



Markus Heikkinen

Konenäön hyödyntäminen automaatioprosessissa ja kuvien tallennus ulkoiselle palvelimelle

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

27.9.2021

Tiivistelmä

Tekijä:	Markus Heikkinen
Otsikko:	Konenäön hyödyntäminen automaatioprosessissa ja kuvien tallennus ulkoiselle palvelimelle
Sivumäärä:	23 sivua + 1 liite
Aika:	27.9.2021
Tutkinto:	Insinööri (AMK)
Tutkinto-ohjelma:	Sähkö- ja automaatiotekniikka
Suuntautumisvaihtoehto:	Automaatiotekniikka
Ohjaajat:	Lehtori Reijo Leinonen Toimitusjohtaja Markku Lakka

Opinnäytetyön tarkoituksena oli tutkia konenäkökameran hyödyntämistä pakkaustavara-automaatioprosessissa sekä rakentaa automaattinen kuvansiirto konenäkökameralta ulkoiselle palvelimelle.

Opinnäytetyö tehtiin yhteistyössä Laktek Engineering Oy:n kanssa. Tutkimuksessa käytettiin yrityksen ABB:n IRB 14050 -robottia, johon oli integroituna Cognex-konenäkökamera. Kirjallisina lähteinä oli robotiikkaan, konenäköön ja tiedonsiirtoon liittyvää materiaalia

Opinnäytetyössä tehtiin lastausta valvova konenäkösovellus ja pohdittiin, mitä lisäarvoa konenäkö voisi tuoda erilaisiin automaatioprosesseihin.

Opinnäytetyön perusteella kävi selväksi, että konenäköä ja data-analysointia voidaan hyvin käyttää apuna muuan muassa kunnossapidossa ja laadunvalvonnassa. Kaiken kaikkiaan konenäköä voidaan hyödyntää yhä enemmän kaikissa automaatiotekniikan prosesseissa.

Avainsanat: konenäkö, tiedonsiirto, analysointi

Abstract

Author: Markus Heikkinen
Title: Utilization of machine vision in the automation process and copying images to an external server
Number of Pages: 23 pages + 1 appendix
Date: 27 September 2021

Degree: Bachelor of Engineering
Degree Programme: Electrical and Automation Engineering
Specialisation option: Automation Engineering
Instructors: Reijo Leinonen, Principal Lecturer
Markku Lakka, CEO

The purpose of the thesis was to study the utilization of a machine vision camera in the packaging automation process and to build an automatic image transfer from a machine vision camera to an external server.

The thesis was carried out in co-operation with Laktek Engineering Oy. Company's ABB IRB 14050 robot was used in this study with integrated machine vision camera. The written sources used were material related to robotics, machine vision and data transmission.

Loading application was made as a practical example and was considered what value machine vision could bring to automation processes.

Based on the thesis it became clear that machine vision and analysis of data can be used to help maintenance and quality control. Overall machine vision can be increasingly utilized in all automation processes.

Keywords: machine vision, data transfer, analysis

Sisällys

1	Johdanto	1
2	Konenäkö	2
2.1	Konenäön määritelmä	2
2.2	Konenäköjärjestelmän tunnistusmetodeja	3
2.3	Sovelluksia	4
2.4	Konenäköjärjestelmän hyödyntäminen automaatioprosessissa	5
3	Tiedonsiirto	7
4	Lastauksen valvonta konenäkökameralla ja kuvien tallennus palvelimelle	9
4.1	Setupin rakentaminen	9
4.2	Konenäköjärjestelmän konfigurointi	10
4.2.1	FTP-palvelimen tekeminen	10
4.2.2	Yhteyksien konfigurointi	11
4.2.3	Konenäkökameran kuvantunnistustyökalun konfigurointi	13
4.2.4	Esteet ja haasteet	16
4.3	Testaus	18
4.4	Johtopäätökset	20
5	Yhteenveto	23
	Lähteet	24
	Liitteet	26
	Liite 1: FTP-palvelimen luomisen manuaali Windows 10:llä	26

1 Johdanto

Opinnäytetyöhön liittyvän projektin tarkoituksena on suunnitella Laktek Oy:n pääasiallisen sähkö- ja automaatio-suunnittelun rinnalle uusi lisäpalvelu, jossa konenäkö tunnistaa, käsittelee ja siirtää kuvat ulkoiselle palvelimelle. Opinnäytetyö tehdään ABB:n IRB14050 -robotilla, johon on integroitu Cognexin musta-vaikoinen konenäkökamera, imukuppitarttuja ja sivutarttuja. Kameran ohjelmistona käytettiin In-Sight Exploreria.

Projektissa konenäköjärjestelmä valvoo elintarvikepurkkien lastausta lavalle. Simuloinnin helpottamiseksi elintarvikepurkit on korvattu kuutioilla. Tarkoituksena on suunnitella toimiva kokonaisuus kustannustehokkaasti käyttäen mahdollisimman vähän valmiita ratkaisuja. Pelkistetyesti voidaan sanoa, että työssä käytetään vain tietokonetta ja konenäkökameraa, joiden pohjalta pyritään tekemään toimiva kokonaisuus ilman ostettuja lisäpalveluja.

Opinnäytetyön alussa käydään läpi konenäön määritelmän ohella konenäköjärjestelmien erilaisia tunnistusmetodeja, sovelluksia, joissa konenäkö on keskeisessä roolissa sekä pohditaan yleisesti konenäön hyödyntämistä automaatiojärjestelmissä. Kolmannessa kappaleessa käsitellään yleisesti tiedonsiirtoa sekä keskitytään erityisesti tässä projektissa käytettävään FTP (File Transfer Protocol) -tiedonsiirtomenetelmään.

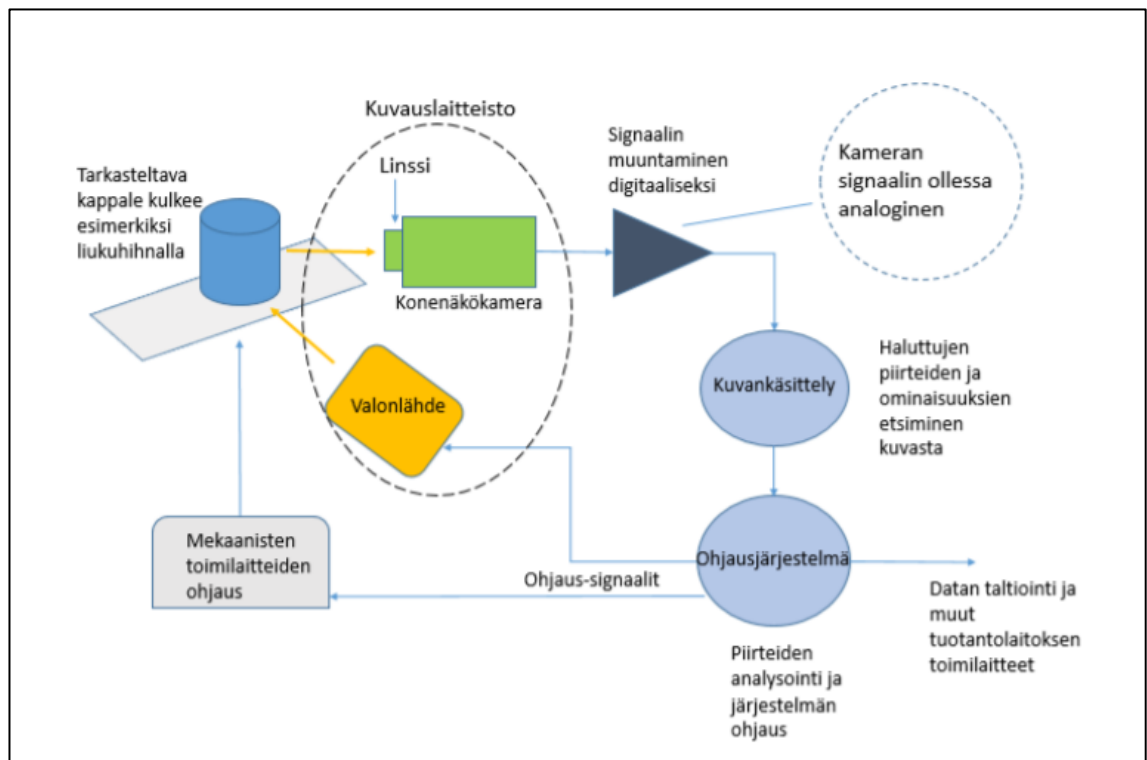
Tämän jälkeen opinnäytetyössä kuvataan projektiin liittyvä esimerkitapaus, jossa konenäkö toimii osana automaatioprosessia. Tämä tapahtuu kameran ohjelmiston kuvantunnistustyökalun avulla, joka vertaa asetettuja raja-arvoja kuvasta saatuihin arvoihin. Lopuksi pohditaan konenäön tuomaa lisäarvoa esimerkitapaukseen sekä yleisesti konenäön merkitystä automaatioprosesseissa nyt ja tulevaisuudessa.

2 Konenäkö

2.1 Konenäön määritelmä

Määritelmät siitä, mitä konenäöllä tarkoitetaan vaihtelevat, mutta yleisesti ottaen on kyse menetelmästä ja teknologiasta, jolla kuvasta poimitaan tietoa. Yksinkertaisimmillaan konenäköä voidaan käyttää ns. hyväksytyt kuva / hylätty kuva -erottelulla, jolloin konenäön kuvantunnistukselle on asetettu jokin raja-arvo, jonka perusteella se joko hyväksyy tai hylkää kuvan. Esimerkiksi raja-arvoksi on voitu asettaa kuuden lasipurkin tunnistaminen ja mikäli konenäkö havaitsee, että purkkeja onkin lavalla jokin muu määrä kuin kuusi, kuva hylätään.

Monimutkaisemmat sovellukset voivat tunnistaa kuvasta kappaleiden muotoja, suuntia, hahmoja ja kasvoja (1;2, s. 5; 3, s. 1.)



Kuva 1. Konenäköjärjestelmän prosessikaavio (4).

Kuvassa 1 on esitetty konenäköjärjestelmän prosessikaavio, joka sisältää yleensä kameran, mahdollisimman tarkkaan vakioidun valaistuksen, tietokoneen tiedon poimintaan ja kuvan prosessointiin sekä digitaaliset ohjaussignaalit liitettyinä automaatioprosessiin.

Valaistuksen mahdollisimman tarkka vakiointi on tärkein prosessin ulkopuolinen osatekijä konenäköjärjestelmän käytössä. Teollisessa käytössä on edelleen hyvin paljon mustavalkokameroita rajoitetulla erottelukyvyllä, jotka voivat valaistuksen vaihdellessa antaa virheellistä tietoa ja sillä tavoin vaikuttaa koko prosessiin sujuvuuteen.

Teknologian ja tietokoneiden laskentatehon kehittymisen myötä kamera voi suorittaa itsenäisesti laskennat eikä tietokonetta välttämättä tarvita.

2.2 Konenäköjärjestelmän tunnistusmetodeja

Kuvankäsittelymetodeja on lukuisia, modernit konenäköjärjestelmät yhdistelevät näitä algoritmeja luotettavuuden lisäämiseksi.

Mittaaminen on perustoiminto, jolla varmistetaan, että kappale on halutun kokoinen. Riippuen kappaleesta mittausheitto saa olla sadasosamillimetrejä esimerkiksi mikroprosessorit, kun taas puunkäsittelylaitoksella voi riittää senttimetrien tarkkuus. (5.)

Pikselien laskenta on konenäkösovelluksissa yksi yleisimmin käytettävistä menetelmistä. Kuvasta lasketaan joko tummia tai vaaleita pikseleitä tietyltä alueelta ja tehdään vertailu oletusarvoon, jonka perusteella kappale hylätään tai hyväksytään. Kappaleen ollessa hyväksytyn rajoissa prosessi jatkuu normaalisti ja hylätty kappale aiheuttaa minimissään hälytyksen tai prosessiin pysäyttämisen.

(5.)

Kappaleen reunantunnistuksella konenäköjärjestelmä reagoi suureen valon intensiteetin muutokseen ja tekee oletuksen, että kappaleen reuna tai ääriviiva on

juuri siinä paikassa. Tähän algoritmiin on yleensä yhdistetty myös esimerkiksi mittaus, jolla varmistetaan, että kappale on sallittujen arvojen sisällä. (5.)

Kuviontunnistuksessa kuvasta etsitään ennakoita määritetyiltä alueelta kappaleita, jotka ovat riittävän lähellä oletukseksi valittua kappaleita. Tähän voidaan lisätä myös samankaltaisten kappaleiden laskenta, jolloin saadaan varsinkin isoissa tuotantoerissä hyväksytyjen ja hylättyjen kappaleiden prosenttiosuudet hyvinkin tarkasti tilastoitua. (5.)

Erilaiset kaksiulotteisten tekstien, viivakoodien tai merkkijonojen tunnistamiseen tarkoitettut konenäköjärjestelmät vertaavat kuvasta saatua dataa mallikappaleeseen ja tulos on yksiselitteisesti hylätty tai hyväksytty. (5.)

3D-kamerat pystyvät tarkastelemaan kappaleen pintoja ja muotoja myös liikkeessä olevasta kappaleesta tehden prosessista entistä nopeampaa ja luotettavampaa. (6.)

Erilaiset lääketieteelliset kuvaukset mm. röntgenkuvaus, magneettikuvaus, PET-kuvaus ja varjoainekuvaukset hyödyntävät osittain konenäköjärjestelmiä ja niiden laskentamalleja.

Viimeisimmät konenäköjärjestelmät pystyvät tunnistamaan ja analysoimaan laajojakin asiakirjakokonaisuuksia, jotka veisivät ihmiseltä moninkertaisesti aikaa.

2.3 Sovelluksia

Pullonpalautusautomaatti on kaikille tuttu järjestelmä, jossa konenäkö on keskeisessä roolissa. Siinä kamera kuvaa pullon viivakoodin ja hylkää tai hyväksyy lopputuloksen. Vanhemmissa versioissa automaatissa yleensä vain yksi kamera, joka kuvaa viivakoodin yhdestä suunnasta yleensä ylhäältä. Järjestelmä ei osaa pyörittää pulloa oikeaan asentoon viivakoodin suhteen vaan tämä jää ihmisen tehtäväksi. Uudemmat versiot sisältävät useamman kameran tai linssin,

joiden avulla konenäkö tunnistaa pullon viivakoodin monesta eri suunnasta. Tämän lisäksi järjestelmän ohjaus pyörittää pulloa hihnan päällä löytääkseen viivakoodin. Kuitti tulostuu automaattisesti, jos hihnalla ei tunnisteta pulloa.

Erilaiset teollisuuskäytössä olevat konenäkösovellukset ovat iso kokonaisuus. Turvajärjestelmässä konenäköjärjestelmälle voidaan asettaa ns. turva-alue, johon ihminen astuessaan laukaisee hätä-seis-kytkimen. Kamera voi tunnistaa liikkeen tai sitten se voi toimia infrapunalla eli ihmisen lämpö laukaisee turvajärjestelmän. Tällainen järjestelmä voi korvata perinteisen aidan tilanteissa, joissa esimerkiksi työskentelevän robotin tai ohjaimen luo pitää mennä useasti päivässä tekemään muutoksia asetuksiin.

Erilaiset kasvojentunnistusmenetelmät ovat arkipäivää nykyään niin turvallisuussovelluksissa kuin älypuhelimissa. Älypuhelimessa käyttäjä opettaa laitteen kasvojensa piirteet, josta älypuhelin tekee tekoälyn ja konenäköjärjestelmän avulla kolmiulotteisen mallin. Näytön lukitusta avattaessa puhelin vertaa tätä tallennetta sillä hetkellä näkyvään kasvokuvaan. Puhelimissa on nykyään jopa kolme kameraa varmistamassa laadukkaat kuvat sekä tekemässä tarkempia biometrisiä eli ihmisen fysiologiaan perustuvia tunnisteita. Muita yksilöiviä biometrisiä tunnisteita ovat esimerkiksi sormenjäljet.

2.4 Konenäköjärjestelmän hyödyntäminen automaatioprosessissa

Konenäköä ja sen dataa voidaan hyödyntää monissa automaatioprosesseissa. Metalliteollisuuden prosessissa voi olla paikkoja, jotka ovat ihmiselle liian kuumia. Tällöin prosessia joudutaan valvomaan etäältä valvomosta. Konenäkökamera voi tällaisessa tilanteessa valvoa prosessia paikan päältä esimerkiksi kappaleen pinnanmuotojen, mittojen ja lämpötilan osalta. Tällä varmistetaan, että kappaleet pysyvät annetuissa raja-arvoissa ja lopputuotteen laatu ei kärsi.

Helpompia toteuttaa ovat prosessit, joihin myös ihminen osallistuu. Tällöin ei myöskään kameralta vaadita niin paljon kalliita teknisiä ominaisuuksia ja järjestelmän ylläpito sekä asennus on helpompaa. Tällainen sovellus voisi olla esimerkiksi pakkauslinjalla, jossa laatuvarmistettu kappale pakataan ja pinotaan.

Tällaisessa tapauksessa voi riittää, että pakettien koot ovat halutuissa arvoissa suhteellisen isolla marginaalilla ja lavojen viivakoodit aikaleimataan, jotta mahdolliset virheet voidaan paikantaa oikeaan paikkaan ja oikeaan aikaan.

Voidaan myös ajatella, että konenäkö valvoo ihmisen tekemisen laatua esimerkiksi kunnossapidon tai asennuksen osalta. Tehdäänkö asiat ohjeiden mukaan, oikeassa järjestyksessä ja ovatko turvajärjestelyt riittävät. Tähän liittyviä ongelmia valvonnan laillisuudesta ja eettisyydestä pohditaan tämän luvun viimeisessä kappaleessa.

Turvallisuus on yksi tärkeimmistä asioista modernissa teollisuuslaitoksessa. Turvallisuutta mietittäessä on aina huomioitava koko yritys ja johdon tehtävänä on varmistaa toiminnan turvallisuus prosessin kaikissa vaiheissa. Tämä on otettava huomioon ennen kuin otetaan käyttöön yhä älykkäämpiä ja itsenäisempiä järjestelmiä kuten esimerkiksi konenäköjärjestelmät. Asennukset, testaus ja käyttöönotto tulee standardisoida, jotta voidaan taata turvalliset työolot ihmisille ja ympäristölle.

Esimerkkejä turvajärjestelmistä ovat hätä-seis-piirit, valoverhot, suoja-aidat, oikosulkusuojat ja sulakkeet. Valoverhon tai suoja-aidan voisi esimerkiksi korvata konenäköjärjestelmällä. Tässä konenäköjärjestelmään määritetään kuva-alue fyysisen alueen sijaan, ja ylitettäessä alueen raja turvajärjestelmä laukeaa ja pysäyttää laitteen.

Turvallisuus ei ole nykyään pelkästään tuotantolaitoksessa tapahtuvaa ennaltaehkäisyä ja fyysisesti läsnä olevien suojelua vaan myös tietoturvaluus on noussut viime vuosina tärkeäksi. Tästä on olemassa paljon esimerkkejä, mutta tietoturvaa ja yksityisyyttä uhkaavia ryhmiä on nykyään kolme: Ammattirikolliset ja hakkerit, joita motivoi raha, protestimeilessä tietoturvaa rikkovien ryhmä sekä valtiot, jotka tarkkailevat kansalaisiaan tai toisia valtioita. (7).

Lähes kaikki automaattioratkaisut ovat nykyään hakkeroitavissa. Mediassakin esillä ollut ydinvoimalan hakkerointi on yksi vakavimmista tietomurroista nykyai-

kaisten automaatiojärjestelmien aikana. Tätä kaikkea torjutaan esimerkiksi turvallisuusyritysten ja valkohattuisten hakkereiden eli taitoan hyvään käyttävien ihmisten avulla. Myös konenäkökameran, tekoälyn ja hienostuneiden algoritmien avulla on mahdollista tunnistaa esimerkiksi sähköpiirien kuvista eroavaisuuksia ja epäloogisuuksia, jotka on mahdollisesti tahallisesti aiheutettu. Tietoturvallisuudessa on meneillään kilpajuoksu tietojärjestelmien suojaajien ja hakkerioiden kesken.

Teknologian ja tekoälyn kehittyessä tulee väistämättä mieleen myös asioiden eettinen puoli. Kuka tekee päätökset saadusta tiedosta ja sen hyödyntämisestä, kun asennus- tai kunnossapitoprosessia valvotaan kameralla? Missä menee ihmisen valvomisen rajat ja kenellä on oikeus nähdä tätä henkilöä koskevaa tietoa? Kenen vastuulla on, jos itseohjautuva eli autonominen kulkuneuvo ajaa ihmisen yli? Nämä kysymykset tulevat lisääntymään koneoppimisen ja tekoälyn kehittyessä. Tarvitaan päivityksiä lakeihin ja sääntöihin, jotta kaikki teknologia saadaan valjastettua parhaaseen mahdolliseen käyttöön myös eettisestä näkökulmasta. On selvää, että tähän tarvitaan kaikkien tieteenalojen asiantuntijoiden yhteistyötä.

3 Tiedonsiirto

Tiedonsiirto on sähköistä tiedon siirtämistä yhteyskanavan avulla järjestelmästä toiseen. Käytössä on erilaisia siirtotapoja ja valinta riippuu datan määrästä sekä millaista dataa lähetetään.

FTP (File Transfer Protocol) on yksi vanhimmista tiedonsiirtomenetelmistä ja sitä käytetään myös tässä työssä. FTP toimii asiakas-palvelin-periaatteella. Asiakas tai asiakkaat lähettävät pyyntöjä palvelimelle halutusta tiedosta tai ohjelmista ja palvelin toimittaa asiakkaan pyytämät tiedot asiakkaalle (8.) FTP-palvelin voi toimia joko passiivisessa tai aktiivisessa tilassa. Erot tilojen välillä ovat siirtoyhteyden muodostamistavassa. Passiivinen valmistautuu siirtämään tietoa, mutta odottaa että asiakas avaa tiedonsiirtoyhteyden. Aktiivinen palvelin avaa itse yhteyden asiakkaaseen ja aloittaa tiedonsiirron. (9.)

Tiedonsiirron ehtona voi olla FTP:ssä salasanan ja käyttäjätunnuksen syöttäminen tai palvelimelle voi olla vapaa pääsy riippuen saatavan tiedon arkaluonteisuudesta. Oikeudet voidaan myös rajata tiettyyn käyttäjäryhmään tai organisaatioon. FTP on tietoturvasoltaan vaatimaton tiedonsiirtotapa, minkä vuoksi siitä ollaan luopumassa monissa yrityksissä. Esimerkiksi Windows 10:ssä FTP:tä voidaan käyttää, mutta oletusasetuksissa se on ei-aktivoitu.

TCP-protokolla (Transmission Control Protocol) on luotettava tiedostonsiirtomenetelmä kahden tietokoneen välille, joka toimii asiakas-palvelin-periaatteella ja jossa on myös elementtejä vikatilanteesta toipumiseen. (10.)

SSH (Secure Shell) on salausmenetelmä, jonka avulla verkkopalveluja voidaan käyttää turvallisesti suojaamattoman verkon yli. SSH käyttää samaa asiakas-palvelin-tapaa kuin FTP. SSH:ssa kaikki data on salattu toisin kuin FTP:ssä. Ohjelman purkuun ja lukemiseen käytetään tekstipohjaisia salauksia ja niitä voidaan lukea esimerkiksi Windowsin komentokehoteella tai PowerShell-ohjelmalla. SFTP (Secure File Transfer Protocol) hyödyntää SSH:n salausprotokollaa ja nostaa FTP:n turvallisuutta merkittävästi. (11.)

HTTP (Hypertext Transfer Protocol) on alun perin suunniteltu WWW (World Wide Web) -sivujen tiedonsiirtoon. Hypertext tarkoittaa, että tekstit ovat linkitetty toisiinsa ja lukijalla on niihin vapaa pääsy. Tällaisia tekstejä näkee esimerkiksi monien sähköpostien allekirjoituksissa, joissa yrityksen internet-sivut on linkitetty suoraan tekstiin. Tähän perustuvat suurelta osin WWW-sivut, jossa kaikki on linkitetty toisiinsa ja napauttamalla mm. kuvaa, tekstiä tai mallia sivuilla vierailija pääsee katsomaan aihepiiriin linkitettyjä internetsivuja. (12.)

P2P (Peer to Peer) on verkkoprotokolla, jossa jokainen verkkoon kytketty asiakaslaite toimii sekä palvelimena että asiakkaana verkon muille jäsenille. Kyseistä järjestelmää käytettiin 2000-luvun alkupuolella, jolloin tiedonsiirtonopeudet olivat nykymittapuulla varsin vaatimattomat. P2P oli tuolloin varsin yleinen tapa ladata musiikkia tai elokuvia toisilta käyttäjiltä. Etuna oli, että resurssit olivat yhteisessä käytössä ja uusien tahojen liittyessä järjestelmään, järjestelmän

kapasiteetti ja siirtonopeus vain kasvoi. Järjestelmä sai osakseen myös paljon kritiikkiä, koska tekijänoikeudet jäivät suurilta osin huomioitta (13).

Peripheral devices eli oheis- tai lisälaitteet ovat laitteita, jotka voidaan yhdistää helposti tietokoneeseen, mutta eivät kuitenkaan kuulu vakiokokoonpanoon kuten keskusyksiköt tai virransyöttöyksiköt. Yleisin käytetty laite on USB (Universal Serial Bus). USB on sarjaväyläarkkitehtuuri, joka liittää oheislaitteet tietokoneeseen asiakas-palvelin-periaatteella (14). Hiiret ja näppäimistöt ovat tästä tyyppiesimerkkejä. Nämä ovat myös ns. Plug and Play -laitteita, jolloin esimerkiksi liitettäessä tulostin tietokoneeseen USB:n avulla, tietokone saa tiedon laitteesta ja mallista. Siten tietokone osaa valita oikeat ohjelmat ja ajurit juuri kyseistä laitetta varten ja tulostin on sekunneissa käyttövalmis. USB:n kautta tapahtuva tiedonsiirto ei tarvitse toimivaa verkkoyhteyttä.

Konenäköjärjestelmissä käytetään yleisesti FTP (File Transfer Protocol) tai SFTP (Secure File Transfer Protocol) -yhteyttä kuvien siirtoon. Konenäköjärjestelmä perinteisesti osana prosessin ohjausta ei vaadi niin paljon suojausta kuin arkaluonteinen data. Toki pilvipalveluiden yhä lisääntyvä käyttö esimerkiksi kuvien tallennusta ja analysointia varten asettavat myös suojaukselle kovempia vaatimuksia.

4 Lastauksen valvonta konenäkökameralla ja kuvien tallennus ulkoiselle palvelimelle

4.1 Setupin rakentaminen

Opinnäytetyön esimerkkitapauksessa simuloitiin tilannetta, jossa elintarvikepurkkeja on aseteltu lavalle erilaisiin muodostelmiin. Opinnäytetyössä käytettiin 3x2-muodostelmaa (kuva 2), jossa harmaat kuutiot vastaavat elintarvikepurkkeja. Konenäkökamera valvoo lastausta laskien kuutioiden määrän sekä yläpinnan samankaltaisuuden hyväksyen tai hyläten kuvan.

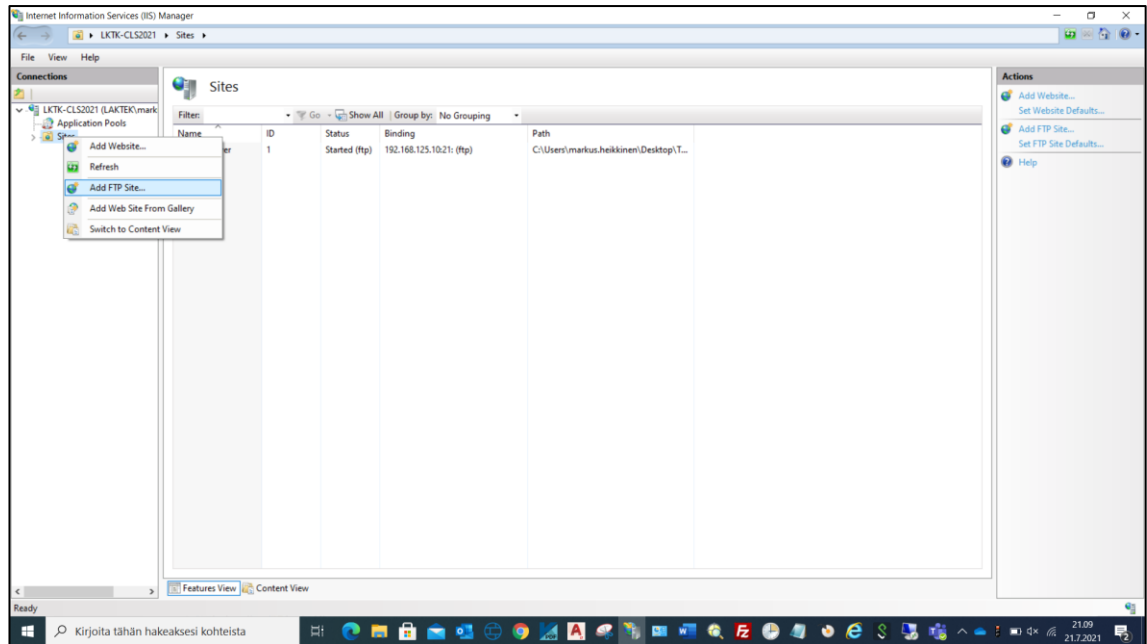


Kuva 2. Asetelma kuudesta harmaasta kuutiosta.

4.2 Konenäköjärjestelmän konfigurointi

4.2.1 FTP-palvelimen tekeminen

Työssä luotiin FTP-palvelin Windowsin omalla IIS (Internet Information Services) - hallintaohjelmalla, koska se oli ilmainen ja varmennettu tapa luoda data-siirtoyhteys ulkoiseen palvelimeen (kuva 3).



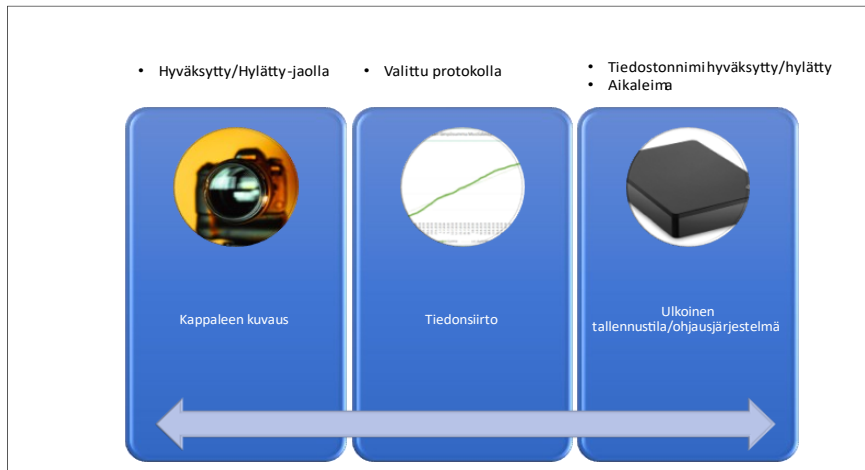
Kuva 3. Windowsin IIS-managerin perusnäyttö ja FTP-palvelimen lisääminen.

FTP-palvelimen luonti oli suoraviivaista ja nopeaa (liite 1):

- palvelimen nimeäminen
- IP-osoitteen määrittäminen
- uuden käyttäjän luominen sekä oikeuksien määrittely
- palvelimen online-tilan varmistus
- kirjautuminen FTP-palvelimen osoitteeseen ja kuvat palvelimelle.

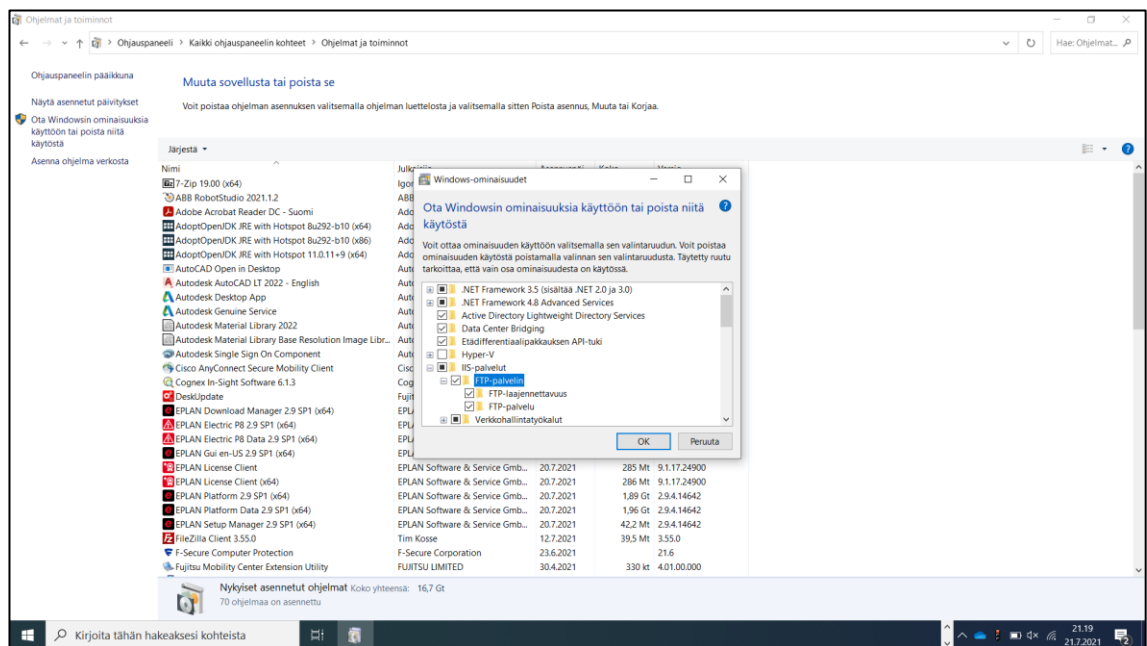
4.2.2 Yhteyksien konfigurointi

Konenäköjärjestelmän tiedonsiirron aloittamisessa ulkoiselle palvelimelle on tärkeä etsiä vakaa ja toimiva tallennustila. Tässäkin on paljon vaihtoehtoja, mutta kameran tiedonsiirto-ominaisuudet rajaavat näitä vaihtoehtoja. Kuvassa 4 on yksinkertaistettu prosessikaavio tiedonsiirrosta. Työssä käytetyn integroidun kameran ainoa tiedonsiirtotapa ulkoiselle palvelimelle oli käyttää FTP:tä. Windowsin Azure-pilvipalvelu mittavine ominaisuuksineen on hyvä vaihtoehto, mutta kyseisen palveluntarjoajan kaikki yhteysprotokollat ovat suojattuja ja niiden hyödyntäminen vaatisi syvää ymmärrystä ohjelmoinnista ja tiedonsiirtomenetelmistä.



Kuva 4. Prosessikaavio tiedonsiirrosta ulkoiselle palvelimelle/ohjausjärjestelmään.

Ensimmäiseksi otettiin käyttöön ohjauspaneelin ohjelma-asetuksista IIS-palvelut ja sen alavalikosta FTP-palvelimen käyttöönotto (kuva 5).



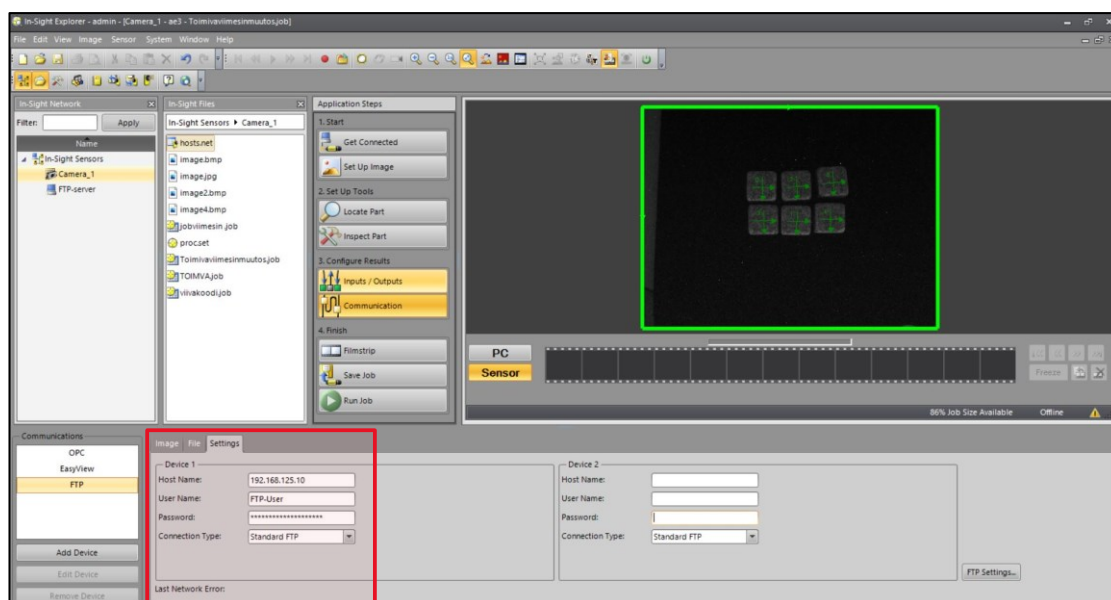
Kuva 5. Windowsin ohjauspaneelin ominaisuuksien hallinta.

FTP valittiin myös siksi, että kyseinen menetelmä oli kameran käyttöoppaan mukaan ainoa mahdollisuus saada kuvat siirtymään automaattisesti kameran laukaisimesta ulkoiseen tallennustilaan. Tässä työssä pääasiallisena ulkoisena

tallennustilana toimikin FTP-palvelin sekä Microsoft Azuren pilvipalvelussa toimiva virtuaalinen tallennustila. Tässä työssä käytettiin FTP:n passiivista tiedonsiirtotapaa.

Seuraavaksi konenäkökameran ohjelmistoon syötettiin FTP-palvelimen IP-osoite 192.168.125.10 Host Name-kohtaan. Konenäkökameran IP-osoite oli 192.168.125.11. FTP-palvelimen asetuksissa käytettiin pienintä mahdollista tiedonsiirtosuojausta, jolla varmistettiin toimiva yhteys palvelimeen.

Tämän jälkeen lisättiin FTP-User-käyttäjätunnus ja salasana, jotka oli luotu aikaisemmin FTP-serverin tekemisen yhteydessä (kuva 6).

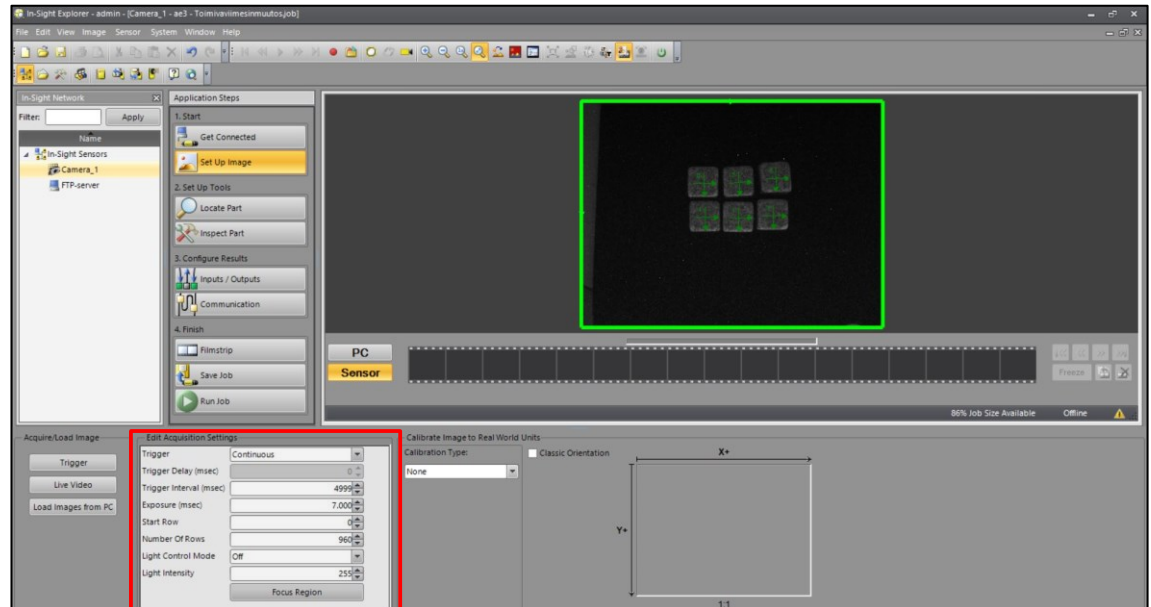


Kuva 6. Kameran ohjelmiston perusnäkö.

4.2.3 Konenäkökameran kuvantunnistustyökalun konfigurointi

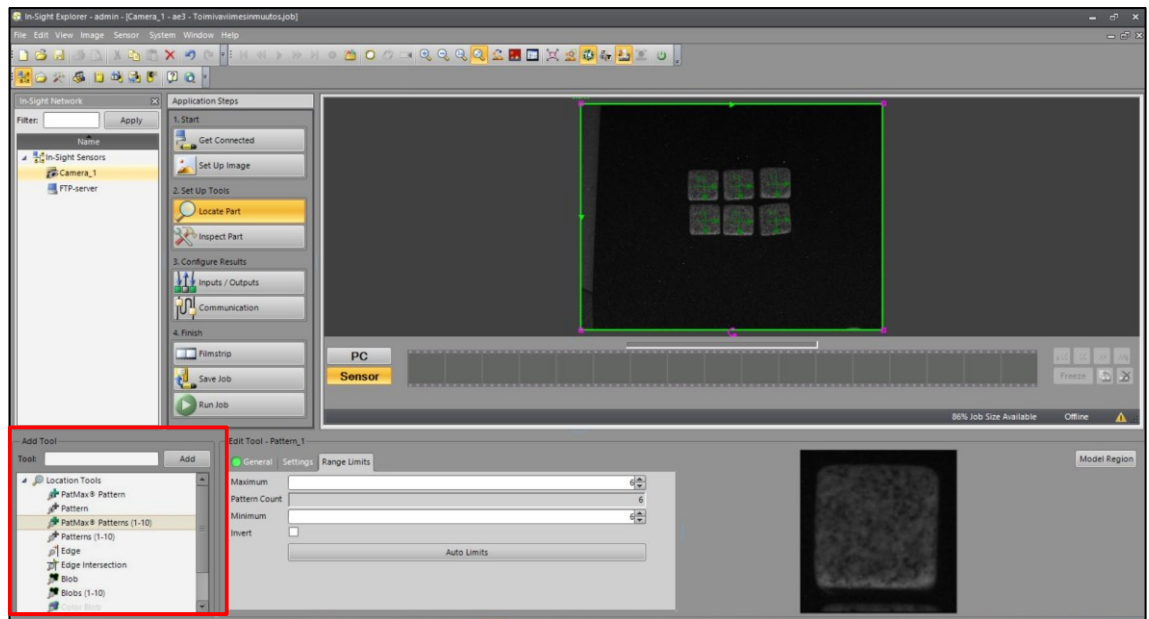
Valaistuksen vakioiminen kameran kuvantunnistustyökalun luottavuuden parantamiseksi on tarkkaa työtä. Konenäkökamera voi antaa virheellisiä tietoja kuvasta, jos valaistus vaihtelee paljon.

Toimivan yhteyden jälkeen kameran ohjelmistosta asetettiin automaattilaukaisin toimimaan seitsemän sekunnin välein. Kuvattavana kohteena oli kuusi kuutiota, joiden tarkoitus oli kuvata elintarvikepurkkeja (kuva 7).



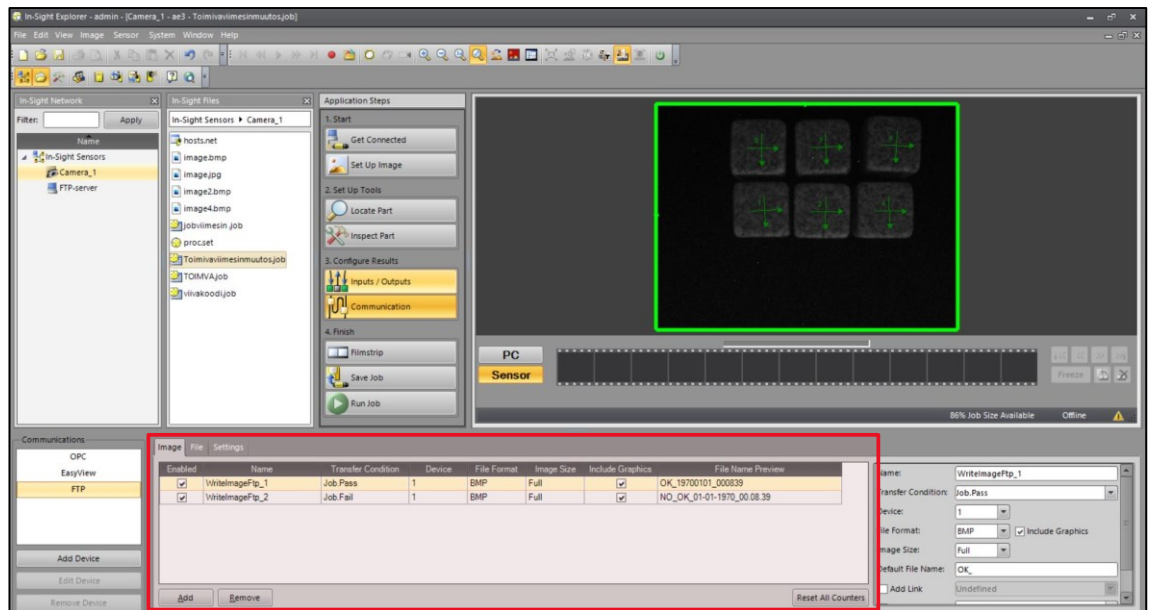
Kuva 7. Automaattinen kameran laukaiseminen seitsemän sekunnin välein.

Kameran kuvantunnistustyökaluista valittiin Pattern Tool -sarjasta se, joka laskee, montako samanlaista muotoa löytyy koko kuvan alueelta (kuva 8). Aluksi työkalulle rajattiin vain yksi neliö, johon ohjelma vertasi viittä muuta. Työkalu etsii tässä tapauksessa samanlaista muotoa, pikselimäärää sekä kappalemäärää. Ohjelmiston oletusarvoja ei tässä vaiheessa muutettu.



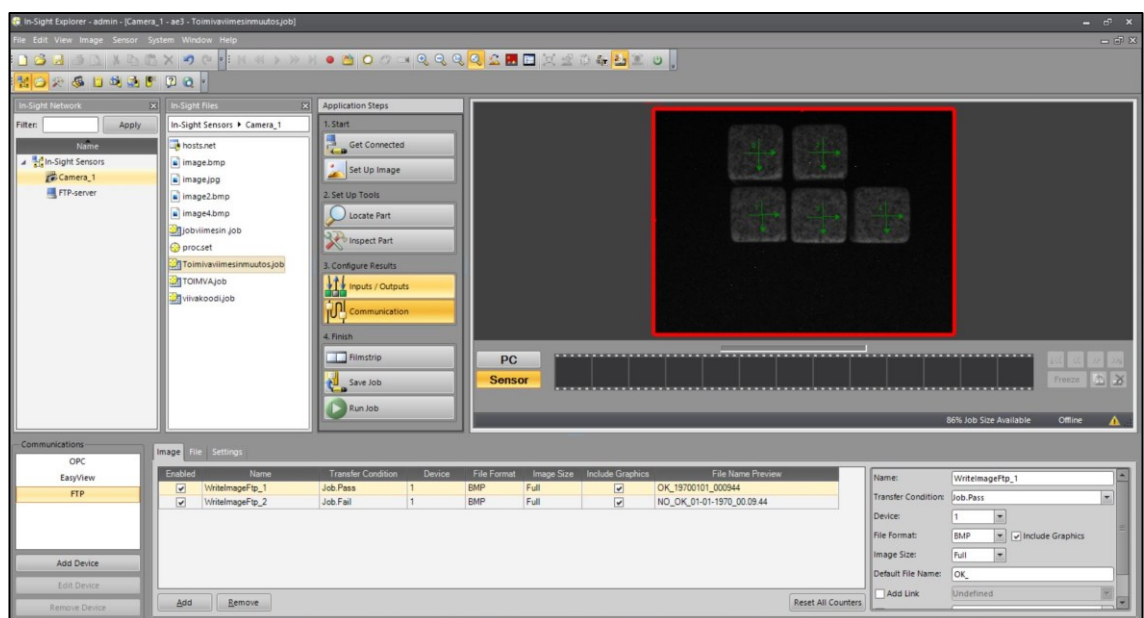
Kuva 8. Kuvantunnistustyökalujen asettaminen neliöiden tunnistamiseksi.

Seuraavaksi käytettiin käskyä `writeimage_ftp`, joka muuntaa kameran laukaisusta otetun kuvan FTP-paketiksi, joka lähtee FTP-palvelimelle automaattisesti järjestelmän ollessa online-tilassa (kuva 9). Siirtoehdoksi laitettiin ohjelmiston valikosta `job_pass` ja `job_faill`. Hyväksyty eli `job_pass` -siirtoehdon täytyessä FTP-palvelimelle tallentui aikaleima, tiedoston nimi `OK_` ja sekä kaksi kuvaa. Toisessa kuvassa näkyi pelkät neliöt, toisessa neliöt sekä vihreä suorakaide kuvan ympärillä.



Kuva 9. Hyväksytty kuva, jossa vihreä suorakaide kuvan ympärillä sekä tiedostonimi OK.

Hylätty eli job_fail kohdalla näkyi aikaleima, tiedoston nimi NO_OK_ sekä punainen suorakaide kuvan ympärillä (kuva 10).



Kuva 10. Hylätty kuva, jossa punainen suorakaide kuvan ympärillä sekä tiedostonimi NO_OK.

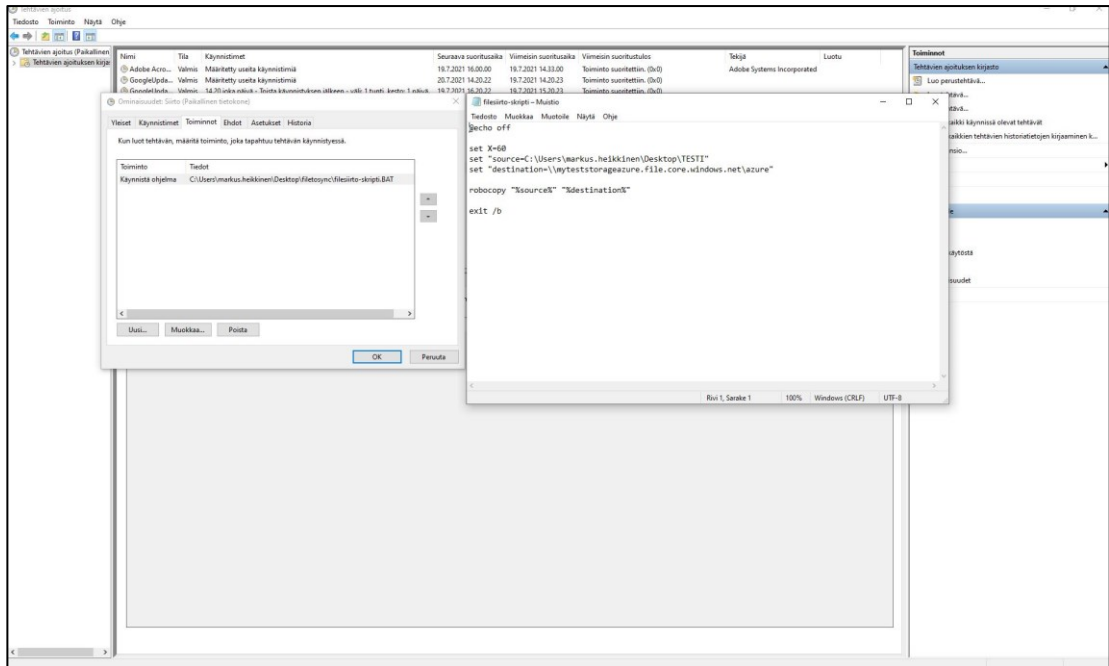
4.2.4 Esteet ja haasteet

Isoimmaksi haasteeksi tässä työssä muodostui toimivan FTP-yhteyden luonti. Työssä jouduttiin käyttämään FTP-yhteyttä, koska kameran ohjelmisto mahdollisti vain tämän tiedonsiirtotavan ulkoiselle palvelimelle. Yhteyden muodostamista vaikeutti se, että kamera oli integroitu ABB:n robotin sisälle ABB:n omien suojausmääritelmien taakse sen sijaan, että kamera olisi oma yksikkönsä, jolloin yhteys kameran nykyaikaisiin tiedonsiirtoportteihin olisi saatu suoraan ja yhteystyyppi tunnistettu automaattisesti. Projektissa oli tarkoituksena käyttää mahdollisimman vähän valmiita palveluja ja tästä johtuen esimerkiksi ABB:n valmiita ohjelmistoja ei haluttu hyödyntää.

Toisena haasteena oli se, ettei Laktek Engineering Oy:llä ollut järjestelmävalvojan oikeuksia turva-asetuksia esimerkiksi Windowsin palomuurin avaamiseen ja suoja-asetusten muuttaminen oli haastavaa. Tästä johtuen yhteys toimi aluksi erittäin huonosti ja vain osa datasta siirtyi palvelimelle ja suurimman osan datan siirtymisestä esti palomuuuri tai jokin muu suoja-asetus.

Microsoftin pilvipalvelua hyödynnettiin, koska se on laajasti käytetty pilvipalvelu lähes rajattomilla laajennusmahdollisuuksilla. Suora yhteys Microsoft Azuren pilvipalveluun ei kuitenkaan onnistunut, koska kameran tiedonsiirto-ominaisuudet olivat rajalliset ja Microsoftin pilvipalvelut ovat hyvin suojattuna.

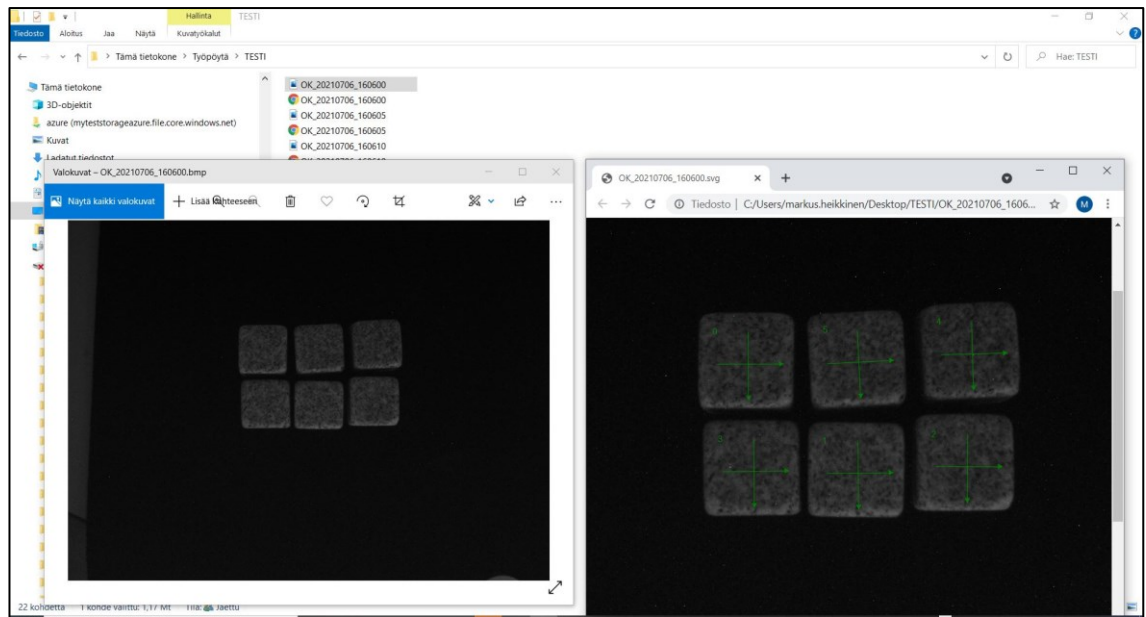
Tiedostojen siirtäminen Windowsista Microsoftin pilvipalveluihin on helppoa vetämällä tai kopioimalla ne, mutta suora yhteys kamerasta Microsoftin pilvipalveluihin ei toiminut vaan edellytti lisäohjelman käyttöä. Tämä lisäohjelma tehtiin Windowsin tehtävien ajoitus -ohjelmalla lisäämällä komentokehoteeseen muuttaman rivin tallennuskäsky (kuva 11).



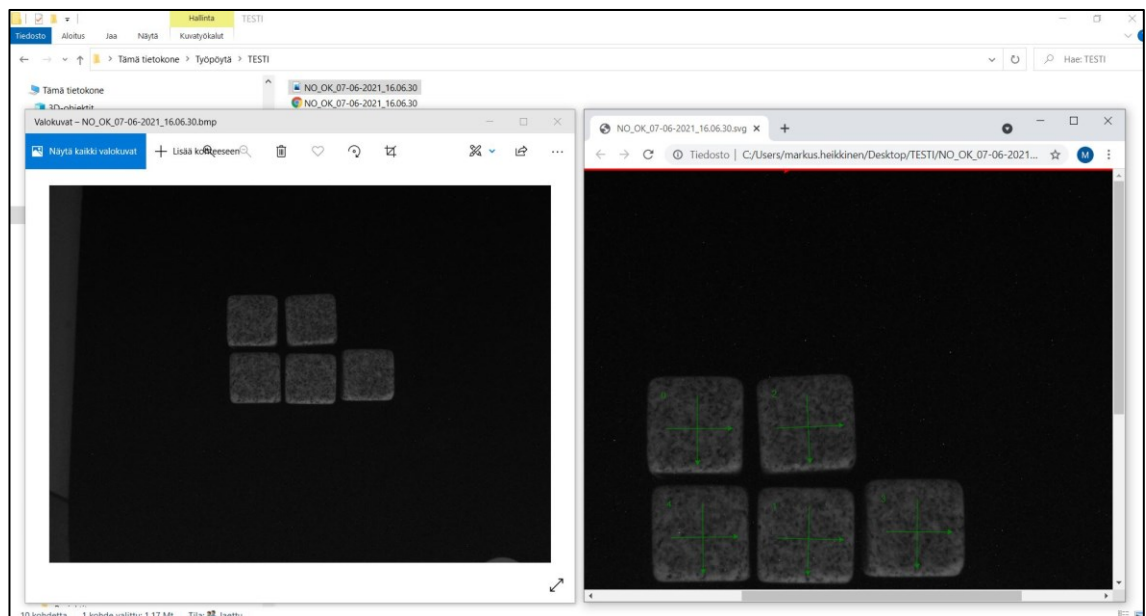
Kuva 11. Ajoitetun tallennuksen tiedostonsiirto-skripti Windowsin tehtävien ajoitus -ohjelmassa.

4.3 Testaus

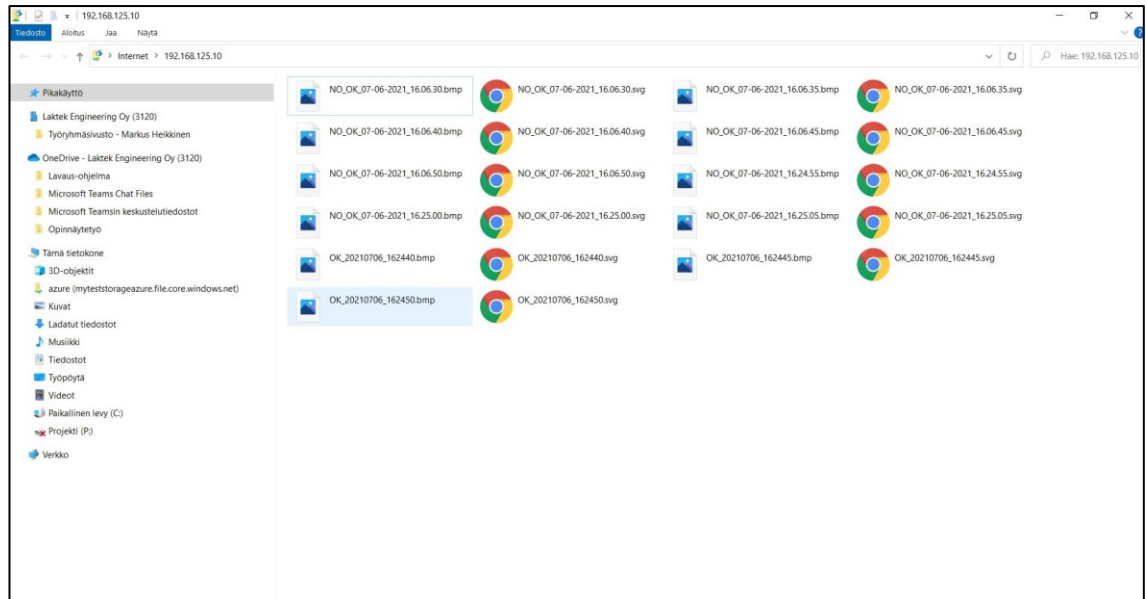
Testausprosessi oli hyvin yksinkertainen. Kamera asetettiin online-tilaan ja tämän jälkeen tarkistettiin, ilmestyisikö kuvia FTP-palvelimelle. Testauksen aikana kuutiomuodostelmasta poistettiin yksi kuutio useampaan kertaan sen varmentamiseksi, että kuvankäsittelyalgoritmi tunnisti virheellisen asetelman eli sen, että kuutioita oli eri määrä kuin kuusi kappaletta. Kuvista 12, 13, 14 ja 15 nähdään, että sovellus toimii; kuvat on nimetty oikein hyväksyty/hylätty -periaatteella, kuvissa on punainen tai vihreä kehys, niissä on aikaleimat ja ne tallentuvat FTP-palvelimelle sekä linkitys Azureen toimii.



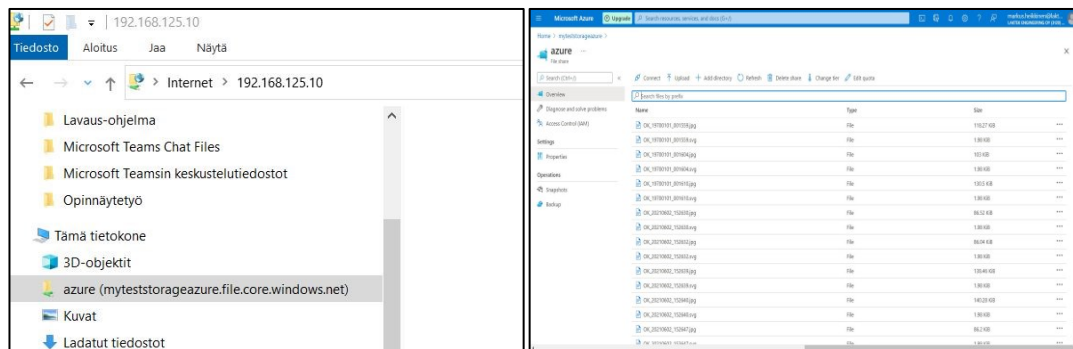
Kuva 12. Hyväksytyissä kuvissa nimi OK_, aikaleima sekä vihreä kehys.



Kuva 13. Hylätyissä kuvissa nimi NO_OK_, aikaleima sekä punainen kehys.



Kuva 14. Kuvat tallentuneet onnistuneesti FTP-palvelimelle.



Kuva 15. Kuvat tallentuneet myös automaattisesti Azuren virtuaalisen tallennustilaan

4.4 Johtopäätökset

Testiajossa kuvat tallentuivat sekä FTP-palvelimelle että Windowsin tehtävien ajoitus -ohjelmalla Microsoft Azuren pilvipalveluun. Tässä esimerkkitapauksessa tausta-ajatuksena on elintarvikkeiden siirtäminen erilaisiin muodostelmiin lavalle. Kivet toimivat tässä elintarvikepurkkeina, jotka kootaan 3x2- , 5x4- tai 3x4- muodostelmiin lavalle, josta ne siirretään jatkokäsittelyyn pakkausta varten.

Mitä hyötyä tästä nimenomaisesta sovelluksesta voisi olla elintarvikkeiden valmistusprosesseissa? Esimerkiksi elintarvikkeiden lastaamisessa konenäkökamera pystyisi havaitsemaan, jos yksikin purkki on mennyt rikki. Kuva tallentuisi pilvipalveluun ja tästä tulisi välittömästi tieto I/O-porttien kautta robotin ohjausjärjestelmään ja robotti pysähtyisi (kuva 16).



Kuva 16. ABB:n IRB14050 -robotin I/O-portit.

Lisäksi elintarvikkeita valmistava yritys voisi hyödyntää konenäköä yhden valmiin kerroksen tai koko lavan viivakoodin lukemisessa, koska elintarvikkeiden valmistus on erittäin hygieenistä ja tarkkuutta vaativaa työtä. Jos kävisi niin, että valmistuksessa olisi mennyt jotain vikaan ja lopputuote olisi esimerkiksi asiakkaan mielestä pahanmakuinen tai muutoin epäkelpo, niin yritys saisi tallentuneiden viivakoodien ja kuvan aikaleimojen perusteella tiedon siitä, milloin mahdollinen vahinko on sattunut ja mitä tuotantoerää se koskee. Näin yritys voisi reagoida heti mahdolliseen ongelmatilanteeseen ja katsoa, toistuuko kyseinen ongelma tietyssä prosessin vaiheessa esimerkiksi pastöroinnissa.

Kolmas yritykselle hyödyllinen asia olisi kappaleiden, erien tai lavojen laskeminen sen varmistamiseksi, että tuotannosta tulee ulos täsmälleen oikea määrä tuotteita.

Nämä edellä mainitut tunnistustilanteet voidaan toteuttaa kahdella eri tavalla; kamera on integroituna robotin tarttujaan tai kamera on omana yksikkönään. Esimerkiksi roboteissa, joiden siirtonopeus on suuri, kamera ei ehdi ottamaan riittävän laadukkaita kuvia ja näissä tilanteissa on järkevämpää käyttää konenäkökameraa omana, erillisenä yksikkönään. Tutkielmassa kamera oli integroitu robotin tarttujaan, koska robotin siirtonopeus oli pieni (kuva 17).



Kuva 17. Konenäkökamera integroituna robotin tarttujaan.

5 Yhteenveto

Opinnäytetyössä keskityttiin suunnittelemaan toimivaa automaattista yhteyttä konenäköjärjestelmästä ulkoiseen tallennustilaan sekä kuvankäsittelyalgoritmia hyväksyty/hylätty -periaatteella. Konenäkökameran käyttö esimerkiksi lastaamisen valvomiseen on opinnäytetyön perusteella mahdollista ja perusteltua. Konenäköjärjestelmä pystyy ihmisen sijaan tekemään yksitoikkoista työtä tauotta vapauttaen henkilöstöresursseja tuottavampaan työhön.

Pilvipalveluihin tallennettujen tietojen analysointi tehostaa prosessia; datasta saadaan pitkän aikavälin tietoa, ennakointikyky paranee ja ongelmat automaatioprosessissa saadaan minimoitua.

Jotta dataa voidaan hyödyntää mahdollisimman hyvin, ei pelkkä konenäkökamera riitä, vaan tarvitaan myös esimerkiksi lämpötila-antureita ilmaisemaan moottorin käyntilämpötiloja ja vertaamaan niitä pitkän aikavälin keskiarvoihin. Näin saadaan tietoja, joiden avulla voidaan havaita prosessin kohdat ja ajat, missä lämpötila nousee raja-arvojen yläpuolelle ja tehdä ennaltaehkäiseväisesti toimintoja, joilla prosessi saadaan toimimaan mahdollisimman tehokkaasti ilman ylimääräisiä pysähdyksiä.

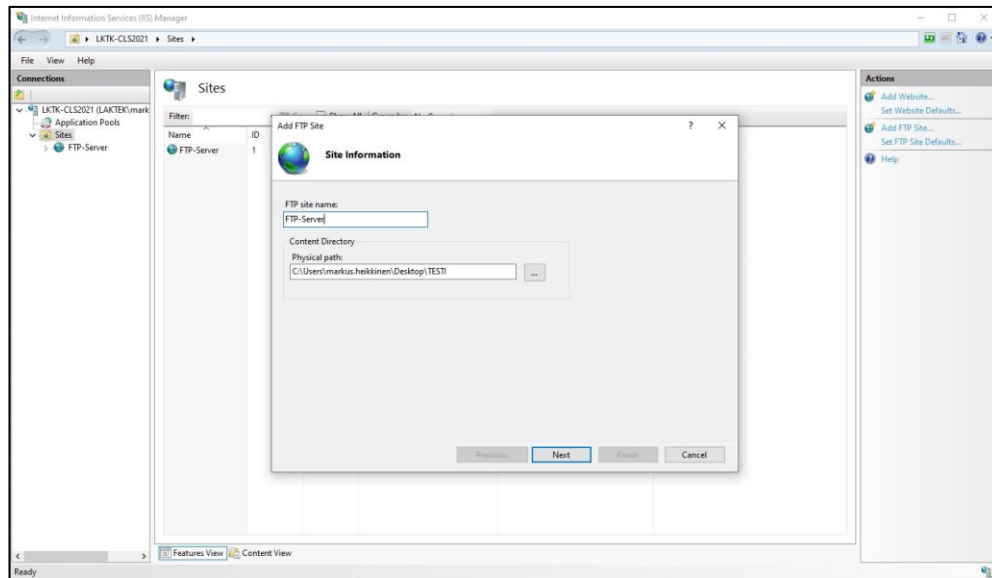
Opinnäytetyön myötä on käynyt selväksi, että konenäköä voi hyödyntää hyvin monipuolisesti erilaisissa automaatioprosesseissa. Perinteiset kuvantunnistusmenetelmät, kuten mittaaminen ja kappaleiden laskeminen ovat saaneet rinnalleen uusia älykkäämpiä menetelmiä, jotka pystyvät yhä tarkemmin havaitsemaan ja analysoimaan kuvan sisältämää informaatiota.

Lähteet

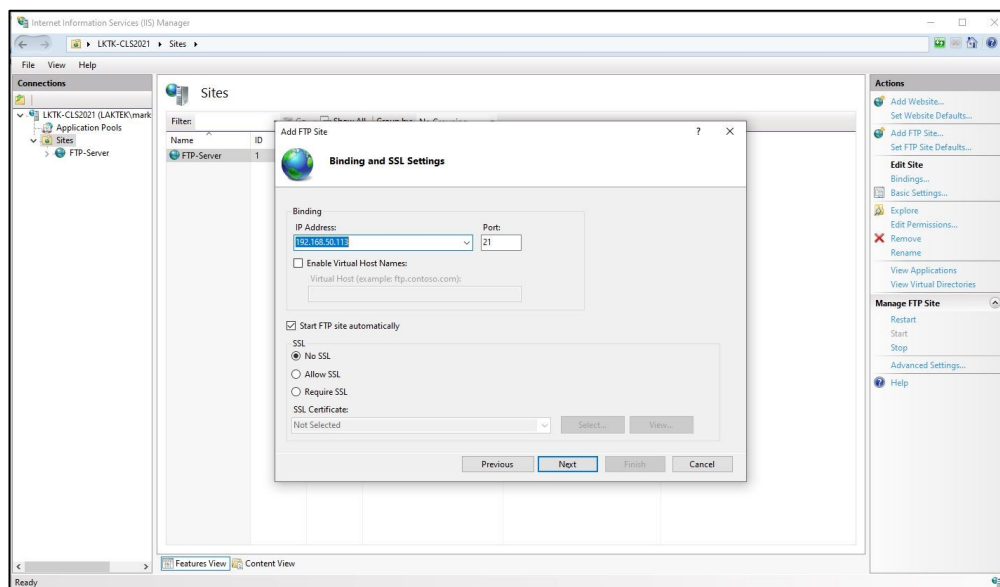
- 1 Beyerer, Jürgen; Puente León, Fernando & Frese, Christian 2016. Machine Vision - Automated Visual Inspection: Theory, Practice and Applications. Berlin: Springer.
- 2 Graves, Mark & Bruce G. Batchelor 2003. Machine Vision for the Inspection of Natural Products. Springer. p. 5.
- 3 Steger, Carsten; Markus Ulrich; Christian Wiedemann 2018. Machine Vision Algorithms and Applications (2nd ed.). Weinheim: Wiley-VCH. p. 1.
- 4 Batchelor, B.G. & Whelan, P.F. (2012) Basic Machine Vision Techniques. Springer, London, s.26-28, 40-60.
- 5 Kuvankäsittelymetodeja. 2020. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/Konen%C3%A4k%C3%B6>>. Luettu 13.8.2021.
- 6 Teollinen kuvankäsittely. 2021. Verkkoaineisto. 3D-konenäkö. <https://www.sick.com/fi/fi/teollinen-kuvankaesittely/c/g114858?q=:Def_Type:ProductFamily>. Luettu 13.8.2021.
- 7 Hyppönen Mikko. TED Global 2011. Verkkoaineisto. <https://www.ted.com/talks/mikko_hypponen_fighting_viruses_defending_the_net>. Luettu 17.8.2021.
- 8 Lyhenneluettelo, Kotus. 2021. Verkkoaineisto. Wikipedia. <<https://fi.wikipedia.org/wiki/FTP>>. Luettu 14.8.2021.
- 9 Palvelin. 2020. Wikipedia. Verkkoaineisto. <<https://fi.wikipedia.org/wiki/Konen%C3%A4k%C3%B6>>. Luettu 14.8.2021.
- 10 TCP. 2020. Wikipedia. <<https://fi.wikipedia.org/wiki/TCP>>. Luettu 16.8.2021.
- 11 SSH. 2020. Wikipedia. <<https://fi.wikipedia.org/wiki/SSH>>. Luettu 16.8.2021.
- 12 HTTP. 2020. Wikipedia. <<https://fi.wikipedia.org/wiki/HTTP>>. Luettu 16.8.2021.
- 13 Vertaisverkko. 2020. Wikipedia. <<https://fi.wikipedia.org/wiki/Vertaisverkko>>. Luettu 16.8.2021.
- 14 USB. 2020. Wikipedia. <<https://fi.wikipedia.org/wiki/USB>>. Luettu 16.8.2021.

Liitteet

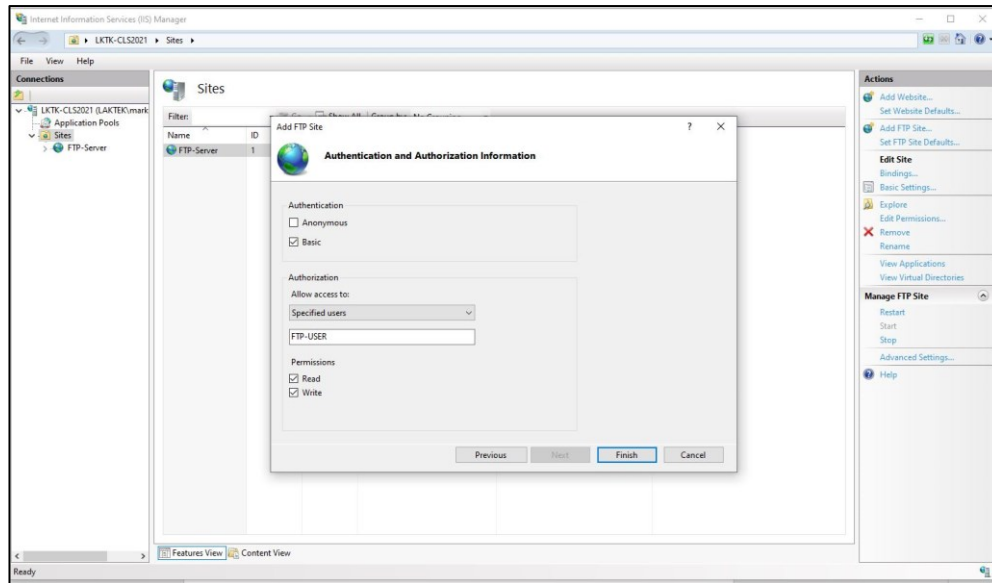
Liite 1: FTP-palvelimen luomisen lyhyt manuaali Windows 10:llä.



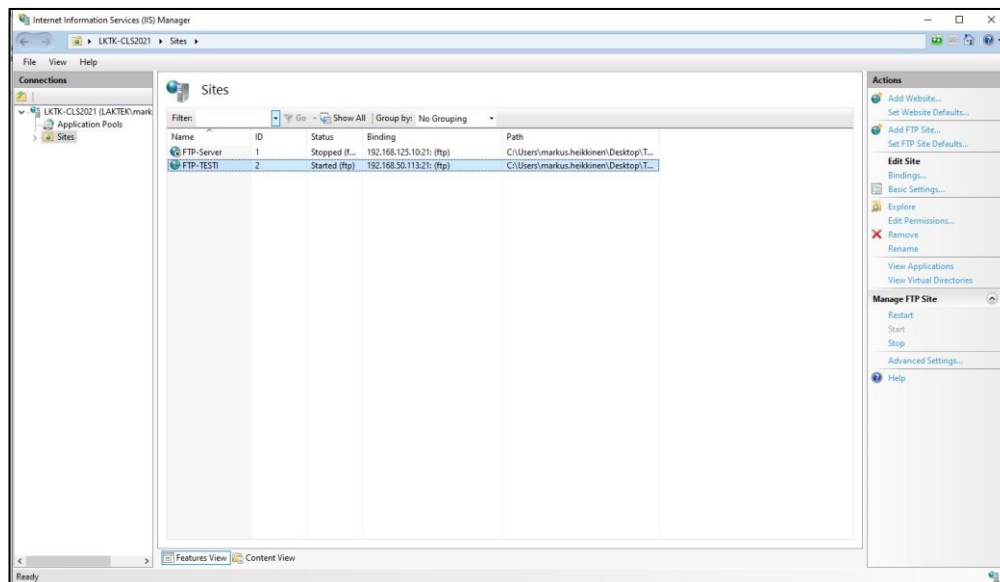
Kuva 1. FTP-palvelimen nimeäminen ja osoitteen määrittäminen.



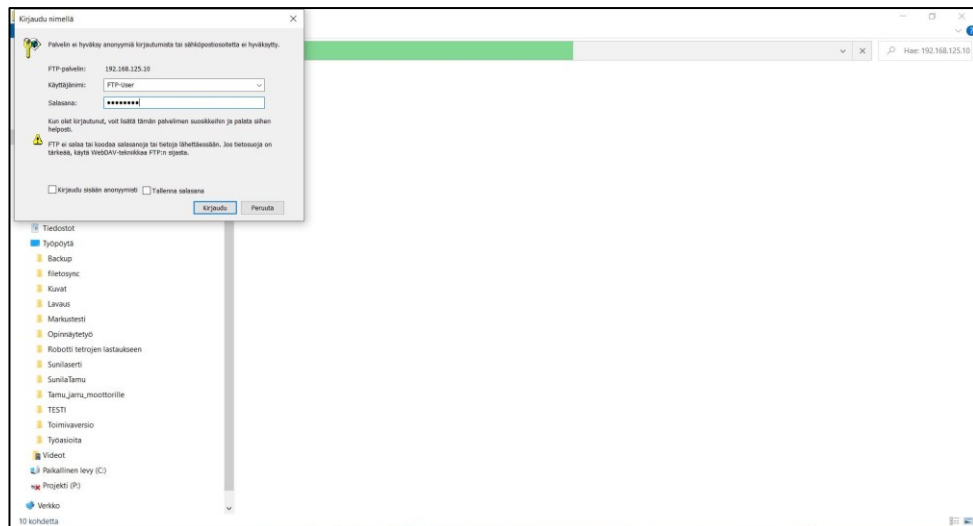
Kuva 2. IP-osoitteen lisääminen sekä SSL-salauksen poisjättö.



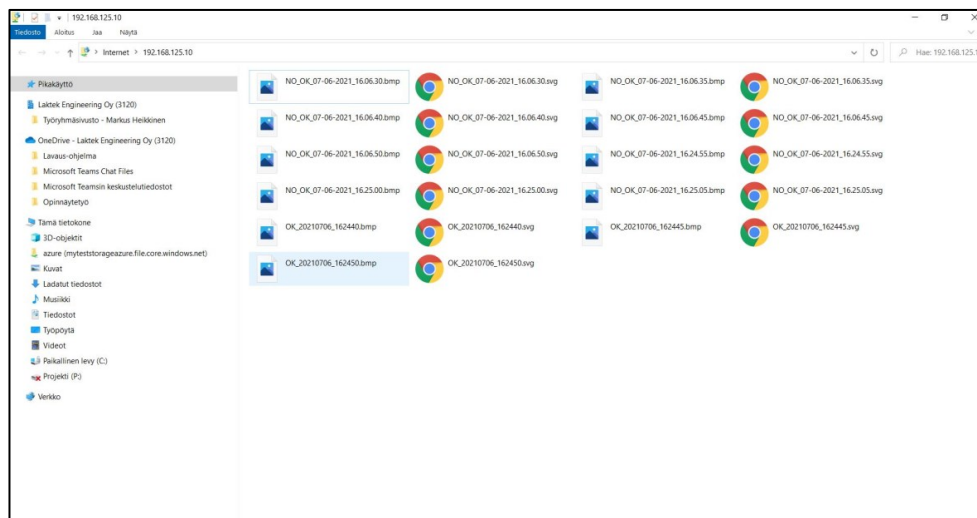
Kuva 3. Uuden käyttäjän lisääminen ja oikeuksien antaminen.



Kuva 4. Loppunäkymä IIS Managerissa ja serverin online-tilan varmistaminen.



Kuva 5. Näkymä uuden käyttäjän kirjautumisesta FTP-palvelimelle.



Kuva 6. Näkymä FTP-palvelimelle kirjautuneena sekä kuvat nimetty joko OK tai NO_OK.