

# Koneälyn käyttö 3D-tulostuksessa virhetilanteiden tunnistamiseksi

TURUN YLIOPISTO  
Tietotekniikan laitos  
TkK-tutkielma  
Tietotekniikka  
Tammikuu 2024  
Jere Kivelä

TURUN YLIOPISTO  
Tietotekniikan laitos

JERE KIVELÄ: Koneälyn käyttö 3D-tulostuksessa virhetilanteiden tunnistamiseksi

TkK-tutkielma, 23 s.  
Tietotekniikka  
Tammikuu 2024

---

3D-tulostus on moderni ja jatkuvasti yleistynyt valmistustapa teollisuudessa ja yksityiskäytössä. Tässä kandidaatintutkielmassa tarkastellaan 3D-tulostuksen virhetilanteita ja niiden tunnistukseen kehitettyjä menetelmiä. Tutkielmassa käsitellään yleisimpien virhetilanteiden syntymistä ja tarjotaan mahdollisia korjauskeinoja. Keskeisenä aiheena on konenäön ja koneälyn toimintaperiaatteet sekä niiden soveltaminen 3D-tulostuksessa. Tutkielma käy läpi eri tutkijaryhmien tuloksia tunnistuksen alalta, korostaen sitä, että ei ole olemassa yhtä oikeaa menetelmää tunnistukseen. Jokaisella menetelmällä on omat vahvuutensa ja heikkoutensa, mikä tekee niistä soveltuvia erilaisiin käyttökohteisiin aina yksittäisen henkilön tarpeista suuriin tulostusoperaatioihin.

Asiasanat: 3D-tulostus, koneäly, konenäkö, virhetilanteet

# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Virhetilanteet</b>	<b>4</b>
2.1	3D-tulostuksen perusteita . . . . .	4
2.2	Virhetilannetyypit . . . . .	5
2.3	Virhetilanteen korjaus . . . . .	7
<b>3</b>	<b>Koneälyn käyttö ja koulutus</b>	<b>9</b>
3.1	Virhetunnistus koneälyllä . . . . .	9
3.2	Koneälyn koulutus . . . . .	10
3.3	Renderöintipohjainen tunnistus . . . . .	12
3.4	Muita algoritmeja . . . . .	13
<b>4</b>	<b>Koneälyn tarkkuus ja vertailu</b>	<b>15</b>
4.1	Tarkkuuden määritelmä . . . . .	15
4.2	Menetelmien vertailu . . . . .	16
<b>5</b>	<b>Pohdintaa</b>	<b>21</b>
<b>6</b>	<b>Yhteenveto</b>	<b>22</b>
	<b>Lähdeluettelo</b>	<b>24</b>

# Kuvat

2.1	Esimerkki spagetin ulkonäöstä. . . . .	6
2.2	Esimerkki reunan nousemisesta. . . . .	6
2.3	Aliekstruusio esimerkki. . . . .	7
3.1	Matriisi . . . . .	11
3.2	Koontikerros . . . . .	11
4.1	Visuaalinen esimerkki YOLO-algoritmista. Algoritmi on jakanut kuva aluksi osiin, jonka jälkeen se on tunnistanut elementtejä kuvasta. . . .	17
4.2	Visuaalinen esimerkki tukivektorikoneen perustasta . . . . .	18
4.3	Visuaalinen esimerkki HOG muunnoksen piirteistä . . . . .	20

# 1 Johdanto

Materiaalia lisäävä valmistus eli 3D-tulostus on vielä uusi menetelmä teollisuudessa. 3D-tulostuksen etuna normaaleihin valmistustapoihin verrattuna on tuotteiden helppo muokattavuus toiveiden mukaiseksi. Myös pienissä tuotantoerissä menetelmä on korvaamaton, kun suhteutetaan tarvittavan alkuinvestoinnin määrää. [1] Lisäävässä valmistuksessa on kuitenkin haasteena laadun tarkkailu. Tulostamisessa voi syntyä ongelmia, ja nämä on pystyttävä havaitsemaan ajoissa, jotta vältytään ylimääräisiltä materiaalikustannuksilta ja säästetään tuotantoajassa. Tätä ongelmaa varten on koulutettu koneälyä, jolla saadaan lisättyä tulostuksen automaatiota sekä vähennettyä materiaali- ja aikahukkaa. [2] Työssä keskitytään pääasiassa FDM-tulostukseen (engl. fused deposition modeling), jossa materiaalia lisätään toisen materiaalin päälle sen ollessa alustalla. Tämä on yleisin 3D-tulostuksessa käytetty tulostusmenetelmä. [3]

Tämä tutkielma pyrkii tunnistamaan 3D-tulostuksessa tapahtuvia virhetilanteita ja niiden syitä. Käydään läpi myös miten koneälyä käytetään näiden tilanteiden tunnistamiseksi. Työssä perehdytään myös siihen, miten koneäly toimii ja miten sitä koulutetaan. Tämän tutkielman tutkimuskysymykset ovat siis seuraavat:

1. *Minkälaisia virhetilanteita kohdataan ja miten niihin reagoidaan?*
2. *Kuinka hyödynnämme koneoppimista niin että se luotettavasti pystyy tunnistamaan nämä virhetilanteet?*

Tutkielman tulokset ovat tärkeitä 3D-tulostuksen yleistyessä niin massatuotan-

nossa kuin yksityisten henkilöiden käytössä, jotta automaation määrää voitaisiin lisätä ja vähentää mm. materiaali- ja aikahävikkiä.

Kysymyksiin on pyritty vastaamaan tutkielman aineiston avulla, joka on saatu suorittamalla kirjallisuuskatsaus seuraavan prosessin avulla. Tiedonhaku suoritettiin syksyllä 2023 käyttäen hakulausetta (Machine Learning OR Image Processing AND 3D Print\* AND Defect OR Error). Hakulausetta käytettiin Google Scholarissa, joka antoi tuhansia aiheeseen sopivia artikkeleita. Samoja avainsanoja käytettiin myös Web of Science- ja IEEE Xplore -tietokannoista, joista löytyi satoja artikkeleita tuhansien sijasta. Artikkeleita tutkimalla kävi kuitenkin selväksi, että monet artikkelit löytyivät myös Google Scholarin kautta. Scholarista haettiin myös muutamaa otteeseen yleistä tietoa koskien 3D-tulostusta ja sen osia, mutta päälähteet ovat kuitenkin löytyneet edellä mainitulla haulla.

Artikkelit rajattiin aluksi sopivan otsikon perusteella. Valituista artikkeleista luettiin tiivistelmä ja käytiin sisältö läpi silmäillen. Tiivistelmän pohjalta valittiin artikkelit, joita luettiin tarkemmin. Pidemmän lukemisen pohjalta päädyttiin tutkielmassa käytettyihin artikkeleihin.

Luvun kaksi kuvaa ovat itse otetut. Kuvassa näkyvän tulostimen malli on Bambu Lab P1P, ja materiaalina käytettiin Bambu Labin valkoista PLA-muovia. Tulostetut mallit ovat vapaasti saatavilla olevia, ei-kaupallisiin tarkoituksiin tehtyjä malleja, jotka valittiin niiden ominaisuuksien vuoksi. Luvun kolme kaaviota on luotu itse materiaalin pohjalta.

Luvussa 2 tarkastellaan yleisimpiä virhetilanteita ja sitä, mistä ne tunnistaa. Käydään myös läpi reagointi näihin virhetilanteisiin: onko mahdollista jatkaa tulostusta vai tarvitaanko toimenpiteitä, jotta laatu pidetään hyvänä? Luvussa 3 käsitellään, miten koneäly tunnistaa virhetilanteen ja mitä sen implementointi vaatii. Miten koneäly on koulutettu niin, että tunnistus on luotettavaa ja tarpeeksi tarkkaa? Luku 4 keskittyy koneällyn virhetilanteiden tarkempaan määrittelyyn 3D-

tulostuksessa. Tarkastellaan myös esimerkkien pohjalta eri tapojen soveltuvuutta tunnistuksessa ja vertaillaan niitä keskenään. Viidennessä luvussa käyn läpi omaa pohdintaani aiheeseen liittyen, ja kuudes luku on yhteenveto.

## 2 Virhetilanteet

3D-tulostus on kerroksittaista materiaalin lisäämistä, joten on tärkeää, että kerroksissa ei tapahdu ongelmia. Kerroksessa aiheutuneet ongelmat saattavat aiheuttaa ongelmia seuraaviin kerroksiin tai jopa koko tulostettavan tuotteen hylkäämisen. Yleisesti ongelmat tapahtuvat kesken tulostusprosessin, jolloin haaskaantuu paljon aikaa ja materiaalia, jos tulostusta ei pysäytetä vaan sen annetaan jatkua loppuun asti. [4] Tarkastellaankin seuraavaksi 3D-tulostuksen perusteita ja sitten yleisimpiä ongelmia perinteisessä 3D-tulostuksessa.

### 2.1 3D-tulostuksen perusteita

3D-tulostus on syntynyt tarpeesta saada tehtyä monimutkaisempia kappaleita, joita on ollut mahdotonta tai erittäin vaikeaa valmistaa perinteisillä valmistustavoilla. Lisäävä valmistus on alusta asti ollut hyvä työkalu tutkijoille prototyyppien tekemiseen, koska se mahdollistaa suhteellisen nopean tavan saada fyysinen kappale ilman erikoiskoneistamista. Lisäksi 3D-tulostus mahdollistaa pienten tuotantoerien valmistuksessa edullisemmat kustannukset verrattuna perinteiseen ruiskupuristukseen. [5]

FDM-tulostus perustuu siihen, että materiaalia syötetään lämpötilaltaan hallittuun suuttimeen, josta se tulee sulaneena ja voidaan tarkasti syöttää ohuena kerroksittain muodostaen lopulta kappaleen. [5] Kappale määräytyy sen CAD-mallin (eng. Computer-aided Design) mukaan. CAD malli pitää kuitenkin muuttaa tulostimelle tarkoitettussa ohjelmistossa g-koodiksi. Ohjelmistossa voi säätää tulostuspara-



metreja, kuten kappaleen täytön ja tulostuslämpötilan halutuksi. Ohjelmisto muuntaa 3D-mallin kerroksiksi ja kerroksien sisällä radoiksi, joita suutin seuraa. G-koodi sisältää siis liikeradat, joita tulostin ymmärtää seurata vaihe vaiheelta, toistaen kappaleen muodon alkuperäisestä CAD-mallissa. [6]

Varsinkin FDM-tulostuksessa on nykyisin todella monia erilaisia muoveja käytössä. Jokaisella muovilla on omat heikkoutensa ja vahvuutensa, joiden perusteella voidaan valita oikeanlainen muovi jokaiseen käyttötarkoitukseen. Muovien eroavaisuudet liittyvät muun muassa sulamislämpötilaan, kerrosvahvuuteen, joustavuuteen ja tietysti hintaan. [5]

## 2.2 Virhetilannetyypit

Yksi ulkoisesti näyttävimmistä virhetilanteista on yleisesti tunnettu termillä "spagetti". Kyseessä on tilanne, jossa tulostin tulostaa materiaalia tyhjän päälle, muodostaen spagetin kaltaisia kasoja. Kuvassa 2.1 voidaan nähdä esimerkki kyseisestä tilanteesta, kun osa on irronnut levystä, ja tulostus on epäonnistunut. Ongelma yleisesti aiheutuu siitä, että tulostaessa uusi materiaali ei tartu edelliseen kerrokseen kunnolla, jolloin muodostuu liian suuri väli tulostettavaan pintaan. Välin ollessa liian suuri, tulostus ei enää onnistu, vaan tulostus tapahtuu nyt tyhjän päällä. Ongelma voi myös aiheutua siitä, että tulostettava kappale irtoaa alustasta, ja näin ollen päästään samaan tilanteeseen, että tulostus tapahtuu tyhjän päällä. [7]

Tuloste saattaa myös irrota kesken tulostuksen levystä. Tämä johtaa usein siihen, että tulostin jatkaa tyhjän päälle tulostamista, koska uudelleen kiinnitys ei ole mahdollista. Ongelma johtuu yleensä liian huonosta kiinnittymisestä levyyn. [7] Ongelmaa voi myös pahentaa tulosteen mahdollinen vääntyminen reunoilta sen jäähtyessä. Vääntyminen tapahtuu, kun muovi jäähtyy sulamislämpötilastaan takaisin kiinteäksi ja samalla kutistuu hieman. Kuvasta 2.2 näkee tilanteen, jossa reuna on alkanut nousemaan jo levystä kappaleen vasemmalla puolella. Tulosteen vääntymi-



Kuva 2.1: Esimerkki spagetin ulkonäös- Kuva 2.2: Esimerkki reunan nousemisesta.

nen on ongelma myös itsessään, koska vaikka tuloste ei irtoaisi levystä, se muuttaa tulosteen geometriaa niin, että se ei vastaa enää suunniteltua tuotetta mitoiltaan. [8]

Yleisiin ongelmiin kuuluu myös tulostimen suuttimen täysi tukkeutuminen tai osittainen tukkeutuminen. Kuva 2.3 voisi toimia esimerkkinä osittaisesta tukkeutumisesta, jossa tukkeuma poistuu tulostuksen aikana ja palaa myöhemmin. Tukkeutuminen voi johtaa joko koko tulostuksen loppumiseen, koska materiaalia ei tule enää ulos, tai osittaisessa tukkeutumisessa se heikentää tulosteen vahvuutta huomattavasti. Tukkeuma voi aiheutua monista eri syistä, kuten epäpuhtauksista muovissa tai ekstruuderin eli muovin suuttimeen työntävien rattaiden lipsumisesta. Tulostuksen kokonaan loppuminen voi myös johtua yksinkertaisesti muovin loppumisesta. [7]

Yli- ja aliekstruusio ovat tilanteita, joissa muovia tulee liikaa tai liian vähän optimaaliseen verrattuna. Molemmat tilanteet vaikuttavat kappaleen pinnan laatuun ja kestävyys. Aliekstruusio saattaa pahimmassa tapauksessa muistuttaa osittain tukkoista suutinta tai yleisemmin jättää kerroksien väliin tyhjää tilaa, kun muovi ei täytä tarvittavaa aluetta kokonaan. Tämä tilanne on nähtävissä kuvassa 2.3 sen ylä- ja alaosassa. Yliekstruusiossa tyhjien välien sijasta muovia tulee liikaa, jolloin



Kuva 2.3: Aliekstruusio esimerkki.

tuloksesta tulee todella epätasainen, koska ylimääräisen muovin täytyy laajentua johonkin suuntaan. Samalla tavalla muovissa oleva piilevä kosteus voi vaikuttaa tilanteeseen. Kosteus muovissa aiheuttaa ongelmia, koska suuttimeen tullessa piilevä kosteus höyrystyy kuumassa suuttimessa. Höyrystyminen vaahdottaa muovia, luoden epätasaisuuksia ja heikentäen tulostetta. [8] [9]

## 2.3 Virhetilanteen korjaus

Varsinkin yli- ja aliekstruusion tapauksissa oleellisinta on kalibroida muovin virtaus niin, että saadaan ideaali nopeus. Tähän myös useasti auttaa tulostuslämpötilan säätö. Lämpötilan nosto auttaa lisäämään virtausta, koska se mahdollistaa nopeamman sulatuksen ja isomman materiaalivirtauksen, kun taas lämpötilan laskeminen voi auttaa laskemaan virtausta. Samoin myös ekstruuderin säätäminen niin, että se työntää uutta muovia suuttimeen vain halutun määrän. Yleensä halutaan tulostaa mahdollisimman nopeasti, joten on tärkeä löytää asetukset, jotka eivät kuitenkaan

ole liian hitaat koneen suorituskykyyn nähden, jolloin voidaan tulostaa mahdollisimman tuottavasti ja laadukkaasti. [9]

Muovin hyvä kiinnittyminen tulostuslevyyn on todella oleellista, kun halutaan välttyä ongelmilta. Levyn on tärkeää pysyä puhtaana, jotta epäpuhtaudet eivät vaikuta kiinnittymiseen. Samoin levyn tasapainon ylläpitäminen on merkittävää, jotta siihen tulostettava muovi olisi tasapaksua riippumatta siitä, missä kohtaa levyä tulostetaan. Lisäksi tulostuskorkeuden kalibrointi on tärkeää niin, että tulostin ei tulosta liian lähellä levyä tai liian kaukana levystä, jolloin kiinnittyminen on optimaalinen sekä tulostuksen että irrotuksen kannalta. Edellä mainitut toimenpiteet ennaltaehkäisevät tulosteen irtoamista, mikä puolestaan estää vääntymistä ja spagetin muodostumista. [10]

## 3 Koneälyn käyttö ja koulutus

3D-tulostuksen yleistyessä on tullut tärkeäksi keksiä tapa monitoroida tulostusprosessia automaattisesti. Yksi käytetyimmistä ja tutkituimmista tavoista on koneälyn käyttö automaation lisäämiseksi, koska se on verrattain helppo tapa lisätä osaksi tulostusprosessia omana komponenttinaan, mutta kuitenkin irrallaan varsinaisesta tulostimesta. Tarkastellaan, mitä tarvitaan koneälyn käyttöönottoa varten ja miten sitä koulutetaan.

### 3.1 Virhetunnistus koneälyllä

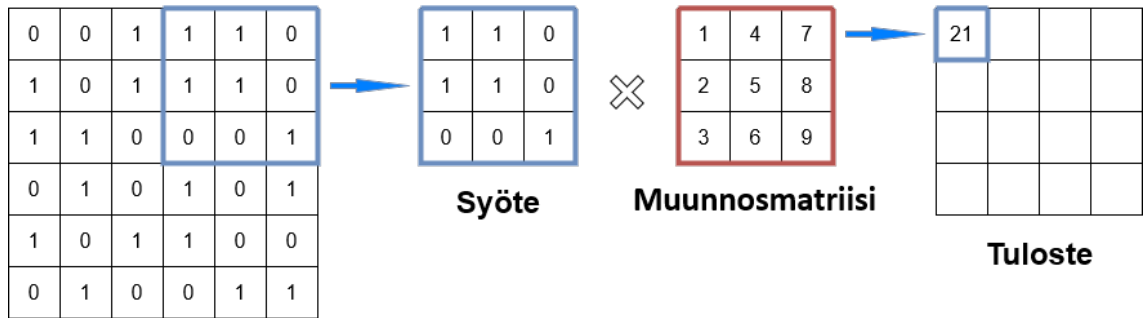
Yleisin tapa toteuttaa konenäkö tulostimeen varsinkin yksityisten henkilöiden käytössä on asentaa tulostimeen erikseen kamera, joka ottaa kuvia kesken tulostusprosessia. Kameraksi riittää normaali webkamera. Korkea resoluutio kamerassa vain lisääisi tarvittavaa prosessointiaikaa ilman suurta hyötyä tarkkuuteen, koska tarvittavat datamäärät kasvaisivat huomattavasti. Otettujen kuvien resoluutiota saatetaankin pienentää jälkikäteen ennen prosessointia juuri sen takia, että prosessointiaikaa saataisiin laskettua huomattavasti. [11] Tärkeää on vain pitää kamera vakaana, jotta se ei aiheuta turhia ongelmia tarkkuudessa. Tarvitaan myös riittävän hyvä valaistus, jotta varsinaiset muodot on helpompi tunnistaa, eikä turhia varjoja ole häiritsemässä prosessia. Tarvitaan myös erikseen oma tietokone, joka on riittävän nopea käymään kuvat läpi kohtuullisessa ajassa. Kuvat ajetaan konvoluutioverkon läpi ja koulutettu algoritmi päättää, onko kuvassa virhetilanne vai ei. [1]

Hyvin koulutetusta koneälystä voidaan saada korvaamatonta apua tulostukseen. Sen lisäksi, että helposti tunnistettavat ongelmat, kuten tulosteen irtoaminen tai spagetin muodostuminen, voidaan tunnistaa, voidaan sitä käyttää myös pienempien ongelmien havaitsemiseen ja parantamiseen. Esimerkiksi yli- ja aliekstruusio tulostuksessa voidaan tunnistaa. Hyvin integroidussa tulostimessa nämä ongelmat voidaan korjata kesken tulostuksen säätämällä tulostimen muovin virtausta. Ekstruusio-ongelmien havaitseminen voi olla vaikeaa ihmissilmällä tulostamisen aikana, joten koneäly avaa tässä suhteessa uusia mahdollisuuksia tulostuksen laadun parantamiseen ja valvomiseen. [11]

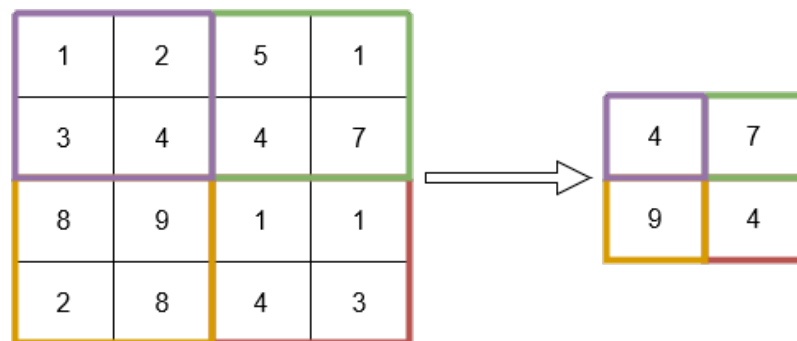
Varsinaisen tekoälyn käyttö perustuu siihen, että otetaan kuvia tulostusprosessin aikana. Otetut kuvat syötetään ennalta koulutettuun konvoluutioneuroverkkoon. Neuroverkon koulutus mahdollistaa sen, että verkon läpi syötetyt kuvat palauttavat arvon. Tätä arvoa voidaan verrata muihin tilanteisiin ja määrittää, onko kyseessä ongelma tai millainen ongelma on kyseessä. Tietyn arvon palautus voi esimerkiksi lisätä virtausta, jos ongelma havaitaan, tai olla tekemättä mitään, jos arvo on normaalin sisällä. Verkon koulutus määrittää, mitä ongelmia voidaan huomata, ja kuinka hyvälaatuinen tunnistus on. [11] [1]

## 3.2 Koneälyn koulutus

Neuroverkon käyttö tunnistamisessa vaatii sen kouluttamisen haluttuun tehtävään. Otetuista kuvista voidaan poistaa väri. Väriin poisto on hyvä tehdä, jotta tunnistus onnistuisi kaikilla tulostemateriaalien väreillä ja helpottaakseen tunnistamisen laatua, kun kaikki käsittely ja koulutus on tehty harmaasävyisillä kuvilla. Myös harmaasävyttämisen etuna on datamäärän pienentyminen, mikä puolestaan nopeuttaa prosessointiaikaa. Haastavin osa kuitenkin on antaa neuroverkolle tarpeeksi dataa, jotta voidaan luotettavasti nojata sen antamiin tuloksiin. Hyvän tuloksen saamiseksi voidaan tarvita tuhansia viitteitä. Neuroverkot toimivat paremmin isomman koulu-



Kuva 3.1: Matriisi



Kuva 3.2: Koontikerros

tusaineiston kanssa. Tätä prosessia voi myös laajentaa, jos on tarve tunnistaa useita eri virhetilanteita samalla neuroverkolla. [11]

Konvoluutioneuroverkko on merkittävin osa virhetunnistusta, joten käydään läpi paremmin sen toimintamallia. Ensimmäinen kerros on varsinainen konvoluutiokerros. Konvoluutiokerros ottaa syötteenä kuvan ja syöttää sen matriisiin läpi. Matriisi käyttää matemaattisia funktioita, jotka ottavat pikseliryhmän ja muuttavat sen pienemmäksi yhdistetyksi osaksi funktioiden pohjalta. Prosessin voi nähdä kuvassa 3.1. Konvoluutiokerros säilyttää kuitenkin tarvittavan datan kuvasta, mutta tekee siitä helpommin käsiteltävän seuraaville vaiheille. [1]

Seuraavana kerroksena toimii koontikerros. Koontikerros ei itsessään tuo lisää parametrejä kuvaan. Tarkoitus tässä vaiheessa on pienentää kuvan kokoa yhdistämällä kokonaisuuksia pienemmäksi. Koonti ottaa mallista osan ja tiivistää sen pienemmäksi. Tiivistyksestä otetaan mallin osasta suurin arvo talteen ja poistetaan

loput. Kuvassa 3.2 on esimerkki tiivistyksestä. Tiivistykseen valitun alueen koko vaikuttaa siihen, kuinka tarkka lopputulos on, koska suuri tiivistys saattaa heikentää mallin tarkkuutta. Tiivistyksen pääasia on kuitenkin vähentää datan määrää ja näin nopeuttaa prosessointiaikaa. 2x2 kokoinen pienennys kokoon 1x1 saattaakin jo pienentää datan määrää jopa 75% ilman suuria haittavaikutuksia, koska viereisten pikselien merkitys tärkeään dataan ei ole niin suuri. [1]

Viimeisenä kerroksena on täysin yhdistetty kerros. Yhdistetyn kerroksen tavoite on muuttaa edellisestä kerroksesta saatu malli yksittäisen sarakkeen kokoiseksi vektoriksi. Malli käytetään funktion läpi, joka hoitaa muutoksen. Mallin muuntaminen yksittäisen sarakkeen kokoiseksi vektoriksi on tärkeä osa prosessia, koska se toimii syötteenä seuraaville edessä oleville neuroverkon osille. Neuroverkko on nimensä mukaisesti täysin yhdistetty, jos sen solmut ovat yhteydessä kaikkiin seuraavan kerroksen solmuihin. Neuroverkon lopussa saadaan jokin loppuarvo tulokselle. Lopputulosta voidaan verrata haluttuun arvoon ja tulon erotuksella voidaan kouluttaa mallia paremmaksi käyttämällä vastavirta-algoritmia. [1]

### 3.3 Renderöintipohjainen tunnistus

On myös kehitetty erilainen tapa arvioida tulostuksen virheellisyyttä. Sen sijasta, että kuvaa käydään läpi ja todennetaan virheellisyys vain ennalta koulutetulla algoritmilla, voidaan jokaista kerrosta verrata virheettömään esimerkkiin. Tämän toteutus eroaa perinteisemmästä tunnistamisesta niin, että siinä käytetään luotua g-koodia, joka sisältää tulostimen tulostukseen tarvittavat liikeradat. G-koodi syötetään johonkin mallinnusohjelmaan niin, että se muodostaa digitaalisen mallin kyseisestä mallista juuri niin kuin tulostin sen tulostaisi. Nyt voidaan tulostuksessa otettuja kuvia verrata suoraan kerros kerrokselta malliin, joka on hyvin lähellä todellisuutta. [12] Jonkin verran yksinkertaisempi lähestyminen on myös verrata valmista kappaletta tehtyyn malliin. Silloin ei ole mahdollista verrata kerroksittain mallia, mutta



voidaan säästää prosessointi- ja tarkistusaikaa. [13]

Etuna tässä lähestymistavassa on todella hienojakoisenkin eroavaisuuden tunnistus. Varsinaista virhetunnistusalgoritmia ei tarvitse kouluttaa niin paljon kuin perinteisempää lähestymistapaa käyttäen. Negatiivisena puolena kuitenkin digitaalisen mallin luominen vaatii paljon prosessointiaikaa ja tehoja, varsinkin jos halutaan se tehdä reaaliajassa. On myös oltava todella tarkkana kameran asennuksessa ja valaistuksessa, jotta se saadaan pysymään tasaisena. Digitaalinen malli pitää myös saada säädetyksi ohjelmistossa niin, että se vastaa samaa katselukulmaa kuin fyysinen kamera, jotta vertailu olisi ylipäättään mahdollista. [12]

### 3.4 Muita algoritmeja

Kuvan tunnistuksessa on käytössä monia erilaisia menetelmiä kuvankäsittelyyn. Eri menetelmillä on omat positiiviset ja negatiiviset vaikutuksensa, jotka määrittävät niiden soveltuvuuden erilaisiin käyttötarkoituksiin. Kuvaa voidaan esimerkiksi käsitellä konvoluutioverkon avulla, mutta on myös muita tapoja toteuttaa kuvantunnistusta.

Nopeaan ja kevyempään tunnistukseen voi käyttää YOLO-pohjaista algoritmia (engl. You Only Look Once). YOLO hyödyntää neuroverkkoa, mutta sen suunnittelussa on pyritty nopeuteen ja keveyteen suurempien kokonaisuuksien tunnistamisessa kuvasta. YOLO käyttää kohtalaisen yksinkertaista konvoluutioverkkoa, jossa kerroksia ei ole kovin monta, mikä edistää prosessointinopeutta. [14]

Tunnistuksessa voidaan myös hyödyntää tukivektorikonetta (engl. support-vector machine). Kuva pitää aluksi muuttaa piirrevektoriksi, jotta sitä voidaan käyttää menetelmässä. Tukivektorikone käyttää matemaattisia vektoreita, joihin määritellään haluttu virhemarginaali. Kun tarpeeksi moni datapiste ylittää marginaalin, tilanne voidaan tulkita virheeksi. Tukivektorikone on jo huomattavasti tarkempi tunnistuksessaan, mutta tämä tarkkuus saavutetaan lisäämällä algoritmin raskautta, ja se

vaatii enemmän resursseja toimiakseen. [15]

Piirrevektoreiden tekemiseen voidaan käyttää HOG-pohjaista algoritmia (engl. Histogram of Oriented Gradients). HOG muodostaa kuvasta histogrammin, jonka avulla kuvan piirteitä voidaan tunnistaa käyttämällä jotain muuta luokittelijaa, kuten edellä mainittua tukivektorikonetta. Muunnettu kuvadata helpottaa piirteiden tunnistamista muille algoritmeille. HOG-algoritmin muokkaaman kuvan avulla voidaan laskea tarvittavaa prosessointi aikaa seuraaville algoritmeille. [16]

## 4 Koneälyn tarkkuus ja vertailu

Eri menetelmillä on omat vahvuutensa ja heikkoutensa, ja niiden soveltuvuus riippuu käytettävän järjestelmän vaatimuksista ja tarkoituksesta. Jokaisella menetelmällä on omat erityispiirteensä, ja valintaan vaikuttavat tekijät voivat sisältää nopeuden, tarkkuuden, resurssien tarpeen ja koulutusprosessin monimutkaisuuden. On tärkeää arvioida tarkasti, mikä menetelmä sopii parhaiten tiettyyn käyttötarkoitukseen ja millaisia kompromisseja ollaan valmiita tekemään. Lisäksi on hyvä ottaa huomioon, että mikään tunnistusjärjestelmä ei ole täysin virheetön, ja sen suorituskykyyn vaikuttavat monet tekijät.

### 4.1 Tarkkuuden määritelmä

3D-tulostuksen virhe ei aina ole yksinkertainen kyllä tai ei -tilanne. Todellisuudessa siinä on paljon välimaastoa ennen kuin hyvästä tulostuksesta päädytään virheeseen. Menetelmiin tarvitsee siis määritellä sopivalla tarkkuudella, mistä virhe oikeastaan alkaa [2]. Kuten Vaezin ja Chuanin tutkimuksessa on todettu, suurempi saturaatio kerrosten välissä lisää tulosteen kestävyyttä, mutta samalla heikentää pinnan laatua. Myös kerrospaksuuden lisääminen samassa saturaatiossa saattaa kasvattaa vetolujuutta, mutta heikentää taivutuslujuutta. Virhe voi olla jossain paikoissa suurenkin marginaalin sisällä [17]. Toisaalta virhe saattaa olla kohtalaisen yksinkertainen kyllä tai ei, kuten esimerkiksi katsottaessa, onko tuloste irronnut levystä.

Menetelmien vertailussa pitääkin olla valppaana, sillä eri menetelmillä voi olla

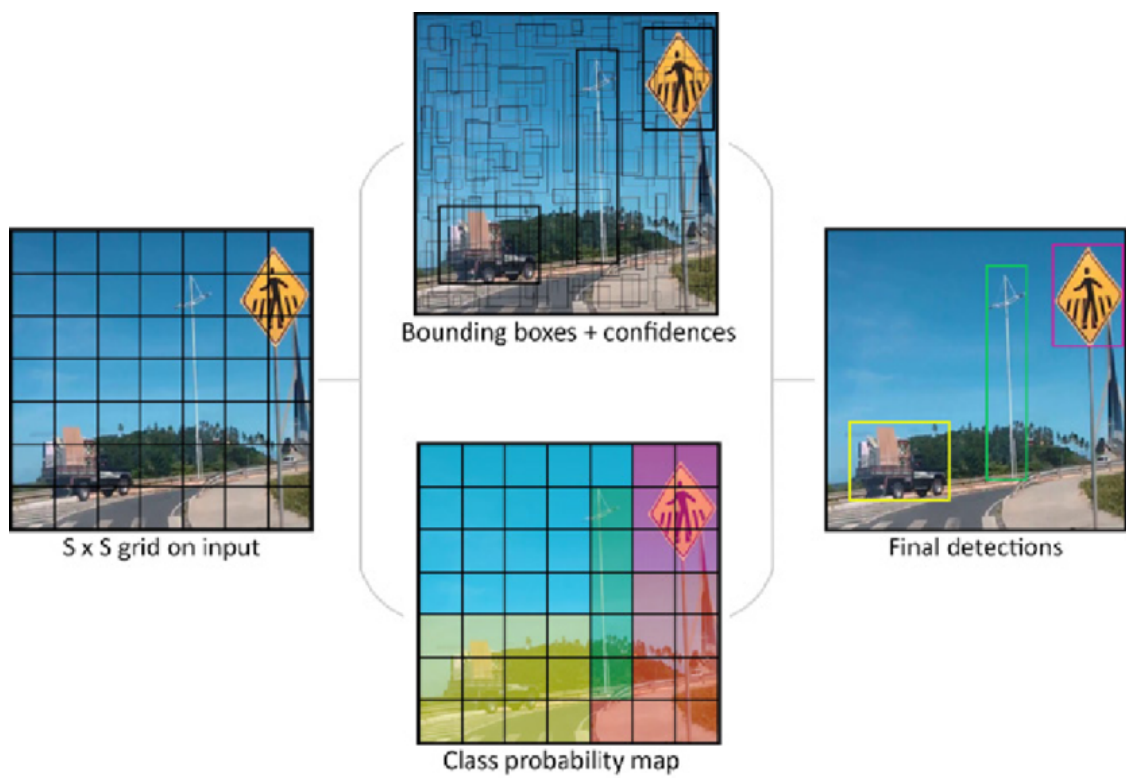
erilainen määritelmä virheelle. Toinen saattaa määrittää, että pienet kosmeettiset virheet eivät vielä tee tulosteesta virheellistä, kun taas toisen mielestä pienikin kosmeettinen virhe tekee tulosteesta virheellisen. Hyvän menetelmän merkkipaalu on kuitenkin se, että se tunnistaa omissa parametreissään virheet hyvällä prosentilla. Tämä mahdollistaa sen, että voidaan olla varmoja siitä, että se osaa itsenäisesti ja varmasti tunnistaa virheet.

## 4.2 Menetelmien vertailu

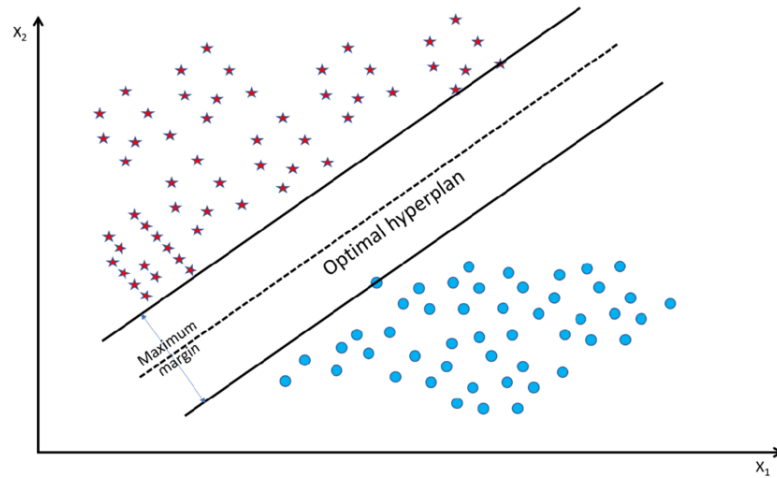
Singaporelaisen tutkimusryhmän kehittämä menetelmä, "detect-remove-replace" (DDR) eli suomeksi "tunnista-poista-korvaa", on esimerkki erittäin teollisella tasolla tapahtuvasta 3D-tulostimien virhetilanteiden tunnistamisesta ja korjaamisesta. Heidän päämääränään on ollut luoda erittäin automatisoitu tunnistusjärjestelmä yleisellä tasolla, mahdollistaen kokonaan automatisoidun tulostuslevyn tyhjennyksen ja uuden tulosteen aloituksen. Koneäköä on käytetty yhdessä syväoppimisen kanssa, käyttäen YOLO-pohjaista algoritmia. YOLO on nopea ja tarkka tunnistamaan suurempia elementtejä kuvasta. Kuvassa 4.1 näkee YOLO-algoritmin toimintaa ja tunnistuselementtejä<sup>1</sup>. Heidän tavoitteensa on tunnistaa vain isot tulosteen keskeyttävät virheet kuten spagetti tai tulosteen irtoaminen, joten ei ole tarpeen käyttää enemmän resursseja käyttävää algoritmia. Heillä kuitenkin osana prosessia on erilinen robottikäsi, joka pystyy tyhjentämään ja puhdistamaan tulostelevyn valmiiksi seuraavaa tulostetta varten tai vaihtamaan koko levyn uuteen. Koko kokonaisuuden toimivuuden kannalta on siis tärkeää, että kaikki komponentit on saatu kommunikoimaan keskenään niin, että koneälyn tunnistuessa virheen suoritetaan levyn puhdistus ja aloitetaan uuden tulosteen tulostus. Heidän saamansa tulokset ovat olleet lupaavia. He raportoivat 85.11 % varmuuden virhetilanteen tunnistuksessa ja 20/21 onnistumisen tulosteen poistamisessa sekä 100 % varmuuden tulostelevyn vaihdossa.

---

<sup>1</sup>Joseph Redmon, 2016 (<https://arxiv.org/pdf/1506.02640.pdf>)



Kuva 4.1: Visuaalinen esimerkki YOLO-algoritmista. Algoritmi on jakanut kuva aluksi osiin, jonka jälkeen se on tunnistanut elementtejä kuvasta.



Kuva 4.2: Visuaalinen esimerkki tukivektorikoneen perustasta

[7]

Tutkijat Dell ja Chang ovat kehittäneet menetelmää, joka soveltuu paremmin yksittäisen henkilön käyttöön kuin edellä mainittu menetelmä. Perustana toimii kuitenkin sama idea, eli konenäköä käytetään yhdessä koulutetun koneälyn kanssa, jotta voidaan tunnistaa virhetilanne. He ovat käyttäneet menetelmässään tukivektorikoneetta. Tukivektorikoneelle ominaista on antaa perustaso, johon voidaan antaa marginaali, mihin vektoreita voidaan verrata. Kuvassa 4.2 näkee tukivektori rakenteen marginaalin toimintaperiaatetta<sup>2</sup>. Tässä tapauksessa tunnistus tapahtuu niin, että otettu kuva tulosteesta jaetaan 16 osaan 4x4 matriisissa, ja jokaista sektoria tarkastellaan virheen mahdollisuuden suhteen. Jos 4 osiota ylittää 10 % virhemarginaalin, voidaan tilanne katsoa virheeksi. Kyseessä on helposti implementoitava ominaisuus, koska kyseessä on vain virhetilanteiden seuranta, eli virheen korjaus jää käyttäjälle tehtäväksi. Tämä on esimerkki yleisemmästä lähestymistavasta 3D-tulostuksen automaattisessa tunnistamisessa sen asentamisen yksinkertaisuuden vuoksi. Tarvi- taan kamera, riittävän hyvä valaistus ja prosessointitehoa. [2] Toinen tutkijaryhmä on käyttänyt samanlaista pohjaa tunnistamiselle, jossa tulostusprosessia kuvataan,

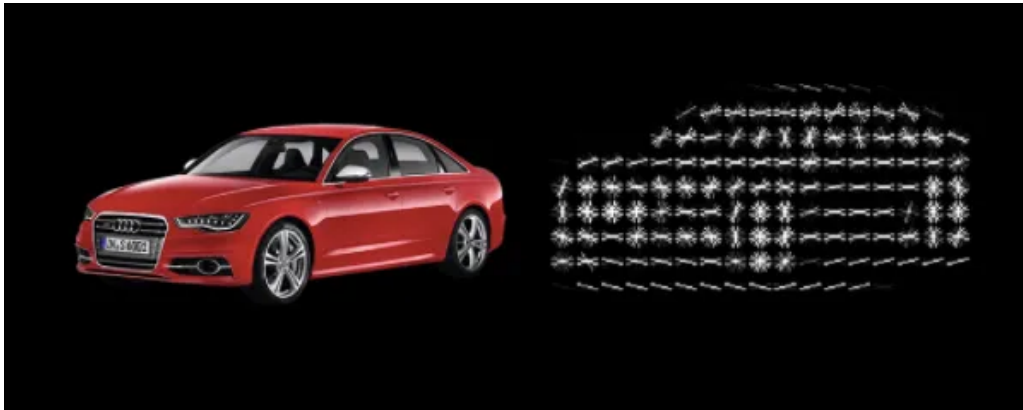
<sup>2</sup>Giuseppe Ciaburro, 2022 ([https://www.researchgate.net/publication/362604337\\_Machine\\_fault\\_detection\\_methods\\_based\\_on\\_machine\\_learning\\_algorithms\\_A\\_review](https://www.researchgate.net/publication/362604337_Machine_fault_detection_methods_based_on_machine_learning_algorithms_A_review))

ja se syötetään neuroverkon läpi. Erona kuitenkin on, että he ovat implementoineet tavan, jolla tulostusparametrejä voidaan muuttaa sen perusteella, mitä neuroverkko huomioi. Tällöin voidaan automaation määrää lisätä jo huomattavasti, koska voidaan jättää ihmiskontakti pois, kun kyseessä on pienimuotoinen ja korjattavissa oleva virhe, kuten ali- tai yliekstruusio. [18]

Petsuik ja Pearce ovat ottaneet kokonaan erilaisen lähestymistavan virhetilanteiden tunnistamiseen verrattuna edellä mainittuihin. He ovat kehittäneet tavan käyttää erillisenä renderöityä mallia ja verrata sitä tulosteeseen. Tämänkaltainen lähestymistapa luo erilaiset ongelmat yrittäessään ratkaista toisia. Menetelmä on todella aikaa vievä asennuksessa ja valmisteluissa verrattuna edellä mainittuihin menetelmiin. Tarvitaan kiinteä kamera-asennus kuten edellisissäkin menetelmissä, mutta nyt on erityisen tärkeää, miten kamera asennetaan. Kameran on oltava samassa kuvakulmassa kuin tietokoneella renderöity versio, jotta vertailu onnistuu hyvin. Valaistuksella on myös suurempi merkitys onnistumisen kannalta, koska varjot väärässä paikassa saattavat koitua ongelmalliseksi. Lisäksi varsinainen renderöinti vie paljon aikaa. Petsuikin ja Pearcen tutkimuksessa käytetyllä keskiluokan tietokoneella yhden tulostuskerroksen renderöintiin kului muutama minuutti. Riippuen siitä, kuinka monta kerrosta tulosteessa on tai kuinka monen kerroksen kohdalla tarkistus tehdään, on tämä prosessi on todella aikaa vievä. Kääntöpuolena kuitenkin on, että tämä renderöinti tarvitsee tehdä vain kerran, jonka jälkeen sitä voidaan käyttää uudelleen. Varsinainen tunnistus tapahtuu erittäin nopeasti käyttämällä HOG-pohjaista tunnistusta. Algoritmi pilkkoo kuvan osiin ja luo niistä histogrammin, josta voidaan tunnistaa kuvan piirteitä. Kuvassa 4.3 voi nähdä visualisoinnin kuvan piirteistä<sup>3</sup>. Kun sekä oikeasta tulosteesta että renderöidystä kerroksesta on luotu histogrammit, niitä on helppo vertailla toisiinsa ja tunnistaa, kuinka suuri eroavaisuus niiden välillä on. Tässä piilee menetelmän etu. Vertailu tapahtuu erittäin nopeasti, kestäen

---

<sup>3</sup>Analytics Vidhya, 2019 (<https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>)



Kuva 4.3: Visuaalinen esimerkki HOG muunnoksen piirteistä

vain sekunteja. Menetelmällä voidaan tunnistaa erittäin pienetkin erot tulosteessa, koska vertaamme suoraan malliin. Edellisissä esimerkeissä on huomattavasti vaikeampi saavuttaa samanlaista tarkkuutta tunnistamisessa. Virheelle pitää kuitenkin antaa jonkinlaiset parametrit, koska pientä eroavaisuutta on aina tiedossa sekä pitää pohtia missä menee virheen raja. Menetelmä ei ole yleisesti ottaen hyvä yhden kappaleen tulostukseen aikaa vievän renderöinnin takia, mutta varsinainen etu piileekin siis sarjatuotannossa. Edellisiin menetelmiin verraten varsinainen tunnistus on kevyt kuormaltaan, mikä mahdollistaa sarjatuotannossa pienen prosessoritehon käytön, koska sen ei tarvitse koko tulostuksen ajan pyörittää raskaampia syväoppimisalgoritmeja. Myös Paschalisin tutkijaryhmä on käyttänyt menetelmää missä fyysistä mallia verrataan digitaaliseen. Erona kuitenkin on, että heidän tapauksessaan valmista kappaletta verrataan CAD-malliin, jolloin vertailu kesken tulostuksen ei onnistu, mutta voidaan nähdä, miten eri parametrit vaikuttavat lopputulokseen. [12] [13]



## 5 Pohdintaa

On selvää, että ei ole yhtä oikeaa ratkaisua kaikkiin eri tilanteisiin. Kun pohditaan, mikä tapa on tietystä tilanteesta sopivin, on tarpeen arvioida asiaa monelta eri kannalta, jotta päästään parhaaseen vaihtoehtoon. Kuten aikaisemmin käytiin läpi, jotkin menetelmät vaativat huomattavasti enemmän alkutyötä kuin toiset, mutta toisaalta varsinainen virhetunnistus saattaa olla kevyempää.

Kysymys tuotteiden valmistusmääristä ja tulostimien määrästä on keskeinen. Yksityishenkilön yhden tulostimen käytössä ei todennäköisesti tarvita kovin tarkkaa tai monimutkaista virhetunnistusta, sillä tulostimen käyttäjä on todennäköisesti valmiimpi puuttumaan tilanteeseen ja korjaamaan sen itse. Kysymys voi olla siitä, että voisi jättää tulostimen yksin toimimaan ja saada ilmoituksen, jos tapahtuu jotain katastrofaalista. Yksityishenkilö, joka ei tulostele jatkuvasti vaan aika ajoin, tuskin haluaa suurta alkusäätöä saadakseen virhetunnistuksen käyttöön ja toimimaan.

Massatuotannossa taas virhetunnistuksen tarkkuus ja monimuotoisuus ovat oleellisia, sillä jokaista tulostinta ei haluta tarkkailla turhaan. Sen sijaan tavoitteena on saada parempaa automaatiota säästämään työtunteja ja materiaalia. Molemmat ovat yrityksen mittakaavassa säästöjä kustannuksiin, jotka aiheutuvat tulostamisesta. Alussa asennuksen monimutkaisuus ei myöskään ole niin suuri este, jos nähdään sen kattavuus pitkällä aikavälillä, jolloin suurempikin alkusijoitus on mahdollista perustella.

## 6 Yhteenveto

Tutkielman tarkoitus oli saada vastaukset seuraaviin tutkimuskysymyksiin:

1. Minkälaisia virhetilanteita kohdataan ja miten niihin reagoidaan?
2. Kuinka hyödynnämme koneoppimista niin että se luotettavasti pystyy tunnistamaan nämä virhetilanteet?

Pohjana käytettiin FDM-tulostusta, ja kävimme läpi sen yleiset ongelmat, kuten spagetin syntyminen, ali- ja yliekstruusio, kappaleen irtoaminen ja suuttimen tukkeutuminen. Ongelmille saatiin löydettyä erilaisia syitä, jotka vaihtelevat korjattavuudeltaan helposta vaikeaan. Osaan riittää tulostimen kalibrointi ja parametrien säätö oikeaan suuntaan virheen perusteella. Perusasiat, kuten tulostelevyn puhtaus ja muovin laatu, on myös pidettävä mielessä.

Koneälyn näkökulmasta hyvin koulutettu tunnistus saattaa osata määrittää, mitä pitää muuttaa, jotta mahdollinen virhe korjataan. Tämä riippuu täysin koneälyn määrittelystä ja siitä, halutaanko tämä antaa koneälylle vai ihmiselle päätettäväksi. Ihmiselle jättäminen helpottaa koneälyn koulutusta, joten siinä pitää miettiä kohdetta tarkkaan.

3D-tulostuksen saralla koneälyllä on monia erilaisia käyttökohteita, kuten mas-  
satuotanto, yksittäisen käyttäjän tarpeet ja kaikki siltä väliltä. Hyvin koulutettu koneäly on välttämätön, jotta tunnistukseen voidaan luottaa. Eri menetelmät käyttävät erilaisia lähestymistapoja päästäkseen haluttuun tulokseen. Esimerkiksi YOLO-algoritmi voi olla nopea mutta ei ehkä kovin tarkka tarkemmassa tunnistamis-

sa. Renderöityä mallia voidaan verrata HOG-pohjaisella tunnistuksella, mikä mahdollistaa pienienkin muutosten havaitsemisen mallissa. Tukivektorikone puolestaan edustaa näiden kahden väliltä olevaa ratkaisua.

Kaikissa näissä menetelmissä luotettavuus perustuu vahvasti koulutukseen, ja hyvin koulutettu algoritmi on luotettava. Myös fyysisten lisälaitteiden asennus ja implementointi ovat oleellisia jokaisen menetelmän kannalta. On tärkeää huomioida, että eri menetelmiä valittaessa on otettava huomioon käyttötarpeet ja tavoitteet saavuttaakseen optimaalisen ratkaisun

# Lähdeluettelo

- [1] M. Farhan Khan, A. Alam, M. Ateeb Siddiqui et al., "Real-time defect detection in 3D printing using machine learning", *Materials Today: Proceedings*, vol. 42, s. 521–528, 2021, Second International Conference on Recent Advances in Materials and Manufacturing 2020, ISSN: 2214-7853. DOI: <https://doi.org/10.1016/j.matpr.2020.10.482>. url: <https://www.sciencedirect.com/science/article/pii/S2214785320381037>.
- [2] U. Delli ja S. Chang, "Automated Process Monitoring in 3D Printing Using Supervised Machine Learning", *Procedia Manufacturing*, vol. 26, s. 865–870, 2018, 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA, ISSN: 2351-9789. DOI: <https://doi.org/10.1016/j.promfg.2018.07.111>. url: <https://www.sciencedirect.com/science/article/pii/S2351978918307820>.
- [3] P. Dudek, "FDM 3D printing technology in manufacturing composite elements", English, *Archives of Metallurgy and Materials*, vol. Vol. 58, iss. 4, s. 1415–1418, 2013. DOI: 10.2478/amm-2013-0186.
- [4] W. Oropallo, "Ten challenges in 3D printing", English, *Engineering with Computers*, vol. 32, s. 135–148, 2016, ISSN: 1435-5663. DOI: 10.1007/s00366-015-0407-0. url: <https://doi.org/10.1007/s00366-015-0407-0>.

- 
- [5] R. B. Kristiawan, F. Imaduddin, D. Ariawan, Ubaidillah ja Z. Arifin, *Open Engineering*, vol. 11, nro 1, s. 639–649, 2021. DOI: doi:10.1515/eng-2021-0063. url: <https://doi.org/10.1515/eng-2021-0063>.
- [6] A. C. Brown ja D. de Beer, ”Development of a stereolithography (STL) slicing and G-code generation algorithm for an entry level 3-D printer”, teoksessa *2013 Africon*, 2013, s. 1–5. DOI: 10.1109/AFRCOON.2013.6757836.
- [7] H.-Y. Li, Y. Ma, M. N. A. B. Miswadi et al., ”Detect-Remove-Replace: A Robotic Solution That Enables Unmanned Continuous 3D Printing”, *IEEE Robotics Automation Magazine*, vol. 29, nro 2, s. 33–45, 2022. DOI: 10.1109/MRA.2021.3103478.
- [8] B. Shaqour, M. Abuabiah ja S. Abdel-Fattah, ”Gaining a better understanding of the extrusion process in fused filament fabrication 3D printing: a review”, *The International Journal of Advanced Manufacturing Technology*, vol. 114, nro 5, s. 1279–1291, 2021. url: <https://doi.org/10.1007/s00170-021-06918-6>.
- [9] V. M. Bruère, A. Lion ja J. Holtmannspötter, ”Under-extrusion challenges for elastic filaments: the influence of moisture on additive manufacturing”, *Progress in Additive Manufacturing*, vol. 7, nro 3, s. 445–452, 2022. url: <https://doi.org/10.1007/s40964-022-00300-y>.
- [10] S. Martin, G.-G. Joamin, S. Janak, S. Stephan ja H. Clemens, ”Effect of the printing bed temperature on the adhesion of parts produced by fused filament fabrication”, *Plastics, Rubber and Composites*, vol. 47, nro 1, s. 17–24, 2018. DOI: 10.1080/14658011.2017.1399531. url: <https://doi.org/10.1080/14658011.2017.1399531>.
- [11] W. Lee, J. Fritsch, A. Maqsood et al., ”Adaptive 3D Printing for In Situ Adjustment of Mechanical Properties”, *Advanced Intelligent Systems*, vol. 5,

- nro 1, s. 2200–229, 2023. DOI: <https://doi.org/10.1002/aisy.202200229>.  
eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.202200229>. url: <https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202200229>.
- [12] A. Petsiuk ja J. M. Pearce, "Towards smart monitored AM: Open source in-situ layer-wise 3D printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine", *Additive Manufacturing*, vol. 52, s. 102–690, 2022, ISSN: 2214-8604. DOI: <https://doi.org/10.1016/j.addma.2022.102690>. url: <https://www.sciencedirect.com/science/article/pii/S2214860422000951>.
- [13] P. Charalampous, I. Kostavelis, T. Kontodina ja D. Tzovaras, "Learning-based error modeling in FDM 3D printing process", *Rapid Prototyping Journal*, vol. 27, nro 3, s. 507–517, 2022. DOI: [10.1108/RPJ-03-2020-0046](https://doi.org/10.1108/RPJ-03-2020-0046).
- [14] P. Jiang, D. Ergu, F. Liu, Y. Cai ja B. Ma, "A Review of Yolo Algorithm Developments", *Procedia Computer Science*, vol. 199, s. 1066–1073, 2022, The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020–2021): Developing Global Digital Economy after COVID-19, ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2022.01.135>. url: <https://www.sciencedirect.com/science/article/pii/S1877050922001363>.
- [15] D. A. Pisner ja D. M. Schnyer, "Chapter 6 - Support vector machine", teokses-  
sa *Machine Learning*, A. Mechelli ja S. Vieira, toim., Academic Press, 2020,  
s. 101–121, ISBN: 978-0-12-815739-8. DOI: <https://doi.org/10.1016/B978-0-12-815739-8.00006-7>. url: <https://www.sciencedirect.com/science/article/pii/B9780128157398000067>.

- 
- [16] W. Zhou, S. Gao, L. Zhang ja X. Lou, "Histogram of Oriented Gradients Feature Extraction From Raw Bayer Pattern Images", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, nro 5, s. 946–950, 2020. DOI: 10.1109/TCSII.2020.2980557.
- [17] M. Vaezi ja C. K. Chua, "Effects of layer thickness and binder saturation level parameters on 3D printing process", *he International Journal of Advanced Manufacturing Technology*, vol. 53, nro 1, s. 275–284, 2011. DOI: 10.1007/s00170-010-2821-1. url: <https://doi.org/10.1007/s00170-010-2821-1>.
- [18] T. S. Tamir, G. Xiong, Q. Fang et al., "Machine-learning-based monitoring and optimization of processing parameters in 3D printing", *International Journal of Computer Integrated Manufacturing*, vol. 36, nro 9, s. 1362–1378, 2023. DOI: 10.1080/0951192X.2022.2145019.