

# Taulukoiden tunnistus osana OCR-pipelinea

TURUN YLIOPISTO  
Tietotekniikan laitos  
TkK-tutkielma  
Syyskuu 2025  
Ilari Larkiola

TURUN YLIOPISTO  
Tietotekniikan laitos

ILARI LARKIOLA: Taulukoiden tunnistus osana OCR-pipelinea

TkK-tutkielma, 26 s.

Syyskuu 2025

---

Tämä tutkimus keskittyy taulukoiden tunnistukseen osana OCR-pipelinea historiallisista dokumenteista. Tutkimuksessa keskityttiin erikseen printattuihin ja käsinpiirrettyihin taulukoihin. Tutkimuksen tavoitteena oli selvittää, kuinka hyvin taulukoiden automaattinen tunnistus onnistuu käyttämällä YOLO11-malliversioihin perustuvia koneoppimismalleja sekä kuinka mallin koko ja koulutusparametrit, kuten epokkien määrä ja neuroverkon syvyys vaikuttavat mallin suorituskykyyn. Aineistona käytettiin Suomen Sukuhistoriallisen Yhdistyksen digitoituja muuttokirjoja, jotka sisälsivät rakenteellisesti vaihtelevia ja visuaalisesti haastavia dokumentteja. Käsin annotoitu aineisto jaettiin kolmeen luokkaan: sivun otsikko, taulukon otsikoriivi ja itse taulukko, mikä mahdollisti mallien kouluttamisen ja niiden suorituskyvyn vertailun eri datatyyppeiden välillä. Mallien tarkkuuteen vaikuttivat merkittävästi koulutusparametrit ja aineiston rakenne. Parhaat tulokset saavutettiin 100 epokin koulutuksella, ja kevyempi YOLO11s-malli osoittautui hyväksi ja resurssitehokkaaksi vaihtoehdoksi, vaikka suurempi YOLO11l-malli tarjosi tietyissä tilanteissa hieman paremman tarkkuuden. Printatulla datalla koulutettu malli kykeni osittain ennustamaan myös käsinpiirrettyä dataa, erityisesti varsinaisen taulukon osalta. Tulokset viittaavat siihen, että hyvin konfiguroidut YOLO-pohjaiset ratkaisut voivat tukea tehokasta taulukoiden tunnistusta ja tiedon keruuta vaihtelevista dokumenttimuodoista osana OCR-järjestelmiä.

Asiasanat: taulukoiden tunnistus, OCR, YOLO, koneoppiminen

# Sisällys

<b>1</b>	<b>Johdanto</b>	<b>1</b>
<b>2</b>	<b>Kirjallisuuskatsaus</b>	<b>3</b>
2.1	OCR-tekniikan perusteet . . . . .	3
2.2	Taulukoiden ja sarakkeiden tunnistus OCR-pipelissä . . . . .	4
2.3	Haasteet taulukoiden tunnistamisessa . . . . .	6
2.4	Aiemmat tutkimukset ja menetelmät . . . . .	7
<b>3</b>	<b>Tutkimusmenetelmät</b>	<b>9</b>
3.1	Data: Printatut ja käsinpiirretyt taulukot . . . . .	9
3.2	Annotointiprosessi . . . . .	11
3.3	Mallin koulutus ja testaus . . . . .	15
3.4	Arviointimenetelmät ja -kriteerit . . . . .	16
<b>4</b>	<b>Tulokset</b>	<b>18</b>
4.1	Mallin suorituskyky printatulla datalla . . . . .	18
4.2	Mallin suorituskyky käsinpiirretyllä datalla . . . . .	20
4.3	Printatulla datalla koulutetun mallin suorituskyky käsinpiirretylle datalle . . . . .	22
<b>5</b>	<b>Yhteenveto</b>	<b>24</b>



# Kuvat

2.1	Kuvassa vasemmalla taulukon tunnistaminen (TD) ja oikealla taulukon rakenteen tunnistaminen (TSR). . . . .	5
3.1	Vasemmalla printattu taulukko, oikealla käsinpiirretty. . . . .	11
3.2	Annotoitu printattu-taulukko. . . . .	13
3.3	Annotoitu käsinpiirretty-taulukko, ilman tableheader-luokkaa. . . . .	13
3.4	Annotoitu käsinpiirretty-taulukko, tableheader-luokalla. . . . .	14
4.1	YOLO11s mallin ennustamia lukemia printatulle testi-aineistolle. Parametrit: 100-epokkia, batch size 16, AdamW-optimointi. . . . .	20
4.2	Printatulla datalla koulutettu YOLO11s mallin ennustamia lukemia käsinpiirretylle testi-aineistolle. Parametrit: 100-epokkia, batch size 16, AdamW-optimointi. . . . .	23

# Taulukot

4.1	YOLO11s- ja YOLO11l-mallien tulokset eri epokkimäärillä. . . . .	19
4.2	YOLO11s- ja YOLO11l-mallien suorituskyky käsinpiirretyllä datalla.	21

# 1 Johdanto

Optinen merkkien tunnistus (OCR, Optical Character Recognition) on kehittynyt merkittävästi viime vuosikymmeninä, ja sen sovellukset ovat laajentuneet monille eri aloille. Taulukoiden tunnistus on erityisen haastava osa OCR-prosessia, koska taulukot sisältävät rakenteellista tietoa, joka on säilytettävä tarkasti [1]. Tässä työssä tarkastellaan taulukoiden tunnistuksen historiaa, nykytilaa, menetelmiä sekä haasteita ja mahdollisia tulevaisuuden ratkaisuja.

Taulukot ovat yleinen tapa esittää tietoa järjestelmällisesti. Niiden tunnistaminen ja käsittely OCR-prosessissa on tärkeää, jotta taulukoiden sisältämä tieto voidaan hyödyntää tehokkaasti. Taulukoiden tunnistaminen sisältää sekä taulukon rakenteen että sisällön tunnistamisen ja säilyttämisen. Nykyiset taulukoiden tunnistusmenetelmät vaihtelevat yksinkertaisista heuristisista menetelmistä kehittyneisiin koneoppimismalleihin. Esimerkiksi perinteiset OCR-järjestelmät käyttävät usein sääntöpohjaisia lähestymistapoja, kun taas nykyaikaiset menetelmät hyödyntävät syväoppimista ja konvoluutioverkkoja (CNN) taulukoiden rakenteen ja sisällön tunnistamiseen. [2] [3]

Taulukoiden tunnistuksessa on useita haasteita, kuten vaihtelevat asettelut, monimutkaiset rakenteet ja heikkolaatuiset kuvat. Lisäksi käsinpiirretyt taulukot tuovat omat haasteensa, koska esimerkiksi käsiala voi vaihdella suuresti. [3] [4]

Vaikka taulukoiden tunnistuksessa on saavutettu merkittäviä edistysaskeleita, on vielä monia tutkimusalueita, jotka kaipaavat lisätutkimusta. Esimerkiksi adaptiiviset

menetelmät, jotka pystyvät käsittelemään erilaisia taulukkoasetteluja ja -tyylejä, ovat lupaava tutkimussuunta. [4]

Tässä työssä luvussa 2 esitetään kirjallisuuskatsaus, jossa käydään läpi mm. OCR-tekniikan perusteita sekä haasteita taulukoiden tunnistuksessa. Luvussa 3 esitellään kokeellisen osan tutkimusmenetelmät, sisältäen käyttämäni data, annotointiprosessi, mallin koulutus ja testaus sekä arviointimenetelmät. Luvussa 4 käyn läpi kokeellisen tutkimukseni tuloksia. Luvussa 5 on työn yhteenveto.

Tutkimuskysymykseni ovat seuraavat:

1. Mitkä ovat yleisimmät haasteet taulukoiden tunnistuksessa OCR- pipelineissa?
2. Miten hyvin printatulla datalla koulutettu malli toimii käsinpiirretyllä datalla?
3. Kuinka neuroverkkomallin koko (mallin parametrien määrä) vaikuttaa tuloksiin?

## 2 Kirjallisuuskatsaus

### 2.1 OCR-tekniikan perusteet

Optinen merkkientunnistus (OCR, Optical Character Recognition) on teknologia, joka muuntaa skannattuja kuvia tai asiakirjoja muokattavaan digitaaliseen muotoon. Tämä prosessi mahdollistaa tekstin käsittelyn ja analysoinnin laskennallisesti, mikä on olennaista asiakirjojen digitalisoinnissa. OCR-teknologia perustuu kolmeen päävaiheeseen: esikäsitteilyyn, merkkien tunnistukseen ja jälkikäsitteilyyn. Esikäsitteilyssä asiakirjakuvat optimoidaan tunnistamista varten, merkkien tunnistus vaihtelee konekirjoitetun ja käsinkirjoitetun tekstin mukaan, ja jälkikäsitteilyssä tunnistettu teksti muokataan haluttuun muotoon. [2] [3]

OCR-teknologia sai alkunsa 1950-luvulla, jolloin skannereita käytettiin kirjojen ja dokumenttien muuntamiseen kuviksi. Varhaiset OCR-järjestelmät pystyivät lukemaan vain rajallisen määrän ennalta määritettyjä merkkejä, ja niiden tarkkuus oli huono. 1960-luvulla kehitettiin uusia algoritmeja ja tietokonenäkötekniikoita, jotka paransivat OCR:n tarkkuutta ja laajensivat tunnistettavien merkkien määrää. [3]

1970-luvulla Ray Kurzweil kehitti monifonttisen OCR-järjestelmän, joka pystyi tunnistamaan useita eri fontteja. Tämä järjestelmä kaupallistettiin ja myytiin myöhemmin Xeroxille. 1980-luvulla kehitettiin älykäs merkkien tunnistus (ICR) ja magneettimustetunnistus (MICR), jotka mahdollistivat käsinkirjoitetun tekstin ja pankkisekkeihin painettujen magneettisten merkkien tunnistamisen. [3]

1990-luvulla henkilökohtaisten tietokoneiden ja internetin yleistymisen johti OCR-tekniikan laajaan käyttöön. OCR-järjestelmiä käytettiin kirjojen, lehtien ja muiden painettujen materiaalien digitalisointiin. 2000-luvun alussa OCR-tekniikka kehittyi edelleen uusien algoritmien ja parannetun laitteiston myötä. Nykyään OCR-järjestelmät pystyvät tunnistamaan laajan valikoiman merkkejä ja kieliä, mukaan lukien käsinkirjoitusta ja monimutkaisia kuvia. [3]

## 2.2 Taulukoiden ja sarakkeiden tunnistus

### OCR-pipelinessa

Taulukoiden ja sarakkeiden tunnistus on erityisen haasteellista OCR-prosessissa, sillä se vaatii sekä visuaalista että rakenteellista asiakirjojen analyysia. Choi ja Jeong [2] esittävät artikkelissaan menetelmiä, joilla taulukot ja tekstialueet voidaan tunnistaa tehokkaasti, mikä parantaa tekstintunnistuksen tarkkuutta. Menetelmät perustuvat siihen, että taulukot paikannetaan kuvassa ja OCR kohdistetaan juuri näihin kohtiin. Taulukoiden tunnistamisessa käytetään usein syviä neuroverkkoja, jotka pystyvät paikantamaan taulukoiden sisäiset rakenteet ja erottamaan ne muusta sisällöstä. [2]

Taulukoiden tunnistaminen (TD, Table Detection) ja taulukoiden rakenteen tunnistaminen (TSR, Table Structure Recognition) ovat kaksi keskeistä tehtävää OCR:ssä. TD keskittyy taulukoiden paikantamiseen kuvissa, TSR tunnistaa niiden sisäiset rakenteet, ja taulukoiden sisällön tunnistaminen (TCR, Table Content Recognition) poimii tekstisisällön taulukoista. Esimerkki taulukoiden tunnistamisesta, sekä taulukoiden rakenteen tunnistamisesta nähdään kuvassa 2.1. Kuvan taulukon pohja on otettu Wikipedia-artikkelista [5], mutta olen annotoinut taulukon, sekä rakenteet itse sen päälle. Nykyiset menetelmät, kuten Table Transformer [4], kohtaavat haasteita esimerkiksi pitkien taulukoiden käsittelyssä ja tyhjien solujen erottami-

nessa merkityksellisistä soluista. Anandin [4] tutkimuksessa esitetään innovatiivinen ratkaisu, joka parantaa taulukkoanalyysin ja -tunnistuksen suorituskykyä. [4]

Elektro-negatiivisuusero	Sidoksen ioniluonne prosentteina	Elektro-negatiivisuusero	Sidoksen ioniluonne prosentteina
0,1	0,5	1,7	51
0,2	1	1,8	55
0,3	2	1,9	59
0,4	4	2,0	63
0,5	6	2,1	67
0,6	9	2,2	70
0,7	12	2,3	74
0,8	15	2,4	76
0,9	19	2,5	79
1,0	22	2,6	82
1,1	26	2,7	84
1,2	30	2,8	86
1,3	34	2,9	88
1,4	39	3,0	89
1,5	43	3,1	91
1,6	47	3,2	92

Kuva 2.1: Kuvassa vasemmalla taulukon tunnistaminen (TD) ja oikealla taulukon rakenteen tunnistaminen (TSR).

OCR:n yhteydessä pipeline tarkoittaa prosessien sarjaa, jota sovelletaan kuvan käsittelyyn tekstin tunnistamiseksi ja poimimiseksi. Tässä on yksinkertaistettu kuvaus tyypillisestä OCR-pipeline-prosessista:

1. **Esikäsittely:** Kuva valmistellaan tekstin tunnistamista varten. Tämä voi sisältää vaiheet kuten kuvan muuntaminen harmaasävyksi, kohinan vähentäminen ja kuvan normalisointi.
2. **Tekstin havaitseminen:** Seuraavaksi etsitään kuvan alueet, jotka sisältävät tekstiä. Tämä tehdään usein koneoppimismalleilla, jotka tunnistavat tekstirivit tai -blokit.
3. **Tekstin segmentointi:** Kun tekstialueet on tunnistettu, ne jaetaan pienempiin yksiköihin, kuten sanoihin tai kirjaimiin. Tämä auttaa eristämään tekstin tunnistamista varten.

4. **Tekstin tunnistus:** Segmentoitu teksti syötetään tunnistusmalliin, joka muuntaa kuvan segmentit varsinaiseksi tekstiksi. Käytössä oleva OCR-malli tulkitsee merkkien muodot ja kääntää ne koneellisesti luettavaksi tekstiksi.
5. **Jälkikäsittely:** Viimeisessä vaiheessa tunnistettu teksti viimeistellään hienosäädöllä. Tämä voi sisältää oikoluvun, muotoilun ja tekstin järjestämisen rakenteelliseen muotoon.

Jokainen OCR-pipeline-prosessin vaihe on tärkeä tarkkuuden ja tehokkuuden varmistamiseksi. Pipeline-lähestymistapa mahdollistaa modulaarisuuden ja joustavuuden, mikä helpottaa OCR-järjestelmien kehittämistä ja ylläpitoa. [2]

## 2.3 Haasteet taulukoiden tunnistamisessa

Taulukoiden tunnistamiseen liittyy useita haasteita, kuten epäsäännölliset rakenteet, yhdistetyt solut ja monimutkaiset sisäkkäiset taulukot. Nämä tekijät vaikeuttavat tehokasta merkkientunnistusta, koska ne vaativat syvällisempää ymmärrystä tekstin ja rakenteiden suhteista. Aiemmat tutkimukset ovat Anandin [4] mukaan osoittaneet, että monet perinteiset OCR-mallit eivät pysty käsittelemään näitä erityistapauksia riittävän tarkasti, mikä luo tarpeen kehittää uusia lähestymistapoja ja malleja, kuten TC-OCR [4], jotka on suunniteltu nimenomaan taulukoiden tunnistamisen haasteita varten. [4]

Taulukon rajausta ja sen löytäminen kuvista on tunnistuksen perusta. Kuitenkin taulukon taustat (kuvassa) ja konteksti missä se esiintyy sekä hyvin erilaiset taulukkorakenteet aiheuttavat ongelmia. Seuraavaksi on esitetty yleisiä haasteita taulukon tunnistamisessa. Ensinnäkin tietyt kuvaan liittyvät tekijät vaikeuttavat taulukon havaitsemisesta. Tällaisia ovat epätasainen tai riittämätön valaistus, kuvan vääristymä esimerkiksi suoruuden osalta (kuva vinossa), epätarkkuus jne. Ne on poistettava kuvan esikäsittelyyn sopivilla työkaluilla. Toiseksi taulukkokuvien huono laatu,

mahdolliset päälletehdyt merkinnät, taulukon puuttuvat osat jne, vaikeuttavat taulukoiden rajojen erottamista. Kolmanneksi taulukoiden hyvin erilaiset tyypit ja rakenteet ovat ongelma niiden tunnistamisessa. Toisiinsa linkitetyt taulukot, useiden muuttujien väliset taulukot, ristiintaulukoinnit jne. Tähän ratkaisuna on usein vain laajan esikoulutusmateriaalin hyödyntäminen mallien opetuksessa. [6]

## 2.4 Aiemmat tutkimukset ja menetelmät

Aiemmat tutkimukset ovat keskittyneet erilaisiin menetelmiin taulukoiden ja tekstin tunnistamiseksi OCR-pipelineissa. Choi ja Jeong [2] esittävät innovatiivisia lähestymistapoja, jotka keskittyvät reaaliaikaiseen taulukoiden ja tekstialueiden tunnistamiseen asiakirjakuvista, parantaen näin perinteisten OCR-järjestelmien suorituskykyä [2]. Choi ja Jeong kuvaavat menetelmää, jossa taulukon löytämiseen kuvasta hyödynnetään Faster R-CNN algoritmia. Solujen sijainti ja koko etsitään segmentointiin soveltuvalla Mask R-CNN algoritmilla. Varsinainen solun teksti poimitaan lopuksi easyOCR-ohjelmalla. Choin ja Jeongin mukaan tehokas taulukon- ja tekstintunnistus on yhdistelmä syväoppimismalleja ja avoimen lähdekoodin ohjelmistoja [2]. TC-OCR-malli on myös merkittävä kehitysaskel, sillä se yhdistää useita koneoppimismalleja ja mahdollistaa tarkemman tunnistamisen ja rakenteen erottamisen, mikä tekee siitä lupaavan vaihtoehdon nykyisille OCR-ratkaisuille [4].

Anandin [4] tutkimuksessa ehdotetaan end-to-end-pipelinea, joka yhdistää seuraavat mallit: DETR, CascadeTabNet ja PP OCR v2. DETR (DEtection TRansformer) on end-to-end-transformeripohjainen ratkaisu objektien tunnistukseen. Se käyttää itsehuomiomekanismeja mallintaakseen vuorovaikutuksia sekvenssin elementtien välillä ja tuottaa suoraan joukon ennusteita raja-alueista sekä luokkamerkinnoistä. CascadeTabNet on edistynyt syväoppimisjärjestelmä, joka käsittelee taulukon tunnistuksen ja rakenteen tunnistuksen yhdellä mallilla. Se suorittaa pikselitason taulukon segmentoinnin ja solujen segmentoinnin, mikä mahdollistaa taulukon

rakenteellisen tiedon poimimisen. PP OCRv2 on suunniteltu saavuttamaan korkea tarkkuus ja laskennallinen tehokkuus käytännön OCR-sovelluksissa. Se käyttää Collaborative Mutual Learning (CML) ja Copy Paste -menetelmiä datan parantamiseen sekä PP-LCNet-verkkoa tekstin tunnistamiseen.

Anandin [4] tutkimuksessa suoritettiin vertailuanalyysi ehdotetun mallin ja Table Transformer (TATR) -mallin välillä TableBank-datasetillä. Tulokset osoittivat, että ehdotettu malli on TATR:ää parempi sekä tarkkuudessa että tehokkuudessa. Malli saavutti vaikuttavan 0.96 IoU:n (Intersection Over Union, IoU:n käyttö kuvataan luvussa 3.4) ja 78 %:n OCR-tarkkuuden, mikä on noin 25 % parannus OCR-tarkkuudessa verrattuna TATR:ään. [4]

Kashinathin [7] artikkelissa esitetään ratkaisu, joka perustuu YOLO V3 (You Only Look Once) -objektintunnistus-malliin taulukoiden tunnistamiseksi ja poimimiseksi skannatuista asiakirjoista. YOLO V3 on objektintunnistus-malli, joka käsittelee kuvia reaaliajassa riittävän nopeasti suhteessa tarkkuuteen, jolloin monimutkaisetkin asiakirjat saadaan digitalisoitua. Hyvä tulos saavutetaan heidän mukaansa huolellisella datan annotoinnilla. Tällöin malli tunnistaa paitsi selkeästi määritellyt taulukot myös epästandardit asetelut, joissa perinteiset rivi- ja sarake-erottimet puuttuvat. Mallin tarkkuutta kuvaavat indeksit olivat seuraavat: PubLayNet-aineistossa malli saavutti keskimääräisen tarkkuuden (mAP - Mean Average Precision) 98 % ja keskimääräisen IoU:n 88,81 % 50 % IoU-kynnyksellä. Marmot-aineistossa keskimääräinen tarkkuus (mAP) oli 95,07 % ja IoU 75,57 %. [7]

## 3 Tutkimusmenetelmät

### 3.1 Data: Printatut ja käsinpiirretyt taulukot

Tutkimukseni kokeellisessa osassa käytän kahta eri tyyppistä dataa: printattuja taulukoita sekä käsinpiirrettyjä taulukoita. Kuvassa 3.1 esitetään esimerkki printatusta sekä käsinpiirretystä taulukosta. Printatussa taulukossa on painettu pohja, jossa on mukana taulukoiden rajat. Käsinpiirretyssä taulukossa rajat on piirretty käsin, esimerkiksi viivoittimen avulla. Näiden taulukoiden lähdeaineisto on peräisin Suomen Sukuhistoriallinen Yhdistys ry:n [8] julkisista kirkonkirjoista, erityisesti suomalaisista muuttokirjoista.

#### Printatut taulukot

Printatut taulukot ovat valittu seuraavista lähteistä:

- Porvoo, Ruotsalainen tuomiokirkkoseurakunta, muuttaneet 1880–1885 (MKO146–147, kuvat 3–45)
- Tornio, muuttaneet 1881–1883 (MKO11, kuvat 1–12)
- Alavus, muuttaneet 1880–1884 (MKO517–518, kuvat 1–35)

#### Aineiston jako:

#### Koulutus-aineisto:

Porvoo (MKO146–147, kuvat 3–23),

Tornio (MKO11, kuvat 1–12),

Alavus (MKO517–518, kuvat 1–19, 33–35)

Yhteensä 55 kuvaa.

**Validointi-aineisto:**

Porvoo (MKO146–147, kuvat 24–27, 32–38),

Alavus (MKO517–518, kuvat 20–28)

Yhteensä 20 kuvaa.

**Testaus-aineisto:**

Porvoo (MKO146–147, kuvat 28–31, 39–45),

Alavus (MKO517–518, kuvat 29–32)

Yhteensä 15 kuvaa.

## Käsinpiirretyt taulukot

Käsinpiirrettyjen taulukoiden lähteet:

- Porvoo, Ruotsalainen tuomiokirkkoseurakunta, muuttaneet 1869–1879 (MKO118–122, kuvat 2–67)
- Pyhämaa, muuttaneet 1864–1887 (MKO93, kuvat 1–24)

**Aineiston jako:**

**Koulutus-aineisto:**

Porvoo (MKO118–122, kuvat 2–29, 56–67),

Pyhämaa (MKO93, kuvat 1–15)

Yhteensä 55 kuvaa.

**Validointi-aineisto:**

Porvoo (MKO118–122, kuvat 30–37, 42–48),

Pyhämaa (MKO93, kuvat 16–20)

Yhteensä 20 kuvaa.

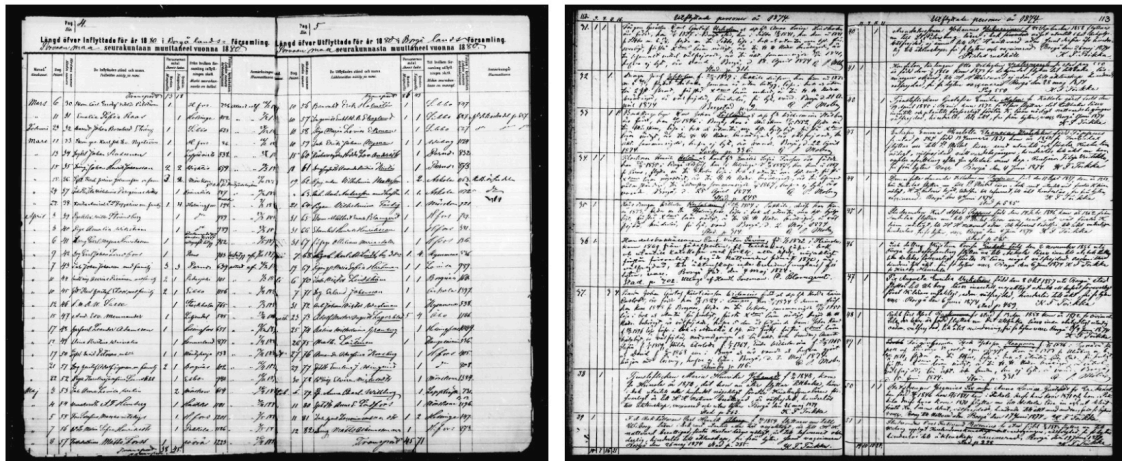
### Testaus-aineisto:

Porvoo (MKO118–122, kuvat 38–41, 49–55),

Pyhämaa (MKO93, kuvat 21–24)

Yhteensä 15 kuvaa.

Tämä aineistojako mahdollistaa tasapainoisen koulutus-, validointi- ja testausvaiheen analyysin ja kehittämisen kannalta.



Kuva 3.1: Vasemmalla printattu taulukko, oikealla käsinpiirretty.

## 3.2 Annotointiprosessi

Tutkimuksen kannalta keskeinen vaihe oli kuvatiedostojen annotointi eli kuvissa esiintyvien rakenteiden merkitseminen koneoppimismallin koulutusta varten. Annotointi suoritettiin Roboflow verkkopohjaisella työkalulla [9], joka on suunniteltu erityisesti konenäköprojekteihin (computer-vision). Roboflow tarjoaa intuitiivisen käyttöliittymän, jossa käyttäjä voi merkitä kuviin halutut kohteet (bounding box) ja

määrittää niille luokat. Työkalu tukee myös datasetin jakamista koulutus-, validointi- ja testiosuuksiin, sekä mahdollistaa automaattisen datan esikäsittelyn (esim. kuvan koon muuttaminen tai augmentointi) mitä ei tässä tutkimuksessa käytetty.

Tutkielmassa käytetty aineisto koostui dokumenttikuvista, jotka jaettiin kolmeen eri datasettiin:

1. **Printattuja** taulukoita sisältävä datasetti
2. **Käsinpiirrettyjä** taulukoita sisältävä datasetti
3. **Yhdistelmä datasetti**, jossa koulutus- ja validointiaineistona toimii printatut taulukot, mutta testiaineistona on käsinpiirrettyjä taulukoita

Tämän jaottelun avulla pyrittiin selvittämään mallin kykyä toimia eri dokumenttityylien kanssa eli tunnistaako se taulukkomuotoisia rakenteita myös silloin, kun visuaalinen tyyli eroaa selvästi koulutusdatasta. Annotointityössä käytettiin kolmea luokkaa:

1. **Header** (sivun otsikko)
2. **Tableheader** (taulukon otsikkorivi)
3. **Table** (varsinainen taulukko)

Nämä luokat valittiin niiden visuaalisuuden ja merkityksensä perusteella. Eri-tyisesti dokumenttien rakenteen tunnistamisen kannalta on tärkeää, että otsikko, taulukon otsikkorivi sekä taulukon sisältö erotetaan toisistaan. Kuvissa 3.2, 3.3 ja 3.4 esitetään annotoituja taulukoita. Keltaiset taulukot ovat sivun otsikko-luokkaa, punaiset taulukot kuvaavat taulukon otsikko-luokkaa ja liljat taulukot varsinaista taulukko-luokkaa.

The image shows two printed tax tables from Finland, side-by-side. Each table has a header with the year and the name of the parish (Borgå Länets församling). The tables contain columns for taxpayer names, addresses, and various tax amounts. The text is in Swedish. The tables are titled 'Längd öfver Uflyttade för år 18 58 i Borgå Länets församling' and 'Längd öfver Uflyttade för år 18 57 i Borgå Länets församling'. The tables are filled with handwritten entries.

Kuva 3.2: Annotoitu printattu-taulukko.

The image shows two handwritten tax tables from Finland, side-by-side. Each table has a header with the year and the name of the parish (Borgå Länets församling). The tables contain columns for taxpayer names, addresses, and various tax amounts. The text is in Swedish. The tables are filled with handwritten entries. The tables are titled 'Längd öfver Uflyttade för år 18 58 i Borgå Länets församling' and 'Längd öfver Uflyttade för år 18 57 i Borgå Länets församling'.

Kuva 3.3: Annotoitu käsinpiirretty-taulukko, ilman tableheader-luokkaa.

The image shows two handwritten tables from 1853. The left table is titled "Pääosan kirjastoarkistot" and "Pääosan muuttaneiden kirjall. vuosi 1853". The right table is titled "Ulkos. muuttaneiden kirjall. vuosi 1853". Both tables have columns for names, dates, and other details, with handwritten entries and some red annotations.

Kuva 3.4: Annotoitu käsinpiirretty-taulukko, tableheader-luokalla.

Manuaalinen annotointi oli työläs mutta välttämätön osa prosessia. Käsin tehty raja- ja luokkajaukset mahdollistaa suuremman tarkkuuden verrattuna automaattisiin menetelmiin, erityisesti kun halutaan selkeästi eroteltuja ja luotettavia merkintöitä. Annotointityön tarkkuudella on merkittävä vaikutus mallin oppimistuloksiin, sillä huolimattomasti tai epätarkasti merkitty aineisto voi heikentää mallin suorituskäytännön [10]. Tämän vuoksi kuvat käytiin huolellisesti läpi, ja erityistä huomiota kiinnitettiin siihen, että luokkarajaukset olivat johdonmukaisia koko datasetissä.

Annotointiprosessin aikana havaittiin, että erityisesti käsinpiirretyissä dokumenteissa luokkien rajojen määrittäminen oli ajoittain haastavaa: esimerkiksi taulukon otsikko ja itse taulukko voivat sulautua visuaalisesti toisiinsa. Tämä lisäsi tarvetta tehdä harkittuja valintoja siitä, missä raja kulkee.

### 3.3 Mallin koulutus ja testaus

Mallin koulutuksessa käytettiin YOLOv11-malliperhettä [11] (You Only Look Once), joka on yksi tunnetuimmista reaaliaikaisiin objektintunnistustehtäviin soveltuvista syväoppimismalleista. Valinta YOLO-malleihin perustui kahteen pääasialliseen tekijään: aiempi kokemus mallien käytöstä sekä niiden käyttöönoton ja koulutuksen helppous yhdessä hyvän dokumentaation kanssa. Näiden ansiosta mallien kouluttaminen ja testaaminen eteni mutkattomasti.

Tutkimuksessa testattiin kahta eri malliversiota:

1. **YOLOv11s** (small), kevyempi ja nopeampi malli, joka soveltuu erityisesti tilanteisiin, joissa laskentatehoa on rajoitetusti
2. **YOLOv11l** (large), joka tarjoaa paremman tarkkuuden, mutta suuremman mallikoonsa vuoksi vaatii enemmän resursseja

Alkuperäiseen suunnitelmaan kuului myös YOLOv11x-mallin koulutus, mutta käytössä olleen laitteiston rajoitukset estivät sen käytön. YOLOv11x vaatii suuremman määrän GPU-muistia kuin mitä oli saatavilla.

Mallin koulutus toteutettiin Google Colab [12] -ympäristössä, jossa käytössä oli NVIDIA Tesla T4 GPU. T4 tarjoaa kohtuullisen suorituskyvyn pienille ja keskikokoisille malleille, mutta sen 16 Gt GPU-muisti ei riittänyt YOLOv11x-mallin kouluttamiseen. Colab-ympäristön etuna oli helppo pääsy GPU-laskentaan sekä helppo integraatio Python- ja YOLO-kirjastoihin.

Aineisto tuotiin Colabiin suoraan Roboflowista, joka mahdollistaa datasetin vienin yhteensopivassa muodossa (tässä tutkimuksessa YOLOv11). Tämän ansiosta datan esikäsittelyyn ei tarvittu lisätyökaluja. Koulutuksen aikana käytettiin YOLO:n omaa Python-kirjastoa, joka tarjoaa valmiit funktiot mallin koulutukseen, validointiin ja evaluointiin. Tämä korostaa YOLO-mallien käytön helppoutta erityisesti koulutuksessa ja testaamisessa. Laitteistosta ja ympäristöstä huolimatta YOLOv11s

ja YOLOv11l -mallit tarjosivat hyvän suorituskyvyn suhteessa niiden laskennallisiin vaatimuksiin. Mallien evaluoinnissa käytettiin mAP (mean Average Precision) -arvoa sekä tarkkuutta ja herkkyyttä (precision & recall), jotka antavat kattavan kuvan mallin toimivuudesta eri dataseiteillä.

### 3.4 Arviointimenetelmät ja -kriteerit

Mallin suorituskykyä arvioitiin konenäön tutkimuksessa yleisesti käytetyillä arviointikriteereillä, jotka mahdollistavat eri malliversioiden ja datasettien vertailun selkein ja mitattavin perustein. Käytetyt arviointikriteerit olivat seuraavat:

1. **Precision** (tarkkuus): Kuinka suuri osa mallin tekemistä tunnistuksista oli oikein
2. **Recall** (herkkyys): Kuinka suuren osan todellisista kohteista malli löysi
3. **mAP** (mean Average Precision): Keskimääräinen tarkkuus kaikkien luokkien yli, tietyllä tarkkuuskynnyksellä mitattuna. Tässä tutkielmassa käytettiin sekä mAP@50 että mAP@50–95 -arvoja

Näistä erityisesti mAP on laajasti käytetty suorituskykyindikaattori objektintunnistustehtävissä. Se perustuu IoU-arvoon (Intersection over Union), joka mittaa mallin ennustaman kohteen ja todellisen kohteen päällekkäisyyttä. IoU lasketaan jakamalla leikkaavan alueen pinta-ala kohteiden yhdistetyn alueen pinta-alalla [6]. Jos IoU-arvo ylittää tietyn kynnyksen (esimerkiksi 0.5), ennuste katsotaan onnistuneeksi. mAP mittaa, kuinka tarkasti ja kattavasti malli pystyy havaitsemaan kohteita eri IoU-kynnyksillä. mAP@50–95 viittaa keskimääräiseen tarkkuuteen, joka lasketaan kymmenen eri IoU-kynnyksen (0.50–0.95, 0.05:n askelvälein) perusteella. Tämä tekee siitä vaativamman ja informatiivisemmän mittarin kuin yksittäinen IoU-kynnys [13].

Kaikki arviointikriteerit laskettiin YOLOv11-kirjaston sisäänrakennetuilla evaluointityökaluilla. Tämä mahdollisti luotettavan ja yhdenmukaisen arvioinnin jokaisen koulutusepokin aikana. Mallin toimivuutta tarkasteltiin sekä validointi- että testidatalla, jotta voitiin havaita esimerkiksi ylisovittamista.

Lisäksi suorituskykyä arvioitiin visuaalisesti: mallin tekemät ennusteet piirrettiin kuviin bounding boxeina, jolloin oli mahdollista arvioida käytännön tarkkuutta ja rajojen täsmällisyyttä. Tämä visuaalinen arviointi täydensi numeerisia tunnuslukuja, ja tarjosi syvällisemmän käsityksen mallin toiminnasta erityisesti visuaalisesti vaihtelevissa aineistoissa. Yhdessä nämä arviointiperusteet muodostivat monipuolisen kuvan mallin luotettavuudesta ja soveltuvuudesta dokumenttirakenteiden automaattiseen tunnistamiseen.

## 4 Tulokset

### 4.1 Mallin suorituskyky printatulla datalla

Tutkimuksen ensimmäisessä vaiheessa arvioitiin kahden YOLO11-mallin, YOLO11s:n ja YOLO11l:n, suorituskykyä printatulla datalla, jossa taulukoiden rakenteet, kuten reunat ja otsikot olivat selkeästi visuaalisesti esillä. Mallien tehtävänä oli tunnistaa kolme eri taulukkotyyppiä: sivun otsikko (header), itse taulukko (table) sekä taulukon otsikko (tableheader). Käytetty aineisto sisälsi 90 kuvaa, joista 55 käytettiin koulutukseen, 20 validointiin ja 15 testaukseen. Vaikka testikuvia oli vain 15, sisälsi se kuitenkin 90 luokitusinstanssia. Kaikki koulutukset tehtiin kuvatarkkuudella  $640 \times 640$  ja AdamW-optimointialgoritmilla, joka määritteli oppimismopeuden automaattisesti.

Mallin suorituskykyyn vaikutti selvästi epokkien määrä. YOLO11s-mallilla testattiin sekä 50 että 100 epokin koulutusta. 50 epokin jälkeen malli saavutti mAP@50–95-arvon 0.95, ja inferenssiaika oli noin 3.8 millisekuntia. Vaikka kokonaisuudessa tulokset olivat hyvät, luokkakohtaisissa tarkkuuksissa näkyi heikompaa suorituskykyä erityisesti header-luokassa (mAP@50–95 = 0.91), ja konfuusiomatriisissa esiintyi kolme väärää luokitusta. Kun koulutusta jatkettiin 100 epokkiin, suorituskyky parani kaikilla mittareilla. Malli saavutti tällöin mAP@50–95-arvon 0.96 ja säilytti nopean inferenssiajan (4.1 ms). Vain yksi virheluokitus havaittiin koko testidatassa. Näiden tulosten perusteella 100 epokkia valittiin jatkokokeiden standardiksi, koska se tarjosi

selvästi paremman tasapainon tarkkuuden ja oppimisen välillä.

Samaa epokkien vaikutusta tutkittiin myös YOLO11l-mallilla, joka on selvästi suurempi ja sisältää noin 25,3 miljoonaa parametria. 50 epokin koulutuksella malli saavutti mAP@50–95-arvon 0.96 ja hyvän luokkakohtaisen tarkkuuden. Kuitenkin konfuusiomatriisi osoitti kaksi virheellistä luokitusta, joissa tableheader-instanssit sekoittuivat taustaan (background). Lisäksi validointituloksissa näkyi hetkittäistä epävakautta, mikä ilmeni erityisesti val / cls-loss-käyrän suurina piikkeinä. Kun koulutusta jatkettiin 100 epokkiin, YOLO11l:n tarkkuus nousi mAP@50–95-arvoon 0.96, eikä väärinluokituksia enää esiintynyt. Inferenssiaika pysyi melkein samana (11s - 13.3 ms, 11l - 13.8 ms), mutta tulokset olivat selvästi vakaampia ja luotettavampia, taulukko 4.1.

Malli	Epokit	mAP@50–95	Virheluokituksia	Inferenssiaika (ms)
YOLO11s	50	0.952	3	3.8
YOLO11s	100	0.963	1	4.1
YOLO11l	50	0.961	2	13.3
YOLO11l	100	0.968	0	13.8

Taulukko 4.1: YOLO11s- ja YOLO11l-mallien tulokset eri epokkimäärillä.

Yhteenvetona voidaan todeta, että sekä YOLO11s että YOLO11l saavuttivat erittäin hyviä tuloksia printatulla datalla. YOLO11l-malli tarjosi hieman korkeamman ennustetarkkuuden, mutta YOLO11s oli huomattavasti nopeampi ja resursitehokkaampi. Erityisesti 100 epokin koulutus paransi molempien mallien kykyä yleistää ja vähensi väärinluokituksia. Näiden havaintojen perusteella YOLO11s soveltuu hyvin tilanteisiin, joissa nopeus ja keveys ovat tärkeitä, kun taas YOLO11l on parempi valinta silloin, kun tarkkuus on kriittinen tekijä ja laskentaresurseja on riittävästi saatavilla.



Kuva 4.1: YOLO11s mallin ennustamia lukemia printatulle testi-aineistolle. Parametrit: 100-epokkia, batch size 16, AdamW-optimointi.

## 4.2 Mallin suorituskyky käsinpiirretyllä datalla

Tutkimuksen toisessa vaiheessa arvioitiin mallien kykyä suoriutua käsinpiirretystä datasta, joissa taulukoiden reunat eivät olleet yhtä selkeitä kuin printatuissa versioissa. Tarkoituksena oli selvittää, miten hyvin mallit kykenevät oppimaan piirteet visuaalisesti haastavammasta aineistosta. Mallien tehtävänä oli taas tunnistaa kolme eri taulukkotyyppiä: sivun otsikko (header), itse taulukko (table) sekä taulukon otsikko (tableheader), mutta tällä kertaa taulukon otsikko (tableheader) ei sisällynytään kaikkiin taulukoihin. Käytetty aineisto sisälsi 90 kuvaa, joista 55 käytettiin koulutukseen, 20 validointiin ja 15 testaukseen. Nämä 15 testikuvaa sisälsivät 68-luokitusinstanssia. Molemmat mallit, kevyempi YOLO11s ja suurempi YOLO11l, koulutettiin samoin parametrein: 100 epokkia, batch size 16, kuvatarkkuus  $640 \times 640$  ja AdamW-optimointi, jonka oppimisnopeus määrittyi automaattisesti.

YOLO11s-malli saavutti käsinpiirretyllä datalla kokonaisuudessaan mAP@50–

95-arvon 0.89, mikä osoittaa hyvää oppimiskykyä. Erityisesti table- ja tableheader-luokissa tarkkuus säilyi korkeana (0.99 ja 0.87), kun taas header-luokassa havaittiin hieman heikompi tulos (0.82). Konfuusiomatriisin mukaan header-luokan tapauksissa tapahtui kaksi virhettä, joissa se sekoitettiin taustaan, ja tableheader-luokassa yksi virheellinen tunnistus. Mallin inferenssiaika säilyi nopeana (4.0 ms), mikä vahvistaa sen sopivuutta nopeaa suorituskkyä vaativiin sovelluksiin, vaikkakin suorituskky ei täysin vastannut printatulla datalla saavutettuja tuloksia.

Vastaavasti YOLO11l-malli koulutettiin samalla käsinpiirretyllä aineistolla. Tämä suurempi malli saavutti hieman matalamman kokonaistarkkuuden mAP@50–95-arvolla 0.88. Luokkakohtaisesti tulokset vaihtelivat: header-luokka sai tarkkuudeksi 0.84, table 0.99 ja tableheader 0.81. Konfuusiomatriisi osoitti, että tableheader-luokassa tapahtui yksi virheluokitus, jossa se sekoitettiin taustaan. Kaikki kahdeksan todellista instanssia luokiteltiin muuten oikein. YOLO11l-mallin inferenssiaika oli selvästi pidempi (11.9 ms), mikä on luonnollinen seuraus sen suuremmasta parametrikooosta (noin 25,3 miljoonaa parametria), taulukko 4.2.

Malli	Epokit	mAP@50–95	Header	Table	Table-header	Virheluokituksia	Inferenssiaika (ms)
YOLO11s	100	0.895	0.820	0.995	0.870	3	4.0
YOLO11l	100	0.885	0.847	0.995	0.811	1	11.9

Taulukko 4.2: YOLO11s- ja YOLO11l-mallien suorituskky käsinpiirretyllä datalla.

Näiden tulosten perusteella voidaan päätellä, että molemmat mallit kykenevät oppimaan käsinpiirrettyihin taulukkoihin, mutta tarkkuus laskee hieman verrattuna printattuun aineistoon. YOLO11s tarjoaa nopeamman ja kevyemmän ratkaisun, joka toimii yllättävän hyvin myös haastavammassa visuaalisessa ympäristössä. YOLO11l tuottaa hieman tarkempia ennusteita tietyissä luokissa, mutta sen hyöty ei ole merkittävä suhteessa lisääntyneeseen laskentakustannukseen tässä kontekstissa.

### 4.3 Printatulla datalla koulutetun mallin suoriutumiskyky käsinpiirretylle datalle

Tutkimuksen viimeisessä vaiheessa selvitettiin, kuinka hyvin printatulla datalla koulutettu YOLO11s-malli kykeni ennustamaan oppimansa käsinpiirrettyyn dataan. Tarkoituksena oli arvioida, oliko malli oppinut pelkän visuaalisen tyylin vai todellisen taulukkorakenteen piirteitä. Tätä varten mallia testattiin täysin käsinpiirretyllä testijoukolla, johon kuului 15 kuvaa ja yhteensä 68 luokitusinstanssia. Käytössä oli aiemmin parhaiten suoriutunut YOLO11s-malli, joka oli koulutettu 100 epokilla ja batch size 16.

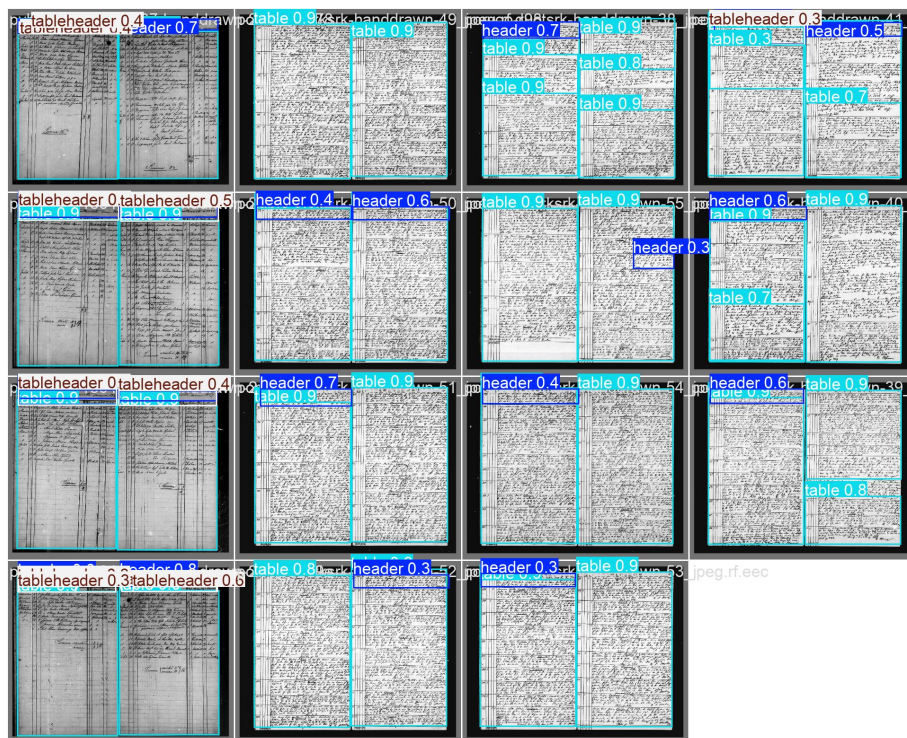
Mallin kokonaissuorituskyky testissä oli kohtalainen:  $mAP@50-95$  jäi arvoon 0.417, ja  $mAP@50$  oli 0.514. Luokkakohtaisissa tuloksissa nähtiin merkittäviä eroja. Table-luokka (eli varsinainen taulukko) suoriutui erittäin hyvin  $mAP@50-95$ -arvolla 0.896 ja täydellisellä recall-arvolla (1.0), mikä viittaa siihen, että mallin kyky tunnistaa taulukkorakenteen pääalue oli säilynyt erittäin vahvana huolimatta visuaalisten piirteiden muutoksista. Sen sijaan header- ja erityisesti tableheader-luokkien kohdalla suorituskyky heikkeni merkittävästi: tableheader-luokan  $mAP@50-95$  putosi arvoon 0.107.

Tähän vaikutti selvästi testidatan sisäinen vaihtelu: käsinpiirretty testijoukko sisälsi kahdentyyppisiä taulukoita, joista toisissa tableheader-luokkaa ei esiintynyt lainkaan. Koulutusvaiheessa malli oli kuitenkin nähnyt vain sellaisia dokumentteja, joissa tableheader oli aina mukana. Tällainen ristiriita taulukkorakenteissa saattoi osaltaan hämmentää mallia ja johtaa heikkoihin tuloksiin kyseisessä luokassa.

Konfuusiomatriisi vahvisti tämän havainnon. Suurin osa virheluokituksista liittyi siihen, että mallilla oli vaikeuksia erottaa tableheader-alueita muista luokista tai taustasta. Ennustekuvissa näkyi useita tapauksia, joissa tableheader-alueet jäivät kokonaan havaitsematta tai tunnistettiin matalalla ennustearvolla. On kuitenkin

huomattavaa, että vaikka mallin tarkkuus tietyillä luokilla heikkeni, sen kyky erottaa taulukkorakenne kokonaisuutena, erityisesti itse Table-luokka säilyi yllättävän hyvin.

Tämä osoittaa, että malli ei ollut täysin sidoksissa oppimaansa visuaaliseen tyyliin, vaan se oli oppinut taulukoiden rakenteellisia piirteitä. Samalla tulokset korostavat, että jos mallilta odotetaan korkeaa tarkkuutta käsinpiirretyissä taulukoissa, se vaatii ainakin osittaista altistusta kyseisenlaisen datan piirteille koulutusvaiheessa. Silti pelkästään printatulla datalla koulutettu YOLO11s-malli pystyi tuottamaan kohtuullisen tarkkoja ennustuksia myös käsinpiirretyllä datalla, mikä tekee siitä lupaavan pohjan jatkokehitykselle erityisesti sellaisissa OCR-pipelineissa, joissa dokumenttien visuaalinen tyyli vaihtelee merkittävästi.



Kuva 4.2: Printatulla datalla koulutettu YOLO11s mallin ennustamia lukemia käsinpiirretylle testi-aineistolle. Parametrit: 100-epokkia, batch size 16, AdamW-optimointi.

## 5 Yhteenveto

Tässä työssä käsiteltiin taulukoiden tunnistusta osana OCR-pipelinea. Työn tavoitteena oli selvittää, kuinka hyvin taulukoiden tunnistus onnistuu sekä printatusta että käsinpiirretystä datasta hyödyntäen YOLO11-malleja. Samalla tutkittiin, kuinka mallin koko ja koulutusparametrit, erityisesti neuroverkon syvyys (kerrokset ja parametrien lukumäärä) sekä epokkien lukumäärä vaikuttavat tuloksiin. Työ jakautui kahteen osioon: kirjallisuuskatsaukseen sekä kokeelliseen osaan.

Kirjallisuuskatsauksessa havaittiin, että taulukoiden tunnistaminen OCR-pipelineissa edellyttää erillisiä käsittelyvaiheita: taulukon paikantamista (TD), rakenteen tunnistamista (TSR) ja sisältötekstin poimintaa (TCR). Nykyiset lähestymistavat perustuvat pääasiassa syväoppimismalleihin, jotka yhdistävät objektin tunnistuksen, segmentoinnin ja tekstintunnistuksen yhtenäiseksi prosessiksi. Tutkimuksissa on osoitettu, että tällaiset ratkaisut parantavat erityisesti monimutkaisten ja rakenteellisesti vaihtelevien taulukoiden tunnistustarkkuutta. Nämä havainnot tukevat myös työn kokeellista osuutta, jossa todettiin, että erityisesti tableheader-luokka vaatii tarkkaa ja rakenteellisesti monipuolista opetusaineistoa hyvän suorituskyvyn saavuttamiseksi.

Kokeellisen osan ensimmäisenä vaiheena oli datan keruu ja sen annotointi. Annotointi on oleellista mallin koulutukselle ja sitä voidaan tässä tapauksessa pitää välttämättömyytenä eli sen vaikutusta ei tässä työssä selvitetty.

Printatulla datalla saavutettiin erittäin hyvä ennustetarkkuus sopivilla mallin

parametreilla. Riittävä epokkien määrä oli 100, ja mallin koolla ei ollut ratkaisevaa merkitystä, sillä sekä kevyempi YOLO11s- että suurempi YOLO11l-malli saavuttivat korkean  $mAP@50-95$ -arvon. Käsiniirretyllä datalla suorituskyky jäi hieman alemmaksi, mutta  $mAP@50-95$ -arvot olivat silti vahvoja (noin 0.9), erityisesti tableluokan osalta, jossa saavutettiin lähes täydellinen tarkkuus.

Huomionarvoista on, että kevyempi YOLO11s (238 kerrosta) suoriutui muutamassa tapauksessa jopa paremmin kuin laajempi YOLO11l (464 kerrosta). Suuri määrä kerroksia ja parametreja ei tuonut lisähyötyä ja saattoi heikentää mallin oppimiskykyä. Tämä voi johtua esimerkiksi ylioppimisesta, jossa malli oppii liiaksi opetusaineiston yksityiskohtia ja kohinaa, jolloin sen suoritus testidatalla heikkenee. Lisäksi suurempi verkko voi helpommin juuttua paikallisiin minimeihin koulutuksen aikana, mikä estää optimaalisen ratkaisun löytymisen.

Viimeisessä tutkimusvaiheessa tutkittiin mallin siirrettävyyttä: miten printatulla datalla koulutettu YOLO11s-malli suoriutuu käsiniirretyn datan testauksessa. Tulokset osoittivat, että malli kykenee oppimaan taulukoiden rakenteen tunnistamista ainakin osittain, erityisesti table-luokka tunnistettiin hyvin ( $mAP@50-95 = 0.896$ ;  $recall = 1.0$ ). Sen sijaan header- ja tableheader-luokissa tarkkuus heikkeni selvästi, mikä viittaa siihen, ettei pelkkä printatun datan oppiminen riitä monimuotoisten käsiniirretyjen rakenteiden tarkkaan tunnistamiseen. Lisäksi testidatassa oli mukana myös sellaisia taulukkoja, joissa tableheader-elementtiä ei esiintynyt ollenkaan kun taas koulutusdatassa se oli aina mukana. Tämä rakenteellinen ero todennäköisesti vaikutti erityisesti tableheader-luokan heikkoon suorituskykyyn.

On kuitenkin tärkeää huomioida, että käytetyt datasetit olivat määrältään pieniä. Tämän vuoksi kaikki saadut tulokset ovat suuntaa antavia, eivätkä ne yksinään riitä laajojen johtopäätösten tekemiseen esimerkiksi eri taulukkorakenteiden yleisestä siirrettävyydestä. Tulokset tarjoavat silti merkittäviä viitteitä siitä, että kevyempi YOLO11s on hyvä kompromissi tarkkuuden, nopeuden ja oppimisen välillä tässä

kontekstissa.

# Lähdeluettelo

- [1] S. Drobac, L. Sinikallio ja E. Hyvönen, ”OCR Pipeline for Transforming Parliamentary Debates into Linked Data: Case ParliamentSampo - Parliament of Finland on the Semantic Web”, *Digital Humanities in the Nordic and Baltic Countries Publications*, vol. 5, s. 287–296, lokakuu 2023. DOI: 10.5617/dhnbpub.10670.
- [2] H. Choi ja J. Jeong, ”Realtime detection of table objects and text areas for OCR preprocessing”, *WSEAS Transactions on Information Science and Applications*, vol. 20, s. 197–205, 2023.
- [3] J. Wang, ”A Study of The OCR Development History and Directions of Development”, *Highlights in Science, Engineering and Technology*, vol. 72, s. 409–415, 2023.
- [4] A. Anand et al., ”TC-OCR: TableCraft OCR for Efficient Detection & Recognition of Table Structure & Content”, teoksessa *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, 2023, s. 11–18.
- [5] Wikipedia contributors, *Elektronegatiivisuus – Wikipedia*, <https://fi.wikipedia.org/wiki/Elektronegatiivisuus>, Viitattu 24.6.2025, 2025.
- [6] C. Yu et al., ”A survey for table recognition based on deep learning”, *Neurocomputing*, vol. 600, 2024. DOI: 10.1016/j.neucom.2024.128154.

- 
- [7] T. Kashinath, T. Jain, Y. Agrawal, T. Anand ja S. Singh, ”End-to-end table structure recognition and extraction in heterogeneous documents”, *Applied Soft Computing*, vol. 123, toukokuu 2022. DOI: 10.1016/j.asoc.2022.108942.
- [8] Suomen Sukuhistoriallinen Yhdistys, *Kirkonkirjat paikkakunnittain*, <https://www.sukuhistoria.fi/sshy/kirjat/Kirkonkirjat/arkisto/paikkakunnittain.htm>, Viitattu 28.3.2025, 2025.
- [9] Roboflow, *Roboflow: Give your software the sense of sight*, <https://roboflow.com>, Viitattu: 28.3.2025, 2025.
- [10] J. F. Mullen, F. Tanner ja P. Sallee, ”Comparing the Effects of Annotation Type on Machine Learning Detection Performance”, teoksessa *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, kesäkuu 2019, s. 855–861. DOI: 10.1109/cvprw.2019.00114.
- [11] Ultralytics, *YOLOv11 Models Documentation*, <https://docs.ultralytics.com/models/yolo11/>, Viitattu: 28.3.2025, 2025.
- [12] Google, *Google Colaboratory*, <https://colab.research.google.com>, Viitattu: 28.3.2025, 2025.
- [13] I. Ahmad et al., ”Deep Learning Based Detector YOLOv5 for Identifying Insect Pests”, *Applied Sciences*, vol. 12, lokakuu 2022. DOI: 10.3390/app121910167.