**Tekijät: Mika Matkaselkä**

**Optical character recognition system - raportti**

**CENTRIA-AMMATTIKORKEAKOULU**

**ATIS19Y**

**2.9.2021-2.12.2021**

Ryhmätyön aiheeksi valikoitui OCR, eli optical character recognition system, aihe oli vaihtoehdoista mielenkiintoisin ja myös muut ryhmät päätyivät tekemään samasta aiheesta työnsä. Opettaja oli sitä mieltä, että on fiksumpaa tehdä samasta työstä kaikkien, jolloin opastaminen on helpompaa. Muita vaihtoehtoja olisivat olleen Datalogger ja optical distance meter.

Tehtävässä tulisi luoda sulautettu järjestelmä Raspberry PI 4 – laitteelle, joka pystyy kuvasta lukemaan tekstiä. Tätä varten kuvasta tulee löytyä kalibrointia varten pisteet, joiden avulla voidaan säätää haluttuja arvoja. Ohjelma osaa kalibroinnin jälkeen kaapata kuvaa esim. web-kamerasta ja otetuista kuvista se tunnistaa kohdealueen, josta lopulta teksti luetaan. Ohjelman tulisi osata tunnistaa ja säätää kuvan mittasuhteita sekä kääntää kuva tarvittaessa oikeaan kulmaan. Otetun kuvan pitäisi siis toimia, vaikka kameralla otettaisiin kuva väärinpäin olevasta kohteesta. Kuvasta on myös tarkoitus poistaa kohinaa ja tarkalla kalibroinnilla pyritään välttämään se, ettei ohjelma lue arvoja tarkoitetun alueen ulkopuolelta. Alkuvaiheessa ohjelmaan tehdään oma käyttöikkuna, mutta lopullisesta työstä tämä on tarkoitus poistaa, jolloin sitä käytetään restapi rajapinnan kautta selaimesta. Tätä kautta pystytään katsomaan kuvan säädettyjä arvoja, otettua kuvaa ja kuvasta luettua tekstiä parametreineen. Tämä tehdään sen vuoksi, että ohjelman arvoja voidaan muokata ja säätää ilman varsinaista käyttöliittymää, jota monesti ei ole käytössä Raspberry PI -laitteissa.

Ryhmässäni ei ollut muita jäseniä kuin minä, joten myös kaikki vastuualueet jäivät minun tehtäväkseni. Päädyin tähän ratkaisuun, koska kuuluin toiseen ryhmään, kuin muut opiskelijat ja ajattelin etteivät aikataulumme sopisi hyvin yhteen. En tuntenut muita kurssilaisia ennakkoon edellä mainituista syistä. Kurssin työtä ei tarvinnut kuitenkaan suorittaa aivan yksin, koska opettajan tarjoamaa materiaalia ja koodia sai käyttää omissa ryhmätöissä.

Työhön tarvittavat ohjelmistot ja välineet: kamera, QT Creator, Visual studio code, Raspberry PI 4, node js, selain, kuvanmuokkaustyökalu, github, bash ja linux-konsoli.

Toteutumisvaiheet

Kurssin alussa tutustuttiin miten Kubuntu-linux toimii ja mitä ohjelmia kurssin aikana tullaan tarvitsemaan. Varsinaista tehtävää varten piti osata muokata kuvaa ja saada sieltä erottumaan haluttuja asioita. Tätä varten opettelimme muokkaamaan kuvasta mustavalkoisen, tai pikemminkin harmaasävykuvan, josta laitteen on helpompi tunnistaa haluttuja muotoja.

Ohjelmaa työstettiin QT Creatorin avulla oppituntien ja kotiläksyjen muodossa. Ohjelmistoa kehitettiin niin, että luotiin ensin käyttöliittymän sisältävä applikaatio, johon lisättiin uusia osia kurssin mittaan. Ohjelmaan lisättiin kuvan lukemista varten koodia, jonka avulla ohjelma pystyi hakemaan kuvan halutusta paikasta. Tähän lisättiin myöhemmin ominaisuus, että kuvasta pystyttiin muuttamaan eri sävyisiä kuvia, kuten harmaasävykuva. Käyttöliittymä luotiin sen vuoksi, että mitattaviin arvoihin tehtävät muutokset saataisiin näkyville ohjelman kehitysvaiheessa. Käyttöikkunaan lisättiin erilaisia kalibrointiarvoja, joiden avulla pystyttiin näkemään mitattavien pallojen koko, etäisyys ja muuttamaan muita raja-arvoja. Näitä arvoja muuttamalla saatiin halutut kohteet joko näkymään korostettuna kuvissa, tai poistettua. Säätöjä muuttamalla sai kontrastia muutettua niin, että kuva oli joko todella tumma, tai vaalea. Tämä mahdollisti ei haluttujen kuvioiden poissulkemisen kuvasta, niin että vain halutun kokoisten ja muotoisten kuvioiden välinen teksti luettaisiin. Kuvioiden ei siis pitänyt olla välttämättä pyöreitä, mutta niiden käyttäminen oli todennäköisesti yksinkertaisinta.

Ohjelmasta oli tarkoitus muuttaa lopulta versio, jonka pystyisi asentamaan Raspberry PI 4:seen, joten ohjelmasta piti poistaa käyttöliittymä ja ohjata sen käyttöä restapi:n avulla selaimen kautta. Vaihtoehtona oli myös luoda ohjelma uusiksi konsolisovelluksena. Ohjelman konfigurointitiedostoa piti siis pystyä muuttamaan konsolin kautta ja restapi haki uudet arvot tämän tiedoston kautta. Restapi:n piti hakea kuva suoraan videolta, mutta oma versioni ei valitettavasti edennyt siihen asti. Restapi haki kyllä muutetut arvot oikeasta tiedostosta, ja ne päivittyivät selaimen configuration sivulle, sekä result sivulla näkyi kuvasta luettu teksti. Tehtävään jäi siis käyttöön testissä käytetty kuva eikä kameran ottama kuva tai videokuva. Sain poistettua käyttöliittymän, mutta en osannut tehdä tästä varsinaisesti konsolikomennolla toimivaa ohjelmaa. Yritin myös konsolin kautta muuttaa konfiguraatiotiedoston arvoja, mutta en osaa käyttää konsolikomentoja kovin hyvin, joten se jäi yrityksen asteelle. Löysin kohdekansion ja tiedoston, mutta en osannut muokata tiedostoa. Sain käynnistettyä node js serverin konsolin kautta ja selaimessa avasin ohjelman eri käyttösivuja, jolloin konsolissa näkyi, että mitä sivuja olin käynyt katsomassa.

Lopputulosten analysointi

Missä onnistuttiin

Sulautetut sovellukset eivät olleet minulle entuudestaan tuttuja, joten kurssi oli todella mielenkiintoinen. Itselläni on kokemusta ohjelmointikursseista, mutta varsinaisesti toimivien ja loppuun asti tehtyjen sovellusten tekemisestä minulla ei ollut kokemusta. Pidin paljon siitä, että jos omassa ryhmässä (yksin minun tapauksessani) ei ollut päässyt eteenpäin jostain kohtaa, niin gitin kautta pystyi hakemaan toimivia osia tai kokonaisuuksia koodista. Minullakin kävi useampaan otteeseen niin, että onnistuin saamaan oman ohjelmani solmuun, jolloin tätä kautta pystyi korjaamaan ongelmat. Oli hienoa päästä näkemään ensimmäistä kertaa käytännössä, kun saa oman kameran kautta kuvan ruudulle oppitunnilla tehdyn ohjelman avulla. Vaikka ohjelmassani en päässyt ihan tavoiteltuun lopputulokseen, oli tehtävän tekeminen ja yrittäminen silti todella opettavaista. Olin saanut ainakin jossain välissä tehtävän tekemistä kameran, kalibroinnin ja lukuominaisuudet toimimaan. Myös restapi:n käyttäminen oli opettavaista ja onnistuin sitä kautta katsomaan kuvaa ja lukemaan tekstin siitä. Tämän lisäksi sivulle päivittyivät konfiguraatio tiedossa olevat muutetut arvot. Onnistuin myös konsolin kautta käynnistämään node js serverin ja näin sitä kautta sivustolla tapahtuvat muutokset.

Missä mentiin pieleen

Työ jäi siis vajaaksi omalta osaltani, koska en saanut ohjelmaa toimimaan konsolin kautta. Poistin toki käyttöikkunan ja muut ylimääräiset osat toiminnasta. Kuitenkaan lopputulos ei mielestäni ole valmis käytettäväksi sellaisenaan Raspberryllä, koska siitä puuttuu todennäköisesti tarvittavia ominaisuuksia. Node js:n kautta olisi pitänyt kytkeä staattisen kuvan sijaan näkymään kamerasta otettu kuva. Olisin voinut suunnitella työn tekemisen perusteellisemmin ja säännöllisemmin, sekä työn eri vaiheita. Olisin saanut paremman kokonaiskuvan työstä ja mitä lähden milloinkin kehittämään. Näitä asioita on hankala miettiä jälkikäteen kovin tarkasti, saati lähteä korjaamaan asioita, jos tehtäviä asioita ei ole kirjannut tarpeeksi hyvin ylös. Tarpeeksi aikaa käyttämällä tämäkin olisi voinut onnistua. Myöskin osallistuminen toiseen ryhmään olisi voinut olla viisaampi ratkaisu, koska yksin työn tekeminen ja mahdollinen ohjelmointi olisi ollut helpompi suunnitella ryhmässä.

Ajallisesti minulle tuli aika suuria ongelmia tehtävän kanssa, kun tietokoneeni virtalähde hajosi 23.11 enkä vajaaseen viikkoon päässyt käsiksi koulutehtäviini. Sain koneen korjattua 28.11, mutta sitten oli päällekkäin muita tehtäviä ja tentti. Lopputulos ei siis aivan onnistunut, mutta yritin parhaani käytössä olleeseen aikaan nähden.

Parannusehdotukset

Jos kurssilla on jatkossa enemmän opiskelijoiden tekemää koodia, voisin suositella ryhmiä pakolliseksi. Nyt kurssilla eteneminen oli mahdollista, kun käytössä oli myös opettajan tekemää ohjelmaa, mutta yksin koko ohjelman tekeminen voi olla aika vaativa prosessi. Ehkä tuntikohtaisesti voisi olla jokin valmis pohja, jonka päälle sitä omaa koodia kehitettäisiin, jos ohjelmointi ei ole onnistunut, niin seuraavalla tunnilla voisi ladata sitten toimivan korvaavan version. Näinhän tämä aika pitkälti toimikin. Oli mielestäni hyvä, että oppitunneilla käytiin läpi valmista koodia ja mitä sen eri osat ja arvot tekevät. Päädyin käyttämään suurimmaksi osaksi opettajan koodia. Omaa koodia tuli kokeiltua enemmän valmistusvaiheessa, mutta kun yritti saada toimivaa tuotetta valmiiksi, niin opettajan ohjelmat olivat huomattavasti varmempia ja toimivat paremmin, kuin omat yritykseni.

Johonkin voisi olla hyvä mainita, että kurssin suorittamiseen tarvitsee web-kameraa, joka minulta puuttui aluksi kokonaan, jäin tämän takia alussa hieman jälkeen, kun tehtäviä oli turhan tuntuista tehdä ilman kameraa. Huomiona kun kameraa testattiin ensimmäisen kerran päälle VLC:ssä en saanut kuvaa ulos, tämä korjaantui tällä: VLC mediasoittimessa kuvan tunnistus ei toiminut alkuun, löysin nopeasti etsimällä komentoriviin syötettävän koodinpätkän, jolla kameratuki saatiin asennettua VLC:hen. Käytetty koodi oli: sudo apt install vlc-plugin-access-extra

Liiketoimintamahdollisuuksien analysointi.

Liiketoimintamahdollisuuksia voisi löytyä useitakin. Ensimmäisenä tulisi mieleen rekisterikilven lukuohjelma, jota voisi käyttää esim. pysäköinninvalvonnassa tai parkkihallissa. Ohjelma voisi lukea rekisterikilven kameralla ja verrata sitä maksettuihin autopaikkoihin. Rekisterikilpi voisi olla sopivan kokoinen ja tarpeeksi samanlainen jokaisessa autossa, että sen tunnistamista varten ohjelman voisi kalibroida. Ongelmia voisi seurata esim. rekisterikilpien likaisuudesta, jolloin arvoja voisi olla vaikea lukea. Parkkihallissa kamera voisi lukea lippua ostamaan pysähtyneen auton rekisterikilven ja maksuautomaatti voisi saada rekisterikilven sitä kautta, jolloin asiakkaan ei tarvitsisi itse näppäillä rekisterinumeroa.

Ohjelmaa voisi muokata toimimaan myös konttien, junavaunujen yms. numeron tai tekstin lukemiseen, jolloin voitaisiin ohjata näiden lastaus oikeaan paikkaan. Kenties ohjelma voisi toimia osana jotain muuta kokonaisuutta, kuten varastorobottia, joka kuljettaa lähetykset oikealle paikalleen, tai osaa kerätä tavarat oikeasta kohteesta. Robotti siis varmistaisi lukemalla sijaintikyltistä onko kohde varmasti oikea.

Kenties järjestelmää pystyisi käyttämään kulunvalvonnassa, jolloin sillä voisi tarkastaa henkilökortteja ja onko heillä kulkulupaa tietyllä alueella. Henkilökortilla voisi olla sopiva muoto ja kuvasta voitaisiin lukea mahdollisesti nimi, työtehtävä ja turvallisuusluokitus. Kortin kelvatessa voitaisiin aukaista ovet suljetulle alueelle. Nopeasti kuvasta tekstin kerääminen voisi olla hyödyllistä esim. pdf tiedostoista, jolloin teksti saataisiin nopeasti käytettävään muotoon. Myös paperisten tekstien muuttaminen digitaaliseksi tekstiksi voisi olla hyödyllinen sovellus.