

Tekoälyä hyödyntävän sovel- luksen määrittely valaisimien teknisten tietojen analysointiin

Patrik Jokiharju

OPINNÄYTETYÖ
Marraskuu 2022

Dataosaaminen ja tekoäly (ylempi AMK)

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Dataosaaminen ja tekoäly (ylempi AMK)

JOKIHARJU, PATRIK:

Tekoälyä hyödyntävän sovelluksen määrittely valaisimien teknisten tietojen analysointiin

Opinnäytetyö 65 sivua, joista liitteitä 8 sivua
Marraskuu 2022

Valaisimien ja muiden sähkölaitteiden turvallisuustestauksessa käsitellään paljon laitteiden teknistä dokumentaatiota. Standardit asettavat vaatimuksia laitteiden arvokilpien ja muun teknisen dokumentaation sisällölle, ja teknisten tietojen perusteella tehdään testaukseen liittyviä päätöksiä. Teknisen dokumentaation analysointi on perinteisesti tehty manuaalisesti.

Tämän opinnäytetyön tarkoituksena oli tutkia, kuinka tekoälyä voidaan hyödyntää sähkölaitteiden teknisen dokumentaation analysoinnissa. Tavoitteena oli määrittää käytettävät tekoälytekniikat sekä pohtia, kuinka tekoälymallien avulla saatuja tietoja voidaan käyttää testausprosessissa. Työn yhteydessä oli tarkoitus luoda tekoälymallit, joiden avulla arvokilvistä ja teknisestä dokumentaatiosta saataisiin tunnistettua ja eristettyä tarvittavia tietoja.

Opinnäytetyössä päädyttiin käyttämään Azuren Cognitive Services -palvelun pilvipohjaisia tekoälymalleja. Palvelusta valittiin OCR-algoritmi kirjoitetun tekstin tunnistamiseen, mukautettu entiteettitunnistusmalli tekstin luokitteluun sekä mukautettu objektitunnistusmalli arvokilvissä esiintyvien kuvatunnusten tunnistukseen ja luokitteluun. OCR-algoritmin osalta päädyttiin tutkimaan, kuinka erilaiset kuvaolosuhteet vaikuttavat tekstin tunnistukseen. Mukautettujen mallien osalta tutkimuksen yhteydessä kerättiin opetusdatat, valmisteltiin opetukset, opetettiin mallit ja arvioitiin mallien hyvyttä. Tutkimuksessa esitellään tutkimustulokset OCR-algoritmin osalta sekä mukautettujen mallien luomisprosessit ja metriikat. Tutkimuksessa esitetään myös pohdintaa siitä, kuinka mallien avulla saatuja tietoja voitaisiin käyttää testauksen apuna ja miten malleista voitaisiin rakentaa kokonaisvaltainen sovellus.

Tutkimuksen tuloksena todettiin, että tekoälyä voidaan hyödyntää arvokilpien ja teknisen dokumentaation analysoinnissa. Tutkimuksen aikana havaittujen haasteiden myötä tultiin kuitenkin siihen tulokseen, että vaikka tekoälyä voidaan hyödyntää, niin kaikissa tapauksissa se ei välttämättä ole kannattavaa. Tekoälyn tuoma hyöty riippuu paljon siitä, kuinka monimutkaisen laitteen teknisiä tietoja tekoälyä hyödyntävän sovelluksen avulla on tarkoitus analysoida.

Asiasanat: tekoäly, konenäkö, entiteettien tunnistus, Azure

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Master's Degree in Data Expertise and Artificial Intelligence

JOKIHARJU, PATRIK:
Defining an AI Application to Analyse Technical data of Luminaires

Master's thesis 65 pages, appendices 8 pages
November 2022

In the safety testing of luminaires and other electrical equipment, a considerable amount of technical documentation is processed. The standards set requirements for technical documentation, and some of the test values are based on the technical data of the equipment. Data processing has traditionally been done manually.

The purpose of this study was to investigate how artificial intelligence can be used to analyse technical documentation. The goal was to define suitable AI technologies and consider how the information provided by AI models could be used in the testing process.

Cloud-based AI models of Azure Cognitive Services was used in this project. An OCR algorithm was chosen for text recognition, custom named entity recognition model for entity recognition, and custom vision model for symbol recognition. In this thesis, the experimentations with the OCR algorithm and the building and evaluation processes of the custom models are presented.

As a result, it was concluded that AI could be used to analyse the technical documentation of electrical equipment. However, the use of AI powered applications is not necessarily a cost-effective solution especially with low-tech equipment due to large building and maintenance investments.

Key words: artificial intelligence, computer vision, entity recognition, Azure

SISÄLLYS

1	JOHDANTO	6
2	SÄHKÖTURVALLISUUSTESTAUS JA SERTIFIOINTI	7
	2.1 Laitestandardit ja sertifiointijärjestelmät	7
	2.2 Testaus- ja sertifiointiprosessi	8
	2.3 Toiminnan nykytilanne	9
3	AZUREN COGNITIVE SERVICEN PALVELUT JA TEKNIIKAT	11
	3.1 Yleiskatsaus Cognitive Services -palveluihin	11
	3.2 Konenäkö ja konvoluutioneuroverkot	12
	3.3 Nimettyjen entiteettien tunnistus	17
	3.4 Metriikat mukautettujen mallien arviointiin.....	19
4	SOVELLUS	23
	4.1 Sovelluksen tavoite ja perusperiaate.....	23
	4.2 Tuotteiden arvokilpien analysointi	24
	4.3 Kirjoitetun tekstin tunnistus arvokilvistä.....	25
	4.4 Kirjoitetun tekstin luokittelu.....	28
	4.4.1 Opetusdata	29
	4.4.2 Opetuksen valmistelu ja mallin opetus	30
	4.4.3 Opetetun mallin arviointi	34
	4.5 Kuvatunnusten tunnistaminen	39
	4.5.1 Opetusdata	40
	4.5.2 Opetuksen valmistelu ja mallin opetus	41
	4.5.3 Opetetun mallin arviointi	42
	4.6 Kirjallisesta dokumentaatiosta tehtävät tunnistukset	45
	4.7 Tunnistuksista tehtävät päätökset	47
	4.8 Sovelluksen tulevaisuuden kehitysmahdollisuudet.....	51
5	POHDINTA	53
	LÄHTEET.....	56
	LIITTEET	58
	Liite 1. OCR-algoritmin kokeilun tuloksia	58
	Liite 2. Mukautetun entiteettientunnistusmallin metriikat	60
	Liite 3. Entiteettientunnistusmallin kokeilut.....	61
	Liite 4. Mukautetun objektintunnistusmallin metriikat	63
	Liite 5. Mukautetun objektintunnistusmallin testidataa	64

LYHENTEET JA TERMIT

CB	Certification Body
EN	European Standards / European Norm
ENEC	European Norms Electrical Certification
ETICS	European Testing, Inspection and Certification System
IEC	International Electrotechnical Commission
IECEE	IEC System of Conformity Assessment Schemes for Electrotechnical Equipment and Components

1 JOHDANTO

Latteiden sähköturvallisuustestauksessa käsitellään suuria määriä kirjallista dokumentaatiota, kuten arvokilpiä, laitteisiin liittyvää teknistä dokumentaatiota sekä asennusohjeita. Standardit asettavat tiettyjä vaatimuksia arvokilpien ja teknisen dokumentaation sisällölle ja testauksessa käytettävät sähköiset arvot sekä laitteille sovellettavat standardin antamat raja-arvot määräytyvät osittain laitteissa esitettyjen teknisten tietojen mukaan. Teknisten tietojen tutkiminen on perinteisesti tehty manuaalisesti, eli testaaja tutkii kaiken materiaalin läpi, etsien standardin vaatimia tietoja. Tämä prosessi voi olla erittäin aikaa vievää, koska monimutkaisten laitteiden dokumentaatio voi olla usean sadan sivun mittainen.

Tämän opinnäytetyön tarkoituksena oli tutkia, kuinka tekoälyä voidaan hyödyntää valaisimien arvokilpien ja muun teknisen dokumentaation analysoinnissa ja miten saatuja tietoja voidaan käyttää hyväksi valaisimien turvallisuustestauksessa. Tavoitteena oli luoda tekoälymallit, joiden avulla voidaan tunnistaa ja luokitella kirjoitettua tekstiä sekä kuvatunnuksia. Opinnäytetyön tarkoituksena ei ollut luoda täysin toimivaa arvokilpiä ja dokumentaatiota analysoivaa sovellusta, vaan määrittää käytettävät tekniikat ja pohtia tekoälymallien avulla tunnistettujen tietojen pohjalta tehtäviä testausratkaisuja. Määrityksen oli tarkoitus luoda pohja sovellukselle, joka nopeuttaa testausprosessia sekä karsii manuaalisesta työstä aiheutuvia virheitä.

Tutkimuksen alkuvaiheessa päätettiin, että tekoälymallit pyritäisiin luomaan valmiiden pilvipohjaisten palveluiden avulla. Tähän valikoitui Microsoftin Azure-alustalla tarjoama Cognitive Services -palvelu. Palvelusta valikoitui lopulta käytettäväksi valmis malli optiseen merkkien tunnistukseen, mukautettu malli nimettyjen entiteettien tunnistukseen sekä mukautettu malli objektien tunnistukseen.

Opinnäytetyössä esitellään ensin lyhyesti sähkölaitteiden testaus- ja sertifiointiprosessit sekä Cognitive Services -palvelu ja teoriat algoritmien takana. Tämän jälkeen esitellään tekoälymallien toimintaa, mukautettujen mallien luomisprosessit sekä arvioidaan luotujen mallien tehokkuutta. Lopuksi pohditaan, kuinka mallien avulla saatuja tietoja voidaan hyödyntää testausprosessissa.

2 SÄHKÖTURVALLISUUSTESTAUS JA SERTIFIOINTI

2.1 Laitestandardit ja sertifiointijärjestelmät

Sähkölaitteiden turvallisuustestaus perustuu eri laitestandardeihin. Sähköalan standardeja luo ja ylläpitää IEC. IEC-standardit ovat kansainvälisiä laitestandardeja, jotka määräävät kansainvälisellä tasolla tuotteille asetettavat vaatimukset. Kansainvälisten standardien lisäksi on luotu kansallisia standardeja. Kansalliset standardit antavat lisävaatimuksia kansainvälisiin standardeihin sekä joissain tilanteissa voivat jopa muuttaa kansainvälisten standardien vaatimuksia. Eurooppalainen EN-standardeja luova ja ylläpitävä organisaatio on ETICS. EN-standardit ovat eurooppalaisia standardeja, joissa on huomioitu Euroopan erityisvaatimukset. Eurooppalaisista standardeista 85 % perustuu kansainvälisiin IEC-standardeihin ja loppuilla viidellätoista prosentilla standardeista ei ole kansainvälistä vastinetta, vaan standardit ovat luotu ainoastaan Euroopan markkinoille (SESKO RY n.d.).

Useat sähkölaitestandardisarjat koostuvat pääosasta sekä lisäosista. Esimerkiksi tutkimustyössä käytettävässä valaisinstandardisarjassa on opinnäytetyön kirjoitushetkellä pääosan lisäksi 23 lisäosaa. Standardisarjojen pääosa esittää yleiset vaatimukset ja testit. Lisäosat taas määrittävät testijärjestyksen sekä antavat laitekohtaisia lisävaatimuksia. Valaisinstandardisarjan lisäosia on tehty esimerkiksi upotettaville valaisimille, siirrettäville valaisimille sekä katu- ja tievalaisimille. Yksittäiset standardit taas koostuvat eri osioista. Valaisinstandardissa osioita ovat esimerkiksi merkinnät ja ohjeet, rakenne, johdotus sekä pinta- ja ilmajälit.

Kansainvälisten ja kansallisten standardien tapaan sertifiointijärjestelmät jakautuvat myös kansallisiin ja kansainvälisiin sertifiointijärjestelmiin. Kansainvälinen maailmanlaajuinen sertifiointijärjestelmä on IECEE:n hallinnoima CB-järjestelmä, joka perustuu kansainvälisiin IEC-standardeihin. Euroopan tasolla tärkein sertifiointijärjestelmä on ETICS:n ylläpitämä ENEC-sertifiointijärjestelmä. Pääpiirteittäin ENEC-sertifiointi eroaa CB-sertifioinnista siten, että ENEC-sertifiointissa tulee huomioida myös kansallisten standardien vaatimukset. Suurin ero on kuitenkin

kin siinä, että ENEC-sertifiointi vanhenee, kun taas CB-sertifikaatti ei periaatteessa vanhene koskaan. ENEC-järjestelmässä on mukana myös seurantatestaukset, joissa vuosittain tarkastetaan, että tuote täyttää tietyiltä osin edelleen standardin vaatimukset, eikä tuotteisiin ole tehty luvattomia muutoksia valmistajan puolelta. ENEC-järjestelmän lisäksi Euroopassa on myös useita muita kansallisia sertifiointijärjestelmiä, kuten FI-järjestelmä Suomessa ja S-järjestelmä Ruotsissa. Näitä järjestelmiä hallinnoivat paikalliset sertifiointielimet, mutta testaus perustuu silti suurelta osin IEC-standardeihin.

Erilaiset standardit, standardien päivitykset sekä erilaiset sertifiointijärjestelmät aiheuttavat testauksen ja raportoinnin automatisoinnille ylläpitohaasteita. Jos yritys ottaa testauksessa käyttöön tekoälyä ja koneoppimista hyödyntävän sovelluksen, olisi sovelluksen päivityksestä huolehdittava tasaisin väliajoin ja yrityksellä olisi hyvä olla nimetty henkilö vain tätä tehtävää varten. Sovelluksen päivitys on haastava tehtävä, koska standardeja päivitetään melko usein ja standardien päivityksistä johtuvat mahdolliset sovellukseen tehtävät muutokset tulee aina validoida päivitysten jälkeen.

2.2 Testaus- ja sertifiointiprosessi

Testaus- ja sertifiointiprosessi voidaan jakaa karkeasti neljään vaiheeseen: testaus, tarkastus, sertifiointikatselmus ja sertifiointipäätös. Testaus- ja tarkastusvaihe suoritetaan laboratorion toimesta ja sertifiointikatselmus sekä sertifiointipäätös sertifiointielimen toimesta. Laboratorio ja sertifiointielin ovat toisistaan erillisiä toimijoita, eikä sertifiointielin saa olla mukana testausprosessissa. Tämän opinnäytetyön tutkimus keskittyy testausvaiheeseen, mutta työn tuloksia voidaan käyttää myös tarkastusvaiheen avuksi.

Testausvaiheessa tuotteelle suoritetaan standardin vaatimat testit ja tarkastukset. Testausta suorittava henkilö tekee testit standardin ohjeistamalla tavalla, standardin määräämässä järjestyksessä. Osa testeistä on silmämääräisiä tarkistuksia ja osa erilaisilla laitteilla ja mittareilla suoritettavia testejä ja mittauksia. Mittaustulokset tulee aina merkitä selkeästi raakadataan. Tarkastusten, testien ja mittausten perusteella, testausta suorittava henkilö täyttää testausraporttipohjaa.

Raportointipohjat ovat yleensä IEC:n virallisia raporttipohjia, jotka perustuvat tiettyjen standardien tiettyihin versioihin. Raporttipohjien ulkoasu sekä raporttipohjiin tehtävät merkinnät ovat tarkasti säänneltyjä. Testauksen onnistumisen kannalta tärkeitä asioita ovat standardien tuntemus, tarkkuus sekä oikeat ja tarkastetut mittalaitteet sekä ohjelmistot. Kaikki mittalaitteet tulevat kalibroida tasaisin väliajoin ja laitteiden mittauserävarmuuden tulee osua standardeissa määriteltyjen rajojen sisälle. Samaan tapaan kaikki testauksessa käytettävät ohjelmistot tulee validoida, jotta varmistetaan ohjelmistojen antamien tulosten oikeellisuudesta.

Valmis testausraportti tarkastetaan laboratorion toimesta testauksen valmistuttua. Tarkastusvaiheessa varmistetaan, että kaikki standardin vaatimat tarkastukset, testit ja mittaukset on suoritettu asiaankuuluvasti sekä tarkastetaan, että testausraportti on täytetty oikein ja selkeästi.

Laboratorion lopputarkastuksen jälkeen raportti lähetetään sertifiointielimen arvioitavaksi. Sertifiointielin tekee oman tarkastuksensa raportille sekä päätöksen sertifioinnista. Sertifiointipäätökseen ei vaikuta pelkästään laitteen testauksen tulokset, vaan sertifiointin yhteydessä varmistetaan myös valmistajan tehtaiden ja toiminnan laadusta.

2.3 Toiminnan nykytilanne

Tällä hetkellä kaikki edellisessä luvussa mainitut vaiheet ovat hyvin pitkälti manuaalista työtä. Tuotteiden arvokilvet ja tekninen aineisto tarkastetaan testausta suorittavan henkilön toimesta manuaalisesti ja verrataan standardin vaatimukseen. Hyvin usein joudutaan lukemaan koko tekninen aineisto läpi, jotta sieltä löydetään standardin vaatimat kiellot ja ohjeistukset. Aineisto käydään usein kokonaisuudessaan läpi myös tarkastusvaiheessa, jotta varmistetaan, ettei testausvaiheessa ole jäänyt oleellisia asioita huomioimatta.

Manuaalisen työn ongelmia ovat tähän kuluva aika, virhealttius sekä testauksen kokonaisuuden hahmottaminen. Monimutkaisten laitteiden tekninen dokumentaatio voi olla monitasavuinen. Tämän kokoisen aineiston läpikäynti vie paljon

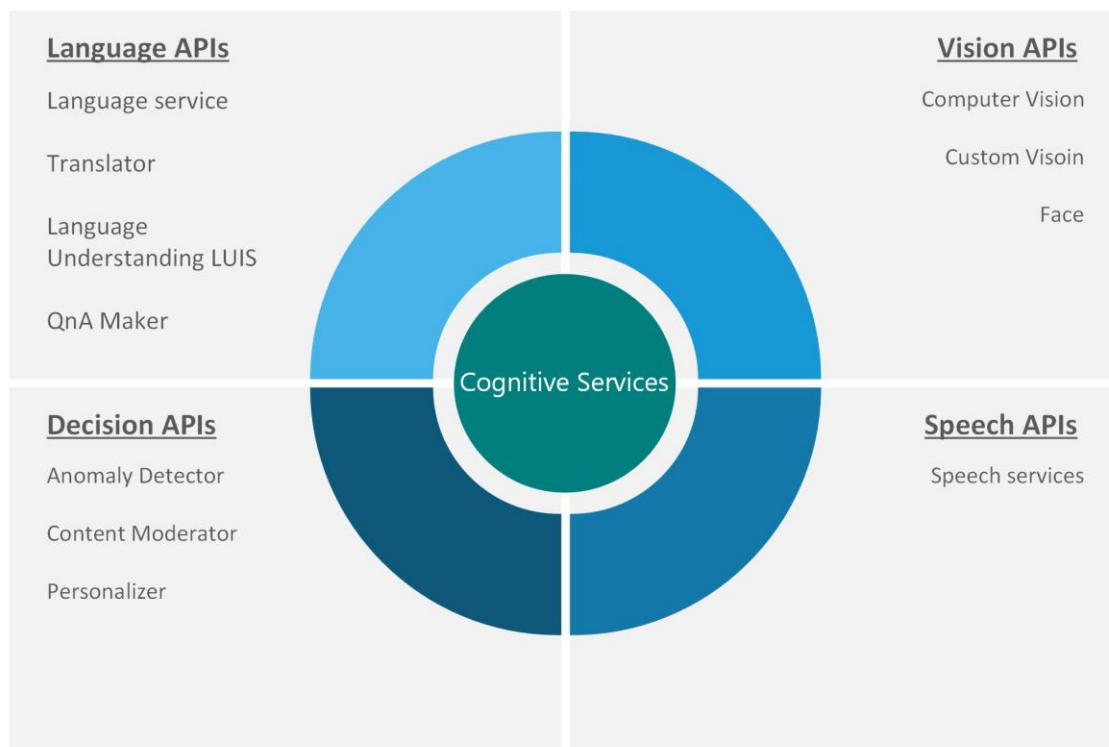
aikaa, koska aineiston seasta joudutaan etsimään standardin vaatimia oleellisia asioita. Usein työtä voidaan helpottaa esimerkiksi PDF-lukijan hakutoiminnolla, mutta aina tämä ei ole mahdollista. Aineiston koko vaikuttaa myös virheiden määrään, eli välttämättä ei löydetä kaikkea oleellista tietoa. Koska testaaaja joutuu keskittymään yksittäisten tietojen etsintään, voi kokonaiskuva laitteesta hämärtyä, jolloin testaaaja ei välttämättä havaitse kaikkia dokumentaation puutteita.

3 AZUREN COGNITIVE SERVICEN PALVELUT JA TEKNIIKAT

3.1 Yleiskatsaus Cognitive Services -palveluihin

Cognitive Services on Microsoftin tarjoama kokoelma pilvipohjaisia tekoälypalveluja Azure-alustalla. Cognitive Servicen käyttäjät saavat käyttöönsä Microsoftin valmiiksi kehittämiä tekoälymalleja, joihin päästään käsiksi joko kutsumalla niitä ohjelmistorajapintojen kautta, olemalla vuorovaikutuksessa näiden kanssa ohjelmistokehityspakettien avulla ja joidenkin palveluiden osalta malleihin voidaan olla yhteydessä selainpohjaisten sovellusten kautta. Ohjelmistokehityspakettien tuemia ohjelmointikieliä ovat C#, Go, Java, JavaScript, Python ja R. (Moniz ym. 2021, 1).

Cognitive Service -palvelut ovat jaoteltu kuvion 1 mukaisesti neljään peruskategoriaan: näkö (vision), puhe (speech), kieli (language) ja päätös (decision). Nämä peruskategoriat sisältävät erilaisia palveluja erilaisiin sovelluksiin.



KUVIO 1. Cognitive Services -kategoriat ja kategorioiden palvelut.

Cognitive Services -palvelut ovat pääsääntöisesti maksullisia. Azure tarjoaa palveluiden käyttöön ilmaisen tason, joka on pääsääntöisesti tarkoitettu yksittäisille henkilöille ja opiskelijoille, jotka haluavat tehdä palvelussa kokeiluja. Palveluiden kustannuksia voidaan arvioida Azuren tarjoamien työkalujen avulla, sekä palvelutaso voidaan valita palvelujen käytön mukaisesti. (Plan and manage costs of an Azure Cognitive Search service 2022.)

Cognitive Services -palvelussa on otettu huomioon palvelun turvallisuus. Turvallisuuteen kannattaa kiinnittää huomiota, koska kyseessä on pilvipalvelu. Koska käyttäjän ja palvelun välillä voi liikkua hyvinkin arkaluontoista dataa, on tärkeää, että ulkopuoliset tahot eivät pääse käsiksi dataan. Azure käyttää useita eri suojaustekniikoita pitääkseen palvelun turvallisena (Azure security baseline for Cognitive Services 2022).

Cognitive Services -palvelu tuo useita hyötyjä käyttäjälle. Koska palvelussa on laaja valikoima valmiiksi opetettuja tekoälyalgoritmeja, voidaan suuri määrä toimintoja suorittaa ilman valtavan datamäärän keräämistä ja algoritmien opettamista. Tämä säästää käyttäjän ajan lisäksi myös kustannuksia, koska suuren datamäärän varastoiminen sekä monimutkaisten algoritmien opetus vaativat suuret laite- tai palveluinvestoinnit. Lisäksi palvelu tuo tekoälyalgoritmit käyttöön suuremmalle yleisölle, koska palvelun käyttöön ei tarvita laajaa teknistä osaamista, vaan perustietämyksellä saadaan käyttöön tehokkaita työkaluja. Palvelun huonona puolena on se, että valmiita algoritmeja ei voida säätää. Tämän takia palvelun algoritmit eivät sovellu spesifeihin ja monimutkaisiin sovelluksiin.

Tässä opinnäytetyössä hyödynnettiin konenäkökategorian valmista OCR-algoritmia, konenäkökategorian mukautettua konenäköpalvelua sekä kielikategorian mukautettua entiteettitunnistuspalvelua.

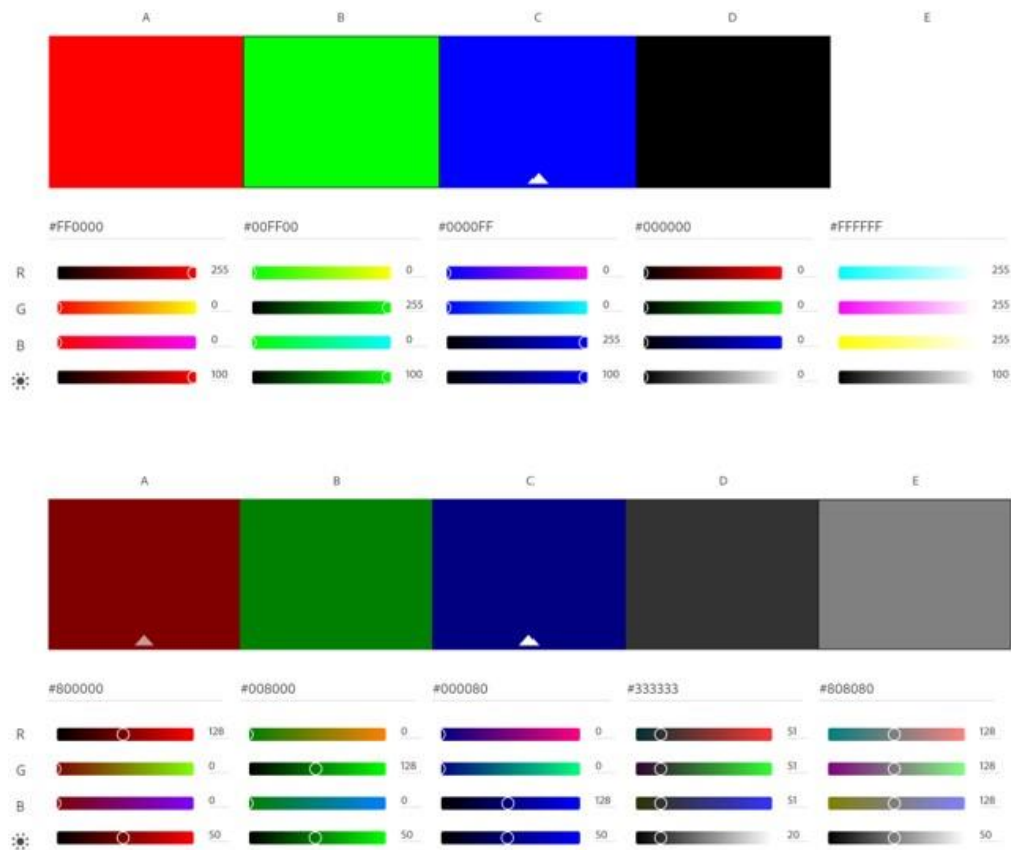
3.2 Konenäkö ja konvoluutioneuroverkot

Konenäkö perustuu pohjimmiltaan tietokoneen kykyyn käsitellä digitaalisessa kuvassa esiintyvien pikseleiden dataa ja pikselien välisiä suhteita. Digitaalinen kuva koostuu useista pikseleistä, esimerkiksi 100 x 100 pikselin kuvassa on yhteensä

10 000 pikseliä. Jokaiselle kuvan pikselille voidaan määrittää paikka sekä intensiteetti.

Esimerkiksi mustavalkoisen kuvan yksi pikseli saa arvon $I(x, y)$, jossa (x, y) on pikselin sijainti kuvassa ja I on pikselin intensiteetti-arvo, joka voi olla välillä 0–255. Intensiteetti-arvo on verrannollinen valon voimakkuuteen, joka vaikutti kameran sensoriin kuvanottohetkellä ja tallennettiin vastaavaan pikseliin. Täysin mustan pikselin intensiteetti-arvo on 0 ja täysin valkoisen pikselin 255. Mustavalkoista kuvaa voidaan täten käsitellä kaksiulotteisena intensiteettitaulukkona. Värikuva eroaa mustavalkokuvasta siten, että pikselin intensiteetti-arvo koostuu kolmesta arvosta: punaisen, vihreän ja sinisen värin intensiteeteistä. Tällöin pikseli saa paikka-arvon (x, y) sekä intensiteetti-arvon (r, g, b) . Esimerkiksi täysin punainen kuva saa intensiteetti-arvon $(255, 0, 0)$ ja täysin sininen kuva $(0, 0, 255)$. Koska värikuvan pikselin intensiteetti-arvo koostuu kolmen värikanavan arvoista, voidaan värikuvaa käsitellä moniulotteisena taulukkona tai kokoelmana kaksiulotteisiä taulukkoja, missä jokainen kaksiulotteinen taulukko vastaa yhtä värikanavaa. (Peters 2017, 12.)

Pikselin saamia intensiteetti-arvoja on havainnollistettu kuvan 1 avulla. Kuvassa kullakin värillä on neljä arvoa: R, G, B ja kirkkaus. Ylemmän väripaletin punainen, vihreä ja sininen ovat niin sanottuja todellisia arvoja, eli esimerkiksi punaisen värin RGB-arvo on $(255, 0, 0)$. Alemmassa väripaletissa on havainnollistettu kirkkauden vaikutusta RGB-arvoihin. Koska kirkkaus vaikuttaa myös värikanavien arvoon, ei pikselillä tarvitse olla erillistä kirkkaustietoa. Kuvasta voidaan nähdä myös, että valkoisella, mustalla ja harmaan eri sävyillä kaikki RGB-arvot ovat samat. Tästä syystä mustavalkoisen kuvan pikseli saa vain yhden intensiteetti-arvon.

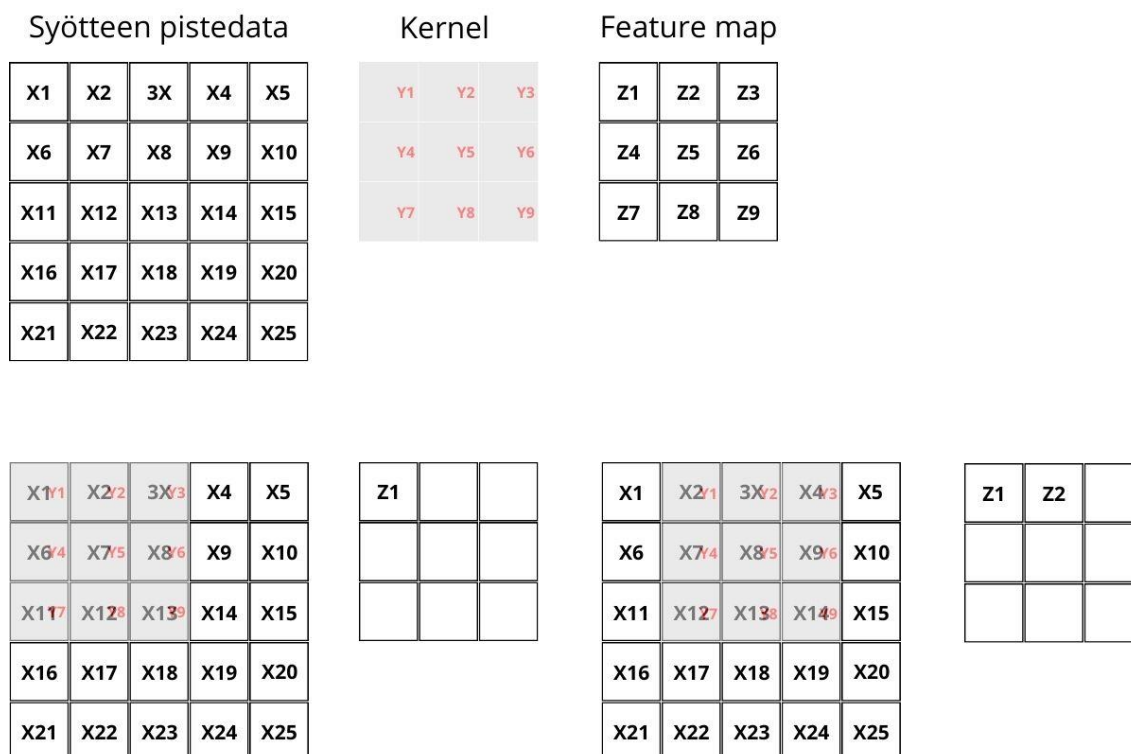


KUVA 1. Havainnollistava kuva digitaalisen kuvan pikselien saamista intensiteetti-arvoista.

Useat konenäön algoritmit ovat rakennettu konvoluutioneuroverkkojen avulla. Konvoluutioneuroverkossa käytettävä konvoluutio on tehokas tapa poimia kuvista ominaisuuksia, joiden avulla algoritmin tekemää kuvantunnistusta voidaan tehostaa. Tutkitaan seuraavaksi konvoluutiota mustavalkoisen kuvan osalta.

Konvoluutiossa keskeinen käsite on kernel. Kernel on käytännössä kaksiulotteinen matriisi painotuksilla. Kernelin avulla käydään läpi syötteen (mustavalkokuva) kaksiulotteista, pistedatasta koostuvaa matriisia. Kernel, eli pienempi matriisi liikkuu syötteen pistedatasta koostuvan suuremman matriisin poikki ja tekee jokaisen pysähdyksen kohdalla ruudun kolmanteen matriisiin. Kun kernel on käynyt läpi koko syötteen pistedatataulukon, eli on suoritettu yksi konvoluutio, saadaan uusi matriisi, jota kutsutaan ominaisuuskartaksi (feature map). Ominaisuuskartan datapisteiden arvot määräytyvät syötteen datapisteiden arvojen ja kernelin painotusten tulon mukaan. (Shafkat 2018.)

Kuvassa 2 havainnollistetaan kernelin liikkumista syötteen pistedatamatriisissa ja liikkumisen lopputuloksena syntyvää ominaisuuskarttaa. Ominaisuuskartan pisteen arvo Z on käytännössä syötteen yhden alueen pistearvojen X painotettu summa, jossa painokertoimet ovat kernelin arvot Y . X :n arvo on välillä 0–255.



KUVA 2. Esimerkki konvoluutiosta. Kernel käy läpi syötteen pistedatamatriisin, minkä tuloksena saadaan ominaisuuskartta (feature map).

Edellinen esimerkki on yksinkertaistettu esimerkki mustavalkoisesta kuvasta, jossa on vain yksi kanava. Värillisessä kuvassa kanavia taas on kolme: punainen, vihreä ja sininen. Tällöin konvoluutio tehdään jokaiselle kanavalle erikseen. Syötteen jokaisen kanavan pistedatataulukkoa käydään läpi omilla kernelillä, joilla jokaisella on erilaiset painokertoimet. Jokaisen kolmen kanavan prosessoitu tulos summataan suodattimen toimesta siten, että lopputuloksena on vain yksi ulostulokanava. (Shafkat 2018.)

Toinen konvoluutioon liittyvä keskeinen käsite on padding. Padding tarkoittaa käytännössä sitä, että syötteen pistedatamatriisiin reunoille lisätään ylimääräisiä pisteitä. Nämä pisteet ovat niin sanottuja valesiteitä, joille annetaan yleensä arvo 0. Tämä tehdään, jotta konvoluutiossa ei menetettäisi kuvan reunoilla olevaa, tunnistuksen kannalta merkittävää dataa. Toinen syy on se, että paddingin

avulla saadaan konvoluution tuloksena syntyvästä ominaisuuskartan matriisista samankokoinen, kuin syötekuvan matriisi. (Shafkat 2018.)

Konvoluutio tapahtuu mallin konvoluutiokerroksessa. Yhdessä konvoluutiokerroksessa voi olla useita kerneleitä. Jokaisella kernelillä on tällöin oma tehtävänsä, eli jokainen kernel tunnistaa syötekuvasta erilaisia ominaisuuksia. Tunnistettavia ominaisuuksia voivat olla esimerkiksi reunat, neliöt ja ympyrät. Mallissa voi olla myös useita konvoluutiokerroksia. Ensimmäisen konvoluutiokerroksen syöte on tunnistettava kuva, toisen konvoluutiokerroksen syöte on ensimmäisen konvoluutiokerroksen ulostulo, kolmannen konvoluutiokerroksen syöte on toisen konvoluutiokerroksen ulostulo ja niin edespäin.

Konvoluutiokerroksen yhteydessä käytetään usein pooling-kerrosta. Tämän kerroksen tarkoituksena on yksinkertaisesti pienentää konvoluutiokerrokselta toiselle siirtyvän datan määrää, pienentämällä ominaisuuskartan matriisin kokoa eli resoluutiota (Pramoditha 2022). Tämä tarkoittaa kääntäen sitä, että kerroksen avulla poimitaan tunnistuksen kannalta tärkeimmät ominaisuudet. Datamäärän pienentämisen lisäksi, kerroksen tarkoitus on tehdä mallista paikallisesti invariantti syötteen pienille muutoksille, kuten skaalaukselle ja asennolle, ja kerroksen ydintoimintona on välittää enemmän siitä, havaitaanko ominaisuus, eikä siitä, mikä on ominaisuuden sijainti syötteessä (Salvaris, Dean & Tok 2018).

Miksi sitten käyttää pooling-kerrosta, kun ominaisuuskartan kokoa on ensin kasvatettu padding-operaation avulla? Syy tähän on se, että pooling-kerroksen avulla saadaan ominaisuuskarttaa pienennettyä hallitusti ja älykkäästi, säilyttämällä sen sisältämät relevantit tiedot. Jos pienennys tehtäisiin pelkän konvoluution avulla, tunnistuksen kannalta merkittävää dataa katoaisi konvoluutiokerrosten välillä hallitsemattomasti.

Konvoluutioneuroverkon viimeinen osa on niin sanottu luokittelukerros (fully connected layer). Tämän kerroksen syötteenä toimivat konvoluutio- ja poolig-kerrosten läpikäynnin tuloksena syntyneet, korkeamman tason ominaisuuksia sisältävien ominaisuuskarttojen vektorit (Pokhrel 2019). Kun luokittelukerroksia laiteetaan useita peräkkäin, syntyy neuroverkko, jossa dataa siirretään verkossa eteenpäin neuronien painotusten ja aktivointifunktioiden tulosten mukaisesti.

Lopputuloksena kuva luokitellaan siihen luokkaan, jonka todennäköisyys on suurin.

Konvoluutioneuroverkon yksi hyöty on se, että verkossa on huomattavasti vähemmän yhteyksiä kuin perinteisessä neuroverkossa. Perinteisessä neuroverkossa jokainen kerros on täysin kytketty jokaiseen seuraavan kerroksen neurooniin. Kun kerrosten ja neuronien määrää lisätään, lisääntyy samalla verkossa esiintyvien parametrien määrä räjähdysmäisesti. Pienemmän parametrimääränsä takia, konvoluutioneuroverkko käyttää vähemmän muistia ja laskentatehoa verrattuna perinteiseen neuroverkkoon. (Salvaris ym. 2018.)

Toinen hyöty on niin kutsuttu parametrijako. Perinteisessä neuroverkossa neuroneiden sisääntulosignaalin painotukset määritellään vain kyseiselle neuronille, kun taas konvoluutioneuroverkossa käytetään samaa kerneliä monta kertaa. Sen sijaan, että opetetaan jokaisen neuronin kohdalla erilaiset parametrit, opetetaan joukko kerneleitä ja sovelletaan näitä kaikkiin neuroneihin. (Salvaris ym. 2018.)

3.3 Nimettyjen entiteettien tunnistus

Nimettyjen entiteettien tunnistus (named entity recognition) eli NER on yksi luonnollisen kielen prosessoinnin (natural language processing eli NLP) osa-alue. Nimettyjen entiteettien tunnistus tarkoittaa käytännössä sitä, että jäsentelemättömän tekstin seasta löydetään entiteettejä, kuten paikkojen nimiä, ihmisten nimiä, päivämääriä ja kellonaikoja. NER ei ole pelkästään itsenäinen työkalu tiedon poimintaan, vaan sillä on olennainen rooli myös muissa luonnollisen kielen prosessoinnin sovelluksissa, kuten tekstin ymmärryksessä, tiedon haussa, automaattisessa tekstin yhteenvedossa, kysymyksiin vastaamisessa sekä koneellisissa käännöksissä. (Li, J., Sun, Han & Li, C. 2022, 50.)

Li ym. (2022) mukaan entiteettien tunnistukseen on neljä lähestymistapaa. Ensimmäinen on sääntöihin perustuva lähestymistapa, jossa opetukseen ei tarvita käyttäjän merkkäamaa dataa, vaan tunnistukseen vaikuttavat säännöt määritellään manuaalisesti. Toinen on ohjaamattomaan oppimiseen perustuva lähestymistapa, joka perustuu ohjaamattomiin algoritmeihin ja jossa ei myöskään tarvita

käyttäjän merkkamaa opetusdataa. Kolmas lähestymistapa on ominaisuuksiin perustuva ohjattu oppiminen, joka perustuu entiteettien ominaisuuksiin ja opetusdatan merkkaamiseen. Neljäs lähestymistapa on syväoppimisen algoritmit, jotka tunnistavat käsittelemättömästä datasta automaattisesti tunnistukseen tarvittavia ominaisuuksia. (Li ym. 2022, 53–54.)

Manuaalisesti määritellyt säännöt on vanhin entiteettien tunnistusta lähestyvä tapa. Tässä tavassa turvaututaan manuaalisesti luotuihin sääntöihin, joiden perusteella entiteettien tunnistus suoritetaan. Useimmat manuaalista sääntöjen määrittelyä hyödyntävät järjestelmät perustuvat semanttisten ja syntaktisten sääntöjen manuaaliselle määrittelylle. Lähestymistapa toimii silloin, kun voidaan tunnistaa koko entiteettien tunnistukseen tarvittava sanasto. Säännöt voidaan kuitenkin luoda vain tietyn toimialueen sanastolle kerrallaan, joten jokaiselle toimialueelle on luotava omat sääntönsä. (Li ym. 2022, 53.)

Ohjaamattoman oppimisen lähestymistapa perustuu tyypillisesti ryhmittelyyn eli klusterointiin. Käytännössä tämä tarkoittaa sitä, että nimetyt entiteetit voidaan kerätä klusterityhmistä niiden kontekstin samankaltaisuuden perusteella. Kaikki ohjaamatonta oppimista hyödyntävät tavat perustuvat sanastoresursseihin, sanastollisiin rakenteisiin sekä tilastoihin, jotka ovat laskettu analysoimalla suurta joukkoa erilaisia tekstejä. (Nadeau & Sekine 2007.)

Ominaisuuksiin perustuva ohjattu oppiminen perustuu merkattuun opetusdataan ja entiteettien ominaisuuksien määrittelyyn. Tätä lähestymistapaa sovelletaan varsinkin silloin, kun entiteettejä on tarkoitus luokitella useampaan luokkaan. Peruseriaate on, että opetusdataan merkataan entiteetit ja näiden luokat. Malli oppii opetusdatan ja entiteettien ominaisuuksien avulla entiteettien rakenteet ja luo tunnistukseen säännöt. Ominaisuuksiin perustuvassa ohjatun oppimisen lähestymistavassa, ensiarvoisen tärkeää on entiteettien ominaisuuksien määrittäminen. (Li ym. 2022, 53–54.)

Nimettyjen entiteettien tunnistuksen tapauksessa, ominaisuuksilla tarkoitetaan algoritmien käyttöön tarkoitettuja sanojen kuvaajia tai tunnusomaisia attribuutteja. Sana voi saada esimerkiksi numeerisen attribuutin, joka kertoo sanassa esiintyvien merkkien määrän, sekä Boolean-attribuutin, jonka arvoksi voidaan

määrittää esimerkiksi tosi, kun sana alkaa isolla kirjaimella ja epätosi, kun sana alkaa pienellä kirjaimella. Entiteettien tunnistuksessa ja luokittelussa hyödynnetään tällöin tekoälymallin luomia tunnistussääntöjä yhdessä entiteettien ominaisuuksien kanssa. (Nadeau & Sekine 2007.)

Syväoppimisen algoritmeista on tullut hallitseva toteutustapa nimettyjen entiteettien tunnistuksessa. Li ym. (2022) mukaan syväoppimisen algoritmit tuovat kolme oleellista hyötyä nimettyjen entiteettien tunnistukseen, verrattuna muihin tekniikoihin. Ensinnäkin NER hyötyy epälineaarista muunnoksista, jotka luovat syötteestä epälineaarisen kuvauksen ulostuloa varten. Epälineaaristen funktioiden avulla malli voi oppia datasta monimutkaisempia piirteitä. Toiseksi syväoppimisen algoritmien avulla säästetään huomattavasti entiteettien ominaisuuksien suunnitteluun ja määrittämiseen kuluvaan aikaan ja resursseihin, koska syväoppimisen algoritmit kykenevät automaattisesti tunnistamaan datasta erilaisia piirteitä ja ominaisuuksia. Kolmanneksi syväoppimisen algoritmien avulla voidaan luoda hyvinkin monimutkaisia NER-järjestelmiä. (Li ym. 2022, 54.)

Nimettyjen entiteettien tunnistukseen ja luokitukseen voidaan käyttää useita erilaisia syväoppimisen algoritmeja. Li ym. (2022) esittää yleisen nimettyjen entiteettien tunnistukseen käytettävän syväoppimismallin taksonomian, joka koostuu syötteiden hajautetusta esityksestä, kontekstienkooderista sekä merkkien eli tagien dekodeerista. Näihin eri osa-alueisiin voidaan käyttää erilaisia algoritmeja riippuen mallin käyttösovelluksesta. (Li ym. 2022, 54.)

3.4 Metriikat mukautettujen mallien arviointiin

Cognitive Services -palvelussa luotujen mukautettujen mallien hyvyttä voidaan arvioida erilaisilla metriikoilla, joita eri palvelut tuottavat automaattisesti. Palvelujen tuottamia mittareita ovat sisäinen tarkkuus (precision), herkkyys (recall), F1-tulos, sekaannusmatriisi sekä objektintunnistusmallin osalta keskimääräinen tarkkuus (AP).

Sekaannusmatriisi on taulukkoesitys mallin tekemistä luokitteluluennuksista. Sekaannusmatriisin avulla huomataan, miltä osin malli suoriutui hyvin ja miltä

osin malli teki virheitä. Sekaannusmatriisiin liittyy neljä keskeistä käsitettä, jotka ovat oikea negatiivinen luokitus, oikea positiivinen luokitus, väärä negatiivinen luokitus ja väärä positiivinen luokitus (Grigoriev 2021). Kuvassa 3 on esitetty yksinkertainen kahden luokan luokittelutilannetta kuvaava sekaannusmatriisi.

		Mallin ennusteet	
		Kyllä	Ei
Todelliset arvot	Kyllä	oikea positiivinen	väärä negatiivinen
	Ei	väärä positiivinen	oikea negatiivinen

KUVA 3. Esimerkki kahden luokan luokituksen sekaannusmatriisista.

Mallin sisäinen tarkkuus ilmoittaa, kuinka suuri osa kaikista positiivisiksi luokitelluista tapauksista oli oikeasti positiivisia. Sisäinen tarkkuus voidaan laskea sekaannusmatriisin avulla. Matemaattisesti sisäinen tarkkuus lasketaan kaavalla

$$\text{sisäinen tarkkuus} = \frac{\text{oikeat positiiviset}}{(\text{oikeat positiiviset} + \text{väävät positiiviset})}, \quad (1)$$

eli jakamalla oikeat positiiviset luokitukset kaikilla positiivisilla luokituksilla, eli oikeiden positiivisten luokitusten ja väärin positiivisten luokitusten summalla. Mitä suurempi mallin sisäinen tarkkuus on, sitä varmempi mallin tekemä positiivinen luokitus on. (Grigoriev 2021.)

Mallin herkkyys lasketaan myös sekaannusmatriisin avulla. Herkkyys kertoo, kuinka suuren osan kaikista positiivista tapauksista mallin onnistui luokittelemaan positiivisiksi. Matemaattisesti herkkyys lasketaan kaavalla

$$\text{herkkyys} = \frac{\text{oikeat positiiviset}}{(\text{oikeat positiiviset} + \text{väärät negatiiviset})}, \quad (2)$$

eli jakamalla oikeat positiiviset luokitukset oikeiden positiivisten luokitusten ja väärin negatiivisten luokitusten summalla. Mitä suurempi mallin herkkyys on, sitä todennäköisemmin malli löytää relevantin tiedon. (Grigoriev 2021.)

Tutkitaan vielä, kuinka mallin sisäinen tarkkuus ja herkkyys vaikuttavat mallin suorituskykyyn, kun kyseessä on entiteettien tunnistus ja arvioitavana on metriikat yhden entiteetin osalta. Taulukossa 1 on esitetty tilanteita, joissa tarkkuus ja herkkyys on joko suuri tai pieni. Hyvin usein joudutaan arvioimaan, onko mallin sisäinen tarkkuus vai herkkyys tärkeämpi mallin toiminnan kannalta, koska toista kasvattamalla, toinen yleensä pienenee (Grigoriev 2021).

Taulukko 1. Sisäisen tarkkuuden ja herkkyyden vaikutus mallin suorituskykyyn. (Evaluation metrics for custom named entity recognition models 2022.)

Herkkyys	Tarkkuus	Tulkinta
Suuri	Suuri	Entiteetti tunnistetaan hyvin
Pieni	Suuri	Entiteettiä ei aina tunnisteta, mutta jos tunnistetaan, on tunnistuksen luotettavuus korkea
Suuri	Pieni	Entiteetti tunnistetaan hyvin, mutta tunnistuksen luotettavuus on matala, koska entiteetti tunnistetaan joskus toisena
Pieni	Pieni	Entiteettiä ei yleensä löydetä ja jos löydetään, tunnistuksen luotettavuus on matala

F1-tulos kokoaa mallin sisäisen tarkkuuden ja herkkyyden yhdeksi luvuksi. F1-tulos on hyvä mittari, kun pyritään tasapainottamaan mallin tarkkuus ja herkkyys. Matemaattisesti F1-tulos lasketaan kaavalla

$$F1 = \frac{2 \times \text{sisäinen tarkkuus} \times \text{herkkyys}}{(\text{sisäinen tarkkuus} + \text{herkkyys})}. \quad (3)$$

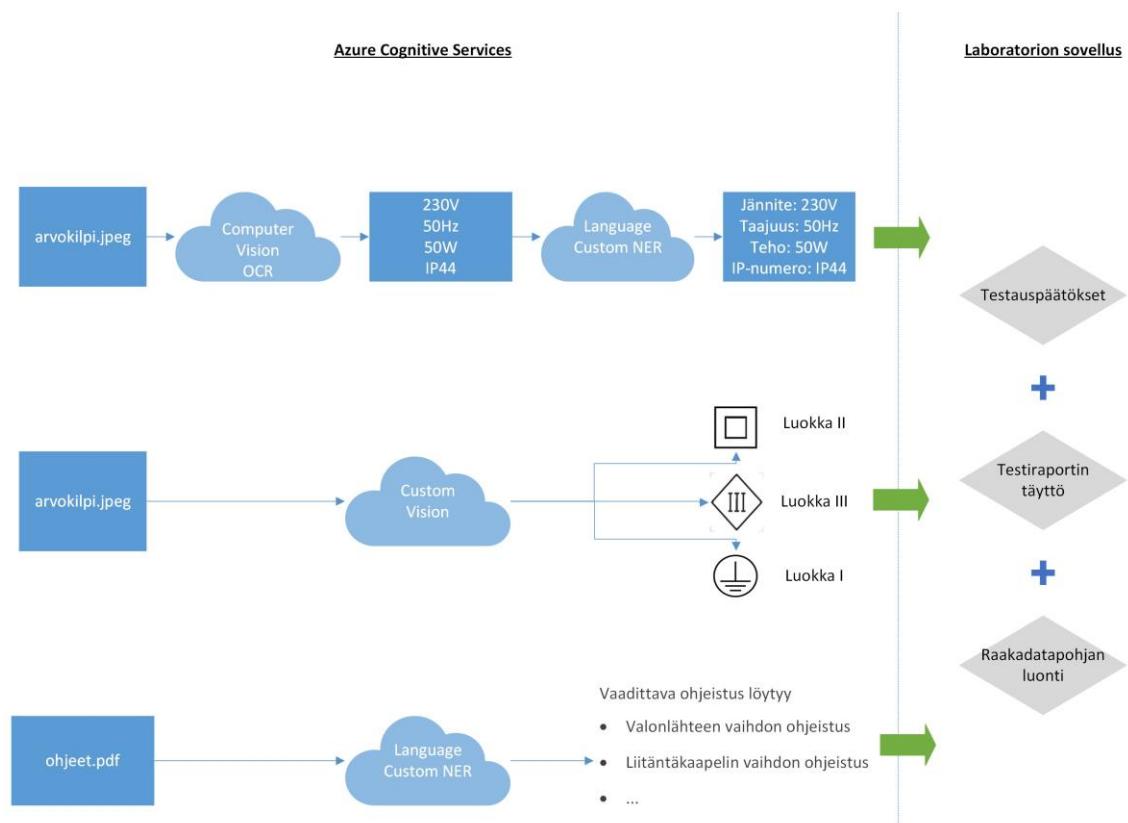
Sen sijaan, että valittaisiin tunnistuksen kynnyisarvo, joka kasvattaa joko tarkkuutta tai herkkyyttä, voidaan valita tunnistuksen kynnyisarvo, joka maksimoi F1-tuloksen. (Grigoriev 2021.)

Objektintunnistusmallia arvioidessa, yksi metriikoista on keskimääräinen tarkkuus. Keskimääräinen tarkkuus ottaa luokittelun lisäksi huomioon myös mallin tekemän objektien paikanmäärittelyn. Keskimääräinen tarkkuus lasketaan useiden muiden metriikoiden avulla, kuten sisäisen tarkkuuden, herkkyyden sekä rajauslaatikoiden suhteen avulla. Matemaattisesti keskimääräisen tarkkuuden voidaan käytännössä ajatella olevan sisäisen tarkkuuden ja herkkyyden funktion kuvaajan (precision-recall curve) alle jäävä alue. (Anwar n.d.)

4 SOVELLUS

4.1 Sovelluksen tavoite ja peruseriaate

Opinnäytetyön tavoitteena oli luoda pohja tekoälyä hyödyntävälle sovellukselle, joka nopeuttaa valaisimien ja muiden sähkölaitteiden sähköturvallisuustestausprosessia. Sovelluksen perustoimintaperiaate esitetään kuviossa 2. Sovelluksen tekoälyosuus jakautuu karkeasti kolmeen eri osa-alueeseen ja nämä osa-alueet jakautuvat eri toimintoihin. Ensimmäinen osa-alue analysoi kirjoitettua tekstiä tuotteiden arvokilvistä, toinen osa-alue analysoi arvokilvissä esiintyviä kuvaturunuksia ja kolmas osa-alue tunnistaa standardin vaatimat ohjeistukset kirjallisesta dokumentaatiosta.



KUVIO 1. Sovelluksen toimintaperiaate ja käytetyt Cognitive Service -palvelut.

Sovelluksen määrittelyssä keskityttiin kuvion 2 vasemmanpuoleiseen osioon, eli siihen, mitä toimintoja tulee suorittaa milläkin Cognitive Services -työkalulla. Mää-

rittelyssä otettiin myös kantaa siihen, miten Cognitive Services -palvelusta saatuja tuloksia voidaan käyttää hyväksi, mutta tulosten hyödyntämisen tekninen toteutus jätettiin tulevaisuuden haasteeksi.

Sovelluksen määrittelyssä käytettiin pohjana valaisinstandardin IEC 60598-1:2020 vaatimuksia. Vaikka määrittely tehtiin valaisinstandardin 2020-version pohjalta, voidaan samoilla toiminnoilla ja työkaluilla skaalata sovellus tulevaisuuden uusia vaatimuksia vastaavaksi sekä toteuttaa samanlainen sovellus toisen laitestandardin vaatimuksia vastaavaksi.

4.2 Tuotteiden arvokilpien analysointi

Laitteiden turvallisuus- ja suorituskykystandardit määrittelevät, mitä tietoja laitteissa tulee olla. Standardit usein määrittelevät myös, missä kohdassa laitetta tiettyjen merkintöjen tulee olla näkyvissä. Käytännöllisistä syistä, laitevalmistajat pyrkivät laittamaan mahdollisimman paljon vaadituista merkinnöistä laitteen arvokilpeen. Kaikkia standardin vaatimia merkintöjä ei voida merkitä laitteen arvokilpeen, vaan ne tulee merkitä laitteen tietyn osan läheisyyteen. Osa vaadituista merkinnöistä on tekstiä ja osa standardissa määriteltyjä kuvatunnuksia.

Merkinnät voidaan karkeasti jakaa informatiivisiin merkintöihin sekä varoittaviin merkintöihin. Varoittavia merkintöjä ovat esimerkiksi sähköiskun vaarasta varoitettava kuvatunnus sekä valonlähteen tuijotuskiellosta kertova kuvatunnus. Informatiivisia merkintöjä ovat esimerkiksi jännite, teho, IP-numero sekä tyyppimerkintä. Jokaisessa laitteessa ei tarvitse olla kaikkia standardissa mainittuja merkintöjä, vaan merkinnät määräytyvät osittain laitteen ominaisuuksien mukaan. Valaisimissa on kuitenkin joitain merkintöjä, jotka tulevat aina olla merkittynä riippumatta valaisimen rakenteesta. Näitä merkintöjä ovat alkuperämerkintä, tyyppi-merkintä, nimellistehon merkintä sekä nimellisjännitteen merkintä. Muita usein käytettäviä merkintöjä ovat suojausluokan merkintä, ympäristön käyttölämpötilan merkintä sekä IP-numero, joka ilmaisee laitteen suojaustason vierasesineitä, pölyä ja vettä vastaan.

Testauksessa keskitytään ensisijaisesti siihen, että tuotteesta löytyvät kaikki kyseiseen tuotteeseen vaadittavat merkinnät. Merkinnöillä on testauksessa myös toinen oleellinen rooli. Niiden perusteella määräytyvät osittain testausmenetelmät ja testauksessa käytettävät arvot, kuten testijännite.

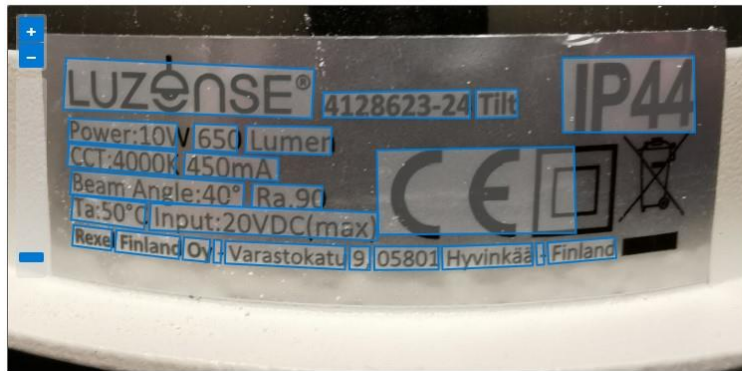
4.3 Kirjoitetun tekstin tunnistus arvokilvistä

Kirjoitetun tekstin tunnistukseen käytettiin Cognitive Servicen näkökategorian OCR-algoritmia, joka on luotu optiseen merkkien tunnistukseen. Algoritmia voidaan käyttää joko Rest API:n välityksellä, ohjelmistokehityspakettien avulla tai selainpohjaisen Vision Studio -palvelun avulla. Vision Studio eroaa muista tavoista siten, että tätä ei voida yhdistää käyttäjän omaan sovellukseen, vaan palvelu on itsenäinen. Kaikki vuorovaikutustavat ottavat kuitenkin yhteyden samaan algoritmiin.

Read API:n syöte on joko kuvatiedosto tai dokumenttiedosto seuraavanlaisilla ominaisuuksilla:

- Tiedostomuodon tulee olla JPEG, PNG, BMP, PDF tai TIFF.
- PDF- ja TIFF-tiedostojen enimmäispituus on 2000 sivua.
- Kuvatiedostojen tiedostokoon tulee olla enintään 500 MB, vähintään 50 x 50 pikseliä ja enintään 10000 x 10000 pikseliä. PDF-dokumenttien tiedostokoossa ei ole rajoituksia.
- Pienin tekstinkorkeus, joka voidaan tunnistaa, on 12 pikseliä 1024 x 768 kokoisessa kuvassa. (Call the Computer Vision 3.2 GA Read API 2022.)

Algoritmi on valmiiksi opetettu tunnistamaan koneellisesti ja käsin tuotettua tekstiä. Koska algoritmi ei ole käyttäjän säädettävissä, tutkimus keskittyi valmiin algoritmin tunnistuskykyyn. Tunnistuskykyä tutkittiin Vision Studio -palvelun avulla, koska palvelu mahdollistaa kuvien nopean analysoinnin. Palvelu tarjoaa JSON-tulosteen lisäksi myös tulosteen, josta nähdään pelkästään tunnistetut tekstit sekä visualisointi tunnistetuista sanoista. Kuvassa 4 on esimerkki Vision Studio -palvelun tulosteista.



Detected attributes JSON

```

LUZENSE®
IP44
4128623-24 Tilt
Power:10W 650 Lumen
CCT:4000K 450mA
Beam-Angle:40° Ra.90
CEL
Ta:50°C Input:20VDC(max)
Rexel Finland Oy - Varastokatu 9, 05801 Hyvinkää - Finland

```

KUVA 4. Esimerkki OCR-algoritmin tekstimuotoisesta tulosteesta.

Mitkä seikat sitten vaikuttavat arvokilvistä tehtävään merkkien tunnistukseen. Kysymykseen etsittiin vastausta pohtimalla ja kokeilemalla, kuinka seuraavat kuvaolosuhteet sekä kuvissa esiintyvät asiat vaikuttavat tunnistuksen lopputulokseen:

- valomäärä
- valonlähteiden heijastukset
- kuvauskulma
- arvokilven asettelu (rypyssä vai suorassa)
- tekstin väri suhteessa pohjan väriin
- tekstin koko.

Kuten kaikessa valokuvauksessa, valomäärä vaikuttaa paljon kuvan laatuun. Jos kameran sensoriin tuulee liikaa valoa tai liian vähän valoa, katoaa kuvasta yksityiskohdat. Järjestyskameralla kuvatessa voidaan käyttää hyväksi histogrammia, joka kertoo kameran sensoriin tulevan valomäärän tasapainon. Histogrammin vasemmalla reunassa ovat tummat sävyt ja oikeassa reunassa vaaleat sävyt. Tämä tarkoittaa käytännössä sitä, että jos histogrammin kuvaaja on kokonaan vasemmalla, valokuvasta tulee täysin musta ja jos kuvaaja on kokonaan oikealla, valokuvasta tulee täysin valkoinen. Kummassakaan kuvassa ei tällöin ole yksityiskohtia. Hyvä perussääntö on valottaa kuva siten, että histogrammi saadaan mahdollisimman tasaisesti näiden kahden ääripään väliin, jolloin kuvassa olevan informaation määrä saadaan maksimoitua.

Huonosta valaistuksesta aiheutuvien, arvokilvissä esiintyvien valonlähteiden heijastuksien huomattiin vaikuttavan negatiivisesti merkkien tunnistukseen. Tunnistusalgoritmi ei tehnyt tunnistusta ollenkaan tai teki väärän tunnistuksen heijastuksen alle jäävistä merkeistä, samaan tapaan kuin ihmissilmä. Heijastukset saadaan minimoitua oikein valaistulla kuvauspaikalla. Kuvauspaikan valaistus tulee toteuttaa erittäin hajotetulla valolla, jolloin kirkkaita pisteitä ei esiinny ollenkaan. Toinen heijastuksiin vaikuttava seikka on valonlähteiden asettelu. Valonlähteitä ei tulisi asettaa suoraan kuvattavan kohteen päälle. Parhaiten heijastukset saadaan minimoitua suorittamalla arvokilpien kuvaus niin sanotun valolaatikon sisällä, jolloin arvokilven pintaan saadaan tasainen hajotettu valo.

Jos arvokilven kuvauspaikka on huono, eli valonlähteet heijastuvat kirkkaana arvokilven pinnasta, tai arvokilpi on liimattu tuotteessa hankalaan paikkaan, voidaan kuvaus joutua suorittamaan muusta kulmasta, kuin suoraan ylhäältä päin. Kuvauskulman vaikutusta merkkien tunnistukseen tutkittiin suorittamalla tunnistus kuvalle, joka oli otettu noin 45 asteen kulmassa suhteessa arvokilven tason normaaliin. Tästä kulmasta otetusta kuvasta algoritmi tunnisti tekstit hyvin. Tätä pienemmästä kulmasta ei arvokilpeä kannata laitteen testauksen yhteydessä kuvata, koska raportointimielessä arvokilpi tulisi kuvata mahdollisimman kohtisuoraan ylhäältä päin.

Heijastusten lisäksi toinen tunnistukseen negatiivisesti vaikuttavan seikan huomattiin olevan arvokilven asettelu, eli onko arvokilpi liimattu suoraan tasaiselle alustalle, esiintyykö arvokilvessä ryppyjä tai onko arvokilpi liimattu epätasaiselle pinnalle. Arvokilvessä esiintyneet rypyt aiheuttivat vääriä tunnistuksia, eli ryppyjen alle jäävät sanat ja merkit tunnistettiin väärin. Arvokilven epätasainen kiinnityspinta ei niinkään aiheuttanut vääriä tunnistuksia, mutta aiheutti toisen yllättävän tunnistusongelman. Kokeiluissa huomattiin, että arvokilven tekstit voivat tiettyissä olosuhteissa heijastua valaisimen rungon pinnasta. Tunnistusalgoritmi tunnisti nämä heijastukset merkeiksi, jolloin tulosteessa esiintyi ylimääräisiä attribuutteja. Kyseiset tunnistusongelmat voidaan korjata sillä, että arvokilpi asetetaan kuvaustilanteessa mahdollisimman tasaisesti, mahdollisimman tasaiselle alustalle ja kuva otetaan kohtisuoraan arvokilven yläpuolelta. Joissain tapauksissa arvokilpi joudutaan irrottamaan laitteesta kuvaustilanteen ajaksi, parhaimman mahdollisen lopputuloksen aikaansaamiseksi.

Vaikka arvokilven tekstin ja pohjan värien suhde sekä tekstin koko vaikuttavat tunnistukseen, ei näitä kuitenkaan tutkittu erikseen kokeiluiden yhteydessä. Tämä johtui siitä, että arvokilvissä on yleisesti ottaen aina tummaa tekstiä vaalealla pohjalla tai vastaavasti vaaleaa tekstiä tummalla pohjalla. Tekstin koon vaikutusta ei tutkittu, koska yleensä standardit määrittävät arvokilvissä esiintyvälle tekstille vähimmäiskoon.

Vaikka kokeiluissa esiintyi tilanteita, joissa algoritmi ei kyennyt täydelliseen merkkien tunnistukseen, on OCR-algoritmi kuitenkin erittäin tehokas työkalu. Paras mahdollinen tunnistus saadaan tehtyä, kun arvokilven pintaan tulee riittävästi valoa ja kuva on valotettu oikein, pinnasta ei heijastu valonlähteitä, kuva otetaan suoraan arvokilven yläpuolelta ja arvokilpi on asetettu tasaisesti tasaiselle alustalle. On muistettava, että määriteltävän sovelluksen käyttöpaikka on testauslaboratorio, jossa edellytetään muutenkin hyviä ja tarkkoja kuvia tuotteista, joten laboratoriolta voidaan olettaa olevan kuvaukseen soveltuva kuvauspaikka sekä osaamista tarkkojen kuvien ottamiseen. Esimerkkejä suoritetuista kokeiluista sekä näiden tuloksista on esitetty liitteessä 1.

4.4 Kirjoitetun tekstin luokittelu

Merkkien tunnistamisen lisäksi sovelluksen tulee myös ymmärtää tekstiä. Tässä tapauksessa ymmärrys tarkoittaa tunnistetun tekstin eri osien luokittelua.

Tekstin luokittelu toteutettiin kielikategorian Custom NER -palvelulla eli mukautetulla entiteettitunnistus palvelulla. Tämän palvelun avulla käyttäjä voi itse opettaa algoritmin tunnistamaan tekstistä haluamansa entiteetit. Sovellus määriteltiin tunnistamaan arvokilven kirjoitetusta tekstistä nimellisteho, nimellinen syöttöjännite, nimellistajuus, IP-numero sekä käyttölämpötila. Kyseiset attribuutit valittiin, koska nämä esiintyvät yleensä valaisimien arvokilvissä. Lisäksi näiden attribuuttien perusteella voidaan tehdä useita testauspäätöksiä.

Algoritmin opetusprosessiin sisältyi viisi vaihetta: entiteettien määrittely, datan valinta ja kerääminen, datan valmistelu, algoritmin opetus ja algoritmin arviointi.

Datan valmistelu ja algoritmin opetus tehtiin Language Studio -palvelussa. Palvelusta saatiin myös opetetun mallin metriikat, joiden avulla voitiin arvioida opetetun mallin hyvyyttä.

4.4.1 Opetusdata

Algoritmin opetus aloitettiin opetusdatan keräämisellä. Azuren dokumentaatiosta löytyi ohjeita liittyen opetusdataan. Dokumentaation mukaan seuraavat asiat tulee ottaa huomioon parhaimman lopputuloksen saavuttamiseksi:

- Opetusdatan tulisi olla todellista oikean elämän dataa. Opetusdatana voidaan periaatteessa käyttää myös synteettistä eli itse luotua, todellista dataa vastaavaa dataa, mutta tällöin mallin suorituskyky todellisissa tilanteissa voi kärsiä.
- Opetusdatan tulisi olla mahdollisimman monimuotoista, kuitenkin siten, ettei datan monimuotoisuus eroa liikaa todellisen datan monimuotoisuudesta. Toisin sanoen, opetusdatan tulisi sisältää esimerkkejä mahdollisimman laajasti todellisessa elämässä esiintyvistä muunnelmista.
- Opetusdatan tulisi olla mahdollisimman monipuolista, jotta vältytään mallin ylisovittamiselta.
- Opetusdatan ei tulisi sisältää dokumenttien kaksoiskappaleita. Kaksoiskappaleet vaikuttavat negatiivisesti opetusprosessiin, mallin arviointimitareihin sekä mallin suorituskykyyn. (How to prepare data and define a schema for custom NER 2022.)

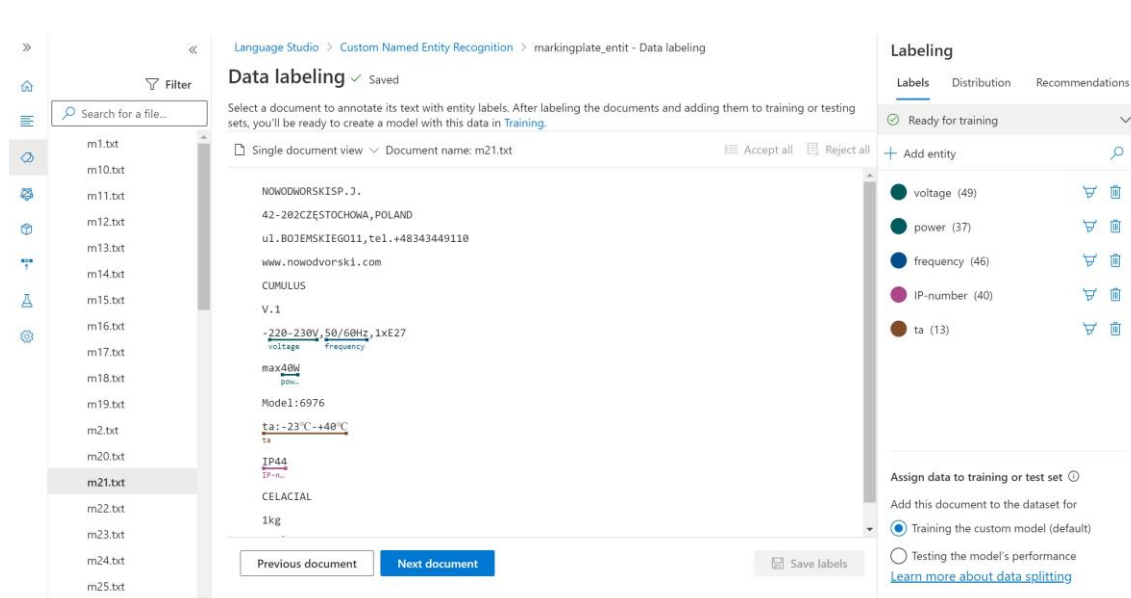
Opetusdatana käytettiin aidoista arvokilvistä OCR-algoritmin avulla saatuja tekstintunnistuksen tuloksia. Datan keräys ja valmistelu tehtiin käytännössä siten, että osa arvokilpikuvista ladattiin Tukesin vaarallisten tuotteiden rekisteristä ja osa arvokilpikuvista otettiin erilaisista laitteista. Kun kuvapankki oli kerätty, jokaisesta kuvasta tunnistettiin teksti OCR-algoritmin avulla ja jokaisesta tunnistuksesta tehtiin oma tekstitiedostonsa.

Suurin osa opetusdatasta oli siinä muodossa, kuin se saatiin OCR-algoritmin tulosteena. Osaan instansseista tehtiin kuitenkin pieniä muutoksia. Esimerkiksi

joissain opetusdatan arvokilvissä sähköiset arvot olivat ilmoitettu taulukkomuodossa. Tällöin ORC-algoritmin tulosteessa entiteetin numeerinen arvo ja yksikkö eivät olleet peräkkäin. Tämän kaltainen datarakenne olisi vaikeuttanut seuraavassa vaiheessa tehtävää entiteettien merkkausta. Dataa arvioidessa huomattiin myös, että IP-numeroita ja käyttölämpötilamerkintöjä oli vähän suhteessa muihin entiteetteihin. Näitä kahta entiteettiä lisättiin opetusdataan, jotta entiteettejä sisältäviä instansseja saataisiin lisää, mikä parantaa algoritmin suorituskykyä.

4.4.2 Opetuksen valmistelu ja mallin opetus

Opetusdatan keräämisen jälkeen opetusdataan merkattiin opetettavat entiteetit. Merkkaaminen tehtiin Language Studio -palvelussa. Kuvassa 5 nähdään esimerkki datanmerkintätyökalusta sekä eri entiteettien merkkaustavoista.



KUVA 5. Esimerkki eri entiteettien merkkaustavoista ja Language Studio -palvelun datanmerkkaustyökalusta.

Azuren dokumentaatiosta löytyi opetusdatan laadun ohjeistuksen lisäksi myös ohjeistus koskien entiteettien merkkausta. Dokumentaation mukaan entiteetit tulisi merkitä dataan mahdollisimman tarkasti, johdonmukaisesti ja täydellisesti, jotta opetettu malli toimisi mahdollisimman hyvin.

- Jokainen entiteetti tulee merkata aina oikean tyyppisenä.

- Datasta tulee merkata vain se, mitä halutaan tunnistaa. Ylimääräisen datan merkkauksesta tulee välttää.
- Samalla entiteetillä tulee olla sama tunniste läpi koko opetusdatan.
- Kaikki opetusdatassa esiintyvät entiteetin instanssit tulee merkata. (Label your data in Language Studio 2022.)

Ennen varsinaista entiteettien merkkauksta, jokaiselle entiteetille päätettiin merkkauksäännöt. Merkkauksäännöillä pyrittiin lähinnä siihen, että jokainen entiteetti olisi merkattu yhdenmukaisesti läpi koko opetusdatan. Säännöt olivat tarpeen, koska entiteettien kirjoitustyyli vaihteli hieman eri arvokilvissä.

Syöttöjännite oli useimmissa arvokilvissä merkitty jännitealueena, esimerkiksi 220–240 V, mutta muutamassa arvokilvessä vain tietynä lukuna, esimerkiksi 230 V. Yhteistä näillä merkintätavoilla oli se, että ensimmäinen merkki oli numero, yleensä vielä numero kaksi, ja viimeinen merkki kirjain V. Tämän seurauksena jännite-entiteetti päätettiin merkata dataan siten, että entiteetti alkoi aina numerolla ja päättyi jännitteen yksikköön eli V-kirjaimen.

Syöttöjännite-entiteetin tunnistuksen suurimman haasteen oletettiin olevan arvokilvissä esiintyvät muut jännitemerkinnät, esimerkiksi toisiopiirin jännitemerkintä. Tapaukset, joissa valaisimen arvokilvessä olisi kaksi jännitemerkintää ovat erittäin harvinaisia, joten tätä ei koettu suureksi ongelmaksi.

Tehoentiteetti päätettiin merkata aina siten, että entiteetti alkoi numerolla ja päättyi tehon yksikköön eli W-kirjaimen. Entiteetille riitti yksinkertainen sääntö, koska valaisimien arvokilvissä on melkein aina vain yksi tehomerkinä, joka on muotoa $n W$ (n ollessa tehon numeerinen arvo). Poikkeuksena ovat valaisimet, joissa valonlähteenä on lamppu. Tällöin yhden lampun valaisimessa, teho ilmoitetaan yleensä sallittuna maksimitehona muodossa $\text{max. } n W$ ja esimerkiksi kolmen lampun valaisimessa muodossa $3 \times \text{max. } n W$. Tehoentiteetin kohdalla tehtiin kuitenkin päätös jättää lamppujen tuomat lisämerkinnät pois, koska ainoastaan kokonaisteholla on merkitystä testauspäätöksissä.

Tehoentiteetin tunnistuksen haasteena huomattiin, että muoto $n W$ voi esiintyä myös valaisimien arvokilvissä esiintyvissä valmistuserätunnuksissa, koska $n W$

voi tarkoittaa myös valmistusviikkoa (n ollessa viikkonumero). Tämän arvioitiin voivan aiheuttaa vääriä positiivisia tunnistuksia.

Taajuusentiteetti merkattiin siten, että entiteetti alkoi aina numerolla ja päättyi taajuuden yksikköön Hz. Taajuusentiteetin osalta merkkauksäännön luonti ja yhtenäinen merkkaustapa olivat erityisen tärkeitä, koska taajuusmerkintä esiintyi arvokilvissä useissa eri kirjoitusmuodoissa. Arvokilvissä esiintyi taajuusmerkinnän kirjoitusmuotoja, kuten n Hz, n-n Hz, n / n Hz ja n Hz / n Hz (n ollessa taajuuden numeerinen arvo, yleensä 50 tai 60).

IP-numeron merkkauksen säännöksi määriteltiin yksinkertaisesti, että entiteetti oli aina muotoa IPYX, kahden viimeisen merkin ollessa numeroita, esimerkiksi IP20 tai IP44. Entiteetin merkkauksen yhteydessä ei huomattu seikkoja, jotka saattaisivat huonontaa entiteetin tunnistuskykyä. IP-numero esiintyi yleisesti ottaen omana tietona, eli omana rivinä tekstitiedostossa, mutta joissain harvoissa tapauksissa IP-merkintä oli sisällytetty tuotteen mallimerkintään. Tämän ei kuitenkaan oletettu aiheuttavan suuria haasteita, koska entiteetti oli selkeä.

Käyttölämpötilamerkintä eroaa hieman edellä mainituista entiteeteistä. Kun edelliset entiteetit rakentuvat pohjimmiltaan kahdesta osasta, numeerisesta arvosta sekä yksiköstä tai tunnuksesta, käyttölämpötilamerkintä koostuu kolmesta osasta, jotka ovat käyttölämpötilan tunnus "ta", lämpötilan tai lämpötila-alueen numeerinen arvo sekä lämpötilan johdannaisyksikkö °C. Entiteetti päätettiin merkata siis siten, että se alkoi aina käyttölämpötilan tunnuksesta "ta" ja loppui lämpötilan johdannaisyksikköön °C.

Lämpötilaentiteetti on lisäksi ainoa opetetuista entiteeteistä, jonka tunnistus olisi periaatteessa voitu suorittaa Language -palvelun valmiilla entiteettientunnistusalgoritmeilla, joka tunnistaa tekstistä lämpötilan. Valmista lämpötilantunnistusta ei kuitenkaan käytetty kahdesta syystä. Ensimmäinen syy oli se, että merkkauksäännöt pyrittiin tekemään yleispäteviksi. Tavoitteena oli, että samoja merkkauksääntöjä voitaisiin soveltaa riippumatta siitä, millä tavoin tai minkä palvelun avulla algoritmi opetetaan. Toinen syy oli sovelluksen tulevaisuuden soveltamisalueet. Otetaan esimerkki, jossa laboratoriollla on valmis sovellus, joka tunnistaa valaisimien arvokilpien informaatiota. Jos sovellusta halutaan laajentaa tunnistamaan

myös esimerkiksi liitäntälaitteiden arvokilpien informaatiota, on valmiin algoritmin päivitys kustannustehokkaampi vaihtoehto kuin kokonaan uuden algoritmin opettaminen. Liitäntälaitteissa voi olla käyttölämpötilamerkinnän lisäksi usein ilmoitettu myös laitteen kotelon maksimilämpötila, jonka tunnus on "tc". Jos lämpötilamerkinnän tunnusta, eli tässä tapauksessa merkintöjä "ta" ja "tc", ei oteta huomioon ja tunnistettavana entiteettinä on pelkästään lämpötila-arvo, olisi tunnistusalgoritmin tulosteessa kaksi tunnuksentonta lämpötila-arvoa. Kun ajatellaan sovelluksen käyttötarkoitusta, kahden tunnuksettoman lämpötilatiedon sisältävästä tulosteesta ei ole apua testauspäätösten teossa.

Entiteettien merkkiaussäätöjä luodessa on muutenkin hyvä pitää mielessä mahdolliset tulevaisuuden käyttöalueet. Eri sähkölaitteissa esiintyy hyvin paljon samoja merkintöjä ja eri laitestandardeissa olevat merkintävaatimukset ovat hyvin samankaltaisia. Jokaisen laitteen kohdalla ei kannata luoda aina uusia entiteettejä, vaan kannattaa hyödyntää jo valmiiksi määriteltyjä entiteettejä. Kun merkintäsäännöissä on huomioitu tulevaisuuden tarpeet, on samojen entiteettien soveltaminen eri laitteiden yhteydessä huomattavasti suoraviivaisempaa.

Datan merkkaamisen jälkeen, ennen mallin varsinaista opetusta, data jaettiin opetusdataan, testidataan ja myöhemmin käytettävään arviointidataan. Myöhemmin käytettävään arviointidataan valittiin manuaalisesti viisi eri arvokilpeä. Valinnassa kiinnitettiin huomiota siihen, että kaikki valitut arvokilvet olisivat mahdollisimman erilaisia ja että valituissa arvokilvissä esiintyisi jokainen entiteetti ainakin kerran.

Opetus- ja testidatan jaossa käytettiin yleisesti käytettyä suhdetta 80 / 20: 80 % datasta määriteltiin opetusdataksi ja 20 % testidataksi. Opetusdatalla tarkoitetaan sitä dataa, jolla mallin varsinainen opetus tehdään. Testidatan avulla luodaan mallin metriikat, eli arvioidaan mallin hyvyttä. Mallin testaus on osa opetusprosessia ja tapahtuu automaattisesti opetuksen yhteydessä. Language Studio teki jaon automaattisesti ja 80 / 20 opetusdatan ja testidatan suhde oli oletusasetuksena. Suhde olisi voitu määrittää myös itse, mutta tähän ei katsottu olevan tarvetta.

Lopullisessa opetus- ja testidatassa oli mukana 71:n eri arvokilven tekstimuotoinen data. Näissä esiintyivät opetettavat entiteetit seuraavasti:

- jännite-entiteetti 72 kertaa
- tehoentiteetti 61 kertaa
- taajuusentiteetti 69 kertaa
- IP-numero entiteetti 62 kertaa
- käyttölämpötilaentiteetti 32 kertaa.

4.4.3 Opetetun mallin arviointi

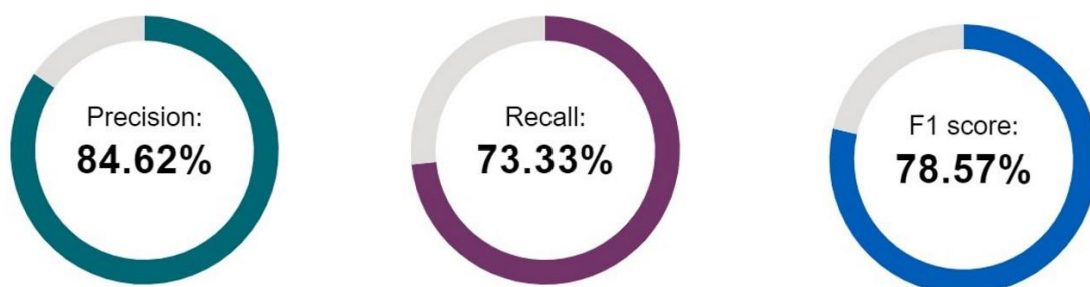
Language Studio -palvelu loi opetetusta mallista metriikat, joiden avulla opetetun mallin hyvyttä voitiin arvioida. Palvelusta nähtiin perusmetriikka koko mallin osalta sekä yksittäisten entiteettien osalta. Metriikan lisäksi opetusdatasta nähtiin perustietoja, kuten dokumenttien kokonaismäärä, miten dokumentit olivat jaettu opetus- ja testidataan sekä miten eri entiteetit olivat jakautuneet opetus- ja testidatan osalta.

Metriikoiden sekä opetus- ja testidatan perustietojen lisäksi, palvelu antoi myös ohjeistusta liittyen mallin parantamiseen. Ohjeistuksessa oli viisi kohtaa, joista nähtiin malliin ja opetusdataan liittyviä ongelmia sekä suosituksia. Mallin arviointi aloitettiin käymällä kyseinen ohjeistus läpi.

Listalla oli kaksi huomiota vaativaa ongelmaa. Ensimmäinen palvelun antama ongelmailmoitus oli, että entiteettien tyypit olivat liian lähellä toisiaan, eli malli sekoitti entiteetit toisiinsa. Tähän ei tietenkään pystytty vaikuttamaan, koska entiteetit määriteltiin standardin vaatimusten mukaan. Tätä ongelmaa pystyttiin tarkemmin tutkimaan sekaannusmatriisin avulla. Toinen palvelun antama huomio mallista oli, että entiteetit eivät olleet tasapainossa, eli joitain entiteettejä oli huomattavasti vähemmän kuin toisia. Lisätiedoista saatiin selvitettyä, että kyseessä oli käyttölämpötilaentiteetti, jota oli datassa noin puolet vähemmän, kuin muita entiteettejä. Tämä oli tiedossa jo dataa valmistellessa. Ongelma voitaisiin korjata lisäämällä opetusdataan lisää arvokilpiä, joissa on käyttölämpötilamerkintä.

Listassa oli ongelmien lisäksi kolme kohtaa, joissa kerrottiin, mitkä asiat olivat kunnossa. Tästä nähtiin, että opetusjoukossa oli tarpeeksi dataa, data sisälsi kaikki entiteettityypit ja entiteetit olivat jakautuneet tasaisesti opetus- ja testaus-dataan.

Seuraavaksi arvioitiin mallin sisäistä tarkkuutta (precision), herkkyyttä (recall) sekä F1-tulosta. Kuvassa 6 on esitetty koko mallia koskevat mittarit.



KUVA 6. Koko entiteettitunnistusmallia koskevat sisäinen tarkkuus, herkkyys sekä F1-tulos.

Mittareita arvioidessa tulee ottaa huomioon, mihin tarkoitukseen mallia käytetään ja miten väärät tunnistukset vaikuttavat haluttuun lopputulokseen. Tässä tapauksessa mallin haluttiin löytävän tietyt entiteetit sekä luokittelevan ne. Luokittelun perusteella haluttiin tehdä testaukseen liittyviä päätöksiä. Kun ajatellaan vääristä luokitteluista aiheutuvia ongelmia, ne voivat aiheuttaa sen, että testejä tehdään väärillä parametreilla. Kuvan 6 mukaiset koko mallia koskevat metriikat ovat usean luokan luokittelussa suuntaa antavia. Paremman kuvan mallin hyvyydestä saadaan, kun suorituskkyä tutkitaan entiteettitasolla.

Kuvassa 7 on esitetty tarkkuus, herkkyys ja F1-tulos jokaisen eri entiteetin kohdalla. Näiden metriikoiden avulla pystyttiin paremmin tulkitsemaa mallin hyvyyttä ja mallin tekemiä virheitä.

Entity name ↑	Precision %	Recall %	F1 score	Training labels	Testing labels
IP-number	91.67	91.67	0.917	50	12
frequency	66.67	66.67	0.667	54	15
power	100.00	33.33	0.5	49	12
ta	83.33	83.33	0.833	26	6
voltage	93.33	93.33	0.933	57	15

KUVA 7. Eri entiteettien metriikat.

Kuvasta 7 huomataan, että malli suoriutui melko hyvin IP-numeron ja jännitteen tunnistuksessa. Myös käyttölämpötilan tunnistus sujui melko hyvin. Taajuuden ja tehon tunnistuksessa mallilla taas oli hiukan ongelmia. Melkein kaikkien entiteettien osalta tarkkuus ja herkkyys olivat samat, mutta tuloksista voidaan kuitenkin nähdä yksi entiteetti, jonka tulokset poikkesivat muista. Tehoentiteetin osalta tarkkuus oli 100 %, mutta herkkyys vain 33,33 %. Tämä tarkoittaa käytännössä sitä, että kaikista mallin tunnistamista tehoentiteeteistä kaikki olivat oikeasti tehoentiteettejä, mutta mallilta jäi tunnistamatta noin kaksi kolmasosaa kaikista esiintyneistä tehoentiteeteistä. Toisin sanoen, kun mallilla haetaan arvokilvestä tehomerkinä, voidaan olla melko varmoja, että jos malli löytää tehomerkin, on tämä oikeasti tehomerkinä. Jos taas mallin tuloksen mukaan arvokilvessä ei ole tehomerkinä, on noin 66 % todennäköisyys, että arvokilvessä onkin tehomerkinä.

Parhaiten kyseistä mallia voitiin arvioida sekaannusmatriisin avulla. Sekaannusmatriisin avulla voitiin paremmin arvioida mallin tarkkuutta yksittäisten entiteettien kohdalla. Entiteettikohtaisten tarkkuuksien lisäksi, sekaannusmatriisista nähtiin hyvin, minkälaisia mallin tekemät virheet olivat. Kuvassa 8 on esitetty mallin sekaannusmatriisi.

Cell colors: 0.0 0.2 0.4 0.6 0.8 1

		Labeled as					
		\$none	IP-number	power	voltage	frequency	ta
Predicted as	\$none	-	0	0.583	0.025	0.019	0
	IP-number	0	1	0	0	0	0.056
	power	0	0	0.333	0	0	0
	voltage	0	0	0	0.975	0.019	0
	frequency	0.079	0	0.083	0	0.962	0
	ta	0	0	0	0	0	0.944

KUVA 8. Luodun entiteettitunnistusmallin sekaannusmatriisi.

Sekaannusmatriisista nähdään, että malli suoriutui hyvin IP-numeron, jännitteen, taajuuden sekä käyttölämpötilan tunnistuksessa. Edellä mainittujen entiteettien osalta vääriä luokituksia oli suhteellisen vähän. Kuten aikaisemmin todettiin, mallin arvioinnissa on hyvä pohtia, kuinka väärät tunnistukset vaikuttavat lopputulokseen. Sekaannusmatriisista nähdään, että malli ei sekoittanut IP-numeroa ja jännitemerkintää muihin entiteetteihin. Taajuusmerkintä sekoittui hyvin vähän jännitemerkintään, käyttölämpötilamerkintä hyvin vähän IP-numeron kanssa ja tehomerkinä hyvin vähän taajuusmerkinnän kanssa.

Kun ajatellaan lopullisen sovelluksen toimintaa, on erittäin tärkeää, että malli ei sekoita entiteettejä toisiinsa. IP-numeron arvolla ei voida tehdä testejä, jotka tehdään jännitearvon mukaan ja laitteen lämpötiloja ei voida arvioida taajuusarvon mukaan. Väärän luokittelun sijaan on parempi, että malli ei tee luokittelua ollenkaan. Tällöin sovelluksen käyttäjä voi itse täyttää puuttuvat tiedot. Luokittelemattomat instanssit näkyvät sekaannusmatriisissa luokkana "\$none". Kun mallia arvioitiin sekaannusmatriisin avulla, mallin hyvydestä saatiin huomattavasti positiivisempi kuva, kuin pelkästään sisäisen tarkkuuden, herkkyyden ja F1-tuloksen mittareilla.

Metriikoiden analysoinnin jälkeen mallia testattiin vielä arvokilvillä, joita ei ollut mukana opetus- ja testidatassa. Testauksessa keskityttiin opetusdatan valmistelun aikana huomattuihin mahdollisiin ongelmakohtiin, joita olivat:

- kaksi eri jännitemerkintää
- valmistuserätunnus, joka voi sekoittua tehomerkinä
- IP-numero on sisällytetty tuotteen mallimerkinä
- taajuusmerkinnän eri kirjoitusmuotojen tunnistus.

Mallia testattiin ensin hyvin yksinkertaisen arvokilven datalla, josta löytyi kaikki määritellyt entiteetit. Arvokilven dataa muokattiin tämän jälkeen siten, että kaikki tunnistettavat entiteetit olivat tekstitiedostossa peräkkäin. Yksinkertaisesta arvokilvestä malli tunnistoi kaikki entiteetit 100 % varmuudella. Kaikki entiteetit tunnistettiin myös silloin, kun kaikki entiteetit olivat kirjoitettu peräkkäin ja entiteetit olivat erotettu välilyönneillä. Kun entiteettejä ei ollut erotettu välilyönneillä, mallin luokittelukyky oli melko heikko. Kaikkia entiteettejä ei tunnistettu ja entiteetit sekoittuivat toisiinsa.

Yksinkertaisten arvokilpien jälkeen testattiin mahdollisten ongelmakohtien tunnistusta. Ensimmäiseksi testattiin kaksi jännitemerkintää sisältävän arvokilven tunnistusta. Testauksessa huomattiin, että esimerkiksi toisiojännitteen merkintä 12 V ja 12 VDC ei vaikuta tunnistukseen. Malli tunnisti ainoastaan syöttöjännitteen merkinnän.

Seuraavaksi testattiin, kuinka malli suoriutui jos arvokilvessä on valmistuserätunnus, esimerkiksi 15W22. Testauksessa huomattiin, että malli tulkitsee valmistuserätunnuksen kokonaisuudessaan tehomerkinäksi, kuten aiemmin oli arveltu. Malli kylläkin tunnisti myös oikean tehomerkin, joten tulosteessa oli kaksi eri tehomerkinettä: oikea tehomerkinettä esimerkiksi 50 W sekä väärin tunnistettu tehomerkinettä 15W22.

Tämän jälkeen testattiin, kuinka malli selviää IP-numeron tunnistuksesta, kun IP-numero on sisällytetty valaisimen mallimerkinettä. Huomattiin, että jos IP-numeroa ei ollut mitenkään erotettu mallimerkinettästä, eli mallimerkinettä ja IP-numero olivat samaa merkkijonoa, malli ei tunnistanut IP-numeroa. Jos taas IP-numero oli erotettu mallimerkinettästä esimerkiksi välilyönnillä tai yhdysmerkillä, malli tunnisti IP-numeron.

Viimeiseksi testattiin taajuusmerkinnän eri kirjoitusmuotojen vaikutusta taajuusmerkinnän tunnistukseen. Testattuja kirjoitusmuotoja olivat n-n Hz, n / n Hz ja n Hz / n Hz (n ollessa taajuuden numeerinen arvo). Testausta ei tehty kolmen eri arvokilven datalla, vaan yhden arvokilven dataa muokattiin siten, että saatiin kaikki eri kirjoitusmuotovariaatiot. Kirjoitusmuotojen n-n Hz ja n / n Hz osalta, malli palautti yhden tehomerkinettä, esimerkiksi 50–60 Hz ja 50 / 60 Hz. Muodon n Hz / n Hz tapauksessa, malli palautti kaksi eri taajuusmerkintää, esimerkiksi 50 Hz ja 60 Hz.

Opetetun mallin kaikki metriikat ovat esitetty liitteessä 2 sekä esimerkkejä tehtyjen testausten tuloksista liitteessä 3.

4.5 Kuvatunnusten tunnistaminen

Kirjoitetun tekstin tunnistamisen ja luokittelun lisäksi, sovellus määriteltiin tunnistamaan ja luokittelemaan osa valaisimien arvokilvissä esiintyvistä kuvatunnuksista.

Kuvatunnusten tunnistus ja luokittelu toteutettiin näkökategorian Custom Vision -palvelun avulla. Palvelun avulla käyttäjä voi luoda oman objektinluokittimen haluamilleen luokittelukategorioille. Sovellus määriteltiin tunnistamaan ja luokittelemaan suojamaadoituksen kuvatunnus, joka periaatteessa kertoo, että laitteen suojausluokka on luokkaa yksi, suojausluokan kaksi kuvatunnus sekä suojausluokan kolme kuvatunnus. Kyseiset kuvatunnukset valittiin, koska nämä näyttävät erittäin suurta roolia laitteiden turvallisuustestauksessa ja näiden mukaan tehdään useita tärkeitä testauspäätöksiä. Kuvatunnukset kertovat käytännössä valaisimien kosketeltavien osien ja verkkojännitteen välisen suojauksen tason.

Kuten entiteettien tunnistuksessa, tämänkin algoritmin opetusprosessiin sisältyi viisi vaihetta: tunnistettavien kuvatunnusten määrittely, datan valinta ja kerääminen, datan valmistelu, algoritmin opetus ja algoritmin arviointi. Datan valmistelu ja algoritmin opetus tehtiin Custom Vision -palvelussa, josta saatiin myös opetetun mallin metriikat.

Palvelu hyödyntää niin sanottua siirrettyä oppimista (transfer learning). Siirretty oppiminen tarkoittaa käytännössä sitä, että mallia ei tarvitse opettaa kokonaan uudelleen uutta luokittelutehtävää varten, vaan opetuksessa hyödynnetään jo olemassa olevaa mallia. Konvoluutioneuroverkot mahdollistavat siirretyn oppimisen, koska osaa toista luokittelutehtävää varten opetetusta konvoluutioneuroverkosta voidaan käyttää sellaisenaan uutta luokitustehtävää varten. (Salvaris ym. 2018.)

Tästä hyvin yksinkertaistettu esimerkki on se, että käytetään OCR-algoritmia hyväksi tässä opinnäytetyössä opetetun objektintunnistimen luomisessa. Kuten aikaisemmin mainittiin, konvoluutiokerrosten tarkoitus on etsiä kuvasta erilaisia ominaisuuksia, kuten vaakaviivoja, pystyviivoja ja kaaria, eli tässä tapauksessa ominaisuuksia, joista numerot ja kirjaimet koostuvat. Uusi malli taas määritellään

tunnistamaan objekteja, jotka koostuvat käytännössä samoista elementeistä, kuin numerot ja kirjaimet. Täten uuden mallin opetuksessa voidaan käyttää suoraan OCR-konvoluutioneuroverkon osia.

4.5.1 Opetusdata

Algoritmin opetukseen käytettiin samoja arvokilpikuvia, kuin tekstintunnistusosiossa. Opetuskuvia ei kuitenkaan ollut tarpeeksi jokaiselle luokalle. Tämän takia, opetuskuvien määrää lisättiin augmentoimalla.

Augmentointi tarkoittaa sitä, että yhdestä kuvasta tehdään useita eri versioita. Kuvaa voidaan esimerkiksi kiertää, siitä voidaan tehdä peilikuva, sen kontrastia voidaan muuttaa ja kuvaan voidaan lisätä kohinaa. Näin yhdestä kuvasta saadaan tehtyä erilaisia versioita, mikä lisää opetusdatan määrää. Augmentoidun opetusdatan avulla voidaan parantaa mallin suorituskykyä sekä vähentää mallin ylioppimista. Eri augmentointitapoja, varsinkin geometrisiä augmentointeja käytettäessä tulee varmistaa augmentointitavan soveltuvuus kyseessä olevaan tilanteeseen soveltuvaksi. (Shorten & Khoshgoftaar 2019, 1–7.)

Azuren dokumentaatiosta löytyi muutamia opetusdataa koskevia ohjeistuksia. Suositelluksi opetuskuvien vähimmäismääräksi ohjeissa mainittiin 30 kuvaa jokaista kategoriasta kohden. Jotta malli toimisi parhaalla mahdollisella tavalla, opetuskuvissa tulisi esiintyä vaihtelua kameran kuvakulmassa, kirkkaudessa, taustassa, visuaalisessa tyylissä, koossa ja tyypissä. Lisäksi olisi toivottavaa, että luokiteltava kohde esiintyisi kuvissa yksin sekä yhdessä ei-luokiteltavien kuvien kanssa. Opetuskuvilla tulee olla seuraavat ominaisuudet:

- Kuvan tiedostomuodon tulee olla JPEG, PNG, BMP tai GIF.
- Tiedoston enimmäiskoon tulee olla 6 MB (ennustettavien kuvien 4 MB).
- Kuvan pitkän sivun tulee olla vähintään 256 pikseliä. Palvelu suurentaa automaattisesti tätä pienemmät kuvat. (Quickstart: Build an object detector with the Custom Vision website 2022.)

Lopullisessa opetukseen käytetyssä datassa oli yhteensä 150 kuvaa, 50 kuvaa jokaista tunnistettavaa kuvatunnusta kohden.

4.5.2 Opetuksen valmistelu ja mallin opetus

Datan keräämisen jälkeen kuvia voitiin alkaa merkkamaan. Kuvien merkkaminen tehtiin Custom Vision -palvelussa. Kuvien merkkaminen oli sinänsä hyvin suoraviivainen prosessi. Kuviin piirrettiin tunnistettavien kuvatunnusten ympärille rajauslaatikot ja määritettiin rajauslaatikon sisälle jäävän kuvatunnuksen luokka. Kuvassa 9 on esimerkki kuvan merkkaamisesta ja kuvatunnuksen luokan määrittämisestä.



KUVA 9. Esimerkki kuvatunnusten tunnistamista varten tehdystä opetuskuvan merkkauksesta.

Kuvatunnusten merkkaamisessa ei ilmennyt suuria haasteita. Ainoa haaste oli, että joissain arvokilvissä eri kuvatunnukset ja tekstit olivat merkitty hyvin tiiviisti. Tämä hankaloitti hieman rajauslaatikoiden tekoa, koska rajauslaatikon sisään haluttiin ainoastaan tunnistettava kuvatunnus, eikä mitään muuta. Opetusdatan valmistelun yhteydessä myös todettiin, että kuvatunnusten merkkaminen kuviin oli erittäin aikaa vievää työtä.

Kuvatunnusten merkkaamisen jälkeen voitiin aloittaa mallin opetus. Custom Vision -palvelussa voitiin valita joko pikaopetus tai edistynyt opetus. Pikaopetus on mallin nopea opetus, jonka avulla saadaan tuntumaa mallin toimivuudesta. Edistynyt opetus taas kestää kauemmin ja mallista tulee parempi.

Mallin opetusparametreja ei pystytty palvelussa juurikaan säätämään. Ainoat parametrit, joita voitiin säätää, olivat ennustuksen todennäköisyyden raja-arvo sekä rajauslaatikoiden päällekkäisyyden raja-arvo. Nämä arvot jätettiin oletusasetuksiin, todennäköisyyden raja-arvoksi 50 % ja rajauslaatikoiden päällekkäisyyden raja-arvoksi 30 %. Todennäköisyyden raja-arvo tarkoittaa käytännössä sitä, että

jos malli ennustaa kuvatunnuksen kuuluvan tiettyyn luokkaan vähintään 50 % todennäköisyydellä, lasketaan tämä päteväksi luokitukseksi. Rajauslaatikoiden päällekkäisyyden raja-arvo taas liittyy kuvatunnuksen sijaintiin kuvassa. Jos mallin määrittelemä rajauslaatikko on vähintään 30 % opetusvaiheessa tehdyn rajauslaatikon päällä, katsotaan mallin onnistuneen kuvatunnuksen paikan määrittämisessä. Raja-arvoja käytettiin metriikoiden laskentaan.

4.5.3 Opetetun mallin arviointi

Opetuksen jälkeen mallia pystyttiin arvioimaan palvelun tarjoamilla metriikoilla. Palvelusta nähtiin koko mallin sisäinen tarkkuus, herkkyys sekä tarkkuuden keskiarvo (mAP). Koska kyseessä oli taas useamman luokan luokitus, koko mallia koskevasta metriikasta saatiin tuntumaa mallin suorituskykyyn.

Mallin herkkyys sekä tarkkuuden keskiarvo olivat molemmat 100 %. Tämä tarkoittaa käytännössä sitä, että malli löysi testidatasta kaikki merkatut kuvatunnukset sekä määrittä niiden paikat ennen opetusta määritellyllä minimitarkkuudella. Mallin sisäiseksi tarkkuudeksi saatiin 93,8 % eli hieman yli yhdeksän kymmenestä mallin tekemästä tunnistuksesta oli relevantteja tunnistuksia. Sisäistä tarkkuutta lähdettiin tutkimaan tarkemmin kategoriatason metriikoilla. Kuvassa 10 on esitetty sekä koko mallin metriikat, että kategoriatason metriikat.



Performance Per Tag

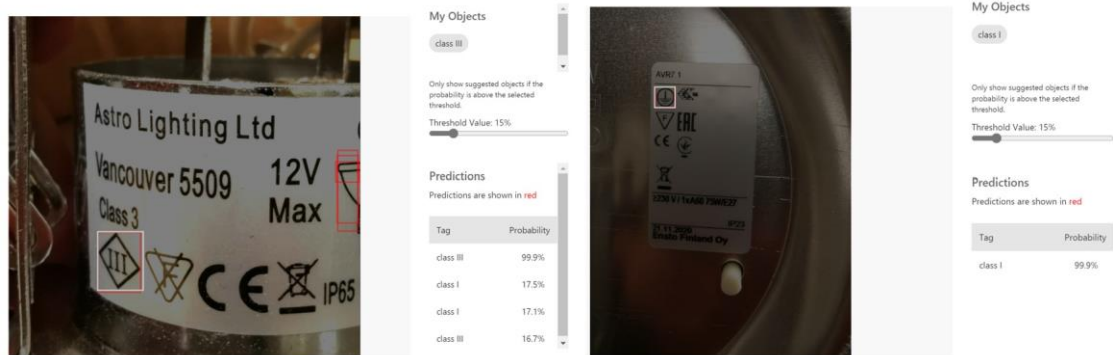
Tag	Precision	Recall	A.P.	Image count
class II	100.0%	100.0%	100.0%	50
class III	90.9%	100.0%	100.0%	50
class I	90.9%	100.0%	100.0%	50

KUVA 10. Luodun objektinluokittimen metriikat koko mallin osalta sekä kategoriatasolla.

Kuvan 10 kategoriatason metriikoista nähdään, että luokan II kuvatunnuksen osalta kaikki kolme mittaria näyttivät 100 %, joten tämän kategorian tuloksia ei analysoitu syvemmin. Analysointi keskittyi luokan I ja luokan III kuvatunnusten luokittelun tuloksiin, koska näissä tarkkuus oli 90,9 %.

Kategoriakohtainen analysointi aloitettiin luokasta I, eli suojamaadoituksen kuvatunnuksesta. Luokan tiedoista nähtiin, millä tavalla malli oli tunnistanut kuvatunnukset testidatan kuvista. Tässä kohtaa huomattiin myös, että opetusdatan ja testidatan jakosuhteena oli käytetty 80 / 20 jakoa, eli 80 % prosenttia kuvista käytettiin opetusdataan ja 20 % kuvista testidataan. Analysoinnin tulos oli hieman yllättävä. Malli oli luokitellut kaikki kuvissa esiintyneet kuvatunnukset oikein sekä löytänyt näiden paikat kuvista. Eli kategorian testidatan analysoinnin perusteella, tarkkuuden olisi pitänyt olla 100 %. Mistä sitten johtui, että tarkkuudeksi oli laskettu vain 90,9 %?

Tämä selvisi, kun analysoitiin luokan III testidatasta tehtyjä tunnistuksia. Huomattiin, että malli oli tehnyt yhden testidatan kuvan osalta vääriä positiivisia tunnistuksia. Kyseinen kuva oli augmentoitu kuva, joka oli rajattu alkuperäisestä koko arvokilven sisältävästä kuvasta. Tämän rajatun kuvan reunaan oli jäänyt osia kuvatunnuksesta, jota ei ollut määritelty tunnistettavan. Malli kuitenkin tulkitse, että tässä olisi ollut kaksi luokan I kuvatunnusta sekä yksi luokan III kuvatunnus. Tunnistukset olivat vääriä positiivisia tunnistuksia, joten ne vaikuttivat koko mallin sisäiseen tarkkuuteen sekä lisäksi luokan I ja luokan III kategorioiden tarkkuuteen. Kuvassa 11 on esitetty väärät tunnistukset sisältävä testidatan kuva sekä oikean tunnistuksen sisältävä testidatan kuva.



KUVA 11. Esimerkkejä mallin testidatan kuviin tekemistä tunnistuksista.

Mallin suorituskykyyn voitiin olla tyytyväisiä, vaikka sovelluksen käyttötarkoituksen kannalta on tärkeää, että malli ei tee vääriä positiivisia tunnistuksia. Kuvan 9 kaltaiset väärät tunnistukset saadaan todennäköisesti minimoitua tehokkaasti opetusdataa lisäämällä. Valmiissa sovelluksessa voitaisiin lisäksi kontrolloida, millä varmuudella tunnistus olisi tehtävä, jotta tunnistus vietäisiin päätöksentekovaiheeseen. Kuten kuvasta 11 huomataan, oikean positiivisen tunnistuksen varmuus oli 99,9 % ja väärin positiivisten tunnistusten varmuus noin 17 %.

Luodun mallin kaikki metriikat ovat esitetty liitteessä 4. Liitteessä 5 on esitetty lisää esimerkkejä mallin testidataan tekemistä tunnistuksista.

4.6 Kirjallisesta dokumentaatiosta tehtävät tunnistukset

Laitteeseen merkittävien tietojen lisäksi, laitestandardit vaativat tiettyjen tietojen löytymistä laitteen kirjallisesta dokumentaatiosta. Dokumentaatioon vaadittavat tiedot voivat olla esimerkiksi yksittäisiä arvoja, kuten jännite tai lämpötila, kuvatunnuksia tai kokonaisia lauseita. Valaisinstandardissa ohjeisiin vaadittavia tietoja ovat muun muassa valonlähteen vaihtoa koskeva kirjallinen ohjeistus, liitäntäjohdon vaihtoa koskeva kirjallinen ohjeistus sekä valaisimien asennukseen liittyviä ohjeistuksia.

Kirjallista dokumentaatiota analysoivaa mallia ei tehty tämän opinnäytetyön yhteydessä. Suurin syy tähän oli se, että kirjallisen dokumentaation analysointiin voitaisiin käyttää samoja tekniikoita, kuin arvokilpien analysointiin. Dokumentaatioon vaadittavan kirjallisen ohjeistuksen tunnistus voitaisiin toteuttaa mukautetulla entiteettitunnistusmallilla, koska esimerkiksi Azuressa yhden entiteetin suurin sallittu merkkimäärä on 500 merkkiä (Custom named entity recognition service limits 2022). Dokumentaatioon vaadittavat kuvatunnukset taas voitaisiin tunnistaa saman kaltaisella objektinluokittimella, joka luotiin arvokilpiä varten. Toinen syy oli se, että opinnäytetyötä varten ei saatu kerättyä tarpeeksi opetusdataa.

Kirjallisen dokumentaation analysoinnin suurin haaste on sanallisten ohjeistuksien kirjoitustavoissa esiintyvät variaatiot. Standardit eivät nimittäin vaadi kaikkien vaatimusten osalta käytettävän tiettyä kirjoitusasua. Otetaan esimerkiksi valaisinstandardin vaatima kirjallinen ohjeistus liittyen valonlähteen vaihtoon. Jos valaisimen rakenne on sen kaltainen, että valonlähteeseen ei pääse käsiksi rikkomatta valaisinta tai sen osia, tulee dokumentaatioissa olla ei-vaihdettavaan valonlähteeseen liittyvä ohjeistus. Ohjeistus on kaksiosainen: ensimmäisessä osassa todetaan, että valaisimen valonlähde ei voi vaihtaa ja toisessa osassa ohjeistetaan, että koko valaisin on vaihdettava uuteen valonlähteen rikkoutuessa. Standardi antaa tähän esimerkkilauseen, joka kuuluu kutakuinkin: ”Tämän valaisimen valonlähde ei voi vaihtaa. Kun valonlähde tulee käyttöikänsä päähän, koko valaisin on vaihdettava.” Standardi ei kuitenkaan vaadi käytettäväksi tiettyä sanamuotoa.

Kyseisestä ohjeistuksesta voi esiintyä monia eri variaatioita. Jos katsotaan pelkästään ohjeistuksen ensimmäistä osaa, voi kirjoitusasu olla esimerkiksi: ”valaisimen valonlähde ei ole vaihdettavissa”, ”tämän valaisimen LED-moduulia ei voida vaihtaa” tai ”LED-valonlähde ei ole vaihdettavissa”. Kaikki esimerkit ovat melko erilaisia, mutta tarkoittavat kaikki samaa ja täyttävät standardin vaatimukset. Entiteettien tunnistuksen näkökulmasta tämä tarkoittaa, että tietty entiteetti voi esiintyä useassa eri muodossa. Tällöin tarvitaan hyvin paljon opetusdataa, joka sisältää tarpeeksi monta esimerkkiä kaikista erilaisista kirjoitusvariaatioista.

Tunnistusta voi tuki lähestyä myös toisesta näkökulmasta. Sen sijaan, että luotaisiin entiteetti, joka on kokonainen lause, voitaisiin kirjallisesta dokumentaatiosta etsiä tiettyjä avainsanoja. Avainsanojen avulla voitaisiin eristää sanat sisältävät lauseet ja näin löytää vaaditut ohjeistukset. Jos käytetään esimerkkinä taas valonlähteen vaihtoa koskevaa ohjeistusta, voitaisiin avainsanoiksi valita ”valonlähde”, ”LED”, LED-moduuli” ja ”LED-valonlähde”. Mallin löytäessä avainsanat dokumentaatiosta, eristettäisiin kaikki avainsanat sisältävät lauseet tai tietty määrä avainsanan edeltäviä ja jälkeen tulevia sanoja. Eristetyt lauseet näytettäisiin sovelluksen käyttäjälle, joka voisi tehdä lopullisen päätöksen siitä, täyttääkö jokin lauseista standardin asettavat vaatimukset. Tämän toteutustavan avulla ei saataisi yhtä automatisoitua sovellusta kuin entiteettien tunnistuksella, mutta opetusdataa tarvittaisiin huomattavasti vähemmän, koska avainsanojen variaatioita olisi huomattavasti vähemmän, kuin kokonaisten lauseiden variaatioita.

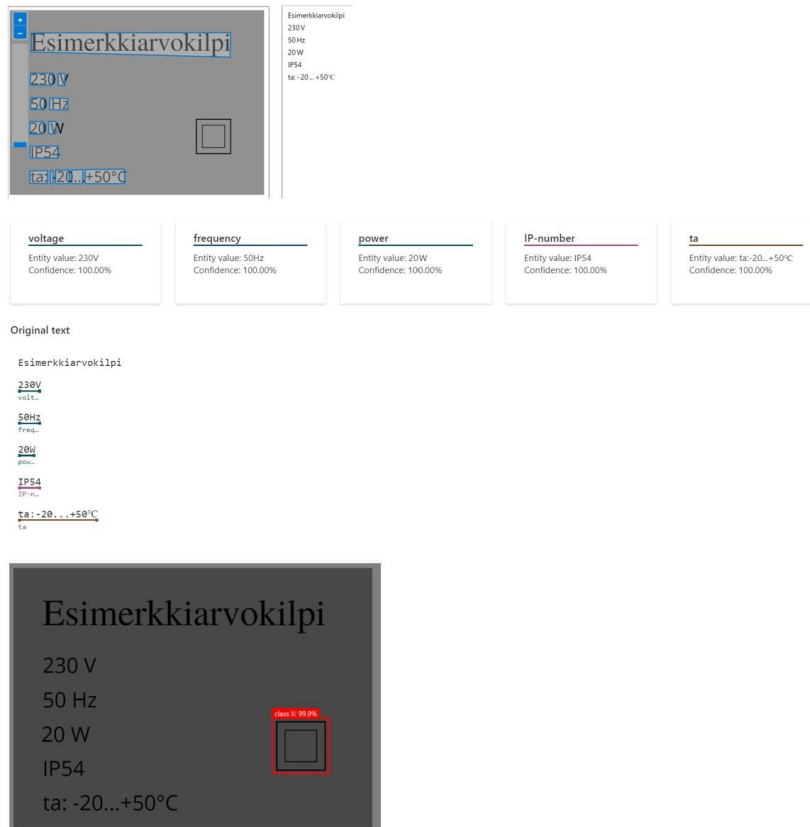
Toinen tapa voisi olla yhdistää entiteettien tunnistus sekä avainsanojen tunnistus. Malli voisi toimia käytännössä niin, että ensin tehdään avainsanojen tunnistus ja avainsanojen sisältävien lauseiden eristys. Sen sijaan, että entiteetin tunnistus tehtäisiin koko kirjallisesta dokumentaatiosta, tunnistus tehtäisiin vain eristetyille lauseille. Tällöin päätöksenteon näkökulmasta voitaisiin käyttää hyödyksi mallin tekemän entiteetin tunnistuksen varmuutta: tunnistus, jolla olisi suurin varmuus jatkaisi lopulliseen päätöksentekoon.

4.7 Tunnistuksista tehtävät päätökset

Opinnäytetyössä oli käytettävien tekniikoiden määrittelyn lisäksi tarkoitus pohtia, miten tekoälyalgoritmien tuloksia voitaisiin hyödyntää testaus- ja tarkastusprosessissa. Sovelluksen perimmäinen tarkoitus olisi analysoida arvokilpiä ja teknistä dokumentaatiota tekoälymallien avulla ja näiden tietojen perusteella auttaa testausinsinööriä tekemään testaukseen liittyviä päätöksiä. Analysoinnin tuloksena saatujen tietojen avulla voitaisiin rakentaa valmista raakadatapohjaa, täyttää testausraporttia sekä määrittää täyttävätkö arvokilvissä ja dokumentaatiossa esiintyvät tiedot standardin vaatimukset.

Sovellus voisi toimia esimerkiksi kuvan 12 mukaisesti. Arvokilpi syötettäisiin ensin OCR-malliin, joka tunnistaisi arvokilvestä kirjoitetun tekstin. OCR-algoritmin tulos lähetettäisiin seuraavaksi entiteettitunnistusmallille, joka tunnistaisi tekstistä määritellyt entiteetit. Arvokilpi lähetettäisiin samanaikaisesti objektitunnistusmallille, joka tunnistaisi arvokilvestä määritellyt kuvatunnukset. Kuten kuvasta 12 huomataan, arvokilvestä saadaan tekoälymallien avulla eristettyä seuraavat tiedot:

- Valaisimen syöttöjännite on 230 V.
- Valaisimen syöttötaajuus on 50 Hz.
- Valaisimen teho on 20 W.
- Valaisimen IP-luokka on IP54.
- Valaisimen käyttölämpötila-alue on -20°C...+50°C.
- Valaisimen suojausluokka on luokka II.



KUVA 12. Esimerkkiarvokilvestä saatavat tiedot käyttäen OCR-algoritmia, entiteettien tunnistusta sekä objektien tunnistusta.

Mitä kyseisillä tiedoilla voidaan sitten tehdä. Ensinnäkin tietojen avulla voidaan arvioida täyttävätkö valaisimen arvokilven merkinnät standardin vaatimukset, eli löytyvätkö arvokilvestä kaikki kyseiselle valaisintyypille vaaditut merkinnät. Tätä ei voida tehdä suoraan tekoälymallien tulosten avulla, koska kaikkiin valaisimiin ei vaadita kaikkia standardissa mainittuja merkintöjä. Jotta tulosten avulla voitaisiin arvioida merkintöjen standardinmukaisuutta, tulisi lopullisessa sovelluksessa olla jonkinlainen aloitussivu, johon testaaja täyttää kyseisen valaisimen perustiedot.

Vaihtoehtoinen toteutus arvokilven merkintöjen standardinmukaisuuden arvioimiseen olisi se, että standardin kaikista merkintävaatimuksista luotaisiin esimerkiksi taulukko, jonka toisessa sarakkeessa on kaikki standardissa mainitut merkintävaatimukset ja toisessa kaikki arvokilvestä tunnistetut merkinnät. Testausinsinööri voisi täten näitä kahta saraketta vertailemalla analysoida arvokilven merkintöjen standardinmukaisuutta.

Kirjallisen dokumentaation analysoinnissa paras tapa olisi ensin määrittää, mitä tietoja sovelluksen tulisi dokumentaatiosta etsiä. Sovelluksen ensisijainen hyöty dokumentaation analysoinnissa olisi, että monisivuisista dokumenteista ei tarvitsisi manuaalisesti etsiä yhtä tiettyä lausetta, joka voi olla käytännössä missä kohdassa hyvänsä dokumenttia. Tämä säästäisi paljon aikaa niin testaajalta, kuin testausraportin tarkastajalta.

Standardinmukaisuuden arvioimisen lisäksi tuloksia voitaisiin käyttää testaus suunnitelman tekoon. Arvokilvestä ja kirjallisesta dokumentaatiosta saatujen tietojen avulla voitaisiin määrittää, mitä testejä tuotteelle tulisi tehdä ja mitä arvoja testeissä tulisi käyttää. Automaattisesti luodun testaus suunnitelman avulla saataisiin kuva testausprosessin laajuudesta ja voitaisiin tehdä esimerkiksi testauksen tarjouksia.

Luotua testaus suunnitelmaa pystyttäisiin myös käyttämään raakadatan pohjana. Raakadatalta tarkoitetaan testauksessa sitä dataa, jota tuotetaan testauksen yhteydessä. Raakadata voi olla yksittäisten testien tuloksia, testaajan kirjaamia huomioita tai yksinkertaisia hyväksytyt tai hylätyt kommentteja. Laadukkaan raakadatan luominen on yksi testauksen tärkeimmistä asioista. Raakadatan perimmäinen tarkoitus on se, että testaus voidaan toistaa samanlaisena pelkästään alkuperäisen testauksen raakadatan perusteella. Virallisiin testausraportteihin päätyy vain murto-osa testauksen aikana kertyneestä datasta, joten testausraporttien pohjalta on vaikeaa toistaa testausta tai analysoida testausmenetelmiä.

Valmis, automaattisesti tuotettu raakadatapohja auttaisi testaajaa testausprosessin aikana. Raakadatapohja toimisi käytännössä testaajan muistilistana ja testausohjeena. Testaajalta säästyisi testausprosessissa aikaa, kun raakadataan olisi valmiiksi kirjattu tehtävät testit, testijärjestys sekä testauksessa käytettävät arvot. Valmiiden raakadatapohjien avulla saataisiin myös yhtenäistettyä laboratorioissa tuotettavan raakadatan ulkomuotoa. Tästä on apua, kun raakadatatista joudutaan myöhemmin etsimään yksittäisten testien tuloksia.

Kolmanneksi arvokilvestä ja kirjallisesta dokumentaatiosta saatuja tietoja voitaisiin käyttää virallisten testausraporttipohjien automaattiseen täyttöön. Viralliset testausraporttipohjat ovat word-dokumentteja ja pohjat koostuvat suurelta osin

nelisarakkeisista taulukoista. Yksi rivi käsittää tietyn standardikohdan yhden vaatimuksen. Taulukossa 2 on esitetty esimerkki testausraporttipohjan rakenteesta.

Taulukko 2. Esimerkki virallisen testausraporttipohjan rakenteesta.

Kohta	Vaatus + testi	Tulos / Huomio	Päätös
4.1	Valaisimen IP-luokka jos muu kuin IP20...:	IP54	P
4.2	Roiskevedeltä suojatun valaisimen sisällä ei vettä		P
7	Valaisimen suojamaadoitus	luokka II	N/A

Ensimmäinen sarakeista kertoo standardin kohdan, josta vaatimus löytyy ja toinen sarakeista kertoo tiivistetysti standardin kohdan vaatimuksen. Näihin kahteen sarakeeseen ei raportin täyttäjällä saa tehdä minkäänlaisia muutoksia. Kolmas sarake on tarkoitettu mittaustuloksille ja raportin täyttäjän huomioille. Tähän sarakeeseen ei pääsääntöisesti tarvitse kirjoittaa mitään, poikkeuksena taulukon 2 esimerkin ensimmäisen vaatimuksen kaltainen tilanne, jossa vaatimuksen perässä on pistejono ja kaksoispiste. Tällöin kommenttikentän täyttäminen on pakollista. Vaikka kommentointi ei ole joka kohdassa pakollista, on se kuitenkin erittäin suositeltavaa.

Viimeiseen sarakeeseen merkitään päätös standardin mukaisuudesta. Merkintä P (Pass) tarkoittaa, että tuote täyttää kyseisen kohdan vaatimukset ja N/A (not applicable) tarkoittaa, että kyseinen standardinkohta ei koske raportoinnin kohteena olevaa tuotetta. Myös merkintä F (fail) on olemassa, mutta tätä ei käytetä virallisissa testausraporteissa.

Arvokilvistä ja teknisestä dokumentaatiosta saatujen tietojen avulla voitaisiin siis automaattisesti täyttää tietyiltä osin testausraportin huomio- ja päätössarakkeita. Automaattista täyttöä voitaisiin hyödyntää esimerkiksi testausraportin kohtiin, jotka ottavat kantaa merkintöihin, kuten taulukon 2 ensimmäisessä standardinkohdassa. Kohdassa pyydetään täyttämään valaisimen IP-luokka raporttiin, joten kommenttikenttään voidaan automaattisesti täyttää arvokilven analysoinnin tuloksena saatu IP54-merkintä sekä päätöskohtaan kirjata P.

Edellä mainittu esimerkki on tilanne, jossa standardi vaatii, että tietyn merkinnän tai ohjeistuksen tulee löytyä valaisimen arvokilvestä tai teknisestä dokumentaatiosta. Analysoinnin perusteella todetaan, että kyseinen merkintä löytyy ja täytetään raporttiin asiaankuuluvat kohdat. Todellisissa raporteissa on päätössarakkeissa kuitenkin yleensä N/A-merkintöjä yhtä paljon kuin P-merkintöjä. Analysoinnin tuloksena saatuja tietoja voitaisiin käyttää myös N/A-merkintöjen automaattiseen täyttöön. Tällöin päätöksenteko olisi niin sanotusti käänteinen, eli arvokilven ja dokumentaation analysoinnin perusteella saatujen tietojen avulla suljettaisiin pois tiettyjä standardinkohtia.

Esimerkkinä tästä on taulukon 2 viimeinen rivi, jossa käsitellään valaisimen suojamaadoitusta. Koska arvokilven analysoinnin tuloksena on saatu tieto, että valaisimen suojausluokka on luokka II, eli valaisimessa ei ole suojamaadoitusta, voidaan kaikki suojamaadoitusta käsittelevät raportin kohdat merkitä automaattisesti N/A-merkinnällä. Tämä säästäisi huomattavasti raportin täyttämiseen kuluva aikaa verrattuna siihen, että N/A-merkintöjä lisättäisiin raporttiin manuaalisesti.

4.8 Sovelluksen tulevaisuuden kehitysmahdollisuudet

Sovelluksen kehityksessä ensimmäinen askel olisi luonnollisesti täydentää tekoälymalleja siten, että ne tunnistaisivat arvokilvistä ja teknisestä dokumentaatiosta kaikki valaisinstandardissa vaadittavat tiedot. Tähän tullaan tarvitsemaan suuri määrä dataa ja työtunteja, varsinkin entiteettien tunnistuksen osalta.

Seuraava luonnollinen askel olisi laajentaa sovellusta siten, että sitä voitaisiin käyttää myös muiden laitteiden testauksessa. Tässä voitaisiin hyödyntää jo olemassa olevia tekoälymalleja ja esimerkiksi luoda perusmalli, joka tunnistaisi yleisimmät sähkölaitestandardissa vaaditut merkinnät. Uusille laitteille soveltuvia tekoälymalleja voitaisiin täten luoda laajentamalla perusmallia.

Vaikka tässä opinnäytetyössä keskityttiin ainoastaan arvokilpien ja dokumentaation analysointiin, voitaisiin varsinkin konenäköä hyödyntää myös muissa testauksen osa-alueissa. Konenäön avulla voitaisiin esimerkiksi analysoida tuotteiden rakenteellisia vaatimuksia, esimerkiksi tunnistaa piirilevyiltä eri piirit. Konenäön hyödyntämistä laitteiden rakenteen tutkintaan tulisi kuitenkin pohtia tarkkaan: mitä osa-alueita pystytään riittävällä tarkkuudella tutkimaan ja minkä osa-alueiden tutkiminen konenäöllä on ylipäätään järkevää.

Sovelluksen päätöksenteon kannalta mielenkiintoinen kehitteillä oleva osa-alue ovat älykkäät standardit, joita kutsutaan myös koneluettaviksi standardeiksi tai SMART-standardeiksi (Standards Machine Applicable, Readable and Transferable). Älykkäät standardit voisivat käytännössä mahdollistaa sen, että tekoälymallien tulokset voisivat keskustella suoraan itse standardin kanssa. Tämä helpottaisiin huomattavasti lopullisen sovelluksen rakentamista, verrattuna tilanteeseen, jossa standardit ovat saatavilla vain PDF-tiedostoina. Standardien ollessa saatavilla vain niin sanotusti ihmisen tulkittavissa olevassa muodossa, joudutaan sovellusta rakentaessa siirtämään kaikki standardin vaatimukset manuaalisesti sovellukseen sekä määrittämään eri standardinkohtien suhteet toisiinsa. Vaatimusten manuaalinen siirtäminen vaatii huomattavan määrän resursseja.

5 POHDINTA

Opinnäytetyön tarkoituksena oli tutkia, kuinka tekoälyä voidaan hyödyntää valaisimien arvokilpien ja muun teknisen dokumentaation analysoinnissa ja miten saatuja tietoja voidaan käyttää hyväksi valaisimien turvallisuustestauksessa. Tavoitteena oli luoda tekoälymallit, joiden avulla voidaan tunnistaa ja luokitella kirjoitettua tekstiä sekä kuvatunnuksia.

Opinnäytetyössä esitettyjen tulosten perusteella voidaan huomata, että arvokilpiä ja teknistä dokumentaatiota pystytään analysoimaan tekoälyalgoritmien avulla. Tuloksista voidaan huomata myös, että analysointiin ei välttämättä tarvitse rakentaa täysin uusia tekoälymalleja, vaan valmiiden tekoälypalvelujen avulla saadaan toteutettua toimivia sovelluksia.

Tavoite luoda toimivat tekoälymallit saavutettiin osittain. Tutkimuksen aikana saatiin luotua mallit, jotka tunnistavat arvokilvistä yksinkertaisia entiteettejä, kuten jännitteen ja taajuuden sekä kolme erilaista kuvatunnustusta. Alkuperäisenä tavoitteena oli kuitenkin luoda malli, joka pystyisi analysoimaan myös kirjallista dokumentaatiota ja dokumentaatiossa esiintyviä monimutkaisempia entiteettejä. Kyseistä mallia ei lopulta pystytty luomaan opetusdatan vähyyden vuoksi. Tutkimuksen yhteydessä kuitenkin todettiin, että samoilla tekniikoilla on mahdollista luoda malli, joka tunnistaa monimutkaisempia entiteettejä, esimerkiksi kokonaisia lauseita. Lisäksi samoja tekniikoita voidaan käyttää myös muiden sähkölaitteiden tietojen analysointiin.

Kun ajatellaan opinnäytetyössä käytettyjä tekniikoita, eli optista merkkien tunnistusta, objektien tunnistusta sekä entiteettien tunnistusta, on entiteettien tunnistus näistä haastavin osa-alue. Tämä johtuu siitä, että varsinkin kirjallisessa dokumentaatiossa esiintyvä sanasto voi olla hyvinkin monimuotoista sekä alakohdista, eli sanastoa ei esiinny juurikaan muussa yhteydessä. Tämä tarkoittaa käytännössä sitä, että entiteettientunnistusmallin opetukseen tarvittava datamäärä on erittäin suuri ja malli tulee opettaa hyvin pitkälti ohjatun oppimisen menetelmin. Näin ollen tuleekin pohtia, onko mallin luomiseen vaadittava ajallinen panostus, eli datan keräämiseen ja merkkaamiseen kuluva aika kannattavaa. Kysymystä

tulisi kirjoittajan mielestä pohtia siltä kannalta, kuinka monimutkaisten laitteiden tietoja mallilla on tarkoitus analysoida. Mitä monimutkaisempi laite on, sitä kannattavampaa olisi rakentaa automatiikkaa laitteiden tietojen analysointiin, kun taas yksinkertaisten laitteiden osalta, automatiikan rakennus ei välttämättä ole kannattavaa.

Arvokilpien ja teknisen dokumentaation analysointia voidaan lähestyä myös toisesta näkökulmasta. Sen sijaan, että luotaisiin malleja, jotka tunnistaisivat kaikki erilaiset kirjoitus- ja muotoiluvariaatiot, joita arvokilvissä ja teknisessä dokumentaatioissa esiintyy, voitaisiin asiaa lähestyä standardisoinnin näkökulmasta. Standardeissa voisi esimerkiksi määrittää tietyn ohjeistuksen tarkan kirjoitusasun, jolloin entiteettien tunnistus olisi helpompaa. Tai jos asiaa viedään vielä pidemmälle, laitteiden ohjeet voisivat olla lomakemuodossa, jossa yhdessä kentässä olisi aina tietty ohjeistus. Tällöin analysointiin voitaisiin käyttää yksinkertaisempaa tekniikkaa kuin entiteettien tunnistus.

Optisen merkkien tunnistuksen ja objektien tunnistuksen osalta ei havaittu samankaltaisia haasteita, kuin entiteettien tunnistuksen osalta. Valmiita OCR-algoritmeja on kehitetty erittäin paljon ja nämä valmiit ratkaisut ovat erittäin tehokkaita tunnistamaan tekstiä kuvista. Kuvatunnusten tunnistusta taas helpottaa se, että kaikki standardeissa vaaditut kuvatunnukset ovat määritelty standardeissa, joten näiden kohdalla ei esiinny suurta variaatiota.

Opinnäytetyössä tutkittiin melko pintapuolisesti, miten arvokilvistä ja teknisestä dokumentaatiosta saatuja tietoja voidaan käytännössä käyttää testausprosessin apuna. Tietojen käytön yksityiskohtainen määrittely vaatisi huomattavasti enemmän aikaa, kuin mitä tämän tutkimuksen tekoon oli varattu. Jos tutkimuksen pohjalta halutaan rakentaa toimiva sovellus, joka analysoi valaisimien tietoja ja tekee näiden tietojen pohjalta testauspäätöksiä, tulisi tietojen yhteys eri standardinkohtiin määrittää hyvinkin tarkasti. Esimerkiksi laitteen syöttöjännitemerkintä vaikuttaa usean testin osalta siihen, millä jännitteellä testi tehdään ja mitä raja-arvoja vaatimuksenmukaisuuden arvioinnissa tulee soveltaa. Jännitemerkinnän avulla voidaan myös tehdä tiettyjä oletuksia laitteen rakenteesta. Myös tutkimuksessa esitetyt kuvatunnukset vaikuttavat hyvin laajasti standardien vaatimukseen.

Näiden avulla voidaan määrittää rakenteellisia vaatimuksia, testeissä käytettäviä arvoja sekä sovellettavia raja-arvoja yhdessä jännitemerkintöjen kanssa.

Kuten tutkimuksessa todettiin, on toimivan ja laboratorio-olosuhteisiin kelpaavan sovelluksen rakentamisessa ja ylläpidossa tiettyjä haasteita. Rakennusvaiheen yksi suurimmista haasteista on edellä kuvattu standardien vaatimusten määrittely sovellukseen. Määrittely on ainakin tällä hetkellä tehtävä manuaalisesti, mikä on erittäin aikaa vievää. Ylläpidon haasteet liittyvät myös manuaaliseen vaatimusten määrittelyyn. Standardit päivittyvät tietyin väliajoin, jolloin myös standardien vaatimukset tulee päivittää sovellukseen. Varsinkin suurten standardipäivitysten yhteydessä, on erittäin haastavaa määrittellä, mihin kaikkiin standardin kohtiin päivityksessä esitetyt muutokset oikeasti vaikuttavat. Päivitykset tulee myös joka kerta validoida, jotta varmistutaan siitä, että sovellus toimii oikein ja luotettavasti. Tulevaisuudessa älykkäät standardit voivat helpottaa vaatimusten määrittelyä ja täten sovelluksen rakentamista ja päivitystä.

Tutkimuksen perusteella kirjoittajan näkemys tekoälyn hyödyntämisestä valaisimien tai muiden sähkölaitteiden teknisten tietojen analysoinnissa on se, että tekoälyä voidaan kyllä hyödyntää, mutta laboratorioiden tulisi pohtia tarkasti sen tuomia hyötyjä ja haittoja. Tekoälyä hyödyntävän sovelluksen rakentaminen voi olla perusteltua monimutkaisten laitteiden teknisen dokumentaation analysointiin. Näiden laitteiden tekniset tiedostot ovat usein monisivuisia ja standardit asettavat teknisille tiedoille monia eri vaatimuksia. Tällöin tekoälyä hyödyntävän sovelluksen avulla voidaan saada aikaan huomattavaa ajansäästöä. Kun taas kyseessä on yksinkertainen laite, jonka teknisille tiedoille ei ole suurta määrää vaatimuksia ja tiedostot ovat vain muutaman sivun mittaisia, voi kokonaisuudessaan olla järkevämpää pysyä manuaalisessa analysoinnissa.

LÄHTEET

Anwar, A. n.d. What is Average Precision in Object Detection & Localization Algorithms and how to calculate it? A step-by-step visual guide to understanding the mean average precision for object detection and localization algorithms. Towards Data Science. Viitattu 21.11.2022. <https://towardsdatascience.com/what-is-average-precision-in-object-detection-localization-algorithms-and-how-to-calculate-it-3f330efe697b>

Azure security baseline for Cognitive Services. 2022. Microsoft Learn. Verkkosivu. Viitattu 13.10.2022. <https://learn.microsoft.com/en-us/security/benchmark/azure/baselines/cognitive-services-security-baseline>

Call the Computer Vision 3.2 GA Read API. 2022. Microsoft Learn. Verkkosivu. Viitattu 13.11.2022. <https://learn.microsoft.com/en-us/azure/cognitive-services/computer-vision/how-to/call-read-api?source=recommendations>

Custom named entity recognition service limits. 2022. Microsoft Learn. Verkkosivu. Viitattu 13.11.2022. <https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/custom-named-entity-recognition/service-limits>

Evaluation metrics for custom named entity recognition models. 2022. Microsoft Learn. Verkkosivu. Viitattu 22.11.2022. <https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/custom-named-entity-recognition/concepts/evaluation-metrics>

Grigoriev, A. 2021. Machine Learning Bootcamp. E-kirja. Shelter Island, NY: Manning Publications Co. Viitattu 21.11.2022. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/machine-learning-book-camp/9781617296819/>

How to prepare data and define a schema for custom NER. 2022. Microsoft Learn. Verkkosivu. Viitattu 20.10.2022. <https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/custom-named-entity-recognition/how-to/design-schema>

Label your data in Language Studio. 2022. Microsoft Learn. Verkkosivu. Viitattu 20.10.2022. <https://learn.microsoft.com/en-us/azure/cognitive-services/language-service/custom-named-entity-recognition/how-to/tag-data>

Li, J., Sun, A., Han, J. & Li, C. 2022. A Survey on Deep Learning for Named Entity Recognition. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 34, NO. 1, 50–70. Viitattu 18.11.2022. Vaatii käyttöoikeuden. <https://ieeexplore-ieee-org.libproxy.tuni.fi/stamp/stamp.jsp?tp=&arnumber=9039685&tag=1>

Moniz, A., Gordon, M., Bergum, I., Chang, M. & Grant, G. 2021. Beginning Azure Cognitive Services. Data-Driven Decision Making Through Artificial Intelligence. 1st ed. E-kirja. New York: Apress. Viitattu 12.10.2022. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/beginning-azure-cognitive/9781484271766/>

Nadeau, D. & Sekine, S. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes: International Journal of Linguistics and Language Resources* Vol. 30, 3–26.

Peters, J. 2017. *Foundations of Computer Vision. Computational Geometry, Visual Image Structure and Object Shape Detection*. Intelligent Systems Reference Library 124. Cham: Springer International Publishing AG.

Plan and manage costs of an Azure Cognitive Search service. 2022. Microsoft Learn. Verkkosivu. Viitattu 13.10.2022. <https://learn.microsoft.com/en-us/azure/search/search-sku-manage-costs>

Pokhrel, S. 2019. *Beginners Guide to Convolutional Neural Networks: Learn about the important components that make up a Convolutional Neural Network*. Towards Data Science 19.9.2019. Viitattu 16.11.2022. Vaatii käyttöoikeudet. <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d>

Pramoditha, R. 2022. *Convolutional Neural Network (CNN) Architecture Explained in Plain English Using Simple Diagrams*. Neural Networks and Deep Learning Course: Part 23. Towards Data Science 20.6.2022. Viitattu 15.11.2022. Vaatii käyttöoikeuden. <https://towardsdatascience.com/convolutional-neural-network-cnn-architecture-explained-in-plain-english-using-simple-diagrams-e5de17eacc8f>

Quickstart: Build an object detector with the Custom Vision website. 2022. Microsoft Learn. Verkkosivu. Viitattu 27.10.2022. <https://learn.microsoft.com/en-us/azure/cognitive-services/custom-vision-service/get-started-build-detector>

Salvaris, M., Dean, D. & Tok, W. H. 2018. *Deep Learning with Azure: Building and Deploying Artificial Intelligence Solutions on the Microsoft AI Platform*. E-kirja. Kalifornia: Apress Berkley. Viitattu 16.11.2022. Vaatii käyttöoikeuden. <https://learning.oreilly.com/library/view/deep-learning-with/9781484236796/>

SESKO RY. n.d. SFS-/IEC-/EN-STANDARDIT. Verkkosivu. Viitattu 9.10.2022. <https://sesko.fi/standardit/sfs-iec-en-standardit>

Shafkat, I. 2022. *Intuitively Understanding Convolutions for Deep Learning: Exploring the strong visual hierarchies that makes them work*. Towards Data Science 1.6.2018. Viitattu 15.11.2022. <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>

Shorten, C. & Khoshgoftaar, T.M. 2019. A survey on Image Data Augmentation for Deep Learning. *Journal of big data* Vol.6 (1), artikkeli 60. Viitattu 13.10.2022. <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>

LIITTEET

Liite 1. OCR-algoritmin kokeilun tuloksia

1 (2)

Onnistunut tunnistus normaalissa kuvaustilanteessa



AIRM
 LUMI II 320 IP44
 Snro 4297206 • A2LUFF • E-nr 7502890
 12W/840 VA • 220-240 V ~ 50/60 Hz
 Airam Electric Oy Ab
 Sementtitehtaankatu 6
 FI-04260, Kerava • www.airam.fi
 17W21

Onnistunut tunnistus noin 45 asteen kuvauskulmassa



~240V
 OPTOTRONIC® INTELLIGENT
 220 ...
 PRI
 OTi DALI 80/220-240/24 TW
 Constant voltage LED Power Supply
 4
 Prated[W]
 Lated [V]
 UN/IN
 IN[A]
 ta [°C]
 80
 24.2
 220-240V
 0/50/60 Hz
 0.39
 0.95
 -20...45

jatkuu

Epäonnistunut tunnistus arvokilvessä olevan ryppyn takia



FTLIGHT

Snro:4228749

128601/PO7817

230V LED-nauha

7W/m 480lm/m IP65 4000K 10m

1,5m kumikaapeli IP44-sukoruipalla

CE

Manufacturer / Valmistuttaja / Importer: Edutto.

Kannistontie 5, 36220 Kangasala FINLAND

Epäonnistunut tunnistus rungon heijastuksen takia



IS8

Brilone Leuchten GmbH & Co. KG

Im Kissen 2, D-59929 Brilon

Typ 3351-015

IP44

230V~50Hz

LED 13W

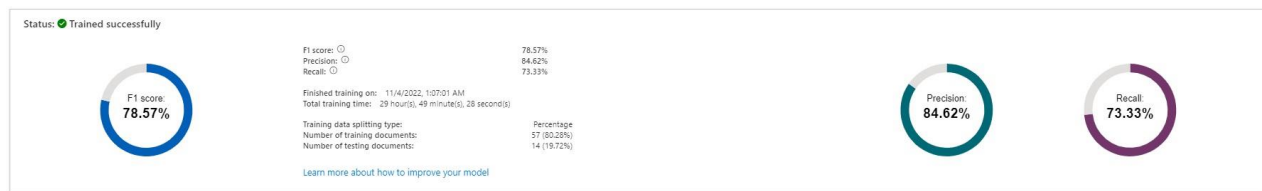
CE

BN154715

03/21

03131

Liite 2. Mukautetun entiteettitunnistusmallin metriikat

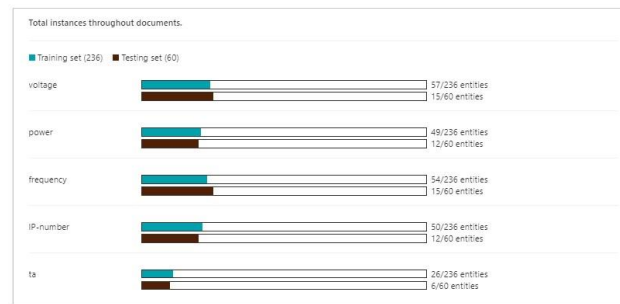
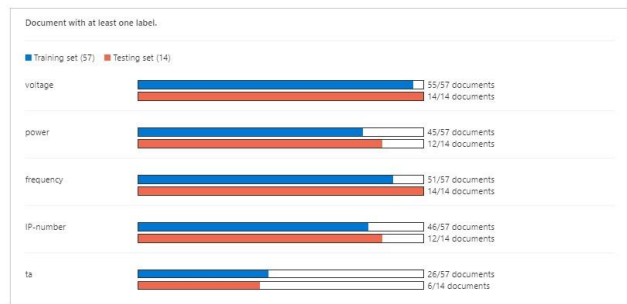


Guidance

To improve your model's performance, review these issues and follow any recommendations.

- **Unclear distinction between entity types in training set.**
When the training data is similar for multiple entity types, it can lead to lower accuracy because the entity types may be frequently misclassified as each other.
7 similar entity type(s). [View Details](#)
- ▲ **Unbalanced entity types within training or test set.**
When sampling bias causes an inaccurate representation of an entity type's frequency, it can lead to lower accuracy due to the model expecting that entity type to occur too often or too little.
1 unbalanced entity type(s). [View Details](#)
- ✓ Training set has enough data. [View Details](#)
- ✓ All entity types are present in test set. [View Details](#)
- ✓ Entity types are evenly distributed between training and test sets. [View Details](#)

Entity name ↑ ↓	Precision % ▼	Recall % ▼	F1 score ▼	Training labels ▼	Testing labels ▼
IP-number	91.67	91.67	0.917	50	12
frequency	66.67	66.67	0.667	54	15
power	100.00	33.33	0.5	49	12
ta	83.33	83.33	0.833	26	6
voltage	93.33	93.33	0.933	57	15



Labeled as

Predicted as	\$none	IP-number	power	voltage	frequency	ta
\$none	-	0	0.583	0.025	0.019	0
IP-number	0	1	0	0	0	0.056
power	0	0	0.333	0	0	0
voltage	0	0	0	0.975	0.019	0
frequency	0.079	0	0.083	0	0.962	0
ta	0	0	0	0	0	0.944

Liite 3. Entiteettitunnistusmallin kokeilut

1 (2)

<p><u>power</u></p> <p>Entity value: 21W Confidence: 100.00%</p>	<p><u>voltage</u></p> <p>Entity value: 220-240V Confidence: 100.00%</p>	<p><u>frequency</u></p> <p>Entity value: 50/60Hz Confidence: 100.00%</p>	<p><u>IP-number</u></p> <p>Entity value: IP44 Confidence: 100.00%</p>	<p><u>ta</u></p> <p>Entity value: ta: -20...+50 Confidence: 100.00%</p>
--	---	--	---	---

Original text

Briloner
9610.A1CF
21W 220-240V~ 50/60Hz IP44 ta: -20...+50
power voltage frequency IP-no. ta
Made in Germany
-OCE

<p><u>power</u></p> <p>Entity value: 21W220 Confidence: 100.00%</p>	<p><u>voltage</u></p> <p>Entity value: -240V Confidence: 100.00%</p>	<p><u>frequency</u></p> <p>Entity value: 50/60Hz P44ta Confidence: 100.00%</p>	<p><u>ta</u></p> <p>Entity value: : -20...+50 Confidence: 100.00%</p>
---	--	--	---

Original text

Briloner
9610.A1CF
21W220-240V~50/60HzIP44ta: -20...+50
power voltage frequency ta
Made in Germany
-OCE

<p><u>power</u></p> <p>Entity value: 21W Confidence: 100.00%</p>	<p><u>voltage</u></p> <p>Entity value: 220-240V Confidence: 100.00%</p>	<p><u>frequency</u></p> <p>Entity value: 50/60Hz Confidence: 100.00%</p>	<p><u>IP-number</u></p> <p>Entity value: IP44 Confidence: 100.00%</p>	<p><u>ta</u></p> <p>Entity value: ta: -20...+50 Confidence: 100.00%</p>
--	---	--	---	---

Original text

Briloner
9610354.A1CFBC
21W
power
220-240V~50/60Hz
voltage frequency
A1ramElectricoyAb
Made in Germany
-OCE
IP44
IP-no.
ta: -20...+50
ta

<p><u>voltage</u></p> <p>Entity value: 220-240V Confidence: 100.00%</p>	<p><u>frequency</u></p> <p>Entity value: 50/60Hz Confidence: 100.00%</p>	<p><u>power</u></p> <p>Entity value: 12W Confidence: 100.00%</p>	<p><u>ta</u></p> <p>Entity value: ta:-20°C-40°C Confidence: 100.00%</p>
---	--	--	---

Original text

Prolich A55FRIP44
220-240V
voltage
50/60Hz 12W
frequency power
ta: -20°C-40°C
ta
Germany
floodlight with sensor
CE

jatkuu

IP-number Entity value: IP44 Confidence: 100.00%	voltage Entity value: 220-240V Confidence: 100.00%	frequency Entity value: 50/60Hz Confidence: 100.00%	power Entity value: 12W Confidence: 100.00%	ta Entity value: ta:-20°C-40°C Confidence: 100.00%
---	---	--	--	---

Original text

Prolich A55FR-IP44
220-240V
 voltage
50/60Hz 12W
 frequency pow_
ta: -20°C-40°C
 ta
 Germany
 floodlight with sensor
 CE

voltage Entity value: 220-240V Confidence: 100.00%	frequency Entity value: 50/60Hz Confidence: 100.00%	power Entity value: 12W Confidence: 100.00%	IP-number Entity value: IP44 Confidence: 100.00%	power Entity value: 15W22 Confidence: 100.00%
---	--	--	---	--

Original text

Ledvance
220-240V 50/60Hz 12W
 voltage frequency pow_
IP44
 IP-n_
 indoor and outdoor use
 class II
 Batch 15W22
 power

power Entity value: 18W Confidence: 100.00%	voltage Entity value: 220-240V Confidence: 100.00%	frequency Entity value: 50-Hz Confidence: 100.00%	frequency Entity value: 60Hz Confidence: 100.00%	IP-number Entity value: IP5 Confidence: 100.00%
--	---	--	---	--

Original text

EUROLAMP
 REF.No:145-20016/BL200CP05/
18W, 220-240V
 pow. voltage
50Hz/60Hz
 freq. freq.
IP5 44000K
 IP-_
 CE
 EUROLAMP

power Entity value: 18W Confidence: 100.00%	voltage Entity value: 220-240V Confidence: 100.00%	frequency Entity value: 50-60Hz Confidence: 100.00%	IP-number Entity value: IP5 Confidence: 100.00%
--	---	--	--

Original text

EUROLAMP
 REF.No:145-20016/BL200CP05/
18W, 220-240V
 pow. voltage
50-60Hz
 frequency
IP5 44000K
 IP-_
 CE
 EUROLAMP

Liite 4. Mukautetun objektintunnistusmallin metriikat



Performance Per Tag

Tag	Precision	Recall	A.P.	Image count
class II	100.0%	100.0%	100.0%	50 
class III	90.9%	100.0%	100.0%	50 
class I	90.9%	100.0%	100.0%	50 

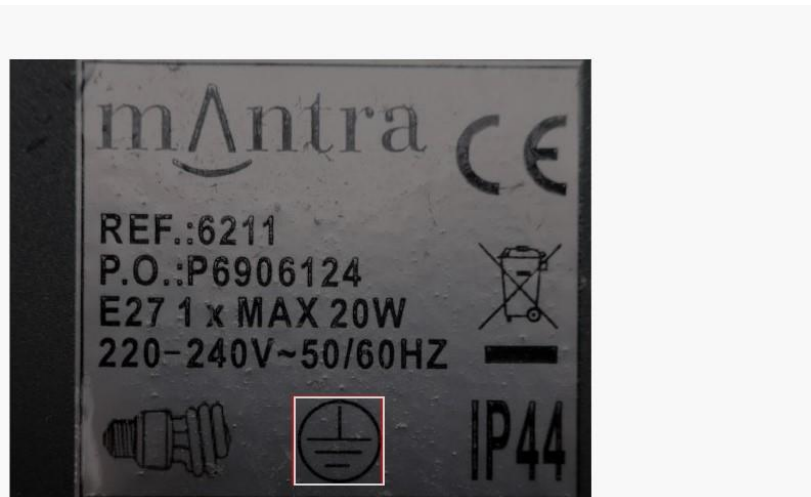
Iterations

Probability Threshold: 50% Overlap Threshold: 30% 

Liite 5. Mukautetun objektintunnistusmallin testidataa

1 (2)

Suojamaadoituksen kuvatunnuksen onnistunut tunnistus



Only show suggested objects if the probability is above the selected threshold.

Threshold Value: 15%

Predictions

Predictions are shown in red

Tag	Probability
class I	99.9%

Luokan II kuvatunnuksen onnistunut tunnistus



Only show suggested objects if the probability is above the selected threshold.

Threshold Value: 15%

Predictions

Predictions are shown in red

Tag	Probability
class II	99.9%

jatkuu

Luokan III kuvatunnuksen onnistunut tunnistus



Only show suggested objects if the probability is above the selected threshold.

Threshold Value: 15%

Predictions

Predictions are shown in red

Tag	Probability
class III	99.9%

Väärä tunnistuksia



Only show suggested objects if the probability is above the selected threshold.

Threshold Value: 15%

Predictions

Predictions are shown in red

Tag	Probability
class III	99.9%
class I	17.5%
class I	17.1%
class III	16.7%