OCR-tekniikoiden vertailu kiinan, korean ja japanin kielissä

Turun yliopisto
Tietotekniikan laitos
LuK-tutkielma
Tietojenkäsittelytiede
Toukokuu 2026
Eemeli Suojanen

TURUN YLIOPISTO
Tietotekniikan laitos

Eemeli Suojanen: OCR-tekniikoiden vertailu kiinan, korean ja japanin kielissä

LuK-tutkielma, 21 s.
Tietojenkäsittelytiede
Toukokuu 2026

Tämä tutkielma käsittelee optisen merkintunnistuksen (OCR) toimintaa kiinan, japanin ja korean kielissä keskittyen eri OCR-tekniikoiden suorituskykyyn ja niiden soveltuvuuteen monimutkaisissa merkkijärjestelmissä. Tutkielmassa tarkastellaan perinteisiä, koneoppivia ja syväoppivia OCR-menetelmiä sekä kiinan, japanin ja korean kielten (CJK) erityispiirteitä, kuten suuria merkkimääriä, visuaalisesti samankaltaisia merkkejä ja tekstin orientaation vaikutusta tunnistukseen. Lisäksi tutkielmassa vertaillaan OCR-järjestelmien suorituskykyä kirjallisuuskatsauksen perusteella hyödyntäen aiempien tutkimusten tuloksia ja vertailuja.
Tutkielmassa havaittiin, että OCR-järjestelmien tarkkuus riippuu merkittävästi käytetystä menetelmästä ja tunnistettavasta kielestä. Perinteiset ja koneoppivat menetelmät kärsivät erityisesti segmentointivirheistä, kun taas syväoppivat menetelmät kykenevät käsittelemään tekstiä kokonaisuutena ja saavuttavat tasaisemman suorituskyvyn eri kielissä. Lisäksi tekstin asettelu, kuvien laatu ja merkkien visuaalinen samankaltaisuus vaikuttavat merkittävästi tunnistuksen tarkkuuteen. Tulosten perusteella syväoppivat OCR-ratkaisut soveltuvat parhaiten CJK-kielten kaltaisten monimutkaisten kirjoitusjärjestelmien käsittelyyn.

Asiasanat: OCR, optinen merkintunnistus, kiinan kieli, japanin kieli, korean kieli

Sisällys

1 Johdanto

Optinen merkintunnistus (eng. Optical character recognition, OCR) on nykyään tärkeä teknologia yhteiskunnan käyttämissä digitaalisissa työkaluissa ja sovelluksissa. OCR:ää käytetään esimerkiksi fyysisten dokumenttien digitalisointiin ja arkistointiin sekä lomakkeiden automaattiseen käsittelyyn.[1] Nämä sovellukset vaativat, että paperilla oleva teksti siirtyy täysin virheettömänä digitaaliseen muotoon. Tämän takia on tärkeää, että OCR onnistuu tarkasti ilman virheitä.

Tekstin asettelu vaikuttaa OCR-työkalujen toimintaan. Jotkin työkalut voivat tunnistaa tekstiä vain yhteen suuntaan kerralla eli vain vaakasuunnassa olevaa tekstiä tai vain pystysuunnassa olevaa tekstiä. Tässä tutkielmassa perehdytään eri OCR-tekniikkoihin ja erityisesti niiden toimintaan kiinan, japanin ja korean (CJK) kielissä. Nämä kielet ovat erityisasemassa OCR:ssä, koska niiden merkistö on huomattavasti latinalaista merkistöä monimutkaisempi. Tästä johtuen CJK-kielten merkintunnistuksessa jo yhden viivan virhe yhdessä merkissä voi muuttaa merkin merkityksen täysin. Tämä voi johtaa siihen, että koko tekstin merkitys vääristyy, jolloin OCR-tuloksen hyödyntäminen muuttuu käytännössä mahdottomaksi.

Johdannon jälkeen tutkielmassa käsitellään OCR:n taustaa ja CJK kielten erityispiirteitä. Luvussa 3 syvennytään eri OCR tekniikoihin ja luvun alaluvuissa keskitytään tarkemmin jokaiseen kolmesta pääjaottelusta OCR-tekniikoille. Tämän jälkeen luvussa 4 suoritetaan empiirinen testi ja analysoidaan eri OCR-tekniikoita käyt-

täviä työkaluja. Lopuksi luvussa 5 koostetaan tutkielman päätelmät ja suoritetaan yhteenveto.

Tutkielmassa sovelletaan tutkimusmenetelmänä systemaattista kirjallisuuskatsausta, jota täydennetään pienimuotoisella empiirisellä kokeella luvussa 4. Tarkoituksena on tarkastella ja vertailla eri OCR-tekniikoihin perustuvien työkalujen suorituskykyä sekä muodostaa käytännönläheinen käsitys niiden soveltuvuudesta erityisesti CJK-kielten käsittelyyn. Tavoitteena on siten tuottaa perusteltuja näkökulmia OCR-työkalujen valintaan sekä vastata asetettuihin tutkimuskysymyksiin. Empiirisessä kokeessa käytetty toteutus on saatavilla GitLab-repositoriossa (ks. `https://gitlab.utu.fi/eesuoj/ocr_test/-/tree/37bf361b30c8b310b0392d584f83379a5b0c42ee/`).

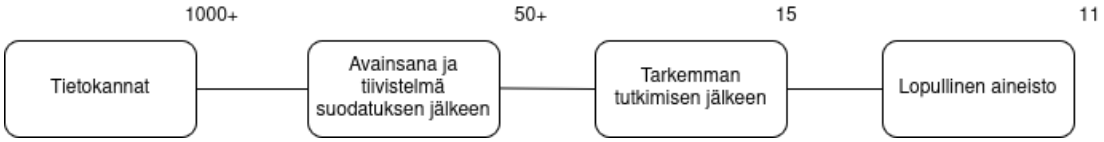
TK1: Mitkä ovat yleisimmät OCR virheet CJK kielissä ja miten ne vaikuttavat OCR tuloksen käyttöön?

TK2: Eri OCR-tekniikoiden tarkkuus erot?

TK3: Miten tekstin asettelu vaikuttaa OCR tulokseen?

Tutkimusaineisto koostuu pääasiassa vertaisarvioiduista tieteellisistä artikkeleista ja kirjojen osista. Kaikki käytetty aineisto on englanninkielistä ja se on haettu keskeisistä tieteellisistä tietokannoista, kuten Google Scholar, IEEE Xplore ja Web of Science. Hakuprosessissa hyödynnettiin keskeisiä käsitteitä ja niiden synonyymejä, kuten OCR, CJK, optical character recognition, benchmark, evaluation, accuracy, recognition rate ja multilingual OCR. Näistä muodostettiin erilaisia hakulausekkeita, kuten OCR CJK benchmark, Chinese OCR benchmark, Japanese OCR benchmark ja Korean OCR benchmark. Lisäksi tehtiin erillisiä hakuja OCR-teknologian kehi-

tyshistoriaan liittyen (esim. OCR development history), jotta tutkimukselle saatiin riittävä teoreettinen tausta.



Kuva 1.1: Aineistohaku kuvattuna

Hakujen tuloksena saatiin tuhansia osumia, joita rajattiin systemaattisesti otsikoiden ja avainsanojen perusteella. Alustavan suodatuksen jälkeen tarkempaan tarkasteluun valikoitui yli 50 lähdettä. Näistä merkittävä osa hylättiin, koska ne eivät sisältäneet vertailua eri OCR-työkalujen välillä tai esittäneet numeerisia tuloksia suorituskyvystä, mikä teki niistä tämän tutkimuksen kannalta vähemmän relevantteja. Lopulliseen analyysiin valittiin 11 lähdettä, jotka täyttivät asetetut kriteerit ja mahdollistivat luotettavan vertailun OCR-järjestelmien suorituskyvystä.

2 OCR yleisesti

OCR-teknologia kehitettiin 1950-luvulla, jolloin sitä käytettiin tunnistamaan standardisoitua tekstiä kirjoista tai dokumenteista otetuista kuvista. Varhaiset OCR-järjestelmät olivat tarkkuudeltaan rajallisia ja pystyivät tunnistamaan vain suppean joukon yleisimpiä merkkejä. Suorituskyky kuitenkin parani, kun kehitettiin hienostuneempia algoritmeja ja konenäkötekniikoita. Tämän lisäksi OCR-järjestelmien tunnistamien merkkien määrä kasvoi valtavasti. Seuraava merkkipaalu OCR-teknologian kehityksessä tapahtui 1980-luvun aikana, kun alettiin kehittämään koneoppivia OCR-tekniikoita. Koneoppivat OCR-tekniikat vakiintuivat osaksi OCR-tekniikoita 1990-luvulla. Ensimmäiset syväoppivat OCR-mallit kehitettiin 2010-luvulla. [1]

OCR-tekniikat voidaan jakaa kolmeen pääjaotteluun: perinteinen OCR, Koneoppiva OCR ja Syväoppiva OCR [2]. Perinteinen OCR kattaa kaikki OCR-menetelmät, jotka eivät käytä mitään koneoppimista [3]. Esimerkiksi sääntöpohjainen OCR kuuluu perinteisen OCR:n alle. Sääntöpohjaisessa OCR:ssä tunnistus perustuu kuvan samankaltaisuuteen ennalta määritettyjen piirteiden kanssa. Koska sääntöpohjainen OCR ei hyödynnä koneoppimista, se kykynee ainoastaan vertaamaan piirteiden samankaltaisuutta ennalta määritettyyn malliin. [3]

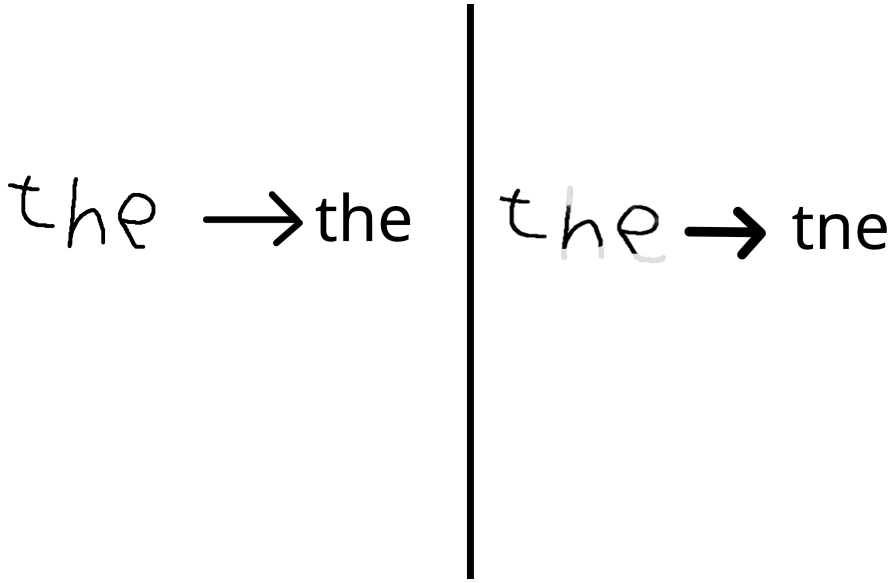
Koneoppiva OCR kattaa allensa kaikki koneoppimista käyttävät OCR-menetelmät, mutta ei mitään, mikä sisältää syväoppimista [1]. Esimerkki koneoppi-

vasta OCR:stä on tukivektorikone (SVM). Koneoppivan OCR:n piirteet, jotka malli oppii, ovat ennalta määritettyjä [2].

Syväoppivaan OCR:ään kuuluvat kaikki OCR-menetelmät, joissa käytetään neurovekkoja [4]. Esimerkiksi konvoluutioneuroverkko (CNN), joka toimii yhdessä sekvenssimallin kanssa, on syväoppiva OCR malli[4] [5]. Sekvenssimalli voi olla esimerkiksi transformer-arkkitehtuurin perustuva malli. Syväoppiva OCR käsittelee koko tekstin kuvana ilman, että sitä tarvitsee manuaalisesti segmentoida. CNN oppii itse piirteet kuvasta, ja sekvenssimalli tulkitsee merkkijonon järjestyksen.

Segmentointi tarkoittaa OCR-järjestelmissä vaihetta, joissa kuva jaetaan pienempiin osiin ennen varsinaista tunnistusta. Tyypillisesti tekstialue pilkotaan ensin riveihin ja tämän jälkeen sanoihin tai yksittäisiin merkkeihin, jotta jokainen osa voidaan tunnistaa erikseen. Perinteisessä ja koneoppivissa OCR-järjestelmissä segmentointi tehdään eksplisiittisesti ennen luokittelua, jolloin tunnistus perustuu yksittäisten merkkien analysointiin [6]. Segmentointi ei kuitenkaan ole aina yksiselitteistä, sillä merkkien rajat voivat olla epäselviä esimerkiksi silloin, kun merkit koskevat toisiaan tai kuvan laatu on heikko. Tällöin segmentointivirheet vaikuttavat suoraan tunnistuksen lopputulokseen. Tämän vuoksi segmentointi on keskeinen tekijä erityisesti perinteisten OCR-järjestelmien suorituskyvyssä [5], [7].

Hyvälaatuisissa englanninkielisissä teksteissä kaikki tekniikat toimivat erittäin hyvin, mutta jos tekstissä on vähänkin sumeita kohtia, varsinkin perinteisten tekniikoiden tarkkuus heikentyy huomattavasti [3].



Kuva 2.1: Esimerkki siitä, miten kuvassa oleva epätarkkuus voi vaikuttaa OCR tulokseen

Kuvassa 2.1 näkyy kuinka valaistuksen aiheuttama heijastuminen voi vaikuttaa OCR tulokseen. Kone- ja syväoppivat OCR menetelmät ovat huomattavasti perinteisiä OCR-menetelmiä sietokykyisempiä kohinalle [2]. Kuvausympäristön valaistus voi johtaa siihen, että kuvassa oleva teksti ei ole enää OCR luettavaa. Heijastumista voivat aiheuttaa esimerkiksi, jos kuvia otetaan salaman kanssa.

Latinalaista aakkosjärjestelmää käyttävissä kielissä on tyypillisesti 26 perusmerkkiä, ja suomen kielessä tähän joukkoon lisätään kolme kirjainta (å, ä, ö), jolloin suomalaisessa merkkijärjestelmässä on yhteensä 29 merkkiä. Japanin kielessä käytetään rinnakkain kolmea kirjoitusjärjestelmää: hiraganaa, katakanaa ja kanjia. Hiraganassa ja katakanassa on kummassakin 46 perusmerkkiä, kun taas kanji-merkkien määrä on huomattavasti suurempi; Unicode-standardi sisältää yli 90 000 kiinalaisperäistä merkkiä, joista osa jakaa saman koodipisteen japanin kanjien kanssa. Näin ollen kanji-järjestelmässä on yli 3 000-kertainen määrä merkkejä verrattuna suomalaiseen aakkostoon.

Korean kielen hangul-järjestelmä perustuu tavumerkkeihin, jotka muodostetaan yhdistämällä alkukonsonantti, vokaali ja mahdollinen loppukonsonantti. Alkukon-

sonantteja on 19, vokaaleja 21 ja loppukonsonantteja 28 (mukaan lukien nollaloppu), kuten taulukossa 2.1 esitetään. Näiden yhdistelmien perusteella mahdollisia hangul-tavumerkkejä on yhteensä $19 \times 21 \times 28 = 11\ 172$, ja kaikki nämä sisältyvät Unicode-standardiin. Näin ollen hangul-järjestelmässä on yli 300-kertainen määrä mahdollisia tavumerkkejä verrattuna suomalaiseen merkkijärjestelmään.

Taulukko 2.1: Hangulin tavumerkkien muodostukseen käytettävät Unicode-jamot. Näistä muodostuu yhteensä $19 \times 21 \times 28 = 11\ 172$ Hangul-tavumerkkiä.

Alkukonsonantti		Vokaali		Loppukonsonantti	
ᄀ	ᄁ	ᅡ	ᅢ		ᆨ
ᄂ	ᄃ	ᅣ	ᅤ	ᆩ	ᆪ
ᄄ	ᄅ	ᅥ	ᅦ	ᆫ	ᆬ
ᄆ	ᄇ	ᅧ	ᅨ	ᆭ	ᆮ
ᄈ	ᄉ	ᅩ	ᅪ	ᆯ	ᆰ
ᄊ	ᄋ	ᅫ	ᅬ	ᆱ	ᆲ
ᄌ	ᄍ	ᅭ	ᅮ	ᆳ	ᆴ
ᄎ	ᄏ	ᅯ	ᅰ	ᆵ	ᆶ
ᄐ	ᄑ	ᅱ	ᅲ	ᆷ	ᆸ
	ᄒ	ᅳ	ᅴ	ᆹ	ᆺ
			ᅵ	ᆻ	ᆼ
				ᆽ	ᆾ
				ᆿ	ᇀ
				ᇁ	ᇂ

Koska kiinan ja japanin kielissä käytetään osittain samoja merkkejä, mutta niiden merkitykset voivat erota kielestä riippuen, OCR-järjestelmien on välttämätöntä hyödyntää kielikontekstia tulkinnan tukena. Esimerkiksi merkkijono 勉強 tarkoittaa japanissa opiskelua, kun taas kiinassa se viittaa pakottamiseen. Merkit ovat visuaalisesti identtiset ja jakavat saman Unicode-koodipisteen, mutta semanttinen sisäl-

tö määräytyy kielen perusteella. Vastaavasti merkkijono 手紙 tarkoittaa japanissa kirjettä, kun taas kiinassa se viittaa vessapaperiin. Tällaiset tapaukset korostavat kontekstuaalisen analyysin merkitystä monikielisessä tekstintunnistuksessa.

Lisäksi OCR:n kannalta haastavia ovat visuaalisesti hyvin samankaltaiset merkit, joilla on kuitenkin eri merkitys. Esimerkiksi kiinalaiset merkit 未 (wèi) ja 末 (mò) eroavat toisistaan ainoastaan yhden vaakaviivan pituuden osalta, mutta niiden merkitykset ovat täysin erilaiset: 未 viittaa ajalliseen keskeneräisyyteen, kun taas 末 tarkoittaa loppua tai päätöstä. Tällaiset hienovaraiset visuaaliset erot asettavat korkeita vaatimuksia sekä kuvan tarkkuudelle että mallin kyvylle erottaa pieniä yksityiskohtia.

Edellä kuvatut piirteet, yhdessä merkkijärjestelmien laajuuden kanssa, muodostavat merkittäviä haasteita OCR-malleille. Jokainen yksittäinen merkki käsitellään tyypillisesti omana luokkanaan, mikä tarkoittaa tuhansia tai jopa kymmeniä tuhansia luokkia kielikohtaisessa mallissa. Tämä kasvattaa huomattavasti tarvittavan koulutusdatan määrää, sillä jokaisesta luokasta on oltava riittävästi esimerkkejä luotettavan tunnistuksen saavuttamiseksi. Lisäksi merkkien suuri yksityiskohtaisuus lisää virhealttiutta: jo yhden viivan puuttuminen tai vääristyminen voi muuttaa merkin toiseksi. Tästä syystä matala resoluutio tai kuvanlaadun heikkeneminen vaikeuttaa tunnistusta merkittävästi. Lopuksi on huomattava, että erityisesti kanjeilla ja kiinalaisilla merkeillä voi olla useita kontekstisidonnaisia merkityksiä, mikä lisää vaatimuksia OCR-prosessin jälkeiselle kielelliselle analyysille ja merkityserottelulle.

3 OCR-tekniikat

3.1 Perinteinen OCR

Perinteinen OCR perustuu ennalta määrättyihin sääntöihin ja piirteisiin [5]. Piirteet ja luokat määritellään ennalta, eikä OCR-mallille syötettävä data vaikuta niiden muodostumiseen. Perinteiseen OCR-menetelmään perustuva työkalu, joka on suunniteltu käsittelemään yhdellä tietyllä fontilla painettuja standardoituja dokumentteja, sisältää kaikki kyseisen fontin merkit, joita voi esiintyä. Tällainen työkalu vertailee jokaista merkkiä sisältämäänsä merkkilistaan ja sen perusteella ilmoittaa joko lähimmän vastineen tai että merkkiä ei voitu tunnistaa. Ennen kuin malli alkaa vertailemaan merkkejä se käyttää geometrista heuristiikkaa merkkien segmentointiin kuvasta. Segmentoinnin jälkeen malli alkaa vertailla kuvan merkkejä mallin merkkilistaan [7]. Kuvassa 3.1 on esimerkki yksinkertaistetusta Template matching algoritmista. Tämä algoritmi vertaa annettua kuvaa kuvatietokantaan, joka sisältää kullekin numerolle (0-9) kuvia. Algoritmi käyttää vertailuun Hamming etäisyyttä ja palauttaa käyttäjälle sen numeron, jonka kuvilla on pienin etäisyys annettuun kuvaan. Algoritmi toteuttaa ainoastaan OCR-menetelmän eikä sisällä segmentointia, vaan olettaa, että sille syötettävä kuva on jo segmentoitu yhden merkin kokoiseksi.

```
from PIL import Image
import numpy as np
from pathlib import Path

W, H = 64, 64

def prep(p):
    a = np.array(Image.open(p).convert("L").resize((W, H),
        Image.NEAREST))
    return (a < 128).astype(np.uint8)

x = prep("input.png")

best_label, best_dist = None, 10**18
for p in Path("dataset").glob("*.png"):
    label = Path(p).stem
    dist = int(np.count_nonzero(x ^ prep(p)))
    if dist < best_dist:
        best_dist, best_label = dist, label

print(best_label)
```

Kuva 3.1: Template matching -menetelmän toteutus Pythonilla.

3.2 Koneoppiva OCR

Koneoppiva OCR perustuu selkeään segmentoitiin ja käsin määriteltyihin piirteisiin, joiden avulla merkit luokitellaan opetusdatan perusteella. Tyypillinen arkkitehtuuri sisältää komponenttien erottelun (eng. connected component analysis), tekstirivien ja sanojen tunnistuksen, merkkisegmentoinin sekä piirteiden hahmottamisen ennen varsinaista luokittelua. Esimerkiksi Tesseractin varhaisessa monikielisessä versiossa merkit esitetään polygonisina ääriviivoina, joista muodostetaan piirrevektoreita, kuten sijaintia, suuntaa ja pituutta kuvaavia ominaisuuksia. Luokittelu toteutetaan prototyyppipohjaisella generatiivisella menetelmällä kaksivaiheisesti hyödyntäen tarkkaa etäisyyslaskentaa sekä luokkien karsintamenetelmää (engl. class pruner) [8]. Järjestelmä hyödyntää sanakirjaa ja kielimallia hakutilan rajaamiseen, ja segmentointi on keskeinen osa koko tunnistusprosessia. Segmentointiin voidaan käyt-

tää esimerkiksi projektioprofiileihin perustuvia menetelmiä, mutta ne kärsivät ali- ja ylisegmentoinnista erityisesti toisissaan kiinniolevien tai monimutkaisten merkkien tapauksessa [7]. Koneoppivassa OCR:ssä eri vaiheet (segmentointi, piirteiden irrotus, luokittelu ja kielimalli) ovat rakenteellisesti erillisiä, ja optimointi tapahtuu usein osakohtaisesti eikä koko järjestelmän tasolla.

Koneoppivaan luokitteluun ja eksplisiittiseen merkkisegmentointiin perustuvat OCR-järjestelmät muodostavat siirtymävaiheen perinteisen ja syväoppivan OCR:n välillä. Näissä järjestelmissä tunnistusprosessi koostuu useista erillisistä vaiheista, kuten esikäsittelystä, merkkien segmentoinnista, piirteiden muodostamisesta ja luokittelusta. Piirteet määritellään tyypillisesti ennalta ja kuvaavat esimerkiksi merkkien geometrisia ominaisuuksia, minkä jälkeen luokittelija vertaa näitä piirteitä opetusdatasta muodostettuihin malleihin. Tesseractin varhainen monikielinen toteutus on esimerkki tällaisesta lähestymistavasta, jossa merkkien tunnistus perustuu piirrevektoreihin ja prototyyppipohjaiseen luokitteluun useille kielille [8].

Kielikohtaista suorituskykyä tarkasteltaessa koneoppivat OCR-järjestelmät osoittavat, että sama järjestelmä pystyy käsittelemään useita kirjoitusjärjestelmiä, mutta tulokset vaihtelevat merkittävästi kielten välillä. Esimerkiksi monikielistä OCR:ää käsittelevässä tutkimuksessa Tesseract saavuttaa korkean tarkkuuden sekä englannille (99.5 %) että kiinalle (96.23 %) (ks. taulukko 3.1), mikä osoittaa, että koneoppivat menetelmät voivat yltää hyvään suorituskykyyn useissa kielissä, kun käytössä on riittävästi opetusdataa ja selkeä merkkijoukko [8]. Toisaalta uudemmassa monikielisessä vertailussa havaittiin, että koneoppiviin menetelmiin perustuvat järjestelmät, kuten kaksivaiheiset tai hybridimallit, saavuttavat eri kielissä vaihtelevia tuloksia (ks. taulukko 3.1). Esimerkiksi englannissa tarkkuus voi olla yli 90 %, kun taas kiinassa ja koreassa tarkkuus jää selvästi alemmaksi samassa järjestelmässä [7]. Tämä viittaa siihen, että koneoppivien OCR-järjestelmien suorituskyky riippuu vahvasti sekä kielestä että käytetystä malli arkkitehtuurista.

Taulukko 3.1: OCR-järjestelmien suorituskyky eri kielissä

Vuosi	Järjestelmä	Tyyppi	Kielituki	EN %	ZH %	KO %	Lähde
2025	Projection Profile OCR	Perinteinen	Yksikielinen	24.98	78.23	18.95	[7]
2009	Tesseract	Koneoppiva	Monikielinen (manuaalinen)	99.5	96.23	–	[8]
2025	Mixed Neural OCR	Koneoppiva	Monikielinen (yhteinen malli)	93.86	81.86	63.93	[7]
2025	Two-Step OCR	Koneoppiva	Monikielinen (yhteinen malli)	96.78	64.85	57.69	[7]
2025	Proposed Deep OCR	Syväoppiva	Monikielinen (yhteinen malli)	50.58	86.21	77.65	[7]

Näiden tutkimusten perusteella koneoppivan OCR:n keskeinen rajoite monikielisessä ympäristössä on se, että tunnistus perustuu usein eksplisiittiseen merkkisegmentointiin ennen varsinaista luokittelua. Mikäli segmentointi epäonnistuu, virhe siirtyy suoraan tunnistusvaiheeseen, mikä heikentää lopputulosta erityisesti niissä kielissä, joissa merkkien rakenne on monimutkainen tai vaihteleva. Tämän vuoksi koneoppivat OCR-järjestelmät soveltuvat parhaiten tilanteisiin, joissa dokumenttien rakenne ja käytetty kirjoitusjärjestelmä ovat suhteellisen vakiintuneita [7], [8].

3.3 Syväoppiva OCR

Syväoppiva OCR perustuu monikerroksisiin neuroverkkoihin, joissa piirteiden oppiminen ja luokittelu toteutetaan yhtenäisenä end-to-end-prosessina. Nykyiset järjestelmät hyödyntävät konvoluutioneuroverkkoja (CNN) tekstikuvan visuaalisten piirteiden oppimiseen sekä toistoverkkoja (RNN), kuten Long Short-Term Memory (LSTM)- ja Bidirectional Long Short-Term Memory (BiLSTM) -kerroksia, tekstin sekvenssiriippuvuuksien mallintamiseen [7]. LSTM-pohjaisissa ratkaisuissa tekstikuva käsitellään sekvenssinä, jossa konvoluutiokerrokset muodostavat piirre-esityksiä tekstikuvan kapeista osista ja LSTM-kerrokset tuottavat merkkiluokkia näiden piirteiden perusteella [6].

Yleinen ratkaisu on Long Short-Term Memory–Connectionist Temporal Classification (LSTM-CTC) -arkkitehtuuri, jossa Connectionist Temporal Classification (CTC) mahdollistaa merkkisekvenssien oppimisen ilman eksplisiittistä merkkirajojen annotointia [7]. Tutkimusten mukaan sama merkki voi esiintyä useissa peräkkäisissä kuvaviipaleissa, minkä vuoksi LSTM-kerrokset tuottavat usein saman symbolin useita kertoja. CTC-kerros yhdistää nämä toistuvat symbolit ja muodostaa lopullisen tekstisekvenssin ilman erillistä merkkisegmentointia [6]. Tällöin segmentointi on implisiittistä ja malli optimoi suoraan koko sanan tai tekstirivin tunnistustulosta.

Lisäksi encoder–decoder- ja attention-mekanismeihin perustuvissa malleissa encoder muodostaa koko syötekuvaa kuvaavan piirrevektorin, jonka perusteella decoder tuottaa merkkisekvenssin kohdistamalla huomion tekstikuvan eri osiin vaiheittain [6]. Monikielisessä kontekstissa syväoppiva lähestymistapa mahdollistaa suurten merkkijoukkojen, kuten kiinan ja korean, tehokkaamman käsittelyn kuin perinteiset eksplisiittiseen segmentointiin perustuvat menetelmät [7]. Samalla tutkimuksissa todetaan, että suurten merkistöjen käsittely on edelleen haastavaa, koska yksittäiset kuvaviipaleet sisältävät vain rajallisesti informaatiota koko merkistä [6].

Vaikka Tesseractin alkuperäinen luokitin perustui perinteiseen yhdistettyjen komponenttien analyysiin, sen monikielinen sovittaminen osoitti, että suurten merkkijoukkojen käsittely voidaan integroida OCR-järjestelmiin ilman voimakasta kielikohtaista rakenteellista segmentointia [8]. Tutkimuksessa korostetaan erityisesti, että kiinan, korean ja arabian kaltaiset kirjoitusjärjestelmät asettavat OCR-järjestelmille haasteita merkkien suuren määrän, merkkien yhdistymisen ja epäselvien merkkirajojen vuoksi [8].

Monikielisessä tekstintunnistuksessa syväoppivat OCR-mallit ovat osoittaneet kykenevänsä käsittelemään useita kirjoitusjärjestelmiä samassa järjestelmässä. Monikielisiä haasteita käsittelevät tutkimukset hyödyntävät laajoja, useita kieliä ja kirjoitusjärjestelmiä sisältäviä aineistoja, ja niiden tulokset osoittavat, että syväoppi-

mismenetelmät kykenevät yleistämään eri kieliin ilman kielikohtaisia malleja [9], [10]. Tämä on merkittävä ero koneoppiviin menetelmiin verrattuna, joissa kielikohtainen optimointi on usein välttämätöntä.

Kielikohtaisia tuloksia tarkasteltaessa syväoppivat OCR-järjestelmät saavuttavat tasaisempaa suorituskykyä eri kielissä kuin segmentointiin perustuvat menetelmät. Erot kielten välillä eivät kuitenkaan katoa kokonaan. Esimerkiksi monikielisessä OCR-järjestelmässä saavutetaan kiinan (86.21 %) ja korean (77.65 %) osalta korkeampi tarkkuus kuin useissa koneoppivissa ratkaisuissa (ks. taulukko 3.1), mutta samalla englannin tarkkuus voi jäädä selvästi alemmaksi samassa mallissa [7]. Tämä osoittaa, että vaikka syväoppivat mallit yleistyvät paremmin eri kirjoitusjärjestelmiin, niiden suorituskyky riippuu edelleen opetusdatasta ja mallin optimoinnista.

Lisäksi joissakin tutkimuksissa syväoppivia OCR-järjestelmiä on laajennettu käyttämällä vahvistusoppimista merkkisegmentoinnin optimointiin monikielisessä ympäristössä. Tällainen lähestymistapa mahdollistaa sen, että järjestelmä oppii automaattisesti optimaaliset segmentointistrategiat eri kielille, mikä parantaa suorituskykyä erityisesti monikielisissä sovelluksissa [6]. Näiden ominaisuuksien vuoksi syväoppivat OCR-järjestelmät ovat nykyisin keskeinen lähestymistapa monikielisessä tekstintunnistuksessa sekä tutkimuksessa että käytännön sovelluksissa.

4 OCR-työkalujen vertailu

Tässä luvussa tarkastellaan eri OCR-tekniikoiden sietokykyä kuvassa esiintyville häiriöille testin avulla. Tarkasteltavia häiriöitä ovat esimerkiksi kuvan kohina sekä kuvan kallistuminen. Näitä häiriötekijöitä simuloitiin testissä muokkaamalla alkuperäisiä kuvia hallitusti.

Testissä vertailtiin PaddleOCR- ja Tesseract-työkaluja. Tesseract-työkalulla kuvat käsiteltiin kahdesti käyttäen kahta eri tunnistussuunta-asetusta. Häiriöitä lisättiin kuviin kääntämällä niitä vaiheittain -5 asteen kulmasta $+5$ asteen kulmaan yhden asteen välein sekä lisäämällä valkoista kohinaa kahdella eri voimakkuustasolla. Testissä käytettiin PaddleOCR:stä versiota 2.6.2 ja Tesseractistä versiota 4.1.0.

PaddleOCR on avoimen lähdekoodin optinen tekstintunnistusjärjestelmä, joka perustuu PaddlePaddle-syväoppimisalustaan ja hyödyntää neuroverkkoja tekstin havaitsemiseen, tunnistamiseen sekä rakenteen analysointiin kuvista. Se tukee useita kieliä sekä erilaisia tekstityyppejä, kuten painettua ja käsinkirjoitettua tekstiä. Järjestelmän vahvuuksiin kuuluu korkea tarkkuus ja kyky muuntaa kuvat helposti jatkokäsiteltävään digitaaliseen muotoon [5].

Testissä työkalujen suorituskykyä arvioitiin CER-metriikalla (Character Error Rate), joka perustuu Levenshtein-etäisyyteen. Levenshtein-etäisyys kuvaa kahden merkkijonon välistä minimimäärää operaatioita (lisäys tai poisto), joilla toinen merkkijono voidaan muuntaa toiseksi [11]. Ennen etäisyyden laskemista merkkijonot normalisoitiin poistamalla rivinvaihdot ja välilyönnit sekä yhdistämällä teksti

yhdeksi merkkijonoksi. Näin varmistettiin, ettei mittari vääristy OCR-työkalujen erilaisista tekstin muotoiluista. CER-arvo laskettiin jakamalla Levenshtein-etäisyys oikean merkkijonon pituudella.

Seuraavaksi esitetään testin tulokset eri kulmilla ja kohinatasoilla. Tulokset on esitetty CER-arvoina, joissa pienempi arvo tarkoittaa parempaa tunnistustarkkuutta. Tarkastelun tavoitteena on selvittää, miten kuvan kallistuminen ja kohina vaikuttavat eri OCR-työkalujen suorituskykyyn.

Tulokset eri kulmilla ilman kohinaa on esitetty taulukossa 4.1, kun taas kohinatasoilla 10 ja 20 saadut tulokset on esitetty taulukoissa 4.2 ja 4.3. Lisäksi yhteenveto tulosten mediaaneista ja keskihajonnoista on esitetty taulukossa 4.4.

Tuloksista havaitaan, että PaddleOCR saavuttaa kaikissa testitilanteissa vakaan CER-arvon 0.050 riippumatta kulmasta tai kohinatasosta (taulukot 4.1–4.3). Tämä viittaa siihen, että kyseinen työkalu on erittäin sietokykyinen testissä käytettyjä häiriöitä vastaan.

Tesseractin suorituskyky vaihtelee merkittävästi käytetyn tunnistussuunnan mukaan. Vaakasuunnassa CER-arvot sijoittuvat pääosin välille 0.700–1.350, ja tuloksissa on havaittavissa lievää heikkenemistä erityisesti suuremmilla kulmilla, kun taas kohinan vaikutus jää suhteellisen vähäiseksi kulman vaikutukseen verrattuna (taulukot 4.1–4.3). Pystysuuntaisella asetuksella suorituskyky on selvästi heikompi, sillä CER-arvot sijoittuvat pääosin välille 1.450–2.100 ja pysyvät melko vakaina eri kulmilla ja kohinatasoilla. Tämä viittaa siihen, että pystysuuntainen asetus tuottaa heikkoja tuloksia häiriötekijöistä riippumatta (taulukot 4.1–4.3).

Taulukossa 4.4 esitetyt mediaani- ja keskihajonta-arvot tukevat edellä kuvattuja havaintoja. PaddleOCR:n keskihajonta on nolla kaikissa tapauksissa, mikä korostaa tulosten vakautta, kun taas Tesseractin tuloksissa esiintyy jonkin verran vaihtelua erityisesti vaakasuuntaisessa asetuksessa.

On myös huomionarvoista, että PaddleOCR:n CER-arvo pysyy täysin vakiona kaikissa testitilanteissa (taulukot 4.1–4.3). Tämä voi viitata joko työkalun erittäin hyvään sietokykyyn tai siihen, että käytetty testiaineisto tai mittausmenetelmä ei ole riittävän herkkä havaitsemaan pieniä eroja suorituskyvyssä.

Taulukko 4.1: CER eri kulmilla (kohina = 0)

Työkalu	**-5°**	**-4°**	**-3°**	**-2°**	**-1°**	**0°**	**1°**	**2°**	**3°**	**4°**	**5°**
PaddleOCR	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Tesseract Vaaka	0.95	0.85	0.70	0.80	0.75	0.80	0.80	0.85	1.25	0.85	1.30
Tesseract Pysty	1.45	1.90	1.45	1.85	1.80	2.00	2.00	2.00	2.05	2.05	2.10

Taulukko 4.2: CER eri kulmilla (kohina = 10)

Työkalu	**-5°**	**-4°**	**-3°**	**-2°**	**-1°**	**0°**	**1°**	**2°**	**3°**	**4°**	**5°**
PaddleOCR	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Tesseract Vaaka	0.95	0.85	0.75	0.80	0.80	0.80	0.85	0.85	1.30	0.85	1.35
Tesseract Pysty	1.45	1.90	1.45	1.90	2.00	1.90	2.00	2.00	2.10	1.95	2.10

Taulukko 4.3: CER eri kulmilla (kohina = 20)

Työkalu	**-5°**	**-4°**	**-3°**	**-2°**	**-1°**	**0°**	**1°**	**2°**	**3°**	**4°**	**5°**
PaddleOCR	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
Tesseract Vaaka	0.85	0.70	0.75	0.80	0.80	0.80	0.85	1.25	1.30	0.85	0.85
Tesseract Pysty	1.90	1.90	1.85	1.85	2.00	2.00	2.10	2.10	2.05	1.95	2.00

Taulukko 4.4: CER-mediaani ja keskihajonta kohinatasoittain

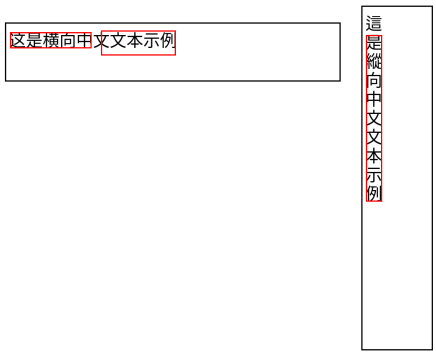
Työkalu	**Kohina = 0**	**Kohina = 10**	**Kohina = 20**
PaddleOCR	0.05 ± 0.00	0.05 ± 0.00	0.05 ± 0.00
Tesseract Vaaka	0.85 ± 0.19	0.85 ± 0.20	0.85 ± 0.19
Tesseract Pysty	2.00 ± 0.22	1.95 ± 0.22	2.00 ± 0.09

Tulosten havainnollistamiseksi tarkastellaan OCR-työkalujen tuottamia tekstialueiden tunnistuksia kohinattomissa kuvissa. Erityinen huomio kiinnitetään sekä suoraan orientaatioon (0°) että pieniin kiertokulmiin (±5°), jotta voidaan arvioida orientaation vaikutusta tekstialueiden havaitsemiseen. Punaiset rajaukset kuvaavat tunnistettuja tekstialueita suorassa tapauksessa, kun taas kiertokulmia havainnollistavat kuvat esitetään ilman tunnistuslaatikoita (kuva 4.1).

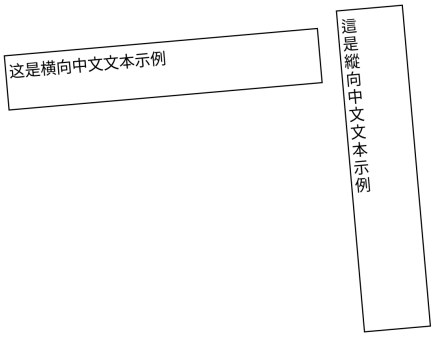
Tesseractin tulokset vaihtelevat merkittävästi käytetystä asetuksesta riippuen: vaakasuuntainen asetus pilkkoo tekstin useisiin virheellisiin segmentteihin (kuva 4.1a), kun taas pystysuuntainen asetus tuottaa visuaalisesti yhtenäisemmät rajaukset (kuva 4.1e). Näistä huolimatta jo pienet kiertokulmat (±5°) heikentävät segmentoinnin laatua selvästi, mikä viittaa menetelmän herkkyyteen tekstin orientaatiolle (kuva 4.1). Tämä näkyy myös heikkona merkkitunnistuksena ja korkeina CER-arvoina.

PaddleOCR puolestaan tunnistaa tekstialueet sekä vaaka- että pystysuunnassa yhtenäisinä ja rakenteellisesti johdonmukaisina kokonaisuuksina (kuva 4.1c). Menetelmä säilyttää suorituskykynsä paremmin myös pienillä kiertokulmilla, mikä ilmenee vakaampana segmentointina eri orientaatioissa (kuva 4.1). Tämä selittää sen matalan ja tasaisen CER-tason verrattuna Tesseractiin.

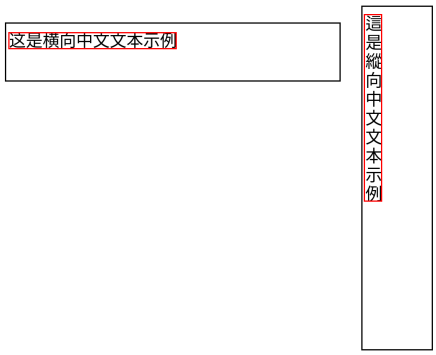
Kokonaisuudessaan työkalujen väliset erot syntyvät pitkälti jo tekstialueiden havaitsemisvaiheessa ennen varsinaista merkkien tunnistusta. Tulokset osoittavat, että erityisesti orientaation muutokset korostavat robustin esikäsittelyn ja tekstin suuntauksen tunnistamisen merkitystä OCR-prosessissa.



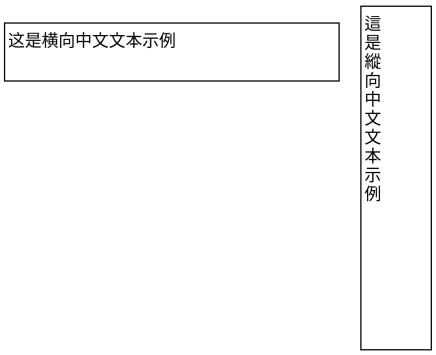
(a) Tesseract (vaakasuunta)



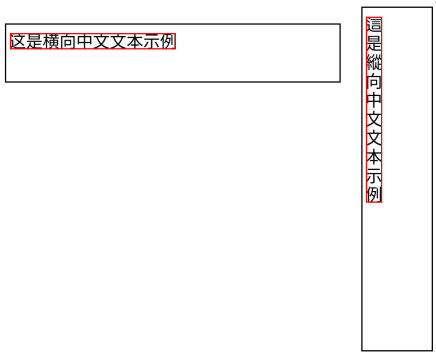
(b) Kierto +5°, ei kohinaa



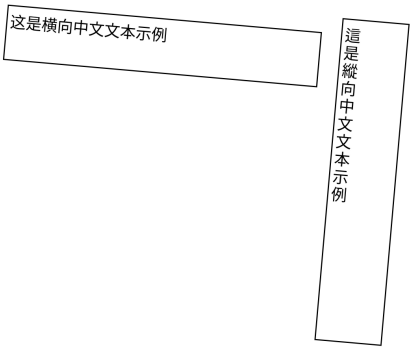
(c) PaddleOCR (0°, ei kohinaa)



(d) Alkuperäinen kuva (0°, ei kohinaa)



(e) Tesseract (pystysuunta)



(f) Kierto −5°, ei kohinaa

Kuva 4.1: OCR-työkalujen tekstialueiden tunnistus kohinattomissa kuvissa eri orientaatioissa: suora tapaus (0°) tunnistuslaatikoineen sekä kiertokulmat (+5° ja −5°) ilman laatikoita

5 Yhteenveto

Tutkielmassa havaittiin, että OCR-tekniikoiden suorituskyky riippuu merkittävästi sekä käytetystä menetelmästä että tunnistettavasta kielestä. Erityisesti kiinan, japanin ja korean kaltaiset monimutkaiset merkkijärjestelmät heikentävät tunnistustarkkuutta, koska merkkien suuri määrä ja visuaalinen samankaltaisuus lisäävät virheiden riskiä. Perinteiset ja koneoppivat OCR-menetelmät kärsivät erityisesti segmentointiin liittyvistä ongelmista, kun taas syväoppivat menetelmät pystyvät käsittelemään tekstiä kokonaisuutena ja saavuttavat tasaisemman suorituskyvyn eri kielissä. Empiirisen testin perusteella syväoppiva PaddleOCR osoittautui selvästi sietokykyisemmäksi häiriöitä vastaan, kun taas Tesseractin suorituskyky vaihteli tekstin asettelun ja tunnistussuunnan mukaan.

TK1: Mitkä ovat yleisimmät OCR virheet CJK kielissä ja miten ne vaikuttavat OCR tuloksen käyttöön?

Yleisimmät OCR-virheet CJK-kielissä liittyvät merkkien visuaaliseen samankaltaisuuteen, suuriin merkkimääriin sekä kuvien laatuun. Pienetkin erot merkeissä, kuten yksittäisen viivan puuttuminen tai pituus, voivat muuttaa merkin merkityksen täysin. Lisäksi kohina, epätarkkuus ja valaistusvirheet heikentävät tunnistusta merkittävästi. Näiden virheiden seurauksena OCR-tuloksesta voi tulla käytännössä käyttökelvoton, koska yksittäiset virheet voivat muuttaa koko tekstin merkityksen.

TK2: Eri OCR-tekniikoiden tarkkuus erot?

OCR-tekniikoiden välillä on merkittäviä eroja tarkkuudessa. Perinteiset menetelmät toimivat hyvin vain selkeissä ja standardoiduissa tilanteissa, mutta niiden suorituskyky heikkenee nopeasti häiriöiden lisääntyessä. Koneoppivat menetelmät parantavat tarkkuutta, mutta ovat edelleen riippuvaisia erillisistä käsittelyvaiheista, kuten segmentoinnista. Syväoppivat OCR-menetelmät saavuttavat parhaan ja tasaisimman tarkkuuden, koska ne oppivat piirteet suoraan datasta ja käsittelevät tekstin kokonaisuutena ilman eksplisiittistä segmentointia.

TK3: Miten tekstin asettelu vaikuttaa OCR tulokseen?

Tekstin asettelu vaikuttaa merkittävästi OCR-tulokseen, erityisesti menetelmissä, jotka olettavat tietyn lukusuunnan. Testin perusteella tunnistussuunnan valinta voi heikentää tuloksia huomattavasti, vaikka tekstialueet tunnistettaisiin oikein. Lisäksi kuvan kallistuminen ja tekstin suunta vaikuttavat tunnistuksen tarkkuuteen enemmän kuin kohina. Syväoppivat menetelmät ovat vähemmän herkkiä asettelulle, kun taas perinteisemmät menetelmät voivat epäonnistua, jos tekstin rakenne poikkeaa odotetusta.

Jatkotutkimuksessa voitaisiin tarkastella modernien OCR-järjestelmien suorituskykyä manga- ja webtoon-materiaalissa kiinan, japanin ja korean kielissä. Erityisesti tutkimuksessa voitaisiin analysoida puhekuplien, pystysuuntaisen tekstin, erikoisfonttien sekä kuvituksen kanssa päällekkäisen tekstin vaikutusta tekstintunnistuksen tarkkuuteen. Manga- ja webtoon-aineistot muodostavat OCR-järjestelmille huomattavasti perinteisiä dokumentteja ja kirjoja monimutkaisemman tutkimusympäristön, sillä niiden visuaalinen rakenne, tekstin sijoittelu ja typografiset ratkaisut vaihtelevat merkittävästi ilman laajaa standardisointia.

Lähdeluettelo

[1] J. Wang, "A study of the OCR development history and directions of development", *Highlights in Science, Engineering and Technology*, vol. 72, s. 409–415, 15. joulukuuta 2023. DOI: 10.54097/bm665j77.

[2] G. S. Hukkeri, R. H. Goudar, P. Janagond ja P. S. Patil, "Machine learning in OCR technology: Performance analysis of different OCR methods for slide-to-text conversion in lecture videos", *International Journal of Advanced Computer Science and Applications*, vol. 13, nro 8, 2022. DOI: 10.14569/IJACSA.2022.0130839.

[3] P. Jain, K. Taneja ja H. Taneja, "Which OCR toolset is good and why : A comparative study", *Kuwait Journal of Science*, vol. 48, nro 2, 5. huhtikuuta 2021. DOI: 10.48129/kjs.v48i2.9589.

[4] M. Schlüter, C. Tepper, C. Briese, O. Kroeger, R. Vicente-Garcia ja J. Krüger, "Deep learning-based optical character recognition for identifying on-label printed part numbers of used automotive parts: A comparative study of open source and commercial methods", teoksessa *Sustainable Manufacturing as a Driver for Growth*, H. Kohl, G. Seliger, F. Dietrich ja S. Mur, toim., Cham: Springer Nature Switzerland, 2025, s. 527–534, ISBN: 978-3-031-77429-4. DOI: 10.1007/978-3-031-77429-4_58.

[5] S. A. Francis ja M. Sangeetha, "A comparison study on optical character recognition models in mathematical equations and in any language", *Results*

in Control and Optimization, vol. 18, s. 100 532, 1. maaliskuuta 2025. DOI: 10.1016/j.rico.2025.100532.

[6] J. Park, E. Lee, Y. Kim, I. Kang, H. I. Koo ja N. I. Cho, "Multi-Lingual Optical Character Recognition System Using the Reinforcement Learning of Character Segmenter", *IEEE Access*, vol. 8, s. 174 437–174 448, 2020. DOI: 10.1109/ACCESS.2020.3025769.

[7] F. S. Binunya ja H. Zhou, "Multilingual text recognition and assistance for low-resource languages using computer vision", *Open Access Library Journal*, vol. 12, nro 6, 30. kesäkuuta 2025. DOI: 10.4236/oalib.1113574.

[8] R. Smith, D. Antonova ja D.-S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR", teoksessa *Proceedings of the International Workshop on Multilingual OCR*, sarja MOCR '09, New York, NY, USA: Association for Computing Machinery, 25. heinäkuuta 2009, s. 1–8, ISBN: 978-1-60558-698-4. DOI: 10.1145/1577802.1577804.

[9] N. Nayef et al., "ICDAR2017 Robust Reading Challenge on Multi-Lingual Scene Text Detection and Script Identification - RRC-MLT", teoksessa *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, marraskuu 2017, s. 1454–1459. DOI: 10.1109/ICDAR.2017.237.

[10] N. Nayef et al., "ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition —RRC-MLT-2019", teoksessa *2019 International Conference on Document Analysis and Recognition (ICDAR)*, syyskuu 2019, s. 1582–1587. DOI: 10.1109/ICDAR.2019.00254.

[11] A. A. Kulkarni ja N. Kiyavash, "Nonasymptotic Upper Bounds for Deletion Correcting Codes", *IEEE Transactions on Information Theory*, vol. 59, nro 8, s. 5115–5130, elokuu 2013. DOI: 10.1109/TIT.2013.2257917.