konenäkö ja Kuvantunnistus. Bilot.group. Viitattu 2.11. Saatavissa: <https://bilot.group/fi/articles/konenako-ja-kuvantunnistus/>

Scandinavianphoto. 2023. Suljinaika – miten kameran suljin toimii. scandinavianphoto. viitattu 18.12. Saatavissa: <https://www.scandinavianphoto.fi/tietopankki/suljinaika>

Siersbaek, P. 2019. Näin käytetään kameran ISO-asetuksia. Digikuva. Viitattu 10.12.2023. Saatavissa: <https://digi-kuva.fi/valokuvaustekniikka/nain-kaytetaan-kameran-iso-asetuksia>

Stora Enso. 2022. Inkeröisten Kartonkitehdas. Stora Enso WeShare

Stora Enso. 2023a. Historia. Viitattu 30.12.2023. Saatavissa: <https://www.storaenso.com/fi-fi/about-stora-enso/our-history>

Stora Enso. 2023b. Tietoa Stora Ensosta. Viitattu 30.12 Saatavissa:

<https://www.storaenso.com/fi-fi/about-stora-enso>

Stora Enso. 2023c. CVP. Stora Enso WeShare

Vorenkamp, T. 2018. Understanding Exposure, part 3: Shutter Speed. Viitattu 20.12.2023.

Saatavissa: <https://www.bhphotovideo.com/explora/photography/tips-and-solutions/understanding-exposure-part-3-shutter-speed>

Winter, A. 2023. Tekoäly ja neuroverkot. Viitattu 3.11.2023 Saatavissa:

<https://www.itewiki.fi/p/tekoaly-ja-neuroverkot>