

# OHJELMISTOROBOTIIKAN YHDISTÄMINEN KONEOPPIMISEEN



Ammattikorkeakoulututkinnon opinnäytetyö

Hämeenlinnan korkeakoulukeskus  
Tietojenkäsittelyn koulutusohjelma

Syksy, 2019

Jenny Mustonen

Tietojenkäsittelyn koulutusohjelma  
Hämeenlinnan korkeakoulukeskus

---

<b>Tekijä</b>	Jenny Mustonen	<b>Vuosi</b> 2019
<b>Työn nimi</b>	Ohjelmistorobotiikan yhdistäminen koneoppimiseen	
<b>Työn ohjaaja</b>	Lasse Seppänen	

---

## TIIVISTELMÄ

Tämän opinnäytetyön tavoitteena oli tutkia, kuinka rakennetaan yksinkertainen laskuja lukeva ohjelmistorobotti. Lisäksi tavoitteena oli tutkia, miten tällaista ohjelmistorobottia voidaan opettaa koneoppimisen avulla. Opinnäytetyön tavoitteena oli myös kasvattaa osaamista ohjelmistorobotiikasta työelämää ajatellen. Työ toteutettiin toimeksiantajan CGI Suomen idean perusteella.

Opinnäytetyön teoriaosassa käydään läpi ohjelmistorobotiikan kehityksen vaiheita yhtenä automaation tasona. Työssä kerrotaan perusteita automatisoitavien prosessien valinnasta ja kehityksestä. Lisäksi työssä on kerrottu käytännönläheisin esimerkein ohjelmistorobotiikan hyötyjä sekä haasteita. Ohjelmistorobotiikkaa voidaan kehittää useiden eri työkalujen avulla, joista työssä on esitelty yleisimmät. Tekoälyn osalta työssä on kerrottu sen yhteiskunnallisista vaikutuksista sekä esimerkein tämän hetkisestä tilasta. Lisäksi työssä on lyhyesti esitelty työhön koneoppimisen työkaluksi valittu Microsoft Azure Machine Learning.

Opinnäytetyön toiminnallisessa osassa kehitettiin UiPath-ohjelmistolla PDF-dokumentteja käsittelevä ja lukeva ohjelmistorobotti. Kehityksen aikana ilmenneistä haasteista johtuen ohjelmistorobotin kehitys jäi kesken ja työn tavoitteeksi määritelty koneoppimisen yhdistämisen tutkiminen jouduttiin jättämään käytännössä pois työstä. Tästä huolimatta ohjelmistorobotin kehityksen myötä opinnäytetyön tavoitteet osaamisen kasvattamiseksi toteutuivat. Opinnäytetyön teoriaosuus antoi myös hyvän kuvan tekoälyn ja koneoppimisen perusteista.

**Avainsanat** ohjelmistorobotiikka, RPA, koneoppiminen, tekoäly, UiPath

**Sivut** 39 sivua

Degree Programme in Business Information Technology  
Hämeenlinna University Centre

---

<b>Author</b>	Jenny Mustonen	<b>Year</b> 2019
<b>Subject</b>	Combining robotic process automation with machine learning	
<b>Supervisor</b>	Lasse Seppänen	

---

ABSTRACT

The purpose of this thesis was to study how to build a simple invoice-reading software robot. A further aim was to investigate how such a software robot can be taught through machine learning. The aim of the thesis was also to increase the knowledge of software robotics for working life. The work was carried out based on the idea of CGI Suomi.

The theory part of this thesis deals with the development of software robotics as one level of automation. The paper discusses the basics of selecting and developing automated processes. In addition, practical examples of the benefits and challenges of software robotics have been discussed. Software robotics can be developed with a variety of tools, the most common of which are presented in the thesis. As far as artificial intelligence is concerned, the work discusses its social effects and examples of its current state. In addition, Microsoft Azure Machine Learning, a machine learning tool for work, is briefly introduced.

In the functional part of the thesis, a PDF-document-processing and reading software robot was developed with UiPath software. Due to the challenges during the development, the development of the software robot was not completed and the study of combining machine learning with the goal of the work had to be left out of practice. Nevertheless, with the development of the software robot, the aims of the thesis to increase knowledge were realized. The theoretical part of the thesis also gave a good idea of the basics of artificial intelligence and machine learning.

**Keywords** Robotic Process Automation, RPA, machine learning, UiPath

**Pages** 39 pages

## KÄSITTEET

AI	Artificial Intelligence, tekoäly. Inhimillistä ajattelua jäljittelevä kone.
Algoritmi	Yksityiskohtainen kuvaus tai ohje siitä, miten tehtävä suoritetaan.
API	Application Programming Interface, ohjelmointirajapinta, jonka kautta sovellukset voivat kommunikoida keskenään.
CV	Computer vision, konenäkö
ML	Machine learning, koneoppiminen
OCR	Optical character recognition, optinen tekstintunnistus
Pilvipalvelu	Internetin yli käytettävä palvelu, jossa tiedot ovat palvelun tarjoavan yrityksen palvelimella eivätkä käyttäjän omalla tietokoneella tai oman yrityksen palvelimella.
RPA	Robotic Process Automation, ohjelmistorobotiikka

# SISÄLLYS

1	JOHDANTO.....	1
2	OHJELMISTOROBOTIIKKA .....	2
2.1	Ohjelmistorobotiikan evoluutio .....	2
2.2	Optinen tekstintunnistus ohjelmistorobotiikassa.....	4
2.3	Automatisoitavien prosessien valinta .....	4
2.4	Ohjelmistorobotiikan hyötyjä .....	5
2.5	Ohjelmistorobotiikan haasteita .....	6
2.6	Ohjelmistorobotiikan työkaluja.....	7
2.6.1	UiPath .....	7
2.6.2	BluePrism.....	8
2.6.3	Automation Anywhere .....	9
2.6.4	Robot Framework.....	9
3	KONEOPPIMINEN.....	10
3.1	Tekoälyn vaikutukset yhteiskuntaamme .....	10
3.2	Koneoppiminen yhtenä tekoälyn mallina .....	11
3.3	Koneoppiminen teknisesti.....	12
3.4	Netflix-palvelun suositteluominaisuus koneoppimisen esimerkkinä .....	13
3.5	Tekoälyyn liittyviä uhkia.....	14
3.6	Microsoft Azure Machine Learning.....	15
4	OHJELMISTOROBOTIN SUUNNITTELU JA KEHITYS .....	17
4.1	Ohjelmistorobotin suunnittelu.....	17
4.2	Ohjelmistorobotin kehitys.....	18
4.2.1	Käsiteltävien tiedostojen määrittely .....	20
4.2.2	PDF-dokumenttien lukeminen .....	22
4.2.3	Kuvien eriyttäminen ja vertailu .....	25
4.2.4	Laskun lukeminen konenäöllä .....	29
4.2.5	Kuvien vertailu pikseleittäin .....	30
4.2.6	Viitteen paikannus laskulta .....	31
5	TULOKSET .....	32
5.1	Tiedon etsiminen.....	32
5.2	Virhetilanteiden paikantaminen .....	32
5.3	Logon vertaaminen pikseli pikseliltä .....	32
5.4	Logon lukeminen konenäöllä .....	33
5.5	Koneoppimisen hyödyntäminen ohjelmistorobotiikassa .....	34
6	YHTEENVETO .....	35
	LÄHTEET.....	36

## 1 JOHDANTO

Tämän opinnäytetyön tarkoitus on johdattaa lukijansa ohjelmistorobotiikan ja koneoppimisen maailmaan sekä kasvattaa ymmärrystä näistä teknologioista. Ohjelmistorobotiikka, Robotic Process Automation (RPA) on noussut yhdeksi ohjelmistoalan polttavimmista aiheista viime vuosien aikana. Ohjelmistorobotiikan pääajatuksena on automatisoida ihmisten suorittamia rutiininomaisia, toistuvia työtehtäviä. Robotti suorittaa annettuja tehtäviä kuten ihminenkin niitä suorittaisi. Robotin työskennellessä voit nähdä ohjelmistojen avautuvan, tekstin kirjoittuvan näytölle ja painikkeiden painuvan näytöllä hiiren klikkauksen jälkeen.

Ohjelmistorobotti käsittelee annettujen sääntöjen mukaisesti sellaiset tehtävät, jotka on mahdollista käsitellä säännönmukaisesti. Poikkeaman ilmetessä robotti ohjaa kyseisen tapauksen manuaaliseen käsittelyyn. Esimerkki tästä on automatisoitu laskun käsittely, jossa laskun laskuttaja tunnistetaan logon perusteella. Robotille on määritelty tarvittavien tietojen sijainti tunnettujen laskuttajien laskupohjilla. Kun laskuttaja on tunnistettu, etsii robotti tarvittavat tiedot sijaintitietojen perusteella. Jos robotti ei löydä tarvittavia tietoja, ohjataan lasku manuaaliseen käsittelyyn. Yhtenä ongelmana tällaisessa robotissa on tilanne, jolloin samalla laskuttajalla on käytössään erilaisia laskupohjia ja robotti osaa käsitellä näistä vain yhden. Tämän opinnäytetyön tarkoituksena on tutkia mahdollisuuksia rakentaa tällaiselle robotille kyvykkyys oppia poikkeamien manuaalisesta käsittelystä ja näin ollen ratkaista yhä useampi tapaus rutiininomaisesti samalla automaatiotasoa vähitellen kasvattaen.

Opinnäytetyö on toteutettu yhteistyössä CGI Suomen Intelligent automation -yksikön kanssa. CGI Suomi Oy on kansainvälisesti toimiva täyden palvelun IT-palvelutoimittaja, jonka pääkonttori sijaitsee Kanadassa. Intelligent automation -yksikössä tuotetaan asiakkaille innovatiivisia ohjelmistorobotiikkaratkaisuja, joissa on hyödynnetty muun muassa koneoppimista. Tyypillisesti koneoppimista on kohdistettu ohjelmistorobotin käyttämään tietovarastoon. Yksikön jatkuvan kehityksen kohteena ovat eri tavat koneoppimisen hyödyntämiseksi ohjelmistorobotin oman sääntöpohjaisen logiikan kehittämiseksi. (CGI Suomi Oy, n.d.)

Työssä on tarkoitus rakentaa yksinkertainen ohjelmistorobotti esimerkkinä mainittuun laskun käsittelyyn. Robottiin pyritään yhdistämään koneoppimista, jonka avulla manuaalisesti kirjattu tieto opettaa robottia. Työssä kehitettyä robottia ei viimeistellä työelämän tarpeisiin soveltuvaaksi. Sen toimintaa ei rakenneta vikasietoiseksi eikä sitä testata. Tässä opinnäytetyössä vastataan kysymyksiin:

- Miten rakennetaan laskuja lukeva ohjelmistorobotti?
- Miten laskujen käsittelyä voidaan parantaa koneoppimisella?

## 2 OHJELMISTOROBOTIIKKA

Automaatiolla tarkoitetaan teknologiaa, jonka avulla ihmisten osuus erilaisissa prosesseissa voidaan minimoida tai jopa poistaa kokonaan. Termin automaatio uskotaan syntyneen jo 1940-luvulla ja nykyään automaatio on yleistynyt koskettamaan lähestulkoon kaikkea ja kaikkia arkielämässämme. Nykypäivänä automaatio on saavuttanut niin vahvan sijan ohjelmistokehityksessä, että useat uudet teknologiat ovat kehittyneet sen pohjalta. Ohjelmistorobotiikka on yksi näistä. (Tripathi, 2018, s. 6 - 9)

Ohjelmistorobotiikalla tarkoitetaan lyhyesti sanottuna ohjelmistoa, joka jäljittelee ihmistä suorittaen prosessin säännönmukaisia vaiheita järjestyksessään. Ohjelmistorobotiikalla voidaan ratkaista monimutkaisempia päätöstä vaativia vaiheita, mikäli robottia ohjelmoitaessa tätä varten on määriteltävä tarkat ehdot. Perinteisen automaation ja ohjelmistorobotiikan selkein ero on ohjelmistorobottien itsenäisyys. Ohjelmistorobotit eivät vaadi järjestelmäintegraatiota ja näin ollen prosessien automatisoinnista on tullut entistä helpompaa. Muihin automaatioteknologioihin verrattuna ohjelmistorobotiikassa yhtenä selkeänä etuna on yhteensopivuus erilaisten ohjelmistojen ja rajapintojen kanssa. (Tripathi, 2018, ss. 10 - 11)

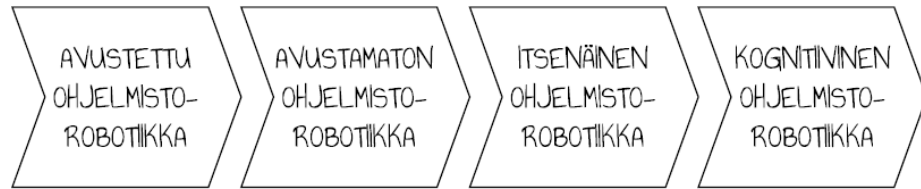
Ohjelmistorobotiikan avulla voidaan käsitellä niin massoiittaiset ja usein toistuvat yksinkertaisemmat kuin monimutkaisemmat prosessit. Tämän myötä ihmiset voivat keskittyä mielekkäämpiin ja arvoa tuottaviin tehtäviin näiden yksitoikkoisten rutiinitehtävien sijaan. Ohjelmistorobotiikan avulla näiden rutiinitehtävien inhimillisten erehdysten määrä minimoituu, sillä robotti kykenee käsittelemään identtisesti kerran luotujen ehtojen mukaisesti loputtoman määrän tehtäviä. Ohjelmistorobotti kykenee työskentelemään virheettömästi vaikkapa ympäri vuorokauden. (Tripathi, 2018, ss. 10 - 11; Månsson, 2017; ks. myös UiPath, n.d.a)

Tällä hetkellä ohjelmistorobotiikka on suosituinta pankki-, vakuutus-, telekommunikaatio- ja sähköalan yrityksissä. Suosio selittyy näiden alojen käyttämällä perinteisillä ohjelmistoilla, joita on helppo automatisoida ohjelmistorobotiikalla ilman järjestelmäintegraatioita tai uusien ohjelmistojen hankintaa. Vuonna 2018 ohjelmistorobotiikan käyttö kasvoi 63,1 prosenttia ja tänä vuonna 2019 kasvun on odotettu jatkuvan vieläkin kiivaampana. (Soumik, 2019)

### 2.1 Ohjelmistorobotiikan evoluutio

AIMDek Technologiesin (2018) mukaan ohjelmistorobotiikan evoluution voidaan kuvailla koostuvan neljästä kuvassa 1 esitellystä, selkeästi erottavasta vaiheesta. Ensimmäinen ja yksinkertaisin vaihe on avustettu ohjelmistorobotiikka. Vaiheesta käytetään myös nimitystä RPA 1.0. Tällöin automaatio tapahtuu selkeästi näkyvästi tietokoneen näytöllä ja tarvitsee toimiakseen ihmisen apua. Tällainen robotti voidaan esimerkiksi käynnistää

työntekijän toimesta aina tarvittaessa ja sille voidaan antaa tiettyjä tietoja, valintoja tai käskyjä robotin toimiessa. Avustetulla ohjelmistorobotiikalla voidaan automatisoida yksinkertaisia prosesseja, kuten tiedon siirtoa järjestelmästä toiseen. (AIMDek Technologies, 2018)



Kuva 1. Ohjelmistorobotiikan evoluution vaiheet

Avustamaton ohjelmistorobotiikka on avustetun robotiikan seuraava vaihe. Tätä kutsutaan myös nimellä RPA 2.0. Avustamaton robotti toimii ilman ihmisen vaikutusta ja se voi toimia myös taustalla. Tällainen robotti voi käynnistyä ajastetusti tietyllä aikataululla tai muulla halutulla logiikalla. Tällaiset robotit voivat toimia vuorokauden ympäri vapauttaen työntekijän kokonaan muihin tehtäviin. Kuvassa 2 on esitetty avustetun ja avustamattoman ohjelmistorobotin erot vertaillen. (AIMDek Technologies, 2018)

AVUSTETTU RPA		AVUSTAMATON RPA
SUORITTAAN TEHTÄVIÄ YKSITTÄISELLE TYÖNTEKIJÄLLE	MITÄ	AUTOMATISOI TAUSTATEHTÄVIÄ
TYÖNTEKIJÄ KÄYNNISTÄÄ ROBOTIN JA ANTAA SILLE TARVITTAVIA TIETOJA TAI KÄSKYJÄ	MITEN	TOIMII TAUSTALLA SÄÄNNÖNMUKAISESTI
TYÖNTEKIJÄ KÄYNNISTÄÄ ROBOTIN AINA TARPEEN TULLEN	MILLOIN	TOIMII ILMAN TYÖNTEKIJÄN TOIMIA ENNALTA MÄÄRÄTYLLÄ LOGIIKALLA
NOPEUTTAA ASIAKASPALVELUA JA KASVATTAASIAKASTYYTYVÄISYYTTÄ	MIKSI	VAPAUTTAA TYÖNTEKIJÄT MIELEKKÄÄMPIIN TEHTÄVIIN ALENTAEN KUSTANNUKSIA JA PARANTAEN LAATUA

Kuva 2. Avustetun ja avustamattoman ohjelmistorobotin erot (Automation Anywhere, n.d.)

Seuraava ohjelmistorobotiikan vaihe RPA 3.0., itsenäinen ohjelmistorobotiikka, on tällä hetkellä viimeisin ohjelmistorobotiikan muoto. Tässä ohjelmistorobotiikkaan yhdistetään tekoälyä, kuten koneoppimista ja koneäköä. Tekoälyn avulla ohjelmistorobottia voidaan muokata koko ajan toimivammaksi ja kattavammaksi. (AIMDek Technologies, 2018)



Evoluution viimeisin vaihe, kognitiivinen ohjelmistorobotiikka eli RPA 4.0. on vasta kehittyneessä. Sen avulla ohjelmistorobotiikkaan yhdistetään useita erilaisia teknologioita, kuten esimerkiksi tiedonlouhintaa, tekstin analysointia, koneoppimista ja luonnollisen kielen käsittelyä. (AIMDek Technologies, 2018)

## 2.2 Optinen tekstintunnistus ohjelmistorobotiikassa

Ihmisen lukiessa dokumenttia hän tunnistaa dokumentista erilaisia sanoja sekä kuvia ja tämän perusteella tekee jatkopäätelmiä. Ohjelmistorobotti pyrkii optisen tekstintunnistuksen avulla toimimaan samoin; tunnistamaan dokumentin, esimerkiksi laskun, sisällön ja sen perusteella suorittamaan halutut jatkotoimet. Tekstintunnistus, Optical Character Recognition (OCR) käsitteenä tarkoittaa sekä koneellisesti tuotetun tekstin tunnistamista, että käsinkirjoitetun tekstin tunnistamista (Intelligent Character Recognition - ICR). OCR-työkalut tunnistavat dokumentin rakenteen, tunnistavat dokumentista tekstiä, taulukoita, kuvia ja niin edelleen sekä muuntavat nämä koneen ymmärtämäksi tekstiksi. Tekniikkaa käytetään yleisesti skannattujen asiakirjojen, kuten esimerkiksi laskujen ja tiliotteiden käsittelyyn. (Docparser, n.d.; Techopedia, n.d.)

Tekstintunnistuksella voidaan tunnistaa sekä koneellista tekstiä että käsinkirjoitettua tekstiä. Tekniikalla voidaan tunnistaa tekstiä esimerkiksi myös valokuvasta. Tunnistuksen luotettavuus on riippuvainen luettavan dokumentin laadusta. Tekstintunnistuksen luotettavuus on laadukkaasta dokumentista koneellista tekstiä luettaessa yli 99 prosenttia ja käsinkirjoitettua tekstiä luettaessa 90 prosenttia. Dokumentin laadun laskiessa myös luotettavuus laskee 80 prosenttiin. (Kameswara Rao ym., 2019)

## 2.3 Automatisoitavien prosessien valinta

Automatisointi ei ole itseisarvo. Ennen prosessien automatisointia on tarpeellista miettiä, onko automatisoitavaksi ajateltua tehtävää tarpeen tehdä lainkaan. Jos tehtävä todetaan edelleen tarpeelliseksi, on seuraavaksi yhtenäistettävä prosessit ennen automatisointia. Toimimattoman prosessin automatisointi ei ole järkevää. Kun prosessit on kehitetty tarkoituksen mukaisiksi ja toimiviksi, voidaan tehtävät automatisoida ketterästi ohjelmistorobotiikan avulla. (Kaarlejärvi, 2017)

On prosesseja, jotka kannattaisi automatisoida ja prosesseja, jotka on mahdollista automatisoida. Toistuvat ja paljon aikaa vievät prosessit sekä prosessit, joissa on suuri mahdollisuus virheisiin ovat erityisesti sellaisia, jotka kannattaisi automatisoida. On kuitenkin tärkeää miettiä mitä voidaan automatisoida. Jotta prosessi voidaan automatisoida, on sen oltava selkeä, sääntöjä noudattava ja johdonmukainen. Esimerkiksi prosessia, jossa esiintyy vaihtelua tai joka vaatii ihmisen tilannekohtaista päättelykykyä, ei voida automatisoida. Koska automaation tavoitteena on aina myös

kustannustehokkuus, on prosessin automatisoinnin hyötyjen oltava suurempia, kuin siitä aiheutuvat kustannukset. Kustannuksia laskettaessa on muistettava huomioida kehityskustannusten lisäksi ylläpito tulevaisuudessa. (Tripathi, 2018, ss. 7 - 8; Månsson, 2017)

#### 2.4 Ohjelmistorobotiikan hyötyjä

Ohjelmistorobotiikan suurimpina hyötyinä aiemmin mainitun täsmällisyyden tuoman laadukkuuden lisäksi ovat muun muassa tapahtumien raportoitavuus, kustannustehokkuus, nopeus, ketteryys ja monipuolisuus. Tämän lisäksi yksi huomionarvoinen hyöty on työntekijöiden työtyytyväisyys. Ohjelmistorobotiikan vähentäessä puuduttavia rutiinitehtäviä, voivat työntekijät keskittyä mielekkäämpiin tehtäviin. (Tripathi, 2018, ss. 11 - 14)

Ainoana etuna ei kuitenkaan ole ainoastaan nykyisten prosessien automatisointi. Ohjelmistorobotiikan avulla voidaan luoda uusia prosesseja, joita nykyisin ei vielä ole olemassa. Näillä prosesseilla voidaan suorittaa erilaisia tarkistuksia laadun parantamiseksi. Ohjelmistorobotiikan avulla voidaan myös hyödyntää vanhempia tietojärjestelmiä paremmin koko niiden elinkaaren ajan ilman suuria investointeja vanhaan tietojärjestelmään. (Kääriäinen ym., 2018, s. 12; ks. myös Månsson, 2017)

Soumik (2019) mainitsee artikkelissaan hyvän esimerkin laadun parantamisesta pankkialalla ilman suoranaista rahallista tai ajallista säästöä. Aiemmin ihmisten suorittamien tehtävien virheiden määrä on laskenut merkittävästi ohjelmistorobotiikan käyttöönoton myötä. Muutoksen ei ole vielä todettu tuoneen rahallista säästöä, mutta laadussa merkitys on huomattava. (Soumik, 2019)

Brandallin esimerkin mukaan myyjien ajankäyttöä tutkittaessa kuvan 3 mukaisesti osa tehtävistä on automatisoitavissa. Tarkemmin katsottuna automatisoitavia tehtäviä on yhteensä enemmän kuin mitä myyjä käyttää itse myyntityöhön. Tämä on hyvä esimerkki siitä, miten oikeanlaisten asioiden automatisointi vapauttaa aikaa tärkeämpiin ja mielekkäämpiin työtehtäviin. Ei kuitenkaan ole itsestäänselvyys, että kaikki nämä automatisoitavissa olevat tehtävät automatisoitaisiin. Aiemmin automaatiota on käytetty isojen ja raskaiden prosessien automatisointiin. Ohjelmistorobotiikan myötä myös pienempiä, yksittäisen työntekijän tehtäviä on mahdollista automatisoida kustannustehokkaasti. (Brandall, 2018)



Kuva 3. Myyjien ajankäyttö (Brandall, 2018)

## 2.5 Ohjelmistorobotiikan haasteita

Monen ensimmäinen ajatus robotiikasta ja automaatiosta on pelko oman työpaikan korvaantumisesta. Teollisuudessa robotit ovat poistaneet rutiinitöitä tuotannosta. Ohjelmistorobotiikalla voidaan saada sama aikaan toimistotyössä. On kuitenkin hyvä muistaa, että automaatiolla pyritään pääasiassa korvaamaan toistuvia, työntekijälle mielenkiinnottomia työtehtäviä. Tällöin työntekijä vapautuu muihin mielekkäisiin tehtäviin kasvattaen työviihtyvyyttä. (AIMDek Technologies, 2018; CGI Suomi Oy, 2016)

Automaation lisääntyessä osaamisen tarve muokkautuu, sillä jatkossa teknisiä osaajia tarvitaan entistä enemmän. Tässä haasteeksi voi muodostua osaavien työntekijöiden vähyys alalla. Työpaikkojen poistumisen sijaan ohjelmistorobotiikka pikemminkin muokkaa nykyisiä työmarkkinoita. Digitalisaation eri osa-alueiden vaikutukset ovat hyvin samankaltaisia eikä vaikutuksia ole helppo erottaa toisistaan. (AIMDek Technologies, 2018; ks. myös Työ- ja elinkeinoministeriö, 2018, s. 10)

Ohjelmistorobotiikkaan liittyy toki myös muita haasteita. Raution (Elisa Oyj, 2019) mukaan kokemusten perusteella ohjelmistorobotiikkaprojekteista nousee esille kolme tyypillisintä haastetta. Projektien ensimmäiset haasteet liittyvät aiemmin mainittuun automatisoitavien prosessien valintaan. Ohjelmistorobotiikan järkevä hyödyntäminen edellyttää yksinkertaisia ja suoraviivaisia prosesseja. Usein tarkemman tarkastelun myötä huomataan tämän toteutuvan vain 90-prosenttisesti. Mikäli tähän vaiheeseen ei kiinnitetä tarpeeksi huomiota, saatetaan kehityksen myöhemmissä vaiheissa kohdata ongelmia. Pahimmillaan kehitetty ohjelmistorobotti osoitautuu turhaksi tai toimimattomaksi. (Elisa Oyj, 2019)

Seuraava ohjelmistorobotiikan ongelma Raution (Elisa Oyj, 2019) mukaan on robotin hallinta ja valvonta. On muistettava, ettei robotti kykene tekemään päätöksiä, valintoja tai kyseenalaistamaan tekemäänsä. Robotti on tehtävässään täydellinen. Se toistaa tekemänsä väsymättömästi eikä tee koskaan virheitä. Robotti voi toimia virheellisesti, mikäli sen kehityksessä on tapahtunut virhe. Tällöin robotti systemaattisesti toistaa virheensä. Se kuitenkin noudattaa vain sille annettua ohjetta, joka on virheellinen. Oh-

jelmistorobotiikkaprojektia suunniteltaessa on siis muistettava varata tarpeeksi aikaa itse kehitystyön lisäksi robotin valvontaan ja hallintaan. (Elisa Oyj, 2019; CGI Suomi Oy, 2016)

Kolmas tyypillinen ongelma on ohjelmistorobotin kustannuksien arviointi. Kehitykseen käytettävät työkalut ovat maksullisia ja niiden kustannusrakenteet voivat olla hyvin erilaisia. On tärkeää projektin aloitusta huomioida kaikki robotin kehityksen kustannukset ja verrata niitä prosessin tämän hetkisiin kustannuksiin. Markkinoilla on useita erilaisia ratkaisuja ohjelmistorobotiikan toteuttamiseen, joten on perehdyttävä huolella eri vaihtoehtoihin ja niiden kustannuksiin. Kustannuksia laskettaessa on muistettava huomioida sekä työkalun hankintaan, että ylläpitoon liittyvät kulut. (Elisa Oyj, 2019)

## 2.6 Ohjelmistorobotiikan työkaluja

Markkinoilla on jo useita erilaisia työkaluja ohjelmistorobottien kehitykseen, joista tässä työssä esitellään lyhyesti kolme kaupallista työkalua ja yksi avoimen lähdekoodin työkalu. Valitut kaupalliset työkalut ovat alansa markkinajohtajia ja valikoituneet esiteltäviksi sen perusteella. Markkinoilla on kuitenkin useita muita työkaluja näiden seuraavaksi esiteltyjen lisäksi. (Edureka!, 2019)

### 2.6.1 UiPath

Työkaluista suosituin UiPath tarjoaa tässä esitellyistä kolmesta graafisen editorin omaavista työkaluista parhaan mahdollisuuden harjoitella ohjelmiston käyttöä ilman maksullisia lisenssejä. Ohjelmistosta on saatavilla täysin maksuton versio sekä kattavat oppaat ohjelmiston käytön opiskeluun. UiPathin käyttöliittymä on selkeä eikä ohjelmiston käyttäminen vaadi välttämättä ohjelmointitaitoja. Tämän myötä ohjelmisto ei kuitenkaan ole laajennettavissa ohjelmoinnin avulla yhtä laajasti kuin BluePrism tai Automation Anywhere. (Edureka!, 2019; UiPath, n.d.b; RPA Training, n.d.; Omnisys, 2019)

Työkalu on suunniteltu nopeaan kehitykseen sekä siihen on mahdollista yhdistää erilaisia tekoälyominaisuuksia. Maksuton versio on helposti laajennettavissa tarvittaessa myöhemmin maksulliseen versioon. Työkalu ei sovellu luotettavasti isoihin robotiikkakokonaisuuksiin, joissa sen toimintavarmuus on selkeästi BluePrismiä ja Automation Anywhereä alhaisempi. UiPath on siis parhaimmillaan pienissä ja keskisuurissa automatisointiprojekteissa. (Edureka!, 2019; UiPath, n.d.b; RPA Training, n.d.; Omnisys, 2019)

UiPath koostuu kolmesta eri osasta: UiPath Studio, UiPath Robot ja UiPath Orchestrator. Studion selkeän, graafisen käyttöliittymän avulla ohjelmointitaidottomankin on mahdollista kehittää ohjelmistorobotteja. Käyttöliittymä ohjaa käyttäjää virhetilanteissa. Robotin suorittamat vaiheet määritellään vuokaaviomaisesti käyttäen erilaisia aktiviteetteja raahaa-ja-pudota-tyyppisesti. Studiota on mahdollista käyttää selainpohjaisesti, joten sen käyttö onnistuu työasemariippumattomasti. Siitä on saatavilla kuitenkin myös työpöytäversio. Studiolla saman robotin kehitys onnistuu useammalta henkilöltä samanaikaisesti, joten se soveltuu tiimien käyttöön. Studiolla on mahdollista luoda pohja robotille nauhoittamalla, jonka avulla kehitys on entistäkin helpompaa. (Tripathi, 2018, ss. 20 - 21, s. 24; UiPath, n.d.b)

UiPath Robot on työpöytäsovellus, jolla voidaan käynnistää Studiolla luotu robotti. Studiolla ohjelmoitujen robottien hallinnointi onnistuu selainpohjaisen Orchestratorin kautta. Orchestratorilla voidaan ajastaa ja seurata kaikkien robottien toimintaa. Usein samasta robotista on kehitetty useita eri versioita. Orchestrator mahdollistaa eri versioiden hallinnan. (Tripathi, 2018, s. 21, ss. 25 - 26)

UiPathilla on oma Academy oppimisympäristö, UiForum-keskustelupalsta sekä UiPath Go! -verkkosivusto. Academy sisältää useita erilaisia verkkokursseja eri aiheista. Kurseista suurin osa on ilmaisia ja kenen tahansa käytettävissä. Keskustelupalstalla käyttäjät voivat kysyä apua ongelmiinsa toisiltaan ja jakaa omia työnkulkujaan keskusteluiden yhteydessä. Keskustelut ovat julkisia, joten myös muut samoja ongelmia kohdanneet käyttäjät voivat löytää avun jo käydystä keskustelusta. UiPath Go! -sivustolle käyttäjät voivat ladata omia yksittäisiä aktiviteettejaan, työnkulkujaan tai kokonaisprojekteja ohjeineen. Muut käyttäjät voivat ladata näitä komponentteja omaan käyttöönsä hyödyntäen niitä oman robottinsa rakentamisessa. (UiPath Academy, n.d.b.; UiPath Forum, n.d.; UiPath Go!, n.d.)

## 2.6.2 BluePrism

Kuten UiPathin, myös BluePrismien käyttäminen onnistuu ilman syvällisiä ohjelmointitaitoja. Työkalu vaatii kuitenkin UiPathia enemmän ohjelmointiosaamista ja se perustuu C#-ohjelmointikieleen. BluePrismistä ei ole saatavilla lainkaan kokeiluversiota, joten työkalun käyttö vaatii aina lisenssin ostamista. Ohjelmistoa käytetään työpöytäsovelluksen kautta, joten käyttö vaatii ohjelmiston asennuksen. Työkalusta puuttuu UiPathin ja Automation Anywheren kaltainen mahdollisuus prosessien nauhoitukseen, joka nopeuttaa prosessien automatisointia. Työkalu soveltuu Automation Anywheren kaltaisesti isompiin tarpeisiin säilyttäen kuitenkin tällöinkin luotettavuutensa. (RPA Training, n.d.)

### 2.6.3 Automation Anywhere

Mainituista kolmesta työkalusta Automation Anywhere vaatii vankimman ohjelmointiosaamisen, eikä työkalussa ole graafisia ominaisuuksia kuten UiPathissa tai BluePrismissä. Automation Anywhere pohjautuu Microsoftin teknologioihin. Kuten BluePrismiä myös Automation Anywhereä käytetään työpöytäsovelluksen kautta. Automation Anywherestä on saatavilla 30 päivän kokeiluversio, jonka jälkeen ohjelmiston käyttäminen vaatii lisenssin ostamista. Automation Anywhere säilyttää luotettavan toimintavarmuutensa isommissakin projekteissa. (RPA Training, n.d.; Omnisys, 2019)

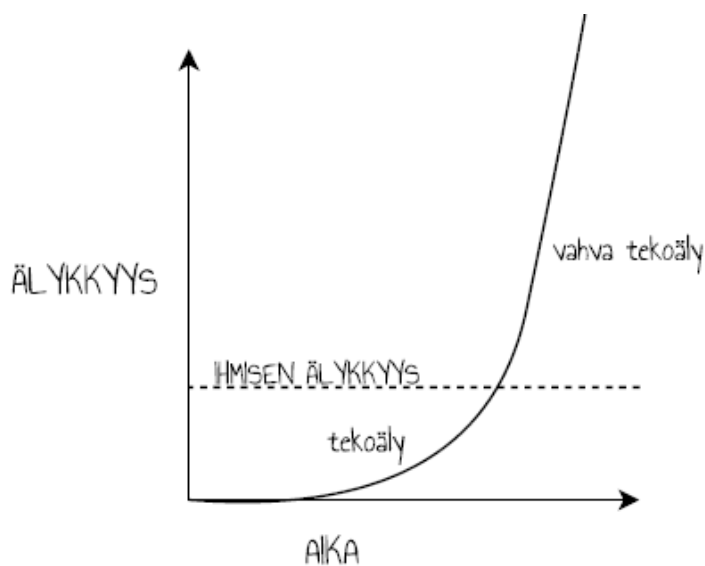
### 2.6.4 Robot Framework

Robot Framework on alun perin testauksen automatisointiin luotu avoimen lähdekoodin työkalu. Maaliskuussa 2018 viisi ohjelmistoalan yritystä aloittivat kehityshankkeen, jonka tavoitteena oli luoda Robot Frameworkista avoimen lähdekoodin ekosysteemi ohjelmistorobotiikkaan. Avoimen lähdekoodinsa vuoksi Robot Framework on joustava ja monipuolinen työkalu ohjelmistorobotiikkaratkaisuihin. Avoimen lähdekoodin etuna on nimenomaan joustavuus. Yksinkertaisimmillaan avoimen lähdekoodin sovellus on hyvin yksinkertainen ja kevyt, mutta sovellus on kuitenkin laajennettavissa lähes loputtomasti tarpeen mukaan. Robot Framework perustuu ohjelmointikieli Pythoniin ja kaupallisiin työkaluihin verrattuna siitä puuttuu täysin graafinen editori. (Laitila, 2018; Knowit, 2018; Robot Framework Foundation, n.d.)

### 3 KONEOPPIMINEN

Tässä luvussa kerrotaan esimerkein mitä koneoppiminen on ja kuinka koneoppiminen näkyy tällä hetkellä yhteiskunnassamme. Koska koneoppiminen on yksi tekoälyn malleista, kerrotaan luvussa perusteita myös tekoälystä. Luvussa avataan ihmisten kokemia tekoälyyn liittyviä uhkia ja sekä kerrotaan työhön valitusta työkalusta.

Siinä missä ohjelmistorobotti vain noudattaa poikkeuksetta sille annettuja sääntöjä, pystyy tekoäly päättämään, kuinka toimia uusissa tilanteissa. Tekoäly pystyy toimintaan, jonka perinteisesti on ajateltu vaativan ihmisälyä. Tekoäly oppii koko ajan omasta tekemisestään entistäkin älykkäämmäksi. Vaikka tämä kuulostaakin ilmiömäiseltä, ei tekoäly kuitenkaan käytännössä tällä hetkellä kykene vielä esimerkiksi laajentamaan osaamistaan tai toimimaan ongelmatilanteissa täysin itsenäisesti. Tätä kutsutaan heikoksi tekoälyksi. Kuvassa 4 on havainnollistettu ihmisen älykkyys verrattuna tekoälyyn. Kaikki tämän hetkinen tekoäly on heikkoa eikä vahvaa tekoälyä ole vielä pystytty kehittämään. (Hiila, Tukiainen & Hakola, 2019, s. 18; Jääskeläinen, 2019, s. 11; ks. myös Merilehto, 2018, s. 23)



Kuva 4. Tekoäly verrattuna ihmisen älykkyteen (Siukonen & Neittaanmäki, 2019, s. 276)

#### 3.1 Tekoälyn vaikutukset yhteiskuntaamme

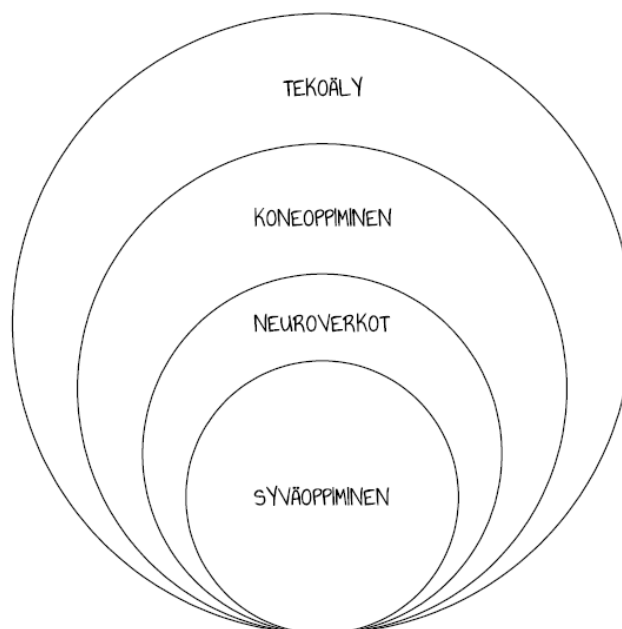
Kukapa ei olisi miettinyt, miten mediassakin paljon esillä ollut tekoäly vaikuttaa omaan työuraan. Ensimmäinen ajatus on helposti kokemus tekoälystä uhkana. Mitä omalle työsuhteelle tapahtuu, kun kone tekee ihmisten työt sekä nopeammin että virheettömämmin? Kuten aiemmin mainittu on tekoälyn tämän hetkinen tilanne kuitenkin vielä kaukana siitä. Historiassa on koettu jo aiemmin useita vastaavia aluksi uhkana koettuja isoja

teollisia mullistuksia, joita on pyritty vastustamaan keinolla millä hyvänsä. Jälkikäteen ajateltuna on voitu kuitenkin huomata muutoksen täysin päinvastaiset vaikutukset, kuten esimerkiksi autoilun salliminen yleisillä teillä Isossa-Britanniassa 1800-luvulla. Tuolloin autoilua vastustaneet hevosvankkurikuskit eivät varmasti voineet kuvitella, kuinka paljon heidän vastustamansa autoilu on myöhemmin tuonutkaan niin työpaikkoja kuin täysin uusia toimialoja. (Hiila ym., 2019, ss. 10 - 11)

Tekoälyyn ja sen yhteiskunnalle tuomiin vaikutuksiin tulisikin pyrkiä suhtautumaan uteliaalla innolla, kauhukuvien maalailun sijaan. Digitalisatiossa on koettu tähän mennessä kolme selkeästi erottuvaa vaihetta. Ensimmäisenä ensimmäisen tietokoneen kehitys, toisena kotitietokoneiden yleistyminen ja kolmantena älylaitteiden yleistyminen sekä erilaisten verkkoalustojen tuomat muutokset omistajuuteen. Tämän kolmannen aallon myötä esimerkiksi aiemmin jokaisen hyllystä löytyneet CD-levyt ovat menettäneet arvostuksensa suoratoistopalveluiden myötä. Neljäs muutosaalto on tuonut mukanaan tekoälyn. (Hiila ym., 2019, ss. 12 - 15)

### 3.2 Koneoppiminen yhtenä tekoälyn mallina

Tekoäly, Artificial Intelligence (AI) koostuu kolmesta erilaisesta mallista kuvan 5 mukaisesti; koneoppiminen, neuroverkot sekä syväoppiminen. Näistä kolmesta koneoppiminen on tällä hetkellä sekä kehittynein että hyödynnetyin. (Hiila ym., 2019, s. 18)



Kuva 5. Tekoälyn mallit (Mattab, 2019)



### 3.3 Koneoppiminen teknisesti

Tieto kätkee sisälleen salaisuuksia erityisesti silloin, kun tietoa on paljon. Isoista tietomääristä voidaan muodostaa erilaisia kaavoja ja malleja. Nämä ovat kuitenkin liian monimutkaisia ihmisen havaittavaksi. Koneoppiminen tutkii tietoa ihmisen puolesta havainnoiden näitä kaavoja tietomäärästä. Pienestä tietomäärästä kaavojen havainnointi voi onnistua ihmiseltäkin, kun taas koneoppiminen ei osaa muodostaa luotettavia kaavoja tietomäärän ollessa liian pieni. Tällöin koneoppimisen muodostama kaava voi olla virheellinen. (Chappell, 2015, s. 3)

Chappell (2015, s. 3) esittelee hyvän esimerkin koneoppimisen muodostamasta virheellisestä kaavasta. Esimerkissä taulukossa 1 on pieni määrä tietoa luottokorttiostoksista. Tämän tiedon perusteella pyritään löytämään kaava, milloin ostokset ovat väärinkäytöksiä ja milloin eivät. Tässä esimerkissä tietoa on niin vähän, että ihminenkin voi yrittää päätellä kaavoja. Selkein kaava tässä tapauksessa olisi, että ostokset ovat väärinkäytöksiä sukunimen alkaessa M-kirjaimella. Tämä ei tietenkään voi pitää paikkaansa. (Chappell, 2015, s. 3)

Taulukko 1. Pieni aineisto luottokorttiostoksia

NIMI	SUMMA	VÄÄRINKÄYTÖS?
LAAKKONEN	2 600,45 EUR	EI
MARTIKAINEN	2 294,58 EUR	KYLLÄ
MIKKOLA	1 003,30 EUR	KYLLÄ
SIPPOLA	8 488,32 EUR	EI

Tiedon määrän lisääntyessä kaavojen luotettavuus paranee. Sen sijaan kyky hahmottaa kaavoja suoraan tiedosta hankaloituu. Tässä koneoppiminen astuu mukaan kuvioon auttamaan kaavojen löytämistä isosta määrästä tietoa. Tämä väärinkäytösten tunnistaminen luottokorttiostoksista on yksi pieni esimerkki siitä, mihin koneoppiminen pystyy. Koneoppimisen avulla voidaan menneiden tapahtumien tutkimisen lisäksi pyrkiä myös ennustamaan tulevaa. (Chappell, 2015, s. 4 - 5)

On hyvä ymmärtää, ettei koneoppiminen kerro suoraan esimerkiksi luottokorttiostosten väärinkäytösten kohdalla onko jokin ostos väärinkäytös vai ei. Koneoppimisen avulla kaavan perusteella saadaan luku nollan ja yhden väliltä. Tämä luku kertoo koneen laskeman todennäköisyyden, onko esimerkiksi luottokorttiostos väärinkäytös vai ei. Luvun ollessa lähellä nollaa, ei ostos todennäköisesti ole väärinkäytös. Luvun lähestyessä yhtä, on todennäköistä, että ostos on väärinkäytös. Koneoppimisen mallin hyödyntämiseksi on tärkeitä mieltä, kuinka mitäkin saatua arvoa käsitellään. Mitä esimerkiksi luottokorttiostosta käsitellään, jos saatu luku on 0,5 eli on yhtä suuri todennäköisyys, että ostos on väärinkäytös kuin että se ei ole.

Onko tärkeämpää estää tuollaisen ostoksen syntyminen, vaikka riskinä on samalla asiakkaan huono kokemus epäonnistuneesta ostosta? Tämän tyyppiset kysymykset ovat osa koneoppimista ja hyvä esimerkki siitä, ettei koneoppiminen tai tekoäly yksin pysty ratkaisemaan kaikkea. (Chappell, 2015, s. 6)

Koneoppimisen avulla kone suoriutuu algoritmien avulla sille annetuista tehtävistä, vaikkei sille olisi kerrottu, kuinka se tapahtuu. Koneoppimisessa kone käsittelee sille syötettyä tietoa ja sen avulla päättelee ja opettelee kuinka toimia. Vaikka koneoppimisen edellytyksenä on suuri tietomäärä, kasvaa koneen älykyys vain tiettyyn pisteeseen saakka. Tiedon on lisäksi oltava laadukasta, joten pelkkä suuri tietomäärä ei riitä. (Hiila ym., 2019, s. 18; ks. myös Merilehto, 2018, s. 26 & s. 195)

Koneoppimisessa oppimiskeinoja ovat ohjattu oppiminen, ohjaamaton oppiminen ja vahvistusoppiminen. Nimensä mukaisesti ohjatussa oppimisessa koneelle kerrotaan oikeat vastaukset. Ohjaamattomassa oppimisessä kone sen sijaan päättelee itse sille annetun datan perusteella. Päätelyssä kone tutkii datan säännönmukaisuuksia sekä suhteita ja toimii sen mukaisesti. Vahvistusoppimisessa koneelle ei kerrota suoraan oikeita vastauksia, mutta annetaan palautetta toiminnasta eri tilanteissa. (Merilehto, 2018, s. 19)

#### 3.4 Netflix-palvelun suositteluominaisuus koneoppimisen esimerkkinä

Käytännöllinen esimerkki koneoppimisen hyödyntämisestä käytännössä on Netflix-suoratoistopalvelun suositteluominaisuus. Netflix tarjoaa kymmeniä tuhansia elokuvia, joten palvelu ei voi näyttää kaikkia elokuvia kerralla. Käyttäjän on mahdotonta etsiä itseään kiinnostavaa sisältöä tällaisesta määrästä sisältöä. Netflixin suosittelu on käyttäjäkohtainen ja sen algoritmit etsivät käyttäjän katsomien ja arvostelemien elokuvien perusteella häntä mahdollisesti kiinnostavaa sisältöä. Netflixin suosittelu kertoo kuvan 6 mukaisesti käyttäjälleen sopivuuden todennäköisyyden prosentteina aiemmin katsottuun sisältöön perustuen. Esimerkissä kuvassa 6 aiemman katseluhistorian perusteella suositellaan 92 prosentin todennäköisyydellä sopivaksi Mad men -sarjaa. Suosittelemisen luotettavuus kasvaa suositteludatan määrän kasvaessa. Netflixin motivaattorina älykkään suosittelun kehitykselle on asiakastyytyväisyyden ja sen myötä kassavirran kasvu. On todennäköisempää säilyttää maksava asiakas palvelun parissa hänen löytäessään jatkuvasti itseään kiinnostavaa sisältöä. (Merilehto, 2018, s. 36; Netflix Suomi, n.d.)



Kuva 6. Netflix-palvelun suosittelu (Netflix Suomi, n.d.)

Tähän esimerkkiin on helppo yhdistää nykyisen tekoälyn tilanteen havainnollistaminen käytännössä. Tekoäly vaatii ihmisen toimiakseen. Netflixin suosittelukaan ei toimisi ilman käyttäjiensä toimia. Katselemalla sisältöä Netflixistä ja erityisesti arvioimalla katsomaansa käyttäjä antaa koneelle tärkeätä dataa suosittelun toimimiseksi. Ilman tätä dataa kone ei voisi suositella juuri kyseiselle käyttäjälle sopivaa sisältöä. (Merilehto, 2018, s. 37)

Tällaisessa oppimisessa havainnollistuu niin sanottu ylisovittamisen riski. Koneen oppiessa opetusdatansa liian hyvin luulee se kaikkien tapausten olevan täysin samanlaisia. Hyvällä koneoppimisen mallilla ylisovittamisen riski on vältettävissä. Tällöin malli oppii soveltamaan oppimaansa dataa yleiseen dataan käytetyn opetusdatan perusteella. (Merilehto, 2018, s. 38)

### 3.5 Tekoölyyn liittyviä uhkia

Kuten ohjelmistorobotiikkaan, myös tekoölyyn liittyvistä uhista yleisin on pelko työpaikkojen katoamisesta. Tekoälyn vaikutuksista työpaikkoihin on esitetty erilaisia arvioita ja poistuvien töiden määräksi on arvioitu 9–47 prosenttia. Vaikutusten arvioiminen on hankalaa, koska kuten robotiikka-kin, tekoäly kasvattaa työn tuottavuutta aiheuttaen lisääntyvän tarpeen teknisten taitojen osaajista. Päätöksenteon siirtyessä osittain ihmiseltä koneelle, on syytä kiinnittää huomiota lisäksi turvallisuuteen sekä eettisiin kysymyksiin. (Siukonen & Neittaanmäki, 2019, s. 261)

Työ- ja elinkeinoministeriön (2018, s. 10) mukaan tekoälyn vaikutukset työelämään vaihtelevat kuitenkin maittain. Sen mukaan esimerkiksi Saksassa vaikutukset ovat Suomea suurempia, perustuen automatisoitavien työpaikkojen arvioituun määrään. Tekoälyn myötä Suomen työpaikoista on arvioitu vuoteen 2030 mennessä poistuvan noin 15 prosenttia. Muutos työn luonteeseen on kuitenkin huomattavasti tätä suurempi. Aiemmin automaation vaikutukset ovat näkyneet matalapalkkaisilla aloilla ja muutoksen myötä on usein siirrytty parempipalkkaisiin töihin. Tekoälyn myötä muutos koskettaa kuitenkin myös korkeasti koulutettuja ja paremmin palkattuja työntekijöitä. (Työ- ja elinkeinoministeriö, 2018, s. 10 - 11)

Aiemmin mainitun työpaikkojen katoamisen pelon lisäksi tekoäly voi yleistyessään ja kehittyessään aiheuttaa todellisia uhkia ja vaaratilanteita. Hyvien ihmisten toimesta tekoälyä pyritään loogisesti käyttämään hyvään. Koska yhteiskunnassamme on ja tulee aina olemaan pahoja ihmisiä, on täysin mahdollista, että nämä pahat ihmiset käyttävät tekoälyä pahaan. Mahdollisesti myöhemmin kehitettävä superälykäs tekoäly voi olla niin tehokas, että saavuttaessaan tavoitteensa ne eivät enää vastaakaan ihmisten sille asettamia tavoitteita. (Merilehto, 2018, ss. 152 - 154)

Uudet teknologiat tuovat mukanaan eettisiä riskejä sekä pohdintaa vastuista. Kuka on vastuussa, jos itsenäisesti toimiva tekoäly aiheuttaa onnettomuuden? Ihmiset toimivat erilaisella älykkyydellä kuin koneet. Tämä on tärkeätä muistaa suunniteltaessa ohjelmistoja ja algoritmeja. Siukonen & Neittaanmäki (2019, s. 263) mainitsevat esimerkissään robottiauton käyttäytymisestä liikennevaloissa. Autolle on määritelty olla ajamatta päin punaista valoa. Tämän perusteella robottiauto saattaakin ratkaisuksi hakeroitua liikennevalojärjestelmään muuttaen valot vihreäksi. Robottiauton mielestä tämä voi olla paras ratkaisu tilanteessa. Ihminen sen sijaan ei toimisi näin. (Siukonen & Neittaanmäki, 2019, ss. 263 - 264)

Ohjelmistot ja algoritmit ovat aina ihmisen tekemiä. Näin ollen niissä voi aina olla virheitä tai ne voivat toimia odottamattomalla tavalla. Ohjelmisto toimii aina niin kuin sen on määritelty toimivan. Aina nämä määrytykset eivät vastaa tavoitetta. Ohjelmistossa saattaa olla puutteita turvallisuudessa, jolloin ohjelmiston muokkaus onnistuu esimerkiksi rikollisten toimesta. Tekoälypohjaiset ohjelmistot toimivat tietyllä algoritmisella tavalla ja tätä on mahdollista hyödyntää tekoälyn harhauttamiseksi. (Siukonen & Neittaanmäki, 2019, ss. 262 - 264)

### 3.6 Microsoft Azure Machine Learning

Markkinoilla on tarjolla erilaisia koneoppimisen ohjelmistoja. Microsoft Azure Machine Learning (myöhemmin Azure ML) on pilvipalveluna toimiva työkalu koneoppimisen hyödyntämiseen. Toimiessaan pilvipalveluna Azure ML voi käsitellä valtavia tietomääriä ja ohjelmiston käyttö onnistuu mistä tahansa. Ohjelmiston käyttö vaatii ainoastaan tietokoneen ja internetiyhteyden. (Chappell, 2015, s. 5)

Azure ML rakentuu muutamista komponenteista. Azure ML Studio on graafinen työkalu prosessien hallintaan. Työkalun avulla sille syötetyn tiedon avulla voidaan prosessoida tietoa, kokeilla erilaisia algoritmeja tietoon sekä testata tuloksena saatua mallia. ML Studion lisäksi Azure ML sisältää erilaisia malleja tiedon esikäsittelyyn, erilaisia algoritmeja sekä rajapinnan, jonka kautta valittu malli voidaan siirtää pilvipalvelu Azureen. (Chappell, 2015, s. 8)

Kuten UiPathissa, myös Azure ML Studiossa erilaisia toimintoja käytetään raahaa-ja-pudota-tyyppisesti graafisessa käyttöliittymässä. Erilaiset toiminnot linkitetään järjestyksessään toisiinsa graafisesti, jonka jälkeen työkalu suorittaa valitut toiminnot. Työkaluun voidaan syöttää esimerkiksi haluttu tieto ja valita halutut käytettävät algoritmit. Tämän jälkeen työkalu etsii ja tarjoaa erilaisia malleja tiedon perusteella. Näistä malleista voidaan valita tarkoitukseen parhaiten soveltuva ja siirtää se tämän jälkeen pilvipalveluun käytettäväksi. (Chappell, 2015, s. 9)

Azure ML on maksullinen työkalu, mutta siitä on saatavilla 30 päivän ilmainen kokeiluversio. Kokeilujakson jälkeen osa toiminnoista on myöhemminkin ilmaisia. Kaikkien ominaisuuksien hyödyntämiseksi Azure ML vaatii maksullisen tilauksen ja hinnoittelu perustuu käytön määrään. (Chappell, 2015, s. 16 - 17)

## 4 OHJELMISTOROBOTIN SUUNNITTELU JA KEHITYS

Johdannossa esimerkkinä mainittu laskun käsittely on tällä hetkellä yksi yleinen ohjelmistorobotiikalla automatisoitu tehtävä. Tämän tyyppisiä ohjelmistorobotteja on jo kehitetty käyttöön. Robotille on kehitysvaiheessa kerrottu tiedossa olevien laskuttajien laskujen rakenne. Tämän perusteella robotti osaa laskuttajan tunnistaessaan etsiä tarvittavat tiedot ennalta määriteltujen sijaintitietojen perusteella. Jos tietoja ei kuitenkaan löydetä oletetuista sijainneista, ohjataan lasku manuaaliseen käsittelyyn. Tässä työssä on tarkoitus rakentaa yksinkertainen laskuja lukeva ohjelmistorobotti sekä tutkia mahdollisuuksia kasvattaa robotin automaatioastetta koneoppimisen avulla.

Ohjelmistorobotin kehitykseen työkaluksi tähän työhön on valittu UiPath ja koneoppimisen työkaluksi Microsoft Azure Machine Learning, jotka ovat toimeksiantajan käytetyimpiä työkaluja. Molemmat ovat alallaan vahvoja toimijoita tarjoten mahdollisuuden ilmaiseen käyttöön. UiPath-ohjelmistosta tarjolla on pysyvästi ilmainen Community-versio ja Azure Machine Learningista 30 päivän ilmainen kokeilujakso, joka todettiin riittäväksi tähän työhön.

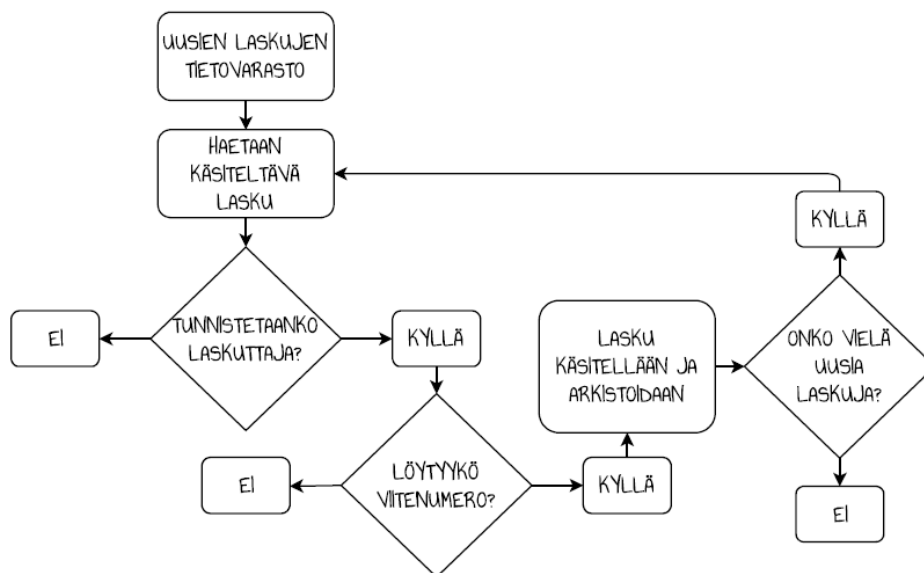
Opinnäytetyön toiminnallinen osuus aloitettiin perehtymällä UiPath-ohjelmistoon Udemyä hankittuun UiPath - Level 1 Robotic Process Automation -kurssiin ja sen harjoitteisiin (Udemy, 2019.b). Kurssilla harjoiteltiin aluksi erilaisia perustoimintoja, jonka jälkeen osaamista syvennettiin eri osa-alueilla lisää. Kurssi antoi hyvät valmiudet ohjelmistorobotin suunnitteluun ja se soveltuu hyvin pientä ohjelmointitaustaa omaavalle. Kurssista on saatavilla perusteet sisältävä edeltävä osa, mikäli ohjelmointitaustaa ennestään ei ole. Level 1 -kurssin jälkeen saatavilla on lisäksi jatko-osa osaamisen syventämiseen. Lisäksi tätä työtä varten käytiin läpi kurssi koneoppimisen työkaluksi valitusta Microsoft Azure Machine Learning -työkalusta (Udemy, 2019.a).

UiPath Academy -oppimisalusta sisältää lukuisia ilmaisia kursseja aihealueittain sekä mahdollisuuden sertifiointiin suorittamiseen. Kurssit on jaoteltu lähtötason perusteella. Academyn kurssit opettavat paljon käytännön vinkkejä ohjelmistorobottien kehitykseen. Udemyn kurssin lisäksi tätä työtä varten osaamista syvennettiin Academyn tarjoamilla kursseilla. Kurseista läpikäytiin Level 1 - Foundation Training sekä AI Computer Vision 2.0. (UiPath Academy, n.d.c; UiPath Academy, n.d.a)

### 4.1 Ohjelmistorobotin suunnittelu

Robotin suunnittelu aloitettiin kuvaamalla prosessi auki karkeasti havainnollistavaan kaavioon kuvassa 7. Koska työssä kehitettävän robotin on tarkoitus olla vain havainnollistava ja tietyn osan suorittava, ei tässä vaiheessa

huomioitu prosessin kaikkia osa-alueita. Robotin on tarkoitus hakea käsiteltävä lasku halutusta tietovarastosta, tässä tapauksessa tiedostokansiossa. Robotti pyrkii tunnistamaan laskuttajan ja tunnistessaan laskuttajan, etsimään laskun viitenumeron. Jos viitenumero löytyy, lasku käsitellään ja arkistoidaan. Tämän jälkeen robotti tutkii, onko käsiteltäviä laskuja lisää. Mikäli kansiossa on vielä laskuja, robotti hakee seuraavan laskun kansiossa käsittelyynsä. Tätä robotti toistaa, kunnes tiedostokansiossa ei ole enää uusia käsiteltäviä laskuja.



Kuva 7. Kaavio laskujen käsittelyn prosessista

## 4.2 Ohjelmistorobotin kehitys

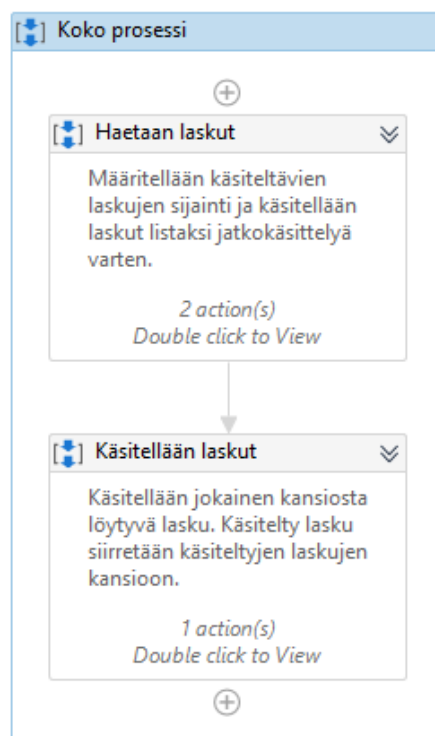
Robotin rakentaminen UiPathissa tapahtuu raahaa-ja-pudota -tyyppisesti (drag and drop). Robotin työvaiheet rakennetaan graafiseen näkymään peräkkäin työnkulkuun (workflow) vuokaaviomaisesti käyttäen valmiita ohjelmiston tarjoamia aktiviteetteja. Robotti voidaan rakentaa kokonaisuudessaan yhden työnkulun sisälle. Isot robottikokonaisuudet voivat rakentua useista toisiinsa linkittyvistä työnkuluista.

Aktiviteetti (activity) määrittelee mitä robotti tekee kussakin työvaiheessa. Samaan tarkoitukseen on usein monia erilaisia aktiviteetteja, jotka eroavat toisistaan esimerkiksi sen suhteen minkälaisen lopputuloksen kyseinen toiminto antaa. Esimerkiksi kuvien vertailua varten on olemassa aktiviteetti, joka laskee kappalemääräisesti, kuinka monta kertaa jokin kuva esiintyy dokumentissa. Toinen aktiviteetti sen sijaan kertoo ainoastaan, löytyykö kuva dokumentista vai ei.

Tietoa voidaan tallentaa ohjelmointikielistä tuttuihin muuttujiin (variables) ja argumentteihin (arguments). Tällöin tallennettua tietoa voidaan myöhemmin muuttaa robotin toiminnan edetessä. Muuttujaan tallennettu

tieto on käytettävissä yksittäisen työnkulun eri aktiviteettien välillä. Argumenttiin tallennettu tieto on käytettävissä eri työnkulkujen välillä. UiPathissa on oletusaktiviteettien lisäksi mahdollista ladata erilaisia paketteja (packages). Paketti sisältää tiettyyn toimintoon tarvittavia aktiviteetteja. Osa paketeista on UiPathin kehittämiä, mutta tarjolla on myös ulkopuolisten kolmansien osapuolien kehittämiä paketteja. Osa paketeista vaatii maksullisia lisenssejä toimiakseen. UiPathin verkkosivuilla paketit on kuvattu selkeästi ohjeineen. Näiden avulla uusien pakettien etsiminen ja käyttäminen onnistuvat, vaikkei aiempaa kokemusta pakettien hakemisesta, asentamisesta tai käyttämisestä olisikaan.

Yksinkertainenkin robotti voi olla hankala hahmotettava kokonaisuutena varsinkin, jos asiaan aiemmin perehtymättömän henkilön on tarkoitus myöhemmin ylläpitää robottia. Usein robotti on aiheellista jakaa useampaan loogiseen osaan, jolloin robotin jatkokehitys on myöhemmin helpompaa. Esimerkiksi kuvassa 8 koko prosessi on jaettu kahteen loogiseen osaan. Esimerkin robotti on hyvin yksinkertainen. Robotille määritellään käsiteltävät laskut, jonka jälkeen robotti käsittelee ne. Tämän jälkeen laskut arkistoidaan.



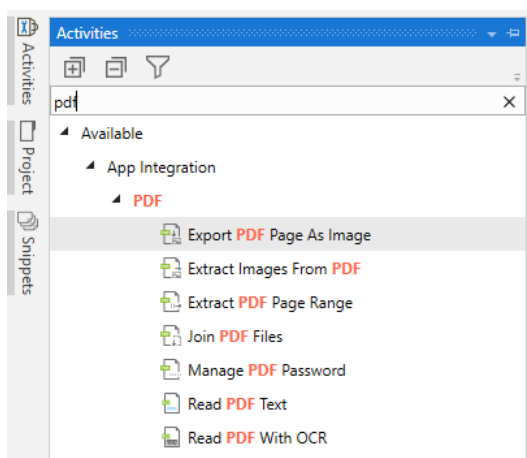
Kuva 8. Kahteen osaan jaettu robotti

Aktiviteetit on hyvä nimetä kuvaavasti. Aktiviteetin nimiä ovat kuvassa 8 "Haetaan laskut" ja "Käsitellään laskut". Lisäksi aktiviteeteille voidaan kirjata kommentti (annotation), jossa voidaan kertoa tarkemmin mitä kyseinen aktiviteetti tekee. Esimerkiksi kuvassa 8 Haetaan laskut -aktiviteetin kommentti on nimen alapuolella vaalean harmaalla tekstillä. Huolellinen



dokumentointi kehitysvaiheessa helpottaa ylläpitoa jatkossa merkittävästi. Aktiviteetin sisältöä päästään tarkemmin tarkastelemaan tuplaklikkaamalla sitä.

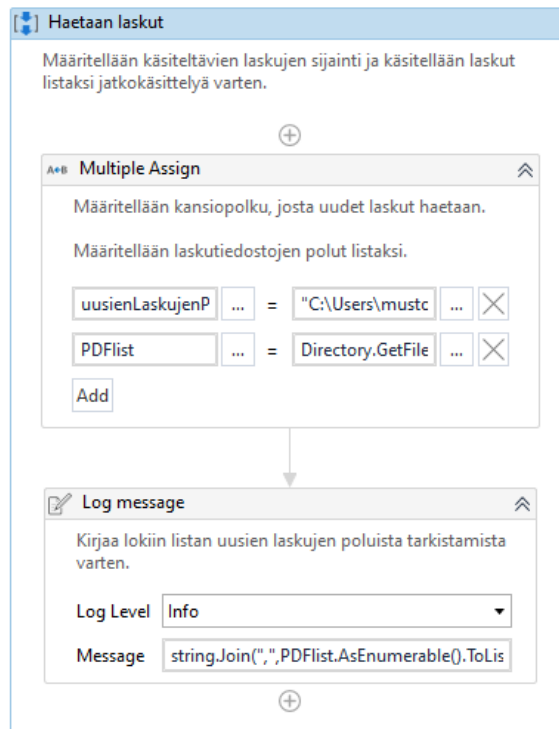
Kun prosessista oli kuvan 7 avulla hahmotettu kehitettävä osuus, aloitettiin robotin kehitys pala kerrallaan alusta alkaen. Ensimmäisenä ladattiin UiPath.PDF.Activities-paketti. Paketteja haetaan ohjelmiston Manage Packages -valikon kautta hakusanojen avulla. Esimerkiksi PDF-dokumenttien käsittelyyn tarvittavat paketit löytyvät hakusanalla ”PDF”. Listalta valitaan haluttu paketti ja se asennetaan Install-painikkeella. Kun tarvittava paketti on asennettu, ovat paketin aktiviteetit valittavissa käyttöön kuvan 9 mukaisesti Activities-valikossa.



Kuva 9. Activities-valikon PDF-dokumenttien aktiviteetit

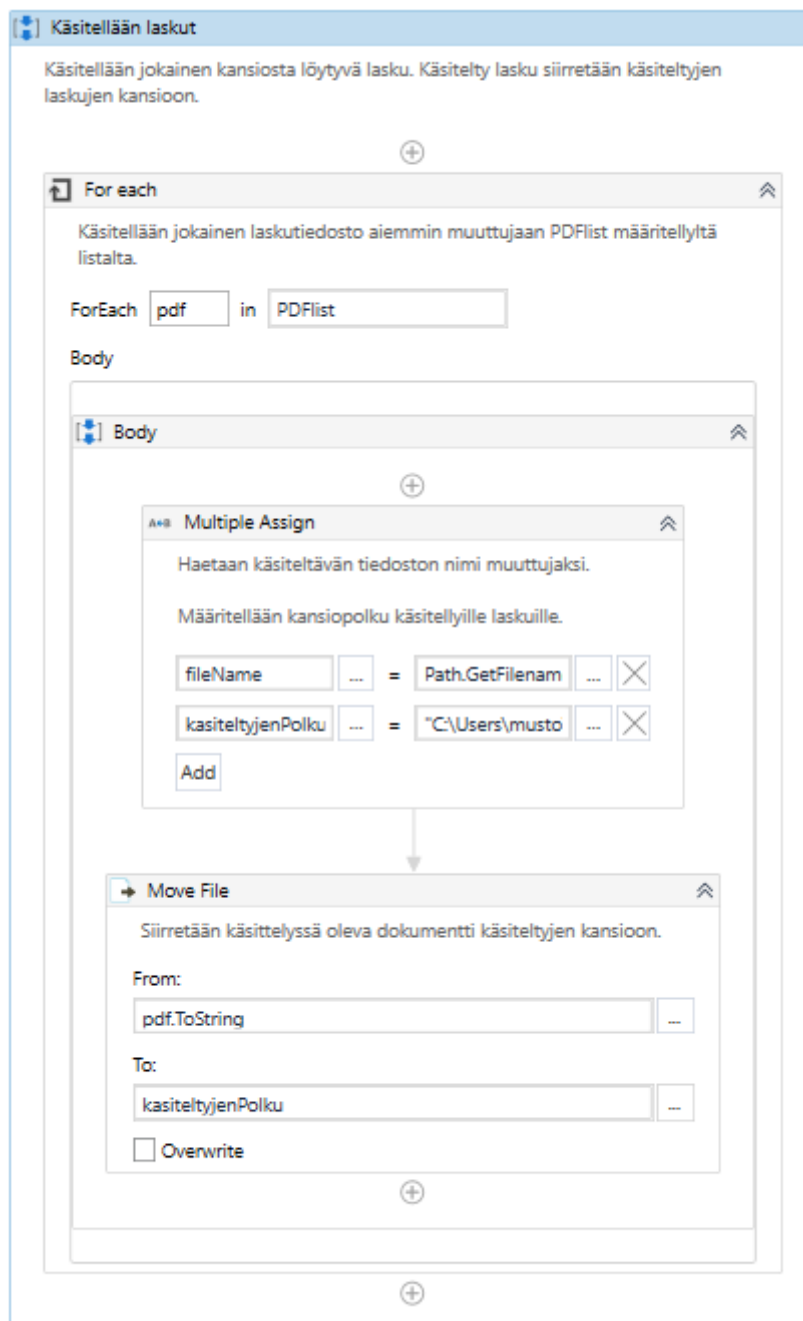
#### 4.2.1 Käsiteltävien tiedostojen määrittely

Robotin rakentaminen aloitettiin luomalla kansio käsiteltäville laskuille. Kansioon tallennettiin esimerkkilaskuja robotin testausta varten. Käytetyt esimerkkilaskut olivat oikeita laskuja muutamilta eri laskuttajilta. Lisäksi haluttiin luoda kansio käsitellyille laskuille. Alkuvaiheessa robotti hakee käsiteltävät laskut ja siirtää ne käsittelyn jälkeen toiseen kansioon. Robotin päänäköymä on esitelty kuvassa 8. Ensimmäinen vaihe, laskukansion määrittely ja tiedostojen hakeminen listaksi, on esitelty kuvassa 10. Tässä vaiheessa robotille kerrotaan uusien laskujen kansiopolku ja se tallennetaan muuttujaan. Kansiosta löytyvät tiedostot haetaan listaksi PDFlist-muuttujaan. Log message -aktiviteetti kirjaa ohjelmiston lokiin PDFlist-muuttujan sisällön. Lokia tarkastellen voidaan tarkistaa, että haettu tieto on oikeassa muodossa.



Kuva 10. Laskujen hakeminen käsittelyyn

Kun käsiteltävät tiedostot on haettu listaksi muuttuinaan, voidaan tiedostoja käsitellä For each -silmukalla. Silmukan sisälle määritellään, mitä jokaiselle käsiteltävälle tiedostolle tehdään. Kuvassa 11 on esitelty kuvassa 8 esitellyn robotin toinen vaihe. Tässä vaiheessa jokainen lasku siirretään käsiteltyjen laskujen kansioon. Multiple Assign -aktiviteetissa määritellään käsiteltävä tiedosto sekä polku, johon lasku siirretään. Move File -aktiviteetille määritellään mistä mihin tiedosto siirretään. Kuvan 11 silmukka toistaa samat toimenpiteet jokaiselle kansiosta löytyneelle laskulle.



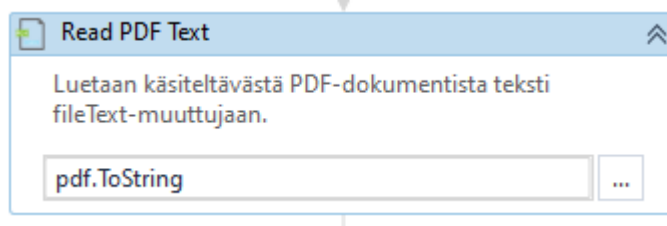
Kuva 11. For each -silmukka

#### 4.2.2 PDF-dokumenttien lukeminen

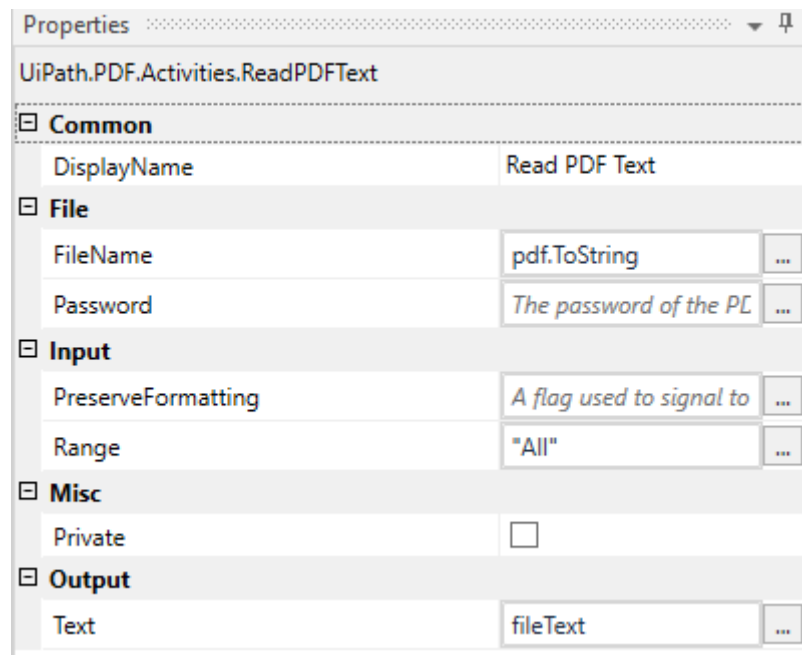
Seuraavaksi testattiin laskujen lukemista lisäämällä aktiviteetteja kuvan 11 silmukan sisälle. PDF-dokumentteja voidaan lukea kahdella tavalla. Dokumentista voidaan lukea pelkkä konekielinen teksti Read PDF Text -aktiviteetilla. PDF-dokumentit eivät kuitenkaan ole välttämättä pelkkää tekstiä ja esimerkiksi tekstiä sisältävää skannattua PDF-dokumenttia luetaan kuin kuvaa. Tällaisen dokumentin lukeminen onnistuu Read PDF With OCR -aktiviteetilla. Aktiviteetti hyödyntää OCR-tekniikkaa eli optista tekstintunnis-

tusta ja lukee tekstiä myös kuvien sisältä. Luettavuuden luotettavuus riippuu muun muassa dokumentin laadusta ja käytetystä fontista. Tällä tekniikalla voidaan lukea myös käsinkirjoitettua tekstiä. OCR-aktiviteetit tarvitsevat toimiakseen OCR-moottorin (OCR engine). Näillä eri tahojen tuottamilla moottoreilla on erilaisia tarkoituksia ja on hyvä perehtyä eri moottoreiden ominaisuuksiin valitessa omaan tarkoitukseen parhaiten toimivaa moottoria.

Kuvan 11 For each -silmukan sisälle, aiempien aktiviteettien väliin, lisättiin aluksi kuvan 12 mukainen Read PDF Text -aktiviteetti. Aktiviteetille määriteltiin käsiteltäväksi tiedostoksi silmukan käsittelyssä oleva PDF-dokumentti. Lisäksi aktiviteetille on määriteltävä kuvan 13 mukaisesti ominaisuuksiin (properties) mitä laskusta luetulle tiedolle tehdään. Aktiviteetin lukema tieto voidaan tallentaa esimerkiksi tekstitiedostoon tai muuttujaan. Ominaisuuksien Output-osion Text-kenttään määritellään tiedon tallentamista varten joko tiedostopolku tai muuttuja. Kuvassa 13 määritellään laskulta luettavan tiedon tallentuvan fileText-nimiseen tekstimuuttujaan.

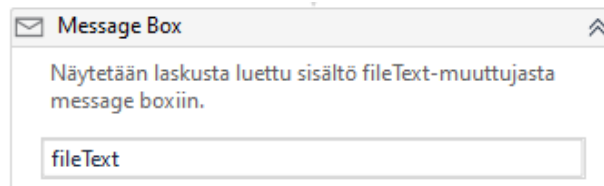


Kuva 12. Read PDF Text -aktiviteetti

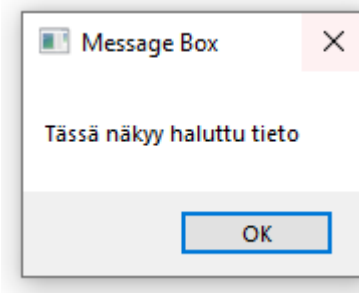


Kuva 13. Read PDF Text -aktiviteetin Properties-osio

Read PDF Text -aktiviteetin lisäksi lisättiin kuvan 14 Message box-aktiiviteetti. Kuvan 10 Log message -aktiviteetin lisäksi Message box -aktiviteetti on käytännöllinen tutkittaessa robotin toimintaa kehitysvaiheessa. Message box -aktiviteetti näyttää kuvan 15 esimerkin mukaisen, halutun tiedon sisältävän ruudun robotin toimiessa. Message boxia käytettäessä robotti etenee toiminnoissaan vasta OK-painikkeen painamisen jälkeen. Tässä tapauksessa Message box -aktiviteetille määriteltiin näytettäväksi tiedoksi laskusta luettu tieto. Tällöin robotin toimiessa voidaan tarkastella kultakin laskulta luettu tieto heti.

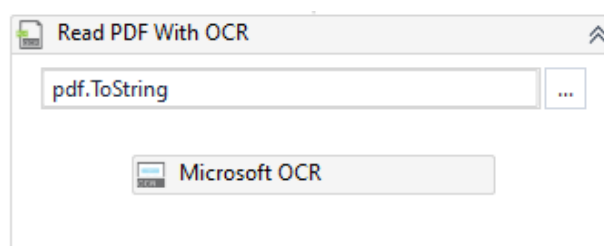


Kuva 14. Message box -aktiviteetti



Kuva 15. Esimerkki Message boxista käytännössä

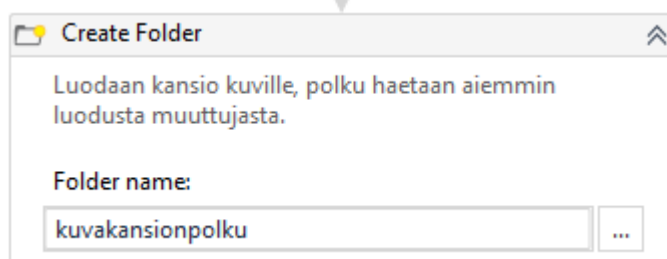
Skannattua PDF-dokumenttia luettaessa pelkällä Read PDF Text -aktiviteetilla robotti onnistui lukemaan vain osan dokumentin tekstistä. Seuraavaksi laskun lukemista testattiin kuvan 16 Read PDF With OCR -aktiviteetilla. Read PDF With OCR -aktiviteettiin on määritelty käsiteltävän dokumentin tiedostopolku sekä valittu käytettäväksi OCR-moottoriksi Microsoft OCR. Moottoreita on mahdollista ladata pakettien mukana aktiviteettien lisäksi. Laskuja luettiin eri moottoreilla ja luotettavin tulos tässä tapauksessa saatiin Microsoft OCR -moottorilla. Ominaisuuksiin määriteltiin jälleen lopputulokseksi tiedon tallentaminen fileText-muuttujaan. Robotti onnistui lukemaan myös skannatun laskun sisällön lähes täysin oikein pieniä virheitä lukuun ottamatta.



Kuva 16. Read PDF With OCR -aktiviteetti

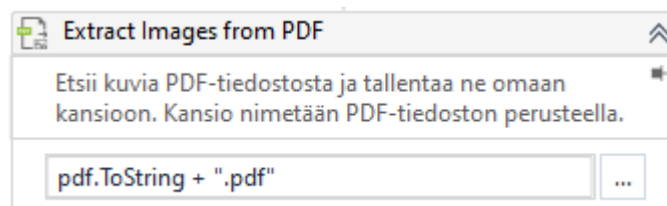
### 4.2.3 Kuvien eriyttäminen ja vertailu

Koska robotin tarkoituksena on pyrkiä tunnistamaan laskun laskuttaja logon perusteella, tutkittiin seuraavaksi PDF-dokumenttien kuvien käsittelyä. Aktiviteetteihin tutustuesssa löydettiin Extract Images from PDF -aktiviteetti, jolla dokumentista voidaan etsiä ja eriyttää kuvia. Laskujen käsittelyyn päätettiin testata tätä aktiviteettia tarkoituksena verrata myöhemmin laskulta löytyneitä kuvia ennalta tiedossa oleviin logoihin. Tässä yhteydessä pohdittiin järkevää tapaa kuvien tallentamiseen jatkossa. Jotta laskun kuvia voidaan verrata logoon tarkoituksenmukaisesti, on tiedettävä mistä laskusta kuvat ovat peräisin. Käsittelyä varten kuvat on tallennettava omaan kansioonsa, jotta niitä voidaan myöhemmin käsitellä silmukan avulla. Käsittely lasku päättyi siirtämään käsiteltyjen laskujen kansioon ja laskulta eriytyvät kuvat tallentamaan laskun tiedostonimen perusteella luotuun kansioon. Tätä varten robottiin lisättiin määrittely kuvien kansio-polusta sekä kuvan 17 Create Folder -aktiviteetti. Create Folder -aktiviteetti luo aiemmin muuttujaksi määriteltyyn polkuun käsiteltävän laskun tiedostonimen mukaisen kansion.



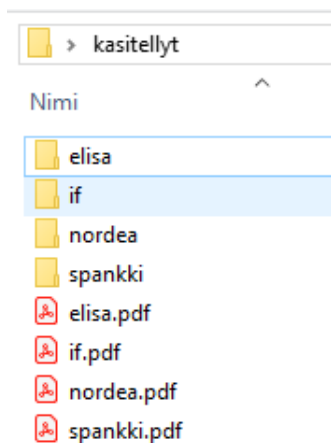
Kuva 17. Create Folder -aktiviteetti

Kun kansio on luotu, eriyttää robotti seuraavaksi kuvat käsiteltävästä dokumentista kuvan 18 aktiviteetilla. Aktiviteetin ominaisuuksiin määriteltiin kuvassa 18 näkyvän käsiteltävän tiedoston lisäksi kansion polku, johon kuvat halutaan tallennettavan.



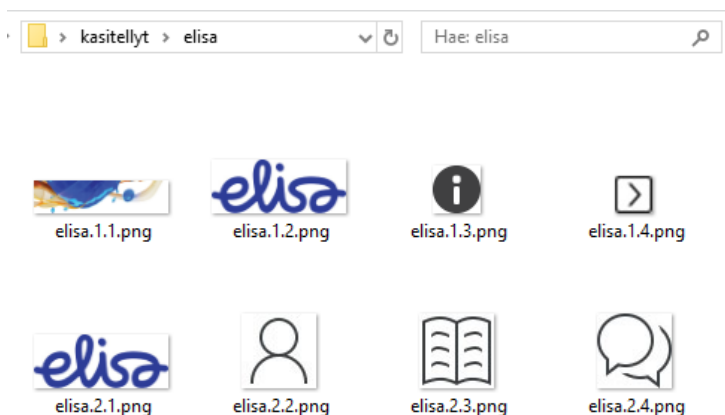
Kuva 18. Extract Images from PDF -aktiviteetti

Näiden aktiviteettien lisäyksen jälkeen robotti loi käsiteltyjen laskujen kansioon kullekin laskulle oman kuvakansion, siirsi laskun käsiteltyjen laskujen kansioon ja erotteli kultakin esimerkkilaskulta löytyneet kuvat omiin kansioihinsa. Kuvassa 19 nähdään neljä käsiteltyä laskua ja niille muodostetut kuvakansiot.



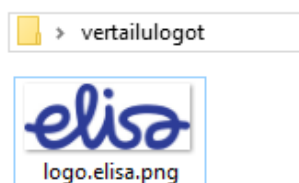
Kuva 19. Käsitellyt laskut ja kuvakansiot

Robotin eräältä esimerkkilaskulta löytämät kuvat (kuva 20) on nimetty alkuperäisen tiedostonimen mukaisesti juoksevalla numeroinnilla. Kuvan nimen ensimmäinen numero kertoo miltä sivulta kuva on löydetty. Pisteiden jälkeinen toinen numero on juokseva järjestysnumero, monesko sivulta löytynyt kuva on kyseessä.



Kuva 20. Laskulta löydetyt kuvat

Koska aktiviteetin avulla laskulta onnistuttiin eriyttämään kyseisen laskun logo, tutkittiin seuraavaksi, kuinka tätä kuvaa voidaan vertailla tiedossa olevaan logoon. Vertailua varten erään laskuttajan logo tallennettiin omaan kansioonsa kuvatiedostona (kuva 21). Vertailtava kuva tallennettiin laskulta löytyneistä kuvista, joten vertailun pitäisi näiden kahden kuvan vertailussa tunnistaa kyseessä olevan täysin sama logo.



Kuva 21. Vertailtava logo

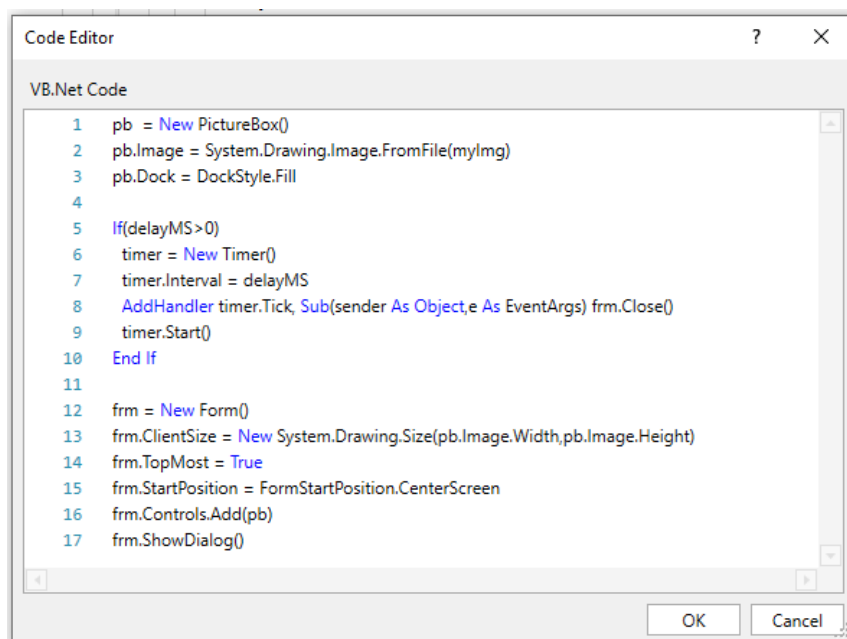
Tähän tarkoitukseen sopivia kuvien käsittelyyn käytettäviä aktiviteetteja ovat esimerkiksi Find Image Matches - tai Image Exists -aktiviteetit. Kuvien vertailuun kokeiltiin molempia aktiviteetteja, mutta vertailua ei onnistuttu rakentamaan toimivaksi. Find Image Matches -aktiviteetti etsii lukumääräisesti, kuinka monta kertaa verrattava kuva esiintyy dokumentissa. Image Exists -aktiviteetti palauttaa tiedon esiintyykö verrattava kuva vertailukohteessa.

Ensin kuvia vertailtiin Image Exists -aktiviteetin avulla, jolloin aktiviteetti palauttaa tiedon täsmääkö kuva verrattavaan logoon. Aktiviteetti palautti virheellisesti jatkuvasti tiedon, ettei kuva täsmää. Molempia aktiviteetteja kokeiltiin myös logon etsimiseen suoraan PDF-dokumentista. Tätäkään ei onnistuttu rakentamaan toimivaksi eikä robotti tunnistanut logoa dokumentista. Tässä vaiheessa löytyi muutamia tarkoitukseen sopivia valmiita paketteja ohjeineen, jotka kuitenkin jouduttiin sulkemaan pois maksullisen lisenssin tarpeen vuoksi.

Kun kuvien vertailua ei saatu toimivaksi, pyrittiin syytä tähän tutkimaan tarkemmin. Ensimmäisenä tarkistettava asiana oli tallennettujen muuttujien sisällön oikeellisuus. Robotin toimintaa on tekstin osalta mahdollista tutkia kehitysvaiheessa aiemmin mainituilla Log message - ja Message box -aktiviteeteilla. Nämä aktiviteetit soveltuvat ainoastaan tekstinkäsittelyyn. Seuraavaksi tutkittiin ja testattiin eri vaihtoehtoja, joilla kulloinkin verrattava kuva saataisiin esille ennen vertailua. Tämän avulla olisi mahdollista tarkistaa kuvan olevan oikeassa muodossa ja lisäksi tarkistaa vertailuaktiviteetin palauttaman vastauksen oikeellisuus.

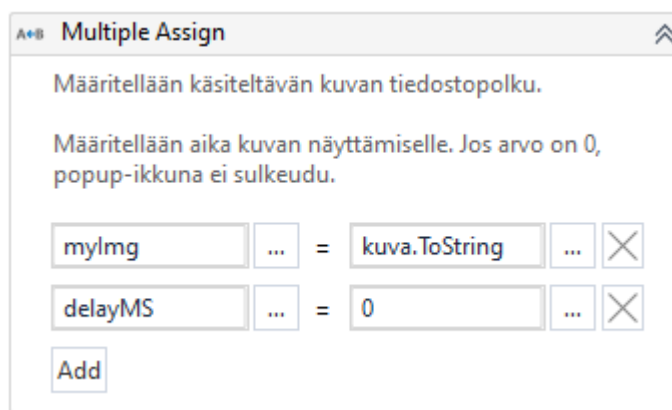
Aiemmin laskulta eriteltyt kuvat haettiin kansioista listaksi käsittelyä varten. Kuvien käsittely toteutettiin jälleen For each -silmukalla yksi kerrallaan. Ennen kuvien vertailua pyrittiin rakentamaan toiminto, joka näyttää käsiteltävän kuvan Message box -tyyppisessä ikkunassa. UiPathin keskustelupalstalta löytyi aiheeseen liittyvä keskustelu, jonka ohjeita sovellettiin tähän tapaukseen. Robotti voi hyödyntää erillistä koodia Invoke code -aktiviteetilla. Aktiviteetin sisälle lisättiin kuvan 22 mukainen koodi.





Kuva 22. Invoke code -aktiviteetin editori

Lisätyn koodin tarvitsemat myImg- ja delayMS-muuttujat määriteltiin lisäksi aiemmin (kuva 23). Käsiteltävä kuva muunnettiin silmukan käsittelyssä olevasta kuvasta. Muuttujalla delayMS määritellään kuvan näkyvyyden aika ja mikäli muuttujan arvo on 0, ei kuva sulkeudu ilman käyttäjän toimia. Tässä tapauksessa arvoksi määriteltiin 0, jotta kuva on näkyvillä, kunnes käyttäjä sulkee kuvaikkunan.



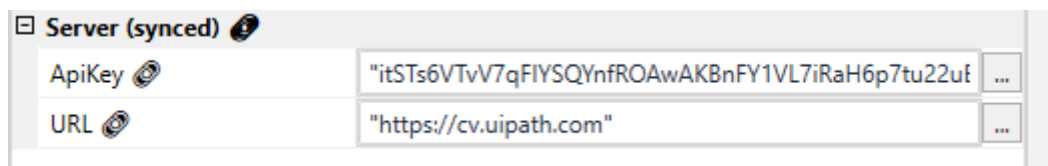
Kuva 23. Koodin tarvitsemien muuttujien määrittely

Kun koodin lisäys saatiin sovellettua robottiin toimivasti, testattiin käsiteltävän kuvan näyttämistä. Käsiteltävä kuva näkyi oikein, joten syy kuvien vertailun epäonnistumiseen rajattiin johtuvan jostain muusta kuin virheellisestä muuttujien sisällöstä. Kuitenkin edelleen kuvan näyttämisen ja vertailun jälkeen kuvia vertaava aktiviteetti palautti False-arvon eli sen mukaan kuvat eivät täsmänneet, vaikka ainakin yksi laskulta löytyneistä kuvista täsmäsi täysin verrattavaan logoon.

Tässä vaiheessa robotin rakentamiseen ja erilaisten vertailuaktiviteettien testaukseen oli kulunut paljon aikaa, joten jouduttiin toteamaan koneoppimisen käytännön osuuden jäävän toissijaiseksi. Jotta robotti saataisiin toimivaksi edes jollain tasolla, päädyttiin perehtymään lisää UiPath Academystä löytyviin ilmaisiin kursseihin.

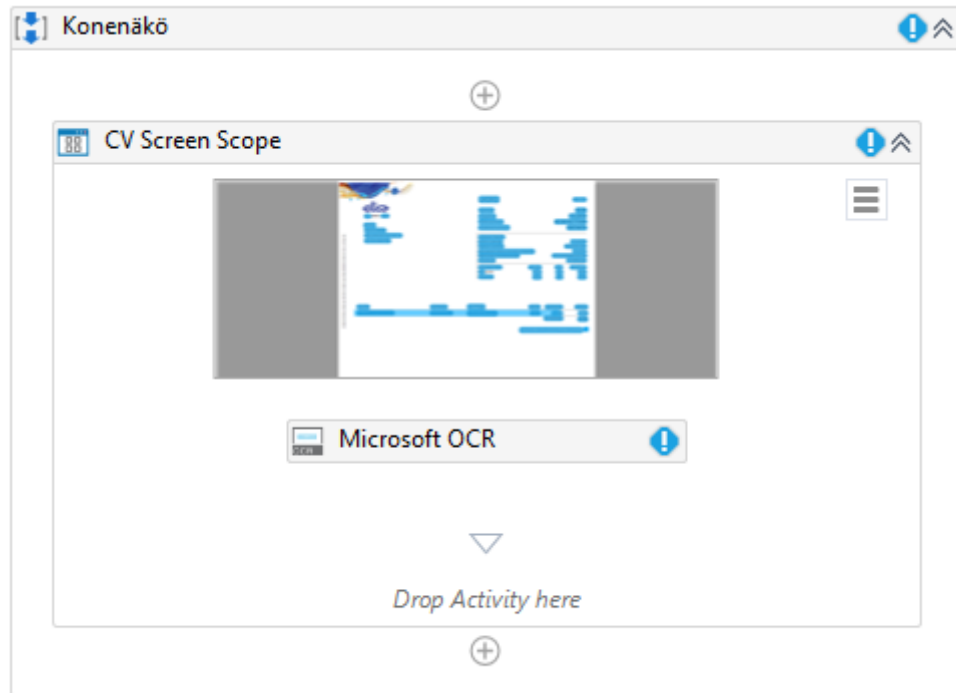
#### 4.2.4 Laskun lukeminen konenäöllä

UiPath Academyn AI Computer Vision -kurssin jälkeen logoa pyrittiin tunnistamaan dokumentista UiPathin oman konenäön avulla. Tätä varten laddattiin UiPath.AI.ComputerVision.Activities -paketti. Tähän mennessä käytetyistä paketeista konenäön aktiviteetit ovat ensimmäisiä, jotka vaativat API-avaimen eli rajapinta-avaimen sekä käytettävän palvelimen verkko-osoitteen. API-avain on saatavilla veloitusetta verkko-osoitteesta cloud.uipath.com löytyvän portaalin Licenses-osiosta. UiPath tarjoaa tähän tarkoitukseen soveltuvan ilmaisen palvelimen osoitteessa <https://cv.uipath.com>. Koneäköä käytettäessä kaikki aktiviteetit kerätään yhden CV Screen Scope -aktiviteetin sisään. Mainitut API-avain ja palvelin määritellään aktiviteetin ominaisuuksiin kuvan 24 mukaisesti.



Kuva 24. API-avaimen ja palvelimen määrittely

Koneäköön tunnistamia elementtejä testattiin kuvan 25 Screen Scope -aktiviteetilla. Aktiviteetin Indicate on screen -ominaisuudella onnistuttiin tunnistamaan PDF-dokumentissa kuvassa 26 sinisellä korostetut elementit. Koska konenäön avullakaan logoa ei onnistuttu tunnistamaan, jatkettiin ratkaisun hakemista edelleen.



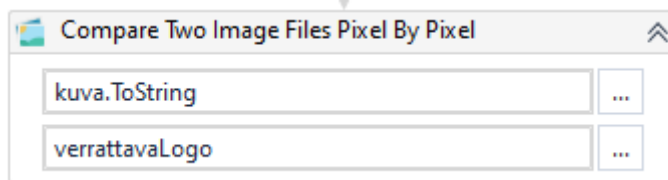
Kuva 25. CV Screen Scope -aktiviteetti



Kuva 26. Konenäön tunnistamat elementit Elisa Oyj:n laskulta

#### 4.2.5 Kuvien vertailu pikseleittäin

Tähän mennessä käytettyjen kurssien, UiPathin ohjeiden tai Community-keskustelupalstan avulla logon tunnistusta ei onnistuttu rakentamaan toimivaksi. Lopulta UiPath Go! -sivustolta löytyi ohjelmistoon ladattava komponentti, jonka kerrottiin vertailevan kahta kuvaa pikseli pikseliltä ja palauttavan tiedon täsmäävätkö kuvat toisiinsa. Testausta varten komponentti ladattiin verkkosivulta tietokoneelle ja asennettiin UiPath-ohjelmistoon. Komponentin mukana asentuneen Compare Two Image Files Pixel by Pixel -aktiviteetin (kuva 27) avulla kuvien vertailu saatiin vihdoinkin toimimaan onnistuneesti ja robotti palautti kuvien täsmätessä odotetun True-arvon.



Kuva 27. Compare Two Image Files Pixel by Pixel -aktiiviteetti

#### 4.2.6 Viitteen paikannus laskulta

Kun kuvien vertailu viimein toimi aivan työn loppuvaiheessa, pyrittiin laskusta vielä jäljellä olevan ajan puitteissa paikallistamaan viitenumero. Viitenumeroa pyrittiin tunnistamaan laskusta muutamien eri aktiiviteetin. Viitenumero ei kuitenkaan missään vaiheessa palautunut aktiiviteeteilta kokonaisuudessaan. Parhaassakin tuloksessa viitenumero palautui ilman viimeistä numeroa. Koska kuvien vertailun onnistunut versio rakentui vasta aivan työn viimeisinä päivinä, ei viitteen hakemiseen ja sen ongelmatilanteiden ratkaisemiseen jäänyt juurikaan aikaa. Nopeiden testien jälkeen viitenumeron paikallistaminen jäi siis kesken.

## 5 TULOKSET

Työssä ei onnistuttu rakentamaan täysin toimivaa ohjelmistorobottia alkuperäisen tavoitteen mukaisesti. Kuvan tunnistuksen ja vertailun eri versioiden hakemisessa ja testaamisessa kului huomattavasti ennakoitua enemmän aikaa, joten tämä luonnollisesti vaikutti myöhempisiin vaiheisiin käytettävissä olevaan aikaan. Koska aika kului ohjelmistorobotin ongelmien ratkaisuun ja kehittämiseen, koneoppimisen käytännön osuus jäi työstä pois kokonaan. Seuraavissa kappaleissa on esitetty työn aikana tehtyjä huomioita sekä ajatuksia jatkokehitystä varten.

### 5.1 Tiedon etsiminen

Ohjelmistorobottia kehittäessä tiedon ja ohjeiden etsiminen eri lähteistä sekä sen soveltaminen omaan tarkoitukseen osoittautui suurimmaksi aikaa vieneeksi osaksi. On osattava etsiä tietoa, jonka olemassa olosta ei ole varmuutta. Robotin kehitys pysähtyi useaksi viikoksi kuvan tunnistamiseen ja vertailuun, sillä vaikka tietoa ja ohjeita etsittiin usein eri hakusanoin eri lähteistä, ei ratkaisua ongelmaan löytynyt. UiPathin keskustelupalstalta löytyi useita täysin vastaavanlaiseen ongelmaan liittyviä keskustelunaloituksia, mutta ei vastauksia. Muutoin keskustelupalsta toimi tärkeänä työkaluna UiPath Go! -sivuston ohella uusien vaihtoehtojen etsimiseen ja kokeilemiseen. Tiedonhakeminen eri lähteistä kehittyi työn toiminnallisen osan aikana huomattavasti, kun aiheeseen tutustuttaessa eri lähteet, niiden sisällöt, käyttäminen ja käytetty sanasto tulivat tutuiksi.

### 5.2 Virhetilanteiden paikantaminen

Ohjelmistorobotin kehityksen yhteydessä erilaisia vikatilanteita ja virheilmoitusten syitä tutkittiin käyttäen Log Message- ja Message box -aktiiviteetteja. Nämä aktiiviteetit ovat erityisen hyödyllisiä missä tahansa ohjelmistorobotin kehityksessä tuoden esille robotin kulloinkin käsittelyssä olevan tiedon. Aktiiviteettien avulla robotin toimintaa on myös helppo seurata lokitiedoista. Varsinkin alkuvaiheessa näitä aktiiviteetteja käytettiin paljon, jonka seurauksena vikatilanteiden ja eri aktiiviteettien riippuvuuksien hahmotus kehittyi työn edetessä. Tämän pohjajymärryksen avulla onnistuttiin rakentamaan soveltaen vastaava toiminne kuvan näyttämiseen. Tämän myötä koodin hyödyntäminen valmiiden aktiiviteettien lisäksi tuli tutuksi.

### 5.3 Logon vertaaminen pikseli pikseliltä

Kuvien vertaaminen onnistui lopulta Compare Two Image Files Pixel by Pixel -aktiiviteetilla, mutta käytännössä tämän aktiiviteetin käyttäminen vastaavanlaiseen robottiin ei ole järkevää. Aktiiviteetti vertaa kuvia ni-

mensä mukaisesti pikseli pikseliltä, joten niiden on vastattava toisiaan täysin. Robotin kehityksessä tätä testattiin muuntamalla verrattavan kuvan kokoa, jonka jälkeen kuvat eivät aktiviteetin mukaan enää täsmänneet.

Koska käsiteltävät laskut ovat usein skannattuja, ei niiden laatu ole riittävä tällaisen aktiviteetin käyttämiseksi. OCR-työkaluissa on usein mahdollisuus määrittellä ominaisuuksiin arvo väliltä 0–1, joka määrittelee toiminnan tarkkuuden. Kuvantunnistus olisikin parasta toteuttaa toimivalla aktiviteetilla, jossa kuvan vastaavuus voidaan määrittellä vastaavalla tavalla. Tällöin tunnistusta voitaisiin testata erilaisilla arvoilla etsien tarkoitukseen sopivin arvo. Esimerkiksi OCR-työkaluissa oletusarvona on yleisesti 0,8, joka on tämänkin työn aktiviteettien kokeilujen perusteella osoittautunut toimivaksi. Kuvantunnistuksessa tämä tarkoittaisi, että kuvien pitäisi täsmätä 80 prosentin tarkkuudella.

#### 5.4 Logon lukeminen konenäöllä

Vertailun vuoksi työn viimeisinä päivinä päädyttiin testaamaan vielä konenäön Screen Scope -tunnistamista kahden muun eri laskuttajan laskuun (kuvat 28 ja 29), joista se tunnisti normaalisti myös logot. Koska näistä laskusta tunnistettiin myös logot, heräsi epäily aiemmin koko ajan esimerkkinä käytetyn Elisa Oyj:n laskun rakenteen ongelmallisuudesta.



Kuva 28. Konenäön tunnistamat elementit Nordea Bank Oyj:n laskulta



Kuva 29. Konenäön tunnistamat elementit S-Pankki Oy:n laskulta

Laskun rakenteen ongelmallisuutta tukee myös viitenumeron tunnistamisen ongelmat. Esimerkkinä käytetyn Elisa Oyj:n laskulta viitettä ei onnistuttu lukemaan kokonaisuudessaan. Kahden muun laskuttajan laskulta vastaavilla aktiviteeteilla viite kuitenkin tunnistui normaalisti kokonaan. Tämä huomio kävi ilmi vasta työn viimeisinä päivinä, joten aikaa muiden

laskujen tutkimiseen ei enää tässä vaiheessa ollut. Vastaavassa kehityksessä jatkossa olisikin syytä alusta alkaen testata robotin toimintaa useilla erilaisilla testilaskuilla, erityisesti ongelmatilanteiden ilmetessä.

### 5.5 Koneoppimisen hyödyntäminen ohjelmistorobotiikassa

Koneoppimisen hyödyntämistä ohjelmistorobotiikkaan robotin opettamiseksi ei ehditty tutkimaan tai testaamaan käytännössä. Koneoppimisen osalta käytännön osuus jäi kurssien yhteydessä suoritettuihin harjoitukseen. UiPathissa tietoa voidaan hakea robotin käyttöön erilaisista tiedostoista. Näiden tiedostojen avulla robotille voidaan kertoa esimerkiksi viitenumeron sijainti laskulla. Koneoppimisen yhdistämistä ohjelmistorobotiikkaan olisi hyvä jatkaa tutkien mahdollisuuksia muokata näitä tiedostoja koneoppimisen avulla. Tämä vaatisi tarkemmin tietoa minkälaista tietoa koneoppimisen avulla voidaan tuottaa ja miten tätä tietoa voitaisiin hyödyntää robotin opettamiseksi.

## 6 YHTEENVETO

Koko opintojeni tavoitteena oli sekä alan vaihto, että työnantajan vaihto, joten omana opinnäytetyön tavoitteenani oli kasvattaa omaa osaamistani mahdollisimman paljon työllistymistä ja työelämää ajatellen. Etsiessäni aiheita opinnäytetyölleni, tavoitteenani oli löytää työelämälähtöinen aihe. Koska ohjelmistorobotiikka oli aiheena kiinnostava niin opintojen kuin tulevan työllistymisenkin kannalta, valikoitui aihe tämän perusteella. Vaikka koneoppiminen ei työtä aloittaessa ollutkaan tuttua, koin sen ajankohdattaiseksi ja tutustumisen arvoiseksi.

Aihevalinta tähän työhön osoittautui mielenkiintoiseksi, opettavaiseksi ja todella ajankohtaiseksi, mutta teknisen toteutuksen osalta liian laajaksi. Työtä aloittaessani aiheista ohjelmistorobotiikka oli käytännössä tuttua yhdestä aiemmasta kurssista. Koneoppimisesta pohjatiedot olivat lähes mitättömät. UiPathin aiemman kokemukseni perusteella oli olettamus, että yksinkertaisen ohjelmistorobotin kehitys työn puitteissa olisi mahdollista. PDF-tiedostojen käsittely oli kuitenkin täysin uutta ja osoittautui käytännössä huomattavasti oletettua haastavammaksi.

Teoriaosaa varten materiaaliin tutustuminen ja teorian kirjoittaminen kasvattivat ymmärrystäni molemmista aiheista tarkemmin ja ymmärsin sen pohjalta myös koneoppimisen pääpiirteet. Lähdemateriaalia etsiessäni pyrin etsimään ajankohtaista, luotettavaa, monipuolista ja mielenkiintoista tietoa. Koska ohjelmistorobotiikka on teknologiana uusi, oli siitä saatavilla oleva materiaali pääosin sähköisessä muodossa. Aiheesta löytyi kuitenkin tästä huolimatta paljon luotettavaa ja käytännönläheistä materiaalia. Omaan oppimiseeni peilaten onnistuin tavoitteessani teorian osalta mielestäni hyvin.

Kokonaisuutena omat tavoitteeni opinnäytetyölle täyttyivät, vaikka työhön aluksi tavoitteeksi asetetut ohjelmistorobotin kehitys sekä tähän koneoppimisen yhdistämisen mahdollisuuksien tutkiminen jäivätkin kesken. Opinnäytetyön tekeminen opetti todella paljon työelämässä ohjelmistorobotin kehitykseen tarvittavia taitoja. Työn edetessä huomasin, että vaikka aiheesta on saatavilla paljon materiaalia ja ohjeita, ei vaikeimpiin kysymyksiin ole välttämättä saatavilla ratkaisuja julkisesti. Ohjelmistorobotiikkaa on kuvattu mahdollisuutena automatisoida prosesseja ilman syvällisiä ohjelmointitaitoja. Työ kuitenkin auttoi ymmärtämään ohjelmoinnin perusteiden ymmärryksen etua ja tärkeyttä ohjelmistorobottikehityksessä. Ohjelmistorobotiikan käytännön taitojen sain lisäksi monipuolisen peruskäsitöksen tekoälystä sekä koneoppimisesta.



## LÄHTEET

AIMDek Technologies. (2018). Evolution of Robotic Process Automation (RPA): The Path to Cognitive RPA. Haettu 23.10.2019 osoitteesta <https://medium.com/@AIMDekTech/evolution-of-robotic-process-automation-the-path-to-cognitive-rpa-c3bd52c8b865>

Automation Anywhere. (n.d.). #1 Platform for attended or unattended RPA. Haettu 22.10.2019 osoitteesta <https://www.automationanywhere.com/solutions/attended-vs-unattended-rpa>

Brandall, B. (2018). Robotic Process Automation and the Death of Data Entry. Haettu 24.10.2019 osoitteesta <https://www.process.st/robotic-process-automation/>

CGI Suomi Oy. (n.d.). CGI yrityksenä. Haettu 2.9.2019 osoitteesta <https://www.cgi.fi/fi/cgi-yrityksena>

CGI Suomi Oy. (2016). Tuomiopäivä toimistorutiineille – Ohjelmistorobotiikan lyhyt oppimäärä. Haettu 25.10.2019 osoitteesta <https://www.cgi.fi/fi/blogi/tuomiopaiva-toimistorutiineille-ohjelmistorobotiikan-lyhyt-oppimaara>

Chappel, D. (2015). Introducing Azure Machine Learning. Haettu 21.10.2019 osoitteesta [https://download.microsoft.com/download/3/B/9/3B9FBA69-8AAD-4707-830F-6C70A545C389/Introducing\\_Azure\\_Machine\\_Learning.pdf](https://download.microsoft.com/download/3/B/9/3B9FBA69-8AAD-4707-830F-6C70A545C389/Introducing_Azure_Machine_Learning.pdf)

Docparser. (n.d.) What Is OCR And What Is It Used For? Haettu 27.9.2019 osoitteesta <https://docparser.com/blog/what-is-ocr/>

edureka! (2019). RPA Tools List and Comparison - Leaders in RPA Software. Haettu 16.9.2019 osoitteesta <https://www.edureka.co/blog/rpa-tools-list-and-comparison/>

Elisa Oyj. (2019). Ohjelmistorobotiikka - oikotie keinoälyn hyötyihin? Haettu 25.10.2019 osoitteesta <https://hub.elisa.fi/ohjelmistorobotiikka-oikotie-keinoalyn-hyotyyihin/>

Hiila, I., Tukiainen, M. & Hakola, I. (2019). *Tiimiäly - opas muuttuvaan työelämään*. Jyväskylä: Tuuma-kustannus.

Jääskeläinen, A. (2019). *Mitä tapahtuu huomenna, kun tekoäly poistaa järjettömyydet?* Helsinki: WSOY.

- Kaarlejärvi, S. (2017). RPA – robotiikalla parempaan arkeen. Haettu 24.10.2019 osoitteesta <https://www.efima.com/blogi/rpa-robotiikalla-parempaan-arkeen/>
- Kameswara Rao, T., Yashwanth Chowdary, K., Koushik Chowdary, I., Prasanna Kumar, K. & Ramesh, C. (2019). Optical Character Recognition from Printed Text Images. Haettu 11.10.2019 osoitteesta <https://doi.org/10.32628/CSEIT1952175>
- Knowit. (2018). Ohjelmistorobotiikkaa Robot Framework -teknologialla - tunnetko avoimen lähdekoodin edut? Haettu 4.9.2019 osoitteesta <https://we.knowit.fi/solutions-fi/ohjelmistorobotiikkaa-robot-framework-teknologialla-tunnetko-avoimen-lahdekoodin-edut>
- Kääriäinen, J., Aihkisalo, T., Halén, M., Holmström, H., Jurmu, P., Matinmikko, T., Seppälä, T., Tihinen, M. & Tirronen, J. (2018). Ohjelmistorobotiikka ja tekoäly - soveltamisen askelmerkkejä. Haettu 4.9.2019 osoitteesta <http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/161123/65-2018-Ohjelmistorobotiikka%20ja%20tekoaly.pdf>
- Laitila, T. (2018). Suomalaiset rakentavat ohjelmistorobottia - avoin koodi haastaa kilpailijat. Haettu 4.9.2019 osoitteesta <https://www.tivi.fi/uutiset/suomalaiset-rakentavat-ohjelmistorobottia-avoin-koodi-haastaa-kilpailijat/ca2f6c13-45eb-3dd2-ba5e-7988e5d65483>
- Mattab. (2019). Artificial Intelligence, Enough of the hype! What is it?. Haettu 22.10.2019 osoitteesta <https://community.hpe.com/t5/hpe-blog-uk-ireland/Artificial-Intelligence-Enough-of-the-hype-What-is-it/bap/7046672#.Xa8VI0YzaUI>
- Merilehto, A. (2018). *Tekoäly: matkaopas johtajalle*. Helsinki: Alma Talent 2018.
- Månsson, D. (2017). Ohjelmistorobotiikkaa käytännönläheisesti - mistä oikeasti on kysymys? Haettu 4.9.2019 osoitteesta <https://www.azets.fi/blogi/ohjelmistorobotiikkaa-kaytannonlaheisesti/>
- Netflix Suomi. (n.d.). Mad Men. Haettu 1.11.2019 osoitteesta <https://www.netflix.com/fi/title/70136135>
- Omnisys. (2019). RPA - Comparing UiPath, Automation Anywhere & Blue Prism. Haettu 18.9.2019 osoitteesta <http://www.omnisysolutions.com/rpa-comparing-ui-path-automation-anywhere-blueprism/>
- Robot Framework Foundation. (n.d.). ROBOT FRAME WORK/ >>RPA. Haettu 4.9.2019 osoitteesta <https://robotframework.org/rpa/>

RPA Training. (n.d.). Blue Prism vs Automation Anywhere vs UiPath. Haettu 18.9.2019 osoitteesta <https://www.rpatraining.co.in/blue-prism-vs-automation-anywhere-vs-uipath/>

Siukonen, T. & Neittaanmäki, P. (2019). *Mitä tulisi tietää tekoälystä*. Jyväskylä: Docendo Oy.

Soumik, R. (2019). RPA is now the world's fastest growing enterprise software category. Haettu 2.9.2019 osoitteesta <https://techwireasia.com/2019/07/rpa-is-now-the-worlds-fastest-growing-enterprise-software-category/>

Techopedia. (n.d.). Optical Character Recognition (OCR). Haettu 27.9.2019 osoitteesta <https://www.techopedia.com/definition/31623/optical-character-recognition-ocr>

Tripathi, A. M. (2018). *Learning Robotic Process Automation : Create Software Robots and Automate Business Processes with the Leading RPA Tool - UiPath*. Haettu 26.8.2019. Ebook Central -tietokanta.

Työ- ja elinkeinoministeriö. (2018). Tekoälyajan työ. Haettu 25.10.2019 osoitteesta [http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/160931/19\\_18\\_TEM\\_Tekoalyajan\\_tyo\\_WEB.pdf?sequence=1&isAllowed=y](http://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/160931/19_18_TEM_Tekoalyajan_tyo_WEB.pdf?sequence=1&isAllowed=y)

Udemy. (2019.a). A-Z Machine Learning using Azure Machine Learning (AzureML). Haettu 20.10.2019 osoitteesta <https://www.udemy.com/course/machine-learning-using-azureml/>

Udemy. (2019.b). UiPath - Level 1 Robotic Process Automation. Haettu 3.10.2019 osoitteesta <https://www.udemy.com/course/uipath-robotic-process-automation/>

UiPath. (n.d.a). Robotic Process Automation (RPA) Haettu 2.9.2019 osoitteesta <https://www.uipath.com/rpa/robotic-process-automation>

UiPath. (n.d.b). UiPath Enterprise RPA Platform. Haettu 16.9.2019 osoitteesta <https://www.uipath.com/product/platform>

UiPath Academy. (n.d.a). AI Computer Vision 2.0. Haettu 12.11.2019 osoitteesta <https://academy.uipath.com/learn/course/external/view/elearning/639/AIComputerVision20>

UiPath Academy. (n.d.b). Browse UiPath's Courses & Learning Plans. Haettu 14.11.2019 osoitteesta <https://academy.uipath.com/learn>

UiPath Academy. (n.d.c). Level 1 - Foundation Training - 2018.3. Haettu 18.9.2019 osoitteesta <https://academy.uipath.com/learn/course/external/view/elearning/511/Level1-FoundationTraining-20183>

UiPath Forum. (n.d.). Ui Forum. Haettu 14.11.2019 osoitteesta <https://forum.uipath.com/>

UiPath Go!. (n.d.). UiPath Go! - Automation made easy! Haettu 14.11.2019 osoitteesta <https://go.uipath.com/>