



# jamk

## Koneoppimisen ja konenäön sovellukset ja menetelmät

Samppa Alanen

Opinnäytetyö, ylempi AMK  
Toukokuu 2024  
Insinööri (ylempi AMK), robotiikka

**Alanen, Sampsa**

## **Koneoppimisen ja konenäön sovellukset ja menetelmät**

Jyväskylä: Jyväskylän ammattikorkeakoulu. Toukokuu 2024, 152 sivua

Tekniikan ala. Robotiikka. Opinnäytetyö ylempi AMK

Julkaisun kieli: suomi

Julkaisulupa avoimessa verkossa: kyllä

### **Tiivistelmä**

Automaation ja robotiikan lisääntyminen teollisuudessa on kasvussa ja varsinkin joustavaan tuotantoon pyritään panostamaan. Järjestelmien joustavuutta pystytään lisäämään konenäkösovellusten avulla, milloin tuotteen ei tarvitse olla aina samassa paikassa tai asennossa. Automaatioasteen lisääntyessä manuaalinen työ pienenee, mikä tarkoittaa ihmisen tekemän laadunvalvonnan vähenemistä. Tästä syystä myös automaattista laadunvalvontaa tarvitaan yhä enemmän tuotteiden kokoonpanoon ja lopputarkastukseen. Koneenäköjärjestelmiä on ollut markkinoilla jo pitkään ja niitä hyödynnetään laajasti. Koneoppimisen yleistymisen myötä järjestelmiin on kuitenkin tullut uusia toimintamalleja, millä pystytään tekemään sovelluksia myös kohteisiin, mihin perinteiset sääntöpohjaiset järjestelmät eivät toimi.

Tavoitteena oli selvittää koneoppimiseen, konenäköön ja niiden yhdistämiseen liittyvät perusteet ja käytännöt. Toteutustavaksi valittiin kirjallisuusselvitys menetelmien mahdollisuuksien ja haasteiden selvittämiseksi, järjestelmäselvitys menetelmien ominaisuuksien vertailemiseksi, sekä kyselytutkimus yritysten näkökulman selvittämiseksi koneoppimisen hyödyntämismahdollisuuksista. Kirjallisuuskatsauksen tavoitteena oli selvittää koneoppimisen ja konenäön perusteet ja toimintatavat, minkä perusteella pystyttiin syventämään konvoluutioneuroverkkojen rakenteeseen ja käyttömahdollisuuksiin. Järjestelmätestaukseen valittiin avoimen lähdekoodin järjestelmä, sekä kaksi kaupallista järjestelmää. Tavoitteena oli selvittää järjestelmien eroja ja käyttömahdollisuuksia.

Tuloksista voitiin päätellä, että kirjallisuuskatsauksen, järjestelmävertailun ja kyselytutkimuksen tulokset ja näkemykset koneoppimisen hyödyntämiskohteista ja haasteista ovat linjassa. Koneoppimisen hyödyntämisessä on haasteita, mitkä ovat tärkeä tiedostaa menetelmää käytettäessä, mutta mahdollisuudet ovat erittäin laajat. Avoimen lähdekoodin järjestelmiä pystyy hyödyntämään laajasti erilaisiin sovelluskohteisiin, mutta käyttö voi olla huomattavasti haastavampaa, kuin kaupallisilla järjestelmillä. Kaupallisista järjestelmistä löytyy käyttöä helpottavia graafisia käyttöliittymiä ja helppokäyttötoimintoja mallin tekemiseen. Koneoppimista hyödynnetään esimerkiksi laadunvalvonnassa, luokittelussa ja merkkien tunnistuksessa. Koneoppimista voidaan hyödyntää erittäin laajasti ja työkalujen kehitys on nopeaa. Kuitenkin perinteisilläkin konenäkötoiminnoilla on omat vahvuutensa, mihin koneoppimista ei pystytä tai kannata hyödyntää. Tällaisia kohteita ovat esimerkiksi mittaukset ja tuotteiden paikoitukset, sekä nopeaa analyysiä vaativat kohteet.

### **Avainsanat (asiasanat)**

Koneoppiminen, konenäkö, kuvankäsittely, konvoluutioneuroverkko

### **Muut tiedot (salassa pidettävät liitteet)**

**Alanen, Samppa**

### **Applications and methods of machine learning and machine vision**

Jyväskylä: JAMK University of Applied Sciences, May 2024, 152 pages

Master's Degree in Robotics. Master's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

### **Abstract**

Automation and robotics are increasing in industry and flexible production is a hot topic. The flexibility of the systems can be increased with the help of machine vision applications. As automation increases, manual work decreases, which means a decrease in human quality control. For this reason, automatic quality control is also increasingly needed for product assembly and final inspection. Machine vision systems have been on the market for a long time and are widely used. However, with the spread of machine learning new operating models have been added to the systems with which it is possible to make applications even for objects where traditional rule-based systems do not work.

The aim was to find the basics and practices related to machine learning, machine vision and their combination. The method of implementation was a literature study to find out the possibilities and challenges of the methods, a system testing to compare the properties of the methods and a survey to find out the perspective of companies on the possibilities of utilizing machine learning. The aim of the literature review was to find out the basics and practices of machine learning and machine vision, which made it possible to delve deeper into the structure and possibilities of using convolutional neural networks. An open-source system and two commercial systems were selected for system testing. The aim was to find out the differences and possibilities of using the systems.

From the results it was possible to conclude that the results and views of the literature review, system comparison and survey on the utilization targets and challenges of machine learning are in line. There are challenges in utilizing machine learning, which are important to be aware of when using the method, but the possibilities are very wide.

Open-source systems can be utilized for a wide range of applications, but the use can be much more challenging than commercial systems. Commercial systems have graphical user interfaces and easy-to-use functions for creating a model. Machine learning is widely used in quality control, classification, and character recognition. However, even traditional machine vision functions have their own strengths which machine learning cannot or should not be used for. Such sites include measurements and product positioning as well as sites requiring quick analysis.

### **Keywords/tags (subjects)**

Machine learning, machine vision, convolution, neural network

### **Miscellaneous (Confidential information)**

## Sisältö

<b>1</b>	<b>Johdanto</b> .....	<b>7</b>
1.1	Toimeksiantaja .....	8
<b>2</b>	<b>Tutkimusasetelma</b> .....	<b>9</b>
2.1	Tutkimuskysymykset .....	9
2.2	Tutkimus- ja analyysimenetelmät .....	10
<b>3</b>	<b>Koneoppiminen</b> .....	<b>11</b>
3.1	Mallin opettaminen ilman merkattua aineistoa .....	13
3.2	Mallien opettaminen esimerkkien pohjalta .....	14
3.2.1	Osittain ohjattu oppiminen .....	15
3.2.2	Itseohjattu oppiminen .....	16
3.3	Vahvistusoppiminen .....	17
3.4	Koneoppimisen algoritmit .....	17
3.4.1	Neuroverkot .....	28
3.4.2	Koneoppimismallin toiminnan arviointi .....	34
<b>4</b>	<b>Konenäkö</b> .....	<b>39</b>
4.1	Konenäköjärjestelmän rakenne .....	40
4.1.1	Kamerakennon rakenne .....	41
4.1.2	Optiikat .....	49
4.1.3	Valaistuksen suunnittelu .....	50
4.1.4	Kuvien prosessointi .....	50
4.2	Konenäköfunktiot ja niiden hyödyntäminen .....	51
4.2.1	Konenäköjärjestelmien haasteet .....	72
<b>5</b>	<b>Koneoppiminen kuvien tunnistuksessa</b> .....	<b>73</b>
5.1	Yleiset periaatteet koneoppimismallien hyödyntämiseen .....	81
5.2	Koneoppimista hyödyntävät kaupalliset konenäköjärjestelmät .....	89
5.2.1	Koneoppimista hyödyntäviä tuotteita .....	90
<b>6</b>	<b>Järjestelmien vertailu</b> .....	<b>92</b>
6.1	Avoimen lähdekoodin koneoppimistestaus .....	92
6.2	MVTec Deep Learning Tool -koneoppimistyökalun testaus .....	104
6.3	Cognex VisionSuite -ohjelmiston testaus .....	111
6.4	Järjestelmätestausten yhteenveto .....	120



<b>7</b>	<b>Kyselytutkimuksen tulokset .....</b>	<b>121</b>
<b>8</b>	<b>Johtopäätökset.....</b>	<b>124</b>
<b>9</b>	<b>Pohdinta.....</b>	<b>128</b>
9.1	Luotettavuus .....	129
9.2	Jatkokehitys.....	130
	<b>Lähteet .....</b>	<b>131</b>
	<b>Liitteet .....</b>	<b>142</b>
	Liite 1. Kyselytutkimus.....	142
 <b>Kuviot</b>		
	Kuvio 1. Koneoppimisen tyyllilajit.....	12
	Kuvio 2. Ohjaamattoman oppimisen käyttäminen hedelmien luokitteluun .....	13
	Kuvio 3. Ohjatun oppimisen käyttäminen hedelmien tunnistukseen .....	14
	Kuvio 4. Koneoppimisprojektin työmäärät .....	15
	Kuvio 5. Lineaarisen regressiomallin hyödyntäminen toimitusajan määrittämiseksi .....	19
	Kuvio 6. Epälineaarisen aineiston kuvaajat.....	20
	Kuvio 7. Sigmoid-funktion hyödyntämisestä kahden aineiston luokitteluun.....	21
	Kuvio 8. Decision tree -mallin käyttäminen eläinten luokitteluun .....	22
	Kuvio 9. Päättöpuun hyödyntäminen tietojoukkueesta.....	22
	Kuvio 10. Yli- ja alioppimisen esimerkkikuvaajat. ....	23
	Kuvio 11. Aineiston käsittely Random forest -toiminnolla viidellä eri puumallilla.....	24
	Kuvio 12. KNN mallin hyödyntäminen luokittelussa.....	25
	Kuvio 13. Tukivektorikoneen esimerkki kaksiulotteisesta aineistosta .....	26
	Kuvio 14. Tukivektorikoneen esimerkki kolmiulotteisesta aineistosta .....	26
	Kuvio 15. K-mean menetelmän esimerkkikuvat .....	27
	Kuvio 16. Neuronin toiminta.....	28
	Kuvio 17. Yksinkertaisen neuroverkon rakenne numeroiden tunnistamiseen .....	29
	Kuvio 18. Neuroverkon painokertoimet .....	30
	Kuvio 19. Aktivointifunktioiden kuvaajia .....	31
	Kuvio 20. Neuroverkon forward propagation -menetelmän toimintaperiaate.....	32
	Kuvio 21. Backpropagation-menetelmän toimintakuvaus .....	33
	Kuvio 22. Aineiston jakaminen mallin testausta ja validointia varten.....	34
	Kuvio 23. Ristiin validointi kuudella osiolla.....	35
	Kuvio 24. Precision- ja recal-toimintojen erojen esittely.....	36

Kuvio 25. Opetusaineiston jakaminen pienempiin eriin .....	39
Kuvio 26. Esimerkki älykameran käytöstä.....	41
Kuvio 27. Kamerakennon toimintaperiaate.....	41
Kuvio 28. CCD-kennon toimintaperiaate .....	42
Kuvio 29. CMOS-kennon toimintaperiaate .....	43
Kuvio 30. Pikselin mikrolinssi ja värisuodin.....	44
Kuvio 31. Väri- ja harmaasävykennon rakenne .....	45
Kuvio 32. Kolmella kennolla varustettu värikamera .....	45
Kuvio 33. Värikuvan muodostuminen pikseleistä.....	46
Kuvio 34. Harmaasävyarvot 2-, 4- ja 8-bittisellä muunnoksella .....	47
Kuvio 35. RGB-kuvan värisävyt.....	48
Kuvio 36. Väriarvojen vektorit .....	49
Kuvio 37. Polttovälin vaikutus kuva-alaan .....	49
Kuvio 38. Harmaasävyarvojen histogrammit erilaisissa kuvissa.....	52
Kuvio 39. Konvoluutioesimerkki .....	53
Kuvio 40. 5 x 5 Gaussin suodatin keskihajonnalla yksi .....	54
Kuvio 41. Sobel-toiminnolla tehty reunojen tunnistus.....	56
Kuvio 42. Roberts ja Prewit -toimintojen kernelit .....	56
Kuvio 43. Ensimmäisen ja toisen kertaluvun derivaatan hyödyntäminen muutosten havaitsemiseen .....	57
Kuvio 44. Reunojen tunnistus derivaatan avulla .....	58
Kuvio 45. 5 x 5-kernelinen LoG-toiminnon muoto ja kerneli.....	59
Kuvio 46. Laplace-suodatettu ja skaalattu kuva .....	60
Kuvio 47. Kynnystys yhdellä ja kahdella raja-arvolla. ....	61
Kuvio 48. Kynnestyksen hyödyntäminen harmaasävykuviin. ....	62
Kuvio 49. Keskiarvoistettu harmaasävykuva muunnettuna binäärikuvaksi .....	62
Kuvio 50. Asiakirjan yksinkertainen ja keskiarvoistettu kynnystys.....	63
Kuvio 51. Erosion-toiminto. ....	64
Kuvio 52. Dilation-toiminto.....	65
Kuvio 53. Openin- ja Closing-toiminnot. ....	66
Kuvio 54. Reunapisteiden tunnistussuunnat ja esimerkkikuva .....	67
Kuvio 55. Bounding box ja muodon orientaatio .....	67
Kuvio 56. Muotoja ja niiden ominaisuuksia .....	68
Kuvio 57. Mittaustoiminnon esimerkkikuva .....	69
Kuvio 58. Mittausalueen suunnan vaikutus tunnistettavaan harmaasävyarvoon .....	69

Kuvio 59. Grid-kalibrointi Cognex In-Sight Explorer ohjelmassa .....	70
Kuvio 60. Tuotteen ominaisuuksien ja mittojen selvittäminen perinteisillä konenäkötoiminnoilla .....	71
Kuvio 61. Konenäkötoimintojen valintataulukko .....	72
Kuvio 62. Perinteisen konenäköjärjestelmän haasteet .....	73
Kuvio 63. Vinossa olevan muodon tunnistus konvoluutiotoiminnolla.....	74
Kuvio 64. Esimerkki täyttö-toiminnosta 3 x 3 kernelillä .....	75
Kuvio 65. Kernelin askelmäärityksen vaikutus kuvakokoon .....	75
Kuvio 66. Konvoluution ja Pooling-toiminnon tulos esimerkikuvasta .....	76
Kuvio 67. Flatten-toiminnon esimerkik kuva .....	77
Kuvio 68. Konvoluutioneruooverkon esimerkikirakenne .....	77
Kuvio 69. Mallien tarkkuus ImageNet-1k validointiaineistolla suhteessa kokoon ja laskentaan.....	78
Kuvio 70. AlexNetin rakenne.....	79
Kuvio 71. R-CNN-mallin toimintaesimerkki.....	80
Kuvio 72. YOLO toiminnolla tunstettuja kohteita .....	81
Kuvio 73. Avoimen lähdekoodin ympäristöjä koneoppimismallien tekemiseen.....	82
Kuvio 74. VG-16 konvoluutioneurooverkon rakenne .....	84
Kuvio 75. VGG-16 konvoluutioneurooverkon parametrien ja aktivointitiarvojen muistinkäyttö .....	85
Kuvio 76. Procemex paperikoneen päänvientiköysien laadunvalvonta .....	86
Kuvio 77. Beckhoff TwinCAT3 koneoppimismallin opettaminen ja hyödyntäminen .....	87
Kuvio 78. Astrocyte-työkalun esimerkki tuotteen kärjen merkkaisesta.....	89
Kuvio 79. Perinteisen ja koneoppimistahyödyntävät konenäköjärjestelmän vertailu.....	90
Kuvio 80. Kuvien lukeminen ja skaalaus Python-ympäristössä. ....	93
Kuvio 81. Testiaineiston kuvien lukeminen Python ympäristössä.....	94
Kuvio 82. Kuvien harmaasävyarvojen muuttaminen välille 0...1.....	94
Kuvio 83. Konvoluutioneurooverkon rakenne .....	95
Kuvio 84. Python-ympäristön konvoluutioneruooverkon rakenne ja parametrien määrä.....	95
Kuvio 85. Koneoppimismallin opetusmäärittely ja hirstorian tallentaminen.....	96
Kuvio 86. Mallin opetushistorian piirtäminen .....	96
Kuvio 87. Mallin tarkkuuden kehittyminen eri opetusvaiheen edetessä. ....	97
Kuvio 88. Näytettävien testikuvien valitseminen kuva-aineistosta.....	97
Kuvio 89. Tulosteet näytettävistä testikuvista.....	98
Kuvio 90. Confusion matrix -taulukko testiaineistosta Python-ympäristössä .....	98
Kuvio 91. Confusion matrix -taulukon kuvien tulostaminen. ....	99
Kuvio 92. Heatmap-kuvan esittäminen Python-ympäristössä .....	100

Kuvio 93. Mallin kerrosten lukitseminen siirto-opetusta varten Python-ympäristössä.....	100
Kuvio 94. Esiopetetun Xception-mallin hyödyntäminen Python-ympäristössä .....	101
Kuvio 95. Python-ympäristössä tehdyn koneoppimismallin muistinkäyttö opetuksen aikana.	103
Kuvio 96. Halcon ohjelmointikielet ja liityntärajapinnat .....	104
Kuvio 97. Aineiston valinta Deep learning tool -työkalulla.....	105
Kuvio 98. Opetusaineiston valitseminen ja tarkastaminen Deep learning tool -työkalulla. ....	106
Kuvio 99. Koneoppimismallin määrittäminen Deep learning tool -työkalulla.....	106
Kuvio 100. Mallin opetushistorian tarkastelu Deep learning tool -työkalulla .....	107
Kuvio 101. Tulosten varmentaminen Deep learning tool -työkalulla.....	108
Kuvio 102. Heatmap-kuvien esittäminen Deep learning tool -työkalulla.....	108
Kuvio 103. Deep learning tool:in muistinkäyttö mallin opettamisen aikana. ....	109
Kuvio 104. Muistinkäyttöraja ResNet-50-mallin kanssa. ....	109
Kuvio 105. Deep learning tool muoton valinta ja segmentointityökalut .....	110
Kuvio 106. Aineiston valinta ViDI-työkalulla .....	112
Kuvio 107. Valitun ainesiton tarkastelu ViDI-työkalulla.....	112
Kuvio 108. Kuvien luokkamäärittely ViDI-työkalulla.....	113
Kuvio 109. Opetusparametrien määrittäminen ViDI-työkalulla. ....	114
Kuvio 110. Tulosten tarkastelu ViDi-työkalulla.....	114
Kuvio 111. Koneoppimismallin tulosten tarkastelu ViDI-työkalulla .....	115
Kuvio 112. ViDi-työkalun supervised-toiminto. ....	116
Kuvio 113. Testiainesiton analysointi ViDi-työkalulla .....	116
Kuvio 114. Mallin opetusaineiston kasvattaminen ViDi-työkalulla. ....	117
Kuvio 115. ViDi EasyBuilder työkalun pattern- ja syötekuva valinta. ....	118
Kuvio 116. Opetusaineiston valinta EasyBuilder-näkymässä .....	118
Kuvio 117. Opetusaineiston tarkastelu EasyBuilder-näkymässä.....	119

## Taulukot

Taulukko 1. Lineaarisen ja logistisen regressiomallin erot. ....	21
Taulukko 2. Aktivointifunktioiden esittely .....	31
Taulukko 3. Koneoppimisessa hyödynnettävien algoritmien listaus.....	33
Taulukko 4. Aineiston hyödyntäminen mallin sovittamisessa.....	35

Taulukko 5. Confusin matrix -taulukkoesimerkki.....	37
Taulukko 7. Bechoff TwinCAT3 ML-kirjaston käytettävissä olevista algoritmeista .....	88
Taulukko 8. Aineiston käsittelyyn soveltuvia ohjelmistoja ohajattua oppimista varten.....	102
Taulukko 9. Python ympäristön testaustulokset .....	103
Taulukko 10. MVTec Deep Larning Tool testaustulokset .....	111
Taulukko 11. Cognex VisionSuite testaustulokset .....	120

# 1 Johdanto

Tekoälystä keskustelu ja uutisointi on jo osa jokapäiväistä elämäämme. Tekoälyltä odotetaan paljon ja sen toivotaan mullistavan maailmaa monella osa-alueella. Tekoälyn toivotaan helpottavan ja korvaavan rutiinityötehtäviä, tunnistavan puhetta ja tekstiä, helpottavan ja parantavan lääketieteellisiä diagnooseja ja tukevan tieteellistä tutkimusta. (Goodfellow, Bengio & Courville 2016, 1.) Tekoälyn mahdollisten hyödyntämiskohteiden takia siihen panostetaan runsaasti tutkimuksessa, mutta myös teollisuudessa. Tekoälyn teknologiajohtajat, joihin voidaan lukea mukaan Google, Facebook ja Amazon, kasvattavat asemaansa ja ovat jopa saaneet jo huolestuttavan aseman tekoälyn kehityksen parissa. Tekoälyyn liittyy myös pelkoa työpakkojen vähentymisestä ja siitä, että tekoäly tulee lähivuosikymmeninä jopa ihmistä älykkäämmäksi supertekoälyksi. Tekoälyyn liittyvässä keskustelussa on kuitenkin paljon liiallista innostusta tai mainostusta ja monesti tekoälystä puhuttaessa taustalla saattaakin olla vain normaalia digitaaliseen teknologiaan liittyvää kehittymistä. (Pietikäinen & Silvén 2023, 6.)

Tekoälyn mahdollisuuksista käydään paljon keskustelua, mutta käytännön teollisia sovelluksia on toistaiseksi vähän. Eniten viime aikoina esille noussut tekoälyn, tai paremminkin koneoppimisen, hyödyntämiskohta teollisuudessa on erilaiset konenäköjärjestelmiin liittyvät ratkaisut. Tehokkailla tietokoneilla ja työkaluilla pystytään hyödyntämään syväoppivia neuroverkkoja, joilla saadaan aikaan helposti näyttäviä tuloksia. Suorituskyky ei kuitenkaan ole vielä sillä tasolla, että kaikkia järjestelmiä pystyttäisiin hyödyntämään todellisiin sovelluksiin, tai laskentaresurssin tarve on huomattavasti suurempi, mitä perinteisillä menetelmillä. Tämän takia varsinkin konenäköjärjestelmissä on syytä tutustua ja selvittää vaihtoehtoiset menetelmät ja ratkaisut, sekä tunnistaa koneoppimisen hyödyntämisen haasteet ja rajoitteet. Monesti perinteinen järjestelmä saattaa olla sovelluksen kannalta riittävän hyvä ja kustannustehokas. (Latokartano, Karvonen, Skriko, Holamo, Christophe, Ahonen, Haapakoski, Partanen, Lempiäinen, Kampiainen, Paasio, Liljamo & Närhi 2023, 68; Pietikäinen & Silvén 2023, 7.)

Koneoppimista pystytään hyödyntämään konenäköjärjestelmissä muun muassa laaduntarkkailussa haastavissa kohteissa, missä perinteiset ehtopohjaiset konenäköfunktiot eivät toimi. Lisäksi koneoppimista hyödyntävillä järjestelmillä pystytään lyhentämään perinteisiin konenäkötoimintoihin

pohjautuvien sovellusten kehitys- ja testausaika. Koneoppimisen myötä myös erilaiset helppokäyttötoiminnot ovat mahdollistaneet sovellusten nopeamman kehittämisen. (Latokartano ym. 2023, 68.)

Suomen kilpailukykyä pyritään nostamaan automatisoinnin ja robotiikan avulla. Uutisoinnissa on paljon esillä suomen robotti-investoinnit ja miten automaatio ja robotiikka tulevat muuttamaan tuotantoa ja varsinkin joustava tuotanto on paljon esillä (Welling 2023). Suomessa on investoitu viime vuosina paljon yhtyeistyörobotiikkaan, mikä soveltuu rakenteen, ohjelmoinnin, liikuteltavuuden ja turvatoimien ansiosta hyvin vaihtelevaan ympäristöön (Tervola 2023). Automaatioasteen nostaminen ja robottien lisääminen tarkoittaa yleensä myös kameratekniikan lisäämistä. Kameralla tuotannosta pystytään tekemään joustavampaa, kun tuotteiden ei tarvitse olla aina aseteltuna samaan kohtaan. Laitteiden lisääntyessä ja ihmisen tekemien työvaiheiden vähentyessä myös ihmisen tekemä visuaalisen tarkastus tuotteelle vähenee. Tämän takia myös kameralla tehtävät laaduntarkastukset linjastolla tai valmiiseen tuotteeseen tulee lisääntymään.

## **1.1 Toimeksiantaja**

Työn toimeksiantajana toimii Jyväskylän ammattikorkeakoulu. Työn aihepiiri ja selvitettävät asiat nousivat esille vuosina 2021–2023 toteutetusta coADDVA – ADDing Value by Computing in Manufacturing -hankkeesta, minkä tarkoituksena oli kehittää valmistavaan tuotantoon ja kunnossapittoon liittyvää kyvykkyyttä laskennan avulla. Hankkeen yhtenä toimenpidepakettina oli koneoppimiseen pohjautuvat konenäköjärjestelmät. Koneoppimiseen liittyviä konenäköjärjestelmiä ei ollut Jamkilla käytössä ennen hankkeen aloittamista, minkä takia aiheesta ei ollut juurikaan ennakkotietoa. Hankkeen edetessä koneoppimiseen ja erilaisiin järjestelmiin liittyviä haasteita ja kysymyksiä nousi esille. Lisäksi hankkeeseen kuului investointihanke, minkä ansiosta Jamkille saatiin hankittua uusia koneoppimista hyödyntäviä konenäköjärjestelmiä ja ohjelmistoja. Jamkin henkilöstön tiedon ja osaamisen kasvattamisen lisäksi konenäköjärjestelmien ja varsinkin koneoppimista hyödyntävien järjestelmien tuntemus paikallisissa yrityksissä on varsin pientä. Jyväskylän ammattikorkeakoulu tarjoaa yrityksille konenäköjärjestelmien palveluita, joten aihepiirin tutkimisella saadaan mahdollisesti vietyä tietämystä myös paikallisille yrityksille.

## 2 Tutkimusasetelma

Konenäköjärjestelmät kehittyvät nopealla vauhdilla. Tähän kuvioon on viimevuosina tullut lisäksi koneoppiminen, missä kehitysvauhti on vielä nopeampaa. Koneoppimisen hyödyntämiseen on paljon ympäristöjä ja kirjastoja, sivustot ovat täynnä erilaisia ohjeistuksia ja esimerkkejä, mutta varsinaisia käytännön, tai ainakaan teolliseen tarkoitukseen tehtyjä sovelluksia löytyy vähemmän. Aihepiiriin voi olla haastavaa tutustua ja selvittää, mikä tieto on oleellista ja millaisia toimintoja on tarjolla. Tutkimustyön aiheena oli selvittää koneoppimiseen ja konenäköön liittyvät perusteet, sekä koneoppimisen hyödyntämismahdollisuudet konenäköjärjestelmissä. Kirjallisuuskatsauksen ja järjestelmätestausten avulla pyrittiin samaan selville järjestelmien eroja ja käyttömahdollisuuksia, sekä selvittää valintoihin vaikuttavia asioita. Selvitysten perusteella pyritään kasvattamaan toimeksiantajan, sekä yritysten tietoisuutta konenäköön ja koneoppimiseen liittyviin haasteisiin ja mahdollisuuksiin.

### 2.1 Tutkimuskysymykset

Kokonaisuutena koneoppiminen ja konenäkö ovat erittäin laajoja aihealueita. Materiaalia on tarjolla erittäin paljon, eikä kaikkea pysty sovittamaan yhteen selvitystyöhön. Aihe rajattiin koneoppimiseen ja konenäköön liittyviin perusteisiin ja käytännön ratkaisuihin ja tutkimuskysymykset laadittiin liittymään mahdollisimman läheisesti yritysten ja konenäköjärjestelmistä kiinnostuneiden tahojen tarpeisiin.

Selvitystyön tutkimuskysymyksiksi valittiin:

1. Millaisia mahdollisuuksia ja haasteita koneoppimista hyödyntävissä konenäköjärjestelmissä on teollisuusyrityksille?
2. Miten yritys pystyy hyödyntämään koneoppimista tuotteiden laadunvalvonnassa?
  - Millaisia helppokäyttötoimintoja järjestelmistä löytyy?
  - Miten kaupalliset järjestelmät eroavat avoimen lähdekoodin järjestelmistä?
3. Millaisissa sovelluskohteissa erilaisia koneoppimispohjaisia menetelmiä kannattaa hyödyntää ja milloin kannattaa suosia perinteisiä konenäkömenetelmiä.



## 2.2 Tutkimus- ja analyysimenetelmät

Työn tutkimusmenetelmäksi valittiin työn luonteen perusteella kehittävä tutkimus. Koneäköä ja koneoppimista on tutkittu erittäin paljon, joten painopiste selvitystyössä oli tarvittavan tiedon kerääminen yhteen ja löytää menetelmät ja ratkaisut tutkimuskysymyksiin. Kuten Toikko ja Rantanen (2009, 21–22.) toteavat, tutkimusmenetelmien ja kehittämistoiminnan erona on luotettavuuden ja tulosten arviointi. Tutkimusmenetelmissä luotettavuus arvioidaan tieteellisten periaatteiden mukaisesti, mutta kehittämistoiminnassa pyritään käytännönläheisempään ratkaisuun ja tulosten arviointiin. Tutkimus- ja kehittämistoiminnan yhdistelmästä voidaan käyttää termejä tutkimuksellinen kehittäminen tai kehittävä tutkimus. Termeillä kuvataan siis tutkimuksen ja kehittämistoiminnan yhteyttä. Tutkimuksellisessa kehittämistoiminnassa on yleistä, että käytännön ongelmat ja kysymykset ohjaavat tiedontuotantoa ja kehittävässä tutkimuksessa toiminta kulkee tutkimuksellisesta kysymyksenasettelusta ja metodologisesta tarkastelusta käytännönläheisempää kehittämistoimintaa. (Toikko & Rantanen 2009, 18–22.)

Kirjallisuudessa kehittämistyön menetelmät voidaan jakaa määrällisiin ja laadullisiin menetelmiin. Määrällisillä menetelmillä pyritään selvittämään esimerkiksi pitääkö jokin teoria paikkansa. Teoriasta tehdään hypoteeseja, joita selvitetään esimerkiksi kyselytutkimuksella. Kerätyn aineiston analysointi tapahtuu tilastollisilla menetelmillä. Laadullisessa menetelmässä tavoitteena on selvittää tai ymmärtää paremmin jotakin tutkittavana olevaa asiaa, mitä ei tunneta vielä hyvin. Kehitystyössä voi olla kuitenkin hyödyllistä hyödyntää useampia menetelmiä, joilla saadaan erilaisia näkökulmia ja ideoita. Tällaisessa tapauksessa määritelmä määrällisestä tai laadullisesta menetelmästä ei ole enää relevantti. Työssä voidaan käyttää esimerkiksi kyselyitä, haastatteluja, havainnoiteja ja benchmarkingia. (Ojasalo, Moilanen & Ritalahti 2015, 40, 104–105.)

Selvitystyön kehitysmenetelminä sovellettiin sekä määrällistä että laadullista menetelmää. Työn tietoperustan aineistonkeruu ja kahden aihepiirin yhdistäminen toteutettiin dokumenttianalyysinä. Käytännön testauksiin sovellettiin Benchmarking-menetelmää, millä tarkoitetaan esimerkiksi tuotteen, palvelun, ominaisuuden tai toiminnan vertailua parhaimman ratkaisun tai vaihtoedon löytämiseksi (Ojasalo, Moilanen & Ritalahti 2015, 186).

Tutkimustyön tulosten arvioimiseksi päätettiin tehdä myös kyselytutkimus yritysten näkökulmista koneoppimisen mahdollisuuksista ja haasteista konenäkösovelluksissa. Kyselyn tutkimusasetelmaksi valittiin poikkileikkausaineistolla tehty tutkimus. Poikkileikkaustutkimus tarkoittaa, että kysely suoritetaan yhtenä ajankohtana ja vastaajia on useita. Toinen menetelmä olisi seurantatutkimus, missä kysely suoritetaan vähintään kahteen kertaan vastaajajoukon pysyessä samana. Poikkileikkaustutkimusta voidaan hyödyntää, kun kysymykset liittyvät asioiden tai ilmiöiden esiintyvyyteen. (Valli 2018.)

Tutkimustyön eettisinä periaatteina on käytetty Jamkin eettistä ohjeistusta, mikä pohjautuu Tutkimuseettisen neuvottelukunnan HTK-ohjeistukseen. Hyvän tieteellisen käytännön periaatteita ovat luotettavuus, rehellisyys, arvostus ja vastuunkanto. Tieteellisen työn tekemisessä on tärkeää, että työ suunnitellaan, toteutetaan ja dokumentoidaan huolellisesti. Toimintaan liittyvistä mahdollisista luvista ja suostumuksista on myös huolehdittava, ennen tutkimusaineiston keruun aloittamista. (Hyvä tieteellinen käytäntö ja sen loukkausepäilyjen käsitteleminen Suomessa 2023, 11–13.) Kyselytutkimukseen osallistuneille ilmoitettiin selkeästi mihin vastauksen tuloksia ja selvitystyötä tullaan hyödyntämään. Vastaukset myös kerättiin täysin anonymisti, eli vastaajista ei kerätty henkilö- tai yhteystietoja, millä vastaajien yksityisyys tai yritys saataisiin tietoon. Myös tämä ilmoitettiin vastaajille selkeästi.

Aineiston hankintaan pyrittiin käyttämään alkuperäisistä lähteistä löytyvää tietoa. Tämän lisäksi lähteiden alkuperää ja laatua pyrittiin arvioimaan kriittisesti ja aineistolle pyrittiin löytämään myös useampia lähteitä laadun varmistamiseksi. Aineistoksi valittiin niin painettuja kirjoja, artikkeleita, järjestelmätoimittajien verkkosivuja sekä uudempaa aineistoa tutkimusten ja blogikirjoitusten muodossa.

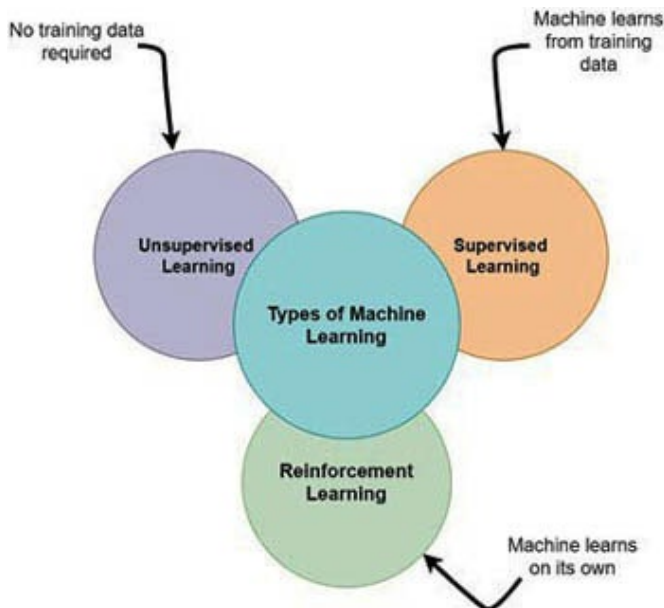
### **3 Koneoppiminen**

Koneoppimisessa tapahtui suuri läpimurto vuonna 2012, joka perustui internetistä saatavaan datamäärään merkittävään lisääntymiseen, tietokoneiden laskentatehon huimaan kehittymiseen, sekä jo 1980-luvulla tehtyihin menetelmäkeksintöihin. Merkittäviä menetelmiä ovat olleet muun muassa vastavirta-algoritmia käyttävät monikerrosneuroverkko ja CNN, eli konvoluutioneuroverkko, sekä niiden kehittyneemmät versiot. Mallien ja laskentatehon avulla osoitettiin, että käyttämällä monikerroksista oppivaa neuroverkkoa ja suurta määrää opetusaineistoa mallin kouluttamiseen

päästään huomattavasti aikaisempia menetelmiä parempaan lopputulokseen. (Pietikäinen & Silvén 2023, 8.)

Teollisuudessa tekoäly ja koneoppiminen tarkoittaa laitteille ja roboteille samaa, mitä oppiminen on luonnostaan ihmiselle. Laitteet oppivat tekemään päätöksiä ja ennusteita esimerkkien kautta. Monikerroksisten neuroverkkojen ja suurten aineistomäärien lisäksi laitteiden laskentakyky ja esiopetetut neuroverkkomallit ovat mahdollistaneet laskennan suoraan esimerkiksi kameroiden prosessoreilla. Tämä mahdollistaa koneoppimisen hyödyntämisen myös pienemmillä kuva-aineistoilla. (Cognex 2022, 2–3.)

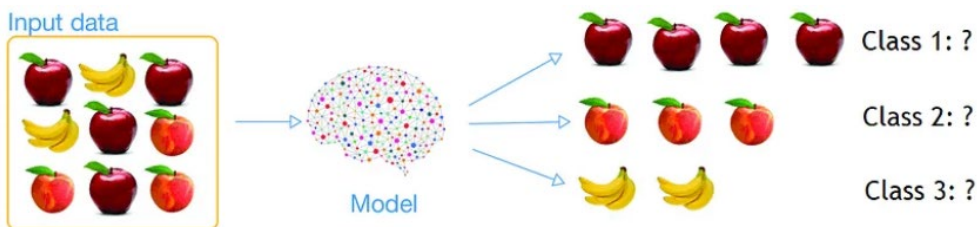
Koneoppiminen voidaan jakaa yleisesti kolmeen eri luokkaan. Luokat ovat Supervised learning, Unsupervised learning ja Reinforcement learning (ks. kuvio 1). Jokaisella menetelmällä on omat vahvuudet ja heikkoudet minkä lisäksi menetelmän valinta riippuu käytössä olevasta aineistosta ja laskentaraudasta. Mallia valitessa onkin tärkeää, että menetelmien ominaisuudet ovat selvillä, parhaan tuloksen saavuttamiseksi. (Doshi, Hiran, Jain & Lakhwani 2021, kappale 1.)



Kuvio 1. Koneoppimisen tyyliäjit (Doshi ja muut 2021, kappale 1.)

### 3.1 Mallin opettaminen ilman merkattua aineistoa

Unsupervised learning, eli ohjaamaton oppiminen tarkoittaa käytännössä sitä, että malli oppii tunnistamaan luokittelemattomasta opetusaineistosta poikkeamia tai samankaltaisuuksia. Ohjaamattomassa oppimisessä aineistoa ei esikäsitellä erillisiin luokkiin, tai siihen ei tehdä merkintöjä, millä ohjattaisiin mallin oppimisen kohteita. Ohjaamattomassa oppimisessä aineistosta pystytään löytämään yhtäläisyyksiä, joiden olemassaolo ei ole selvillä tai vaatisi erittäin paljon aineiston manuaalista tutkimista (ks. kuvio 2). Tämän takia ohjaamatonta oppimista hyödynnetään muun muassa data-analytiikassa, konenäössä tuotteiden lajittelussa, lääketieteellisessä kuvantamisessa, asiakas-segmentoinnissa ja poikkeamien havaitsemisessa. Ohjaamattomassa oppimisessä malli pystyy käymään läpi suuria määriä aineistoa ja löytää sieltä epätyypillisiä poikkeamia, joiden perusteella pystytään tekemään tunnistuksia, liittyivät ne sitten laitteiden vikaantumiseen, inhimillisiin virheisiin tai vaikka tietoturvasa havaittuun poikkeamaan. Varsinkin kyberturvallisuuden puolella voi olla haastavaa havaita reaaliaikaisesti poikkeamia tai hyökkäyksiä yrityksen tietoverkkoon. Tällaisessa tapauksessa koulutettua turvallisuusohjelmistoa voidaan hyödyntää muutosten reaaliaikaiseen havaitsemiseen. (What is unsupervised learning? N.d.; Gillis 2023.)



Kuvio 2. Ohjaamattoman oppimisen käyttäminen hedelmien luokitteluun (Bouchard 2020.)

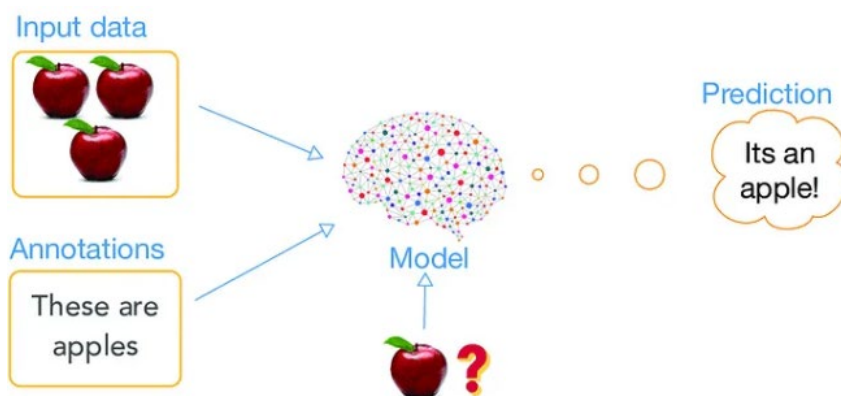
Ohjaamatonta oppimista pystytään hyödyntämään monimutkaisiin tehtäviin varsinkin silloin, kun alkuperäinen aineisto on monimutkaista ja vähemmän jäsenneiltyä. Aineiston monimutkaisuuden lisäksi aineiston koko voi olla niin iso, että tunnistusten määrittely jokaiselle datapisteelle olisi liian työlästä. Tällaisissa tapauksissa ohjaamatonta oppimista on huomattavasti nopeampi hyödyntää, kuin ohjattua oppimista. (Doshi ja muut 2021, kappale 1.)

Ohjaamattoman oppimisen heikkouksia ovat tulosten arvaamattomuus, pidemmän mallien opetusajat ja tunnistusten puutteellisuus. Opetuksen tuloksista voi olla vaikea tehdä päätelmiä, mitä

malli on oppinut ja minkä perusteella tunnistus on tehty. Tulosten todentamiseksi ei ole merkittyä aineistoa, millä mallin toiminta erillisellä aineistolla pystyttäisiin todentamaan heti opetusvaiheessa. Pidemmät mallien opetusajat johtuvat yleisesti mallien tarvitsemasta aineistomäärästä. Ohjaamaton oppiminen vaatii suuremman aineiston luotettavan tuloksen saamiseksi. Lisäksi raakadatasta oppiminen voi olla erittäin aikaa vievää. Aineiston määrä ei aina kuitenkaan paranna tulosta. Suuressa aineistossa voi olla paljon piiloon jääviä ominaisuuksia, tai toisaalta yhtäläisyyksiä löytyy väärin perustein. (Gillis 2023; Doshi ja muut 2021, kappale 1.)

### 3.2 Mallien opettaminen esimerkkien pohjalta

Supervised learning, eli ohjattu oppiminen poikkeaa ohjaamattomasta oppimisesta aineiston käsittelyn osalta. Ohjatun oppimisen perusteena on hyvin luokiteltu ja merkattu, eli annotoitu aineisto. Opetusaineistoon tehdään merkinnät joko luokista, esimerkiksi hedelmien luokittelussa kaikki hedelmien kuvat on nimetty tiettyyn luokkaan (ks. kuvio 3). Aineisto voi olla luokiteltu nimellä tai numeerisesti. Ohjatussa oppimisessa voidaan tehdä myös tarkempia määrittelyitä aineiston suhteen. Monesti pelkästään kokonaisen kuvan määrittely tiettyyn luokkaan on liian suurpiirteinen riittävän tarkkuuden saavuttamiseksi. Jos kuvassa tai aineistossa on vain pieni poikkeama, tai halutaan tunnistaa useampi vikatilanne, kannattaa kuvasta merkata tunnistettavaksi kohdaksi vain viallinen osa. Tuotteessa saattaa esimerkiksi olla toisistaan poikkeavia merkintöjä tai muotoja kokoonpanon mukaan. Tämä tarkoittaa yleensä erittäin suurta työmäärää aineiston käsittelyn osalta. Ohjatun oppimisen hyödyntämien saattaa vaihdella myös paljon käytettävän järjestelmän osalta. Millaisia malleja pystyään opettamaan ja miten aineiston merkinnät tehdään. (Bouchard 2020.)

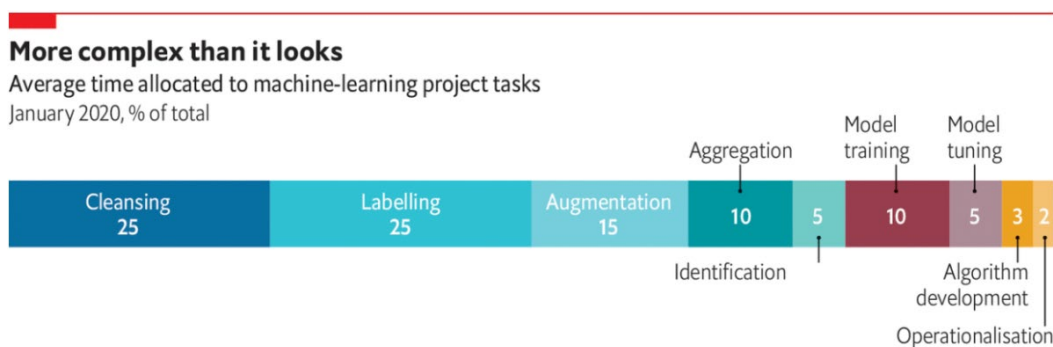


Kuvio 3. Ohjatun oppimisen käyttäminen hedelmien tunnistukseen (Bouchard 2020.)

### 3.2.1 Osittain ohjattu oppiminen

Osittain ohjattu oppiminen, eli semi-supervised learning tarkoittaa, että osa opetusaineistosta on ennalta merkattu tai opetettu järjestelmälle ja osa aineistosta on merkitsemätöntä aineistoa. Merkatulla aineistolla tarkoitetaan, että aineisto on luokiteltu valmiiksi haluttuihin kategorioihin, tai esimerkiksi kuvasta on tunnistettu, millä kohdalla kuvassa on laatupoikkeama. Merkkaamaton aineisto, jota ei ole vielä luokiteltu, pystytään hyödyntämään mallin ennustuksen säännönmukaistamiseen. (Sun, Dong, Mao & Wu 2019, kappale 6.)

Monessa tapauksessa aineiston kerääminen saattaa olla helppoa ja esimerkiksi kuva-aineistoa saadaan hyvistä tuotteista suoraan tuotantolinjastolta. Mutta kuten Sun ja muut (2019), sekä Doshi ja muut (2021) toteavat, kerätyn aineiston läpikäynti ja nimeäminen on työlästä ja aikaa vievää. Aineiston keräämien ja käsittely saattaa kestää jopa yli 70 % koko koneoppimisprojektin kestosta (ks. kuvio 4). Tämän lisäksi aineiston käsittelyllä on erittäin suuri merkitys koneoppimismallin tunnistusvarmuuteen. Usein koneoppimismallin parametrien muuttamisella ja parantamisella saadaan muutoksia aikaan tunnistustarkkuudessa vain tiettyyn pisteeseen asti. Tämän jälkeen parametrien lisääminen ja mallin syventäminen vain lisää laskenta-aikaa, mutta tarkkuuteen ei saada isoja muutoksia. Usein ratkaisuna on aineiston lisääminen ja ennen kaikkea parantaminen. (Eckstein 2022.)



Kuvio 4. Koneoppimisprojektin työmäärät (Eckstein 2022.)

Aineiston koko saattaa olla nopeasti kymmeniä tuhansia kuvia, joista pitäisi käydä läpi, mitkä ovat hyviä ja mitkä huonoja näytteitä, missä kohtaa kuvassa tunnistetaan poikkeama ja kuvat pitäisi nimetä tai lajitella omiin kansioihin. Osittain ohjatussa oppimisessä hyödynnetään aluksi merkattua

aineistoa mallin opettamiseen ja tämän jälkeen mallia lähdetään parantamaan merkkamattomalla aineistolla erilaisia oppimismenetelmiä hyödyntäen. Menetelmää kutsutaan kaksivaiheiseksi oppimiseksi, two-step learning. (Sun ym. 2019, kappale 6.)

Mallin opettamisessa voidaan hyödyntää erilaisia oppimismenetelmiä. Yksi yleisimmin hyödynnettävistä oppimismenetelmistä on puolustava oppiminen, self-training. Malli opetetaan ensin merkityllä aineistolla ja tämän jälkeen mallia hyödynnetään ennustamaan luokkia merkitsemättömästä aineistosta. Uudet tunnistukset voidaan merkata ja lisätä kuulumaan oikeaan luokkaan, jolloin merkittyjen näytteiden määrä kasvaa ja malli voidaan opettaa uudestaan isommalla aineistomäärällä. Tätä toimintoa voidaan toistaa tarvittava määrä, että mallin tunnistustarkkuus on riittävällä tasolla. (Sun ym. 2019, kappale 6.)

Aina koneoppimismallia ei kannata lähteä opettamaan alusta asti uudestaan, jos sitä halutaan tarkentaa, tai opetusaineistoa on kerätty lisää. Jos aineistolla on jo opetettu malli, voidaan malli siirto-opettaa uudestaan. Menetelmästä käytetään termiä Transfer Learning. Siirto-opettamisessa voidaan käyttää apuna myös valmiita malleja, joiden opettamiseen on käytetty erittäin laajoja opetusaineistoja. Malli ei itsessään anna parasta mahdollista tulosta tiettyyn toimenpiteeseen, koska opetusaineisto saattaa poiketa todellisesta sovelluskohteesta. Valmista mallia lähdetään opettamaan ja mukauttamaan pienellä otannalla uutta valmiiksi merkattua aineistoa. Siirto-opettamisessa mallin ylin kerros vapautetaan, mutta alempien kerrosten painokertoimet pidetään lukittuna. (Kneusel 2021, kappale 14.) Siirtymäoppimista kannattaa hyödyntää, kun saatavilla olevaa aineistoa on vähän tai tiedetään, että tunnistuksessa on samankaltaisuuksia aikaisemmin opetettuun aineistoon. (Sun ym. 2019.) Yleisesti käytössä olevissa valmiiksi opetetuissa malleissa on käytetty yleisesti muun muassa ImageNet-, CIFAR- ja MNIST-kuvasarjoja (Huilgo 2023).

### **3.2.2 Itseohjattu oppiminen**

Itseohjattu oppiminen, eli self-supervised learning, kouluttaa itseään merkitsemättömän aineiston avulla. Itseohjattu oppiminen perustuu aineiston analysointiin ja toistuvien mallien etsimiseen. Toistuvien mallinen mukaan järjestelmä pystyy tekemään johtopäätöksiä ja tunnistamaan vastaavia toimintoja uudesta aineistosta. Itseohjatusta oppimisesta on hyötyä silloin, kun mallin koulutukseen ei ole valmiina luokiteltua opetusaineistoa tai jos prosessista ei ole riittävästi ennakkotietoa opettamisen tekemiseksi. Itseohjatun oppimisen etuna onkin sen helppokäyttöisyys. Kun

opetusaineistoa ei tarvitse luokitella tai jokaista kuvaa käsitellä ja merkata virheellinen kohta erikseen asiantuntijan toimesta, nopeutuu mallin opettaminen huomattavasti. (Bergmann 2023.)

Itseohjattua oppimista hyödynnetään tällä hetkellä varsinkin kyberturvallisuuden puolella. Ihmiselle saattaa olla haastavaa havainnoida ja tunnistaa tietomurrot riittävän nopeasti ja tehokkaasti. Itseohjattua oppimista hyödyntävä järjestelmä analysoi järjestelmän toimintaa ja oppii tunnistamaan sieltä tietyt toimintamallit. Jos järjestelmään tulee poikkeamia, pystyy malli tunnistamaan ne huomattavasti ihmistä nopeammin. Itseohjattu oppiminen on yksi eniten tutkittu ja kehittyvä koneoppimisen muoto ja sen hyödyntäminen uskotaan lisääntyvän muun muassa lääketieteessä, autonomisessa ajamisessa, robotiikassa, kielen ymmärtämisessä ja kuvantunnistuksessa. (Bergmann 2023.)

### **3.3 Vahvistusoppiminen**

Vahvistusoppiminen, eli reinforcement learning, on palautteeseen tai palkkioon perustuvaa koneoppimista. Koneoppimismalli pyrkii parantamaan tulosta palkkion tai onnistumisen kautta. Vahvistusoppimisen avulla järjestelmä pyrkii tekemään oikeita päätöksiä tilanteissa saadakseen suurimman mahdollisen pitkän aikavälin palkkion, eli onnistumaan tehtävästä mahdollisimman tehokkaasti, oli se sitten esimerkiksi tarkkuutta, nopeutta tai niiden yhdistelmää vaativaa. Vahvistusoppimisessa käytetään monesti termiä agentti kuvaamaan päätöstä tekevää järjestelmää. Agentti voi olla esimerkiksi jokin kone, robotti tai vaikka tietokonepeli. Agentti oppii itsenäisesti tekemiensä valintojen perusteella, jolloin ennalta opetettua tai merkittävää aineistoa ei tarvita. Oppimisvaiheessa agentille annetaan palkkio onnistuneesta suorituksesta, kun taas virheellisestä suorituksesta annetaan rangaistus. (Brown & Zai 2020, kappale 1.)

### **3.4 Koneoppimisen algoritmit**

Koneoppimisessa hyödynnetään laajasti erilaisia menetelmiä regressio- ja luokitteluongelmien ratkaisemiseen. Menetelmän valinta tulee tehdä aineiston ja siitä tunnistettavien ominaisuuksien mukaan. Asiaan perehtyvälle saattaa herätä kysymys, miksi menetelmiä on niin laajasti käytössä, eikä menetelmissä ole keskitytty vai yhteen tai muutamaankin menetelmään? Vastaus tähän kysymykseen liittyy käsiteltävään aineistoon ja menetelmien hyviin ominaisuuksiin tietyn tyyppisen aineiston käsittelyssä. Toiset menetelmän toimivat paremmin suurten aineistojen kanssa, toiset taas



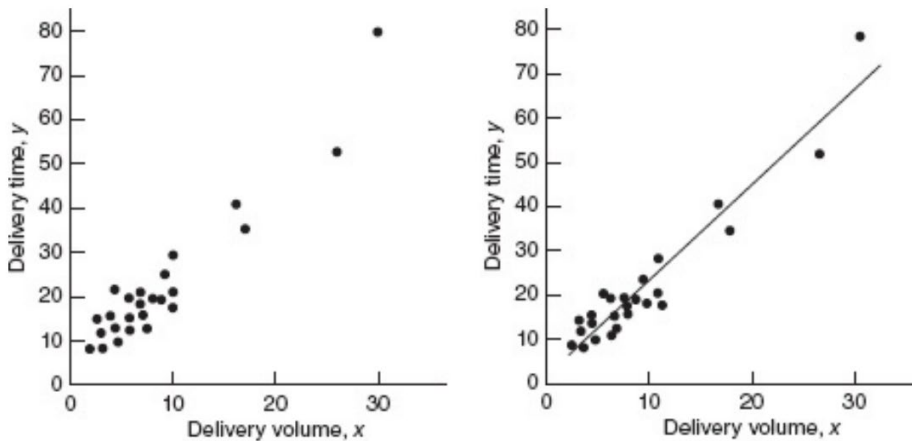
huomattavasti nopeammin, jos malli sopii käsiteltävään aineistoon hyvin. Tämän takia menetelmän valinta on oleellinen osa koneoppimista. Menetelmiä voi myös joutua vertailemaan parempien tulosten saavuttamiseksi. Menetelmät voidaan jakaa kahteen luokkaan, ennustukseen ja luokitteluun, englanninkieliset nimet Prediction ja Classification. Ennusteissa halutaan saada jokin numeerinen arvo, esimerkiksi ostosten hinta. Tällaista mallia voidaan kutsua myös regressiomalliksi. Luokittelussa mallin tehtävänä on tunnistaa eri luokkiin kuuluvien aineistojen ominaisuuksia ja tehdä päätelmä, mihin luokkaan syöte kuuluu. (Shmueli, Bruce, Deokar & Patel 2023, 8–9, 20.)

### **Regressiomallit koneoppimisessa**

Regressiomalleja hyödynnetään usein muuttujien riippuvuussuhteiden tarkasteluun. Aineiston tunnettujen pisteiden avulla halutaan määrittellä, mikä arvo todennäköisesti olisi uusien, tuntemattomien syötteiden tulos. Regressiomallit sopivat joustavuutensa ansiosta erittäin hyvin riippuvuussuhteiden tarkasteluun. Mallin hyödyntämisessä on kuitenkin omat rajoitteensa, mistä käyttäjän on hyvä olla tietoinen. (Regressioanalyysin rajoitteet 2003.)

Lineaarista regressiomallia käytetään yleisesti tunnistamaan jatkuvuutta yhden tai useamman muuttujan suhteen. Malli pyrkii sovittamaan aineiston päälle suoran, mikä vastaa mahdollisimman hyvin syötteiden tulosta. Viivan sovittaminen voidaan laskea esimerkiksi pienimmän neliösumman menetelmällä. Lineaarista regressiomallia voi hyödyntää, kun muuttujia pystytään mittaamaan jatkuvalla tasolla. Tällaisia muuttujia voivat olla esimerkiksi aika, myyntimäärät, paino tai testaustulokset. Pistekaaviota hyödyntäen voi tarkastella, onko muuttujien välillä lineaarista suhdetta. Tiedoissa ei saisi olla merkittäviä poikkeavuuksia, mitkä helposti sekoittavat mallin sovittamista. Lisäksi havaintopisteiden välillä ei pitäisi olla riippuvuutta. (What is linear regression? N.d.)

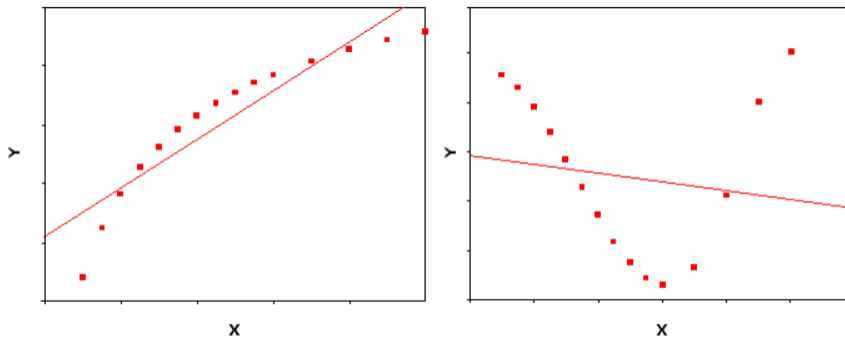
Mallin avulla voidaan esimerkiksi tarkastella juoma-automaattien huoltamiseen ja täyttämiseen menevää aikaa suhteessa toimitusmääriin. Sovittamalla tarkastelupisteet taulukkoon huomataan, että pisteiden kautta pystytään piirtämään viiva, minkä kuvastaa muuttujien välistä suhdetta (ks. kuvio 5). Sovitetun viivan perusteella saadaan laskentakaava, minkä avulla pystytään ennustamaan muilla syötteillä saatavat arvot. Eli kaavasta pystyttäisiin ennustamaan toimitusajat tietyille määrälle virvoitusjuomia. (Montgomery, Peck & Vining 2012, 17–19.)



Kuvio 5. Lineaarisen regressiomallin hyödyntäminen toimitusajan määrittämiseksi (Montgomery ja muut 2012, 18.)

Hyvin usein regressiomalli on vain likiarvo muuttujien välisestä suhteesta. Tämän takia malliin määritetään yleensä myös muuttujien poikkeama. Poikkeama voidaan ajatella tilastollisena virheenä, mikä selittyy esimerkiksi fysikaalisesta, kemiallisesta tai järjestelmän teknisestä ratkaisusta. (Montgomery ja muut 2012, 17–19.)

Aina aineistoa ei kuitenkaan pystytä määrittelemään riittävän tarkasti yhden viivan perusteella. Monesti vastemuuttuja ja ennustemuuttuja liittyvät toisiinsa epälineaarisen funktion kautta. Epälineaarisen regression avulla voidaan arvioida malleja, joissa on mielivaltaisia suhteita muuttujien välillä (ks. kuvio 6). (Montgomery ja muut 2012, 378.) Malli sovitetaan hyödyntäen esimerkiksi logaritmisia-, trigonometrisia-, eksponentti- ja potenssifunktioita. Tällä tavalla myös kaarevia muotoja pystytään tunnistamaan. Epälineaarinen regressio on lineaarista monimutkaisempi, koska funktio luodaan sarjalla approksimaatioita. Mallia sovitetaan kerta toisensa jälkeen aineiston päälle ja pisteiden välinen neliösumma pyritään saamaan mahdollisimman pieneksi. Neliösumma tarkoittaa mittaa, kuinka paljon havainnot eroavat epälinearisesta funktiosta, jota käytetään ennustamaan arvoa. (Kenton 2023.)



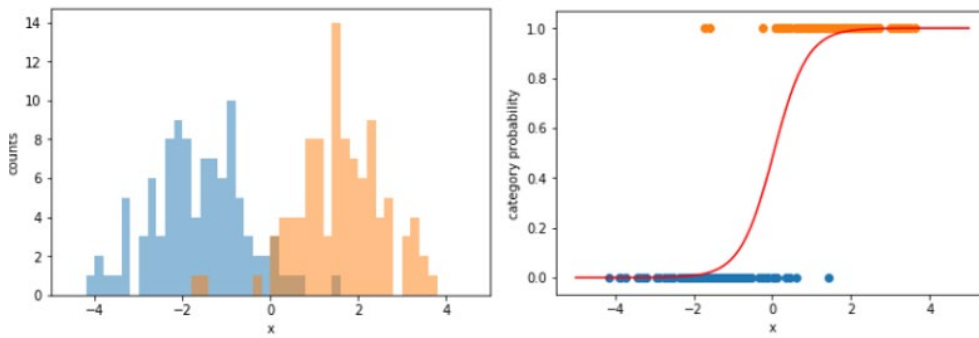
Kuvio 6. Epälineaarisen aineiston kuvaajat (Regressioanalyysin rajoitteet 2003.)

### Logistinen regressio

Logistista regressiomallia käytetään usein, kun aineisto on jaoteltavissa selkeästi kahteen eri luokkaan. Malli ei pyri ennustamaan lineaarisen regression tapaisesti jatkuvia numeerisia määreitä, vaan arvioi tapahtuman todennäköisyyden ja tuloksena on 0/1 tai esimerkiksi tosi/epätosi. Malli analysoi aineiston yhden tai useamman muuttujan välisen suhteen ja tekee luokittelun. Mallia voidaan käyttää esimerkiksi sydänkohtausten todennäköisyyden ennustamiseen, kun tarkasteltavina muuttujina on henkilön paino ja liikunnallisuus. Mallissa käytetään sigmoid-funktiota ennusteiden ja todennäköisyyksien arvioimiseen. Sigmoid-funktio viittaa s-muotoiseen käyrään, millä mikä tahansa numeerinen arvo saadaan muutettu välille 0–1. Funktiossa käytetään kynnsarvoa, millä voidaan päättää, milloin tulos muutetaan arvoksi 0 tai 1. Arvon ollessa 0,5, sitä pienemmät arvot saavat luokittelun 0 ja suuremmat saavat arvon 1. Sigmoid-funktiota kutsutaan logistisen regression aktivointifunktioksi ja se voidaan esittää yhtälöllä 1:

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Missä  $e$  on luonnollinen logaritmi ja  $x$  on syötteen arvo. Jos syöte  $x$  arvo on ääretön,  $e^{-x}$ :n arvo lähestyy nollaa ja tulos lähestyy arvoa yksi. Jos syöte on miinusmerkkinen, lähestyy  $e^{-x}$ :n arvo ääretöntä ja tulos lähestyy nollaa. Kuviossa 7 on esitetty sigmoid-funktion kuvaajalla eroteltu aineisto. (Kanade 2022; What Is Logistic Regression?)



Kuvio 7. Sigmoid-funktion hyödyntämisestä kahden aineiston luokitteluun (Bernet 2019.)

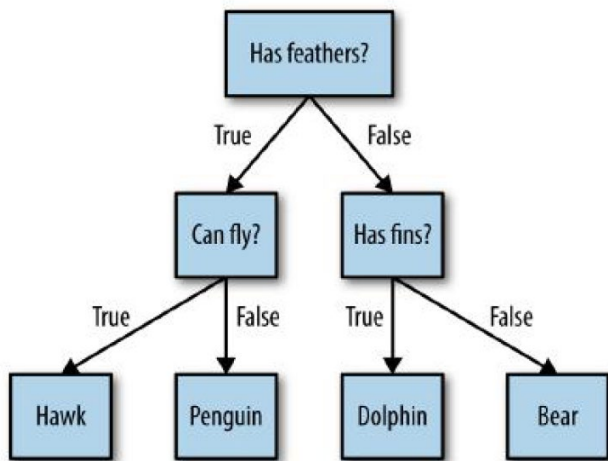
Alla olevassa taulukossa 1 on vertailtu lineaarisen ja logistisen regressiomallin eroja.

Taulukko 1. Lineaarisen ja logistisen regressiomallin erot (Doshi ja muut 2021.)

Nro	Lineaarinen regressio	Logistinen regressio
1	Jatkuva numeerinen määrä	Diskreetti määrä
2	Ennustetut arvot ovat kohdemuuttujan keskiarvo syötemuuttujien annetuilla arvoilla.	Ennustetut arvot ovat kohdemuuttujan tietyn tason todennäköisyys syötemuuttujien annetuilla arvoilla.
3	Regressio-ongelmien ratkaiseminen	Luokitteluongelmien ratkaiseminen
4	Kuvaaja on suora viiva	Kuvaaja on s-muotoinen kuvaaja
5	Käytetään numeerisia muuttujia	Käytetään kategorista muuttujaa
6	Edellyttää, että riippuvaisten ja riippumattomien muuttujien välillä on selkeä suhde	Muuttujien välillä ei tarvitse olla suhdetta

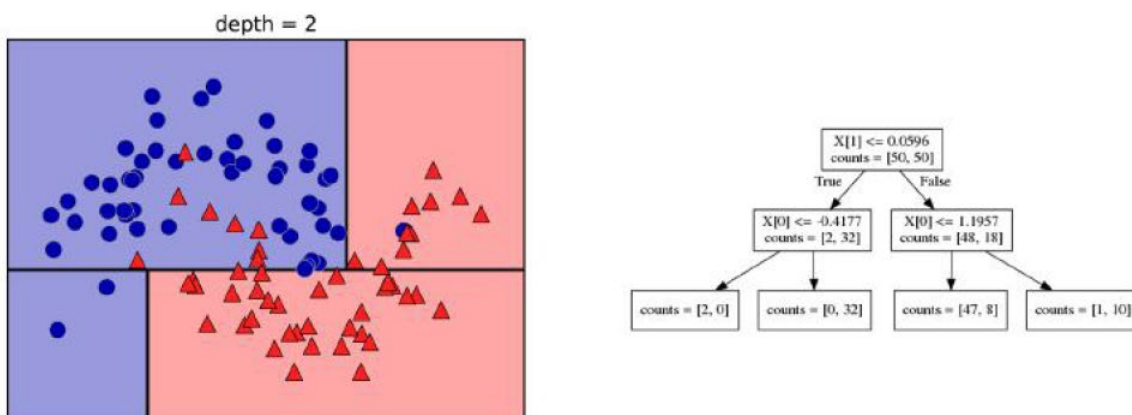
### Decision tree -mallin hyödyntäminen luokittelussa

Decision tree on hierarkkinen puurakenne, mitä voidaan hyödyntää erilaisissa luokittelutehtävissä. Puurakenteessa jokainen oksakohta tarkoittaa päätöstä tietyn ominaisuuden perusteella. Mallin rakenne on helppo visualisoida ja tulkita, minkä takia sitä käytetään yleisesti yksinkertaisten rakenteiden kanssa. Mallia pystytään hyödyntämään sekä regressio-ongelmissa että luokittelussa, missä päätös tehdään vain tiettyjen luokkien mukaisesti. Jokaisessa oksakohdassa esitetään kysymys tai ehto, minkä perusteella edetään seuraavaan oksakohtaan. Esimerkiksi eläinten luokittelu voisi mennä kuviossa 8 esitetyn rakenteen mukaisesti, missä kolmella kysymyksellä luokitellaan eläin neljään kategoriaan. (Müller & Guido 2016, 71–72.)



Kuvio 8. Decision tree -mallin käyttäminen eläinten luokitteluun (Müller & Guido 2016, 71.)

Mallia hyödynnettäessä koneoppimisessa, oksakohtien ehdot määrittyvät mallista. Eli kysymyksiä tai ehtoja ei tarvitse muodostaa manuaalisesti. Kuviossa 9 on esimerkki päätöspuun hyödyntämisessä jatkuvasta aineistosta. (Müller & Guido 2016, 71-73.)

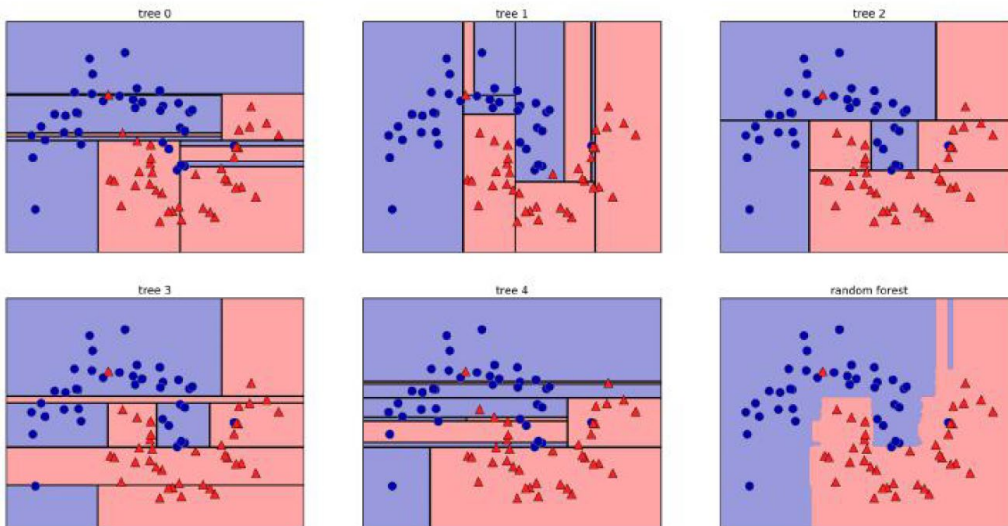


Kuvio 9. Päätöspuun hyödyntäminen tietojoukkueesta. (Müller & Guido 2016, 73.)

### Random forest -malli

Regressioniin ja luokitteluun liittyy yleensä yli- tai alioppiminen. Malli voi oppia tunnistamaan ope- tusaineiston varsin hyvin, mutta uudella aineistolla malli ei enää toimikkaan. Varsinkin yhden De- cision tree:n käyttämisessä on riski mallin ylioppimiselle, jos aineistoa pilkotaan liian pitkälle. Malli saattaa toimia vain täysin identtisellä aineistolla, mitä mallin opettamiseen on käytetty. Kuviossa



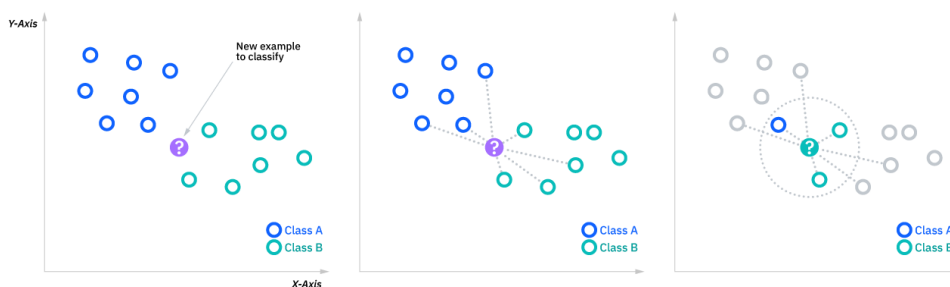


Kuvio 11. Aineiston käsittely Random forest -toiminnolla viidellä eri puumallilla (Müller & Guido 2016, 86.)

### KNN-menetelmä

KNN-menetelmän nimitys tulee sanoista K-Nearest Neighbor ja menetelmää voidaan käyttää sekä regressio- että luokittelutehtäviin. KNN-tekniikka luokittelee arvopisteet niiden läheisyyden ja yhteyden suhteen muuhun käytettävissä olevaan aineistoon. Algoritmin toiminta perustuu siihen, että samantyyppinen aineisto päätellään olevan lähellä toisiaan. Lähellä toisiaan olevat tiedot kuuluvat samaan ryhmään, kun taas aineistosta kaukana oleva piste kuuluu todennäköisesti johonkin toiseen ryhmään. KNN-malli on kohtalaisen helppo ymmärtää ja käyttää, minkä lisäksi siinä on yleensä nopea laskenta-aika. Tämän takia mallin käyttö erittäin yleistä ja yksi suosituimmista koneoppimisalgoritmeista. KNN-malli tutkii aineiston jakautumisen ja pyrkii jakamaan pisteet omiin ryhmiin mallille annettujen lähtötietojen perusteella. Lähtötietojen perusteella ryhmille myös määritetään tunnisteet. Algoritmi siis pyrkii laskemaan aineiston pisteiden välisen etäisyyden, mikä usein määritetään euklidisella metriikalla, mikä tarkoittaa kahden pisteen etäisyyttä suoralla viivalla. Algoritmi määrittelee luokat useimmiten toistuvan arvon tai esimerkiksi keskiarvon perusteella. (Ks. kuvio 12) KNN-mallia pystyy käyttämään vain ohjatun oppimisen kanssa. Eli aineiston pitää olla luokiteltu ja luokille annettu tunnisteet. Lisäksi mallia on ei-parametrinen algoritmi. Tämä tarkoittaa, että aineiston ei oleteta olevan jakautunut jonkin todennäköisyysjakauman mukaisesti. Toinen poikkeava seikka moneen muuhun mallin on, että aineistoa ei jaeta erikseen harjoit-

tus- ja testijoukkoon. Toisaalta aineiston koon kasvaessa myös käsittelyaika pitenee huomattavasti, minkä takia mallia ei käytetä niin paljon esimerkiksi luokittelutehtävissä. Yleinen käyttökohte KNN-mallille on suosittelemoottorit ja kuvantunnistukset. Mallin k-arvo kuvastaa kuinka monta naapuria jokaiselle datapisteelle määritetään, kun pistettä jaetaan eri luokkiin. K-arvon ollessa esimerkiksi kolme, tarkastetaan kolmen lähimmän datapisteen luokittelu. Pienillä k-arvoilla varianssi voi olla varsin suuri, mutta virhe pieni. Suurilla k-arvoilla virheen todennäköisyys on suurempi, mutta varianssia on vähemmän. Arvon valinta riippuu suuresti käsiteltävästä aineistosta. Jos aineistossa on paljon poikkeamia ja hajontaa, toimii suuri k-arvo todennäköisesti paremmin. Yhtenä suosituksena k-arvolle pidetään paritonta lukua. Näin ollen vältetään luokittelun sidoksista useampaan ryhmään. (Nelson 2020; What is the KNN algorithm? N.d.; What is supervised learning? N.d.)



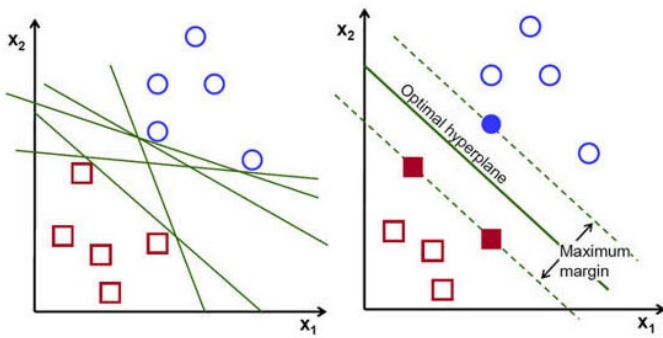
Kuvio 12. KNN mallin hyödyntäminen luokittelussa (What is the KNN algorithm? N.d.)

## Tukivektorikone

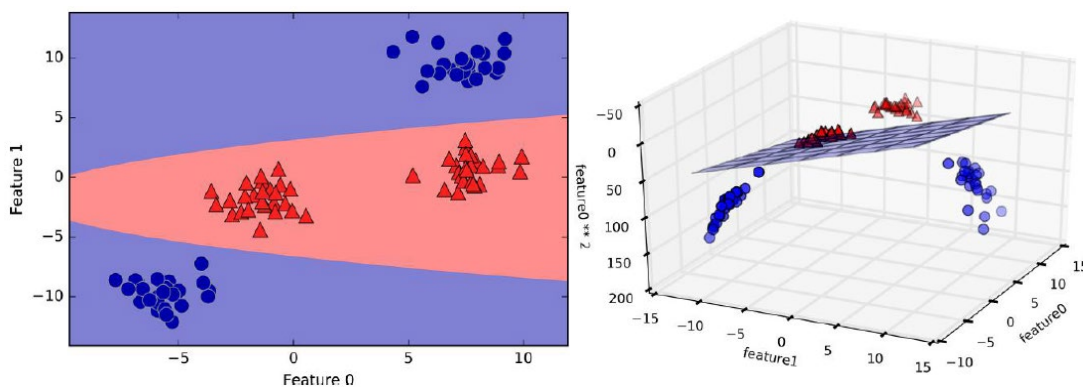
Tukivektorikone, eli support vector machine tai SVM, tarkoittaa kahden tai useamman luokittelun jakoa vektoreilla. Sitä käytetään yleisesti silloin, kun aineiston monimuotoisuuden takia sitä ei pystytä jakamaan yhden vektorin avulla. (Levman 2011, 2–3.) Myös tukivektorikonetta pystytään hyödyntämään sekä luokittelu- että regressio-ongelmien ratkaisemiseen. Pääosin sitä kuitenkin käytetään luokitteluongelmiin. Luokittelu voi tapahtua kahteen tai useampaan luokkaan.

Tukivektorikoneen käyttäminen on suosittua sen laskentatehokkuuden ansiosta. Malli sovittaa viivat näytteiden välille niin, että löydetään optimaalinen hypertaso erottamaan luokat toisistaan. Yksinkertaisessa aineistossa hypertaso voi olla viiva (ks. kuvio 13), mutta monimutkaisemmassa aineistossa hypertaso haetaan useiden ulottuvuuksien avaruudesta (ks. kuvio 14). Tämän ominaisuuden ansiosta mallia pystyy hyödyntämään erittäin monimuotoisenkin aineiston kanssa.





Kuvio 13. Tukivektorikoneen esimerkki kaksiulotteisesta aineistosta (Doshi ja muut 2021.)



Kuvio 14. Tukivektorikoneen esimerkki kolmiulotteisesta aineistosta (Müller & Guido 2016, 96–97.)

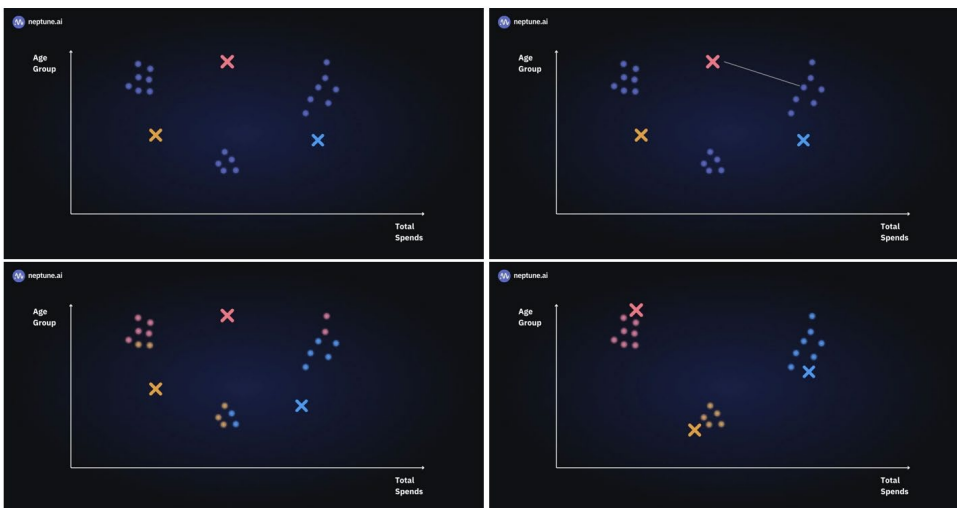
## Naivi-Bayes-algoritmi

Naivi-Bayes-algoritmi on yksinkertainen, mutta tehokas ohjatun oppimisen menetelmä. Menetelmää pystyy hyödyntämään myös erittäin suurten aineistojen kanssa. Menetelmä perustuu Bayesin lauseeseen. Menetelmä hyödyntää todennäköisyyksiä eri luokille, sekä minkä arvon mikäkin piirre saa todennäköisesti kyseisessä luokassa. Naivi-Bayes-luokittelussa ei ole käytössä vain yhtä luokittelijatyyppeä, vaan yleisesti käytetään kolmea eri versiota. Gaussian Naivi-Bayes-luokittelija hyödyntää luokkien keskiarvoa ja keskihajontaa. Multinomial Naivi-Bayes-luokittelija olettaa, että ominaisuudet ovat peräisin multinomiaalisista jakaumista ja sitä voidaan hyödyntää esimerkiksi luonnollisen kielen käsittelyssä. Bernoulli Naivi-Bayes-luokittelijaa hyödynnetään aineiston kanssa, joilla on kaksi arvo, kuten tosi/epätosi tai 1/0. Naivi-Bayes-luokittelua voidaan käyttää muun muassa roskapostin suodattamiseen, asiakirjojen luokitteluun sekä markkinoinnin tukena tekemän

analyysiä mielipiteistä ja asenteista. (Pietikäinen & Silvén 2023, 43; What are Naïve Bayes classifiers? N.d.)

## Klusterointi

Klusteroinnissa aineistosta pyritään löytämään toistuvia ominaisuuksia, minkä perusteella aineisto pystytään jakamaan tiettyihin luokkiin. Samaan ryhmään määritellyllä aineistolla on samankaltaisia ominaisuuksia, mitkä poikkeavat toisen ryhmän ominaisuuksista. Tällaisia luonnollisia luokkia kutsutaan klustereiksi. Klusterointia käytetään yhtenä valvomattoman koneoppimisen algoritmina. Klusterointi voidaan tehdä erilaisilla menetelmillä aineiston mukaan. Osiointi-klusteroinnin yhtenä vaihtoehtona on käyttää K-mean -menetelmään, missä aineisto jaetaan k:n edustamaan määrään klustereita. Aineistoa lähdetään käymään läpi valitsemalla aineistosta pisteet klustereiden arvoiksi. Näitä pisteitä kutsutaan Centroideiksi. Tämän jälkeen aineisto jakautuu valittujen pisteiden mukaisesti ja Centroid-pisteitä lähdetään siirtämään kohti valitun alueen keskiarvoa. Tämän jälkeen alueet muodostetaan uudestaan ja toimenpiteet toistetaan niin usein, että muutoksia ei enää tapahdu (ks. kuvio 15). (Doshi ja muut 2021.)



Kuvio 15. K-mean menetelmän esimerkkikuvat (Sharma 2024.)

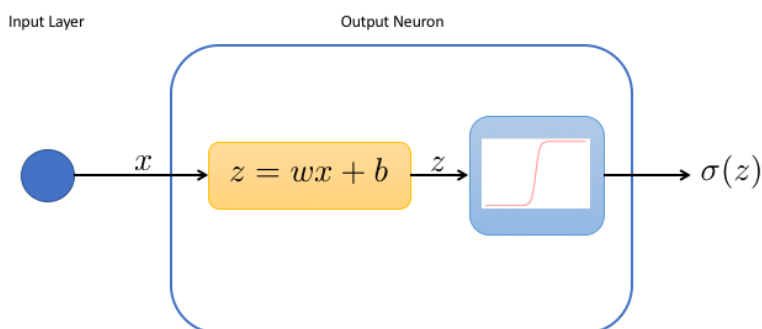
Toinen menetelmä on nimeltään kokoava klusterointi, missä jokaista datapistettä pidetään ensin omana klusterina. Tämän jälkeen klustereiden määrää aletaan vähentää niiden välisten iteratiivis-

ten liitosten kautta. Menetelmä toistaa klustereiden vähentämisen, kunnes saavutetaan menetelmään määritelty haluttu luokittelumäärä. Muita klusterointimenetelmiä ovat muun muassa päällekkäinen klusterointi, missä datapisteet voivat kuulua useampaan klusteriin samanaikaisesti, sekä todennäköisyysklusterointi, missä klusterit muodostetaan todennäköisyysjakaumamenetelmien avulla. (Doshi ja muut 2021, kappale 3.)

### 3.4.1 Neuroverkot

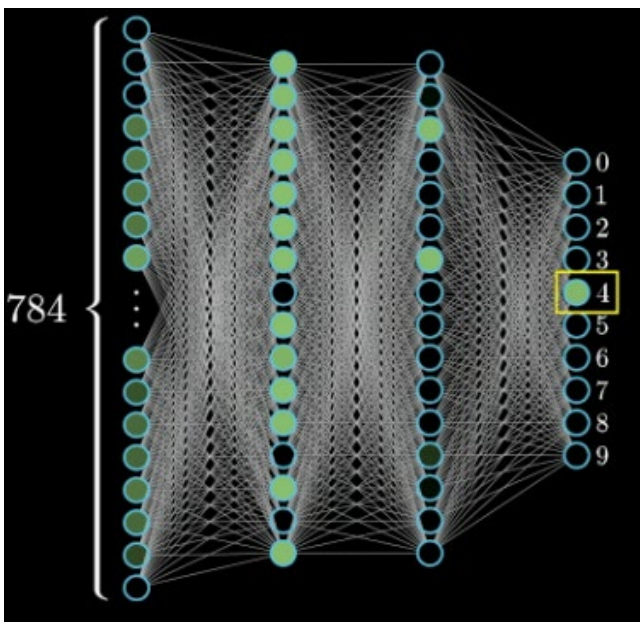
Neuroverkoista puhuttaessa nousee usein esiin neuroverkon kyky mallintaa aivojen toimintaa. Toisin sanoen neuroverkkojen rakenteeseen on pyritty mallintamaan aivojen kaltaista kerrosrakenteiden yhdistymistä solmukohtien avulla. Kerroksissa olevat solmukodat aktivoituvat tietyistä ominaisuuksista ja tieto lähetetään seuraavalle kerrokselle. Neuroverkko oppii tunnistamaan solmukohtien aktivoitumisen ja pystyy tekemään päättelyt tulosten perusteella. (What is supervised learning? N.d.)

Neuroverkko rakentuu nimensä mukaisesti useista yksittäisistä neuroneista. Jokainen neuroni sisältää lineaarisen ja epälineaarisen funktion. Epälineaarisenä, toiselta nimeltä aktivointifunktiona, käytetään yleisesti aikaisemmin esitettyä logistisessa regressiossakin hyödynnettävää Sigmoid-funktiota. Lineaarisen osan  $b$ -muuttujasta käytetään nimitystä bias. Muuttamalla bias:in arvoa, pystytään Sigmoid-funktion horisontaalista kohtaa vaihtamaan. (Ks. kuvio 16) Toisin sanoen arvolla määritellään, missä kohtaa funktion arvo on 0,5. Muuttuja  $w$  on painokerroin tulon muuttujalle  $x$ . Jos painokerrointa muutetaan suuremmaksi, muuttuu Sigmoid-funktion jyrkkyys. Tällaista yksittäistä neuronia kutsutaan nimellä Perseptroni, englanniksi Perceptron. (Bernet, C. 2019, What is a multilayer perceptron (MLP) or a feedforward neural network (FNN)? N.d.)



Kuvio 16. Neuronin toiminta. (Bernet 2019.)

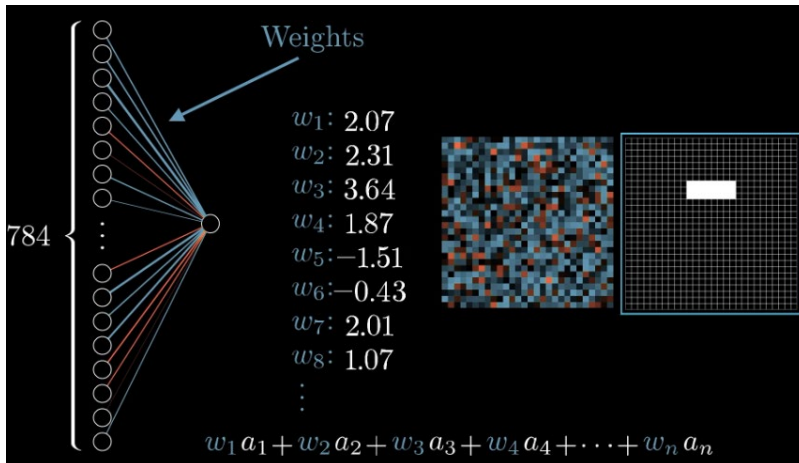
Neuroverkkojen avulla pyritään mallintamaan aineiston ominaisuuksien suhteita usean tason avulla. Neuroverkot koostuvat yleensä useista kerroksista, joiden avulla pystytään tunnistamaan erittäin monimutkaisia ja abstraktejakin piirteitä alkuperäisestä datasta. Yksinkertainen neuroverkko rakentuu yleensä kolmesta, tai useammasta kerroksesta. Ensimmäisenä kerroksena toimii syöte, mihin alkuperäinen aineisto syötetään. Ensimmäisen kerroksen jälkeen tulee piilokerrokset, joihin muodostuu piirteitä ja yhtäläisyyksiä aineiston mukaan. Piilokerrosten jälkeen on lähtökerros, mihin tuotetaan lopullinen ennuste tai luokittelu. Jokaisessa neuronissa on tulo, mihin annetaan syöte tarkasteltavasta aineistosta. Jos käsiteltävänä aineistona olisi esimerkiksi neliön mallinen 28 x 28 pikseliä sisältävä matriisikuva numerosta, neuroverkossa olisi tuloja jokaiselle pikselille eli yhteensä 784 neuronina (ks. kuvio 17). (Sanderson 2017.)



Kuvio 17. Yksinkertaisen neuroverkon rakenne numeroiden tunnistamiseen (Sanderson 2017.)

Neuroverkon piilokerroksissa edellisen neuronikerroksen tulot summataan jokaiseen piilokerroksen neuroniin ja kerrotaan epälineaarilla funktiolla. Piilokerrosten jälkeen esimerkkitapauksessa on kymmenen neuronin, yksi neuronin jokaiselle numerolle välillä 0–9. Jokaisessa neuronissa on arvo välillä 0.0–1.0, mikä kuvastaa miten todennäköisesti syötteenä annettu kuva sisältää kyseisen numeron. Esimerkin neuroverkossa on kaksi piilokerrosta, joista kummassakin on kuusitoista neuronin. Tuloneuronit aktivoivat tietyt piilokerroksen neuronit painokertoimen avulla. Jokainen pai-

nokerroin kuvastaa, miten paljon edellisen kerroksen neuroni vaikuttaa seuraavan kerroksen neuroniin. Painokertoimet voivat olla positiivisia tai negatiivisia. Yksinkertaistaen, jos edellisen kerroksen neuroni on aktiivinen, positiivinen painokerroin tarkoittaa, että myös seuraavan kerroksen neuroni tulisi olla aktiivinen. Negatiivinen painokerroin tarkoittaa, että seuraavan neuronin ei tulisi aktivoitua. Seuraavan kerroksen neuronin arvoksi lasketaan siis edellisen kerroksen neuroneiden summa (ks. kuvio 18).



Kuvio 18. Neuroverkon painokertoimet (Sanderson 2017, muokattu.)

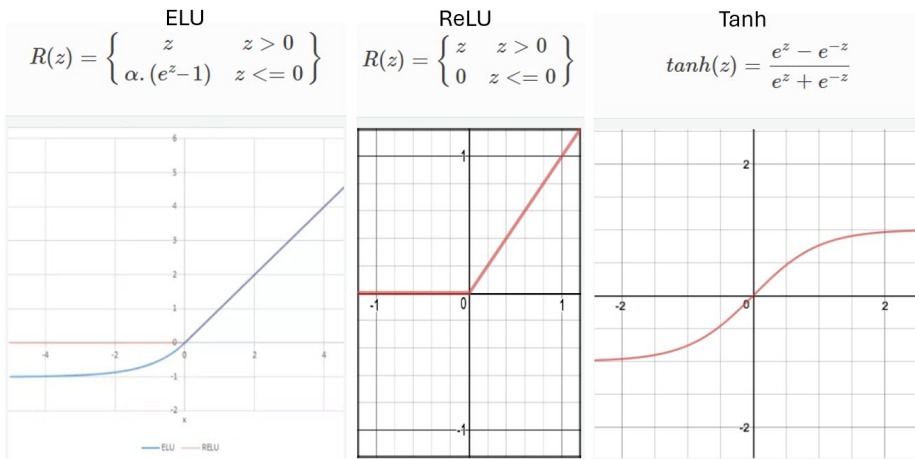
Lasketusta summasta voi näin ollen tulla mikä tahansa luku, mutta arvo halutaan usein muuntaa välille 0 ja 1. Muunnos voidaan tehdä aktivointifunktion, kuten Sigmoid-funktion avulla ja tarvittaessa hyödyntää bias-arvoa siirtämään funktiota haluttuun aktivointikohtaan. Toimenpide toistetaan jokaisen neuronin ja kerroksen osalta. Jos esimerkkikuvan tapauksessa jokaisella neuronilla olisi Sigmoid-aktivointifunktio ja bias-muuttuja, saadaan solmukohtien määrä laskettua yhtälöllä 2:

$$(786 \times 16) + (16 \times 16) + (16 \times 10) = 13002 \quad (2)$$

Tällä neuroverkolla olisi siis 13002 erilaista toimintoa, millä mallin tunnistusta pystytään muuttamaan. (Sanderson 2017.)

Aktivointifunktiona voidaan Sigmoid-funktion lisäksi käyttää myös muita toimintoja. Yleisesti käytettäviä aktivointifunktioita ovat esimerkiksi lineaarinen-, ELU-, ReLU-, Tanh- ja Softmax -funktiot.

Softmax-funktion toiminta on samankaltainen, kun Sigmoid-funktiolla. Softmaxia voidaan kuitenkin käyttää myös useiden luokkien tapauksessa, kun Sigmoid-funktiota hyödynnetään yleensä kahden vaihtoehdon luokittelussa. Lisäksi voidaan käyttää myös lineaarisia aktivointifunktioita. Kuviossa 19 on esitetty tyypillisten aktivointifunktioiden kuvaajia. (Moore n.d.)



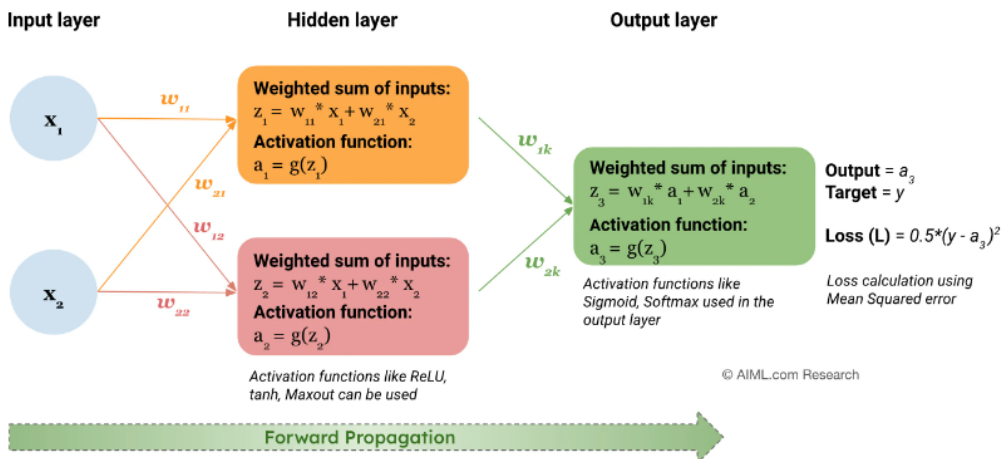
Kuvio 19. Aktivointifunktioiden kuvaajia (Activation Functions n.d.)

Taulukossa 2 on esitetty erilaisten aktivointifunktioiden ominaisuuksia. Taulukkoon on myös kirjattu aktivointitoimintojen hyviä ja huonoja puolia.

Taulukko 2. Aktivointifunktioiden esittely (Verdhan 2021, kappale 1.)

Aktivointi toiminto	Arvo	Positiivista	Haasteet
Sigmoid	[0,1]	(1)Epälineaarinen (2)Helppo käyttää (3)Jatkuva (4)Monotoninen eikä hajoita aktivointeja	(1)Lähtö ei ole keskitetty nollaan (2)Katoavan gradientin ongelma (3)Hidas opettaa
tanh	[-1,1]	(1)Sigmoidin kanssa samankaltainen (2)Gradientti vahvempi	(1)Katoavan gradientin ongelma
ReLU	[0,inf]	(1)Epälineaarinen (2)Helppo laskea ja nopea kouluttaa (3)Ratkaisee katoavan gradientin ongelman	(1)Käytetään vain piilokerroksissa (2)Voi hajoittaa kertoimet (3) $X < 0$ -alueella gradientti on nolla. Siksi painokertoimia ei päivitetä (kuoleva ReLU-ongelma)
Leaky ReLU	$\max(0,x)$	(1)ReLU:n muunnelma (2)Korjaa kuolevan ReLU:n ongelman	(1)Ei voida käyttää monimutkaisissa luokituksissa
ELU	[0,inf]	(1)Vaihtoehto ReLU:lle (2)Lähtö on tasaisempi	(1)Voi hajoittaa kertoimet
Softmax	Laskee todennäköisyydet	Yleensä käytetään lähtökerroksessa	

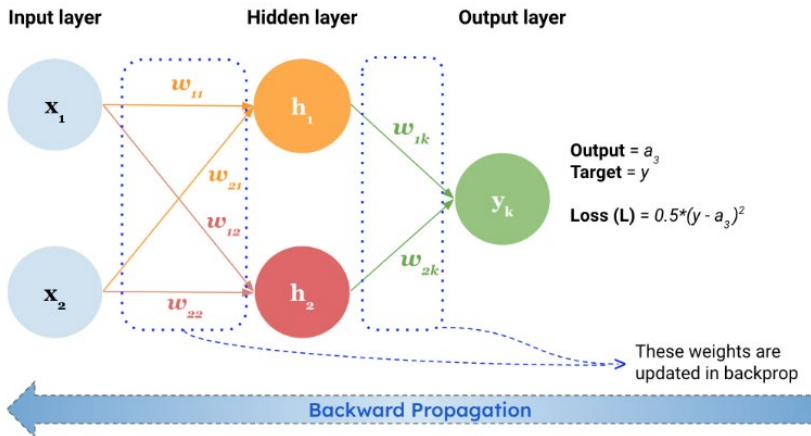
Yksinkertaisista neuroverkoista, missä piilokerroksia on vähän, kutsutaan yleensä MLP-nimellä. Nimitys tulee englannin kielen sanoista Multilayer Perceptron. Yksinkertaisissa neuroverkoissa hyödynnetään prosessia, missä tieto kulkee syötekerroksista piilokerrosten kautta ulostuloon (ks. kuvio 20). Mallilla ei ole palautteita tai toistuvia yhteyksiä. Tällaisen menetelmää kutsutaan forward propagation -termillä. (What is a multilayer perceptron (MLP) or a feedforward neural network (FNN)? N.d.)



Kuvio 20. Neuroverkon forward propagation -menetelmän toimintaperiaate (What is a multilayer perceptron (MLP) or a feedforward neural network (FNN)? N.d.)

Mallin opettaminen tarkaksi vaatii yleensä myös takaisinkytkennän mallin tuloksista tietyillä painokertoimilla. Kun syöte on ajettu läpi, voidaan tulosta verrata todelliseen arvoon ja laskea mallin tekemä virhe. Tämän jälkeen virhe etenee verkon läpi takaperin ja säätää painokertoimia ja bias-arvoja virheen pienentämiseksi (ks. kuvio 21). Tällaista menetelmää kutsutaan backpropagation-termillä. (What is a multilayer perceptron (MLP) or a feedforward neural network (FNN)? N.d.)

Painokertoimien ja bias-arvojen muuttamisessa käytetään Learning rate -arvoa, mikä kuvastaa, miten paljon arvoja korjataan jokaisella iteraatiokierroksella. Learning rate -arvon ollessa pieni, mallin opettamiseen saattaa kulua paljon aikaa. Toisaalta liian isolla arvolla malli ei pysty muuntamaan arvoja riittävän tarkasti, minkä takia mallista tulee epätarkka. Monesti opetuksen alkuvaiheessa arvo voi olla suurempi, minkä jälkeen sitä pienennetään, että päästään entistä tarkempiin tuloksiin. (Kneusel 2021, kappale 9.)



Kuvio 21. Backpropagation-menetelmän toimintakuvaus (What is a multilayer perceptron (MLP) or a feedforward neural network (FNN)? N.d.)

Taulukossa 3 on listaus esitellyistä koneoppimisessa käytettävistä algoritmeista.

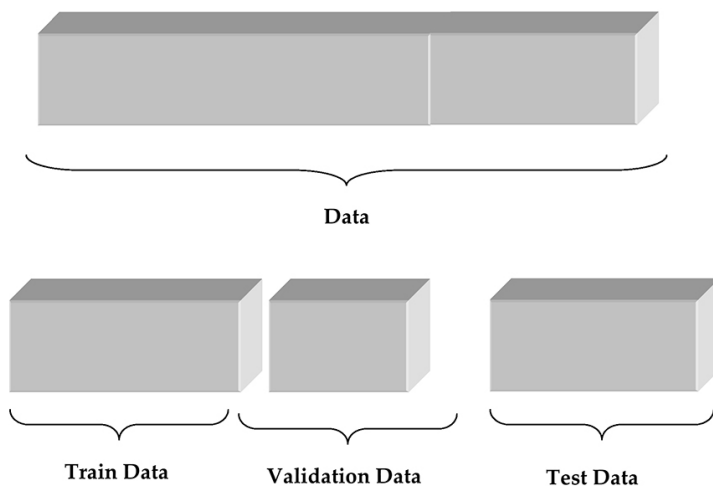
Taulukko 3. Koneoppimisessa hyödynnettävien algoritmien listaus.

Algoritmi	Kuvaus	Regressio	Luokittelu	Ohjaamaton ja ohjattu oppiminen
Lineaarinen regressio	Yksinkertainen suoran sovitus aineiston suhteen. Lähdön arvo numeerinen.	Hyödynnetään regressiomalleissa Sovitetaan vain yksi suora aineiston suhteen	Ei sovellu luokitteluongelmiin	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa
Logistinen regressio	Yksinkertainen algoritmi luokitteluun. Lähtö 1 tai 0.	Ei sovellu regressiomalleihin	Hyödynnetään luokittelussa Arvo vaihtelee välillä 0 - 1 Käytetään aktivointifunktiota esim. Sigmoid-funktiota	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa
Decision Tree	Yksinkertainen puurakenne. Soveltu hyvin selkeän tai pienen aineiston luokitteluun.	Voidaan hyödyntää regressiomalleissa, mutta soveltuu paremmin luokitteluun	Hyödynnetään luokittelussa Yksinkertainen ja nopea Ei sovellu isojen aineistojen käsittelyyn	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa
Random Forest	Decision Tree -mallien yhdistelmä. Soveltu suurten ja haastavien aineistojen käsittelyyn.	Voidaan hyödyntää regressiomalleissa, mutta soveltuu paremmin luokitteluun	Hyödynnetään luokittelussa Soveltu myös isoihin aineistoihin	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa
KNN	Nopea opetus. Toimii vain ohjatun oppimisen kanssa.	Hyödynnetään regressiomalleissa	Hyödynnetään luokittelussa	Voidaan hyödyntää vain ohjatussa oppimisessa
Tukivektorikone SVM	Soveltu haastavan aineiston käsittelyyn.	Voidaan hyödyntää regressiomalleissa	Hyödynnetään luokittelussa Soveltu monimuotoisen aineiston käsittelyyn	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa
Naivi-Bayes	Soveltu suurten aineistojen käsittelyyn. Erilaisia menetelmiä erityyppiselle aineistolle.	Hyödynnetään regressiomalleissa	Hyödynnetään luokittelussa	Voidaan hyödyntää vain ohjatussa oppimisessa
Klusterointi	Voidaan käyttää ohjaamattomassa oppimisessa. Erilaisia menetelmiä erityyppiselle aineistolle.	Ei sovellu regressio-ongelmiin	Hyödynnetään luokittelussa	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa
Neuroverkot	Laaja toiminta erityyppisiin ratkaisuihin. Käytetään paljon konenäköön liittyvissä ratkaisuissa.	Hyödynnetään regressiomalleissa	Hyödynnetään luokittelussa	Voidaan hyödyntää ohjaamattomassa ja ohjatussa oppimisessa



### 3.4.2 Koneoppimismallin toiminnan arviointi

Koneoppimisprosessin oleellisena osana on tunnistaa, miten opetusprosessi toimii ja millaisia tuloksia parametrien muuttaminen aiheuttaa tunnistustuloksiin. Mallia opetetaan ensin tietyllä aineistolla ja painokertoimia korjataan opettamiseen käytettävien toimintojen perusteella. Kun mallia opetetaan useamman kerran samalla aineisto, voi malli helposti ylioppia tunnistamaan aineistossa esiintyviä asioita. Tämän takia onkin tärkeää, ettei opettamista tehdä koko käytettävissä olevalla aineistolla. Osa aineistosta on hyvä jättää mallin toiminnan testaukseen. Tyypillisesti jakona käytetään 70 % aineistosta opettamiseen ja loput 30 % mallin tarkkuuden testaamiseen. Mallin parametrit säädetään opetusaineiston avulla, minkä jälkeen mallin toimivuus varmenneetaan testiaineiston avulla. (Jordan 2017b; Moore n.d.)



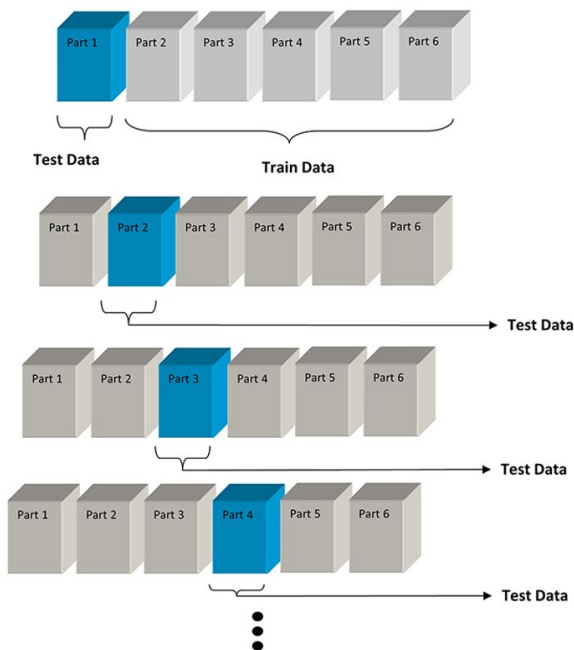
Kuvio 22. Aineiston jakaminen mallin testausta ja validointia varten (Bashin 2023, kappale 5.)

Opetusaineisto voidaan jakaa myös kolmeen osaan, jos aineiston koko on riittävän suuri. Kolmas aineisto käytetään mallin testaamiseen jokaisen opetuskerran jälkeen. Tällaista aineistoa kutsutaan Validation-aineistoksi. Validointi tehdään siis jokaisen opetuskerran jälkeen arvioimaan opetuksen tulosta ja säätämään mallin hyperparametreja. (Ks. kuvio 22) (Bashin 2023, kappale 5.) Taulukossa 4 on esitetty aineiston jakamisen osiot ja niiden käyttötarkoitus.

Taulukko 4. Aineiston hyödyntäminen mallin sovittamisessa

Opetusaineisto	Käytetään mallin sovittamiseen
Validointiaineisto	Käytetään antamaan puolueeton arvio mallin toiminnasta parametrien viritysvaiheessa
Testiaineisto	Käytetään antamaan puolueeton arvio lopullisen mallin toiminnasta

Mallin validoinnissa voidaan käyttää myös Cross-Validation -menetelmää, eli ristiin validointia. Tämä tarkoittaa, että aineisto jaetaan useampaan osajoukkoon. Osioiden määrän pystyy valitsemaan ja määrittämään aineistokoon perusteella. Kuviossa 23 on esitetty aineiston jako kuuteen osaan. Yksi osiosta valitaan testiaineistoksi ja loput viisi käytetään opetusaineistona. Toimenpide toistetaan siten, että jokainen osio on vuorollaan testiaineistona ja mallin toimintavarmuus ilmoitetaan näiden mallien keskiarvona. (Bashin 2023, kappale 5.)



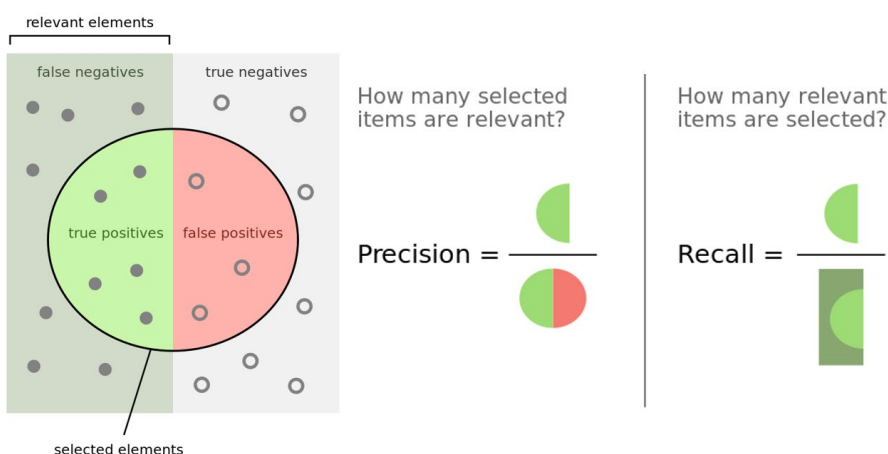
Kuvio 23. Ristiin validointi kuudella osiolla (Bashin 2023, kappale 5.)

Mallin toimivuutta voidaan mitata erittäin monella menetelmällä. Tärkeintä on kuitenkin ymmärtää, minkä tyyppistä menetelmää kannattaa hyödyntää mallin tarkasteluun. Luokittelussa, regressio-ongelmissa ja klusteroinnissa on oman tyyppiset menetelmän, millä mallin toimintaa pystytään

arvioimaan. (Ghosh 2024.) Seuraavissa tekstikappaleissa on esitetty yleisimpiä menetelmiä mallin tarkkuuden määrittämiseen.

Luokittelussa mallin toimintaa mitataan usein loss- tai cost-funktioden avulla. Arvo saadaan vertaamalla ennusteiden virhettä todelliseen arvoon, eli se kuvastaa miten hyvin malli pystyy tunnistamaan syötteen annettun aineiston. Tämä antaa mallille palautetta, minkä perusteella sen parametreja säädetään ja näin ollen pyritään pienentämään loss-funktion arvo mahdollisimman lähelle nollaa. Loss- ja cost-funktioilla on ero, liittyen tarkasteltavaan aineistoon. Loss-funktion tulos on aina yhden opetuskerran tulos, kun taas cost-funktion tulos viittaa koko opetuskerran keskiarvoiseen tulokseen. (What is gradient descent? N.d.)

Toinen tapa mitata mallin toimintaa on määrittellä mallin tarkkuus. Tarkkuuden määrittämisessä käytetään yleensä kolmea eri menetelmää, joita ovat Accuracy, Precision ja Recall. Accuracy saadaan laskettua, kun oikeat ennusteet jaetaan ennusteiden kokonaismäärällä. Tämä menetelmä ei kuitenkaan ole aina paras menetelmä määrittelyyn, varsinkin jos luokkia on paljon ja aineisto ei ole tasaisesti jakautunut eri luokkien välille. Tällaisessa tapauksessa on parempi käyttää Precision -menetelmää, mikä kuvastaa paremmin miten eri luokkiin kohdistuneet arvaukset ovat osuneet oikeaan. Arvo saadaan jakamalla oikeat ennusteet kokonaisennusteiden määrällä. Recall-menetelmä kuvastaa oikeiden ennusteiden suhdetta kaikkiin todellisiin oikeisiin tuloksiin. (Ks. kuvio 24) (Jordan 2017b.)



Kuvio 24. Precision- ja recal-toimintojen erojen esittely (Jordan 2017b.)

Tuloksia voidaan kuvata myös Confusion Matrix -taulukon avulla. Taulukkoon kirjataan tulokset oikeista ja vääristä arvauksista jokaisessa luokassa (ks. taulukko 5). Taulukon avulla on selkeä havainnollistaa väärin tulosten määrää. (Bashin 2023, kappale 5)

Taulukko 5. Confusion matrix -taulukkoesimerkki. (Bashin 2023, kappale 5.)

The decision of the classifier	Actual class	Decision designated as
Class 0	Class 0	True Negative (TN)
Class 0	Class 1	False Negative (FN)
Class 1	Class 1	True Positive (TP)
Class 1	Class 0	False Positive (FP)

Regressiomallien tapauksessa tuloksen arviointia ei voida tehdä samanlaisella oikein / väärin -menetelmällä, kuin luokittelumalleissa. Regressiomallien tapauksessa mallin arvausta pitää verrata numeeriseen arvoon ja päätellä siitä, miten hyvän tuloksen malli tuottaa. Regressiomallien toimivuuden validointiin on olemassa useita erilaisia menetelmiä, mutta yksi ehkä yleisimmin käytetty menetelmä on tilastollinen Mean Square Error, eli MSE-menetelmä. Mitä lähempänä arvot ovat regressiosuoraa, sitä vähemmän mallissa on virhettä. Menetelmä tarkastelee mallin arvauksen ja oikean tuloksen eron neliötä. Jokaisen arvauksen virheen neliöiden summa jaetaan elementtien kokonaismäärällä, mistä saadaan laskettua keskimääräinen neliövirhe. Menetelmästä on olemassa myös Root Mean Square Error, eli RMSE-menetelmä, missä arvosta otetaan vielä neliöjuuri. RMSE voidaan laskea yhtälöllä 3:

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_1)^2} \quad (3)$$

Missä  $n$  = näytteiden määrä  
 $y_i$  = näytteen todellinen arvo  
 $\hat{y}_1$  = mallin arvaus

RMSE-menetelmällä pystytään alentamaan virheasteikkoa MSE-malliin verrattuna. Menetelmällä saatua arvoa käytetään hyödyksi mallin hyperparametrien muokkaamisessa. Tavoitteena on saada virhe nolliin, mutta tämä ei käytännössä ole koskaan mahdollista. (Bashin 2023, kappalle 6.)

Toinen yleisesti käytetty menetelmä regressiomallien tarkasteluun on Mean Absolute Error -menetelmä, eli MAE-menetelmä, minkä avulla pystytään vähentämään yksittäisten paljon poikkeavien arvojen vaikutusta tulokseen. MAE pystytään laskemaan yhtälöllä 4:

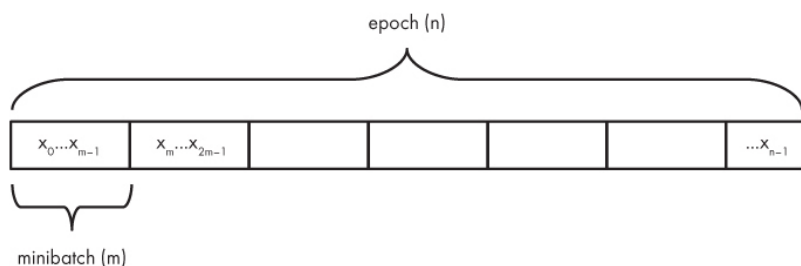
$$MAE = \frac{1}{n} \sum |y_i - \check{y}_1| \quad (4)$$

Missä  $n$  = näytteiden määrä  
 $y_i$  = näytteen todellinen arvo  
 $\check{y}_1$  = mallin arvaus

MAE-menetelmää hyödynnetään yleisesti, kun aineistossa on paljon poikkeamia. (Bashin 2023, kappelle 6.)

Klusterointialgoritmien rakenteen takia etäisyyden määrittämiseen toimivat mittarit ovat hyödyllisiä. Tällaisia menetelmiä ovat muun muassa Dunn Index -, Silhouette Coefficient – ja Elbow-menetelmät. Dunn Index -menetelmä pyrkii tunnistamaan tiiviitä klustereita ja laskee niiden välisen etäisyyden. Lisäksi klustereiden sisäinen pisteiden etäisyys lasketaan. Näistä arvoista saadaan määritettyä indeksiarvo, mikä kuvastaa mallin hyvyttä. Silhouette Coefficient -menetelmässä lasketaan klusterin pisteiden keskimääräinen sisäinen etäisyys. Lisäksi lasketaan erottelu, mikä tarkoittaa etäisyyttä lähimpään naapuriklusteriin. Menetelmän arvo vaihtelee välillä  $-1$  ja  $1$ , missä  $1$  tarkoittaa klustereiden olevan hyvin erillään toisistaan. (Ghosh 2024.)

Opetusaineiston ollessa pieni, voidaan koko aineistoa käyttää mallin opettamiseen. Eli ajaa koko opetussarja mallin läpi ja tehdä hyperparametrien muutokset. Mutta aineiston ollessa iso, voidaan puhua sadoistatuhansista tai jopa miljoonista näytteistä, niin mallin opettaminen menee liian raskaaksi ja aikaa vieväksi, kun jokainen näyte ajetaan mallin läpi, tehdään tarkkuuden arviointi ja päivitetään hyperparametrit. Toisaalta aineistoa halutaan mahdollisimman paljon, että otanta on riittävä ja mallista tulisi tarkempi. Tällaisessa tapauksessa aineisto voidaan jakaa pienempiin eriin, eli Batcheihin, mitkä ajetaan mallin läpi. Yhtä opetuskertaa, kun aineisto on ajettu mallin läpi, kutsutaan termillä Epoch. (Ks. kuvio 25) (Kneusel 2021, kappale 9.)



Kuvio 25. Opetusaineiston jakaminen pienempiin eriin (Kneusel 2021, kappale 9.)

## 4 Konenäkö

Konenäöllä tarkoitetaan tietokonepohjaista kuvien piirteiden tunnistamista ja analysointia (Gonzalez & Woods 2008, 13). Tekniikka on kehittynyt ja sen käyttö on yleistynyt kovaa vauhtia 1960-luvulta lähtien. Laitteiden laskentakyky on kehittynyt huimaa vauhtia, mikä on mahdollistanut monipuolisempien ja raskaampien konenäkösovellusten hyödyntämisen. Tämän lisäksi kamerajärjestelmissä käytettävien komponenttien hinta ja koko on pienentynyt, minkä ansiosta kameroita pystytään asentamaan ja hyödyntämään entistä useammassa kohteissa. Ensimmäiset tuotantoautomaatioissa käytetyt konenäköjärjestelmät liittyivät tuotteiden paikoittamiseen ja mitaamiseen. Tänä päivänä konenäköjärjestelmiä hyödynnetään erittäin laajasti muun muassa tuotteiden tunnistamisessa, dimensiomittauksissa sekä vikojen ja laatuvirheiden havaitsemisessa. (Siltala & Gautam 2023, 172–173.)

Ihminen on luonnostaan hyvä tunnistamaan kuvista ominaisuuksia ja muutoksia. Esimerkiksi, jos kuvaa on muokattu digitaalisesti tai kuvassa on jotain normaalista poikkeavaa, pystyy ihminen tekemään tunnistuksen, vaikka alkuperäinen kuva tai ympäristö ei olisikaan tiedossa. Kuten Azad, Gockel ja Dillmann (2008, 13.) toteavat, konenäköjärjestelmät hyödyntävät monesti vain tiettyä fyysistä ominaisuutta, oli se sitten kolmiomittaus, siluetin leikkaus tai varjojen tunnistaminen. Ihminen sitä vastoin hyödyntää alitajunteisesti aikaisemmin opittuja tietoja, millä kuvasta tai ympäristöstä pystytään tunnistamaan ominaisuudet mahdollisimman tehokkaasti. Ihminen yhdistelee monipuolisesti useita eri muuttujia kuten syvyytsvaikutusta, varjostumia, terävyyttä ja värejä. Näiden kaikkien tietojen yhdistäminen ja analysointi saman aikaisesti tekee silmän ja aivojen yhdistelmästä perinteistä konenäköjärjestelmää selvästi tehokkaamman. Toisaalta konenäköjärjestelmällä pystytään havainnoimaan laajasti asioita, mitä ihmisen on vaikea hahmottaa tai silmällä ei pysty

edes näkemään. Tällaisia kohteita ovat esimerkiksi viivakoodien ja merkintöjen lukeminen tuotteista, kappaleen tarkkojen mittojen tai paikkatiedon selvittäminen, värien analysointi ja mikroskooppisen pienten tai makroskooppisen suurten kohteiden tarkastelu. (Azad ym. 2008, 13–14.) Vaikka kameroiden ominaisuudet on pyritty rakentamaan silmän kaltaiseksi, löytyy myös kamera-laitteita, joilla pystytään kuvaamaan ihmissilmän alueen ulkopuolelta. Tällaisia laitteita ovat esimerkiksi ultravioletti- ja infrapuna-alueella toimivat kamerat. (Gonzalez & Woods 2008, 14.)

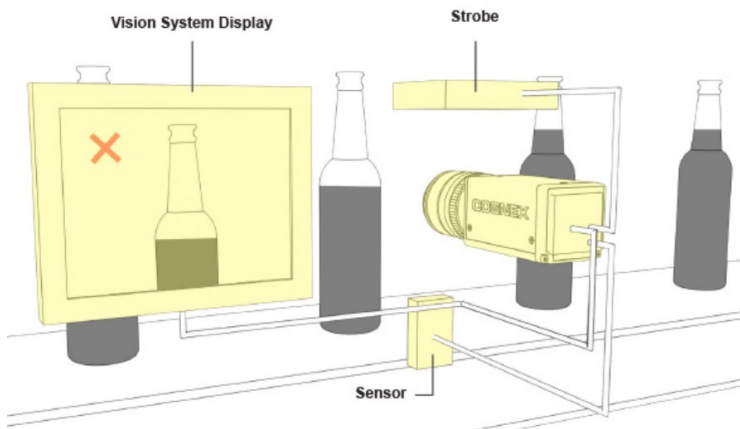
Perinteistä konenäkö hyödynnetään yleensä varsin yksinkertaisten operaatioiden avulla, mitkä eivät tarvitse tekoälyä avuksi. Tekstien pitää olla selkeitä ja samalla kirjaisintyyllillä tehtyjä, tai esimerkiksi viivakoodeja. Muotojen pitää olla ennustettavissa ja sopia tiettyyn malliin. Perinteistä konenäköä ei pysty hyödyntämään esimerkiksi käsin kirjoitettuihin kirjaimiin tai numeroihin vaihtelevien ominaisuuksien takia. Tästä huolimatta perinteisellä konenäöllä on erittäin iso merkitys tuotantoon. Konenäön avulla tuotannon laatua ja tehokkuutta pystytään tehostamaan, tuotteiden läpimenoaika pystytään nopeuttamaan, tuotantohävikkiä syntyy vähemmän ja henkilökunta pystytään siirtämään tärkeämpiin ja kriittisimpiin tehtäviin yksinkertaisista tarkastus- tai kokoonpanotehtävistä. (What Is Machine Vision?)

Kameroita hyödyntävistä järjestelmistä käytetään monesti termejä machine vision ja computer vision. Joissain tapauksissa termit tarkoittavat samankaltaisia järjestelmiä, mutta termeillä on myös omat määrittelynsä. Machine vision -termiä käytettäessä tarkoitetaan teollisuuden järjestelmää, millä pystytään ottamaan kuvia. Computer vision -termillä tarkoitetaan yleensä järjestelmää, missä tietokonetta käytetään kuvan prosessointiin ja siitä saatavan datan hyödyntämiseen. Machine vision -laitteilla saattaa monesti olla myös rajoittuneempi laskentakapasiteetti, kuin computer vision -laitteella. (Yasar, K. 2023.)

#### **4.1 Konenäköjärjestelmän rakenne**

Konenäköjärjestelmä rakentuu yleensä kamerasta, optiikasta, valaistuksesta ja prosessointiyksiköstä (ks. kuvio 26). Kamerassa olevan valoherkän kennon avulla muodostetaan digitaalinen harmaasävy- tai värikuva. Tänäkin päivänä yleisesti käytettävissä olevat kennotekniikat on kehitetty jo 60-luvulla. Yleisesti uusissa kameroissa käytetään tänä päivänä CMOS-kennoja, jonka lyhenne tulee sanoista Complementary metal-oxide-semiconductor. Vanhemmissa kameramalleissa

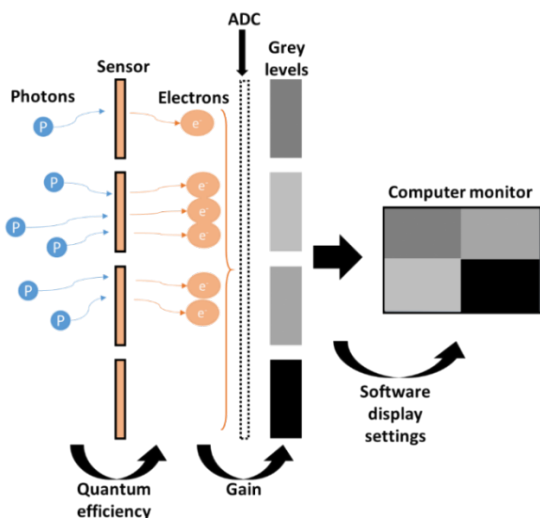
käytettiin Charge-Coupled-device, eli CCD-kennoja. (Siltala & Gautam 2023, 178; Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)



Kuvio 26. Esimerkki älykameran käytöstä (Cognex What is machine vision n.d.)

#### 4.1.1 Kamerakennon rakenne

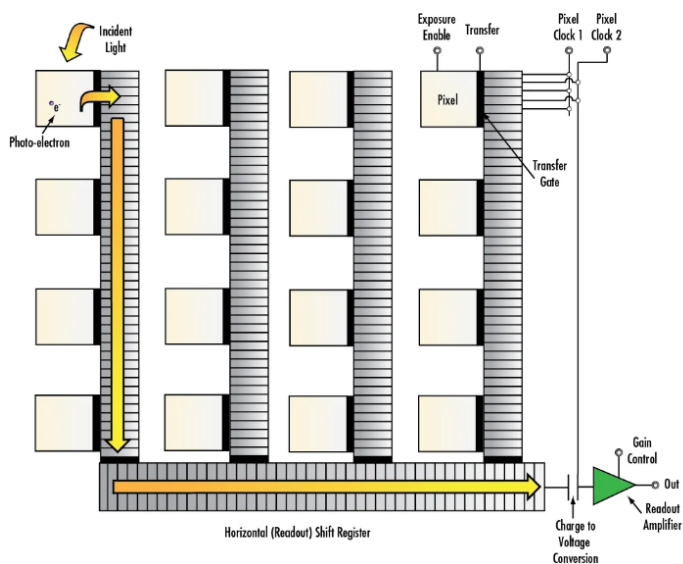
Kameran kenno sisältää useita yksittäisiä piineliöitä, joita kutsutaan pikseleiksi. Toisin sanoen pikseli on kamerakennolla pienin yksittäinen valotettava alue. Mitä enemmän kennolla on pikseleitä, sitä tarkempi kuva pystytään muodostamaan. Anturilla olevat ohuet piikerrokset muuntavat niihin osuvat fotonit elektroneiksi, jotka varastoidaan ja muunnetaan analogisesta digitaaliseksi arvoksi (ks. kuvio 27). (Introduction To Modern Camera Technologies. N.d.)



Kuvio 27. Kamerakennon toimintaperiaate (Introduction to Modern Camera Technologies. N.d.)

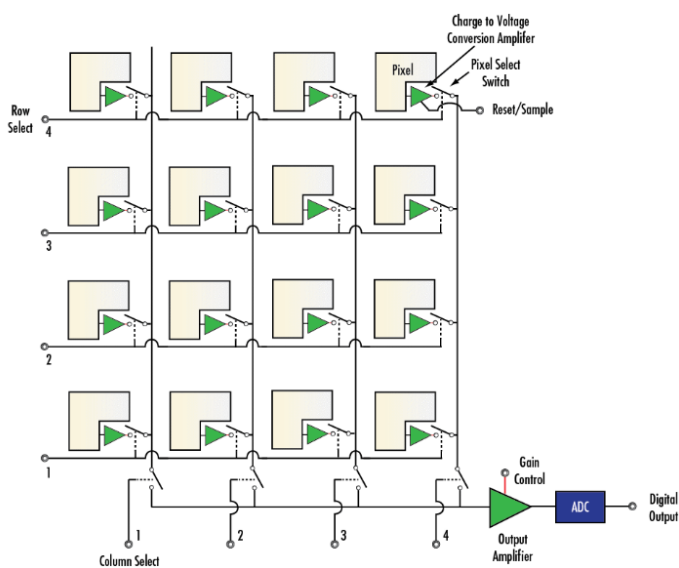


Vaikka CMOS-kennot ovat viime aikoina yleistyneet, kumpaakin tekniikkaa löytyy erilaisista kamerajärjestelmistä. Kummassakin kennotekniikassa hyödynnetään valosähköistä vaikutusta, missä fotonien määrä muutetaan sähköiseksi signaaliksi. Perusero CMOS- ja CCD-kennojen välillä on sähköisen signaalin lukemisessa ja kuvien muodostamisessa. CCD-kennossa pikselit luetaan ensin siirtorekisteriin, mistä ne siirretään tahdistetun pulssin avulla analogiselle vahvistimelle. Vahvistimen jälkeen analogiset arvot muutetaan prosessorille käsiteltävään digitaaliseen muotoon AD-muuntimen avulla. (Ks. kuvio 28) CCD-kennon rakenteen haittana on tiedonsiirron rajoittuminen pakettien siirtonopeuteen. Toisaalta rakenteesta on myös hyötyä. Jokainen paketti vahvistetaan samalla vahvistimella, jolloin kennon herkkyys on tasaisempi koko kuva-alueella. Lisäksi vahvistimista ja muuntimesta aiheutuva häiriön määrä on vähäinen. Vahvistuksia ja muunnoksia tehdään vain rajallinen määrä kennon koosta ja ominaisuuksista riippuen. Rakenteen takia kuvaan voi aiheutua myös häiriötä varauksen siirtymänä viereisiin valoherkkiin pikseleihin. Jokaisella pikselillä on maksimi varaustaso, toiselta nimeltä kaivon syvyys, mikä kuvastaa miten paljon varausta pikseliin pystyy varautumaan. Ylivuoto näkyy kuvissa kirkkaiden kohteiden sumentumisena. (Kumar 2023; Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)



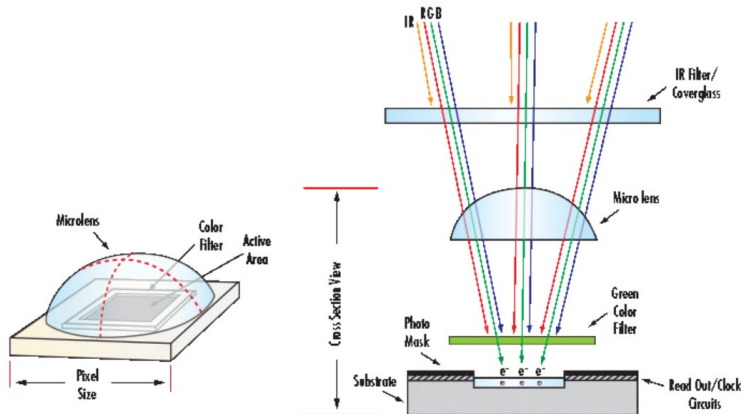
Kuvio 28. CCD-kennon toimintaperiaate. (Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)

CMOS-kennossa jokaisen pikselin varaus vahvistetaan ja muunnetaan jännitesignaaliksi pikselin sisällä. Jokaisessa pikselissä on fotodiodi ja kolme transistoria. Transistoreiden tehtävänä on aktiivoida tiedon lukeminen ja nollata pikselin arvo lukemisen jälkeen, vahvistaa ja muuntaa varaus sekä multipleksoida arvo paikkatiedon mukaan. CMOS-kennossa useita pikseliarvoja pystytään lukemaan samanaikaisesti, tai kennolta voidaan lukea esimerkiksi yksittäisiä pikseliarvoja keskeltä kuvaa. (Ks. kuvio 29) CMOS-kennolta voidaan myös helposti rajata halutun kokoinen kuva-alue. CMOS-kennot ovat erittäin nopeita, mutta kennot eivät ole niin herkkiä, kuin CCD-kennot. Kuvassa esiintyy usein myös paljon enemmän kohinaa, mikä johtuu useista jännitemuuntimista kennon sisällä. Nykyaikaisiin CMOS-kennoihin on pyritty tekemään ominaisuuksia, millä häiriötä pystytään vähentämään ja suodattamaan. CMOS-kennoilla pystytään käyttämään global shutter -toimintoa, millä tarkoitetaan jokaisen pikselin samanaikaista valottamista. Näin ollen kuvaan ei tule vääristymää esimerkiksi liikkeen takia. Monissa kameroissa käytetään kuitenkin rolling shutter -toimintoa, mikä tarkoittaa, että pikselit valotetaan ja luetaan järjestyksessä. Rolling shutter -toiminta aiheuttaa vääristymään varsinkin nopeasti liikkuvien kohteiden kuvauksessa. CMOS-kennoilla on kyky käsitellä korkeita valotasoja ja niiden vaihteluita ilman samanlaista ylivuotoa, kuin CCD-kennossa. Tämän takia CMOS-kennoilla pystytään kuvaamaan paremmin haastavia kohteita, kuten hitsaussaumoja tai valokuituja. Myös tehonkulutuksellisesti CMOS-kenno on taloudellisempi, kuin vastaavan kokoinen CCD-kenno. Tämän takia CMOS-kennojen käyttö varsinkin sulautetuissa järjestelmissä on erittäin suosittua. (Kumar 2023.)



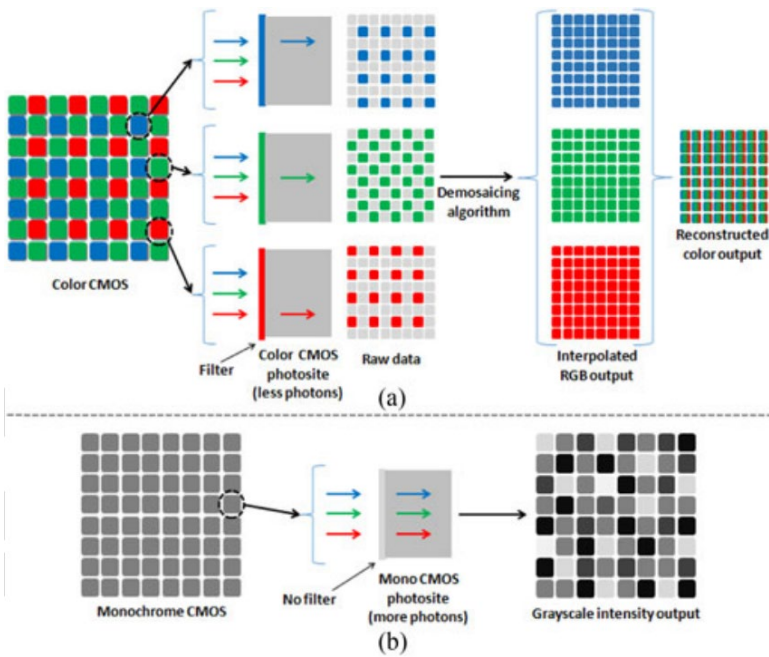
Kuvio 29. CMOS-kennon toimintaperiaate (Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)

Värikameroiden kennorakenne poikkeaa hieman harmaasävykameroiden rakenteesta. Harmaasävykamaroissa jokainen pikseli tuottaa harmaasävykuvan, eli kuvan tarkkuus on suoraan kennolla olevien pikseleiden määrä. Värikamerassa pikseleiden eteen on sijoitettu suotimet, jotka päästävät vain punaisen, vihreän tai sinisen värialueen aallonpituutta läpi. Lisäksi pikselin yläpuolelle voidaan asentaa mikrolinssi, minkä tarkoitus on kerätä valoherkälle alueelle kertyvien fotonien määrää ja kohdistaa ne hyvin halutulle alueelle. (Ks. kuvio 30)



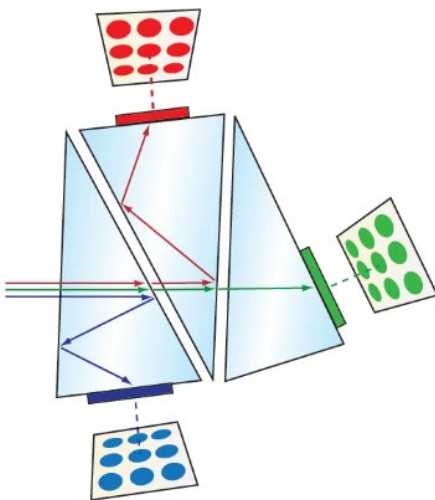
Kuvio 30. Pikselin mikrolinssi ja värisuodin (Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)

Yksikenoisissa värikamaroissa värisuotimet asetellaan pikseleiden päälle tiettyyn järjestykseen. Suotimien matriisikuvion asettelu voi vaihdella valmistajakohtaisesti, mutta yksi yleisin asettelujärjestys on Kodakin kehittänyt Bayer-järjestelmän käyttö. Järjestelmässä käytetään neljän pikselin matriisia missä kaksi kulmapikseliä ovat vihreitä, yksi kulma on punainen ja yksi kulma on sininen. Kuva muodostetaan värikuvaksi laskemalla algoritmeilla esimerkiksi pikselin viereisten pikseleiden keskiarvo. Jos käsiteltävä pikseli on esimerkiksi punainen, tarkastetaan viereisten neljän vihreän ja sinisen pikselin arvot ja kummastakin väristä lasketaan keskiarvo. Lasketuista arvoista jokaiselle kuvan pikselille lasketaan näin ollen punaisen, vihreän ja sinisen värin arvot, eli kuvasta muodostetaan RGB-kuva. Värikameroiden tarkkuus on näin ollen huonompi, kuin vastaavan harmaasävykameran, koska värien tunnistamiseen käytetään keskiarvostettuja väriarvoja. (Ks. kuvio 31) (Wilson 2016, 69)



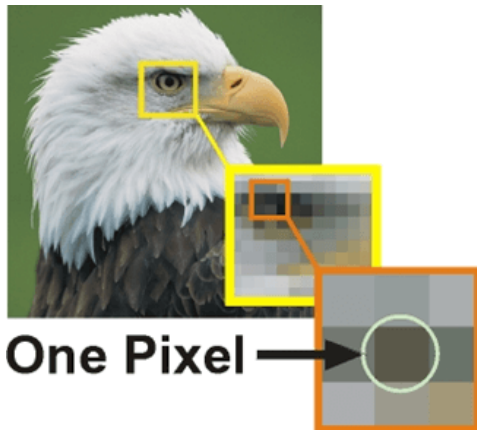
Kuvio 31. Väri- ja harmaasävykennon rakenne (Chen, Gogoi, Lee, Tsai-Lin, Yi, Lu & Hsieh 2019, s. 2.)

Värikameroita on myös kolmella kennolla varustettuina, milloin suotimista aiheutuva tarkkuuden heikentyminen saadaan vältettyä. Kamera on varustettu prismalla, millä eri aallonpituusalueet jaetaan omille kennoille. Tällainen rakenne on monesti huomattavasti kalliimpi, kuin yhden kennon ratkaisu, minkä lisäksi kameroilla saattaa olla alhaisempi valoherkkyys. (Ks. kuvio 32) (Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)



Kuvio 32. Kolmella kennolla varustettu värikamera (Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications n.d.)

Harmaasävykameraa käytettäessä pikseleiden arvot muunnetaan ohjelmallisesti harmaasävyksi ja värikameralla yleensä RGB-arvoksi. Arvoista muodostettu kuva määräytyy ohjelmistoasetuksista, kuten kirkkaudesta ja kontrastista, sekä pikseleiden bittimäärästä, eli syvyysarvosta. Väriarvon lisäksi jokaisella pikselillä on osoitetieto. Osoitetieto kertoo missä kohtaa kuvaa kyseinen pikseli sijaitsee. Kameran kennolta luetut arvot muodostetaan prosessointiyksikössä uudestaan näytettävään muotoon. (Ks. kuvio 33.) (Introduction To Modern Camera Technologies. N.d.)



Kuvio 33. Värikuvan muodostuminen pikseleistä (Gardete 2024.)

Pikselin värisyvyys määräytyy AD-muunnoksen bittien määrästä. Mitä enemmän muuntimessa on bittejä, sitä enemmän värisävyn vaihtoehtoja kuvassa on (ks. kuvio 34). Värisävyyden määrä voidaan laskea yhtälöllä 5:

$$\text{Värisävyyden määrä} = 2^{Bpp} \quad (5)$$

Missä Bpp tarkoittaa Bits per pixel, eli bittien määrä yhdessä pikselissä. Esimerkiksi kahdeksan bittisen harmaasävykuvan jokainen pikseli sisältää 256 eri harmaasävyä. Nolla tarkoittaa mustaa ja 255 arvo tarkoittaa valkoista. Värisävyyden määrä vaikuttaa myös merkittävästi kuvan tarvitsemaan muistimäärään. Kuvan sisältämä data voidaan laskea yhtälöllä 6:

$$\text{Muistivaraus} = \text{rivit} \times \text{sarakkeet} \times Bpp \quad (6)$$

Näin olleen kahdeksan bittisen harmaasävykameran kameran resoluution ollessa esimerkiksi 1000 x 1000 pikseliä, saadaan kuvan muistivarauksen määräksi:

$$\text{Muistivaraus} = 1000 \times 1000 \times 8 \text{ bittiä}$$

$$\text{Muistivaraus} = 8\,000\,000 \text{ bittiä}$$

Arvo voidaan muuttaa tavuksi jakamalla se kahdeksalla ja kilotavuiksi jakamalla luvulla 1024, jos halutaan käyttää kaksikantalukuja. Tässä tapauksessa kuvan koko voitaisiin esittää myös muodossa 976.5625 kb. Jos samassa kamerassa olisikin 24 bittinen muunnin tulisi muistivarauksen määräksi yhtälöllä 7:

$$\text{Muistivaraus} = \frac{(1000 \times 1000 \times 24)}{8 \times 1024} \quad (7)$$

$$\text{Muistivaraus} = 2929.6875 \text{ kb}$$

Eli yhden kuvan muistivaraus olisi lähes 3 megatavua. (Concept of Bits Per Pixel 2024)

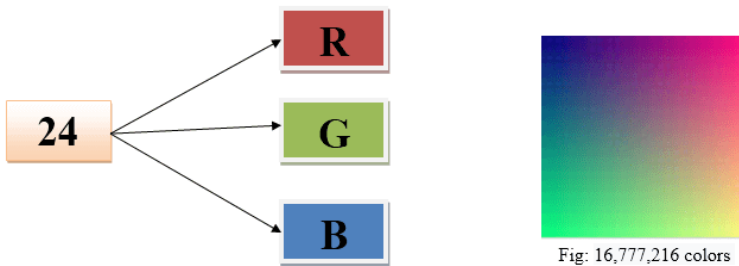


Kuvio 34. Harmaasävyarvot 2-, 4- ja 8-bittisellä muunnoksella. (Imaging Electronics 101: Camera Resolution for Improved Imaging System Performance n.d.)

Värikuvien tapauksessa värisyvyys muodostuu kolmesta kahdeksan bittisestä väriarvosta, punaisesta, vihreästä ja sinisestä, joiden avulla pystytään määrittelemään sekä väri, että intensiteetti (ks. kuvio 35). Yhteensä väriarvoja voi siis olla kaavalla 5 laskettuna:

$$\text{Värisävyyden määrä} = 2^{24}$$

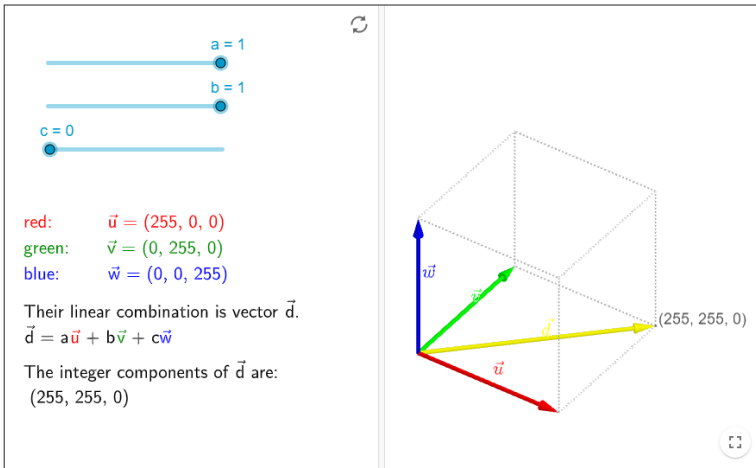
$$\text{Värisävyyden määrä} = 16777216$$



Kuvio 35. RGB-kuvan värisävyt (Types of Images n.d.)

Monesti värikuvia ei kuitenkaan tallenneta tai käsitellä 24 bittisenä sen suuren koon vuoksi. Värikuvissa voidaan käyttää myös 16 bittistä koodausta. 16 bittiä ei kuitenkaan jakaudu tasan kolmelle eri värille, minkä takia punaiselle ja siniselle värille varataan viisi bittiä ja loput kuusi bittiä on käytössä vihreälle värille. (Types of Images) Ihmissilmä pystyy erottelemaan paremmin vihreän eri sävyjä, minkä takia myös kuvissa halutaan tuoda enemmän vihreitä sävyjä esille (Azad ja muut 2008, 14–15).

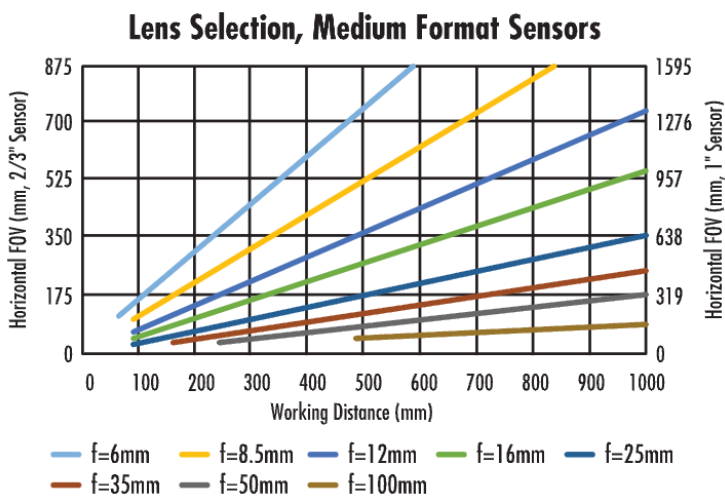
Värikuvan RGB-arvot voidaan esittää myös vektorein (ks. kuvio 36). Punainen, vihreä ja sininen väri voidaan esittää kolmiulotteisen avaruuden vektoreina  $\vec{u}$ ,  $\vec{v}$ , ja  $\vec{w}$ . Näistä kolmesta arvosta pystytään yhdistämään vektori  $\vec{d}$ , millä siis pystytään esittämään kaikki mahdolliset väri- ja intensiteettiarvot. (Geogebra n.d.)



Kuvio 36. Väriarvojen vektorit (Geogebra n.d.)

#### 4.1.2 Optiikat

Jotta kuvattavasta kohteesta tulevat fotonit saataisiin kerättyä ja kohdistettua oikealla tavalla kameran kennolla, tulee kameraan kiinnittää soveltuva optiikka. Optiikan tarkoituksena on tuottaa valovoimainen ja tarkka kuva halutulta kuva-alueelta. Konenäköjärjestelmissä käytettävät optiikat ovat yleisesti kiinteällä polttovälillä, mikä tarkoittaa, että optiikasta ei löydy esimerkiksi järjestelmäkameroista tyypillisiä zoomaavia linsejä. Tarkennuksen suhteen optiikka voi olla manuaalisesti säädettävä tai automaattitarkennukseen soveltuva moottoroitu tai nestemäinen linssi. Sopiva optiikan polttoväli valitaan kameran kennon koon, kameran asennusetäisyyden ja halutun kuva-alan perusteella (ks. kuvio 37). (Siltala & Gautam 2023, 180.)



Kuvio 37. Polttovälin vaikutus kuva-alaan (Basic Lens Selection n.d.)



### 4.1.3 Valaistuksen suunnittelu

Valaistuksen suunnittelulla ja toteutuksella on erittäin suuri merkitys konenäköjärjestelmän luotettavuuteen. Oikeanlaisella valaistuksella saattaa olla ratkaiseva merkitys koko järjestelmän toimintaan. Huonolla tai vääränlaisella valaistuksella saattaa toisaalta pilata koko järjestelmän toiminnan. Valaistuksen tärkein tehtävä on saada haluttu asia tai piirre erottumaan riittävän tehokkaasti. Esimerkiksi kappale halutaan erottaa taustasta mittausta varten, tai sopivalla sivusta tulevalla valaistuksella saadaan pinnassa olevat virheelliset muodot erottumaan paremmin. Valaistuksen ja kuvausympäristön suunnitteluun on hyvä käyttää riittävästi aikaa. Monesti kuvasympäristön olosuhteet ovat vaihtelevat, jos järjestelmää ei koteloida riittävästi. Hallin valaistus saattaa muuttua vuorokauden aikojen mukaan, tai esimerkiksi ikkunoista tuleva luonnonvalo vaikuttaa erittäin paljon valaistukseen. Ihmissilmä tottuu valonmäärään nopeasti ja emme tiedosta, vaikka valaistuksen voimakkuus vaihtelisi ympäristössä. Kamerajärjestelmä ei tällaista sopeutumista aukonsäädön tai valotusajan suhteen osaa tehdä, ellei sille erikseen komenneta käskyä esimerkiksi valotusajan automaattiseen toimintoon. (Siltala & Gautam 2023, 175–177.)

### 4.1.4 Kuvien prosessointi

Laittearkkitehtuurilla tarkoitetaan, missä kameralta saatava kuva-aineisto käsitellään. Vaihtoehtoisia järjestelmiä ovat tietokonepohjaiset järjestelmät, kompaktit konenäköjärjestelmät, älykamerat ja sulautetut järjestelmät. Tietokonepohjaisessa järjestelmässä kameran ottama kuva välitetään tietokoneen prosessorille. Yleisin tapa kuvan prosessointiin on hyödyntää tietokonetta, mutta käsittely voi tapahtua myös esimerkiksi robotin ohjaimella tai erillisellä ohjelmoitavalla logiikalla. Kuvan käsittelystä saatava tulos, oli se sitten tuotteen mitta, paikkatieto tai vaikka viallisen tuotteen tunnistus, voidaan välittää eteenpäin tarvittavan väylätekniikan avulla. Yleisesti järjestelmistä on valittavissa iso määrä erilaisia väylätekniikoita tai perinteisen I/O-liittymän kautta tehtävät kytkennät. Tietokoneella käytettävä ohjelmisto tai konenäkökirjastot voidaan yleensä valita vapaasti tarpeiden mukaisesti. Tietokonepohjaisen järjestelmän valinnassa tulee tietää tarkasti mitä kaikkia ominaisuuksia järjestelmältä vaaditaan, miten laitteet kommunikoivat keskenään, millaisia kirjastoja tarvitaan tunnistuksen toteutukseen ja miten järjestelmä ylläpidetään esimerkiksi päivitysten yhteydessä. (Siltala & Gautam 2023, 177.)

Kompakti konenäköjärjestelmä on hyvin samanlainen ratkaisu, kuin tietokonepohjainen järjestelmä. Tässä tapauksessa laitetoimittaja on valmiiksi määritellyt käytettävän kameraratkaisun ja siihen soveltuvan ohjelmiston kuvan käsittelyyn. Järjestelmät ovat hyvin testattuja ja päivitysten kautta voidaan tarjota uusia ominaisuuksia tai tarvittavia korjauksia havaittuihin puutteisiin. (Siltala & Gautam 2023, 177.)

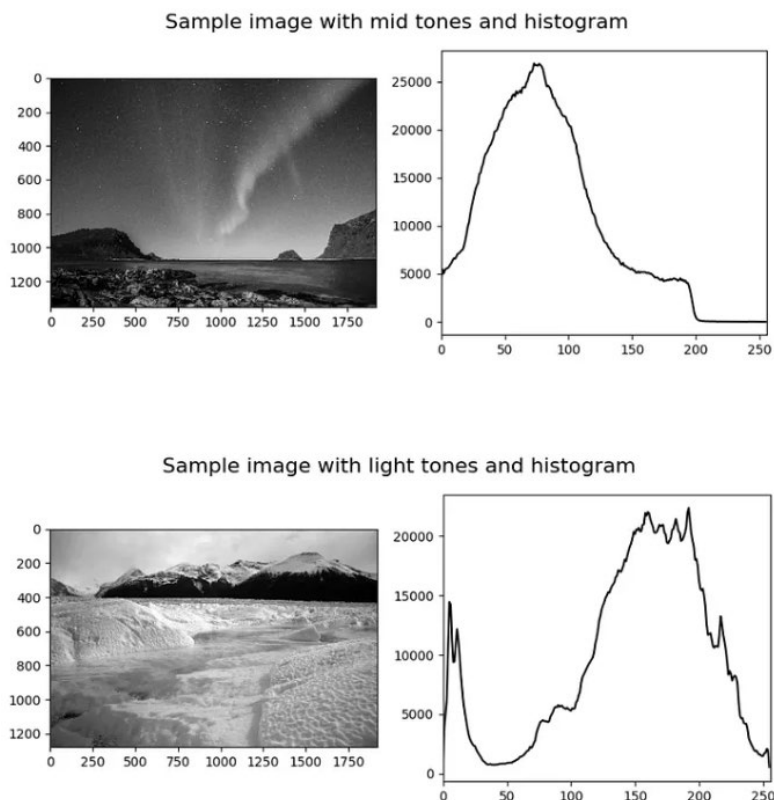
Älykameralla tarkoitetaan laitetta, missä kameralaitteeseen on sijoitettu myös kuvan prosessointiyksikkö. Lisäksi laitteessa saattaa olla integroitu valonlähde, sekä kenttäväylä- ja I/O-liitynnät laitteiden ohjaamista varten. Älykamerat ovat rakenteellisesti yleensä hyvin suojattuja ja koteloituja ja niiden asentaminen haastaviinkin ympäristöihin onnistuu helposti. Älykameroissa on yleensä rajoittuneempi prosessointiteho, kuin PC-pohjaisissa järjestelmissä. Tämän takia myös käytettävissä olevat kirjastot saattavat olla rajoittuneempia. Älykamera voi olla kuitenkin hyvä valinta paikkoihin, mihin tarvitaan vain yksi kamera, tai kamera halutaan liittää suoraan johonkin toiseen järjestelmään, kuten esimerkiksi robotti. (Siltala & Gautam 2023, 178.)

Sulautetut järjestelmät ovat yleisesti johonkin tiettyyn tehtävään optimoituja laitteita. Järjestelmän rakenne ja toiminta on hyvin samankaltainen älykameroiden kanssa. Sulautetut järjestelmät saattavat kuitenkin olla koteloimattomia ja hyvin pelkistettyjä ratkaisuja. Sulautetuissa järjestelmissä kamera on suoraan yhteydessä laitteen prosessointiyksikköön. Näin järjestelmästä saadaan yleensä kompaktin kokoinen ja kuvankäsittely saadaan lähes reaaliaikaiseksi. Esimerkkejä kuvankäsittelyyn soveltuvista sulautetuista järjestelmistä ovat esimerkiksi NVIDIAN Jetson ja TI Jacinto. (Embedded Vision vs. Machine Vision – Everything You Need to Know n.d.)

## **4.2 Konenäköfunktiot ja niiden hyödyntäminen**

Konenäköohjelmistoilla voidaan käsitellä ja analysoida kameroiden tuottamaa kuva-aineistoa. Ohjelmistot sisältävät erilaisia työkalukirjastoja, joista löytyy yleisesti toiminnot kuvan ottamiselle ja siirtämiselle, kuvan käsittelylle ja suodatukselle, sekä kuvan analysointiin ja tulosten esittämiseen. Perinteiset konenäkötoiminnot ovat yleensä sääntöpohjaisia toimintoja, jotka pohjautuvat matemaattisiin toimintoihin ja tulosten vertailuun jotakin ennalta määritettyä raja-arvoa vasten. (Siltala & Gautam, 182.)

Kuvan värisävyjen jakautuminen voidaan esittää graafisena esityksenä histogrammin avulla. Histogrammi on kuvaaja, missä näytetään jokaisen harmaa- tai värisävyyn esiintymismäärä tai intensiteetti. Histogrammin avulla saadaan selkeästi esitettyä eri intensiteettiarvojen jakautuminen kuvassa. Tästä on apua esimerkiksi kuvan kirkkauden ja kontrastin säätämisessä, tai tarvittaessa kameran ja optiikan asetusten muuttamisessa, jos kuva on liian ali- tai ylivalottunut. Lisäksi histogrammin avulla voidaan selvittää kuvan kynnystämiseen sopivaa arvoa. Kuviossa 38 on esitetty kahden erilaisen kuvan harmaasävyarvojen histogrammin. (Azad ja muut 2008, 93–94.)

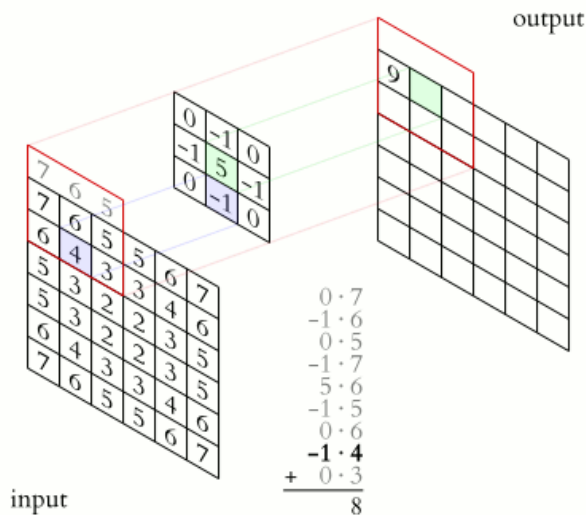


Kuvio 38. Harmaasävyarvojen histogrammit erilaisissa kuvissa (Raghunath 2019.)

### Kuvan esikäsittely suodattimilla

Erilaiset suodattimet ovat erittäin tärkeässä roolissa varsinkin kuvien esikäsittelyssä, ennen varsinaista kuva-analyysiä. Kuvien käsittelyssä törmää usein termiin konvoluutio. Konvoluutio tarkoittaa matemaattista operaatiota kahden funktion välillä, jolla tuotetaan kolmas funktio. Kuvankäsittelyssä konvoluutiolla tarkoitetaan toimintoja, millä pyritään erottelemaan kuvasta haluttuja piirteitä. Esimerkiksi erilaiset suodattimet ovat tällaisia toimintoja. Kuvien käsittelyssä toiminnot

voidaan kuvata kaksiulotteisina matriiseina, joilla käsitellään kuvan jokainen pikseli. Tällaista matriisia kutsutaan myös kerneliksi. Matriisia siirretään pikseli kerrallaan ja jokaisen pikselin, sekä matriisin koosta riippuvien viereisten pikseleiden kesken tehdään laskutoimitukset. (Marques 2011, 203–208; Azad ja muut 2008, 101–102.) Kuviossa 39 on esimerkkilaskelma 3 x 3 matriisilla tehdystä kuvamuunnoksesta.



Kuvio 39. Konvoluutioesimerkki (Gardete 2024.)

Toiminnossa siis sovitetaan 3 x 3 matriisin keskikohta vuorotellen jokaisen kuvan pikselin arvon päälle. Tämän jälkeen matriisin solun arvolla kerrotaan kuvan pikseliarvo ja tulot summataan yhteen. Tällä tavalla saadaan uuden kuvan arvot laskettua pikseli kerrallaan. Esimerkkikuvassa on toimintona kuvan terävöinti. Muita yleisesti käytettäviä suodatustoimintoja terävöinnin lisäksi ovat keskiarvoistus, reunojen tunnistus ja sumennus. Suotimet voidaan jakaa kahteen kategoriaan, alipäästäsuodattimiin. Alipäästäsuodattimet suodattavat nimensä mukaisesti ylempät taajuuDET pois ja päästävät vain alemmat taajuuDET läpi. Tämän takia alipäästäsuodattimilla pystytään vähentämään kuvien häiriöitä ja tasoittamaan kuvaa. Ylipäästäsuodattimet päästävät ylempät taajuuDET läpi ja tällaisia suodattimia voidaan hyödyntää muun muassa kuvien terävyyden parantamisessa ja piirteiden reunojen tunnistamisessa. Alipäästäsuodattimia ovat esimerkiksi keskiarvoistus ja Gaussin-suodatus. Keskiarvoistuksessa konvoluution jälkeinen arvo jaetaan suodattimen arvoilla, milloin saadaan keskiarvostettu tulos uuden kuvan pikseliarvoksi. Keskiarvo-suodatus voidaan esittää esimerkiksi yhtälöllä 8:

$$F_{average} = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (8)$$

Kuvien käsittelyssä Gaussin-suodatus perustuu kaksiulotteiseen ympyräsymmetriseen Gaussin toimintoon, mikä voidaan esittää yhtälöllä 9:

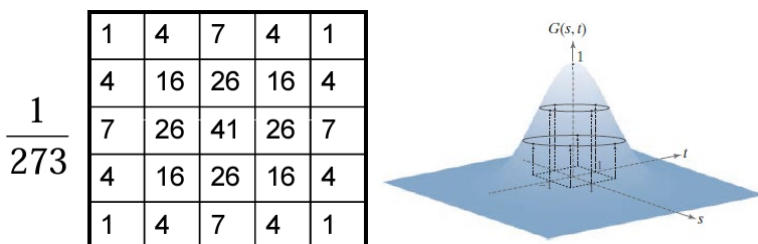
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (9)$$

Missä  $x, y$  = kuvan koordinaatit  
 $\sigma$  = jakauman keskihajonta

Kernelin keskipisteen arvo on suurin ja kertoimet pienenevät suhteessa pisteiden etäisyyteen keskipisteestä. Jakauman keskihajonnalla pystytään muuttamaan jakauman arvoa. Pienemmällä arvolla muoto on terävämpi ja suuremmalla arvolla muodosta tulee laakeampi. Suurempaa keskihajontaa käytettäessä myös kernelin kokoa täytyy kasvattaa, että suodattimen muoto saadaan säilytettyä. 3 x 3 kernelillä ja keskihajonnan arvolla yksi voidaan Gaussin suodatin esittää yhtälöllä 10:

$$F_{Gaussian} = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (10)$$

Suodatin voidaan esittää myös taulukkona, mikä helpottaa lukemista varsinkin suurempien kernelikokojen kanssa. 5 x 5 kokoinen Gaussin suodatin voidaan siis esittää myös kuvio 40 mukaisesti. (Azad ja muut 2008, 103–104; Marques 2011, 237–250)



Kuvio 40. 5 x 5 Gaussin suodatin keskihajonnalla yksi (Gonzales & Woods 2018, 168.)

## Reunantunnistus

Kuvan terävöinti ja reunojen tunnistus voidaan jaotella kahteen luokkaan, mitkä ovat gradienttimenetelmä ja toisen asteen derivaattoihin perustuvat menetelmät. Gradienttimenetelmissä tarkastellaan vierekkäisten pikselien intensiteettiarvoja, minkä perusteella määritetään kaksiulotteinen vektori. Vektorin suunta voidaan määrittää suurinta muutosta kohti, minkä avulla pystytään määrittelemään suurin intensiteetin muutoskohta. Gradientti pystytään määrittelemään yhtälöllä 11:

$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (11)$$

Missä  $x, y$  = kuvan koordinaatit

$\frac{\partial f}{\partial x}$  ja  $\frac{\partial f}{\partial y}$  = osittaisderivaatat

Gradientti määritetään usein vektorin pituuden itseisarvona, mikä voidaan laskea yhtälöllä 12:

$$M(x, y) \approx |g_x| + |g_y| \quad (12)$$

Yksi yleisesti käytettävä menetelmä reunojen tunnistamisessa vaaka ja pystytasossa on Sobel-operaatio. Sobel-operaatiossa käytetään  $3 \times 3$ -kokoisia kerneleitä, mitkä voidaan esittää yhtälöllä 13:

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \quad h_2 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (13)$$

Missä  $h_1$  = y-akselin suunta

$h_2$  = x-akselin suunta

Kuviossa 41 on esitetty Sobel-toiminto harmaasävykuvaan sovitettuna. Kuvassa on alkuperäinen kuva, x-suuntainen reunan tunnistus, y-suuntainen reunan tunnistus sekä yhdistetty reunojen tunnistus. (Gonzales & Woods 2018, 184-185; Sonka, Hlavac & Boule 2008, 135-137)



Kuvio 41. Sobel-toiminnolla tehty reunojen tunnistus (Gonzales & Woods 2018, 721.)

Muita samankaltaisia toimintoja ovat muun muassa Roberts ja Prewitt, missä käytetään hieman erilaista kerneliä (ks. Kuvio 42). (Gonzales & Woods 2018, 719.)

-1	0	0	-1
0	1	1	0

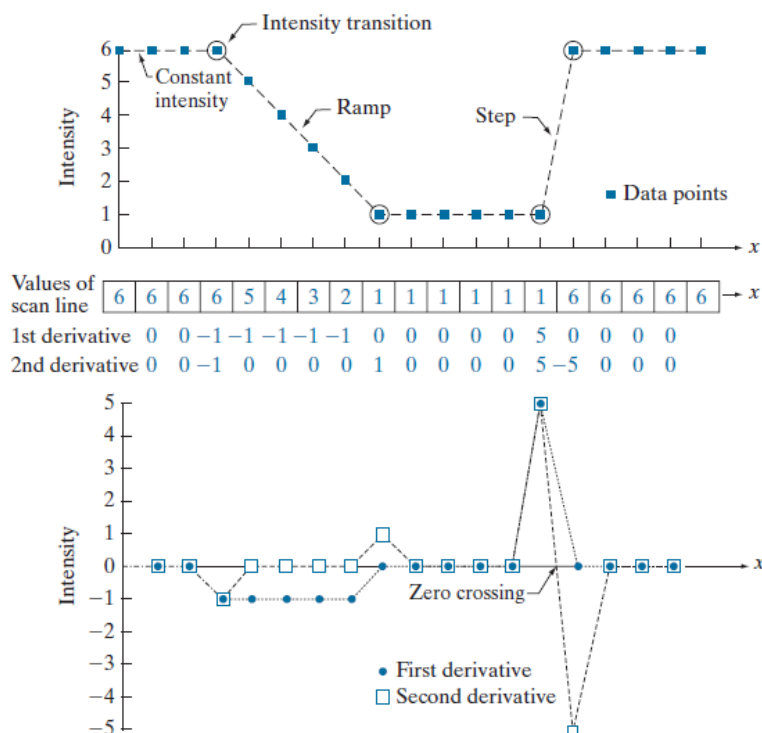
Roberts

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

Prewitt

Kuvio 42. Roberts ja Prewit -toimintojen kernelit (Gonzales & Woods 2018, 719.)

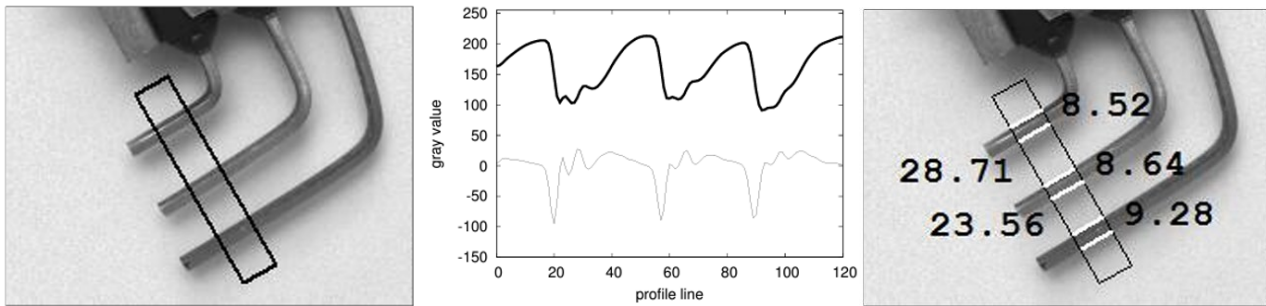
Toisen kertaluvun derivaatan avulla pystytään tunnistamaan intensiteetin muutokset, sekä määrittämään milloin arvo ylittää horisontaalisen akselin. Myös tätä toimintoa pystytään hyödyntämään esimerkiksi reunan tunnistamiseen ja paikantamiseen. Toisen kertaluvun derivaattaa hyödyntämällä reunojen tunnistus on tarkempi, eikä tunnistus ole kuvan tai ominaisuuksien koosta riippuvainen. Menetelmä on kuitenkin laskennallisesti raskaampi, kuin gradienttimenetelmä. Kuviossa 43 on esitetty ensimmäisen ja toisen kertaluvun derivaatan käyttäytyminen yhden intensiteettikuvaajan suhteen. (Gonzales & Woods 2018, 724–726; Sonka ja muut 2008, 139–140.)



Kuvio 43. Ensimmäisen ja toisen kertaluvun derivaatan hyödyntäminen muutosten havaitsemiseen (Gonzales & Woods 2018, 178.)

Kuviossa 44 on esitetty mittaustoiminto komponentin jalkojen etäisyyden mittaamiseen. Kuvajassa on esitetty myös harmaasävyarvo, sekä arvon muunnettu derivaattaa. Derivaatan avulla pystytään määrittämään tunnistusalueen rajat, minkä perusteella lasketaan tunnistettujen alueiden etäisyys. (Solution Guide III-A 1DMeasuring 2024, 10–18.)





Kuvio 44. Reunojen tunnistus derivaatan avulla (Solution Guide III-A 1DMeasuring 2024, 10. Muokattu.)

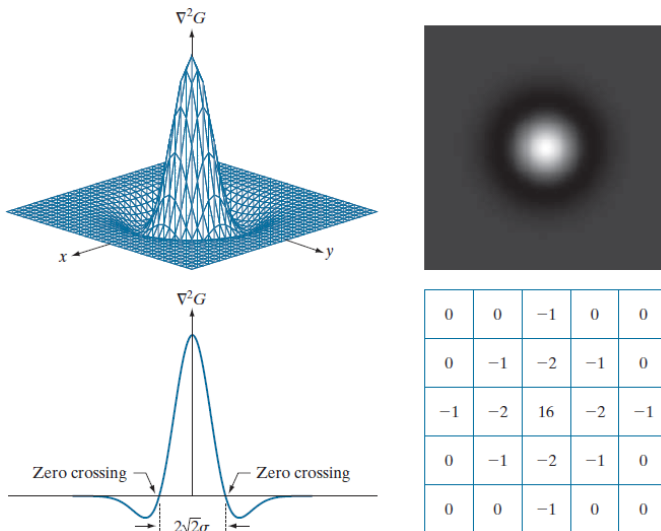
Laplace-toiminnolla pystytään tekemään tarkempia reunojentunnistuksia erikokoisissa kuvissa, sekä kohteissa, missä reunan raja ei ole tarkka. Laplace-muunnosta voidaan hyödyntää esimerkiksi Gaussin-suodattimen kanssa. Tällaista toimintoa kutsutaan LoG-termillä, eli Laplacian of Gaussian, mikä voidaan esittää yhtälöllä 14:

$$h(x, y) = c \left( \frac{x^2 + y^2 - \sigma^2}{\sigma^4} \right) e^{-(x^2 + y^2)/2\sigma^2} \quad (14)$$

missä

- x, y = kuvan koordinaatit
- $\sigma$  = jakauman keskihajonta
- c = normalisoinnin kerroin

Kuviossa 45 on esitetty 5 x 5 matriisilla määritetyn LoG-toiminnon kuvaajat, harmaasävykuva ja matriisin arvot. (Gonzales & Woods 2018, neljäs painos, 724–726; Sonka ja muut 2008, 139–140)



Kuvio 45. 5 x 5-kernelinen LoG-toiminnon muoto ja kerneli (Gonzales & Woods 2018, 726.)

Kernelin arvojen ollessa suuria tai negatiivisia, voi yhteenlaskettu arvo ylittää tai alittaa esimerkiksi kahdeksan bittisen kuvan alueen 0 ... 255. Jos negatiivisia tai liian suuria arvoja ei pystytä käyttämään käsittelyalgoritmin kanssa, tai kun kuva muutetaan näytettävään muotoon ja sen ominaisuudet halutaan säilyttää, kannattaa ali tai yli menevät arvot leikata minimiin tai maksimiin. Tästä saattaa kuitenkin aiheutua oleellisen tiedon katoamista. Arvot pystytään sovittamaan halutulle välille laskemalla siirtämällä alue alkamaan nolasta ja skaalaamalla alue. Nollapisteen siirron pystyy laskemaan yhtälöllä 15:

$$g_m = g - \min(g) \quad (15)$$

Missä  $g_m$  = kuva, missä arvot siirretty nolapisteeseen  
 $g$  = alkuperäinen arvo  
 $\min(g)$  = alueen pienin arvo

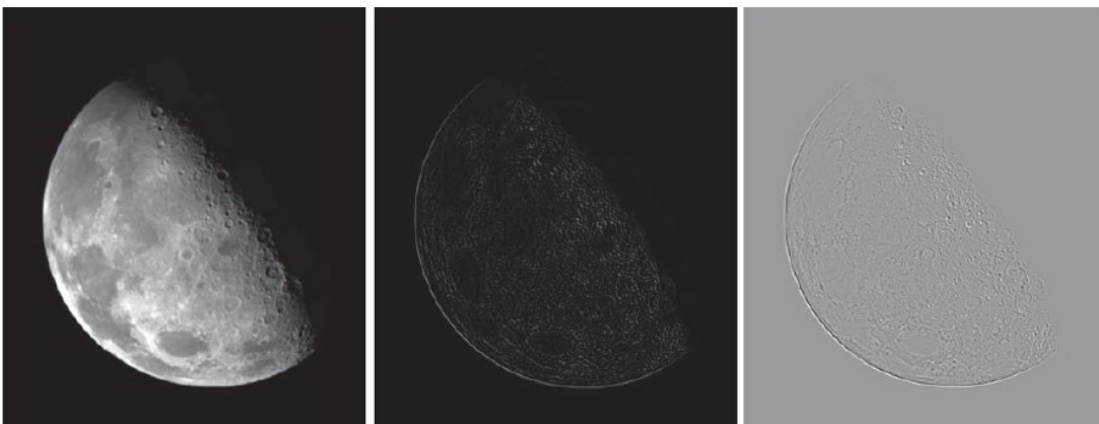
Skaalaus pystytään tekemään yhtälöllä 16:

$$g_s = K[g_m / \max(g_m)] \quad (16)$$

missä  $g_s$  = skaalattu kuva halutulle välille  
 $K$  = pikselin maksimiarvo

(Gonzales & Woods 2018, 91, 180)

Joskus arvot kuitenkin poikkeavat toisistaan niin paljon, että skaalaamalla kuva halutut arvot eivät erotu kunnolla. Kuviossa 46 on esitetty Laplace-toiminnolla suodatettu kuva kuusta, missä nolaa pienemmät arvot on muutettu nolaksi ja kuva, missä arvot on skaalattu välille 0 ... 255.



Kuvio 46. Laplace-suodatettu ja skaalattu kuva (Gonzales & Woods 2018, 181.)

Reunan tunnistus voidaan jakaa kolmeen toimenpiteeseen. Yleisesti kuvaa tarvitsee suodattaa tai muokata, että tunnistettavat piirteet saadaan paremmin esille. Tämän jälkeen käytetään reunan tunnistukseen menetelmää, millä saadaan kaikki tai vain halutun suuntaiset reunat tunnistettua. Viimeisenä vaiheena tulee reunan pisteiden tunnistaminen, eli miten haluttuun reunaan kuuluvat pisteet tunnistetaan ja paikoitetaan kuvasta. (Gonzales & Woods 2018, 714.)

### Kuvien kynnystys

Monesti kuvan käsittely harmaasävy- tai väriarvoilla vaatii liian paljon laskentatehoa. Tämän takia onkin yleistä, että kuvasta muodostetaan yksinkertaisempi binäärinen kuva Threshold, eli kynnystys-toiminnon avulla. Kuvassa olevat alueet tai piirteet voidaan yleensä luokitella pinnan ab-

sorption tai heijastuksen perusteella, kun kuvassa olevat piirteet näkyvät kirkkaampina tai tummempina kohtina. Värikuvissa kynnystys voidaan tehdä myös eri värikanavien perusteella. Kynnystystoiminto on vanhin segmentointityökalu, mutta sen yksinkertaisen ja nopean laskennan takia se on erittäin laajasti käytössä vielä tänäkin päivänä. Kynnystyksessä valitaan raja-arvo, minkä perusteella pikselin arvo muutetaan joko arvoksi 1 tai 0. Kynnystystoiminto voidaan esittää yhtälöllä 17:

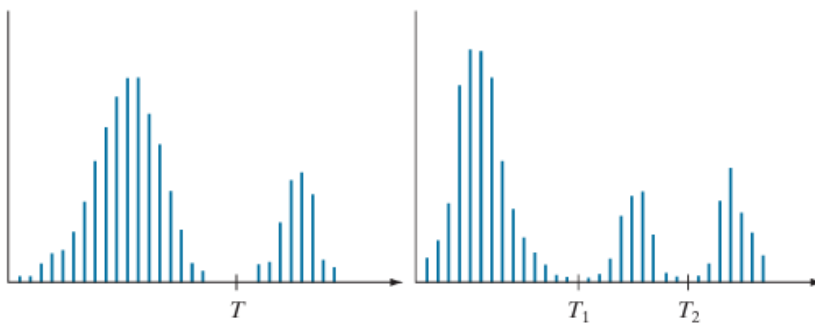
$$g(i, j) = 1 \text{ for } f(i, j) \geq T,$$

$$g(i, j) = 0 \text{ for } f(i, j) < T, \tag{17}$$

Missä  $T$  = kynnestyksen arvo

$g(i, j)$  = binääriseksi muunnettu kuva

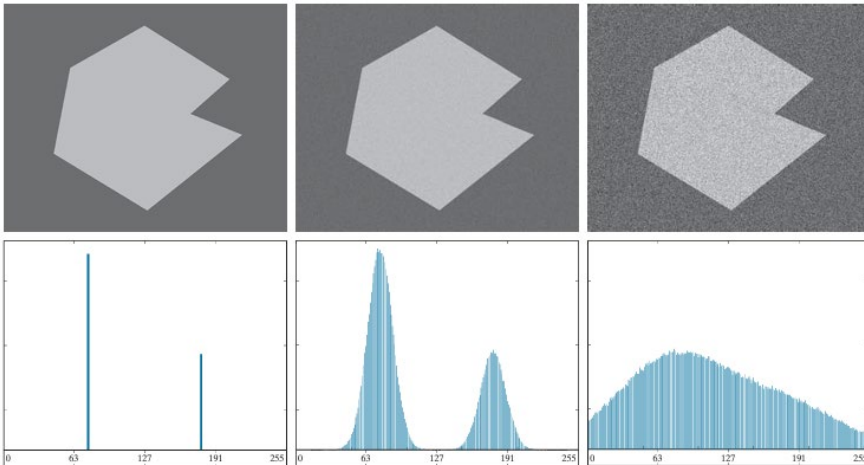
Kynnystys voidaan tehdä myös toisin päin, eli kynnystämällä voidaan helposti valita, mikä alue kuvasta halutaan näyttää arvolla 0 ja 1. Kynnystykseen voidaan valita myös useampia arvoja tarpeen mukaan (ks. kuvio 47). Usein tällainen kynnystys voi olla haastavaa tai mahdotonta, minkä takia kynnystyksessä pitää käyttää lisäksi muita menetelmiä. (Gonzales & Woods 2018, 743)



Kuvio 47. Kynnystys yhdellä ja kahdella raja-arvolla (Gonzales & Woods 2018, 743.)

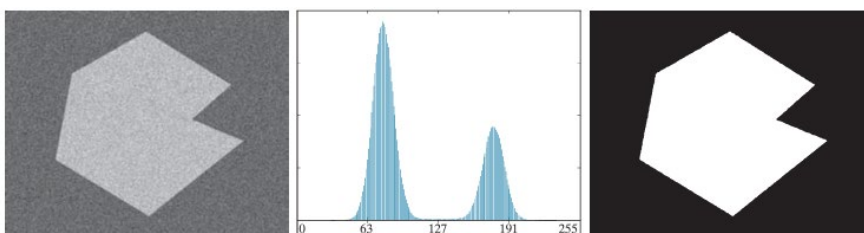
Kun tunnistettavat muodot tai kappaleet eivät ole kosketuksissa toisiinsa, pystyään ne yleensä erottelemaan kynnestyksellä nopeasti ja helposti. Tällaisessakin tapauksessa on kuitenkin valittava kynnestyksen raja-arvo siten, että kuvasta saadaan eroteltua halutut kohteet. Kuvissa saattaa kuitenkin esiintyä häiriötä, mitkä estävät yksinkertaisen kynnystystoiminnon hyödyntämisen. Kuvi-

ossa 48 on esitetty kolme harmaasävykuvaa. Ensimmäisessä kuvassa on erittäin selkeä jako kahden eri alueen välillä. Toisessa kuvassa on häiriötä, mikä näkyy histogrammiesityksessä laajempina otantana eri harmaasävyarvoja. Kuitenkin kynnystysarvo on vielä selkeästi valittavissa erottelemaan kaksi aluetta. Kolmannessa kuvassa häiriötä on jo niin paljon, ettei kynnystykselle pysty määrittämään yhtä raja-arvoa, vaikka ihminen pystyykin kuvasta helposti kertomaan, missä kohtaa muoto erottuu. (Sonka ja muut 2008, 176–183.)



Kuvio 48. Kynnystyksen hyödyntäminen harmaasävykuviin (Gonzales & Woods 2018, 744.)

Haastavissa tapauksissa voidaan hyödyntää esimerkiksi aikaisemmin esitettyä keskiarvosuodatusta sopivan kokoisella kernelillä. Kuviossa 49 on esitetty paljon häiriötä sisältävän harmaasävykuvan keskiarvotettu histogrammikuvaaja, sekä kynnystyksellä muunnettu binäärikuva. (Gonzales & Woods 2018, 761–764; Sonka ja muut 2008, 176–183.)



Kuvio 49. Keskiarvoistettu harmaasävykuva muunnettuna binäärikuvaksi (Gonzales & Woods 2018, s. 753.)

Kuvan valotuksen ollessa epätasainen, voi olla hyödyllistä käyttää liukuviin keskiarvoihin perustuvaa kynnystä. Tässä tapauksessa kuvasta ei tunnisteta vain yhtä arvoa, vaan kuvasta lasketaan alueellisesti parhaiten sopiva kynnystysarvo. Esimerkiksi liukuvaan keskiarvoon perustuvaa kynnystystä käytetään muun muassa asiakirjojen käsittelyssä, kun valaistus ei ole tasainen koko kuva-alueella. Kuviossa 50 on esitetty huonosti valotettu asiakirja koko kuva-alaan perustuvalla ja liukuvalla keskiarvolla kynnystetyt kuvat. (Gonzales & Woods 2018, 761–764; Sonka ja muut 2008, 176–183.)



Kuvio 50. Asiakirjan yksinkertainen ja keskiarvoistettu kynnystys (Gonzales & Woods 2018, 764.)

### Kuvankäsittelyn morfologiset toiminnot

Morfologia tarkoittaa muotojen ja rakenteiden tunnistamista. Kuvankäsittelyssä morfologisilla toimintoilla tarkoitetaan toimintoja, millä pystytään erottamaan kuvista ominaisuuksia, joita voidaan hyödyntää esimerkiksi reunojen tai muotojen tunnistuksessa. Morfologisten toimintojen avulla kuvasta voidaan myös suodattaa, ohentaa, karsia tai muokata ominaisuuksia. Ennen morfologisia toimintoja kuva pitää olla muunnettu binääriseen muotoon kynnystystoiminnon avulla. Yleisiä morfologisia toimintoja ovat Dilation, Erosion, Opening ja Closing. (Gonzalez & Woods 2018, 635; Szeliski 2022, 110–111.)

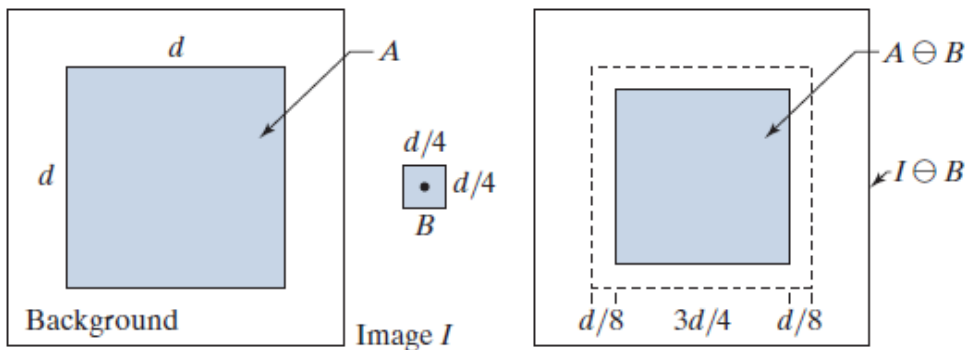
Morfologiset toiminnot perustuvat pistejoukkoihin, joilla voidaan muokata alkuperäistä kuvaa. Pistejoukot voivat olla eri muotoisia ja myös eri ulottuvuuksissa. Morfologisten toimintojen perusominaisuuksia ovat inklusio  $\subset$  tai  $\supset$ , leikkaus  $\cap$ , liitto  $\cup$ , tyhjä joukko  $\emptyset$  ja komponentti  $^c$ . Binäärikuva voidaan käsitellä kaksiulotteisena pistejoukkona, joka voidaan esittää tunnuksella  $Z^2$ . Kuvan pistejoukko voidaan esittää tunnuksella  $A$  ja elementti, millä kuva käsitellään, voidaan esittää tunnuksella  $B$ . Tässä tapauksessa Erosion-toiminto voidaan esittää yhtälöllä 18:

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\} \quad (18)$$

Missä  $z$  = Binäärisen kuvan 1 arvon pikselit

$\emptyset$  = tyhjä joukko

Erosion-toiminnossa siis käydään kuvaa läpi määritetyn kokoisella ja muotoisella pistejoukolla. Binäärisen kuvan tapauksessa pistejoukko voisi olla esimerkiksi 3 x 3 kokoinen elementti, missä jokaisen pikselin arvo olisi yksi. Elementille määritetään myös nollakohta. Nollakohta voi olla missä tahansa kohdassa elementtiä. Kuviossa 51 on käsitelty pistejoukko tietyn kokoisella elementillä, missä nollapiste on määritetty elementin keskelle. Elementtiä siirretään pikseli pikseliltä koko kuva-alueen läpi, keskittämällä elementin nollakohta käsiteltävään pikseliin. Vain siinä tapauksessa, kun jokaisella kuvan pikselillä on sama arvo, kuin elementillä, määritetään uuteen kuvaan pikseliarvo ykköseksi. Erosion-toiminto siis pienentää alkuperäistä kuvaa. Toimintoa voidaan käyttää esimerkiksi suodattamaan kuvasta pois pieniä yksityiskohtia. (Gonzalez & Woods 2018, 636; Sonka ja muut 2008, 658.)

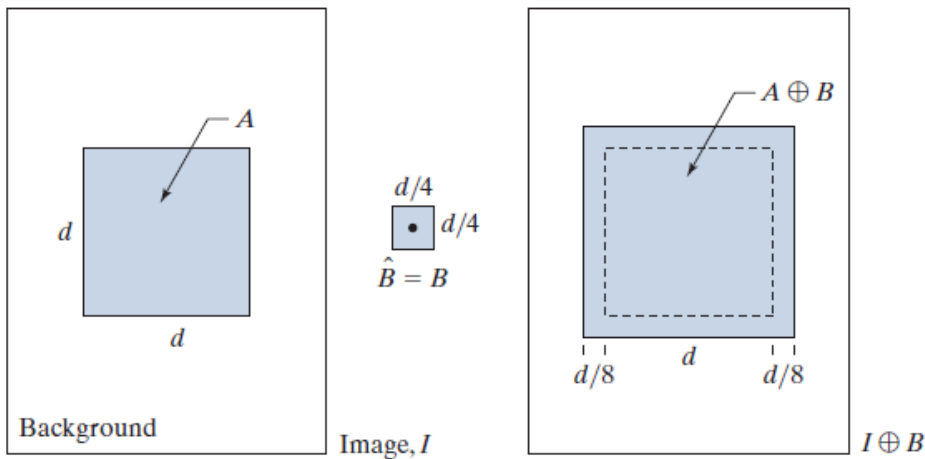


Kuvio 51. Erosion-toiminto (Gonzalez & Woods 2018, 640.)

Dilation-toiminto voidaan esittää samalla tavalla yhtälöllä 19:

$$A \oplus B = \{z | (B)_z \cap A^c \neq \emptyset\} \quad (19)$$

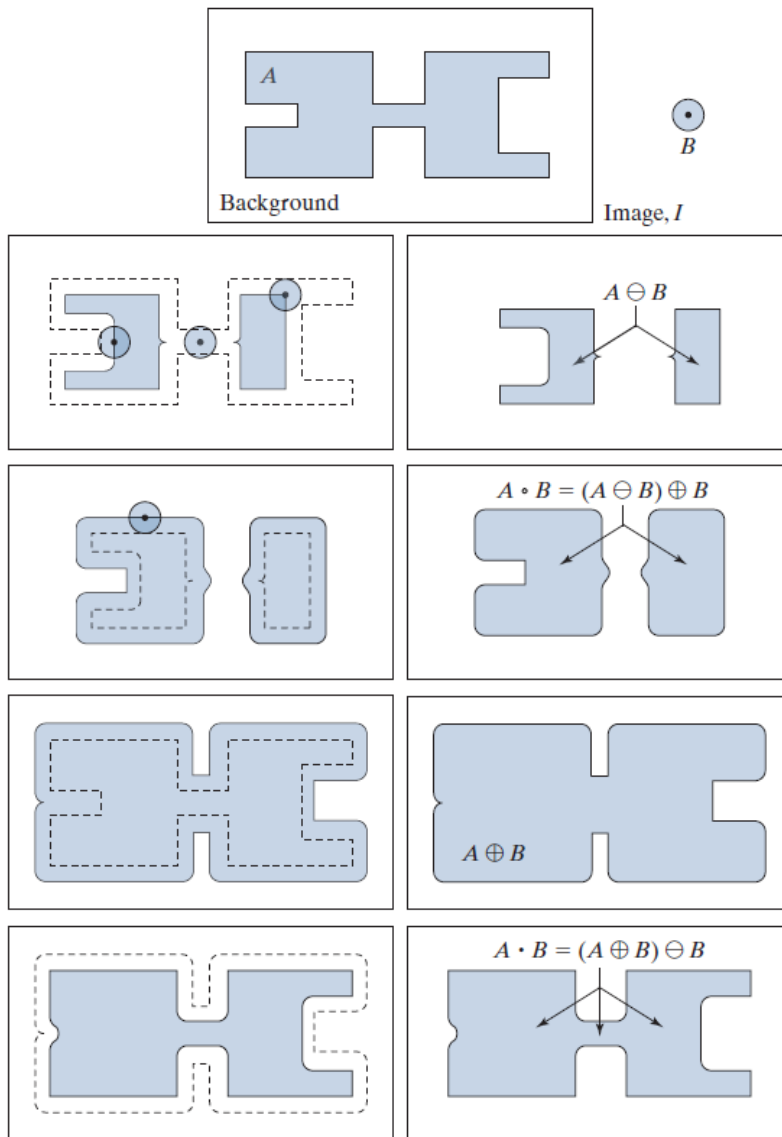
Erosion-toiminnon tapaan Dilation-elementti voi olla eri kokoinen ja muotoinen, minkä lisäksi sille täytyy määrittää nollapiste. Dilation-toiminnossa kuitenkin tarkastellaan, onko jokin elementin pikseliarvo sama, kuin käsiteltävässä kuvassa. Tällaisessa tapauksessa uuteen kuvaan muokataan elementin mukaiset arvot. Kuviossa 52 on esitetty Dilation-toiminto. Kuvasta havaitaan, että toiminto laajentaa kuvaa. Tätä voidaan käyttää hyödyksi esimerkiksi muodossa esiintyvien reikien tai katkosten täyttämiseen. (Gonzalez & Woods 2018, 641–644.)



Kuvio 52. Dilation-toiminto (Gonzalez & Woods 2018, 643.)

Opening- ja Closing-toiminnot ovat käytännössä Erosion- ja Dilation toimintojen hyödyntämistä. Openin-toiminnolla saadaan tasoitettu ääri viivoja, poistettua ohuita ulkonemia ja katkaistua yhdistymiä muotojen välillä. Toiminto toteutetaan tekemällä kuvalle ensin Erosion-toiminto, minkä jälkeen tehdään Dilation-toiminto. Näin kuvaa saadaan ensin pienennettyä ja ohuita kytköksiä katkaistua, minkä jälkeen muotoja kasvatetaan muotojen parantamiseksi Dilation-toiminnolla. Closing-toiminnossa järjestys on päinvastainen, milloin muotoa ensin suurennetaan ja sen jälkeen kutistetaan. Toimenpiteellä saadaan suljettua avoimia kohtia ja tehtyä muodon pinnasta tasaisempia. Kuviossa 53 on esitetty Openin- ja Closing-toiminnot esimerkkimuodon avulla. (Gonzalez & Woods 2018, 644–647.)

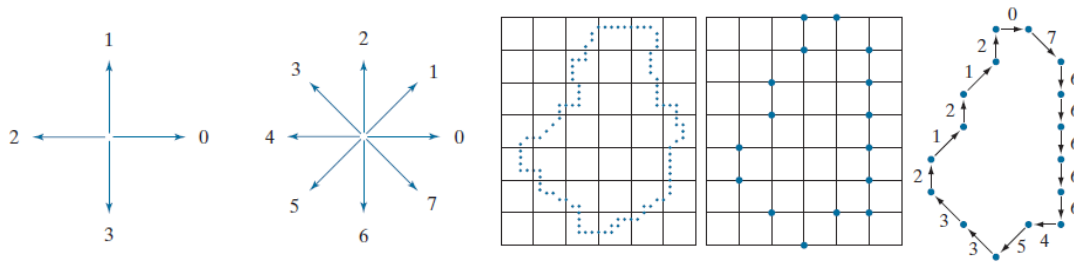




Kuvio 53. Openin- ja Closing-toiminnot (Gonzalez & Woods 2018, 647.)

### Mittaukset ja luokittelut

Kun kuvasta on saatu eroteltu tarvittavat muodot ja rakenteet, tulokset yleensä muutetaan paremmin esitettävään muotoon tarvittavien mittausten tai luokitteluiden takia. Muodon ääriviivojen perusteella pystytään tekemään erilaisia mittauksia ja määrittelyjä. Ääriviivojen tunnistuksessa käytetään neljän tai kahdeksan pisteen tunnistusta (ks. kuvio 54). Neljän pisteen tunnistuksessa tarkastellaan pisteiden liityntää vain vaaka- ja pystytasossa. Kahdeksan pisteen tarkastelussa otetaan huomioon myös viistot liitynnät. Jos pisteestä löytyy liityntä seuraavaan soluun, mikä ei ole tyhjä, voidaan reunan tunnistus tehdä ja toistaa toimenpide. (Gonzalez & Woods 2018, 811–817.)



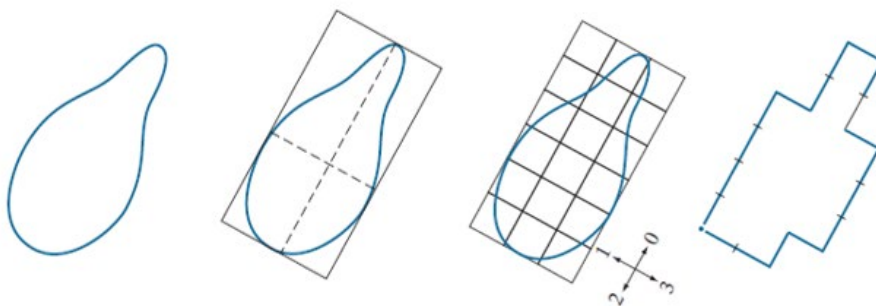
Kuvio 54. Reunapisteiden tunnistussuunnat ja esimerkkikuva (Gonzalez & Woods 2018, 817.)

Reunan pituuden likiarvo saadaan laskettua helposti reunan pikselimäärän perusteella. Vaaka- ja pystytasossa arvo on suoraan pikselin arvo, mutta vinottain liikkuessa pituus pitää kertoa arvolla  $\sqrt{2}$ . Muodon halkaisija saadaan määritettyä yhtälöllä (20):

$$\text{diameter}(B) = \max_{i,j} [D(p_i, p_j)] \quad (20)$$

Missä  $B$  = muodon halkaisija  
 $D$  = muodon ääriviivan pituus  
 $p_i, p_j$  = ääriviivan pisteet

Pisteiden avulla pystytään määrittämään myös bounding box, mikä tarkoittaa pienintä laatikkoa, mikä muodon ympärille pystytään sovittamaan. Muotoa ja mittoja pystytään käyttämään hyödyksi esimerkiksi suunnan ja uuden muodon määrittelyssä (ks. kuvio 55). (Gonzalez & Woods 2018, 832.)



Kuvio 55. Bounding box ja muodon orientaatio (Gonzalez & Woods 2018, 835.)

Kuvasta voidaan myös erotella muotoja niille tyypillisten piirteiden mukaan. Tunnistetusta muodosta voidaan esimerkiksi laskea kompaktius, mikä saadaan laskettua yhtälöllä 21:





$$\text{Kompaktius} = \frac{p^2}{A} \quad (21)$$

Missä:  $p$  = reunaviivan pituus  
 $A$  = pinta-ala

Kompaktiuden lisäksi voidaan tarkastella kappaleen ympyrämuotoisuutta, mikä saadaan laskettua yhtälöllä 22:

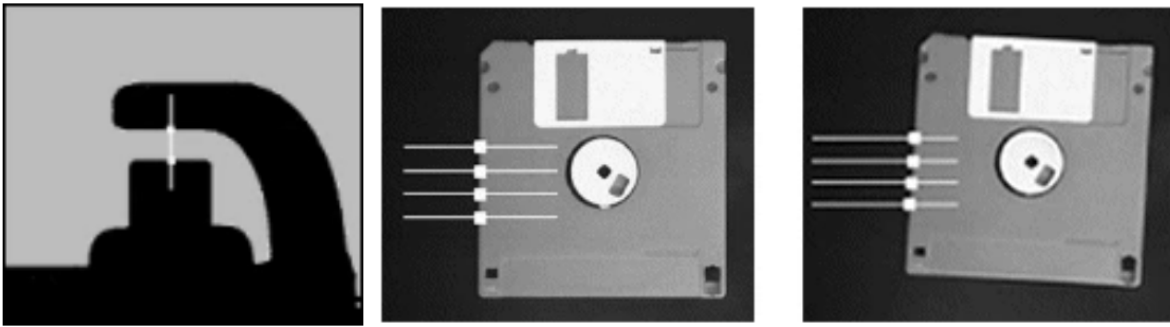
$$\text{Ympyrämuotoisuus} = \frac{4\pi A}{p^2} \quad (22)$$

Kuviossa 56 on esitetty muutama erilainen muoto ja niiden piirteiden ominaisuuksia. (Gonzalez & Woods 2018, 840–842)

Descriptor				
<i>Compactness</i>	10.1701	42.2442	15.9836	13.2308
<i>Circularity</i>	1.2356	0.2975	0.7862	0.9478

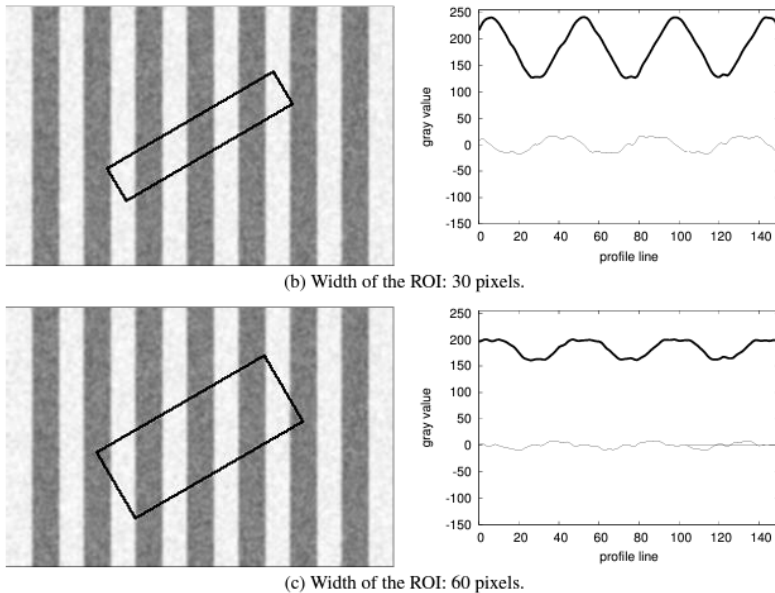
Kuvio 56. Muotoja ja niiden ominaisuuksia (Gonzalez & Woods 2018, 842.)

Muotojen ja reunojen tarkastelun lisäksi erilaiset mittaukset ovat yleisiä. Tuotteesta halutaan esimerkiksi ottaa mitta tietyistä kohtaa laadun varmistamiseksi tai tuotteen tunnistamiseksi. Mittauksessa voidaan hyödyntää reunojen intensiteetin muutoksen tunnistamista. Mittaustoiminto määritetään haluttuun kohtaan kappaletta ja toimenpide tunnistaa reunojen muutoksen. Yleensä mittaukseen voi määrittää yhden tai useamman reunantunnistuksen mitkä tapahtuvat valitulla alueella. Reunan tunnistuspisteiden perusteella voidaan myös määrittää esimerkiksi tuotteen asento. Kuviossa 57 on esitetty sytytystulpan kärkivälin mittaaminen yhden linjan avulla, sekä kappaleen asento reunaviivan pisteiden perusteella. (Image Processing with NI Vision Development Module 2024)



Kuvio 57. Mittaustoiminnon esimerkkikuva (Image Processing with NI Vision Development Module 2024.)

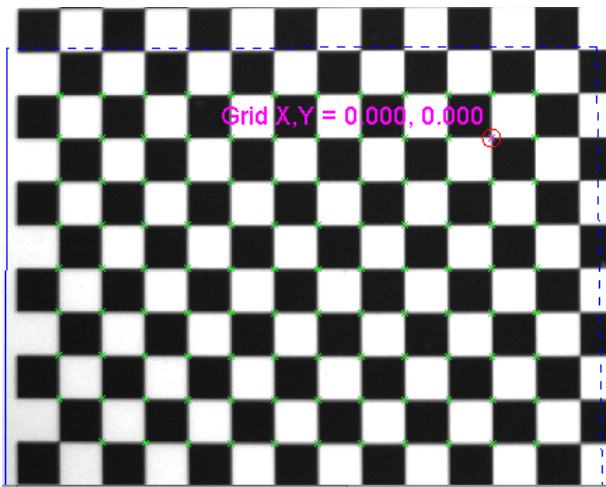
Mittaustoiminto määritetään yleensä useamman pikselin levyiseksi, ettei pienet häiriöt kuvassa aiheuta vääristymää värisävyn tunnistamiseen. Mittaustoiminnon oikean asennon määrittäminen on kuitenkin tärkeä tehdä huolella, koska mittauksen keskiarvoistustoiminto saattaa aiheuttaa muutokseen ei-toivotun ominaisuuden ja muuttaa kuvaajan muutoksen liian tasaiseksi (ks. kuvio 58). (Solution Guide III-A 1DMeasuring 2024, 37.)



Kuvio 58. Mittausalueen suunnan vaikutus tunnistettavaan harmaasävyarvoon (Solution Guide III-A 1DMeasuring 2024, 40.)

### Kuvan kalibrointi

Kuten aiemmin on esitetty, digitaalinen kuva muodostuu pikseleistä, joiden lukumäärä määräytyy kameran kennon ominaisuuksista. Kameran kuvaaman kuva-alueen koko taas määräytyy kameran ominaisuuksista ja käytettävästä optiikasta. Jos kuvasta halutaan tehdä mittauksia, on millimetrien käyttäminen huomattavasti helpompaa ja informatiivisempaa, kuin pikseliarvojen käyttäminen. Kalibroinnin avulla saadaan kuva-alueen tieto muutettu millimetreiksi, minkä lisäksi kalibroinnilla pystytään poistamaan optiikasta aiheutuvaa vääristymää kuva-alueen reunoilla. Kalibrointi voidaan tehdä lineaarisesti määrittämällä, kuinka montaa millimetriä yksi pikseli vastaa. Lineaarissa kalibroinnissa oletetaan, että kuvattava kohde on kohtisuorassa kameran kanssa. Lineaarinen kalibrointi antaa koko kuva-alueella saman arvon, minkä takia kalibrointi ei ole tarkka koko kuva-alueella. Lineaarinen kalibrointi ei myöskään poista optiikasta aiheutuvaa vääristymää. Toinen kalibrointimenetelmä hyödyntää kuva-alueelle asetettua shakkilauta-, tai pistekuviota. Menetelmää kutsutaan grid-kalibroinniksi (ks. kuvio 59). Kuvasta tunnistetaan valkoisten ja mustien neliöiden kulmat tai pistekuvion pisteet. Kun kalibrointikuvion mitat tiedetään, pystytään kuva-alueen kalibrointi määrittämään paremmin koko kuvalle. (Azad ja muut 2008, 72–73. Zhang 2023, 35–37.)



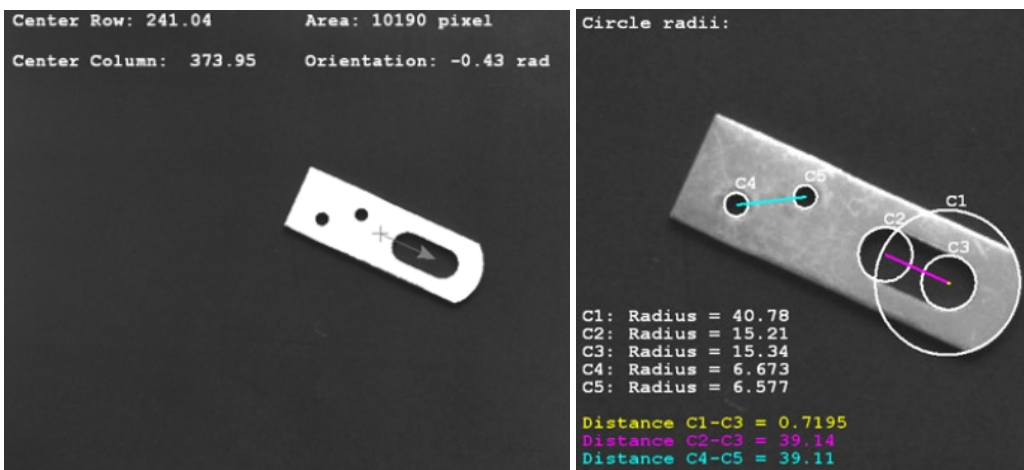
Kuvio 59. Grid-kalibrointi Cognex In-Sight Explorer ohjelmassa (Cognex CalibrateGrid.)

### Analyysi ja lopputulosten esittely

Varsinainen konenäkösovellus koostuu erilaisista suodatuksista ja toiminnoista, millä halutut asiat saadaan erotettua kuvasta. Lopullinen sovellus, sisältää se sitten pituuden, halkaisijan tai pinta-alan mittausta, sävyjen tai muotojen tunnistusta tai vaikka opetetun mallin etsimistä, on tulosten esittäminen ja tarvittaessa lähettäminen toiselle laitteelle erittäin tärkeä osa. Ohjelman toiminta

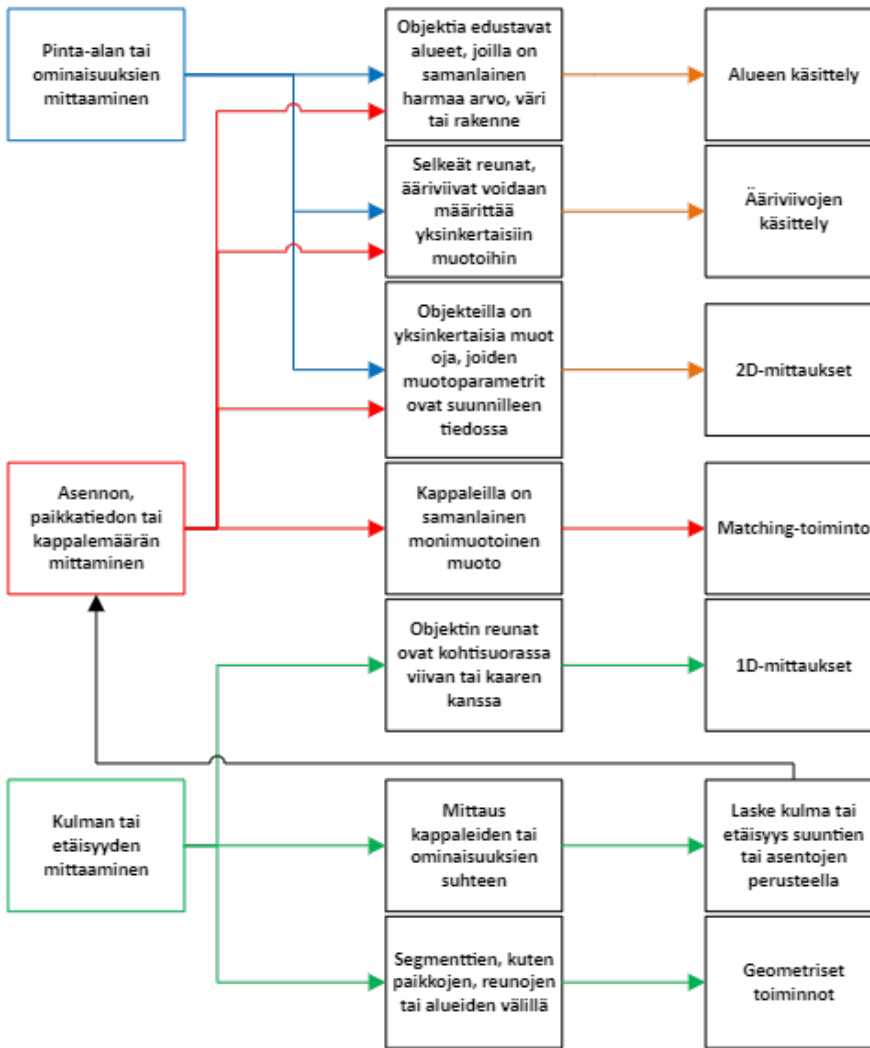
perustuu opetettuihin ja testattuihin raja-arvoihin. Esimerkiksi tuotteen pinta-alalle on voitu asettaa minimi- ja maksimiarvot. Jos tuotteen koko ei täytä tätä ehtoa, poistetaan tuote linjalta. (Silvola & Gautam 2023, 182. Pietikäinen & Silven 2023, 95.)

Reunan harmaasävyarvon tunnistamiseen liittyvän mittaustoiminnon lisäksi perinteisistä konenäköohjelmistoista löytyy yleisesti erilaisia 2D-mittauksia. Tällaisia mittauksia ovat esimerkiksi pinta-alan, paikkatiedon ja kappaleen asennon määrittäminen, kulmamittaukset kappaleen tai eri ominaisuuksien välillä sekä erilaiset mittaukset tunnistettujen ominaisuuksien välillä. Toisin kuin pelkän viivan perustella tehty mittaus, 2D-mittauksissa myös paikkatieto tulee mittauksen mukana. Yleisesti ottaen toiminnot pohjautuvat muotojen ja ominaisuuksien tunnistamiseen. Reunojen ja esimerkiksi ympyrämäisten muotojen tunnistamisen avulla pystytään tekemään erilaisia mittauksia (ks. kuvio 60).



Kuvio 60. Tuotteen ominaisuuksien ja mittojen selvittäminen perinteisillä konenäkötoiminnoilla (Solution Guide III-B 2D Measuring 2024, 8, 10.)

Kuviossa 61 on esitetty valintaohje perinteisten konenäkötoimintojen valitsemiseksi tietyn tyyppiin toimintoihin.

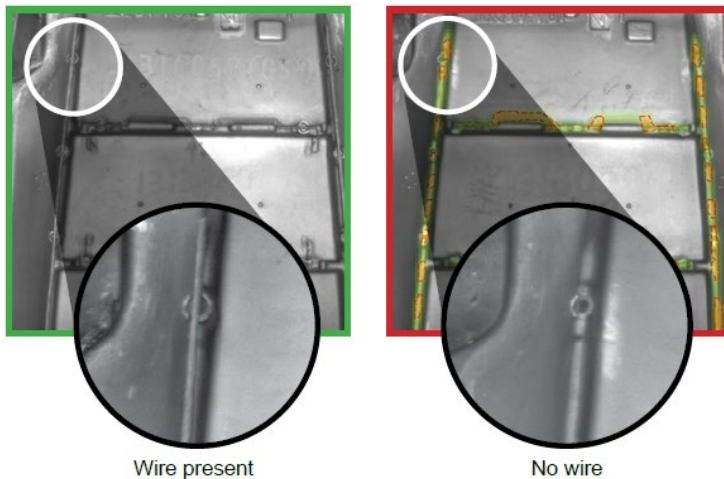


Kuvio 61. Konenäkötoimintojen valintataulukko (Solution Guide III-B 2D Measuring 2024, 28. Muokattu.)

#### 4.2.1 Konenäköjärjestelmien haasteet

Konenäköjärjestelmien toiminnan takia on erittäin tärkeää, että kuvien laatu pysyy tasaisena. Järjestelmät toimivat hyvin tasalaatuisille ja hyvin valmisteluille osille. Laitteistosta, ympäristöstä tai tuotteista johtuvat poikkeamat kuvissa aiheuttavat ongelmia ehtopohjaisille toiminnoille. Jos kuvan kynnystys on tehty tietyllä arvolla ja valaistus tai tuote vaihtuu, ei järjestelmä enää pysty tekemään tarvittavia toimenpiteitä tuotteen tunnistamiseksi tai mittaamiseksi. Haastavuutta voi aiheuttaa esimerkiksi tuotteen lopputarkastus, kun virheiden varianssi voi olla laaja, eikä ehtopohjaisesti pystytä määrittämään kaikkia vikatilanteita. Muita haastavia kohteita ovat muun muassa heijastavat, kiiltävät ja monimuotoiset kappaleet. Konenäköjärjestelmissä on toimintoja,

kuten mittakaavan, tuotteen pyörimisen tai vääristymän korjaus sekä erilaiset suodatusmenetelmät, millä joustavuutta tuotteen tarkastukseen pystytään tekemään. Tästä huolimatta varsinkin monimutkaiset pintatekstuurit tai tuotteet, missä on erilaisia mutta vähäisiä poikkeamia, ovat erittäin haastavia perinteisille konenäkötoiminnoille (ks. kuvio 62). (Cognex 2022.)



Kuvio 62. Perinteisen konenäköjärjestelmän haasteet (Cognex 2022.)

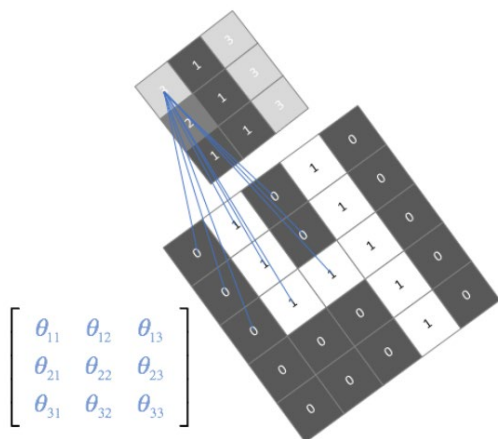
Koneoppimisessa puhutaan paljon opetus- ja testiaineiston laadusta ja määrästä. Myös perinteisten sääntöpohjaisten konenäköfunktioiden tekeminen vaatii riittävän kuva-aineiston. Perinteisten toimintojen hyödyntämisessä on tärkeää huomata, että jokaisesta tunnistettavasta muodosta, mitattavasta tuotteesta tai pintavirhetyypistä täytyy saada kuva-aineistoa ohjelman tekemistä varten. Myös useampi kuva erilaisista tuotteista auttaa ohjelman tekemistä, kun toiminnan pystyy varmentamaan ja testaamaan kuvia muuttamalla. Mitä enemmän ympäristössä tai tuotteessa voi tapahtua vaihtelua, sitä enemmän kuva-aineistoa ja erilaisia variaatioita kuvista tarvitaan toimivan ohjelman tekemiseksi.

## 5 Koneoppiminen kuvien tunnistuksessa

Konenäössä hyödynnettävät neuroverkot hyödyntävät yleensä myös konvoluutiotoimintoja. Tällaisia neuroverkkoja kutsutaan konvoluutioneuroverkoiksi ja termi CNN tulee englannin kielen termeistä convolutional neural network. Haasteena perinteiselle neuroverkolle kuvankäsittelyssä on piirteiden esiintyminen eripuolilla kuvaa ja eri asennoissa. Jos kohde on aina samassa asennossa,



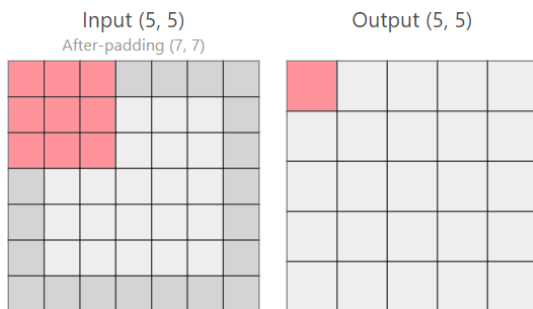
voitaisiin hyödyntää myös perinteistä neuroverkkoa. Konvoluutioiden avulla pystytään tunnistamaan kuvasta muotoja ja hahmoja, eikä niiden tarvitse olla aina samassa paikassa tai asennossa (ks. kuvio 63). Konvoluutioiden avulla pystytään myös pienentämään riittävään tunnistamiseen tarvittavaa parametrimäärää. Neuroverkkoa voidaan hyödyntää tunnistamaan eritasoisia piirteitä kuvista konvoluutiotointojen jälkeen. Alatason tunnistukset voivat olla esimerkiksi reunoja ja viivoja ja ylemmän tason toiminnot monimutkaisempia ja yksityiskohtaisempia ominaisuuksia kuvassa olevasta kohteesta. (Pietikäinen & Silven 2023, 49.)



Kuvio 63. Vinossa olevan muodon tunnistus konvoluutiotointoinnolla (Jordan 2017a)

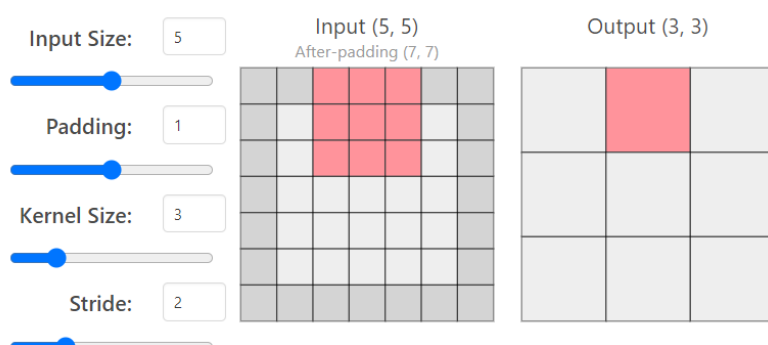
Konvoluutioneroverkko rakentuu siis erilaisista konvoluutioista, joita voivat olla esimerkiksi suodatukset, kuvan terävöinti tai reunojen tunnistus. Syötteeksi annettu kuva käsitellään yhdellä tai useammalla konvoluutiolla. Jokaisen konvoluution lisätään bias-arvo ja tulos ajetaan läpi aktivointifunktiosta. Tällä tavalla käydään läpi koko kuva ja konvoluution jälkeen muodostetaan uusi pistejoukko, mikä toimii syötteenä seuraavalla kerrokselle. Jos eri konvoluutiota on esimerkiksi neljä kappaletta, muodostuu kuvasta neljä uutta piirrekuvaa. Englanninkielinen termi piirrekuvalle on Feature map. Piirrekuvan koko määräytyy konvoluution määrittämisestä. Aikaisemmin esitetyn mukaan kuva voidaan käydä kernelillä läpi pikseli pikseliltä, jolloin piirrekuva on samankokoinen, kuin alkuperäinen kuva. Joissain tapauksissa kerneliä ei kuitenkaan voida käyttää kuvan reunalla kuvaylityksen takia. Tällaisissa tapauksissa kuva pienenee kernelin reunan verran. Jos kuvakoko halutaan säilyttää, voidaan käyttää täyttötoimintoa, mikä tarkoittaa ylimääräisen reunan lisäämistä syötteeksi tulevaan kuvaan. Reuna voidaan laajentaa eri tekniikoilla, mutta yleisimmin käytettävä tekniikka on lisätä nollia symmetrisesti reunojen

ympärille kernelin reunan verran. Esimerkiksi 3 x 3 kernelin kanssa lisätään yksi rivi nolliä syötteen jokaiselle sivulle (ks. kuvio 64). Näin ollen syötteen matriisin koko kasvaa kahdella, mutta lähtö pysyy alkuperäisen kuvan kokoisena. (Wang, Turko, Shaikh, Park, Das, Hohman, Kahng & Chau; Gonzalez & Woods 2018, 965.)



Kuvio 64. Esimerkki täyttö-toiminnosta 3 x 3 kernelillä (Wang ja muut n.d.)

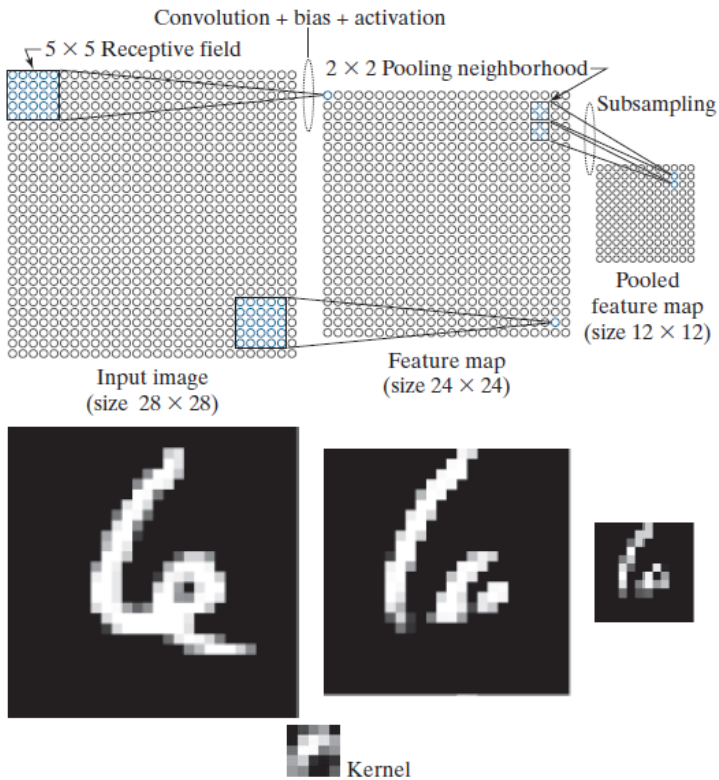
Konvoluutiota ei aina kannata tehdä pikseli pikseliltä, vaan kerneliä voidaan siirtää myös useampi pikseli. Tällä tavalla lähdön koko muuttuu, mutta laskennasta tulee helpompaa ja nopeampaa. Askel kuvastaa, kuinka paljon kerneliä siirretään toimenpiteiden välissä. Esimerkiksi syötteen ollessa 5 x 5, kernelin 3 x 3 ja askeleen ollessa kaksi, tulee lähdöksi 3 x 3 kokoinen matriisi (ks. kuvio 65). (Wang ja muut n.d)



Kuvio 65. Kernelin askelmäärityksen vaikutus kuvakokoon (Wang ja muut n.d)

Konvoluutiotoimintojen välissä käytetään yleensä Pooling-toimintoa, eli kerrostamista. Pooling-toiminnossa konvoluution tulosta yksinkertaistetaan yleensä maksimi- tai keskiarvotoiminnolla. Toiminto voi olla esimerkiksi 2 x 2 kokoinen matriisi, millä käydään kuva läpi ja valitaan neljän

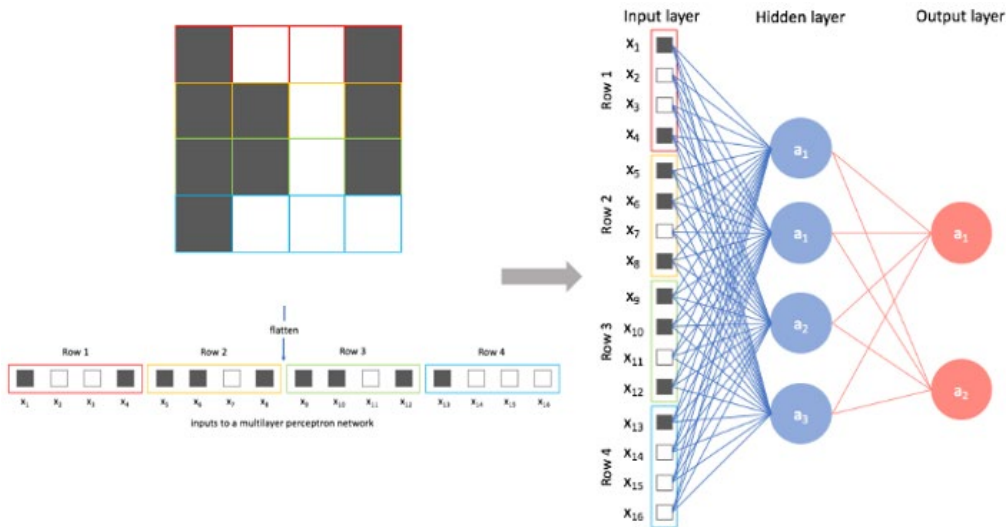
pikselin maksimi- tai keskiarvo. Tällä tavalla kuvakokoa saadaan pienennettyä, mutta kuvan tärkeät ominaisuudet saadaan säilytettyä. Kuviossa 66 on esitetty konvoluution ja Pooling-toiminnon esimerkki piirretystä numerosta. (Gonzalez & Woods 2018, 965–971; Wang ja muut n.d.)



Kuvio 66. Konvoluution ja Pooling-toiminnon tulos esimerkkikuvasta (Gonzalez & Woods 2018, 969.)

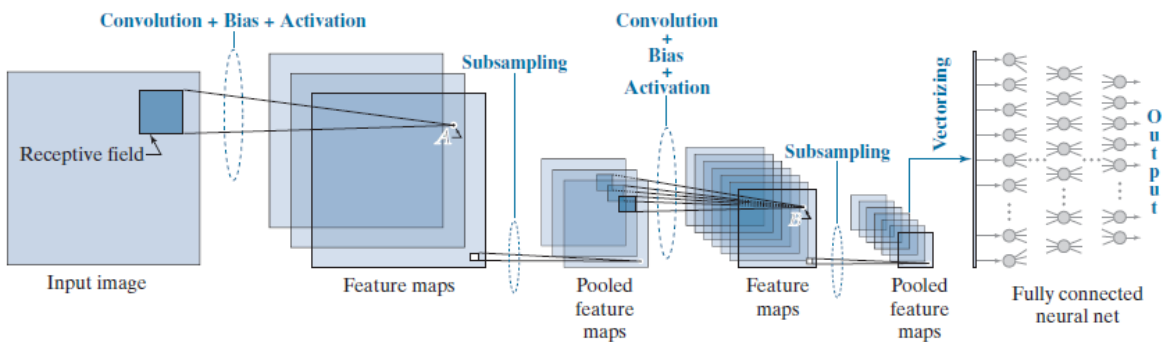
Pooling-toiminnon jälkeinen kuva voidaan edelleen käsitellä halutuilla konvoluutioilla ja toistaa toimenpiteet, kunnes piirteet on saatu jaoteltua riittävälle tasolle. Konvoluutioiden jälkeen viimeinen sarja piirrekuvia täytyy muuttaa neuroverkolle soveltuvaan muotoon. Neuroverkkojen toiminta on suunniteltu toimimaan yksiulotteisen aineiston kanssa, minkä takia kaksiulotteinen taulukko muokataan yhdeksi pitkäksi vektoriksi. Toimenpiteestä käytetään englanninkielistä termiä Flatten. Vektorimuutoksen jälkeen jokainen piirre annetaan syötteenä neuroverkolle. (Gonzalez & Woods 2018, 965–971; Wang ja muut n.d.)

Kuviossa 67 on havainnollistettu Flatten-toimenpidettä. Jos käsiteltävä kuva olisi 4 x 4 kokoinen binäärikuva numerosta yksi, menisi neuroverkon syötteeksi yksi 24 pikseliä pitkä jono. (Jordan 2017a.)



Kuvio 67. Flatten-toiminnon esimerkkikuva (Jordan 2017a.)

Konvoluutioneuroverkon rakenne voisi kokonaisuudessaan olla alla olevan kuvion 68 mukainen.

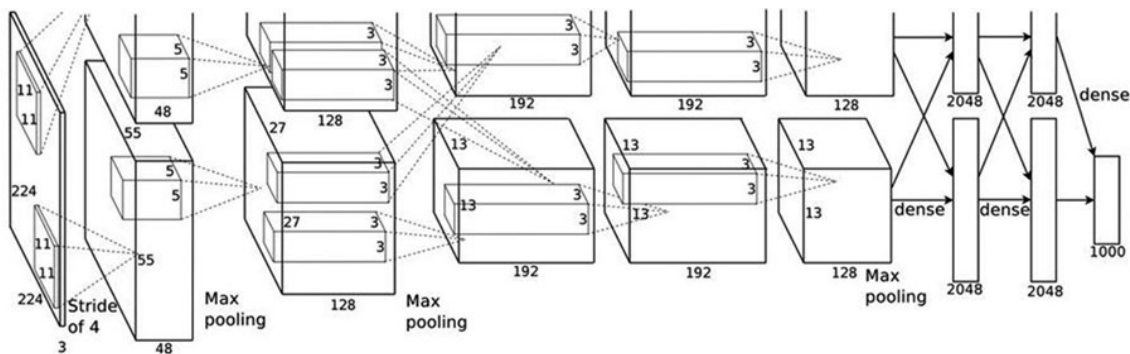


Kuvio 68. Konvoluutioneuroverkon esimerkkirakenne (Gonzalez & Woods 2018, 965.)

Verkon suorituskykyä voidaan parantaa lisäämällä verkkoon useampia kerroksia. Kerrosten lisääminen on tullut mahdolliseksi entistä tehokkaampien tietokoneiden ansiosta, mutta raja kerrosten määrässä on jo havaittavissa. VGGNetin rakenteessa on 16 kerrosta ja GoogLeNetissä 22 kerrosta. Näillä päästiin huomattavasti parempiin luokittelutuloksiin, kuin AlexNetin kahdeksankerroksisella



Aina mallin rakentaminen alusta ei ole kannattavaa. Tutkijat ja tiedemiehet ympäri maailmaa ovat tutkineet ja kehittäneet malleja vuosikausia, minkä tuloksena on tuotettu valmiita neuroverkkorakenteita. Suosittuja neuroverkkomalleja ovat esimerkiksi LesNet-5, AlexNet, VGGNet, GoogLeNet, ResNet, R-CNN, YOLO, SqueezeNet, SegNet, GAN ja moni muu. Esimerkiksi kuviossa 70 esitetyn AlexNetin rakenne koostuu viidestä konvoluutiosta, kolmesta max pooling-toiminnosta sekä kolmesta neuroverkosta. Mallilla on yhteensä 60 miljoonaa parametria ja 650 000 neuronia. (Verdhan 2021, kappale 4.)



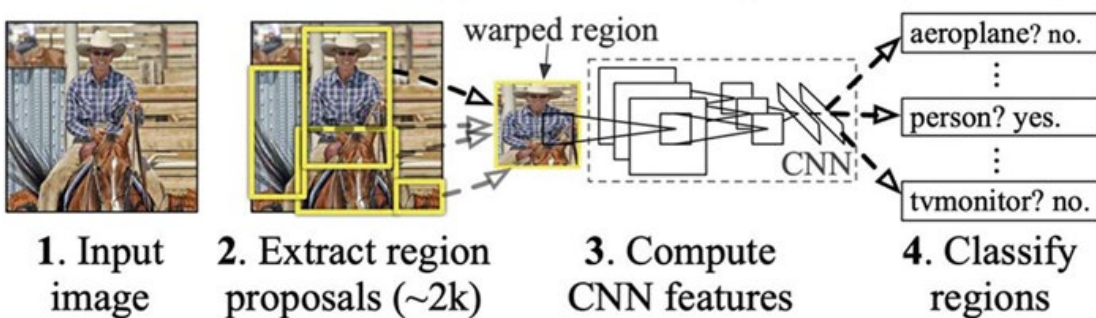
Layer	Operation	Feature Map	Input Size	Kernel Size	Stride	Activation function
Input	Image	1	227x227x3			
1	Convolution	96	55x55x96	11x11	4	ReLU
	Max Pooling	96	27x27x96	3x3	2	ReLU
1	Convolution	256	27x27x256	5x5	1	ReLU
	Max Pooling	256	13x13x256	3x3	2	ReLU
3	Convolution	384	13x13x384	3x3	1	ReLU
4	Convolution	384	13x13x384	3x3	1	ReLU
5	Convolution	256	13x13x256	3x3	1	ReLU
	Max Pooling	256	6x6x256	3x3	2	ReLU
6	Fully Connected		9216			ReLU
7	Fully Connected		4096			ReLU
8	Fully Connected		4096			ReLU
	Output		1000			Softmax

Kuvio 70. AlexNetin rakenne (Verdhan 2021, kappale 4.)

Valmiita konvoluutioneuroverkoilla opetettuja malleja voidaan myös hyödyntää aikaisemmin esitetyn Transfer learning -toiminnon tapaan. Esimerkiksi ImageNetin 1000 eri luokkaa sisältävällä kuva-aineistolla opetettua isoa konvoluutioneuroverkkoa voidaan hyödyntää siirto-opettamalla tunnistamaan omaan sovellukseen tarpeelliset piirteet. Siirto-opetuksessa verkon ylin kerros poistetaan ja se opetetaan halutulla aineistolla. Alempien kerrosten, eli esimerkiksi konvoluutioverkkojen viritys lukitaan, eikä niihin tehdä muutoksia. Tällä tavalla uuden mallin opettaminen tapahtuu huomattavasti nopeammin. Toinen valmiiksi opetettujen mallien hyödyntämistapa on Fine-tuning, eli hienosäätö. Hienosäädön toimintaperiaate on samankaltainen, kuin siirto-opettamisessa. Hienosäädössä alempia kerroksia ei kuitenkaan lukita, vaan myös niiden painokertoimia voidaan

muuttaa. Mallia voidaan hyödyntää esimerkiksi, kun opetettavaa ainestoa on vähän, mutta tiedetään, että otanta on riittävä yleistämään kuvissa esiintyvät piirteet. Tällöin malli voidaan ensin opettaa tietyillä parametreilla ja Learning-rate-arvolla, minkä jälkeen arvoa pienennetään ja saadaan näin ollen tarkempi tulos. Fine-tuning-toimintoa voidaan käyttää myös valmiiksi opettujen, raskaidenkin mallien kanssa. Tässä tapauksessa tiedetään, että alemman tason piirteidentunnistus on jo opetettu hyväksi ja tarkaksi, joten niitä tarvitsee vain hienosäätää hyvin pienillä arvoilla. (Kneusel 2021, kappale 14; Pietikäinen & Silven 2023, 52.)

Koneoppimisen hyödyntämisessä kuvien kanssa saattaa olla ongelmana kohteen paikan ja etäisyyden muuttuminen kameraan nähden. Tällaisessa tapauksessa syöte voi olla vaikea määrittää konvoluutiolle ja neuroverkolle. Toisin sanoen tunnistustarkkuus heikkenee, jos opetetussa aineistossa ei ole erikokoisia näytteitä tunnistettavasta kohteesta. R-CNN-malli, eli Region-based CNN, on kehitetty nimen mukaisesti alueellista tunnistusta varten. R-CNN:n toiminta perustuu määritettyihin alueisiin, mitä kuvassa voi olla jopa 2000. Kuvalle tehdään ensin segmentointi, minkä avulla kuvasta tunnistetaan alueita, missä voisi olla tarkasteltavia kohteita. Tämän jälkeen jokainen alue muokataan mallille sopivaksi ja tehdään CNN:n mallin toimenpiteet (ks. kuvio 71). Mallin rakenteen takia käsittelyajat voivat olla erittäin korkeita, jopa kymmeniä sekunteja, koska käsiteltäviä kuvia voi olla jopa 2000. (Verdhan 2021, kappale 5.)

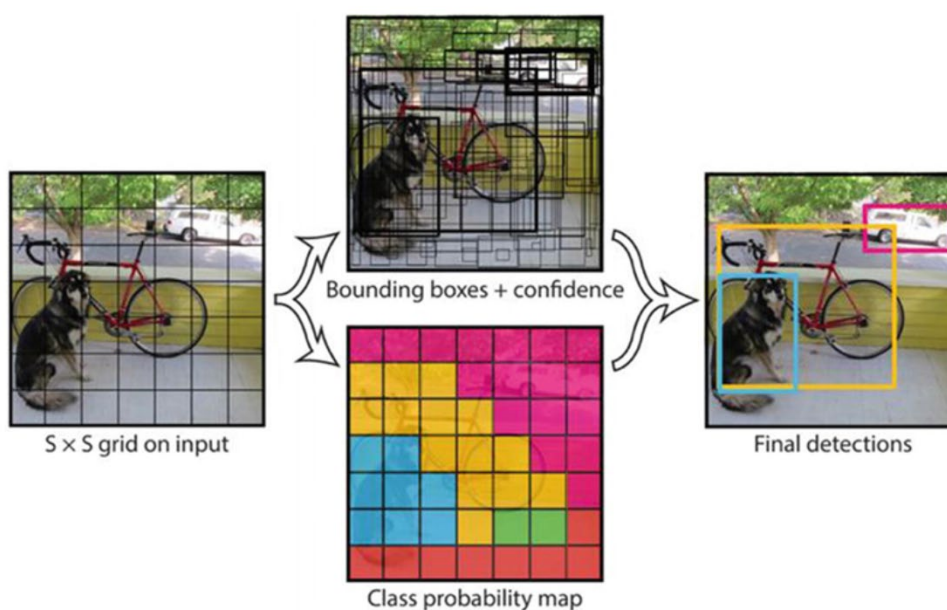


Kuvio 71. R-CNN-mallin toimintaesimerkki (Verdhan 2021, kappale 5.)

Toinen menetelmä kohteiden tunnistamiseen isosta kuvasta on YOLO, eli You Only Look Once -toiminto. YOLO:n toimintaperiaate perustuu kuvan päälle muodostetusta ruudukosta, minkä jokaiseen soluun tehdään tunnistus mahdollisesta hahmosta. YOLO käy läpi kaikki solut ja tekee ennusteen, onko solussa kappaletta. Ne solut, mihin tulee riittävän korkea ennuste, määritetään



tunnistettaviksi kohteiksi ja muodon ympärille muodostetaan laatikko. YOLO pystyy tunnistamaan kuvasta myös useampia kohteita, vaikka ne menisivät päällekkäin toistensa kanssa. Jokaisessa laatikossa siirtyy myös kuvan koko ja paikkatieto, minkä perusteella mallin arvaus pystytään sovittamaan oikeaan kohtaan kuvaa (ks. kuvio 72). Toisin kuin R-CNN:n kanssa, YOLO ajaa konvoluutioneuroverkon vain yhden kerran. Tämä mahdollistaa erittäin nopean analysoinnin ja esimerkiksi videokuvan reaaliaikaisen käsittelyn. YOLO:sta on julkaistu useampia versioita ja konvoluutioneuroverkon malli saattaa vaihdella eri versioiden välillä. (Verdhan 2021, kappale 5.)



Kuvio 72. YOLO toiminnolla tunstettuja kohteita (Verdhan 2021, kappale 5.)

## 5.1 Yleiset periaatteet koneoppimismallien hyödyntämiseen

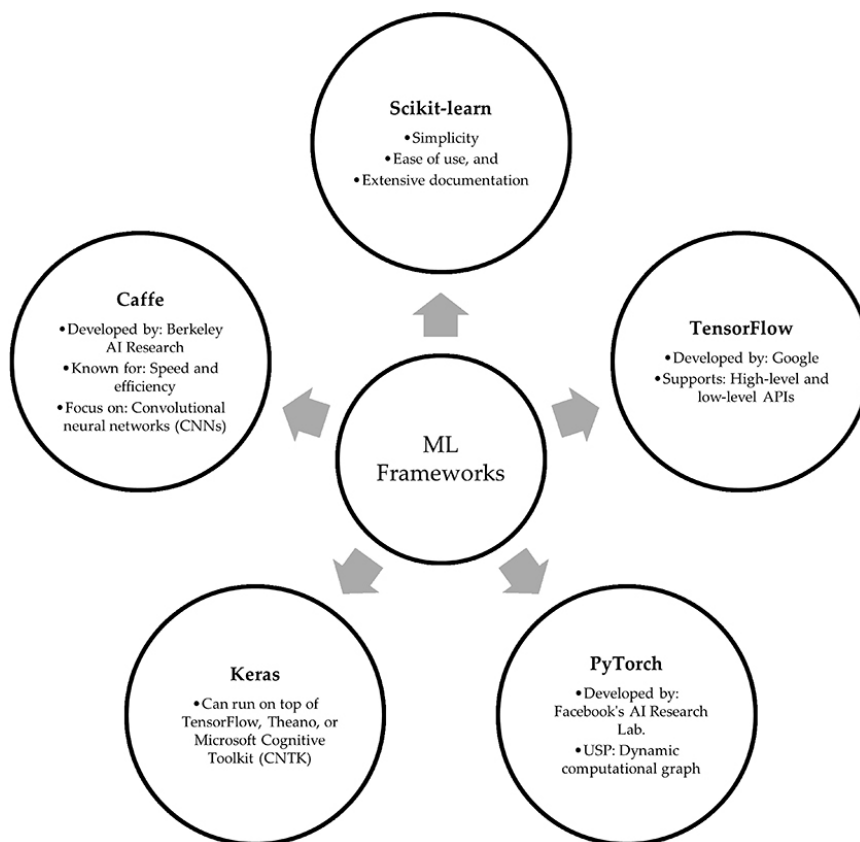
Koneoppimismallien opettamisessa voidaan hyödyntää erilaisia laiteratkaisuja. Yksinkertaisemmat ja pienemmät mallit pystytään opettamaan tietokoneiden prosessoreilla. Tietokoneen prosessoreissa on useita ytimiä, missä laskenta toimii peräkkäisessä järjestyksessä. Tämän kaltainen toiminta ei kuitenkaan sovellu hyvin koneoppimismallien opettamiseen, missä tulee suuria määriä laskutoimituksia, minkä lisäksi aineiston koko saattaa olla erittäin iso. Tietokoneiden näytönohjaimet on suunniteltu rinnakkaisten laskutoimitusten ja suurten aineistojen käsittelyyn, minkä takia niiden hyödyntäminen koneoppimismallien opettamisessa on suosittua. Perinteisten tietokoneiden lisäksi voidaan hyödyntää erillisiä koneoppimistehtäviin kehiteltyjä ASIC-laitekokonaisuuksia. Termin lyhenne tulee englannin kielen sanoista Application specific Integrated Circuits. Laitteiston



rakenne voidaan suunnitella ja optimoida tietyn tyyppiseen toimenpiteeseen, milloin järjestelmästä saadaan tehtyä mahdollisimman tehokas. (Thakare & Umesh 2023, kappale 1.)

Aina ei kannata lähteä hankkimaan tehokkaita komponentteja mallien opettamista varten, jos ai-neiston koko ja mallin tehontarve on liian iso perinteisen tietokoneen käsiteltäväksi. Tänä päivänä voidaan hyödyntää enenemissä määrin pilvipalveluita mallien opettamisessa. Pilvipalveluita tarjoavia yrityksiä on useita ja niistä pystyy vuokraamaan laskentakapasiteettia mallin opettamista var-ten. Pilvipalveluiden laskentakapasiteettia voi vuokrata esimerkiksi Googlelta, Amazonilta ja Mic-rosoftilta. (Pietikäinen & Silven 2023, 53.)

Koneoppimismalleja pystytään opettamaan monessa eri työkaluympäristössä. Ympäristöllä tarkoitetaan työkalua tai kirjastoa, mistä tarvittavat toiminnot mallin opettamiselle löytyy. Ympäristö voi toimia erilaisilla käyttöliittymillä ja ohjelmistoilla. Kuviossa 73 on esitetty yleisesti käytössä olevia avoimen lähdekoodin ympäristöjä. (Harash 2023, kappale 5.)



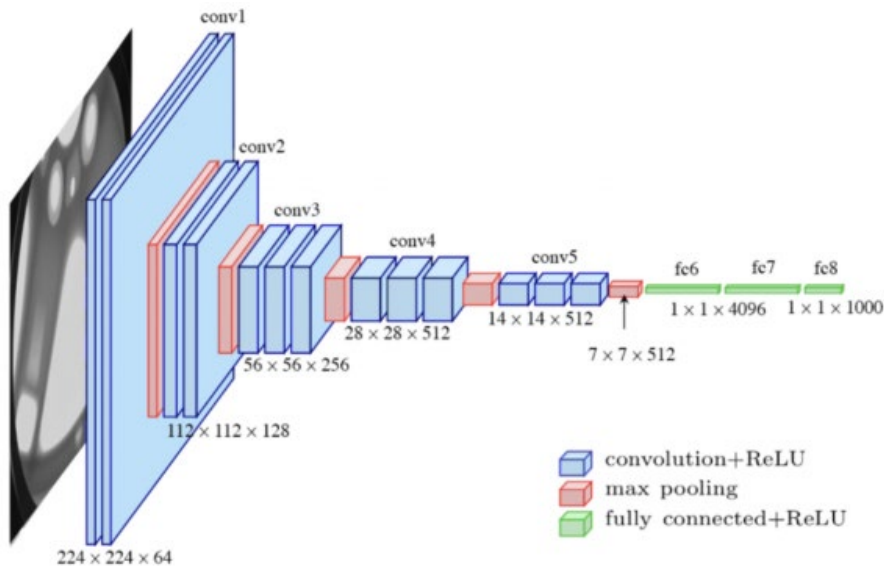
Kuvio 73. Avoimen lähdekoodin ympäristöjä koneoppimismallien tekemiseen (Harash 2023, kappale 5.)

Yksi yleisin ohjelmointikieli koneoppimismallien kehittämiseen on Python. Siinä yhdistyy yleiskäyttöisten ohjelmointikielten tehokkuuden ja Matlabin ja R-ohjelmointikielen komentosarjamainen helppokäyttöisyys. Pythonilta löytyy myös erittäin laajasti tarvittavia kirjastoja tiedostojen lataamiseen ja käsittelyyn, visualisointiin, kuvankäsittelyyn ja koneoppimistyökaluihin. Python mahdollistaa myös käyttöliittymien tekemisen ja liityntärajapinnat muihin ohjelmistoihin ja järjestelmiin. Python-ohjelmointia pystyy tekemään myös suoraan selaimen kautta Jupyter Notebook:lla, mikä on interaktiivinen ympäristö ohjelmakoodin tekemiseen ja ajamiseen. (Müller & Guido 2016, 7.)

### **Mallin opettamisen ja hyödyntämisen haasteet**

Kuten aikaisemmin on todettu, mallin tarkkuutta pystytään parantamaan tekemällä opetusaineistosta mahdollisimman kattavan, että kaikki mahdolliset vaihtoehdot ja ominaisuudet tulee opetettua koneoppimismallille. Tämä tarkoittaa satoja, tuhansia tai jopa satojatuhansia kuvia. Kuvien määrän lisäksi konvoluutioneuroverkon syventäminen useilla kerroksilla auttaa erilaisten piirteiden tunnistamisessa. Tämän lisäksi kuvakoko määrittää, miten paljon käsiteltävää aineistoa on. Jos isolta kuva-alueelta halutaan tunnistaa pieniä kohteita ja poikkeamia, täytyy kuvan resoluutio olla riittävä. Monesti esimerkkitapauksissa kuvan sivun pituus saattaa olla kuitenkin vain kymmeniä tai satoja pikseleitä pitkä (Denneman 2022). Toisin sanoen kuvakoko vaihtelee 0.01–0.1 Mpix välillä. Tämän kokoinen kuva ei ole kovin suuri, jos verrataan teollisuudessa hyödynnettäviin kameroihin mitä löytyy yleisesti väliltä 2–24 Mpix (ACE2 Basic & Pro – matriisikamerat n.d.).

Pieneen kuvakokoon löytyy selitys opettamiseen tarvittavasta muistimäärästä. Jos esimerkiksi tarkastellaan VGG-16 konvoluutioneuroverkon mallia, missä on yhteensä 13 konvoluutiotasoa, viisi max Poolin -toimintoa ja kolme neuroverkkokerrosta (ks. kuvio 74).



Kuvio 74. VGG-16 konvoluutioneuroverkon rakenne (Denneman 2022.)

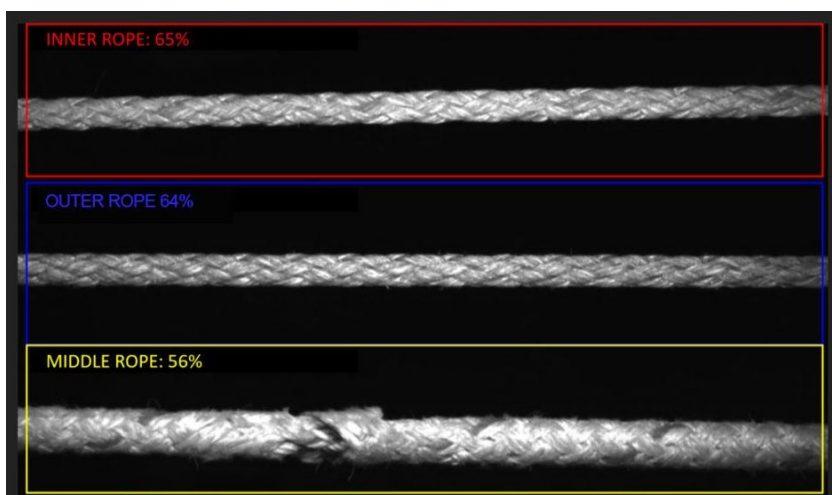
Muistin käyttö voidaan jakaa kahteen luokkaan, mitkä ovat parametrit, eli painokertoimet ja bias-muuttujat, sekä piirrekuvien aktivointitietojen tallennus. Piirrekuvan muistivaraus riippuu syötteen annettavan kuvan koosta, eli leveydestä ja korkeudesta, sekä batch-erän koosta. Kuva- ja eräkoko ei kuitenkaan vaikuta parametrien muistivaraukseen. VGG-16:n syötteen annetaan  $224 \times 224$  kokoinen värikuva. Ensimmäisessä konvoluutiokerroksessa on 64 suodatinta, missä käytetään  $3 \times 3$ -kokoista kerneliä. Värikuvassa käsitellään jokainen väri erikseen, eli punainen, vihreä ja sininen väri käsitellään 64 kertaa  $3 \times 3$ -kernelillä. Tästä saadaan laskettua painokertoimien määräksi  $(3 * 3) * 64 * 3 = 1728$ . Painokertoimien lisäksi, jokaista kerrosta kohti on yksi bias-muuttuja. Tästä saadaan parametrien yhteismääräksi  $1728 + 64 = 1792$ . Parametrit tallennetaan liukulukuina, eli jokainen parametri vie yhteensä neljä tavua tallennustilaa. Ensimmäinen VGG-16 konvoluutioneuroverkon piirrekuva tarvitsee siis yhteensä n. 7KB tallennustilaa. Toisessa kerroksessa on myös 64 suodatinta  $3 \times 3$ -kernelillä. Toiseen kerrokseen tulee syötteen edellisen kerroksen piirrekuvat, eli parametrien määräksi tulee  $(3 * 3) * 64 * 64 = 36864$ . Kun arvoon lisätään 64 kappaletta bias arvoja, tulee tallennustilan tarpeeksi  $36864 + 64 = 36928$  mikä on n. 147 KB. Poolin kerroksessa ei ole tallennettavia parametreja tai kertoimia, mutta kuvakoko muuttuu. Tämä vaikuttaa seuraavien konvoluutiokerrosten kokoon. Samalla tavalla laskettaisiin kaikkien konvoluutiotoimintojen ja neuroverkon kerrosten muistin tarve, kun Flatten-toiminnon jälkeen neuroverkolle tulee syötteen 25088 aktivointitietoa. Kokonaisuudessa VGG-16 tarvitsee siis yhteensä 540 MB



resoluution kuvalle ja laskettaisiin muistintarve samalla tavalla, yhden kuvan käsittely varaisi n. 9,6 Gb ja 32 kuvan sarjalla peräti yli 307 Gb. (Denneman 2022.)

Opetusajan lisäksi kuvien käsittely opetetulla mallilla voi vaatia paljon laskentatehoa ja aikaa. Monessa teollisessa sovelluksessa tahtiaika saattaa olla liian nopea raskaan koneoppimismallin hyödyntämiseen, jos kuvien käsittelyajat nousevat liian korkeaksi. Tällaisissa tapauksissa kuvakokoa täytyy pienentää tai mallia keventää. Joissain tapauksissa voidaan hyödyntää myös videokuvaa ja erillistä analyysiä. Esimerkiksi laadun heikentymisen tai kunnossapidon tarkastelussa ei välttämättä tarvitse tutkia jokaista kappaletta tai saata kuvaa ja analyysiä reaaliaikaisesti.

Esimerkiksi Procemex on hyödyntänyt paperikoneen päänvientiköysien kunnonvalvonnassa koneoppimista (ks. kuvio 76). Haasteeksi järjestelmässä muodostuu köyden liikenopeus. Vaikka kuvaaineistoa saadaan kerätty reaaliaikaisesti, ei koneoppimismallin analyysi toimi riittävän nopeasti näin haastavaan ympäristöön. Tämän takia järjestelmässä hyödynnetään videotallennetta, mikä käsitellään koneoppimismallilla. Tallenteesta pystytään havaitsemaan muutokset ja poikkeamat. (Ilonen & Virkajärvi 2023)

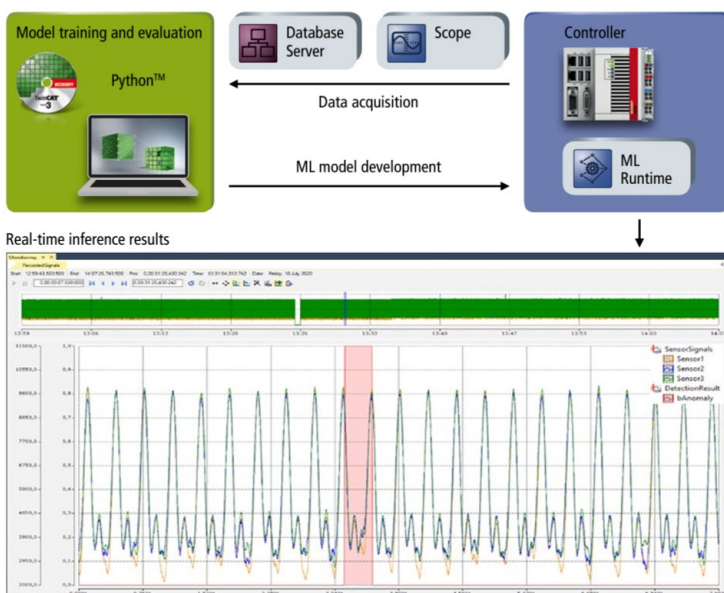


Kuvio 76. Procemex paperikoneen päänvientiköysien laadunvalvonta (Automatic Threading Rope Monitoring.)

## Mallin käyttäminen kuvien tunnistuksessa

Yksinkertainen menetelmä koneoppimisen hyödyntämiseen kuvien tunnistuksessa on käyttää samaa ohjelmistoympäristöä, millä malli on myös opetettu. Python ympäristö soveltuu tähän hyvin, koska ympäristöön voi asentaa esimerkiksi ilmaisen kuvien lukemiseen ja muokkaamiseen tarkoitetun OpenCV -kirjaston. OpenCV on avoimen lähdekoodin konenäkökirjasto, mikä sisältää yli 2500 algoritmiä. OpenCV:tä voi käyttää myös laajasti muissa ohjelmointiympäristöissä ja monella eri käyttöjärjestelmällä. OpenCV:stä löytyy versiot muun muassa C++, Python, Java ja Matlab kieleille, sekä tuki Windowsille, Linuxille, Androidille ja Mac OS:lle. (OpenCV n.d.)

Koneoppimismallia ei ole aina mahdollista tai järkevää suorittaa erillisellä järjestelmällä, vaan malli halutaan sisällyttää esimerkiksi automaatiojärjestelmän tai logiikan käyttöön. Tällaisessa tapauksessa voi olla tarpeen muokata malli toiselle järjestelmälle sopivaksi. Kuviossa 77 on esitetty Beckhoffin järjestelmän toimintakuvaus koneoppimismallien hyödyntämisestä. Aineistoa voidaan kerätä ja tallentaa oikeasta järjestelmästä Beckhoffin omien toimintojen avulla. Tallennetusta aineistosta hyödyntäen opetetaan järjestelmään soveltuva koneoppimismalli Python ympäristössä. Opetettu malli voidaan siirtää Beckhoffin järjestelmään, missä sitä voidaan ajaa reaaliaikaisesti. (PCcontrol The New Automation Technology Magazine 2021.)



Kuvio 77. Beckhoff TwinCAT3 koneoppimismallin opettaminen ja hyödyntäminen (PCcontrol The New Automation Technology Magazine 2021.)

Koneoppimismallit eivät aina sovellu käytettäväksi suoraan mallin opetusvaiheessa tallennetussa muodossa. Tähän on ratkaisuna hyödyntää järjestelmässä olevia muuntimia, tai malli voidaan tallentaa ONNX-muodossa. ONNX tulee sanoista Open Neural Network Exchange ja sen tarkoituksena on tarjota yhteinen kieli erilaisten työkaluympäristöjen luomille koneoppimismalleille. ONNX-malliin pystytään tallentamaan kaikki mallin ominaisuudet ja kohdejärjestelmä osaa tulkita mallin oikealla tavalla. (ONNX Concept 2024) Esimerkiksi Beckhoff TwinCAT3 -ympäristössä tulee koneoppimiskirjaston mukana muunnintyökalu, millä ONNX-tiedostona tallennetut mallit pystytään kääntämään TwinCAT3-ympäristölle sopivaan muotoon. Mallien muunnoksissa on tärkeä selvittää, minkä tyyppisiä malleja järjestelmät tukevat. Taulukossa 7 on esitetty TwinCAT3:n tukemat koneoppimisalgoritmit. Taulukosta nähdään myös, että tällä hetkellä järjestelmässä ei ole esimerkiksi konvoluutioneuroverkkoja käytettävissä. (Samples of ONNX export n.d.)

Taulukko 6. Bechoff TwinCAT3 ML-kirjaston käytettävissä olevista algoritmeista (Samples of ONNX export n.d.)

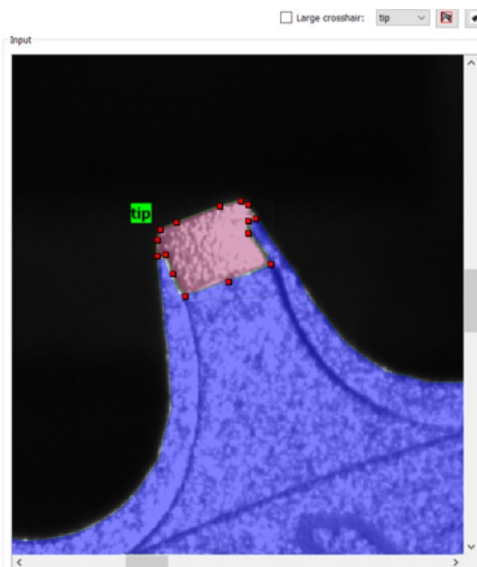
Python package	Model type	Option	Comment
PyTorch	MLP Regressor		
Keras	MLP Regressor		
Scikit-learn	MLP Regressor		
Scikit-learn	MLP Classifier		ONNX graph must be adapted
Scikit-learn	SVR		
Scikit-learn	SVC	decision_function_shape='ovo'	
Scikit-learn	k-means		Meta Key must be entered in ONNX.
Scikit-learn	PCA		
Scikit-learn	Decision Tree Classifier		
Scikit-learn	Decision Tree Regressor		
Scikit-learn	Extra Tree Classifier		
Scikit-learn	Extra Tree Regressor		
Scikit-learn	Extra Trees Classifier		
Scikit-learn	Extra Trees Regressor		
Scikit-learn	Random Forest Classifier		
Scikit-learn	Random Forest Regressor		
LightGBM	Random Forest Regressor		ONNX graph must be adapted
Scikit-learn	Gradient Boosting Classifier		
Scikit-learn	Gradient Boosting Regressor		
Scikit-learn	Hist Gradient Boosting Classifier		
Scikit-learn	Hist Gradient Boosting Regressor		
XGBoost	XGBClassifier	Not all configurations allow an ONNX export	Package version <= 1.5.2 or >= 1.7.4 required
XGBoost	XGBRegressor	Not all configurations allow an ONNX export	Package version <= 1.5.2 or >= 1.7.4 required
LightGBM	LGBMRegressor	Not all configurations allow an ONNX export	
LightGBM	LGBMClassifier		ONNX graph must be adapted

## 5.2 Koneoppimista hyödyntävät kaupalliset konenäköjärjestelmät

Erittäin monet kaupalliset konenäkökirjastot sisältävät tänä päivänä ainakin jonkin tasoisia koneoppimistyökaluja. Usealta kaupalliselta konenäköjärjestelmän ja -kirjastojen toimittajalta löytyy lisäksi myös oma kehitysympäristö, mihin on lisätty koneoppimistoimintoja käyttöä varten lisäosia tai toimintoja. Tällaisia ominaisuuksia ovat esimerkiksi käyttöliittymät aineiston hallintaan ja luokkien merkitsemiseen, sekä mallin opettamiseen ja toiminnan arviointi testiaineiston perusteella. Alla on listattu muutama esimerkki, mistä löytyy vastaavia ominaisuuksia:

- MVTec Halconin Deep learning tool -työkalusta (Halcon deep learning tool n.d)
- Zebran Aurora Imaging COPilot -työkalusta (Zebra's Deep Learning Capabilities n.d.)
- Cognexin ViDi -työkalusta (Visionpro deep learning n.d.)
- SICK:in pilvipalvelupohjaisesta dStudio-työkalusta (Tekoälyratkaisut: automaation todellinen voima n.d)
- Teledyne Dalsa:n Astrocyte-työkalusta (Astrocyte n.d.)

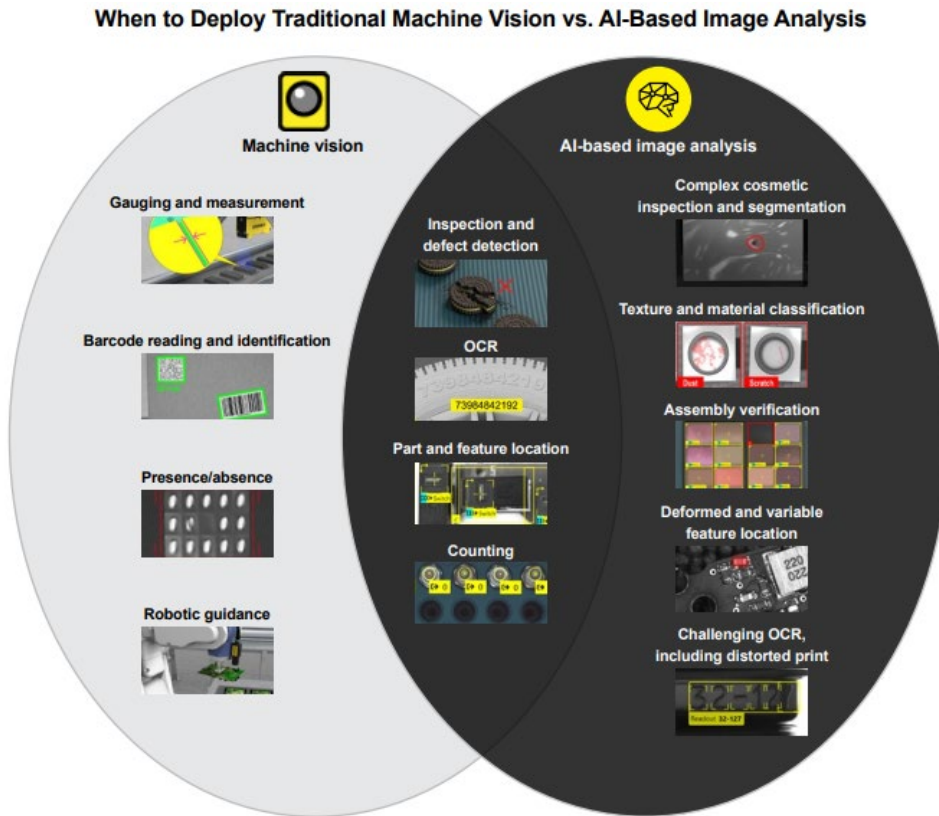
Ohjelmistojen ominaisuuksia on kehitetty helpottamaan ja nopeuttamaan koneoppimismallien tekemistä ja tulosten arviointia. Kuvien merkkaukseen eri luokkiin toimii helposti hiirellä painamalla tai ohjattua oppimista pystyy tekemään piirtämällä kuvioita tunnistettavien alueiden päälle (Visionpro deep learning n.d.). Joistakin työkaluista löytyy lisäksi perinteisiä konenäkötoimintoja hyödyntäviä automaattisia toimintoja esimerkiksi alueiden automaattiseen rajaukseen hahmontunnistuksen perusteella (ks. kuvio 78) (Astrocyte n.d.).



Kuvio 78. Astrocyte-työkalun esimerkki tuotteen kärjen merkkauksesta (Astrocyte n.d.)



Yleisimpiä sovelluskohteita, mihin koneoppimista voidaan hyödyntää kaupallisten toimijoiden sivustojen mukaan ovat merkkientunnistus, erilaiset laadunvalvonnat, tuotteiden tunnistus ja laskeminen, sekä paikoitus esimerkiksi robottipoimintaa varten. Pääpaino on kuitenkin selkeästi haastavissa pintavirheiden ja laadun tarkastussovelluksissa. (Ks. kuvio 79)



Kuvio 79. Perinteisen ja koneoppimistahyödyntävät konenäköjärjestelmän vertailu (AI for factory automation 2022.)

### 5.2.1 Koneoppimista hyödyntäviä tuotteita

Konenäkökameroiden valmistajilta löytyy myös erilaisia tuotteita, missä koneoppimista hyödynnetään hieman eri tavoilla. Koneoppimismallien rakenne ja koulutus voidaan jakaa neljään kategoriaan, mitkä ovat avoimet mallit, rajoitetut mallit, osittaiset mallit ja valmiit mallit. Avoimella mallilla tarkoitetaan järjestelmää, mihin pystyy määrittelemään esimerkiksi konvoluutioneuroverkon rakenteen haluamallaan tavalla. Rajoitettu malli tarkoittaa, että järjestelmässä on käytettävissä tietyntyyppiset rakenteet, mitä pystyy tietyissä rajoissa muokkaamaan. Osittainen malli tarkoittaa val-

miiksi opetettua mallia, mitä voidaan siirto-opettaa pienellä otannalla uusia kuvia valitusta kohteesta. Valmiita malleja hyödyntävissä järjestelmissä ei ole enää mahdollisuutta vaikuttaa malliin ja niissä tehdäänkin muutoksia lähinnä tunnistusparametreihin.

Useassa tuotteessa on jollakin tavalla rajoitettu malli tai viimeistään laitteen laskentakapasiteetti ja muisti rajoittavat mallin ominaisuuksia. Rajoitetuista malleista on esimerkkinä Cognexin D900 -älykamera. Kameran ohjelmointi tapahtuu tietokoneella, mutta kuvien analysointi mukaan lukien koneoppimismallin käsittely, tapahtuu kameralla. Tuotteesta löytyy erilaisia koneoppimismalleja ja opetus voi tapahtua ohjattua tai ohjaamatonta oppimista hyödyntäen. Koneoppimistyökaluina kamerasta löytyy perinteisten konenäkötoimintojen lisäksi koneoppimista hyödyntävä merkkien tunnistus, sekä luokittelu- ja tunnistustyökalut. (In-Sight D900 vision system n.d.)

Osittaisia malleja hyödyntäviä laitteita kutsutaan välillä myös Edge-laitteina, koska laskenta tapahtuu kameralla. Esimerkki tällaisesta laitteesta on Cognex IS2800-älykamera. Laitteen toiminta perustuu esiopetettuun koneoppimismalliin, mitä voidaan opettaa pienellä otannalla uusia kuvia. Laitteesta löytyy perinteisten konenäkötoimintojen lisäksi koneoppimiseen pohjautuva luokittelu-toiminto. Lisäksi laitteesta löytyy koneoppimista hyödyntävä merkkientunnistus. Kummassakin tapauksessa opettaminen tehdään pienellä, kymmenten kuvien opetusaineistolla ja taustalla hyödynnetään Cognexin tekemään esiopetettua koneoppimismallia. (Machine Vision System Sensors Software n.d.)

Täysin valmiita malleja hyödynnetään esimerkiksi monissa kasapoimintasovelluksissa. Photoneo:n AnyPick ja Robominds:in Robobrain, hyödyntävät valmiiksi opetettuja neuroverkkoja poimittavien tuotteiden tunnistamiseen. Neuroverkot on valmiiksi opetettu suurella määrällä erilaista aineistoa, minkä perusteella pystytään tunnistaman kappaleidien tartuntapintoja. Järjestelmiin ei siis tarvitse määrittää 3D-mallia, minkä perusteella kameran tuottamasta pistepilviaineistosta pyritään löytämään vain opetettu muoto. Koneoppimisen avulla kuvasta tunnistetaan vain poimittavaksi soveltuvia tuotteita tai pintoja. (3D-ohjelmistoratkaisut; Every robot needs a brain n.d.)

## 6 Järjestelmien vertailu

Konenäköjärjestelmien vertailua varten valittiin Jamkilla käytössä oleva MVTec Halcon Deep Learning Tool, coADDVA-hankkeessa ostettu Cognex ViDi, sekä avoimen lähdekoodin järjestelmä. Järjestelmävertailun tavoitteena oli selvittää, millaisia toimintoja koneoppimiseen liittyen näistä järjestelmistä löytyy, miten ne poikkeavat toisistaan ja onko järjestelmissä käyttöä helpottavia toimintoja. Selvitettäväksi asioiksi listattiin seuraavat ominaisuudet:

- Miten aineisto valitaan ja luokitellaan
  - o onko opetus, testi ja validointiaineistolle automaattitoimintoja
- Millaisia malleja järjestelmistä löytyy, mitä voidaan hyödyntää
- Pystytäänkö hyödyntämään valmiiden mallien siirto-opetusta (transfer learning)
- Onko järjestelmässä tuki näytönohjaimella tapahtuvaa opettamista ja mallin ajoa varten
- Miten tulosten varmentaminen tapahtuu
  - o arvojen näyttö
  - o kuvaajien piirto
  - o testiaineiston näyttö
  - o virheellisten valintojen näyttö
- Uudet opetuskerrat ja tulosten vertailu

Järjestelmien vertailua varten valittiin yhtenäinen kuva-aineisto, millä toimenpiteiden testaaminen tehdään. Halconin ja Cognexin järjestelmistä löytyy valmiita kuva-aineistoja, mutta nämä jätettiin huomioimatta mahdollisten optimointien takia. Aineistoksi valittiin Kaggle-sivuston kautta ladattu kuvasarja valukappaleista (Dabhi 2020). Kaggle-sivusto on tarkoitettu data-analyytikoille ja koneoppimisesta kiinnostuneille. Kaggle järjestää myös erilaisia kilpailuita, missä kilpailaan muita vastaan ja pyritään löytämään paras tai tehokkain ratkaisu kilpailtavaan aiheeseen. Sivuston kautta voi myös jakaa omia projekteja tai kuten tässä tapauksessa, kuvasarjoja, millä omia sovelluksia pystyy kehittämään ja testaamaan.

### 6.1 Avoimen lähdekoodin koneoppimistestaus

Järjestelmien testaamisen ensimmäisenä vaiheena oli tutustua avoimen lähdekoodin järjestelmään. Järjestelmäksi valittiin Anaconda-ohjelmisto, mihin asennettiin Spyder-kehitysympäristö Python-ohjelmointia varten. Avoimen lähdekoodin järjestelmiin täytyy asentaa itse tarvittavat kirjastot ja koneoppimisen ympäristöksi valittiin TensorFlow. Kirjastojen asennuksessa täytyy olla tarkkana oikeiden versioiden kanssa. Järjestelmän asennuksessa havaittiin haasteita näytönohjai-

men käytön suhteen. TensorFlow-ympäristö osaa hyödyntää nVidian näytönohjaimia, mutta järjestelmästä on kahta eri versiota, toinen prosessorille, toinen näytönohjaimille. Näytönohjainversion lisäksi tietokoneen näytönohjaimella täytyy olla oikeat ajuriversiot asennettuna.

TensorFlow:n lisäksi järjestelmään täytyi asentaa seuraavat kirjastot:

- Numpy: laskenta, taulukot, matemaattiset funktiot
- Matplotlib: visualisointi, kaaviot & kuvaajat
- Seaborn: visualisointi, kaaviot & kuvaajat
- OpenCV: kuvien käsittely, konenäkötoiminnot
- Scikit-learn: data-analytiikka, luokittelu, regressio, klusterointi

Opetuskuvien valitseminen Python ympäristössä tapahtuu lukemalla kansiossa olevat tiedostot omaan taulukkoon. Kansioden määrittämisen jälkeen kaikki kansiossa olevat kuvat luettiin ja tallennettiin taulukkoon. Tallennettavat kuvat myös skaalataan tarvittaessa konvoluutioneuroverkolle valittuun kuvakokoon 300 x 300 pikseliä, mutta tässä tapauksessa kuvat ovat jo valmiiksi oikeassa koossa, joten varsinaista skaalausta ei tapahdu (ks. kuvio 80).

```

#%% Luokittelu kansioden nimien perusteella
CATEGORIES = ['def_front', 'ok_front']
IMG_SIZE = 300

train_data = []
valid_data = []

#Train-datan lukeminen ja koon muutos
for category in CATEGORIES:
    path = os.path.join(folder, category).replace("\\", "/")
    class_num = CATEGORIES.index(category) #ok = 1, def = 0
    for img in os.listdir(path):
        image_data = cv2.imread(os.path.join(path, img), 0)
        image_data = cv2.resize(image_data, (IMG_SIZE, IMG_SIZE)).reshape(1, IMG_SIZE, IMG_SIZE, 1)
        train_data.append([image_data, class_num])
print("Train data: ", len(train_data))

X_train = []
y_train = []

for features, label in train_data:
    X_train.append(features)
    y_train.append(label)

```

Kuvio 80. Kuvien lukeminen ja skaalaus Python-ympäristössä.

Sama toimenpide toistetaan testiaineistolle (ks. kuvio 81). Opetus- ja testiaineistojen kuvien lukumäärä voidaan näyttää käyttämällä print-komentoa. Tässä tapauksessa aineistojen määrät ovat

opetusaineisto 6633 kuvaa ja testiaineisto 715 kuvaa.

```

#####
#Testi-datan lukeminen ja koon muutos
for category in CATEGORIES:
    path = os.path.join(test_data_dir, category)
    class_num = CATEGORIES.index(category) #ok = 1, def = 0
    for img in os.listdir(path):
        image_data = cv2.imread(os.path.join(path,img), 0)
        image_data = cv2.resize(image_data, (IMG_SIZE, IMG_SIZE)).reshape(1,IMG_SIZE,IMG_SIZE,1)
        test_data.append([image_data, class_num])
    print("Test data: ",len(test_data))

random.shuffle(test_data)

X_test = []
y_test = []

for features, label in test_data:
    X_test.append(features)
    y_test.append(label)

```

Kuvio 81. Testiaineiston kuvien lukeminen Python ympäristössä

Pikseleiden harmaasävyarvot muutettiin välille 0...1, jakamalla harmaasävyn arvo luvulla 255 (ks. kuvio 82). Arvot muutettiin sekä opetus- että testiaineistoon. Arvo muutetaan neuroverkon toiminnan takia.

```

#####
X_train = np.array(X_train).reshape(-1,IMG_SIZE,IMG_SIZE,1)
X_train = X_train/255
y_train = np.array(y_train)

X_test = np.array(X_test).reshape(-1,IMG_SIZE,IMG_SIZE,1)
X_test = X_test/255
y_test = np.array(y_test)

```

Kuvio 82. Kuvien harmaasävyarvojen muuttaminen välille 0...1

Konvoluutioneuroverkon rakenteeksi valittiin kolme konvoluutioverkkoa, joiden jälkeen tehdään max Pooling -toiminto. Aktivointifunktioksi valittiin ReLU-funktio. Konvoluutiotoimintojen jälkeen tehdään flatten-toiminto, minkä arvot syötetään 128 neuronia sisältävään neuroverkkoon. Malliin lisättiin myös Dropout (0,5) -toiminto, mikä auttaa parantamaan mallin yleistymiskykyä ja vähentää ylikoulutusta. Dropout-toiminto jättää satunnaisesti osan yksiköistä pois, tässä tapauksessa 50 %. Viimeiseen kerrokseen tulee yksi neuroni, koska luokkia on tässä tapauksessa vain kaksi kappaletta, hyvät ja huono. Viimeisen neuronien aktivointifunktiona käytettiin Sigmoid-funktiota. Lear-

ning rate -hyperparametrin arvoksi valittiin 0.001. Arvo määrittää, miten malli sopeutuu ongelmaan muuttamalla mallin painokertoimia. Lopuksi määritetään mallin nimi, mihin painokertoimet tallennetaan. Tiedostoon tallennetaan parhaan opetuskerran painokertoimien arvot. (ks. kuvio 83)

```

#####
#Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.25, random_state=5)

model = tf.keras.Sequential()
model.add(tf.keras.layers.Conv2D(32, kernel_size=(3,3), input_shape=(300, 300, 1), activation='relu'))
model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2)))
model.add(tf.keras.layers.Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2)))
model.add(tf.keras.layers.Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(tf.keras.layers.MaxPool2D(pool_size=(2,2)))
model.add(tf.keras.layers.Flatten())
model.add(tf.keras.layers.Dense(128, activation='relu'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(1, activation='sigmoid'))
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss=tf.keras.losses.BinaryCrossentropy(),
              metrics=['accuracy'])

log_dir = './logs'
checkpoint_cb = keras.callbacks.ModelCheckpoint('casting.h5', save_weights_only=True,
                                              monitor='val_loss', mode='min', save_best_only=True)

```

Kuvio 83. Konvoluutioneuroverkon rakenne

Kuviossa 84 on kuvattuna mallin rakenne ja parametrien määrä.

Layer (type)	Output Shape	Param #
conv2d_9 (Conv2D)	(None, 298, 298, 32)	320
max_pooling2d_9 (MaxPooling2D)	(None, 149, 149, 32)	0
conv2d_10 (Conv2D)	(None, 147, 147, 32)	9248
max_pooling2d_10 (MaxPooling2D)	(None, 73, 73, 32)	0
conv2d_11 (Conv2D)	(None, 71, 71, 32)	9248
max_pooling2d_11 (MaxPooling2D)	(None, 35, 35, 32)	0
flatten_3 (Flatten)	(None, 39200)	0
dense_6 (Dense)	(None, 128)	5017728
dropout_3 (Dropout)	(None, 128)	0
dense_7 (Dense)	(None, 1)	129
=====		
Total params: 5,036,673		
Trainable params: 5,036,673		
Non-trainable params: 0		

Kuvio 84. Python-ympäristön konvoluutioneuroverkon rakenne ja parametrien määrä

Koneoppimismallin opettamiseen valittiin 8 epoch-opetussarjaa ja näytteiden batch-kooksi valittiin 30. Mallin opettamisen yhteydessä halutaan yleensä tietää, miten malli on käyttäytynyt eri opetuskertojen välillä. Kuvaajan piirtäminen mallin tarkkuudesta opetus ja testiaineistolla on erittäin te-

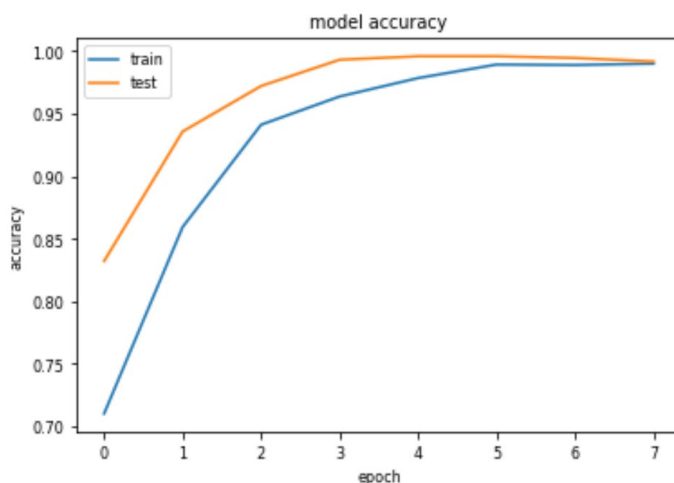
hokas menetelmä esittää opetuksen käyttäytyminen. Kuvaajien piirtämistä varten mallin eri opetusversiot täytyy tallentaa, mikä esimerkkitapauksessa on määritetty history-nimiseen muuttu-  
jaan. (ks. kuvio 85)

```
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=8, batch_size=30, callbacks=[checkpoint_cb])
predictions_train = model.predict(X_train)
predictions_test = model.predict(X_test)
print(accuracy_score(y_test, np.round(predictions_test)))
```

Kuvio 85. Koneoppimismallin opetusmäärittely ja hirstorian tallentaminen

Mallin opettamisvaiheen kuvaajien piirto voidaan toteuttaa Matplotlib-kansiosta löytyvällä plot-toiminnolla. Kuvaajasta nähdään selkeästi, miten mallin tarkkuus kehittyy eri opetuskertojen välillä (ks. kuvio 86). Epoch-määrän nostaminen ei juurikaan vaikuta tulokseen enää viimeisillä opetuskerroilla, minkä takia arvoa ja opetuskertojen määrän kasvattamisesta ei ole hyötyä.

```
%%
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



Kuvio 86. Mallin opetushistorian piirtäminen

Eri opetuskierrosten tulokset nähdään myös ohjelman konsolinäkymästä. Esimerkkitaupauksessa opetusaineiston tarkkuus on parhaimmillaan 98,99 prosenttia viimeisen opetuskerran yhteydessä,

mutta testiaineistolla tarkkuus on ollut parhaimmillaan neljännen opetuskerran yhteydessä tarkkuuden yltyessä 99,3 prosenttiin. (ks. kuvio 87)

```

Train data: 6633
Valid data: 715
Epoch 1/8
188/188 [=====] - 9s 46ms/step - loss: 0.5555 - accuracy: 0.7104 - val_loss: 0.3583 - val_accuracy: 0.8322
Epoch 2/8
188/188 [=====] - 8s 44ms/step - loss: 0.3135 - accuracy: 0.8593 - val_loss: 0.1551 - val_accuracy: 0.9357
Epoch 3/8
188/188 [=====] - 8s 44ms/step - loss: 0.1571 - accuracy: 0.9411 - val_loss: 0.0627 - val_accuracy: 0.9720
Epoch 4/8
188/188 [=====] - 8s 43ms/step - loss: 0.1052 - accuracy: 0.9636 - val_loss: 0.0361 - val_accuracy: 0.9930
Epoch 5/8
188/188 [=====] - 8s 44ms/step - loss: 0.0662 - accuracy: 0.9784 - val_loss: 0.0158 - val_accuracy: 0.9958
Epoch 6/8
188/188 [=====] - 8s 43ms/step - loss: 0.0373 - accuracy: 0.9892 - val_loss: 0.0165 - val_accuracy: 0.9958
Epoch 7/8
188/188 [=====] - 8s 43ms/step - loss: 0.0373 - accuracy: 0.9888 - val_loss: 0.0186 - val_accuracy: 0.9944
Epoch 8/8
188/188 [=====] - 8s 43ms/step - loss: 0.0358 - accuracy: 0.9899 - val_loss: 0.0295 - val_accuracy: 0.9916
0.9916083916083916

```

Kuvio 87. Mallin tarkkuuden kehittyminen eri opetusvaiheen edetessä.

Mallin testaamista varten voidaan ottaa myös näytekuvia testi aineistosta, mitkä voidaan näyttää käyttäjälle mallin tunnistustietojen kanssa. Tällä tavalla voidaan arvioida mallin toimivuutta yksittäisillä kuvilla. Testikuvat arvottiin satunnainen kuuden kuvan sarja, mitkä ajettiin opetetulla koneoppimismallilla. Kuva ja mallin ilmoittama luokka tunnistusprosentilla tulostettiin plot-toiminnolla. (Ks. kuvio 88)

```

###
#Kuvien ja tulosten näyttäminen
plt.gray()
# setting font sizeto 10
plt.rcParams.update({'font.size': 8})

IMG_SIZE = 300
i = 1
for i in range(1, 7):
    x = random.randint(0,715)

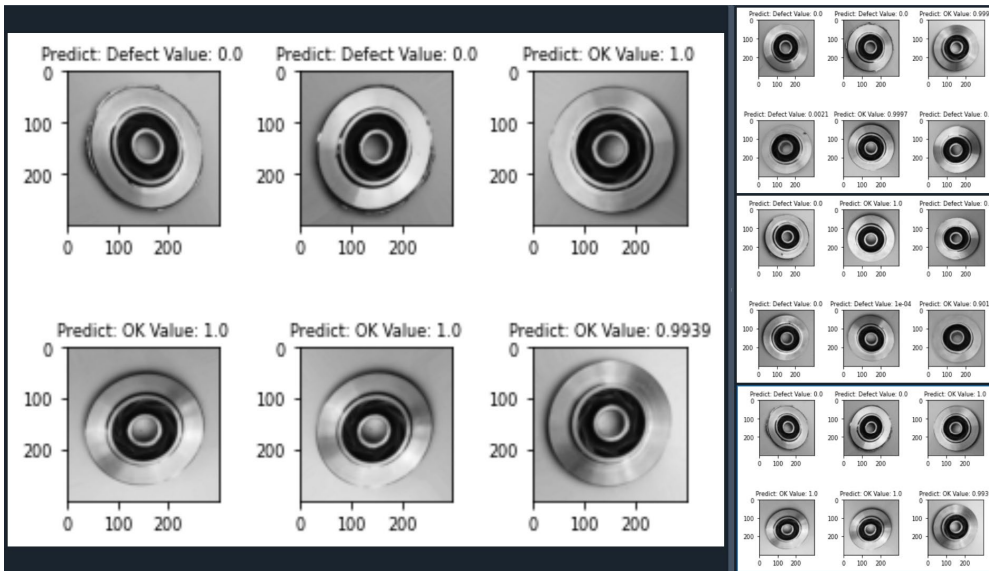
    image = (X_valid[x])
    image_re = np.array(image).reshape(-1, IMG_SIZE, IMG_SIZE, 1)
    test_classes = np.round(model.predict(image_re))
    test_predict = model.predict(np.array([image_re], dtype="float32")[0])
    if test_classes == 0:
        arvaus = "Defect"
        luokka = round(test_predict[0,0],4)
        luokka = str(luokka)
    else:
        arvaus = "Ok"
        luokka = round(test_predict[0,0],4)
        luokka = str(luokka)
    plt.subplot(2,3,i)
    plt.subplots_adjust(hspace=0.8)
    plt.title("Predict: " + arvaus + " " + "Value: " + luokka , fontsize=8)
    plt.imshow(image)
    print("Prediction class: {}".format(test_classes))
    print("Prediction: {}".format(test_predict))
    i += 1

```

Kuvio 88. Näytettävien testikuvien valitseminen kuva-aineistosta.

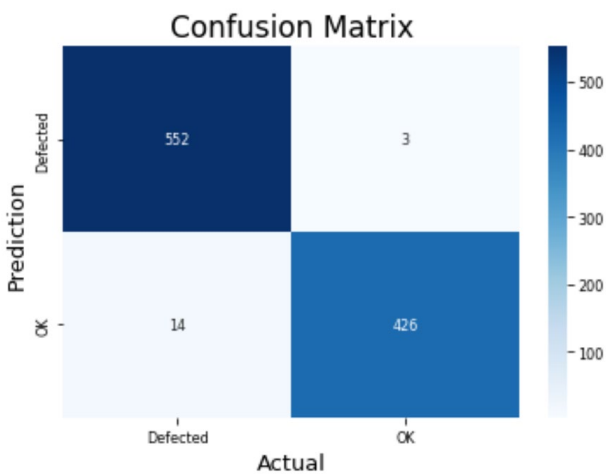


Tulostettuja kuvia voi olla yksi tai useita samassa näkymässä. Näkymään saadaan useampi tuote näytille, kun näyttö jaetaan subplot-toiminnolla. Kuvien luokittelu tapahtui luokittelulla 0 = huono ja 1 = hyvä. Mitä lähempänä arvo on nollaa, sitä todennäköisemmin tuotteessa on virheitä ja mitä lähempänä arvo on ykköstä, sitä todennäköisemmin tuote on hyvä. (Ks. kuvio 89)



Kuvio 89. Tulosteet näytettävistä testikuvista.

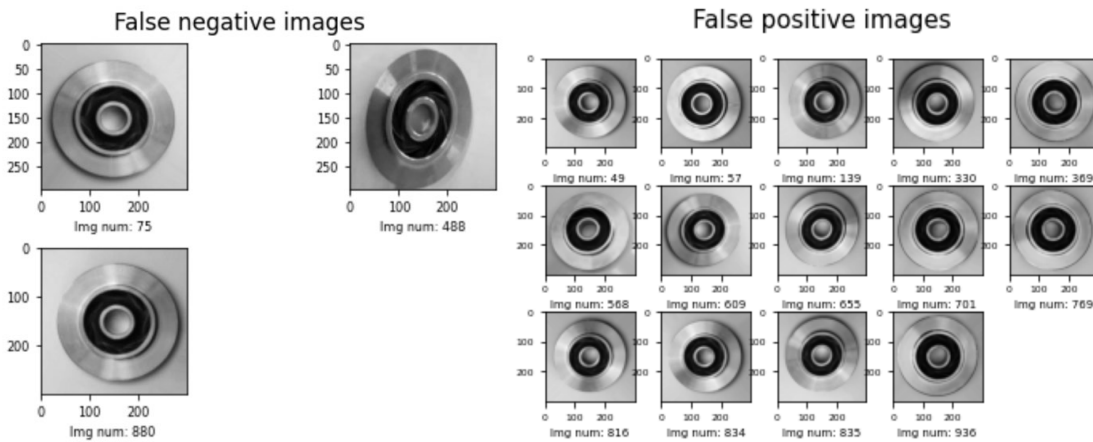
Mallin toimintaa voidaan tarkastella myös confusion matrix -toiminnon avulla, mihin voidaan lisäksi määrittää värisävyt kuvastamaan taulukossa olevia määriä. Taulukon luokkien mukaan voidaan myös tulostaa kuva erillisellä ohjelmakodilla. (Ks. kuvio 90)



Kuvio 90. Confusion matrix -taulukko testiaineistosta Python-ympäristössä

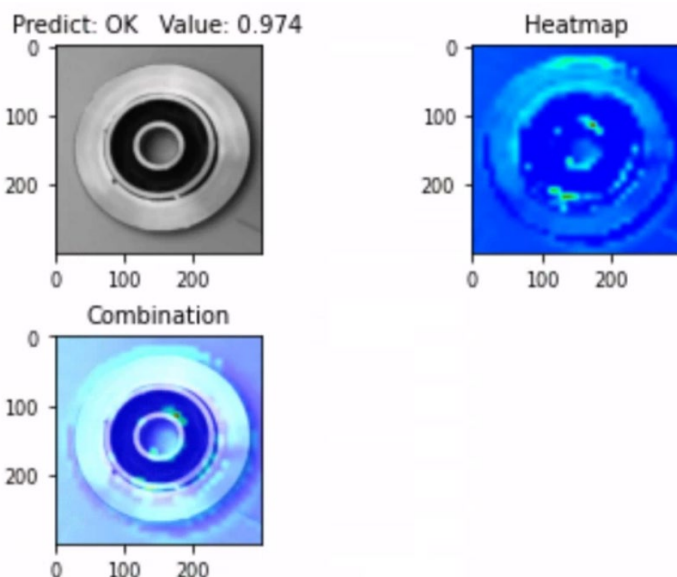
Alla olevassa kuvassa on tulostettuna erikseen false negative- ja false positive -kuvat (ks. kuvio 91).

Toimenpide täytyy suorittaa erillisellä komennolla.



Kuvio 91. Confusion matrix -taulukon kuvien tulostaminen.

Kuviin voidaan piirtää myös konvoluutiokerrosten piirteitä ja näyttää, mihin koneoppimismallin päättely perustuu. Yksi tapa on käyttää heatmap-menetelmää, missä konvoluutiokerroksen kuva näytetään väriarvoina tai sovitetaan alkuperäisen kuvan päälle yhdistelmäkuva. Konvoluutiokerroksen kuva ei ole todennäköisesti samankokoinen, kuin alkuperäinen kuva, minkä takia pienempi kuvakoko vain skaalataan vastaamaan alkuperäistä kuvaa. Tämän takia voi olla hyödyllistä esittää myös pelkkä konvoluutiokerroksen kuva, että nähdään mitä piirteitä neuroverkolle annetaan syötteeksi. Myös useita konvoluutiokerroksia voidaan tulostaa näkyviin, tai tehdä yhdistelmä konvoluutiokuvista ja sovittaa ne alkuperäisen kuvan päälle. Kuviossa 92 on esitetty hyvästä tuotteesta alkuperäinen, heatmap- ja yhdistelmäkuva.



Kuvio 92. Heatmap-kuvan esittäminen Python-ympäristössä

Siirto-opettamisen hyödyntäminen on myös mahdollista. Mallin tasot voidaan lukita muuttamalla Trainable-arvo tilaan False. Toiminnon jälkeen mallin rakenteen tarkastelussa opettavien ja lukittujen parametrien määrä on vaihtunut. Kerroksia pystyy myös lukitsemaan ja aukaisemaan yksitellen. (Ks. kuvio 93)

```
base_model = model
base_model.trainable = False
Total params: 5,036,673
Trainable params: 0
Non-trainable params: 5,036,673
```

Kuvio 93. Mallin kerrosten lukitseminen siirto-opetusta varten Python-ympäristössä

Esimerkiksi TensorFlow:n sivuilta osiosta Transfer learning & fine-tuning löytyy ohjeistus Xception konvoluutioneuroverkon hyödyntämiseen ImageNetin kuvilla. Xception-mallista hyödynnetään vain alemmat kerrokset ja jätetään ylin kerros pois latausvaiheessa. Mallille määritetään tulo-kerrokseksi 150 x 150 kokoinen värikuva ja neruoverkkokerros määritetään uudestaan. Tämän jälkeen malli opetettaisiin valituilla aineistolla ja parametreilla. (Ks. kuvio 94) (Transfer learning and fine-tuning)

```

base_model = keras.applications.Xception(
    weights='imagenet', # Load weights pre-trained on ImageNet.
    input_shape=(150, 150, 3),
    include_top=False)

base_model.trainable = False

inputs = keras.Input(shape=(150, 150, 3))
# We make sure that the base_model is running in inference mode here,
# by passing `training=False`
x = base_model(inputs, training=False)
# Convert features of shape `base_model.output_shape[1:]` to vectors
x = keras.layers.GlobalAveragePooling2D()(x)
# A Dense classifier with a single unit (binary classification)
outputs = keras.layers.Dense(1)(x)
model = keras.Model(inputs, outputs)

model.summary()

```

```

Download data from https://storage.googleapis.com/tensorflow/keras-
applications/xception/
xception_weights_tf_dim_ordering_tf_kernels_notop.h5
83689472/83683744 [=====] - 19s 0us/step
Model: "functional_1"

```

Layer (type)	Output Shape	Param #
input_12 (InputLayer)	[(None, 150, 150, 3)]	0
xception (Functional)	(None, 5, 5, 2048)	20861480
global_average_pooling2d_3 ( (None, 2048)		0
dense_58 (Dense)	(None, 1)	2049

```

Total params: 20,863,529
Trainable params: 2,049
Non-trainable params: 20,861,480

```

Kuvio 94. Esiopetetun Xception-mallin hyödyntäminen Python-ympäristössä

### Yhteenveto Python ympäristön testauksesta

Kirjastojen asennus vaatii selvittelyä varsinkin ensimmäisten asennusten yhteydessä. Mitä pitää asentaa, mitkä ovat yhteensopivia järjestelmiä ja kirjastoja, miten näytönohjainta saa hyödynnettyä mallien laskemiseen ja käsittelyyn?

Opetusaineiston valintaan on useita ratkaisuita, mutta ne täytyy rakentaa itse. Esimerkissä tehty kansioista lukeminen vaatii sen, että kuvat on valmiiksi tallennettu tietyn nimiseen kansioon. Luokkia voi olla useampiakin kuin kaksi. Toinen mahdollinen määrittelytapa on esimerkiksi lukea tiedostonimet ja tallentaa nimen perusteella kuva tiettyyn luokkaan.

Kuvien luokitteluun liittyvä työmäärä onkin varsin merkittävä avoimen lähdekoodin järjestelmissä. Miten kuvat jaetaan oikeisiin kansioihin tai miten kaikkiin kuviin tehdään tarvittavat merkinnät.

Kuvien merkintään voi hyödyntää erillisiä työkaluja, mitä löytyy varsin laajasti internetistä. Varsinkin ohjattua opettamista hyödynnettäessä, jonkinlainen kuvienkäsittelyohjelma on pakollinen. Kuvien käsittelyyn löytyy vaihtoehtoja niin maksullisista, kun myös ilmaiseksi käytettävistä sovelluksista. Alla olevassa taulukossa on listattuna esimerkkejä ilmaisista ja avoimeen lähdekoodiin pohjautuvista, sekä maksullisista ohjelmistoista.

Taulukko 7. Aineiston käsittelyyn soveltuvia ohjelmistoja ohajattua oppimista varten (Top 10 Open Source Data Labelling Tools for Computer Vision 2023 n.d.; 30 best data labeling tools 2024 n.d.)

<b>Ilmaiset ja avoimeen lähdekoodiin perustuvat</b>	<b>Rajoitetut tai maksulliset ohjelmistot</b>
Scalabel.ai	SuperAnnotate
RectLabel	Encord
MakeSense.AI	Kili
VGG Image Annotator (VIA)	Appen
Imglab	Dataloop
CVAT	Amazon SageMaker Ground Truth
LabelImg	Cogito Tech LLC
LabelMe	Labelerr
Tictag.io	Keymakr
VoTT	Labelbox

Avoimen lähdekoodin järjestelmistä löytyy erittäin laajasti erilaisia algoritmeja mallien opettamista varten. Konvoluutioneuroverkkojen rakenteita on helppo muokata ja kokeilla, mutta se vaatii tuntemusta käytettävissä olevista menetelmistä. Mitä mallia kannattaa hyödyntää minkäkin laiseen aineistoon ja miten mallin rakenteesta saadaan tehtyä järkevä.

Avoimen lähdekoodin ohjelmassa myös opetuksen tulosten esittäminen täytyy tehdä itse. Järjestelmästä löytyy erilaisia kirjastoja ja toimintoja, millä opettamiseen liittyvät tiedot saadaan tulostettua arvoina tai havainnointia helpottavina kuvaajina.

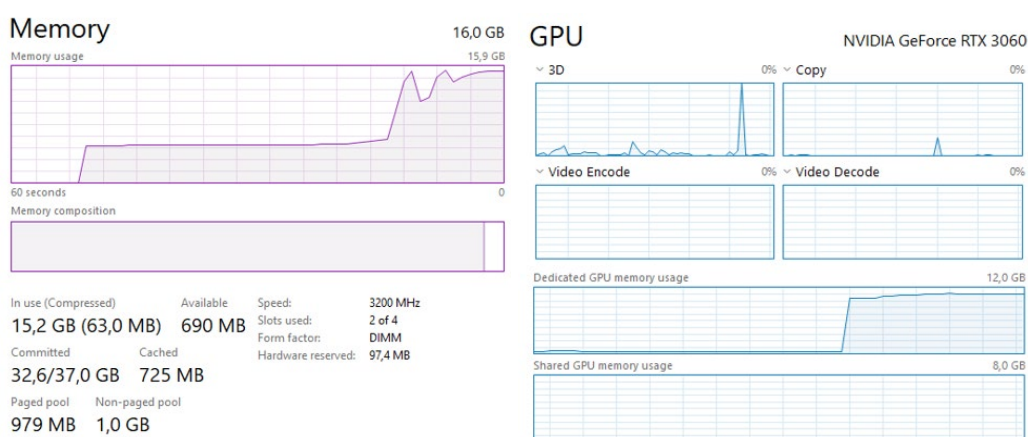
Taulukossa 9 on esitetty yhteenveto avoimen lähdekoodin järjestelmätestauksesta.

## Taulukko 8. Python ympäristön testaustulokset

Anaconda, Spyder IDE & Tensorflow	
Asennus ja tuki	Eri ohjelmistojen ja kirjastojen asennus manuaalisesti. Ohjeistusta löytyy paljon, mutta varsinkin alkuun työlästä. Vaatii kohtalaisen paljon työtä ja selvittelyä.
Aineiston valinta ja luokittelu	Ei automaattitoimintoja. Materiaalin valinta esimerkiksi kansikohtaisesti. Ohjattu oppiminen vaatii erillisen kuvien muokkaus nimeämistyökalun
Millaisia koneoppimismalleja voi hyödyntää	Erittäin monipuolinen ja tehokas tapa tehdä erilaisia malleja. Laskentatehon määrä rajoitteena.
Valmiiden mallien hyödyntäminen ja siirto-opetus	Mahdollisuus käyttää ja siirto-opettaa omia tai valmiita malleja.
Näytönohjaimen hyödyntäminen	Mahdollisuus hyödyntää näytönohjaimia
Tulosten varmentaminen	Tulosten varmentamiseen erilaisia työkaluja. Ohjelman konsolista löytyy mallin tarkkuuteen ja opetuskertoihin liittyvät arvot. Arvot voidaan esittää myös kuvaajilla, vaatii ohjelmakoodin lisäämisen. Näytekuvat Heatmap-toiminnolla mahdollisia.
Uudet opetuskerrat ja tulosten vertailu	Malleja voi opettaa eri parametreilla. Mallien tallentaminen ja laataminen onnistuu. Ei graafista toimintoa eri parametreilla tehtyihin mallien eroihin.

## Muistin käyttö opetuksen yhteydessä

Mallin opettamiseen käytettyä muistia tarkasteltiin tietokoneen tehtävienhallinnan kautta. Mallin opettamisen aloitusvaiheessa tietokoneen keskusmuistin varaus oli 5,1 Gb. Opetusvaiheen aikana keskusmuistin varaus nousi n. 15,2 Gb asti, minkä lisäksi näytönohjaimen muistinkäyttö nousi lähelle 12 Gb:n enimmäismäärää. (Ks. kuvio 95)



Kuvio 95. Python-ympäristössä tehdyn koneoppimismallin muistinkäyttö opetuksen aikana.

## 6.2 MVTec Deep Learning Tool -koneoppimistyökalun testaus

MVTec Halcon on yksi suurimmista konenäkökirjastoista. Halcon-ympäristössä on mahdollista tehdä konenäkösovelluksia useilla ohjelmointikielillä, minkä lisäksi järjestelmästä löytyy erittäin laajat liityntärajapinnat erilaisille kameroiden liityntärajapinnoille (ks. kuvio 96). Lisäksi Halcon-työkalusta löytyy erittäin laaja otanta erilaisia koneoppimismenetelmiä, muun muassa (Halcon a product of MVTec 2023).

- Multi-Layer Perceptrons, MLP
- Support-Vector Machines, SVM
- Gaussian Mixture Models, GMM
- K-Nearest Neighbors (KNN)
- Deep Learning, DL
- Concolutional Neural Networks, CNN

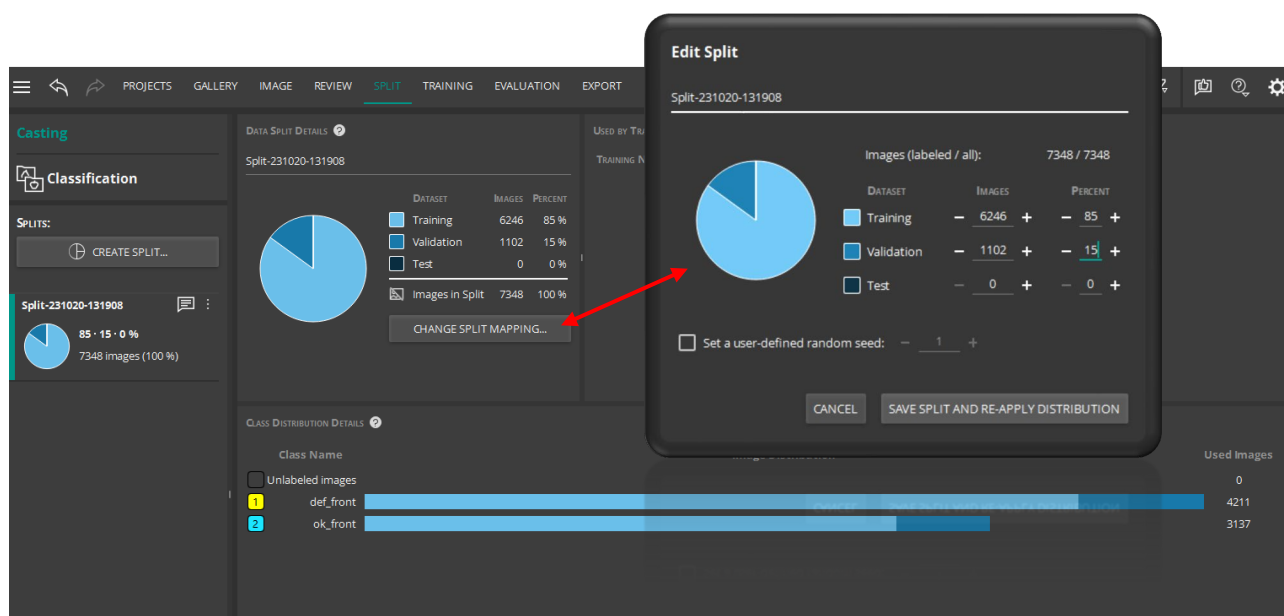
Tämän lisäksi Halcon-ympäristössä pystyy hyödyntämään muissa järjestelmissä tehtyjä koneoppimismalleja ONNX-muodossa.



Kuvio 96. Halcon ohjelmointikielien ja liityntärajapinnat (Halcon the power of machine vision 2015.)

MVTec Deep Learning Tool on erillinen ohjelma Halcon-konenäkösovelluksen rinnalle. Deep Learning Tool tarjoaa käyttäjälle visuaalisen käyttöliittymän mallien opettamiseen, ja ohjelman pystyy lataamaan ilmaiseksi MVTecin sivuilta. Mallien hyödyntäminen kuitenkin vaatii Halcon-ohjelmiston lisenssin. Asennustiedoston mukana tulee iso määrä erilaisia esimerkkiohjelmia sekä kuva-aineistoja mallien testaamiseen.

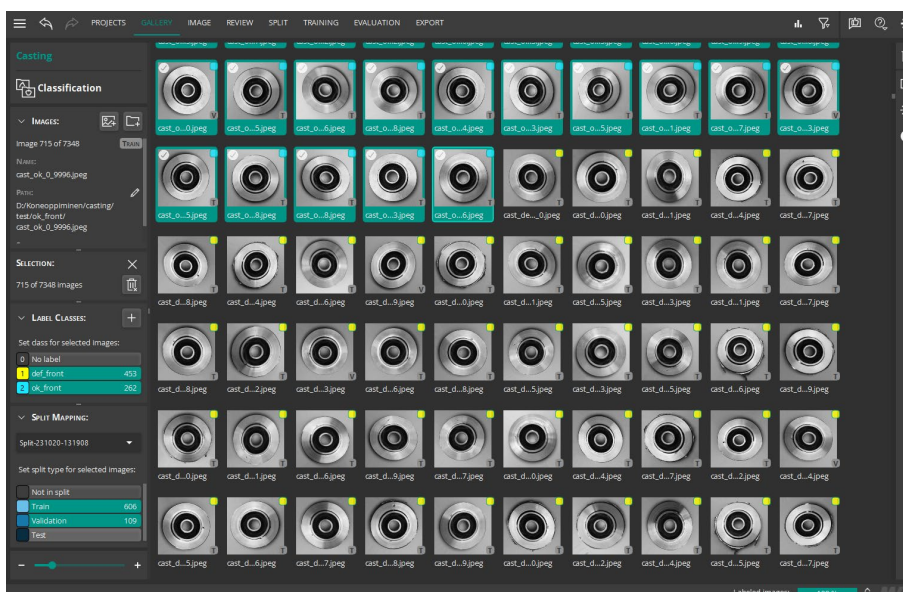
Deep Learning Tool toimii täysin visuaalisen käyttöliittymän avulla. Ensimmäisenä vaiheena on opetusaineiston valinta. Opetusaineiston valinnan voi tehdä valitsemalla kansion tai yksittäisiä kuvatiedostoja. Kansiorakenteen avulla kuvat voidaan tuoda kerralla ilman määrittelyitä tai vaihtoehtoisesti voi tehdä valinnan, että alikansioiden nimien mukaan määritetään eri luokat automaattisesti kuvatiedostoihin. Tästä on hyötyä varsinkin valmiiksi määritettyjen kuva-aineistojen kanssa. Aineiston jakaminen opetus-, validointi- ja testiaineistoon onnistuu helposti määrittämällä avautuvaan ikkunaan kuvien määrä tai prosenttiosuus. Tässä tapauksessa testiaineisto määritettiin nol-laksi, koska testiaineistoksi oli määritetty erillinen kuvasarja, mikä pysyy samana eri järjestelmätes-tauksissa. Valinnan jälkeen näkymästä voidaan tarkastella eri luokkien kuvamäärät ja tehdä tarvittaessa muutoksia. (Ks. kuvio 97)



Kuvio 97. Aineiston valinta Deep learning tool -työkalulla

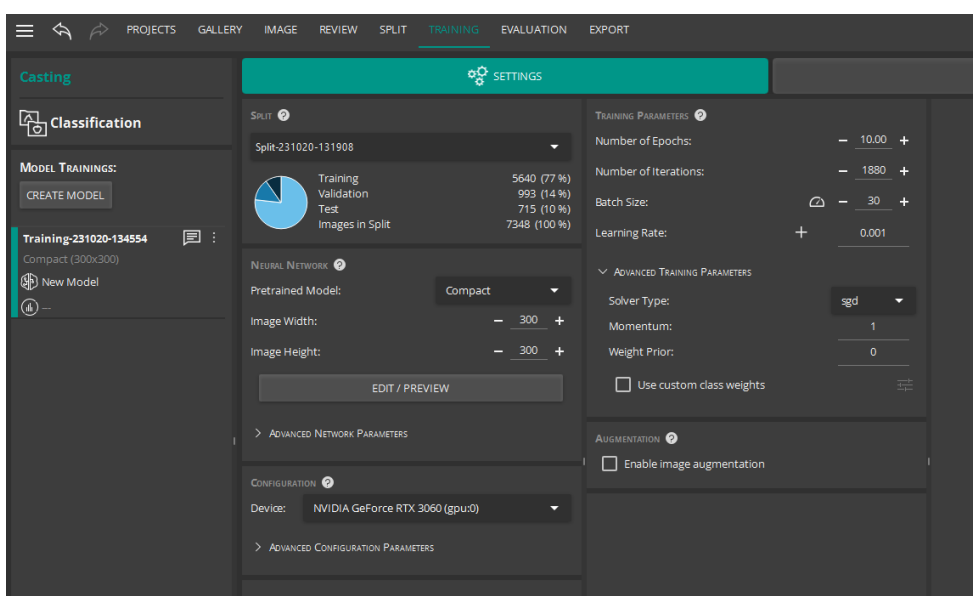
Gallery-välilehdeltä voidaan tarkastella kuva-aineistoa. Valituissa kuvissa on värikoodaus luokka-merkinnän mukaisesti. Yksittäisiä kuvia tai kuvasarjoja pystyy helposti valitsemaan toiseen luokkaan, jos merkintä on väärin. Tässä tapauksessa Testi-aineisto täytyi liittää erikseen toisesta kansio-osta ja merkata testiaineistoksi. (ks. kuvio 98)





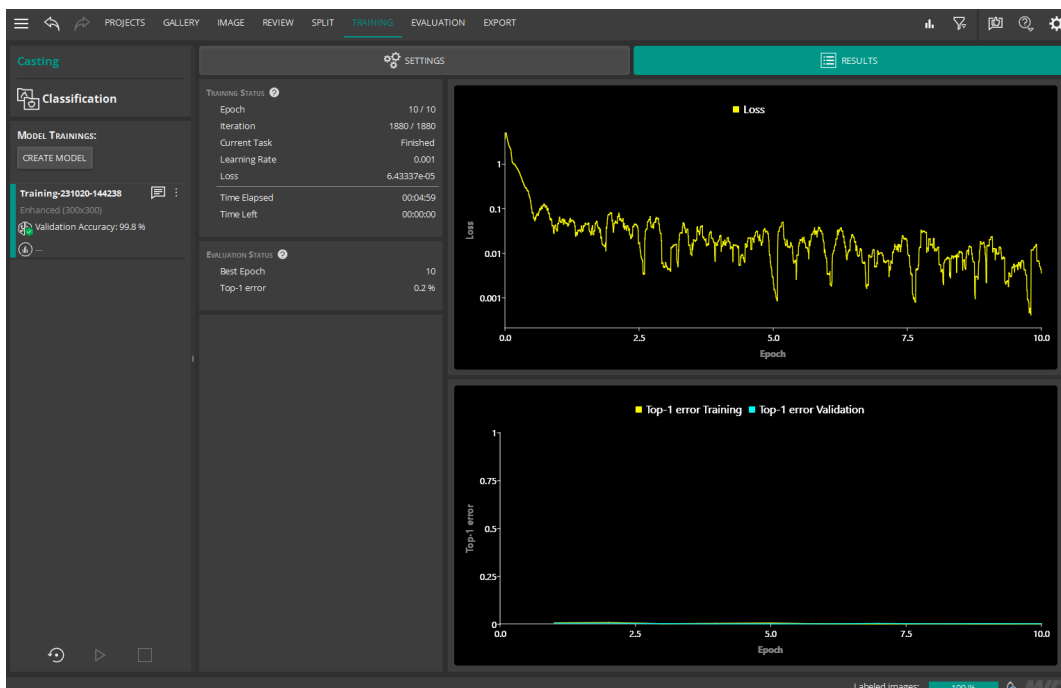
Kuvio 98. Opetusaineiston valitseminen ja tarkastaminen Deep learning tool -työkalulla.

Training-välilehdellä tehdään mallin rakenteen ja opetuksen määrittelyt. Mallille pystyy määrittämään perinteiset opetusparametrit, kuten Epoch- ja Batch size-määrät, Learning rate-arvon sekä optimointiparametrit. Malliin liittyen järjestelmästä löytyy erilaisia esiovetettuja malleja, mitä voidaan hyödyntää siirto-opettamalla, sekä syötteen tulevan kuvan koon määrittäminen. Lisäksi, jos ohjelma tunnistaa yhteensopivan näytönohjaimen, voi valinnan tehdä Device-valinnasta. (Ks. kuvio 99)



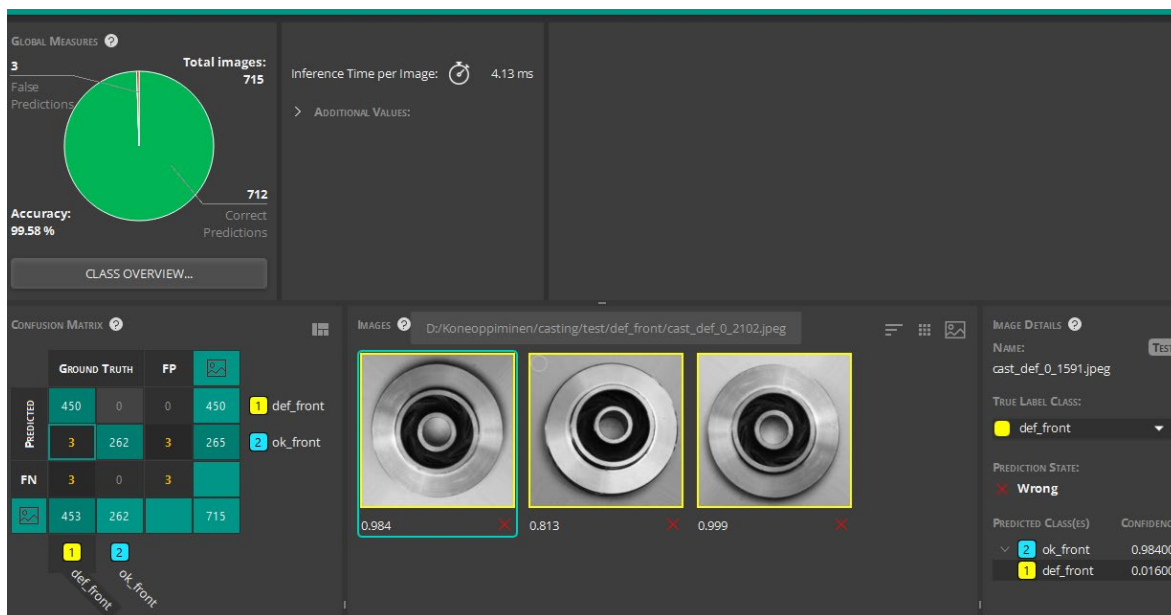
Kuvio 99. Koneoppimismallin määrittäminen Deep learning tool -työkalulla.

Ensimmäinen testaus tehtiin hyödyntämällä Compact -neuroverkkorakennetta, minkä arkkitehtuurista ei löytynyt tarkempia tietoja. Opetuksen aikana näkymään piirtyy Loss-funktion ja Top-error-arvojen kuvaajat. Opetuksen lopuksi järjestelmä näyttää myös Validointi-tuloksen, mikä tässä tapauksessa oli 99,8 %. (Ks. kuvio 100)



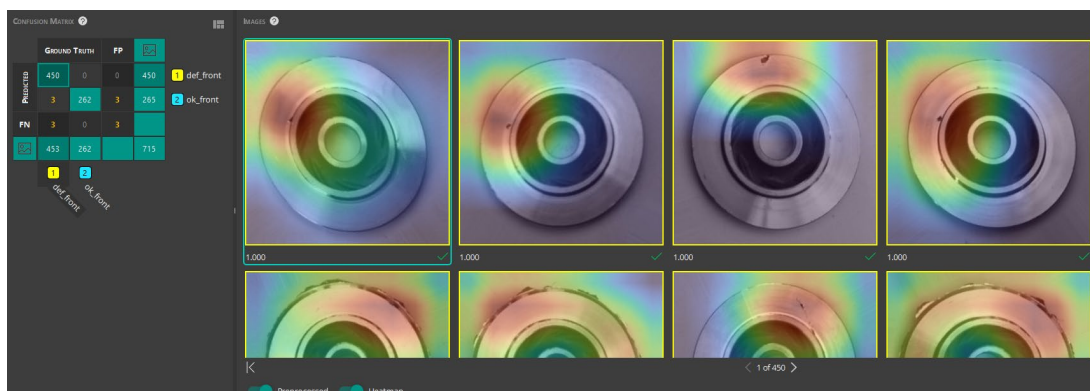
Kuvio 100. Mallin opetushistorian tarkastelu Deep learning tool -työkalulla

Erilliseltä Evaluation-välilehdeltä päästään katsomaan testiaineiston tulokset. Testi-aineiston tarkkuudeksi saatiin 99,58 %. Esimerkkitapauksessa vain kolme kuvaa meni false-negative-osioon. Valitsemalla osion Confusion matrix -kentästä pääsee tarkastelemaan väärin menneet kuvat. Myös oikeat solut voi valita, milloin järjestelmä näyttää kuvat. (Ks. kuvio 101)



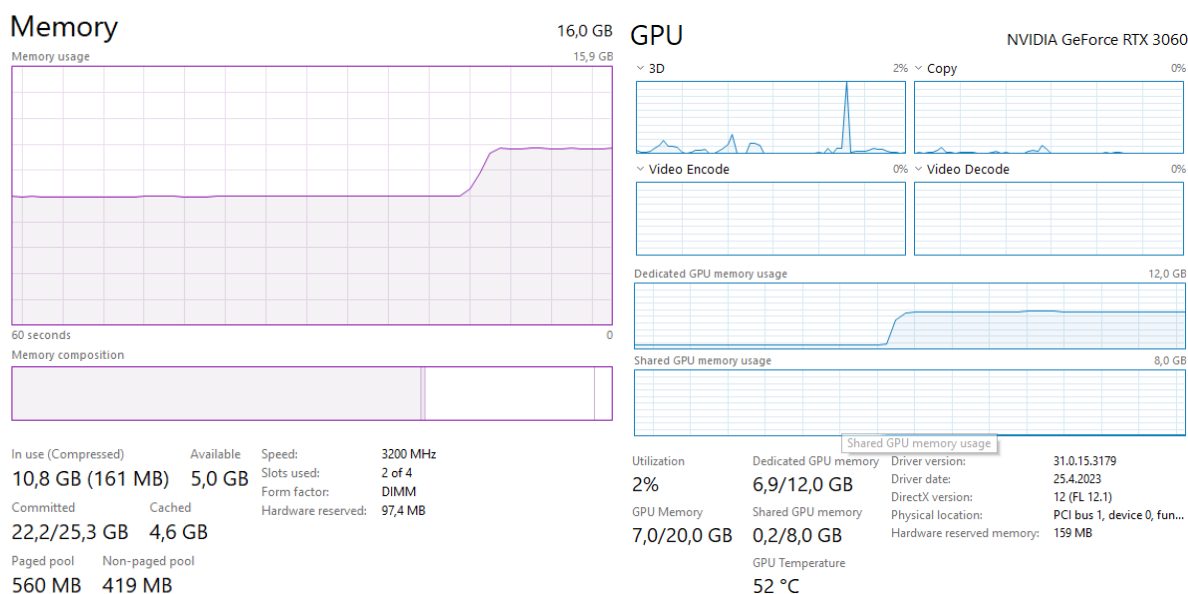
Kuvio 101. Tulosten varmentaminen Deep learning tool -työkalulla

Järjestelmästä löytyy myös Heatmap -toiminto, mikä korostaa kuvasta alueita, minkä perusteella malli on tehnyt päättelyn tietyistä luokasta. Värikoodaus ja korostus helpottavat mallin toiminnan tarkastelua. Vaikka opetus- tai testiaineistolla tulokset olisivat hyvät, mutta kuvista huomaisi, että järjestelmä on oppinut vääriä tai ylimääräisiä piirteitä, voidaan mallia opettaa uudestaan tai lisätä aineistoa opetusdataan. (Ks. kuvio 102)



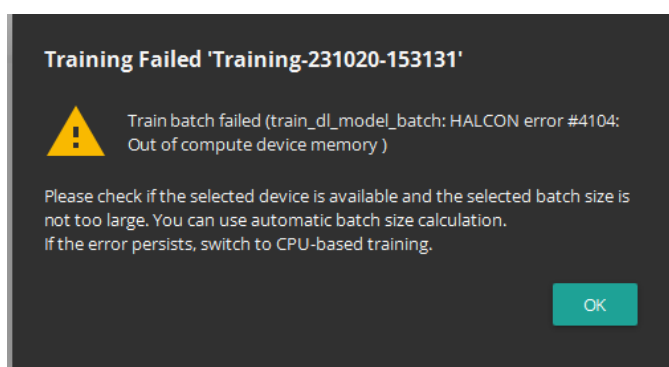
Kuvio 102. Heatmap-kuvien esittäminen Deep learning tool -työkalulla

Opetuksen aikana tietokoneen keskusmuistin käyttö nousi noin 3 Gb ja GPU:n muistinkäyttö nousi noin 6,5 Gb (ks. kuvio 103).



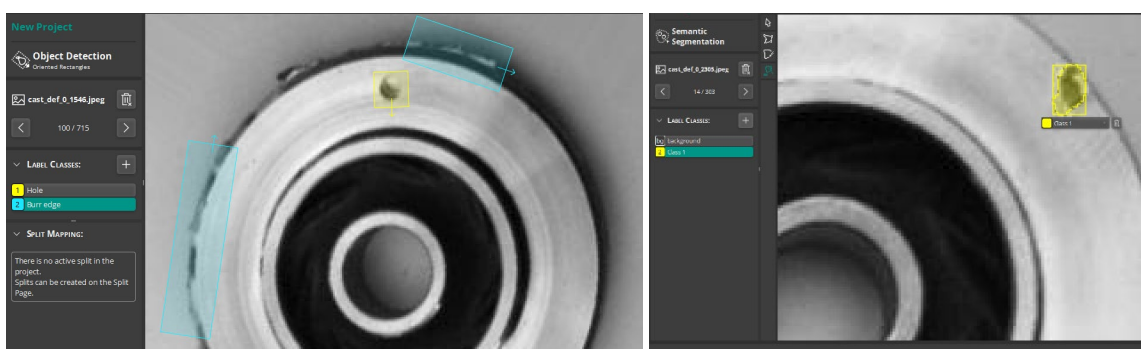
Kuvio 103. Deep learning tool:in muistinkäyttö mallin opettamisen aikana.

Raskaamman mallin yhteydessä havaittiin käytettävissä olevan raudan rajoitteet. Valitsemalla esi-opetetun ResNet-50-mallin ja määrittämällä Batch-kooksi 30, ei järjestelmä pystynyt tekemään mallin opetusta. Pienentämällä koon 15:ksi, pystyi mallin kuitenkin opettamaan. (Ks. kuvio 104)



Kuvio 104. Muistinkäyttöraja ResNet-50-mallin kanssa.

Deep Learning Toolin avulla pystyy tekemään ohjattua oppimista myös erilaisilla kuvien merkkautavoilla tai aluevalinnoilla. Graafinen käyttöliittymä ja aineiston muokkaaminen kuvioiden ja alueiden piirtotoiminnolla toimii hyvin, eikä erillisiä ohjelmistoja aineiston käsittelyyn tarvita. Kuva-alueelta voidaan tehdä manuaalisen hiirellä piirtämisen ja laatikoinen valitsemisen lisäksi tunnustuksia esimerkiksi automaattitoiminnon avulla. Automaattitoiminnossa alueen ympärille rajataan laatikko ja valitaan hiirellä laatikon sisälle rajattua ominaisuutta. Ohjelma laskee automaattisesti muotoihin ja väreihin perustuen, mikä haluttu kuva-alue voisi olla. Uudella hiiren painalluksella alue rajautuu uudestaan. Tällä tavalla saadaan tarkemmin rajattua tiettyjä ominaisuuksia, eikä opetetava ominaisuus ole vain osa valitusta alueesta, kun verrataan neliön piirtoon tunnistettavan kohteen ympärille. (Ks. kuvio 105)



Kuvio 105. Deep learning tool muodon valinta ja segmentointityökalut

### Yhteenveto MVTec Deep Learning tool -ympäristön testauksesta

Järjestelmän asentaminen on helppoa ja tarvittavat aineistot tulevat asennustiedoston mukana. Myös näytönohjaimen hyödyntämiseen tarvittavat lisätoiminnot on sisällytetty pakettiin. Ohjelman käyttäminen on helppoa graafisen käyttöliittymän avulla. Ohjelmaan on myös tarjolla koulutusmateriaalia ja esimerkkiaineistoa, minkä avulla työkalua pystyy käyttämään.

Aineiston valinta ja luokittelu on helppoa ja intuitiivista käyttöliittymän avulla. Luokkien luonti kansiorakenteen perusteella nopeuttaa toimintoja, mutta tarvittaessa kuvien luokkamerkintöjä on helppo vaihtaa yksitellen tai ryhmissä. Myös erilaiset opetusvaihtoehdot onnistuvat kuvista vapaan piirtotoiminnon ja automaattivalintojen avulla.

Koneoppimismallien rakenteiden valinta ei ole niin vapaamuotoista, kuin avoimen lähdekoodin järjestelmissä. Halcon-työkalun puolella on käytössä laajempi otanta erilaisista algoritmeista. Deep Learning Tool -työkalusta löytyy valmiita esiopetettuja malleja mitä siirto-opetetaan soveltumaan uusille kuville. Opetukseen liittyviä parametreja kuitenkin pystyy muokkaamaan.

Tulosten esittäminen on selkeää ja työkalusta löytyy valmiina kuvaajien ja graafien piirto. Kuvien automaattinen näyttö tunnistetiedon ja Heatmap -toiminnon kanssa helpottaa mallin toiminnan tarkastelua.

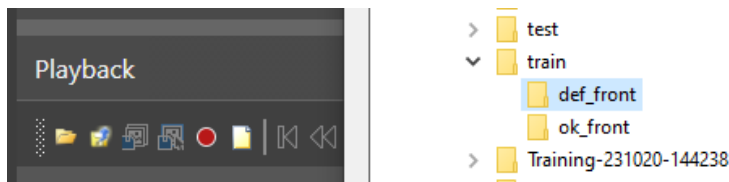
Taulukko 9. MVTEC Deep Learning Tool testaustulokset

MVTEC Deep Learning Tool	
Asennus ja tuki	Asennustiedosto ladattavissa ilmaiseksi MVTECin sivuilta. MVTEC tuki, Suomessa jälleenmyyjänä toimii Stemmer Imaging.
Aineiston valinta ja luokittelu	Kuvat voidaan tuoda suoraan kansioista tai luokitella työkalusta löytyvällä toiminnolla. Kuvien luokittelua helppo vaihtaa hiirellä napauttamalla. Aineisto selkeästi näkyvissä ja värikoodattu. Opetus-, validointi- ja testiaineiston jakaminen onnistuu graafisen valikon kautta.
Millaisia koneoppimismalleja voi hyödyntää	Erittäin monipuolinen ja tehokas tapa tehdä erilaisia malleja. Deep Learning Tool:in puolella jonkin verran rajoituksia.
Valmiiden mallien hyödyntäminen ja siirto-opetus	Mahdollisuus hyödyntää valmiita malleja ja siirto-oppimista.
Näytönohjaimen hyödyntäminen	Mahdollisuus hyödyntää näytönohjaimia.
Tulosten varmentaminen	Tulosten esittäminen kuvaajien avulla. Kuvaajat, confusion matrix -taulukot ja kuvien esitys automaattisesti. Näytekuvat Heatmap-toiminnolla automaattisesti.
Uudet opetuskerrat ja tulosten vertailu	Eri opetuskerrat jäävät ohjelmaan muistiin. Helppo tehdä vertailua eri parametreilla tehtyjen mallien välillä.

### 6.3 Cognex VisionSuite -ohjelmiston testaus

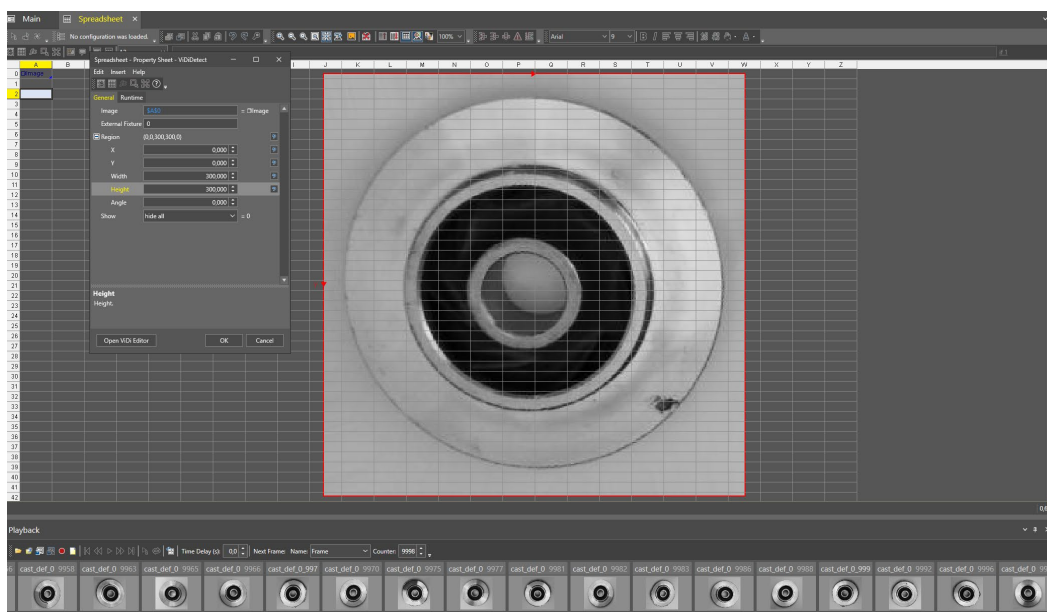
Cognexin koneoppimisjärjestelmä ViDi toimii joko In-Sight Vision Suite- tai VisionPro -ohjelmistolla. VisionPro on enemmän tietokonepohjainen konenäköjärjestelmä, missä kameran tuottama kuva käsitellään ja analysoidaan tietokoneella. VisionSuite-ohjelma toimii Cognexin älykameroiden ohjelmointiin. Tähän selvitykseen valittiin VisionSuite-ohjelmisto Jamkilla olevan Cognex D905 -älykameran takia. Ohjelmiston käyttö onnistuu myös valmiilla kuva-aineistolla, eikä älykameraa ole pakollista käyttää, mikä mahdollisti järjestelmän testauksen samalla kuva-aineistolla muiden järjestelmien kanssa.

VisionSuite-ohjelmaan pystyy lisäämään valmiita kuvasarjoja Playback-valinnan kautta. Saman toiminnon kautta voitaisiin kerätä kuva-aineistoa kameralta ja tallentaa se haluttuun kansioon. Kansioiksi valittiin ensin opetusaineiston vialliset kuvat. (Ks. kuvio 106)



Kuvio 106. Aineiston valinta ViDI-työkalulla

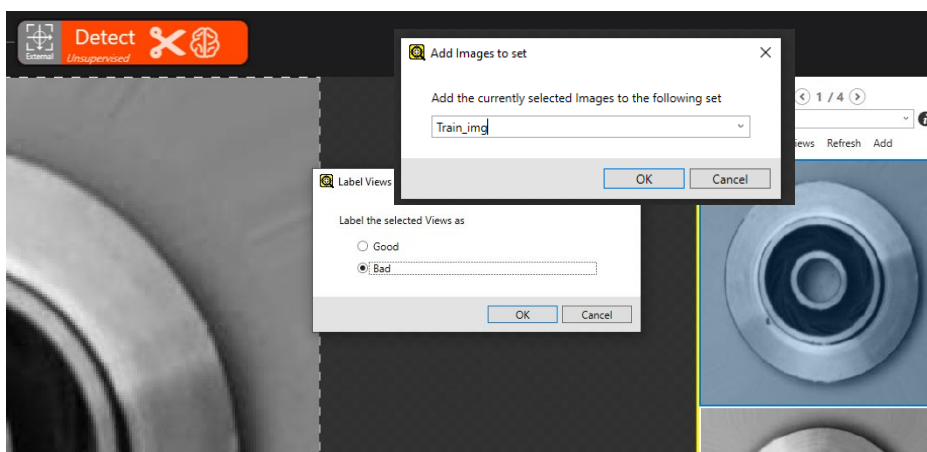
Kuvasarja tulee näkymän alareunaan ja yksittäisiä kuvia pystyy tarkastelemaan valitsemalla haluttu kuva hiirellä aktiiviseksi (ks. kuvio 107).



Kuvio 107. Valitun ainesiton tarkastelu ViDI-työkalulla.

Ohjelmasta löytyy erilaisia toimintoja mallin opettamiselle. Tässä tapauksessa testattiin samankaltaista ohjatun oppimisen menetelmää koko kuvasta, mitä muillakin järjestelmillä testattiin. Cognexin Detect-toiminnolla on sekä unsupervised että supervised ominaisuus. Unsupervised-toimintokaan ei ilmeisesti ole täysin aikaisemmin esitellyn termin mukainen, koska mallille täytyy antaa myös huonoja kuvia. Mallin toiminta ei avaudu käytettävissä olevista materiaaleista, joten tälle

kuvaukselle ei ole täyttä varmuutta. Toiminnolle voidaan kuitenkin lisätä opetettavat kuvat hyvinä ja huonoina näyteinä testiaineistoon. Ensimmäisenä lisätyt kuvat Playback-näkymästä voidaan raahata työkalulle ja määrittää mihin luokkaan kuvat määritetään. Luokaksi voi valita järjestelmästä löytyvät Good/Bad, tai luokkia voi lisätä ja nimetä vapaasti. Kaikki casting-kuvasarjan kuvat lisättiin erikseen opetus- ja testiaineistoihin. (Ks. kuvio 108)



Kuvio 108. Kuvien luokkamäärittely ViDI-työkalulla.

Mallin opetusparametreihin määritellään Epoch-sarjojen määrä, sekä pienimmän tunnistettavan ominaisuuden koko. Ohjelmasta löytyy myös toiminto tunnistettavien kohteiden häiriöiden määrittelyä varten. Tunnistettaville kohteille voidaan määrittää erilaisia kiertymään ja kokoon liittyviä parametreja, millä mallin tunnistustarkkuutta voidaan parantaa, jos tuotteessa esiintyisi variaatioita. Käsiteltävässä aineistossa ei kuitenkaan tällaisia kuvia ollut. Opetusaineistoksi valittiin aikaisemmin lisätty Train\_img-kuva-aineisto, ominaisuuden kooksi valittiin 13.1 pikseliä ja opetus-kertojen määräksi 10. (Ks. kuvio 109)

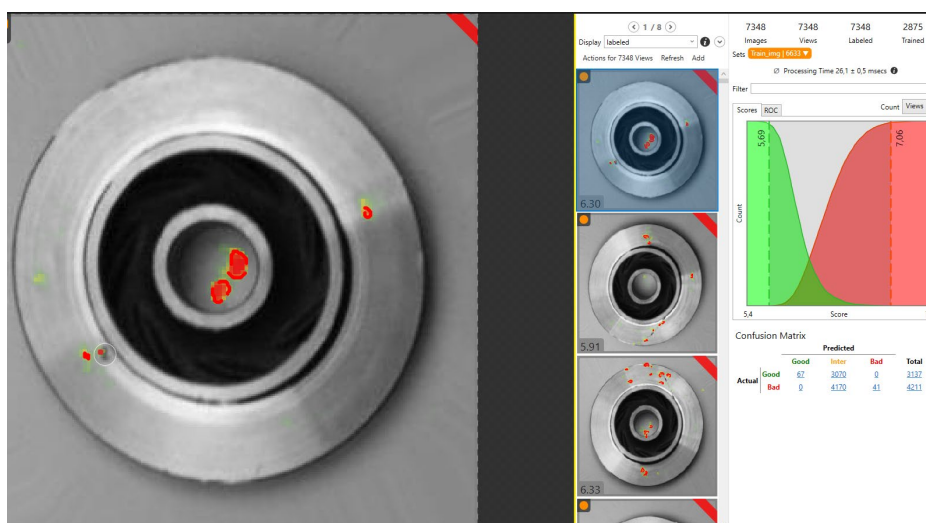




Kuvio 109. Opetusparametrien määrittäminen ViDi-työkalulla.

Opetuksen jälkeen ohjelma näyttää opetus ja testiaineiston kuvat heatmap-tunnisteiden kanssa. Ohjelma näyttää myös confusion matrix -taulukon, sekä kuvaajat eri luokkien tunnistamisesta.

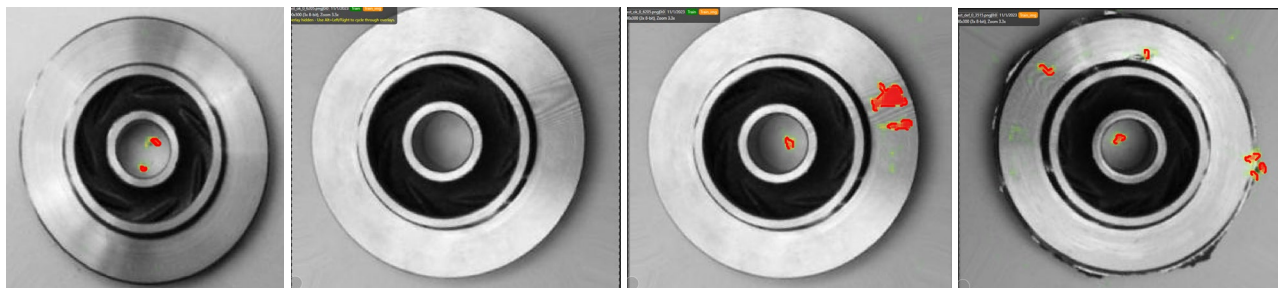
Tunnistuksen raja-arvot on säädetty arvoihin 5,69 ja 7,06. Tässä tapauksessa alueet menevät erittäin paljon päällekkäin ja confusion matrix taulukon Inter-kohdasta nähdään, että erittäin iso osa näytteistä on alueella, missä tunnistustarkkuus ei ole riittävä. (Ks. kuvio 110).



Kuvio 110. Tulsoten tarkastelu ViDi-työkalulla.

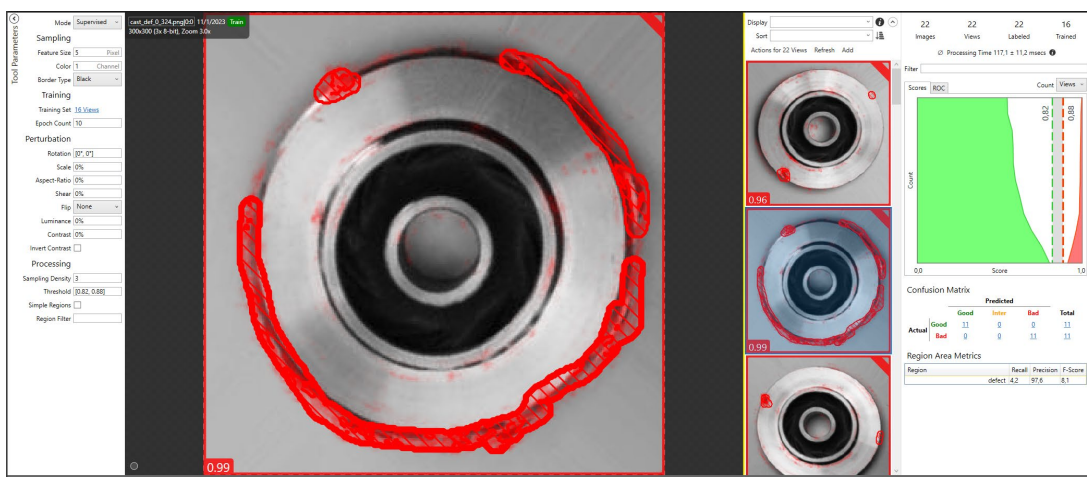
Muuttamalla rajojen arvoja, saadaan mallin tarkkuus paremmaksi Inter-osion osalta. Tämä ei kuitenkaan poista päällekkäisyyksiä tunnistuksessa. Kuva-aineisto ja opetusparametrit eivät tässä tapauksessa sovellu yhteen.

Kuviossa 111 on tuloksia tunnistetuista kohteista. Ensimmäisessä kuvassa malli on tunnistanut kappaleen keskireiän kohdalta muotoja, mitkä eivät vaikuta millään tavalla tuotteen laatuun. Toinen ja kolmas kuva ovat samasta hyvästä tuotteesta mistä nähdään, että kuvasta on tunnistettu heijastumasta ja kuvan skaalauksesta johtuvia muotoja tuotteen pinnasta. Neljännessä kuvassa heatmap-toiminto näyttää, miten huonostakin tuotteesta on tunnistettu ominaisuuksia keskireiän kohdalta, sekä ehjistä pinnasta tulevista heijastuksista. Kuvasta on tunnistettu myös osittain oikeita kohteita. Opetettu malli siis ei toimi kyseiselle kappaleelle tällä kuva-aineistolla.



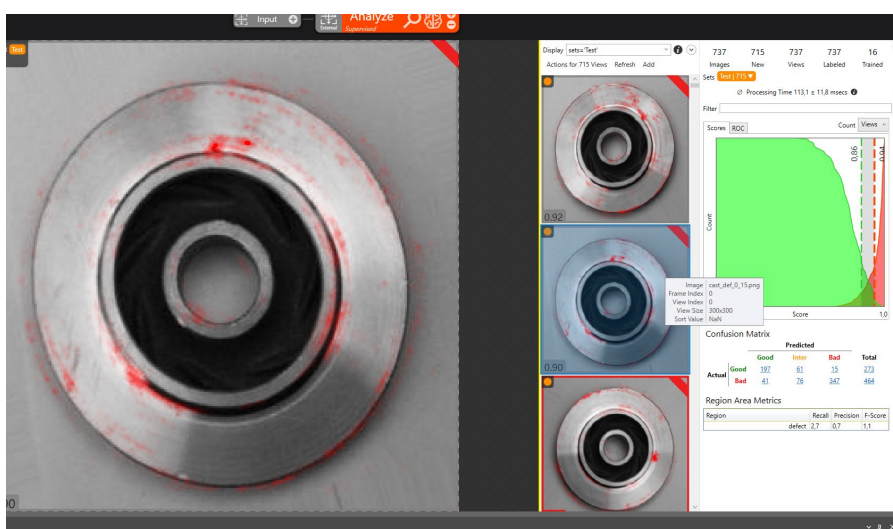
Kuvio 111. Koneoppimismallin tulosten tarkastelu ViDI-työkalulla

Myöskään parametreja muokkaamalla mallin tunnistuksesta ei tullut riittävä, vaan näytteet olivat samankaltaisia ja kuvaajien arvot menivät päällekkäin. Ohjelmaa testattiin myös supervised-toiminnalla, missä hiirellä piirtämällä pystyy merkkamaan haluttuja kohteita kuvista, mitkä koneoppimismalli pyrkii tunnistamaan virheellisiksi ominaisuuksiksi. Toimintaa varten tehtiin uusi projekti uudella kuva-aineistolla. Opetusaineistoksi valittiin 11 huonoa ja 11 hyvää kuvaa. Huonoihin kuviin merkattiin tunnistettavia kohteita hiiren kursorilla piirtäen. Tällä tavalla kuvasta pystyy merkkamaan yhden tai useamman alueen. Merkkamalla alueita, malli ei pyri tunnistamaan esimerkiksi kappaleen keskireiän ominaisuuksia. Opetuksen jälkeen mallin tarkkuus ja kuvaajat vaikuttavat huomattavasti paremmilta. (Ks. kuvio 112)



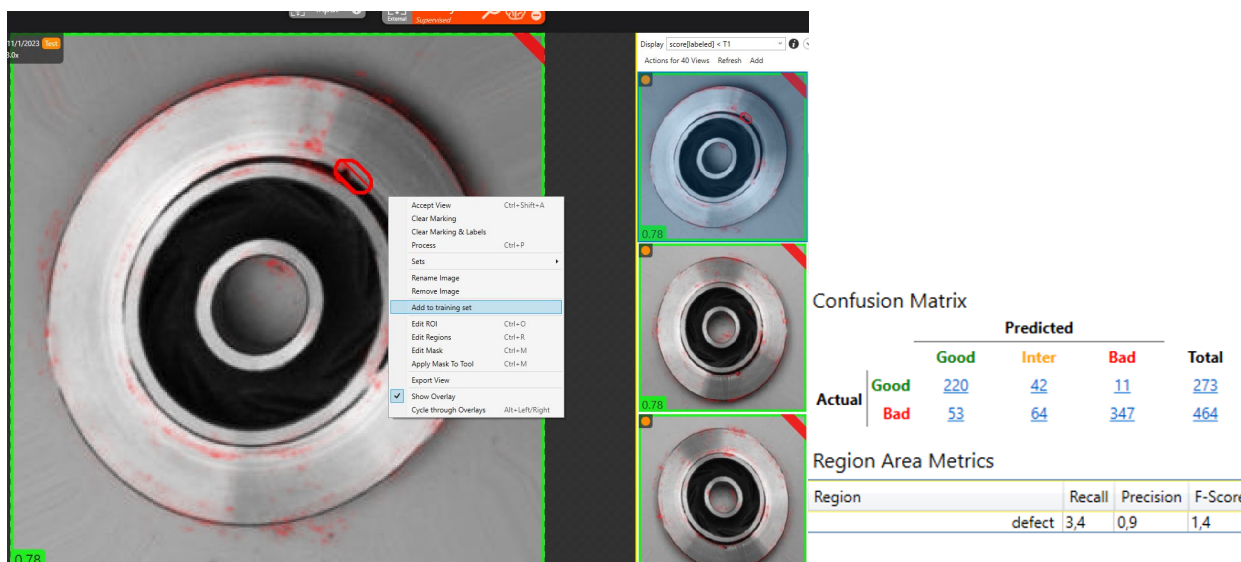
Kuvio 112. ViDi-työkalun supervised-toiminto.

Kuva-aineistoon lisättiin näytteet testiaineistosta, minkä jälkeen kuville tehtiin tunnistus opetetun mallin perusteella. Mallin tarkkuus oli parempi kuin unsupervised-toiminnolla, mutta ei kuitenkaan vielä kovin hyvä. (Ks. kuvio 113)



Kuvio 113. Testiainesiton analysointi ViDi-työkalulla

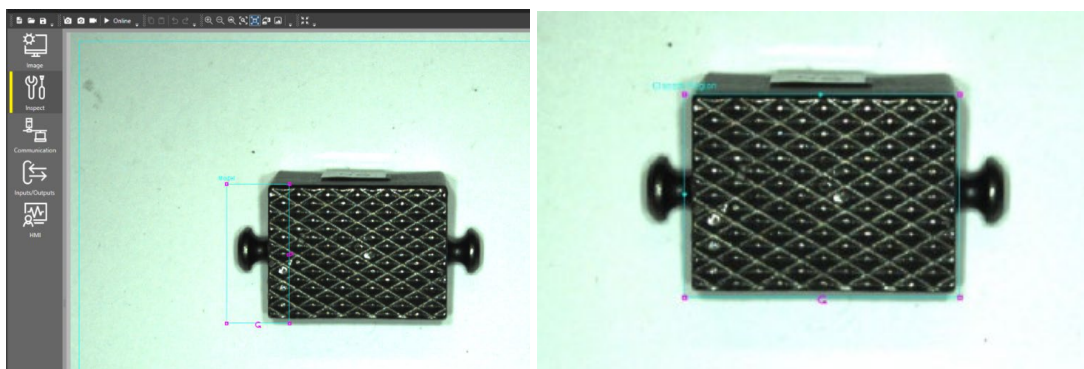
Kuvia läpikäymällä ja keskittymällä varsinkin väriin arvauksiin, voidaan mallin opetusaineistoa kasvattaa. Kuvaa voidaan muokata piirtämällä tunnistettava alue kuvaan, minkä jälkeen kuva voidaan lisätä opetusaineistoksi. Mallin tarkkuus saatiin muutettu arvosta 0,7 arvoon 0,9 lisäämällä muutama lisäkuva opetusaineistoon. (Ks. kuvio 114)



Kuvio 114. Mallin opetusaieston kasvattaminen ViDi-työkalulla.

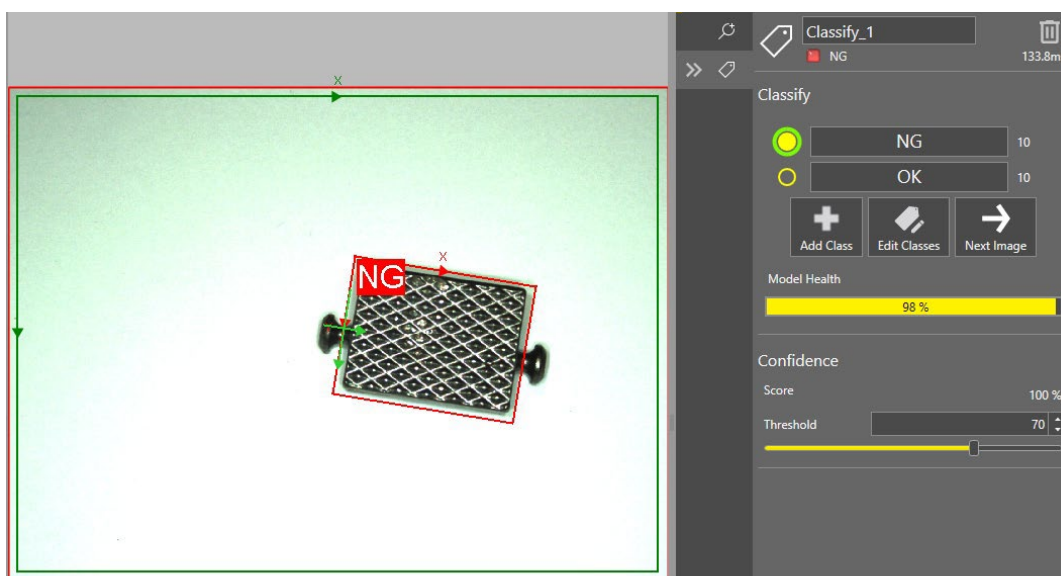
ViDi-työkalusta löytyy myös helpokäyttötoiminto luokittelutehtäviin. Esimerkiksi IS2800-älykameralla pystyy tekemään kevyitä koneoppimismalleja, missä laskenta tapahtuu suoraan kameralla. Cognex kutsuu näitä kameroita EdgeAI-termillä. Kameran käyttämiseen on erillinen EasyBuilder-näkymä, missä kameran asetukset ja ohjelma pystytään tekemään. Kamerasta löytyy luokittelutoiminto, mikä hyödyntää esiopetettua koneoppimismallia, mikä siirto-opetetaan uusilla kuvilla. Kuvien valinta onnistuu helposti käyttöliittymän kautta. Järjestelmän toimintaa testattiin kiinnityspalojen pintavirheiden tunnistamiseen. Opetuskuvat otettiin suoraan käytössä olleella IS2800-kameralla.

Tuotteeseen voidaan määrittää kiintopiste esimerkiksi Pattern-toiminnolla. Tämän kiintopisteen perusteella valitaan kuva alue, mikä annetaan syötteenä koneoppimismallille. Esimerkissä tavoitteena on selvittää kiinnityskappaleen pintavirheet. Toimenpide on perinteiselle sääntöpohjaiselle ratkaisulle erittäin haastava vaihtelevasta muodosta ja valaistuksesta johtuen. Ensin muodolle valitaan kiintopisteeksi reunan tunnistus, minkä perusteella voidaan rajata haluttu kuva-alue mallia varten. (Ks. kuvio 115)



Kuvio 115. ViDi EasyBuilder työkalun pattern- ja syötekuva valinta.

Ohjelman Classify-toiminnon kautta voidaan ottaa uusia kuvia tuotteista ja valita helposti hiirellä mihin luokkaan kuva kuuluu. Järjestelmässä on vakiona kaksi luokkaa NG ja OK, mutta ne voidaan myös nimetä haluamallaan tavalla. Model Health -arvo ilmoittaa mallin tarkkuuden ja sen avulla voidaan päätellä milloin riittävä määrä kuvion saatu kerättyä. (Ks. kuvio 116)



Kuvio 116. Opetusaineiston valinta EasyBuilder-näkymässä

Kuvasarjaa pääsee myös tutkimaan ja tarvittaessa kuvien paikkoja voi vaihtaa eri luokkien välillä, tai virheelliset kuvat voi poistaa. Mallin opettaminen tietyn tyyppisiin toimenpiteisiin onnistuu nopeasti ja helposti, jopa vain muutamilla kuvilla. Toiminnolla ei ehkä saa selville kaikista haastavimpia kohteita, mihin tarvitaan syviä konvoluutioneuroverkkoja, mutta toiminto mahdollistaa usean sääntöpohjaisen toimenpiteen korvaamisen yhdellä opetustoiminnolla. (Ks. kuvio 117)





Kuvio 117. Opetusaineiston tarkastelu EasyBuilder-näkymässä.

## Yhteenveto Cognex ViDi -ympäristön testauksesta

Järjestelmän asentaminen onnistuu helposti yhdellä asennustiedostolla. Näytönohjaimen hyödyntämiseen tarvittavat toiminnot on sisällytetty pakettiin. Ohjelman käyttäminen on helppoa graafisen käyttöliittymän avulla. Cognexin sivuilta on saatavilla opetusmateriaalia ja ohjeita järjestelmän käyttöön.

Aineiston valinnan pystyy tekemään valmiiksi tallennetuista kuvista, tai ottamaan uudet opetuskuvat kameralla. Kuvien luokkien määrittely onnistuu graafisesti, mutta alkuun valinta saattaa olla hieman haastavaa. Luokiteltujen kuvien uudestaan määrittely isompina erinä on haastavaa.

Koneoppimismallien hyödynnettävyys on rajoitettu. Käytettävissä on vain Cognexin määrittelemät koneoppimismallit, eikä niiden rakennetta ole kerrottu. Opetusvaiheessa pystyy määrittelemään tiettyjä parametreja mallin tuloksen parantamiseksi.

Järjestelmä tunnistaa automaattisesti yhteensopivat näytönohjaimet. Lisäksi järjestelmästä löytyy palvelinkäyttömahdollisuus, eli mallien laskennan voi suorittaa myös etänä laskentapalvelimella. Tällainen toiminto vaatii erillisen lisenssin.

Mallin testaus ja tulosten esittäminen toimii automaattisesti kuvaajien ja kuvien heatmap-esitysten avulla. Graafinen kuva auttaa hahmottamaan tunnistusten onnistumista ja määrittämään tunnistuksen raja-arvot, eri luokille. Mallia on helppo opettaa uudestaan merkkamalla edellisen opetuskerran tuloksista kuvia opetusaineistoon.

Taulukko 10. Cognex VisionSuite testaustulokset

Cognex VisionSuite ViDi	
Asennus ja tuki	Asennustiedot ladattavissa Cognexin sivuilta. Vaatii toimiakseen lisenssin. Paljon ohjeistusta ja materiaalia saatavilla kirjautumalla sivustolle. Valmistajan tuki, Suomessa jälleenmyyjänä toimii TR-Electronic
Aineiston valinta ja luokittelu	Aineiston tuonti kansioista onnistuu ja luokittelun voi tehdä yksitellen tai ryhmissä. Erilaisia valintamenetelmiä merkattuihin tai merkkamattomiin kuviin.
Millaisia koneoppimismalleja voi hyödyntää	Koneoppimismallit Cognexin määrittämiä. Mallien rakennetta ei pysty muokkaamaan. Opetusparametreja pystyy muokkaamaan ja eri opetusmenetelmät käytössä (ohjaamaton ja ohjattu)
Valmiiden mallien hyödyntäminen ja siirto-opetus	Vain Cognexin mallit.
Näytönohjaimen hyödyntäminen	Mahdollisuus hyödyntää näytönohjaimia, sekä etäpalvelintä (vaatii erillisen lisenssin).
Tulosten varmentaminen	Tulosten esittäminen kuvaajien avulla. Kuvaajat, confusion matrix -taulukot ja kuvien esitys automaattisesti. Näytekuvat Heatmap-toiminnolla automaattisesti.
Uudet opetuskerrat ja tulosten vertailu	Mallia voi opettaa helposti uudella aineistolla. Uusien kuvien lisääminen opetusaineistoon onnistuu helposti. Mallin opetus työkalukohtainen -> uuden opetuC6:D35ksen voi tehdä valitsemalla uuden työkalun.

## 6.4 Järjestelmätestausten yhteenveto

Avoimen lähdekoodin ja kaupallisten toimijoiden sovelluksista löytyy erittäin paljon samankaltaisuuksia. Python-ympäristöä on hyödynnetty jo pitkään pitkä data-analytiikassa ja koneoppimisessa niin tutkijoiden ja sovelluskehittäjien, kuin myös harrastajien toimesta. Tämä on lisännyt käytettävissä olevia ominaisuuksia, mitä löytyykin varsin laajasti. Käytettävissä olevien kirjastojen ja toimintojen kirjo on erittäin laaja, mikä tuo oman haasteen järjestelmän käyttöön. Kirjastojen ja toimintojen tuntemus täytyy olla hyvällä tasolla, että järjestelmiä pystyy hyödyntämään. Toinen haaste avoimen lähdekoodin järjestelmiin on kirjastojen ja pakettien päivittyminen eri tahtiin, eikä niiden välistä toimintaa aina varmenneta. Tästä aiheutuu haasteita uusien asennusten yhteydessä, eikä ohjelmat välttämättä toimi samalla tavalla, kuin jollakin toisella asennuksella. Avoimen lähdekoodin haasteita aineiston merkinnöissä ja luokittelussa on pyritty ratkaisemaan kaupallisissa järjestelmissä. Toisaalta kaupallisiin järjestelmiin on pyritty tuomaan samoja ominaisuuksia, mitä Python-ympäristössä ja data-analytiikassa hyödynnetään tulosten esittämiseen. Kaupallisista järjestelmistä löytyy yleisesti graafiset käyttöliittymät, mitkä helpottavat ohjelmiston käyttöä ja aineiston luokittelua. Lisäksi järjestelmistä löytyy ilman erillistä asennusta ja toimenpiteitä hyvät toiminnot mallin tarkkuuden ja tulosten varmentamiseen.

Avoimen lähdekoodin mallien hyödyntämismahdollisuudet ovat lähes rajattomat. Malleja pystyy määrittelemään itse tai hyödyntämään internetistä valmiiksi opetettuja malleja. Lisäksi mallien muokkaaminen eri järjestelmiin onnistuu yleisesti avoimen lähdekoodin ympäristöissä. Kaupallisissa järjestelmissä mallit saattavat olla rajoittuneempia, mutta toisaalta sovitettuja kuvankäsittelyyn liittyviin ongelmiin.

Erilaiset helppokäyttötoiminnot mahdollistavat koneoppimisen hyödyntämisen ilman tarkempaa tietämystä koneoppimiseen liittyvistä algoritmeista ja toiminnoista. Tässä tapauksessa rajoittuneemmat ja konenäköönliittyvät toiminnot auttavat sopivan toiminnon tekemisessä. Helppokäyttötoimintojen, kuten aineiston luokittelun, ohjatun oppimisen hyödyntäminen piirtotyökaluilla tai yksinkertaiset luokittelutyökalut valmiiksi opetetuilla malleilla nopeuttavat sovelluksen tekemistä. Perinteisten sääntöpohjaisten konenäkötoimintojen rakentaminen toimivaksi vaatii myös tarkkaa tietämystä järjestelmän toiminnasta. Helppokäyttöisen koneoppimistyökalun avulla tuotteiden luokittelu tai pintavirheiden tunnistaminen onnistuu ilman erillistä tietämystä, mitä järjestelmässä oikeasti tapahtuu.

## **7 Kyselytutkimuksen tulokset**

Kyselytutkimus toteutettiin Jyväskylän ammattikorkeakoulun tarjoamaan Webropol -palvelun kautta. Tulosten käsittelyssä ja arvioinnissa hyödynnettiin Webropolista löytyviä aineistonhallinta- ja raportointityökaluja. Vastaajiksi valittiin Suomessa toimivia yrityksiä, joissa konenäköä hyödynnetään ja Webropol-kyselyn linkkiä lähetettiin vastaajille sähköpostin kautta.

Vastaukset olivat anonyymejä, joten saateviestissä ilmoitettiin, että kyselyä voi jakaa myös eteenpäin. Tästä huolimatta kyselyn vastausmäärä jäi hyvin alhaiseksi, vain seitsemän vastausta. Yhtenä selittäväenä tekijänä saattoi olla koneoppimisen hyödyntämismahdollisuuksien vähäinen tuntemus yrityskentässä, mutta toisaalta kyselytutkimusta ei jätetty kesken vastaamisen aloittamisen jälkeen yhtään kappaletta. Kyselyyn vastauksen aloittaneet siis tekivät kyselyn myös loppuun asti. Kyselyssä käytettiin sekä valmiiksi strukturoituja kysymyksiä että avoimia kysymyksiä. Kyselytutkimuksen kysymykset ja vastaukset on esitetty kokonaisuutena liitteessä 1.



Vastaajat jakautuivat hieman eri sektoreille. Vastaajissa oli mukana laitetoimittajia, koneenrakentajia, kuin myös konenäköjärjestelmiä toteuttavia tahoja. Yrityskentän koko vaihteli pienistä muutamien kymmenen hengen yrityksistä aina yli 10000 henkilöä työllistäviin suuryrityksiin. Yhdessä vastauksessa oli kuitenkin hyvä tarkenne, AI-puolen henkilömäärä suuressa yrityksessä oli kuitenkin vain 20–30 henkilöä. Kokemusta konenäköjärjestelmistä vastaajilla oli hyvin. Yli puolella vastaajista oli kokemusta yli 8 vuotta ja loppuillakin 3–5 vuotta.

Kysymyksessä neljä, kartoitettiin vastaajien kokemuksia koneoppimista hyödyntävistä järjestelmistä suhteessa avoimen lähdekoodin ja kaupallisiin järjestelmiin. Vastaajilla oli vaihtelevasti kokemusta erilaisista järjestelmistä. Osalla vastaajista oli laajemmin kokemusta useista järjestelmistä, mutta oli myös vastaajia, joilla kokemusta vain avoimen lähdekoodin järjestelmistä tai ei juurikaan kokemusta.

Kysymyksessä viisi selvitettiin perinteisten konenäkömenetelmien tulosten varmentamista suhteessa koneoppimisessa hyödynnettävään testi- ja validointiaineistojen käyttämiseen. Vastausten perusteella perinteisiä konenäköjärjestelmiä varmennetaan vaihtelevalla määrällä aineistoa. Kysymyksen asettelu ei ota kantaa, millainen konenäkötoiminto on kyseessä, mikä voi vaikuttaa vastauksiksiin. Vastauksissa kuitenkin puhutaan muutamista kuvista aina tuhansiin kuviin. Yleisotanta vastauksista on, että sovelluskohtaisesti testausaineiston määrä voi vaihdella.

Perussovelluksen testaus voidaan tehdä pienemmällä otannalla, mutta järjestelmän käyttöönoton yhteydessä tulokset testataan pidemmän ajon perusteella.

Kysymykset 6–13 olivat strukturoituja kysymyksiä, miten koneoppiminen on korvannut eri toimintoja konenäköjärjestelmissä. Vastaukseen vastattiin asteikolla 0–10, ei ole korvannut tai korvannut kokonaan. Yhteenvetona vastauksista voidaan todeta, että koneoppimiseen hyödynnetään tällä hetkellä merkkientunnistuksessa ja tuotteiden laaduntarkastuksessa. Kappaletavara-automaatiassa, esimerkiksi tuotteiden paikoituksessa ja robotiikan kanssa vastausten painopiste on hieman enemmän koneoppimisen suuntaan. Koneoppimista hyödynnetään jo näissä toimenpiteissä, mutta se ei ole selvästikään syrjäyttänyt perinteisten järjestelmien hyödyntämistä. Tuotteen mittojen tarkastus, nopeat toimintaympäristöt ja 3D-kuvantaminen koetaan vielä kuuluvan enemmän perinteisten konenäköjärjestelmien toimintapiiriin. Pienten ja suurten tuotantoerien suhteen vastaukset ovat jakaantuneet. Pienten tuotantoerien suhteen vastaajilla on hyvin erilaiset

mielipiteet, mihin varmasti vaikuttaa toimintaympäristö ja järjestelmät, mitä on hyödynnetty. Suurissa tuotantoerissä vastauksia saatiin vain noin puolelta vastaajista, minkä lisäksi vastaukset ovat jakautuneet. Tästä voidaan tehdä johtopäätös, että järjestelmien monimuotoisuuden tuntemus saattaa olla vielä vähäistä ja kysymyksiin on ollut haastetta vastata.

Kysymyksessä 14 kartoitettiin lisäksi muut toimenpiteet, missä koneoppimista on hyödynnetty tai voitaisiin hyödyntää. Vastauksissa nostetaan esille koneoppimisen mahdollisuudet monimutkaisten ja haastavien kohteiden laadunvalvonnassa ja tuotantolaitteiden osien kunnonvalvonnassa sekä turvaratkaisuissa. Vastauksissa myös nostetaan esille perinteisten menetelmien edut muun muassa toistettavuudessa, tarkkuudessa ja mittauksia vaativissa sovelluksissa.

Kysymyksillä 15–16 kartoitettiin koneoppimisen vaikutuksia järjestelmän suunnitteluun ja käyttöönottoon, sekä asiakastyytyvyyteen ja kilpailuetuun verrattuna perinteisiin konenäköratkaisuihin. Suunnitteluun ja käyttöönottoon liittyvissä vastauksissa nousi esille uusien järjestelmien helpokäyttöisyys ja koneoppimisen tuoma joustavuus sekä adaptiivisuus erilaisiin sovelluskohteisiin. Vastauksista nousi esille myös mallien opettamiseen käytettävä aika, haastavien tapausten vaatima laskentateho ja raskaiden mallien hitaus. Sovelluksen kehittämisen ja muutosten tekemisestä nostettiin esille koneoppimisen tuomat hyödyt, mitkä nousevat opetusaineiston käyttämisestä. Sovelluksen tekijän ei tarvitse enää ymmärtää tarkasti konenäköön liittyvien toimintojen yhdistämisestä ja toiminnasta, eikä kuvausympäristöä tarvitse virittää niin tarkasti, kuin perinteisten sääntöpohjaisten toimintojen kanssa. Lisäksi mallien muokkaaminen jatkossa voi olla helpompaa tuotteiden lisääntyessä tai muuttuessa.

Asiakastyytyvyyteen ja kilpailuetuun liittyvissä vastauksissa nousi esille yritysten kiinnostus koneoppimista kohtaan, minkä lisäksi koneoppimisen avulla on pystytty parantamaan aikaisemmin tehtyjä sovelluksia ja tarjoamaan ratkaisuita, mihin aikaisemmin ei ole pystytty vastaamaan perinteisiä konenäköjärjestelmiä hyödyntäen. Asiakastyytyvyyttä koetaan pääsääntöisesti positii-viseksi, tai sitten sillä ei ole vielä ollut merkitystä. Negatiivisia kokemuksia ei vastauksissa kirjattu.

Kysymyksissä 17–18 kartoitettiin yrityksen tulevaisuudennäkymiä koneoppimiseen pohjautuvien järjestelmien hyödyntämisessä, sekä vapaa sana aihepiiriin ja kyselyyn liittyen. Tulevaisuudennäky-

mät koneoppimiselle olivat vastausten perusteella positiiviset. Vastauksissa nostettiin esille edullisten ja helppokäyttöisten sovellusten tuleminen ”koko kansan saataville”, eli järjestelmien hyödyntäminen tulee vaikuttamaan erittäin moneen asiaan. Helpomman käyttöönoton ja sovellusten tekemisen myötä koetaan, että konenäköjärjestelmiä tullaan asentamaan ja käyttöönottamaan enenemissä määriin myös sellaisilla alueilla, missä aikaisemmin kameroita ei ole juurikaan käytetty. Tulevaisuuden kohteina mainittiin myös merkkientunnistus, vaihtelevat olosuhteet, jatkuvien prosessien tilan arviointi sekä tuotantoympäristöjen turvallisuutta tarkkailevat järjestelmät.

Vapaan sanan osioon lisättiin vielä mainintoja AI:n ympärillä pyörivästä hypestä ja pyrkimyksestä soveltaa teknologiaa myös paikoissa, mihin se ei vielä sovellu.

”AI teknologiaa valmistavien yritysten vuosien hypetys aiheesta on saanut aikaan sen, että AI:ta yritetään ängetä paikkoihin, joihin se ei kuulu. Toivotaan, että ei tule kovin suuria pettymyksiä.”

Kyselytutkimuksen tulosten perusteella voidaan todeta, että konenäön parissa työskentelevillä vastaajilla oli hyvin samansuuntaiset ajatukset koneoppimista hyödyntävistä konenäköjärjestelmistä, mitä selvitystyön aikana saatiin kerättyä. Ainoa selkeä poikkeama oli 3D-kuvantamisessa, missä vastaajien mukaan koneoppimista ei juurikaan hyödynnetä, mutta selvityksen perusteella varsinkin kasapoimintasovelluksissa menetelmää on jo hyödynnetty.

## 8 Johtopäätökset

Selvitystyön tutkimuskysymysten tarkoituksena oli selvittää koneoppimista hyödyntävien konenäköjärjestelmien mahdollisuuksia ja haasteita. Näkökulmaksi valittiin yritysten kiinnostus aihealuetta kohtaan ja tarkoituksena oli kartoittaa, mitä yritysten tai aiheesta kiinnostuneiden tulisi tietää koneoppimisesta ja konenäöstä ennen projektin aloittamista. Mahdollisuuksien ja haasteiden lisäksi haluttiin selvittää, millaisia toimintoja järjestelmistä löytyy ja onko niillä pystytty helpottamaan koneoppimiseen liittyviä haasteita.

*Millaisia mahdollisuuksia ja haasteita koneoppimista hyödyntävissä konenäköjärjestelmissä on teollisuusyrityksille?*

Tutkimustyön perusteella voidaan todeta, että tietoperustan ja kirjallisuusselvityksen, järjestelmätestauksen, sekä kyselytutkimuksen tulokset ovat linjassa toistensa kanssa. Kirjallisuusselvityksessä nousseet käytettävissä olevat ominaisuudet ja mahdollisuudet, sekä haasteet tulevat selkeästi esiin järjestelmätestauksessa. Aineiston käsittely on työlästä ja haastavaa ja tulosten kannalta erittäin kriittistä. Selvitystyössä ei itse kerätty kuva-aineistoa, mikä olisi ollut konenäköjärjestelmiin liittyvä toimenpide ja tuonut erittäin paljon lisähaasteita riittävän laajan kuva-aineiston keräämiseen ilman olemassa olevaa tuotantolinjaa, tuotetta tai kuvausympäristöä. Sääntöpohjaisille konenäkötoiminnoille on erittäin kriittistä, että tuotteen kuvaus onnistuu hyvin, eikä ympäristössä ja tuotteessa tapahdu muutoksia. Koneoppimisen avulla pystytään tekemään paremmin vaihtelevaan ympäristöön ja tuotteen variaatioihin soveltuvia tunnistustoimenpiteitä, mutta se ei kuitenkaan poista konenäköympäristön liittyvää tarkkaa suunnittelutyötä ja hyvien kuvien vaikutusta lopputulokseen.

Vaikka järjestelmien laskentateho on lisääntynyt erittäin nopeasti, tulee koneoppimisen hyödyntämisessä kuvien tunnistukseen helposti laskentatehon tai muistin rajallisuus vastaan. Varsinkin perinteisillä tietokoneilla tehtävät mallien opettamiset eivät välttämättä onnistu ison resoluution kuvien ja syvien konvoluutioneuroverkkojen kanssa. Mallin opettamiseen tarvittava kuvamäärä ja verkon parametrit vaativat helposti kymmeniä tai satoja gigatavuja muistia. Tarjolla on kuitenkin maksullisia pilvipalveluita, missä voidaan hyödyntää erillisiä laskentayksiköitä suurienkin mallien opettamiseen.

### *Miten kaupalliset järjestelmät eroavat avoimen lähdekoodin järjestelmistä?*

Koneoppimista voidaan hyödyntää erittäin laajasti avoimen lähdekoodin järjestelmillä, millä tällä hetkellä tapahtuukin suurin osa tutkimus ja kehitystyöstä. Avoimen lähdekoodin järjestelmät ovatkin hyviä työkaluja tutkimuksen ja selvitystöiden tekemiseen, mutta ne saattavat olla haastavia ottaa käyttöön varsinkin ilman aikaisempaa ohjelmointikokemusta. Avoimen lähdekoodin työkalujen käyttämisestä tekee myös haastavaa niiden monipuolisuus ja laaja kirjastomäärä. Kirjastot myös päivittyvät nopeasti, eikä yhteensopivuus ole aina taattua jonkin osan päivittämisen jälkeen. Kaupallisten koneoppimista hyödyntävien konenäköjärjestelmien etuna on niiden helpompi asentaminen ja käyttöönotto. Yleisesti järjestelmiin on tarjolla tukea ja koulutusta, minkä lisäksi työkaluista

löytyy esimerkkiohjelmia, minkä avulla järjestelmää pääsee kokeilemaan. Kaupallisissa järjestelmissä on myös valmiiksi mietitty suunnittelutyön vaiheita ja toiminnoista on pyritty tekemään kuvaavia, myös asiaan vähemmän perehtyneille. Kaupallisten järjestelmien toiminnoilla kestää tulla markkinoille hieman pidempään, kuin avoimen lähdekoodin alustoilla tapahtuvalla kehityksellä, mutta tällä hetkellä kaupallisia järjestelmiä koneoppimismenetelmien kanssa tulee koko ajan lisää.

*Miten yritys pystyy hyödyntämään koneoppimista tuotteiden laadunvalvonnassa?*

*- Millaisia helppokäyttötoimintoja järjestelmistä löytyy?*

Järjestelmien helppokäyttöisyys tiettyihin toimenpiteisiin tulee varmasti lisäämään koneoppimisen hyödyntämiskohteita. Varsinkin konenäössä sovelluksen kehittäjällä tai järjestelmän käyttäjällä ei ole tarvetta päästä kehittämään vapaasti oman mallisia konvoluutioneuroverkkoja, tai hyödyntää jotain muuta koneoppimisen menetelmää, jos mallien toimivuus on pystytty kehittämään ja optimoimaan sellaiselle tasolle, että pienellä siirto-opettamisella järjestelmä saadaan toimimaan halutulla tarkkuudella. Ylimääräisistä ominaisuuksista ja parametreista voi olla joissakin tapauksissa haittaa ja ne saattavat hidastaa sovelluksen tekemistä ja hyödyntämistä.

Helppokäyttöisyyden ja valmiiksi opetettujen mallien ansiosta koneoppimista hyödyntävät järjestelmät saattavat korvata myös perinteisillä konenäkötoiminnoilla tehtäviä sovelluskohteita. Sovelluksen tekeminen voi olla nopeampaa tehdä koneoppimista hyödyntäen, minkä lisäksi järjestelmän muuttaminen operaattorin toimesta on helpompaa, kun esimerkiksi uusi tuote voidaan opettaa tallentamalla kuvia hyvistä ja huonoista näytteistä. Tuotteen muotoon, kuviointiin, väriin tai materiaalin laatuun voi tulla muutoksia, minkä seurauksena perinteinen sääntöpohjainen ohjelma täytyy tehdä monelta osin uudestaan.

Koneoppimismallien hyödyntämismahdollisuudet ovat kuitenkin vielä yleisellä tasolla varsin levälään. Aihepiirin kehitys on erittäin nopeaa, eikä vakiintuneita käytänteitä ole vielä ennättänyt muodostumaan. Laitteistojen laskentatehon nopea kehitys mahdollistaa myös mallien ja erilaisten algoritmien yhdistämisen, sekä yhä syvemmät neuroverkot, minkä takia sopivien rajojen vetäminen mallien laajuuteen ja tarkkuuteen on haastavaa.

*Millaisissa sovelluskohteissa erilaisia koneoppimis pohjaisia menetelmiä kannattaa hyödyntää ja milloin kannattaa suosia perinteisiä konenäkömenetelmiä?*

Koneoppimista hyödyntäviä järjestelmiä pystyy hyödyntämään erityisesti laaduntarkastukseen, tuotteiden luokitteluun, merkkien tunnistamiseen ja tuotteiden älykkääseen poimintaan. Laaduntarkastuksessa tuotteiden virheet voivat monesti olla moninaisia tai tuotteen ulkomuoto ja tekstuuri voivat vaihdella. Tällaisessa tapauksessa sääntöpohjaisten toimintojen käyttäminen voi olla jopa mahdotonta. Koneoppimista hyödyntäen, pystytään näytteiden perusteella opettamaan malli tunnistamaan vain tiettyjä virheitä. Tällä tavalla järjestelmä on joustavampi vaihtelevissakin olosuhteissa. Myös tuotteiden luokittelussa koneoppimisesta on samoja etuja, kuin laaduntarkastuksessa. Koneoppimisen avulla myös sovelluksen kehittäminen voi olla huomattavasti nopeampaa. Sovelluksen tekijän ei tarvitse tehdä erilaisia suodatuksia ja kuvankäsittelyä useiden sääntöpohjaisten toimintojen toteuttamiseksi. Kerätyn kuva-aineiston perusteella malli voidaan parhaassa tapauksessa opettaa tekemään luokittelu vain muutamilla kuvilla. Merkkien tunnistuksessa perinteisen menetelmän toiminta perustuu merkkien samankaltaisuuteen. Pienikin virhe tulosteessa, tai esimerkiksi mustesuihkutulostimen etäisyyden tai kulman muuttuminen vaikuttaa tulosteen kokoon ja asentoon. Koneoppimisen avulla voidaan hyödyntää esiopetettuja malleja, missä on hyödynnetty suuria määriä erilaisilla fonteilla tehtyjä merkintöjä. Mallin ansiosta merkkien ei tarvitse enää olla täysin samanlaisia, kuin perinteisillä menetelmillä tehtynä. Robottien kasapoiminnassa koneoppimista voidaan hyödyntää useiden erilaisten tuotteiden älykkääseen poimintaan, opettamalla pintojen tai muotojen tunnistus suurella määrällä erilaisia tuotteita. Tällä tavalla konenäköjärjestelmä pystyy tunnistamaan poimintapisteen kuva-alueelta ja lähettämään tiedon robotille, vaikka tuotteen 3D-mallia ei olisi opetettu järjestelmälle.

Perinteisten konenäkötoimintojen vahvuus on erityisesti tuotteen mittauksissa ja paikoituksessa. Tuotteen mittausta ei haluta suorittaa opetettuun malliin pohjautuen, vaan kuvasta halutaan tarkka arvo pikseleinä, tai millimetreiksi skaalattuna. Toimenpiteet ovat selkeitä ja nopeita laskea, mutta tarvitsevat laadukkaan kuvausympäristön. Myös selkeissä ominaisuuksien tunnistamisessa sekä esimerkiksi merkintöjen ja viivakoodien lukemisessa, perinteiset sääntöpohjaiset menetelmät toimivat hyvin. Useat merkinnät, kuten viivakoodit, ovat yleensä kamerajärjestelmille helppoa luettavia, niiden merkintätyyppin takia. Sääntöpohjaiset toiminnot ovat myös paljon käytettyjä järjestelmissä, missä laskentatehoa on rajallisesti. Myös erittäin nopeaa kuvaus- ja käsittelytaajuutta

vaativat sovellukset täytyy monesti rakentaa sääntöpohjaisilla toiminnoilla koneoppimisen hyödyntämisen sijaan.

## 9 Pohdinta

Tutkimustyön tekeminen oli haastavaa, mutta mielenkiintoista aihepiirien monimuotoisuuden, konenäköön liittyvien työtehtävien sekä henkilökohtaisiin kiinnostuksenkohteisiin liittyen. Koneoppimisen käytänteet ja algoritmit eivät olleet ennestään tuttuja muuten, kuin osittaisten kokeilujen ja harjoitteiden osalta, joten uuden asian opettelemista ja selvittämistä oli paljon. Kahden teoreettisen asian yhdistämistä varten selvitettävä perustietämys oli myös haastavaa rajata. Mitä perustietämystä koneoppimisesta ja konenäöstä tarvitaan, että voidaan miettiä ja suunnitella konvoluutioneuroverkkojen soveltamista konenäköongelmissa? Kirjallisuusselvitystä tehdessä joutuikin muuttamaan omia ajatuksia työn lopullisesta sisällöstä uusien näkökulmien ja tietojen karttuessa. Useassa asiassa joutui ottamaan lisää näkökulmia huomioon, että menetelmien toiminnan ja yhteyden muihin käsiteltäviin asioihin pystyi selittämään. Tutkimuksessa kuitenkin päästiin sopivaan suhteeseen perustietämystä kummastakin aihealueesta, että konvoluutioneuroverkkojen rakenne ja koneoppimisen mahdollisuudet ja ongelmat kuvankäsittelyssä pystyttiin esittämään.

Tutkimustyön eteneminen meni paljon rinnakkain järjestelmäselvityksen ja kirjallisuuskatsauksen kanssa. Avoimen lähdekoodin ja kaupallisten järjestelmien käyttämiseen ei ollut paljoo aikaisempaa kokemusta ennen selvitystyön tekemistä. Tämän takia järjestelmissä olevia toiminnoista ei ollut tietoa, eikä aihepiiriin ollut riittävästi taustatietämystä, että tutkimustyön olisi pystynyt tekemään täysin kirjallisuusselvityksen ohjaamana. Rinnakkain tehdystä toteutuksesta olikin hyötyä, jotta testauksessa käytettävien toimenpiteiden taustalle saatiin tarvittava määrä teoria-aineistoa.

Tutkimustyön aikataulut ja toteutus olisi vaatinut hieman tarkemman esisuunnittelun ja taustatutkimuksen. Tämän avulla työtä olisi voitu rajata hieman tehokkaammin ja tällä tavalla keskittyä tarkemmin pienempään osa-alueeseen. Työstä tuli varsin laaja osittain työn rajauksen, tai sen puutteellisuuden takia. Toisaalta aihepiirin ajankohtaisuus, omat intressit ja yritysten tarve saada lisätietoa koneoppimisesta kannusti tekemään työn hieman laajemmalla otteella.

Kyselytutkimuksen tavoitteena oli saada yrityksiltä tietoa ja näkemystä koneoppimisen haasteista ja mahdollisuuksista. Kyselytutkimus ei kuitenkaan tavoittanut riittävää vastaajajoukkoa hyvien tulosten saavuttamiseksi. Kyselytutkimusta olisi pitänyt jakaa laajemmalle vastaajajoukolle ja hyödyntää muutakin kanavaa, kuin pelkästään sähköpostiviestiä. Siinä vaiheessa esimerkiksi kyselyn vastauslinkin jakaminen sosiaalisen median kanavilla olisi pitänyt miettiä tarkasti, että näkyvyys olisi mennyt konenäköä hyödyntävien yritysten tietoisuuteen, ettei vastauksiin olisi tullut yleistä näkemystä ja mielipiteitä koneoppimisen hyödyntämisessä. Kyselytutkimus oli kuitenkin vain pieni osa tutkimustyötä tukemaan tai haastamaan selvityksissä saatuja tuloksia.

## 9.1 Luotettavuus

Aihepiirin ympärillä on erittäin paljon toimintaa ja erilaisia tutkimustöitä ja tekstejä julkaistaan erittäin nopeaan tahtiin. Myös uutisointi ja erityisesti generatiiviset tekoälyt ovat tulleet osaksi jokapäiväistä elämäämme. Aihepiirin eteneminen on siis tällä hetkellä erittäin nopeaa. Tästä syystä pari vuotta, tai vain muutama kuukausi sitten kerätty aineisto ja tietämys saattaa olla vanhaa. Menetelmiä vertaillen perusidea ei ole niinkään vanhentumassa, tai kuvantamiseen ja kuvankäsittelyyn liittyvät teorit eivät juurikaan muutu, mutta uusia tapoja niiden hyödyntämiseen tulee nopealla syklillä. Tämän takia jokin työtä tehdessä ollut menetelmä saattaa olla muuttunut ja toimia erittäin hyvin ja luotettavasti kohteeseen, mikä tässä tutkimustyössä on käsitelty toisella tavalla. Koneoppimisen hyödyntämismahdollisuuksien kanssa kannattaakin olla valppaana ja selvittää menetelmien kehittymistä suhteellisen tiheällä syklillä. Jos jokin ei tällä hetkellä ole mahdollista, tai vaikuttaa haastavalta, ei aihepiiriä kannata sulkea vuosiksi sivuun.

Tutkimustyössä on pyritty tuomaan esille tarvittavaa pohjatietoa järjestelmien ja menetelmien hyödyntämiseen, eli lisätä tietämystä aihepiiriin. Kirjallisuusselvityksen, järjestelmätestauksen ja kyselytulosten yhtenäinen linja kuvastaa työn luotettavuutta. Työn pääpaino ei ole ollut esittää tarkkoja järjestelmäominaisuuksia ja vertailla eri järjestelmien tarkkuuksia tietyllä kuva-aineistolla, mutta kuva-aineisto on silti avoimen palvelun kautta ladattavissa ja järjestelmätestaukset ovat myös tarvittaessa mahdollista toistaa. Työn luotettavuuden tarkastelua varten työ olisi hyvä antaa luettavaksi ja arvioitavaksi aihepiirin parissa työskenteleville. Tällä tavalla saataisiin tieto, onko aihepiiri esitetty oikeellisesti ja sopivalla tarkkuudella, vai onko työstä jäänyt jotain oleellista pois.



## 9.2 Jatkokehitys

Tutkimustyössä oli kaksi erittäin laajaa aihealuetta, koneoppiminen ja konenäkö. Työssä on avattu vain menetelmien perusteita ja edetty nopeasti järjestelmätestauksen tasolle. Tämän takia kaikkea aihepiiriin liittyvää ei ole pystytty käymään läpi. Myös järjestelmätestauksien tarkoituksena oli selvittää, minkä kaltaisia toimintoja järjestelmästä löytyy, mutta varsinaista vertailua järjestelmien tai toimintojen välillä ei tehty. Tähän liittyen perinteisten sääntöpohjaisten ja koneoppimista hyödyntävien konenäkötoimintojen tarkempi vertailu olisi varmasti hyödyllinen selvittää. Selvityksen voisi tehdä erilaisia aineistoja käyttämällä ja vertailemalla menetelmien toimintavarmuutta. Lisäksi työssä käytiin läpileikkaus algoritmien ja opetusmenetelmien osalta. Kuvankäsittelyyn liittyvien menetelmien osalta voisi myös tehdä tarkempaa analyysiä menetelmien hyödyistä erilaisen aineiston kanssa.

Toinen esille noussut jatkokehityskohde olisi konvoluutioverkkojen tehostaminen perinteisillä sääntöpohjaisilla konenäkötoiminnoilla. Esimerkiksi suuret kuvat ja raskaat konvoluutioneuroverkot voivat olla ongelmallisia opettaa, minkä lisäksi malli saattaa olla liian raskas nopeampiin toimintaympäristöihin. Suuren kuvan esikäsittely ja esimerkiksi muutoskohdan tunnistaminen voisi tehostaa mallin toimintaa, kun syötteenä voitaisiin määrittää vain kuvassa tapahtunut muutoskohta.

Koneoppimista hyödyntäviä konenäköjärjestelmiä on tullut viime aikoina paljon lisää, ja koko ajan tulee uusia toimijoita, minkä lisäksi toimintojen kirjastot päivittyvät. Järjestelmien tarkempi vertailu ja kuvaus, millaisia toimintoja löytyy ja minkälaiseen ongelmaan niillä pystytään tarjoamaan ratkaisua, kiinnostaisi varmasti yrityksiä, joilla on suunnitteilla konenäkösovelluksen hankinta.

Seuraava suuri läpimurto aihepiiriin ympärillä tulee olemaan kvanttilaskenta. Kvanttilaskentaa selvitetään ja sen parissa tehdään paljon tutkimusta. Konenäköjärjestelmien haasteen, aineiston koon, takia kvanttilaskentaa tullaan varmasti soveltamaan myös konenäön puolella. Tämän takia myös kvanttilaskennan tutkiminen ja selvittämien olisi erittäin hyödyllistä lähitulevaisuutta ajatellen.

## Lähteet

30 best data labeling tools. 2024. Viitattu 10.5.2024. <https://www.superannotate.com/blog/best-data-labeling-tools>

3D-ohjelmistoratkaisut. N.d. Viitattu 25.4.2024. <https://www.oem.fi/tuotteet/konnen%C3%A4k%C3%B6/ohjelmistot/3d-ohjelmistoratkaisut-637594>

ACE2 Basic & Pro – matriisikamerat. N.d. Viitattu 2.5.2024. <https://www.oem.fi/tuotteet/konnen%C3%A4k%C3%B6/kamerat/matriisikamerat-426789/ace-2-basic-pro-matriisikamerat-677707>

Activation Functions. N.d. Viitattu 17.4.2024. [https://ml-cheatsheet.readthedocs.io/en/latest/activation\\_functions.html#softmax](https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html#softmax)

Astrocyte. N.d. Viitattu 20.4.2024. <https://www.teledynedalsa.com/en/products/imaging/vision-software/astrocyte/>

Automatic Threading Rope Monitoring. N.d. Viitattu 10.5.2024. <https://www.procemex.com/vision-applications/automatic-threading-rope-monitoring/>

Azad, P., Gockel, T. & Dillmann, R. 2008. Computer Vision – Principals and Practice. UK: Elektor International Media BV.

Basic Lens Selection. N.d. Artikkelit Edmund Opticsin sivuilla. Viitattu 6.12.2023. <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/how-to-choose-a-variable-magnification-lens/>

Bergmann, D. 2023. What is self-supervised learning? Viitattu 7.9.2023. <https://www.ibm.com/topics/self-supervised-learning>

Bernet, C. 2019. The 1-Neuron Network: Logistic Regression. Viitattu 16.4.2024.

<https://thedatafrog.com/en/articles/logistic-regression/>

Bianco, S., Cadene, R., Celona, L. & Napoletano, P. 2018. Benchmark Analysis of Representative Deep Neural Network Architectures. IEEE Xpres. Doi: 10.1109/ACCESS.2018.2877890. Viitattu 3.2.2024. <https://ieeexplore.ieee.org/document/8506339>, IEEE.

Bouchard, L. 2020. What is Self-Supervised Learning? | Will machines ever be able to learn like humans? Viitattu 5.10.2023. <https://medium.com/what-is-artificial-intelligence/what-is-self-supervised-learning-will-machines-be-able-to-learn-like-humans-d9160f40cdd1>

Brown, B. & Zai, A. 2020. Deep Reinforcement Learning in Action. Manning Publications. Viitattu 26.9.2023. <https://janet.finna.fi>, Skillsoft Books ITPro.

Chen, Y., Gogoi, A., Lee, S., Tsai-Lin, Y., Yi, P., Lu, M. & Hsieh, C. 2019. Coherent Narrow-Band Light Source for Miniature Endoscopes. IEEE Xpres. Doi: 10.1109/JSTQE.2018.2836959. Viitattu 24.10.2023. <https://ieeexplore.ieee.org/document/8359410>, IEEE.

Cognex. 2022. AI For Factory Automation. Viitattu 13.9.2023. <https://www.cognex.com/en-fi/what-is/edge-learning>

Cognex CalibrateGrid. N.d. Viitattu 14.4.2024. [https://support.cognex.com/docs/is\\_611/web/EN/ise/Content/Reference/CalibrateGrid.htm?Highlight=grid](https://support.cognex.com/docs/is_611/web/EN/ise/Content/Reference/CalibrateGrid.htm?Highlight=grid)

Cognex What is machine vision. N.d. Viitattu 27.9.2023. <https://www.cognex.com/what-is/machine-vision/what-is-machine-vision>

Concept of Bits Per Pixel. N.d. Viitattu 5.3.2024. <https://www.javatpoint.com/dip-concept-of-bits-per-pixel>

Dabhi, R. 2020. Casting product image data for quality inspection. Viitattu 4.4.2024. <https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product?rvi=1>

Davies, E. 2012. Computer and Machine Vision: Theory, Algorithms, Practicalities. Neelja painos. London: Elsevier. <https://janet.finna.fi>, EBook Central Academic International Edition.

Denneman, F. 2022. Training vs Inference – Memory Consumption by Neural Networks. Viitattu 5.5.2024. <https://frankdenneman.nl/2022/07/15/training-vs-inference-memory-consumption-by-neural-networks/>

Doshi, R., Hiran, K., Jain, R & Lakhwani, K. 2021. Machine learning: master supervised and unsupervised learning algorithms with real examples. India: BPB Publications. Viitattu 29.9.2023. <https://janet.finna.fi>, Skillssoft Books ITPro.

Eckstein, C. 2022. Deep Learning & AI combined with traditional methods facilitate automation. Vision 2022. Stuttgart. MVTec Software GmbH seminaariesitys.

Embedded Vision vs. Machine Vision – Everything You Need to Know. N.d. Viitattu 6.2.2024. <https://www.technexion.com/resources/embedded-vision-vs-machine-vision-everything-you-need-to-know/>

Every robot needs a brain. N.d. Viitattu 25.4.2024. <https://www.robominds.de/products/robobrain>

Gardete, J. 2024. Convolutional Neural Networks: A Comprehensive Guide. Viitattu 4.4.2024. <https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175>

Geogebra. N.d. Viitattu 5.3.2024. <https://www.geogebra.org/m/Dq2A7aRv>

Ghosh, S. 2024. The Ultimate Guide to Evaluation and Selection of Models in Machine Learning. Viitattu 14.3.2024. <https://neptune.ai/blog/ml-model-evaluation-and-selection>

Gillis, S. 2023. What is unsupervised learning? Viitattu 24.10.2023. <https://www.tech-target.com/searchenterpriseai/definition/unsupervised-learning>

Gonzalez, R. & Woods, R. 2008. Digital Image Processing. 3. p. New Jersey: Person Prentice Hall.

Goodfellow, I. Bengio, Y. & Courville. A. 2016. Deep Learning. MIT Press.

Halcon a product of MVTec. 2023. Viitattu 9.11.2023. [www.mvtec.com/fileadmin/Redaktion/mvtec.com/products/halcon/documentation/solution\\_guide/solution\\_guide\\_ii\\_d\\_classification.pdf](http://www.mvtec.com/fileadmin/Redaktion/mvtec.com/products/halcon/documentation/solution_guide/solution_guide_ii_d_classification.pdf)

Halcon deep learning tool. N.d. Viitattu 7.5.2024. <https://www.mvtec.com/products/deep-learning-tool>

Halcon the power of machine vision. 2015. 7a, uudistettu painos. Viitattu 9.11.2023. <http://download.mvtec.com/halcon-12.0-quick-guide.pdf>

Harash, B. 2023. Machine Learning for Beginners. Toinen painos. Los Angeles: BPB Publications. Viitattu 4.4.2024. <https://janet.finna.fi>, Skillssoft Books ITPro.

Huilgo, P. 2023. Top 4 Pre-Trained Models for Image Classification with Python Code. Viitattu 29.9.2023. <https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/>

Hyvä tieteellinen käytäntö ja sen loukkausepäilyjen käsitteleminen Suomessa. 2023. Helsinki: Tutkimuseettinen neuvottelukunta. Viitattu 7.5.2024. [https://tenk.fi/sites/default/files/2023-03/HTK-ohje\\_2023.pdf](https://tenk.fi/sites/default/files/2023-03/HTK-ohje_2023.pdf)

Ilonen, J. & Virkajärvi, M. 4.4.2023. Vaativat konenäköratkaisut sellu- ja paperiteollisuudessa. Jyväskylän ammattikorkeakoulu. Procemex Oy:n seminaariesitys.

Image Processing with NI Vision Development Module. 2024. Viitattu 10.4.2024.

<https://www.ni.com/en/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/image-processing-with-ni-vision-development-module.html>

Imaging Electronics 101: Camera Resolution for Improved Imaging System Performance. N.d. Viitattu 2.2.2024. <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/camera-resolution-for-improved-imaging-system-performance/>

Imaging Electronics 101: Understanding Camera Sensors for Machine Vision Applications. N.d. Viitattu 20.9.2023. <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/understanding-camera-sensors-for-machine-vision-applications/>

In-Sight D900 vision system. N.d. Viitattu 26.4.2024. <https://www.cognex.com/products/machine-vision/2d-machine-vision-systems/in-sight-d900>

Introduction To Modern Camera Technologies. N.d. Artikkele Teledyne:n sivuilla. Viitattu 6.12.2023 <https://www.photometrics.com/learn/white-papers/introduction-to-modern-camera-technologies>

Jordan, J. 2017a. Convolutional neural network. Viitattu 5.5.2024. <https://www.jeremyjordan.me/convolutional-neural-networks/>

Jordan, J. 2017b. Evaluating a machine learning model. Viitattu 15.4.2024. <https://www.jeremyjordan.me/evaluating-a-machine-learning-model/>

Kanade, V. 2022. What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices. Viitattu 14.1.2024. <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/>

Kenton, W. 2023. What Is Nonlinear Regression? Comparison to Linear Regression. Viitattu 13.10.2024. <https://www.investopedia.com/terms/n/nonlinear-regression.asp>

Kneusel, R. 2021. Practical Deep Learning: A Python-Based Introduction. San Francisco: No Starch Press. <https://janet.finna.fi>, Skillsoft Books ITPro.

Kumar, P. 2023. CMOS vs CCD: Why CMOS Sensors are Ruling the World of Embedded Vision. Viitattu 2.10.2023.

LabVIEW Analytics and Machine Learning Toolkit <https://www.ni.com/fi-fi/shop/product/labview-analytics-and-machine-learning-toolkit.html>

Latokartano, J., Karvonen, H., Skriko, T., Holamo, O., Christophe, F., Ahonen, T., Haapakoski, T., Partanen, A., Lempiäinen, J., Kampiainen, P., Paasio, L., Liljamo, J. & Närhi, J. 2023. Konenäkö. Julkaisussa Teollisuuden robotiikka. Toim. K. Välimäki & M. Niemelä. Helsinki: Suomen Robotiikkayhdistys, 36-75.

Levman, J. 2011. Julkaisussa Support Vector Machines: Data Analysis, Machine Learning and Applications. New York: Nova science publisher. s. 2-3. <https://janet.finna.fi/Record/jamk.993628018506251?sid=4714293623>

Machine Vision System Sensors Software. N.d. Esite ladattu Cognexin tuotesivuilta. Viitattu 26.4.2024. <https://www.cognex.com/products/machine-vision/2d-machine-vision-systems/in-sight-2800>

Marques, O. 2011. Practical Image and Video Processing Using MATLAB. New Jersey: John Willay & Sons. <https://janet.finna.fi>, Ebook Central.

Montgomery, D., Peck, E. & Vining, G. 2012. Introduction to linear regression analysis. Viides painos. New Jersey: John Wiley & Sons.

Moore, T. N.d. Viitattu 17.4.2024. <https://learn.flucoma.org/learn/mlp-parameters/>

Müller, A. & Guido, S. 2016. Introduction to Machine Learning with Python. Sebastopol: O'REILLY.

Nelson, D. 2020. Mikä on KNN (K-Nearest Neighbors)? Viitattu 13.12.2023. <https://www.unite.ai/fi/what-is-k-nearest-neighbors/>

Ojasalo, K., Moilanen, T. & Ritalahti, J. 2015. Kehittämistyön menetelmät: uudenlaista osaamista liiketoimintaan. Helsinki: Sanoma Pro. Viitattu 8.5.2024. <https://janet.finna.fi/>, Ellibs ebook.

ONNX Consept. 2024. ONNX:n esittelysivu. Viitattu 7.5.2024. [https://onnx.ai/onnx/intro/concepts.html?fbclid=IwZXh0bgN-hZW0CMTAAAR1B1i0u\\_q9\\_hy\\_kASzzLlCgdMDmT8qU\\_du2Va35Q9crpQS-id\\_UsL2wDU\\_s\\_aem\\_ATERY-CM-zLkpgTmCb4It7uIVALIM-nPcJGHvng5PKEI5gvvHS4AVtn9D6gP4JY8dWBm6znmQw\\_STI3kkW6BaRyZ](https://onnx.ai/onnx/intro/concepts.html?fbclid=IwZXh0bgN-hZW0CMTAAAR1B1i0u_q9_hy_kASzzLlCgdMDmT8qU_du2Va35Q9crpQS-id_UsL2wDU_s_aem_ATERY-CM-zLkpgTmCb4It7uIVALIM-nPcJGHvng5PKEI5gvvHS4AVtn9D6gP4JY8dWBm6znmQw_STI3kkW6BaRyZ)

OpenCV. N.d. OpenCV esittelysivu. Viitattu 7.5.2024. <https://opencv.org/about/>

PCcontrol The New Automation Technology Magazine. 2021. Beckhoff esite 4/2021. Verl: Beckhoff Automation.

Pietikäinen, M. & Silvén, O. 2023. Miten tekoäly vaikuttaa elämäämme 2050-luvulla. Oulu: Kokenäön ja signaalianalyysin keskus.

Raghunath. 2019. Image Histograms in OpenCV. Viitattu 3.3.2024. <https://medium.com/@rundayala/image-histograms-in-opencv-40ee5969a3b7>

Regressioanalyysin rajoitteet. 2003. Viitattu 13.10.2023. <https://www.fsd.tuni.fi/metelmaopetus/regressio/rajoitteet.html>)

Samples of ONNX export. N.d. Beckhoff Information System sivusto. Viitattu 22.4.2024. [https://infosys.beckhoff.com/english.php?content=../content/1033/tf38x0\\_tc3\\_ml\\_nn\\_inference\\_engine/index.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tf38x0_tc3_ml_nn_inference_engine/index.html&id=)

Sanderson, G. 2017. But what is a Neural Network? Viitattu 14.4.2024. <https://www.3blue1brown.com/lessons/neural-networks>

Sharma, N. 2024. K-Means Clustering Explained. Viitattu 2.3.2024. <https://neptune.ai/blog/k-means-clustering>



Siltala, N. & Gautam, M. 2023. Konenäkö. Julkaisussa Teollisuuden robotiikka. Toim. K. Välimäki & M. Niemelä. Helsinki: Suomen Robotiikkayhdistys, 172–187.

Solution Guide III-A 1D Measuring. 2024. MVTec Halcon esite. Viitattu 14.5.2024.

[www.mvtec.com/fileadmin/Redaktion/mvtec.com/products/halcon/documentation/solution\\_guide/solution\\_guide\\_iii\\_a\\_1d\\_measuring.pdf](http://www.mvtec.com/fileadmin/Redaktion/mvtec.com/products/halcon/documentation/solution_guide/solution_guide_iii_a_1d_measuring.pdf)

Solution Guide III-B 2D Measuring. 2024. MVTec Halcon esite. Viitattu 14.5.2024.

[https://www.mvtec.com/fileadmin/Redaktion/mvtec.com/products/halcon/documentation/solution\\_guide/solution\\_guide\\_iii\\_b\\_2d\\_measuring.pdf](https://www.mvtec.com/fileadmin/Redaktion/mvtec.com/products/halcon/documentation/solution_guide/solution_guide_iii_b_2d_measuring.pdf)

Sonka, M., Hlavac, V. & Boule, R. 2008. Image Processing, Analysis, and Machine Vision. Stamford: Gengage Learning.

Sun, S., Dong, Z., Mao, L. & Wu, L. 2019. Multiview Machine Learning. Springer. Viitattu 29.9.2023.

<https://janet.finna.fi>, Skillssoft Books ITPro.

Szeliski, R. 2022. Computer Vision Algorithms and Application. Toinen painos. Sveitsi: Springer.

<https://janet.finna.fi>, VLeBooks.

Szeliski, R. 2022. Computer Vision Algorithms and Application. Toinen painos. Sveitsi: Springer.

<https://janet.finna.fi>, VLeBooks.

Tekoälyratkaisut: automaation todellinen voima. N.d. Sivustolla oleva esittelyvideo. Viitattu 20.4.2024. <https://www.sick.com/fi/fi/tekoaely/s/artificial-intelligence>

Tervola, J. 2023. Suomen robotti-investoinnit junnaavat paikoillaan – Ruotsin robottitiheys on kaksinkertainen. Viitattu 10.5.2024. <https://www.tekniikkatalous.fi/uutiset/suomen-robotti-investoinnit-junnaavat-paikoillaan-ruotsin-robottitiheys-on-kaksinkertainen/86a1aca3-f1f7-4178-ae77-73ed40587b59>

Thakare, A. & Umesh, S. 2023. Artificial Intelligence Applications and Reconfigurable Architectures. Hoboken: John Willey & Sons. <https://janet.finna.fi>, Skillsoft Books ITPro.

Toikko, T. & Rantanen, T. 2009. Tutkimuksellinen kehittämistoiminta. Tampere: Tampere University Press. Viitattu 8.5.2024. <http://urn.fi/URN:ISBN:978-951-44-7732-4>.

Top 10 Open Source Data Labelling Tools for Computer Vision. 2023. Viitattu 10.5.2024 <https://smartone.ai/blog/top-10-open-source-data-labeling-tools-for-computer-vision/>

Transfer learning and fine-tuning. N.d. Viitattu 25.4.2024. [https://www.tensorflow.org/tutorials/images/transfer\\_learning](https://www.tensorflow.org/tutorials/images/transfer_learning)

Types of Images. N.d. Viitattu 5.3. <https://www.javatpoint.com/dip-types-of-images>

Valli, R. 2018. Aineistonkeruu kyselylomakkeella. Julkaisussa Ikkunoita tutkimusmetodeihin. 1, Metodien valinta ja aineistonkeruu: virikkeitä aloittelevalle tutkijalle. Jyväskylä: PS-kustannus. <https://janet.finna.fi/>, Ellibs ebooks.

Verdhan, V. 2021. Computer Vision Using Deep Learning: Neural Network Architectures with Python and Keras. Limerick: Apress. <https://janet.finna.fi>, Skillsoft Books ITPro.

Visionpro deep learning. N.d. Viitattu 20.4.2024. <https://www.cognex.com/products/machine-vision/vision-software/visionpro-deep-learning>

Wang, J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M. & Chau, P. N.d. What is a Convolutional Neural Network? Viitattu: 4.4.2024. <https://poloclub.github.io/cnn-explainer/>

Welling, A. 2023. Valmet Automotive ei itsekään uskonut uuteen hitsaustekniikkaan – keksintö yllätti kaikki ja nyt se voi mullistaa koko autonrakennuksen. Viitattu 3.10.2023. <https://yle.fi/a/74-20034015>

What are Naïve Bayes classifiers? N.d. Viitattu 20.12.2023. <https://www.ibm.com/topics/naive-bayes>

What is gradient descent? N.d. Viitattu. 19.4.2024. <https://www.ibm.com/topics/gradient-descent>

What is linear regression? N.d. Viitattu 10.11.2023. [https://www.ibm.com/topics/linear-regression?mhsrc=ibmsearch\\_a&mhq=linear%20regression](https://www.ibm.com/topics/linear-regression?mhsrc=ibmsearch_a&mhq=linear%20regression)

What is logistic regression? N.d, Viitattu 14.1.2024. <https://www.ibm.com/topics/logistic-regression>

What is machine vision? N.d. Viitattu 13.9.2023. <https://www.intel.com/content/www/us/en/manufacturing/what-is-machine-vision.html>

What is a multilayer perceptron (MLP) or a feedforward neural network (FNN)? 2024. Viitattu 4.3.2024. <https://aiml.com/what-is-a-multilayer-perceptron-mlp/>

What is supervised learning? N.d. Viitattu 13.12.2023. <https://www.ibm.com/topics/supervised-learning>

What is the KNN algorithm? N.d. Viitattu 13.12.2023. <https://www.ibm.com/topics/knn>

What is unsupervised learning? Viitattu 14.10.2023. [https://www.ibm.com/topics/unsupervised-learning?mhsrc=ibmsearch\\_a&mhq=unsupervised%20learning](https://www.ibm.com/topics/unsupervised-learning?mhsrc=ibmsearch_a&mhq=unsupervised%20learning)

Wilson, P. 2016. Design recipes for FPGAs. Toinen painos. Lontoo: Elsevier. <https://janet.finna.fi>, Ebook Central.

Yasar, K. 2023. Viitattu 13.9.2023. <https://www.techtarget.com/searchenterpriseai/definition/machine-vision-computer-vision>

Zebra's Deep Learning Capabilities. N.d. Viitattu 20.4.2024. [www.zebra.com/content/dam/zebra\\_dam/en/brief/solution/deep-learning-brief-solution-en-us.pdf](http://www.zebra.com/content/dam/zebra_dam/en/brief/solution/deep-learning-brief-solution-en-us.pdf)

Zhang, Y. 2023. 3D Computer Vision: Principles, Algorithms and Applications. Singapore: Springer. <https://janet.finna.fi>, VLeBooks

## Liitteet

### Liite 1. Kyselytutkimus

#### Mihin toimialaan yrityksenne kuuluu?

Vastaajien määrä: 7

Vastaukset
Lasermerkintälaitte valmistaja, koneenrakentaja
Teollisuuskomponenttien toimittaja
Konenäköjärjestelmiä valmistavaan yritykseen
Maahantuojaja/Tukkuliike
Teknolohiateollisuus
Teollisuuden konenäköjärjestelmät
sähkö- ja elektroniikkalaitteet

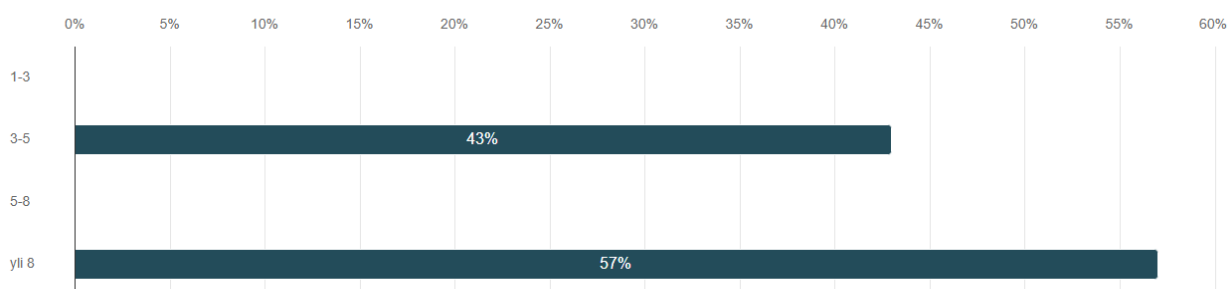
#### Kuinka monta työntekijää yrityksessänne työskentelee?

Vastaajien määrä: 7

Vastaukset
n. 40.
140
noin 19000 henkilöä, joista karkeasti konenäköön tai konenäön AI-sovelluksiin 20 - 30 henkilöä
n. 180
>18000
Vähän vajaat 100.
90

#### Kokemus konenäköjärjestelmistä (vuotta)?

Vastaajien määrä: 7



	n	Prosentti
1-3	0	0,0%
3-5	3	42,9%
5-8	0	0,0%
yli 8	4	57,1%

#### Millaista kokemusta koneoppimiseen pohjautuvista konenäköjärjestelmistä sinulta/yrityksestä löytyy (esim. avoimen lähdekoodin järjestelmät, kaupalliset järjestelmät)?

Vastaajien määrä: 7

Vastaukset
------------

Olen työskennellyt pelkästään konenäön parissa 27 vuotta. Näiden vuosien aikana kertynyt kokemuksia eri valmistajien neuroverkko pohjaisista järjestelmistä. Parhain osaaminen kuitenkin Cognex Edge & Deep Learning ohjelmistoista, joiden kanssa työskennellyt alusta asti n. 10 vuotta.
Kaupalliset kirjastot.
Avoimet esim. Yolo ja kaupalliset esimerkiksi Cognex ja LabView.
Toimin konenäkökomponenttien myynnissä 2D/3D kamerat, sekä laskentatehoon liittyvissä asioissa ym. muut tarvikkeet linssit valot jne.
Ei mainittavaa, yleisellä tasolla seuraamista
Kaupallisista Cognex ja Matlab, avoimista vähän kaikenlaista, pääosin OpenCV ja Tensorflow/pytorch.
omakohtaista kokemusta vain avoimen lähdekoodin järjestelmistä

### Koneoppimisen yhteydessä mallin toiminta usein varmistetaan käymällä läpi opetuksesta erillinen kuva-aineisto. Miten laajalla otannalla perinteisten konenäköfunktioiden toiminta yleensä varmennetaan?

Vastaajien määrä: 7

Vastaukset
Riippuu sovelluksen vaikeusasteesta. Perinteinen konenäkö on huomattavasti herkempää odottamattomille ilmiöille kuvassa. Monesti perussovellus saadaan testauskuntoon muutamilla kuvilla ja hiotaan kuntoon joitakin päiviä kestäväällä ylösajolla oikeassa tuotannossa.
satojen tai mieluusti tuhansien oikeiden ja väärin kuvien kanssa.
Ollaan käytetty 50 - 500 kuvaa/tunnistus
Toimii/Ei toimi. Robotiikkaan yhdistettynä varmistetaan että X-määrä työsyklejä onnistuu ennenkuin todetaan luotettavasti toimivaksi.
Muutamalla
Riippuu sovelluksesta. Perinteisellä konenäöllä voi olla tiedossa että tehty muutos vaikuttaa vain johonkin tiettyyn tapaukseen, jolloin selviää pienemmällä aineistolla. Oppivat järjestelmät vaatii aina isomman testiaineiston, kun toiminta voi muuttua vähän kaikkialla.
valitettavasti en osaa sanoa, arvelisin että tämä on hyvin tapauskohtaista

### Mite paljon seuraavilla osa-alueella koneoppimiseen pohjautuvat järjestelmät ovat korvanneet perinteisiä konenäköjärjestelmiä?

0 1 2 3 4 5 6 7 8 9 10

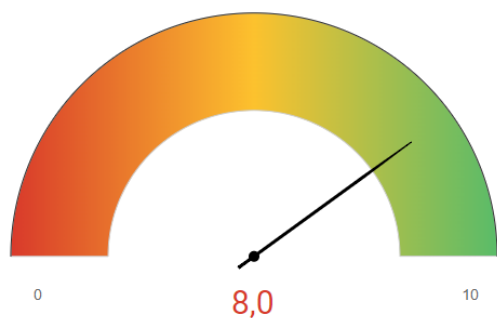
Ei ole korvannut            Korvannut kokonaan

En tiedä

**Merkintunnistus (OCR)**

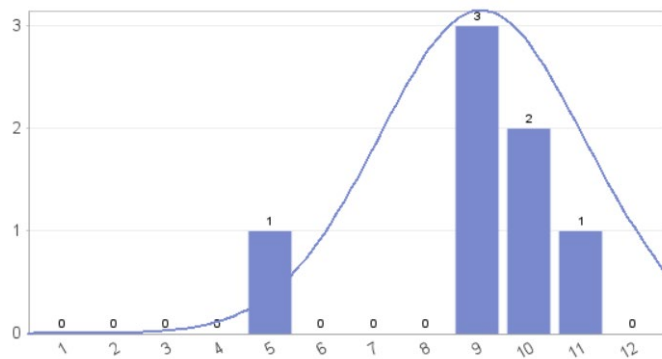
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluettu keskiarvosta



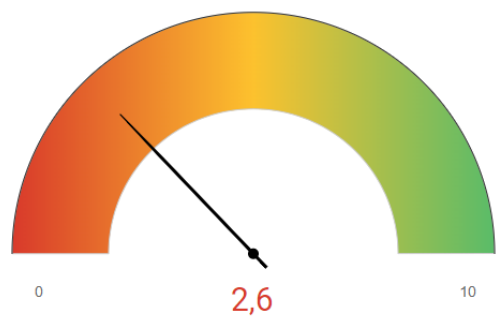
Arvostelijat							Passiiviset		Suositelijat		En tiedä
0	1	2	3	4	5	6	7	8	9	10	
n = 1							n = 3		n = 3		
14,3%							42,8%		42,9%		
0	0	0	0	1	0	0	0	3	2	1	0
0,0%	0,0%	0,0%	0,0%	14,3%	0,0%	0,0%	0,0%	42,8%	28,6%	14,3%	0,0%

Yhteensä		
Vastauksia	NPS	Keskiarvo
7	29	8,0

**Tuotteen mittojen tarkastus**

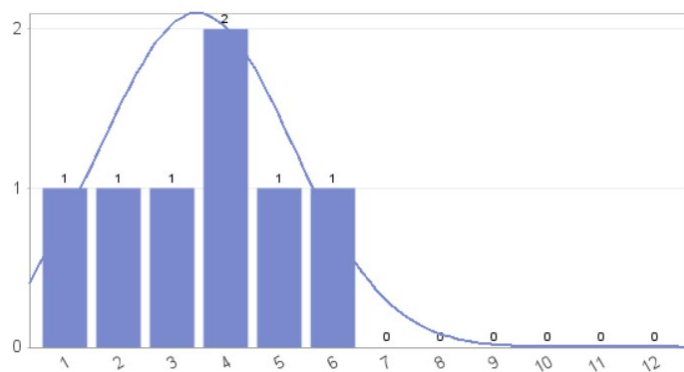
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluettu keskiarvosta



Arvostelijat							Passiiviset		Suositelijat		En tiedä
0	1	2	3	4	5	6	7	8	9	10	
n = 7							n = 0		n = 0		
100,0%							0,0%		0,0%		
1	1	1	2	1	1	0	0	0	0	0	0
14,3%	14,3%	14,3%	28,5%	14,3%	14,3%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

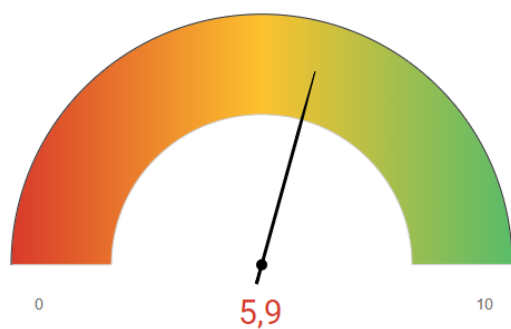
Yhteensä		
Vastauksia	NPS	Keskiarvo
7	-100	2,6



**Laadunvalvonta**

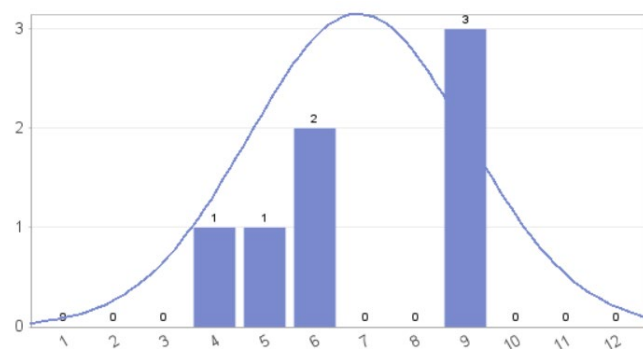
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluetettu keskiarvosta



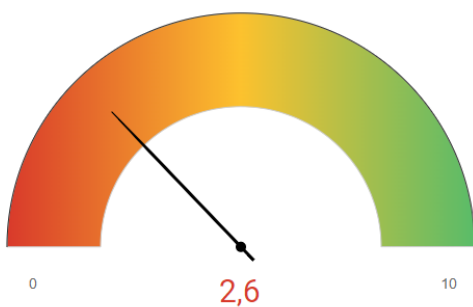
Arvostelijat							Passiiviset		Suositellijat		En tiedä
0	1	2	3	4	5	6	7	8	9	10	
n = 4							n = 3		n = 0		
57,1%							42,9%		0,0%		
0	0	0	1	1	2	0	0	3	0	0	0
0,0%	0,0%	0,0%	14,3%	14,3%	28,6%	0,0%	0,0%	42,8%	0,0%	0,0%	0,0%

Yhteensä		
Vastauksia	NPS	Keskiarvo
7	-57	5,9

**Nopeat toimintaympäristöt**

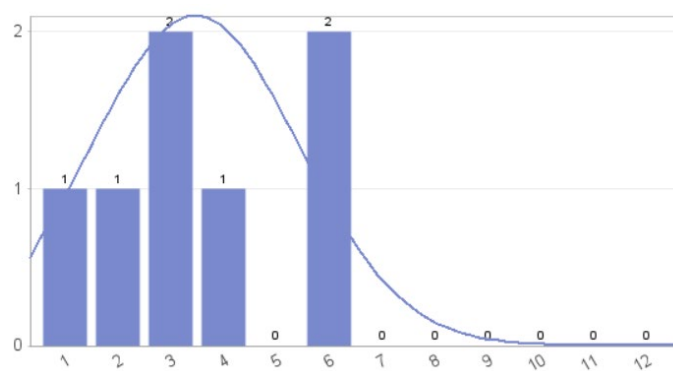
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluetettu keskiarvosta



Arvostelijat							Passiiviset		Suositellijat		En tiedä
0	1	2	3	4	5	6	7	8	9	10	
n = 7							n = 0		n = 0		
100,0%							0,0%		0,0%		
1	1	2	1	0	2	0	0	0	0	0	0
14,3%	14,3%	28,5%	14,3%	0,0%	28,6%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

Yhteensä		
Vastauksia	NPS	Keskiarvo
7	-100	2,6

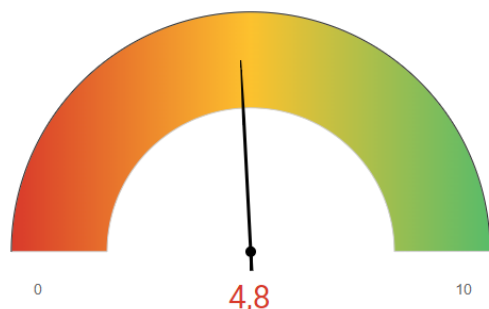




### Kappaletavara-automaatio (tuotteiden paikoitus, robotiikka)

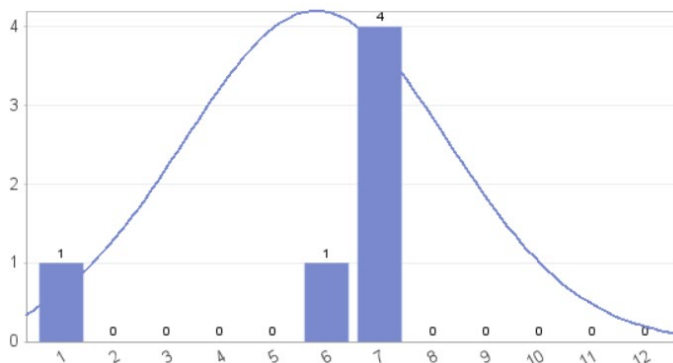
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluettu keskiarvosta



Arvostelijat						Passiiviset		Suositelijat		En tiedä	
0	1	2	3	4	5	6	7	8	9	10	
n = 6						n = 0		n = 0			
100,0%						0,0%		0,0%			
1	0	0	0	0	1	4	0	0	0	0	1
16,7%	0,0%	0,0%	0,0%	0,0%	16,7%	66,6%	0,0%	0,0%	0,0%	0,0%	14,3%

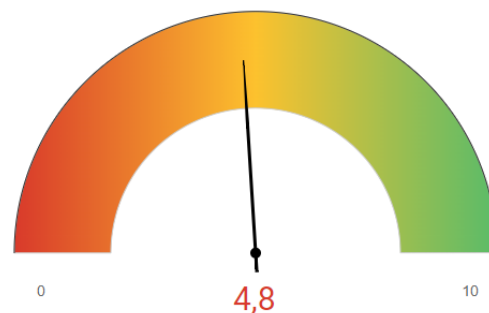
Yhteensä		
Vastauksia	NPS	Keskiarvo
6	-100	4,8



### Pienet tuotantoerät

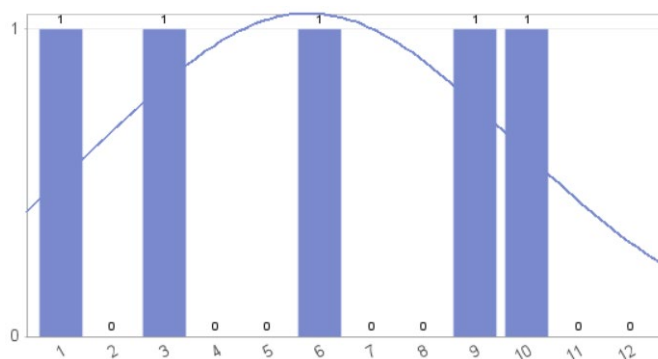
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluettu keskiarvosta



Arvostelijat						Passiiviset		Suositelijat		En tiedä	
0	1	2	3	4	5	6	7	8	9	10	
n = 3						n = 1		n = 1			
60,0%						20,0%		20,0%			
1	0	1	0	0	1	0	0	1	1	0	2
20,0%	0,0%	20,0%	0,0%	0,0%	20,0%	0,0%	0,0%	20,0%	20,0%	0,0%	28,6%

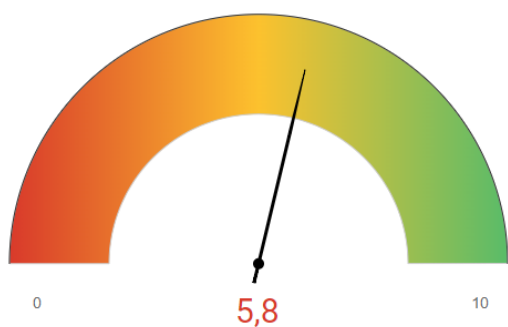
Yhteensä		
Vastauksia	NPS	Keskiarvo
5	-40	4,8



**Suuret tuotantoerät**

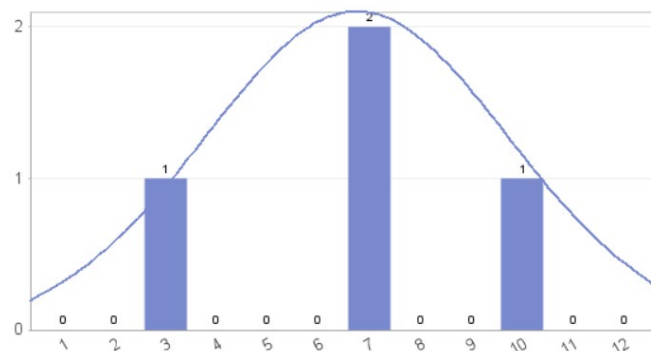
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluettu keskiarvosta



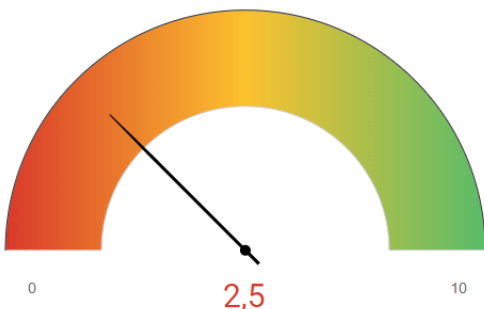
Arvostelijat							Passiiviset		Suositellijat		En tiedä
0	1	2	3	4	5	6	7	8	9	10	
n = 3							n = 0		n = 1		
75,0%							0,0%		25,0%		
0	0	1	0	0	0	2	0	0	1	0	3
0,0%	0,0%	25,0%	0,0%	0,0%	0,0%	50,0%	0,0%	0,0%	25,0%	0,0%	42,9%

Yhteensä		
Vastauksia	NPS	Keskiarvo
4	-50	5,8

**3D-kuvantaminen**

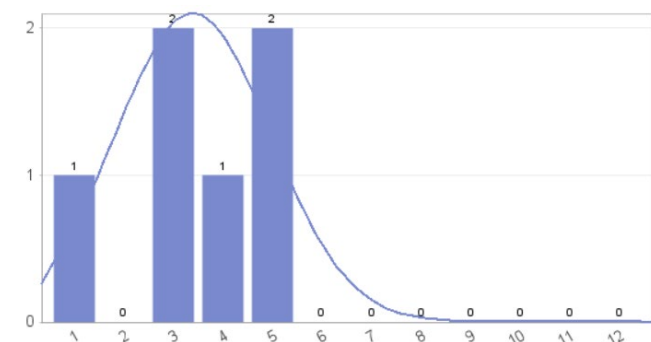
Vastaajien määrä: 7

Vastausvaihtoehto: En tiedä - poisluettu keskiarvosta



Arvostelijat							Passiiviset		Suositellijat		En tiedä
0	1	2	3	4	5	6	7	8	9	10	
n = 6							n = 0		n = 0		
100,0%							0,0%		0,0%		
1	0	2	1	2	0	0	0	0	0	0	1
16,7%	0,0%	33,3%	16,7%	33,3%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	14,3%

Yhteensä		
Vastauksia	NPS	Keskiarvo
6	-100	2,5



**Ovatko koneoppimiseen pohjautuvat järjestelmät korvannet perinteisiä konenäköjärjestelmiä jollain muulla osa-alueella?**

Vastaajien määrä: 6

Vastaukset

AI:lla on oma lokeronsa monimutkaisissa vikojen tunnistus, luokittelu ja hahmontunnistus projekteissa, mutta se ei kykene korvaamaan vielä perinteistä konenäköä toistettavuutta ja tarkkuutta vaativissa sovelluksissa.
-
Visuaalinen tarkastus on nyt AI-pohjaisia ja myös ei-mitattavat suureet, joissa muoto tai hahmo on tärkeä
Anomaly detection on selkeä trendi. Haetaan sitä promillen promillen muutosta esim. laaduntarkkailussa.
Turvallisuus
Tuotantolaitteen osien kunnonvalvonnassa koneoppimiseen pohjautuvilla ratkaisulla on menestyksekkäästi korvattu perinteisiä konenäköä menetelmiä.

### Mitä hyötyä koneoppimiseen pohjautuvista järjestelmistä on suunnittelun ja käyttöönoton kannalta?

Vastaajien määrä: 7

Vastaukset
Aivan viimeisimmät neuroverkkopohjaiset ohjelmistot ovat niin helppokäyttöisiä, että eivät vaadi käyttäjältä juurikaan ymmärrystä konenäön osa-alueelta. Eli aika vähäisellä osaamisella, voi tehdä melko haastaviakin sovelluksia hyödyntäen AI-tekniologiaa.
Tiedetään entuudestaan mitä kappeleita on pystytty opettamaan ja mitkä tuottavat ongelmia. Jos samalaisia järjestelmiä kopioidaan ei tarvitse tehdä työ kuin kerran ja optimoida tarvittaessa ympäristön mukaan.
Nopea toteuttaa, mutta vielä hieman hidas opettaa ja ikävien muotojen kohdalla saattaa arpoa liian pitkään ennen kuin on varma tuloksesta.
Loppuasiakkaan mahdollinen helpompi jatko konfigurointi sovelluksen muuttuessa vuosien mittaan. Mikäli on itse opetettavissa.
Joustavuus, adaptiivisuus,
Yksinkertaisempi sovellus joka on myös nopeammin valmis, kun oppiva järjestelmä korvaa koodin manuaalista rakentamista. Toisaalta ongelmana voi olla datan keräys ja annotointi.
Kuvausympäristöä ei tarvitse suunnitella ja virittää kovin tarkasti, eikä kuvista laskettavia piirteitä tarvitse suunnitella. Käyttöönoton kannalta ei juuri eroa.

### Miten koneoppimisen hyödyntäminen on vaikuttaneet asiakastytyväisyyteenne ja kilpailuunne verrattuna perinteisiin menetelmiin?

Vastaajien määrä: 7

Vastaukset
AI kiinnostaa asiakkaita, mutta varsinaisesti ei ole vaikuttanut vielä merkittävästi.
-
Pystytään päivittämään perinteisellä teknologialla toteutettuja järjestelmiä paremmiksi ja varmemmiksi.
Helpompi ymmärtää vahvuudet, kuin perinteisten systeemien heikkoudet. Asiakastytyväisyys samaa luokkaa.
Ei vielä oikeastaan ole vaikuttanut. Tutkimus ja hyödynnettävyys alkuvaiheessa.
Positiivisesti. Mahdollistaa sellaisia sovelluksia, jotka olisi hyvin vaikea saada toimimaan perinteisellä konenäöllä. Koneoppimista haluttaisiin hyödyntää vähän joka paikassa mainos-arvon takia.
Asiakastytyväisyys on ollut molemmilla menetelmillä hyvä. Koneoppimisen avulla on voitu tarjota konenäköratkaisuja myös kohteisiin, joissa perinteisillä menetelmillä ei olisi saavutettu järkevissä ajassa järkeviä tuloksia.

### Millaisia suunnitelmia teillä on koneoppimiseen pohjautuvien konenäköjärjestelmien hyödyntämiselle tulevaisuudessa? Mihin uusiin käyttökohteisiin aiotte näitä teknologioita soveltaa?

Vastaajien määrä: 7

Vastaukset
Nyt ollaan siinä vaiheessa kehityksessä, että edulliset ja helpot AI järjestelmät tulevat "koko kansan" saataville. Eli tätä menee aivan kaikkialle.
Komponentti myynnin puolesta pystytään laajentamaan eri asiakaskohteisiin. Perinteisen konenäköanalyysin teko vaatii yleensä paljon kokemusta/softaosaamista mutta kaupalliset koneoppimisohjelmit mahdollistavat helpomman käyttöönoton. Tämän takia monet loppukäyttäjätkin voivat opettaa helpommin omat tuotteensa ja tehdä nopeammin analyysia. Sitten on monimutkaisemmat sovellukset mitä ei ennen olla pystytty toteuttamaan perinteisellä analyysillä.
Olipa hyvä kysymys. En ole ennustaja.
Koneoppivat järjestelmät ja niiden päättelykyky tulee varmasti syrjäyttämään kaikki visuaalisen laadun, hahmon, mallin mukaan tehtävät tarkastukset. Systemit tulevat vielä ajan saatossa oppimaan näkemäänsä itse.
Kappaletavaratuotannossa poikkeuksien havaitseminen. AI pohjainen OCR selkeä trendi myös.
Oletettavasti laajenee. Kohteisiin, jotka ovat aina hieman erilaisia, mutta joilla on kuitenkin yhteneväisiä piirteitä. Ja joissa olosuhteet muuttuvat, jolloin perinteisen konenäön lähetymistavat nopeasti rajoittavat toiminnallisuutta.
Erilaiset laadun- tai kunnonvalvontaan päällimmäisinä. Kokeilussa myös erilaisten jatkuvien prosessien valvonta, jossa ei haeta yksittäisiä vikoja/ongelmia, vaan yleisempää arvioita prosessin tilasta.
Konenäköjärjestelmiä ja laadun-/kunnonvalvonnan automaatiota myydään jatkossakin, ja näissä jatketaan koneoppimisen hyödyntämistä osa-alueilla joissa siitä on ilmiselvästi hyötyä. Uusimpana käyttökohteena koneoppimiselle on tuotantoympäristöjen turvallisuutta tarkkailevat järjestelmät.

### Onko aiheesta tai kyselystä muuta sanottavaa?

Vastaajien määrä: 4

Vastaukset
AI teknologiaa valmistavien yritysten vuosien hypetyksessä on saanut aikaan sen, että AI:ta yritetään äängetä paikkoihin, joihin se ei kuulu. Toivotaan, että ei tule kovin suuria pettymyksiä.
-
Toivon huomattavasti läheisempää yhteistyötä jatkossa yrityksen ja JAMKin välillä. Meillä on tarpeita löytää oikean tyyppisiä henkilöitä tekemään maailmaa paremmaksi.
Tähän kyselyyn vastaaminen oli toimenkuvani huomioon ottaen jokseenkin hankalaa.