

**AUTOMAATSE UID TUVASTUSE LOOMINE JA
RAKENDAMINE TOOTMISES**

**CREATION AND IMPLEMENTATION OF AUTOMATIC UID
DETECTION IN PRODUCTION**

RAKENDUSKÕRGHARIDUSTÖÖ

Üliõpilane: Karl-Kristjan Kõverik

Üliõpilaskood: 154584NDFR

Juhendajad: Ago Rootsi; lektor
Ants-Oskar Mäesalu

Tartu 2020

AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud.

Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

"....." 202.....

Autor:

/ allkiri /

Töö vastab rakenduskõrgharidusetöö esitatud nõuetele

"....." 202.....

Juhendaja:

/ allkiri /

Kaitsmisele lubatud

"....."202... .

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina Karl-Kristjan Kõverik (sünnikuupäev: 27.12.1995)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Automaatse UID tuvastuse loomine ja rakendamine tootmises“, mille juhendajad on Ago Rootsi ja Ants-Oskar Mäesalu,

1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.

3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

¹*Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil.*

_____ (allkiri)

_____ (kuupäev)

LÕPUTÖÖ ÜLESANNE

Üliõpilane: Karl-Kristjan Kõverik, 154584NDFR

Õppekava, peeriala: NDFR14/15, Küberfüüsikaline süsteemitehnika

Juhendaja(d): Lektor, Ago Rootsi, +372 5662 9821

Külalisõppejõud , Ants-Oskar Mäesalu, +372 5300 5320

Lõputöö teema:

Automaatse UID tuvastuse loomine ja rakendamine tootmises

Creation and implementation of automatic UID detection in production

Lõputöö põhieesmärgid:

1. UID tuvastussüsteemi raamistiku loomine
2. UID tuvastusprogrammi loomine
3. UID tuvastussüsteemi testimine

Lõputöö etapid ja ajakava:

Nr	Ülesande kirjeldus	Tähtaeg
1.	Lõputöö eelkaitsmine	06.05.2020
2.	Lõputöö esitamine	25.05.2020
3.	Lõputöö kaitsmine	02.06.2020

Töö keel: Eesti keel **Lõputöö esitamise tähtaeg:** "25." mai 2020.a

Üliõpilane: Karl-Kristjan Kõverik ".....".....202....a
/allkiri/

Juhendaja(d): Ago Rootsi ".....".....202....a
/allkiri/

Ants-Oskar Mäesalu ".....".....202....a
/allkiri/

Programmijuht: Helle Hallik ".....".....202....a
/allkiri/

SISUKORD

LÜHENDITE JA TÄHISTE LOETELU	6
1 SISSEJUHATUS.....	7
2 LÄHTEÜLESANNE	9
2.1 Lakkimise tehnoloogiast	9
2.2 Ülesande sisu	9
2.3 Tulenevad lisanõuded.....	10
3 KASUTATAVAD UID TÜÜBID	12
4 ÜLESANDE JAOTUS ALAMÜLESANNETEKS	13
4.1 Sobiva tarkvara valik	13
4.2 Sobiva kaamera valik.....	13
4.3 Töökeskkonna negatiivse mõju vähendamine	14
5 TEHTUD VALIKUD	15
5.1 Tarkvara valik	15
5.2 Kaamera valik	16
6 LAHENDUS.....	18
6.1 Süsteemi lahenduse kavandamine	18
6.2 Süsteemi kavand	20
6.3 Tarkvara ülesehitus.....	20
6.4 Süsteemi ülesehitus	21
7 SÜSTEEMI TESTIMINE	22
7.1 Testimise eesmärk	22
7.2 Testimisele minevad aspektid	22
7.3 Testimise meetodika	23
7.4 Testimise tulemused	24
KOKKUVÕTE	27
SUMMARY.....	29
KASUTATUD KIRJANDUSE LOETELU	31
LISAD	32
LISA 1 – Dahua IPC-HFW5442E-ZE-2712 parameetrid	32

LÜHENDITE JA TÄHISTE LOETELU

HTTP – Hypertext Transfer Protocol (est. Hüperteksti edastusprotokoll)

IP – Internetiprotokoll (ingl. Internet Protocol, IP)

MES - Manufacturing Execution System (est. Tootmise täitesüsteem)

QR- Quick Response (est. ruutkood)

RTP – Real-time Transport Protocol (est. Realaja-transportiprotokoll)

RTSP – Real Time Streaming Protocol (est. Reaalajas Voogedastuse Protokoll)

TCP – Transmission Control Protocol (est. Edastusohje protokoll)

UDP – User Datagram Protocol (est. Kasutajadatagrammi protokoll)

UID – Unique identifier (est. Ainulaadne identifikaator)

USB – Universal Serial Bus (est. Universaalne järjestiksiin)

1 SISSEJUHATUS

Tehnoloogiavaldkondade tänane kiire areng loob üha uusi võimalusi juba olemasolevaid väljakujunenud tootmisprotsesse teha efektiivsemaks. Ka käesoleva lõputöö teema kujunes välja sellisest tootmisliini moderniseerimise vajadusest elektroonikat tootvas Firmas Enics Eesti. Lõputöö teema sai valitud Enics Eesti poolt välja pakutud teemast, mis tuli välja lakiliini kaizenis, kuidas parendada liinitegevust. Ülesandeks on luua süsteem, mis on võimeline tuvastama toodete UID-sid automaatselt. Süsteemi eelduseks on, et süsteem kasutaks kaamerat UID tuvastuseks.

Tööstuses ja mujal, samuti Enicis lakiliinil kasutatakse üldjuhul laser käsiskännereid UID-de lugemiseks. Järgnevalt proovin muuta seda tegevust rohkem automatiseeritumaks. Kui ära võtta käsitsi toote UID skaneerimine ning muuta skaneerimine automaatseks läbi kaamera. Selline tegevus muudaks liini protsessi kiiremaks ja lihtsamaks. Oma töös uurin erinevaid tehnoloogiaid, mis on võimelised automatiseerima vastavat protsessi.

Käesoleva töö ülesandeks on luua programm, mis on võimeline läbi kaamera sisendvoo reaajas tuvastama ja lugema UID ning salvestama UID. Samuti on lõputöö ülesandeks uurida erinevaid kaameraid ning luua raamistik, kuidas peaks kaamera ja programm omavahel töötama. Kuna Enicis lakiliinil asuvad UID erinevate nurkade all, peab süsteem olema võimeline lugema UID-sid erinevate nurkade alt. Selleks tuleb luua süsteem, mis koosneb mitmest kaamerast. Lõputöös tuuakse välja ja analüüsitakse ka erinevad aspekte, mis mõjutavad UID-de tuvastamist reaajas. Nendeks aspektideks on kaugus, kiirus, kaamera nurk, valgus, peegeldus ning kui UID ei ole täielikult näha.

Kaamera valik tänapäeval on väga lai. Kaameraid on erinevate liidestega ning parameetritega. Lõputöö jaoks sai valitud IP kaamera, kuna tootmishoones on IP kaamerat kõige kergem kasutada. IP kaamera kasutab kommunikatsiooniks arvutivõrku, mis on tänapäeval väga laialt levinud ja suuresti eeliseks teiste tüüpide kaamerate ees.

Lõputöö koosneb erinevatest osadest, mida järk-järgult lahendan. Esmalt panen paika lähteülesande ning sellest tulenevad nõuded loodavale süsteemile. Vastavalt nõuetele

loon süsteemi ning uurin, mida on vaja loodava süsteemi jaoks. Jagan lähteülesande erinevateks alamülesanneteks, et oleks kergem orienteeruda erinevates teemades. Järgnevalt uurin kirjandusest, kuidas on varem sarnastele ülesannetele lähenetud. Sellest tulenevalt ja oma teadmistele tuginedes valin vajalikud aspektid, millele tuginedes loodav programm luua. Uurin ka erinevaid kaameraid, et valida sobiv kaamera. Tuleb välja mõelda ka süsteemi kavand ehk milline on süsteemi ülesehitus. Süsteemi kavandi loomisel lähtun lähteülesandest ning kirja pandud nõuetest. Kõige viimase asjana toimub süsteemi testimine erinevate aspektide kohta. Testimise eesmärk on välja uurida, kas ja kuidas süsteem täpsemalt toimib ning leida murekohti ning neid võimalusel parendada.

Lõputöö raames tuuakse välja süsteemi raamistik. Tuuakse välja tähtsamad komponendid ning tehakse tähtsamate komponentidega teste, kinnitamaks nende sobivust. Lõputöö on loodud ettepanekuna ning lõplik süsteem tehakse valmis projekti järgmises faasis.

2 LÄHTEÜLESANNE

2.1 Lakkimise tehnoloogiast

Enics tegeleb elektroonikatoodete tootmisega. Enamik elektroonikaseadmeid monteeritakse tänapäeval trükkplaatidele. Trükkplaadid kaetakse lakikihiga, et kaitsta elektroonikaseadmeid erinevate kahjustavate tegurite eest. Kahjustavateks teguriteks on näiteks niiskus, tolm ja äärmuslikud temperatuurid. Laki pealikandmine tehakse tootmisliinil, mida Enicis nimetatakse "lakiliiniks". Lakiliini töö seisneb, et kõigepealt skaneeritakse toodete UID koodid käsiskänneriga MES'i. MES on süsteem, mis ühendab, jälgib ja juhib keerulisi süsteeme, mida on tarvis tootmisvoolude ja andmevoogude jälgimise jaoks. MES'i sisestatakse UID andmed ning MES logib etapi ära, et saaks järgmisesse etappi liikuda. Seejärel sisestatakse trükkplaadid, mis asuvad rakis, lakiliini alguses olevasse sissesöötusse. Sissesöödus liiguvad automaatselt trükkplaadid räkist konveierliinile. Konveierliinil liiguvad trükkplaadid läbi lakiliini, kust saavad lakikihi peale. Lõppu jõudes lähevad trükkplaadid uuesti räkki ning vajadusel lähevad lakiliini algusesse tagasi ning kaetakse trükkplaadi teine pool ka lakikihiga või trükkplaadid liiguvad uude etappi.

Käsitsi UID skaneerimine on ajakulukas, mille arvelt saaks muuta liinitööd kiiremaks ning vabastaks aega teistele tegevustele. Lakiliini saaks muuta automaatsemaks läbi automaatse UID lugemise, mis toimuks kaamerate vahendusel. Kokkuvõttes on plaatide lakkimine osa Enicis'i tootmisprotsessist ja skaneeritud UID on sisendinfo tootmisvoogude juhtimisel ja optimeerimisel. Seega töökiiruse suurendamine ja vigade vähendamine selles lõigus mõjub positiivselt kogu protsessile.

2.2 Ülesande sisu

Olemasoleva UID käsitsi skaneerimise tehnoloogiat saab oluliselt efektiivsemaks muuta üleminekuga automaatsele skaneerimisele. Tänu millele väheneb käsitöö ning vabaneb aega muudele töödele. Vähenevad inimlikud eksimused. Tagab kõrgema kvaliteedi ja suurema usaldusväarsuse. Lahendusi automaatsele skaneerimisele on mitmeid, kuid juba Enicis'i poolt antud ülesande sobiva lahenduse väljatöötamiseks

eeldas viidet, et see uus tehnoloogia võiks põhineda kaamera kasutamisel. Ülesandes seati väljatöötatavale süsteemile alljärgnevad üldised nõudmised:

- Süsteem on käivitatav kasutaja poolt.
- Süsteem suudab tuvastada erinevaid UID-sid kaamera sisendvoo vahendusel reaajas. Süsteem peab tuvastama EAN-8, EAN-13, Code-39, Code-128 tüüpi 1D-koode ning QR-kood ja DataMatrix tüüpi 2D-koode. Samuti peab omama võimalust edasiarendusteks ilma riistvara väljavahetamiseta ka teiste optiliselt loetavate UID-de lugemiseks.
- Süsteem suudab aru saada, mis andmeid sisaldab UID.
- Süsteem kontrollib kas UID-lt saadud andmed on juba andmebaasis või ei ole. Kontrolli eesmärk on tuvastada, et ühe UID andmete salvestamist ei toimuks mitu korda.
- Süsteem suudab salvestada tuvastatud UID andmed andmebaasi, kui süsteem on kontrollinud, et vastavaid andmeid ei ole andmebaasis.
- Süsteem suudab andmed salvestada MES'i.
- Süsteem suudab kuvada sisendvoo koos UID asukohtadega ning andmetega ekraanile.

2.3 Tulenevad lisanõuded

Enics'i tootmistehnoloogiast vaadeldava lakiliini juures lähtuvad veel alljärgnevad lisatingimused, mida ei saa vaadelda lähteülesandest lahus:

- Süsteem suudab tuvastada kas üks, kaks või rohkem UID-d korraga.
- Süsteem suudab dekodeeritud UID andmed salvestada andmebaasi. Andmebaasi salvestatakse UID andmed, UID tüüp ja salvestamise kellaaeg.
- Süsteemi andmevahetus peab ühilduma MES'iga.
- Süsteem suudab tuvastada UID-sid 30 x 40 sentimeetriliselt alalt.

- Süsteem suudab tuvastada UID kuni 50 sentimeetri kauguselt.
- Süsteem suudab tuvastada UID erinevate nurkade alt.
- Süsteem suudab vastu võtta korraga ühte kui ka rohkemat sisendvoogu ning analüüsida korraga mitme sisendvoo infot.
- Süsteem on suuteline tuvastama UID, mis on läbimõõdult 5 millimeetrit või suuremad.
- Kaamerate paiknemiskoht peab olema selline, et kaamerad ei segaks liini tööd, puhastamist ja hooldust.
- Kaameratel peab olema sobiv valgus parema tulemuse saamiseks UID-de tuvastusel.
- Kaamera ei tohi vibreerida.
- UID dekodeerimine eeldab head arvutusvõimsust.
- Süsteem on võimeline muutma kaadrisagedust vastavalt UID-de tuvastus ja analüüsi kiirusele.
- Süsteemis oleva kaamera pikslitihedus on 1920 x 1080 pikslit või suurem.

3 KASUTATAVAD UID TÜÜBID

UID-d on kodeeritud kas ühe- või kahemõõtmelisena, vastavalt 1D- ja 2D-koodid.

1D-koodid on olemuselt vöötkoodid ehk nad on moodustatud triipudest ja triipude vahelistest alades. 1D-koodide informatsioon on esitatud ribana ja informatsiooni loetakse sellelt piki ühte mõõdet ehk siis piki riba. Vastavalt kodeeritud andmetele, mida 1D-kood sisaldab, muutuvad triipude laius ning tihedus. Üldjuhul on 1D-koodid must-valged. 1D-koode kasutatakse väga laialdaselt erinevates eluvaldkondades. 1D-koode kasutatakse, näiteks transpordis, tööstuses, jaemüügis ja tervishoius. Levinumad 1D-koodide tüübid on EAN-8, EAN-13, Code-39 ja Code-128.

1D-koodi idee pärineb 1932. aastast, kui Wallace Flint pakkus välja, et jaemüügil võiks olla automaatne kassasüsteem. Kuid tollal jäi idee teostus tegemata, kuna Flinti lahendust peeti kalliks ja küllaltki teostamatuks. 1960-ndatel lõpus ja 1970-ndate alguses algas alles suurem 1D-koodide kasutuselevõtt kaubanduses ja tööstuses.[1]

1D-koodide lugemine toimub vöötkoodiskanneri abiga. Vöötkoodiskänner väljastab LED valgust või laserkiiri ning vöötkoodilt peegeldub tagasi valgus. Vöötkoodi valgedelt osadelt peegeldub tagasi rohkem valgust, kui mustadel osadelt ning selle järgi suudab vöötkoodiskänner tuvastada vöötkoodi mustade ja valgete triipude mustri. Mustri teisendab vöötkoodiskänner kümnendsüsteemis olevateks arvudeks ning saadaksegi tootekood.[2]

2D-koode saab jagada kaheks alamrühmaks, milleks on reakoodid ja maatrikskoodid. Reakoodide puhul on paigutatud mitu 1D-koodid üksteise alla ning koodi lugemine toimub reall maailmas. Maatrikskoodide puhul on andmed salvestatud koodi horisontaalselt ja vertikaalselt. Levinumad reakoodid on PDF417 ja CODE49 ning levinumad maatrikskoodid on QR-kood, Aztec ja DataMatrix. Võrreldes 1D-koodidega saab 2D-koodidesse salvestada rohkem infot väiksemale alale. 2D-koodid sisaldavad veaparandusi ehk kui millegi pärast ei ole näha tervet koodi, on kood endiselt loetav. 2D-koodid on üldjuhul loetavad tänu nurgas või keskel asuvale ruudule või ringile, mille põhjal saadakse aru, kus asub 2D-kood.[3]

4 ÜLESANDE JAOTUS ALAMÜLESANNETEKES

Lähteülesande paremini mõistmiseks on lähteülesanne jaotatav järgmisteks osaülesanneteks:

4.1 Sobiva tarkvara valik

Tänapäeval on laialdane valik erinevaid programmeerimiskeeli. Sobiva tarkvara valiku puhul tuleks esmalt uurida, kuidas on varem samalaadseid ülesandeid lahendatud. Uurida, kuidas ja kuidas oleks kasutatav või kohandatav minu ülesande puhul. Sellest lähtudes saab varakult juba nõuandeid kuidas jätkata. Programmeerimiskeele valikul tuleb samuti lähtuda, millise programmeerimiskeelega on loodavat rakendust mugavam ja lihtsam luua. Samuti tuleb jälgida enda eneseoskust, millise programmeerimiskeelega oled tuttav ning oskad kasutada. See aitab kaasa rakenduse kiiremale arengule ning vähemate probleemide tekkimisele. Programmeerimiskeele valikust lähtuvalt saab edasi vaadata erinevaid teeke, mida kasutada.

4.2 Sobiva kaamera valik

Kaamera valikul tuleb lähtuda kaamera võimekusest ning nende olulistest parameetritest. Valitav kaamera peab olema suuteline edastama videovoogu hea pikslitihedusega, et ka väiksemad detailid videovoos oleksid loetavad ja arusaadavad. Kaamera peaks suutma automaatselt fokuseerida, kuna kaamera peab olema suuteline fokuseerima liikuvat objekti. Kaamera puhul on oluline ka, et kaamera suudaks töötada ka väiksema valgustatuse juures nii, et täiendavad valgustid poleks vajalikud või siis lisavalgustid ei suurendaks tehase vastava piirkonna valgustatuse räägust. Kaamera valikul ei tohi unustada ka kaamera suurust ning lähtudes lähteülesande nõuetest kaamera sobivust.

4.3 Töökeskkonna negatiivse mõju vähendamine

Süsteem peab olema töökindel. Paigaldatud kaamera ja tooteliini vahele ei tohi sattuda esemeid, muidu ei suuda süsteem tuvastada UID-sid. Sellest lähtuvalt tuleb süsteem luua selline, et sinna vahele ei satuks vööraid esemeid. Samuti ei tohi kaamerad segada ka otseselt liini tööd. Parema tulemuse saavutamiseks ei tohiks olla koht, kust süsteem tuvastab UID-sid liigselt hämar. Otsene päikesevalgus samuti vähendab süsteemi töö kvaliteeti, kuna UID sildid võivad hakata otsese päikese käes läikima ning UID-de tuvastamine on mingil määral häiritud.

Välja toodud alamülesanded jagunevad kahte gruppi. Esimeseks grupiks, mis sisaldab kahte esimest eelnevalt toodud alamülesannet on spetsiifilisemad ja eeldavad muuhulgas ka tööd kirjandusega ja katsetusi:

- Sobiva tarkvara valik
- Sobiva kaamera valik

Teiseks grupiks on kolmandana välja toodud alamülesanne, mis on tüüpiline inseneriülesanne, mis on lahendatav asukohavaliku ja kaitsekatete kasutamise ja valgustustingimuste parandamisega:

- Töökeskkonna negatiivse mõju vähendamine

5 TEHTUD VALIKUD

Tehtud valikud on seotud etteantud lähteülesande nõuetega ning kirjandusest leitud informatsiooniga. Valikute puhul lähtusin oma praegustest oskustest ning teadmistest ja lisaks uuritud kirjandusest.

5.1 Tarkvara valik

Programmeerimiskeele valikul osutusid sobivateks Python ja Java, kuna mõlema keele kasutamisel on mul mõningane kogemus. Järgnevalt tuli võrrelda neid kahte erinevat programmeerimiskeelt, et välja valida kumba kasutada. Mõlemad programmeerimiskeeled on platvormiülesed ning neil mõlemal on laialdane valik erinevaid teke. Mõlemad eeltoodud programmeerimiskeeled on objektorienteeritud programmeerimiskeeled. Python'i ja Java kasutamine muudab süsteemi platvormiüleseks ehk süsteemi saab kasutada erinevate seadmetega. Erinevus kahe keele vahel on, et Python on dünaamiliselt tüübitud keel ja Java on staatiliselt tüübitud keel. Dünaamiliselt tüübitud keele puhul ei tule deklareerida muutuja tüüpi enne selle kasutamist ehk tüüpi kontroll toimub programmi töö käigus. Staatiliselt tüübitud keele puhul tuleb muutuja deklareerimisel määrata muutuja tüüp ehk tüüpi kontroll toimub koheselt.[4] Sellest tulenevalt on üldjuhul Python'iga kirjutatud programmid lühemad. Uue rakenduse tegemine ning ülesseadmine on Python'iga lihtsam ja kiirem võrreldes Java'ga. Java on kompileeritav keel ja Python on interpreteeritav keel, mille tõttu esimene on teisest oluliselt kiirem. Java sobib rohkem graafiliste kasutajaliidesega programmide, manussüsteemide ja veebirakenduste teenuste tegemiseks. Python on sobilikum andmetöötluste ja masinõpperakenduste tegemiseks.[9][10] Valituks osutus loodava rakenduse jaoks Python, kuna seda on minu arvates lihtsam kasutada ja ka mu varasemad kogemused Python'iga on suuremad.

Lähtuvalt valitud programmeerimiskeelest tuli järgnevalt valida teek, mis oleks võimeline pilte ja videoid töötlemiseks ja analüüsima. Uurides erinevaid Pythoni teke, jäid valikusse kaks teeki: OpenCV ja Scikit-image. Mõlemad teegid on mõeldud kasutamiseks pilditöötlustes ning on avatud lähtekoodiga teegid. Valituks osutus OpenCV, mis on rohkem suunatud reaalsajas arvutinägemisele. Kuna OpenCV-l on

suurem dokumentatsioon ning kogukond, siis sellest lähtuvalt sai ka otsustatud OpenCV kasuks.[5] OpenCV-d kasutatakse erinevate masinõppe ja arvutinägemise rakenduste tegemiseks. OpenCV loodi, et tagada ühtne infrastruktuur erinevate masinõppe ja arvutinägemise rakenduste jaoks.[6] Loodava rakenduse eesmärk on samuti reaajas tuvastada UID-sid sobib OpenCV väga hästi loodava rakenduse loomiseks.

Seejärel tuli uurida võimalusi, kuidas dekodeerida läbi OpenCV edastavast videovoost UID-sid. Selle ülesande lahendamiseks osutus sobivaks teek nimega Zbar. Zbar teek on avatud lähtekoodiga teek. Zbar on mõeldud QR- ja vöotkoodide tuvastamiseks piltidelt ja videotelt. Zbar teek loodi, et oleks stabiilne ja töökindel teek, mis oleks võimeline tuvastama erinevaid QR- ja vöotkoode.[7] Zbar ühildub OpenCV-ga, sellest tulenevalt jäigi valituks Zbar.

5.2 Kaamera valik

Riistvaralise poole pealt sai uuritud erinevaid kaameraid ning nende võimalusi ning kasutusmugavust. Sellest lähtuvalt sai valitud lõputöö tegemise jaoks IP kaamera.

Kaamera valikul lähtusin kasutusvõimalusest ja -mugavusest. Algsed valikuvõimalused olid veebikaamera, IP kaamera või Raspberry Pi kaamera. Veebikaamera all on mõeldud kaamerat, mis on arvutiga ühenduses läbi USB kaabli. IP kaamera all on mõeldud kaamerat, mis saadab videovoogu edasi läbi TCP või UDP/RTP võrgu. Raspberry Pi kaamera all on mõeldud kaamerat, mille ühendamiseks kasutatakse 15-soonelist lintkaablit. Valituks osutus IP kaamera, kuna IP kaameral on hulk häid omadusi, mis muudavad IP kaamera paremini kasutatavaks lähtudes lähteülesande nõuetest.

IP kaamera edastus toimub üle internetivõrgu, mis on tänapäeval olemas pea kõikjal. Saab kasutada nii võrgukaabli ühendust, kui ka Wi-Fi ühendust, mis annab IP kaamerale laiemad ja mugavamad kasutustingimused. Veebikaamera või Raspberry Pi kaamera kasutamine eeldaks lähedal asuvat Raspberry Pi-d või arvutit, kus peal oleks UID tuvastus programm. Raspberry Pi arvutusvõimsus on ka küllaltki väike, mille tulemusena võib ilmneda probleeme. Raspberry Pi kaamera väike objektiiv ei võimalda piisavat valgustundlikust. Sama viga on ka veebikaameratel. IP kaamera puhul võib

arvuti, milles töötab tuvastus programm olla kaugemal kaamerast võrreldes teistsuguste ühendustega kaamerate puhul.

IP kaamerate eeliseks on automaatava, mis hoiab pildi heleduse suhteliselt paigas, kui mingil põhjusel peaks keskmisest suurem valgus peale paistma. Veebikaamerate puhul automaatava on lisatud kallimatele mudelitele ning kusjuures ainult mõnele üksikule mudelile. IP kaameratel on tavaliselt võrreldes veebikaameratega parem optika ja resolutsioon.[8] Paljudel IP kaameratel on ka infrapunavalgustid, mis tagavad parema tulemuse hämaras keskkonnas.

Valitud IP kaameraks osutus Dahua IPC-HFW5442E-ZE-2712, millega testisin loodavat rakendust. Kaamera parameetrid on välja toodud tootelehel (Lisa 1) ning alljärgnevalt on välja toodud valitud kaamera olulisemad parameetrid:

Sensor	1/1.8" 4 Megapixel progressive scan CMOS
Maksimaalne resolutsioon	2688 × 1520
Elektroniline säriaeg	Auto/Manual, 1/3~1/100000 s
Minimaalne valgustus	0.0024 Lux/F1.8 (Color,1/3 s,30 IRE) 0.059 Lux/F1.8 (Color,1/30 s,30 IRE) 0 Lux/F1.8 (IR on)
Objektiiv	Motoriseeritud 2,7-12mm objektiiv
Optiline suum	4,4x
Pakkimisviisid	H.265+/H.265/H.264+/H.264/H.264B/H.264H/MJPEG(sub stream)
Kujutise dünaamiline diapason	BLC / HLC / WDR (140 dB)
Toide	12 V DC või PoE

Tabel 5.1 Dahua IPC-HFW5442E-ZE-2712 parameetrid

6 LAHENDUS

Lahenduse loomisel sai arvesse võetud lähteülesande nõudeid ning lisanõudeid. Katsete tulemused on samuti olulised, mida arvesse võtta ning tänu millele muuta süsteemi ülesehitust. Järgnevalt toon välja erinevad komponendid ning üleseehituse, milline peaks töötav süsteem olema.

6.1 Süsteemi lahenduse kavandamine

Süsteemi komponentideks on kaamerad, ekraan, UID tuvastus programm ja andmebaas. Süsteemi komponendid on valitud lähteülesande põhjal ning arvesse on võetud määratud nõuded. Süsteemi komponentideks olevad kaamerad tuli lähteülesandest, et loodav süsteem peaks kaamerate abiga tuvastama UID-sid. Ekraan sai valitud süsteemi puhul, et testimise käigus eelkõige kontrollida süsteemi töötamist. Ekraani hilisem lisamine on valikuline, et kontrollida üldist süsteemi tööd. Ekraan ei pea tingimata asetsema liini juures. Võib olla kuskil kaugemal, näiteks arvuti juures. Peakomponendiks süsteemis on UID tuvastus programm, mis täidab süsteemi põhiülesandeid ja funktsioone. Süsteemi disain on UID tuvastusprogrammi ümber ehitatud. Andmebaas on andmete säilitamiseks ja edastuseks. MES on andmete edasi saatmiseks ning toote järgmisesse etappi logimiseks. Alljärgnevalt on välja toodud süsteemi komponendid ning nende oskused ja omadused:

- **Kaamerad**

Annavad sisendvoo UID tuvastusprogrammile.

- **Ekraan**

Võtab vastu videovoogu UID tuvastusprogrammilt.

- **UID tuvastusprogramm**

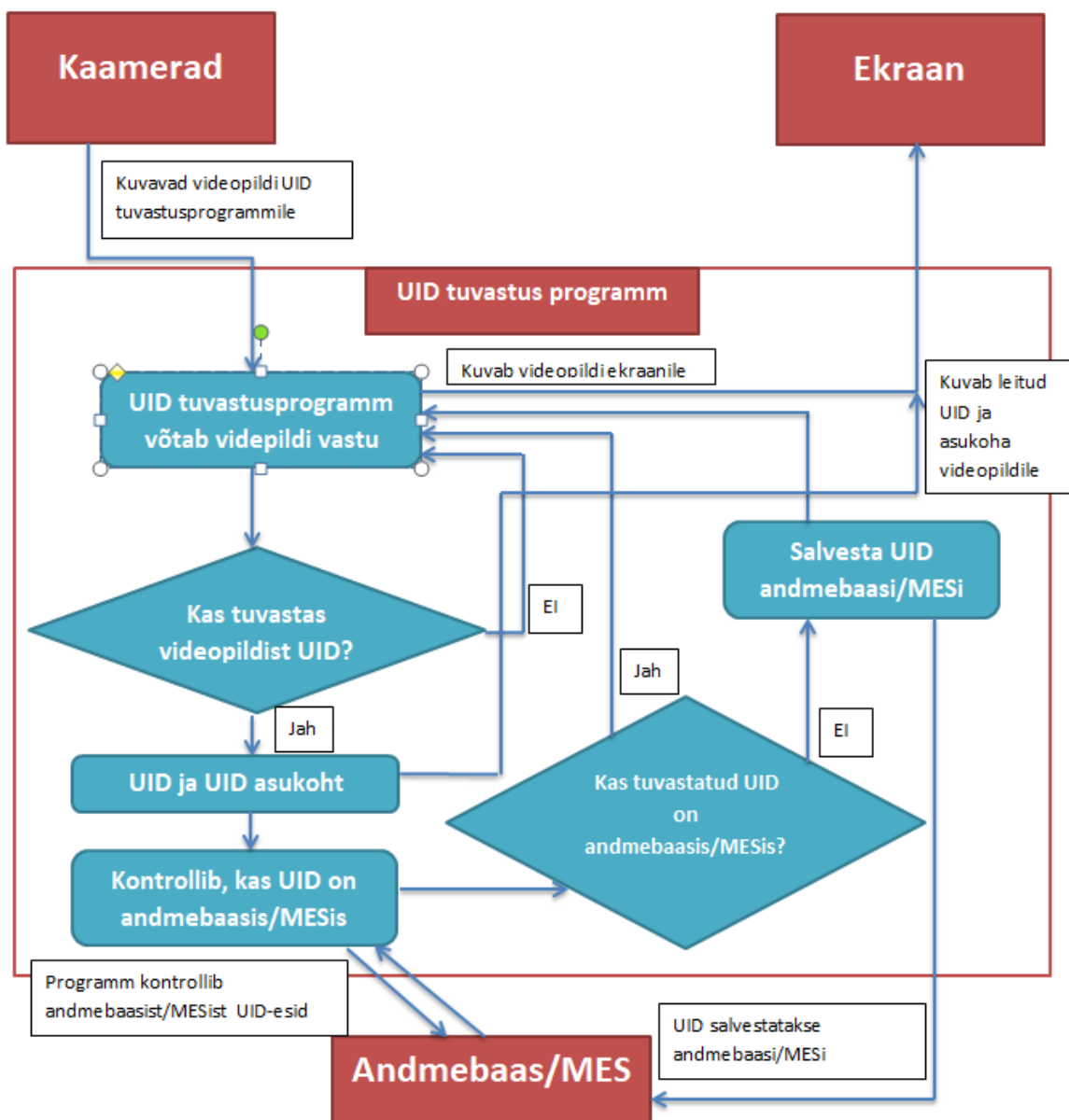
Tuvastab UID asukohta videovoos. Loeb ja dekodeerib videovoos oleva UID andmeid. Edastab videovoogu ekraanile ning kui leiab UID, sel juhul lisab videovoole UID asukoha ja andmed. Saadab UID andmed edasi andmebaasi/MES'i. Võrdleb leitud UID andmed andmebaasis/MES'is olevate andmetega ning teeb kindlaks, kas leitud UID andmed on juba andmebaasis/MES'is või ei ole. Kui andmeid ei ole andmebaasis/MES'is, salvestab programm andmed andmebaasi/MES'i.

- **Andmebaas**

Võtab vastu andmeid UID tuvastusprogrammilt.

- **MES**

Võtab vastu andmeid UID tuvastusprogrammilt.



Joonis 6.1 Joonis süsteemi toimimise kohta

6.2 Süsteemi kavand

Süsteemi kavandi loomist alustasin alamsüsteemide paika panemisest ning määrasin tingimused alamsüsteemidele. Süsteem koosneb neljast alamsüsteemist, milleks on kaamerad, UID tuvastusprogramm, ekraan ja andmebaas. Kaamera peab olema suuteline andma sisendvoogu UID tuvastus programmile. Süsteemi kõige tähtsamaks alamsüsteemiks on UID tuvastusprogramm. Millel on kõige rohkem nõutud funktsioone. Programm peab suhtlema kaameraga ning olema suuteline võtma kaameralt sisendvoogu vastu. Programm annab ekraanile edasi väljundvoogu, et oleks võimalik kontrollida programmi tööd. Programm peab tuvastama sisendvoost UID-sid. Sellest tulenevalt peab olema suuteline suhtlema andmebaasiga. Programm peab olema suuteline tuvastama andmebaasis olevat infot, et kontrollida kas leitud UID andmed on juba andmebaasi salvestatud. Kui ei ole, peab programm olema suuteline salvestama leitud info andmebaasi. Andmebaas peab olema võimeline võtma vastu programmilt infot.

Kui oli määratud alamsüsteemidele nõutavad funktsioonid, sai üldisele süsteemile määrata põhinõuded ja lisanõuded. Põhilisteks nõueteks on, et kasutaja peab süsteemi käima panema ning hooldama väliselt süsteemi. Põhilisteks lisanõueteks on, et süsteem peab olema võimeline tuvastama UID-sid, mis on 0,5 cm laiad või laiemad ning kaamera peab tuvastama UID-sid kuni 50 cm kauguselt.

6.3 Tarkvara ülesehitus

Lähteülesande jaoks loodud tarkvara sai kirjutatud Python'i programmeerimiskeeles. Tarkvara loomisel sai kasutatud OpenCV ja Zbar teeki. OpenCV teeki sai kasutatud kaamera sisendvoo vastuvõtmiseks ning kuvamiseks. Zbar teeki sai kasutatud videovoolt UID-de leidmiseks. UID andmete salvestamiseks kasutasin MongoDB andmebaasi.

Esmalt sai loodud programm, mis suudab tuvastada UID tavaliselt JPG või PNG formaadiga pildilt. Kui olin saanud valmis programmi, mis suudab tuvastada UID pildilt, asusin valmisolevat programmi muutma, et oleks võimalik tuvastada UID arvuti veebikaameraga. Selleks tuli muuta koodi, mitte avama faili, vaid vastu võtma kaamera sisendvoogu. Tänu openCV oli see küllaltki lihtne, kuna openCV on vastavad

käsud olemas. Kui suutsin programmiga näha arvuti veebikaamera pilti ning suutis tuvastada UID-sid, sai edasi liikuda järgmise etapi juurde.

Järgmisena tuli uurida, et kuidas vastu võtta sisendvoogu IP kaameralt. Selleks tuli muuta sisendvoo asukohta. Esmalt sai kasutatud telefoni, kuhu sai tõmmatud IP kaamera äpp. Mis muutis telefoni, justkui IP kaameraks. Sai kasutatud HTTP protokolliga, et edastada sisendvoogu programmile. Kui sai testitud, et saab kasutada IP kaamerat sisendvoona, sai soetatud IP kaamera. Kuna IP kaamera suutis edastada sisendvoogu läbi RTSP protokolliga, sai sisendvoogu vastu võetud RTSP protokolliga. RTSP kasuks sai otsustatud, sest vastava protokolliga eesmärk on edastada videovoogu reaajas. Selleks tuli ühendada IP kaamera samasse arvutivõrku, kus asub arvuti, millel on sisendvoo vastuvõtu ning UID tuvastusprogramm.

Seejärel sai programmile lisatud andmebaas, kuhu saaks salvestada UID-delt tuvastatud andmed, et testida programmi tööd. Kuna olen varem kasutanud MongoDB andmebaasi, otsustasin kasutada salvestamiseks endale tuttavat andmebaasi.

6.4 Süsteemi ülesehitus

Lähteülesande kirjelduse, nõuete ja testimise tulemuste põhjal tuleb süsteemi loomisel kasutada mitu kaamerat. Mitme kaamera eesmärk on tagada UID tuvastust erinevate nurkade alt, et ei jääks ühtegi UID tuvastamata. Ehk tuleks jälgida toote viite erinevate külge. Nendeks viieks küljeks oleks neli äärmist külge ning pealt vaade. Süsteem vajab ka võrgulüliti, et oleks võimalik ühest sisse tulevast võrgukaablist jagada võrgukaablit igasse kaamerasse. Kaamerad peaksid asetsema ennem toote lakkimise masinasse minemist. Sel juhul oleksid kaamerad kaitstud lakkimisest tulenevate aurude eest kaitstud. Vastav lahendus tagaks paremad valgustingimused ning vähesema vibratsiooni. Kaamerad tuleb paigaldada mõnekümne sentimeetri kõrgusele, et ei jääks liinil liikuvatele toodetele ette. Arvuti, kus peal on UID tuvastusprogramm, võib paigutada erinevatesse kohtadesse. Võib paigutada nii liini juurde, kui ka mõnda teise ruumi. See oleneb kasutaja jaoks kasutajamugavusest ja millised tingimused on tagatud.

7 SÜSTEEMI TESTIMINE

Kontrollimaks valitud lahenduse sobivust käesoleva ülesande lahendamiseks, viisin läbi rea teste. Testide käigus kontrollisin kaamera ja tarkvara sobivust, süsteemi tööd kaamera erinevate seadistuste puhul. Selgitasin välja kuidas üks või teine kaamera või tarkvara seadistus mõjutab süsteemi tööd, et selgitada välja sobivad seadistuse vahemikud ja kombinatsioonid.

7.1 Testimise eesmärk

Testide eesmärk on tuvastada optimaalne keskkond, kus kasutada loodud rakendust. Loodud tarkvaraga ning valitud kaameraga teha hulk erinevaid teste, et leida tegureid, mis mõjutavad UID leidmist sisendvoost. Teste seejärel analüüsida, et leida seoseid erinevate tegurite vahel.

7.2 Testimisele minevad aspektid

Testimisel kontrollisin seitset erinevat aspekti. Esimeseks aspektiks sai valitud, et testida milliseid UID-sid suudab süsteem tuvastada. Teiseks ja tõenäoliselt üheks olulisimaks aspektiks on kindlasti kaugus, kui kaugelt või ligidalt suudab programm tuvastada UID-d. Kusjuures vähem tähtsam ei ole ka kolmandaks aspektiks valitud UID tuvastamise kiirus. Kuna tooted liinil liiguvad, siis peab suutma tarkvara leida UID küllaltki kiiresti. Neljandaks aspektiks sai valitud kaamera nurk vastavalt UID-le. Kuna UID võib olla tootel viltuselt ning mitte 90 kraadise nurga all. Selleks sai proovitud erinevate nurkade alt leida QR- ja vöökoodi. Viierendaks aspektiks testisin, kuidas mõjutab valgustus UID leidmist. Samuti uurisin, kas on vahet otsesel ja hajunud valgusel. Kuuendaks aspektiks uurisin, kuidas mõjutab tulemust, kui UID asub läikival pinnal. Seitsmendaks ja viimaseks aspektiks on, et kui UID ei ole tootelt täielikult näha. Aegajalt satuvad UID-ed erinevatele pakenditele kortsutatult ning sai proovitud, kui hästi taolisel juhul suudab programm UID-ed tuvastada. Testimisel vaatlesin

tarkvara ja kaamerat koos, kuna kaamera töörežiimid ja programmi seadistused osutusid omavahel seotuks.

7.3 Testimise metoodika

Testide metoodika varieerus erinevate aspektide puhul. Vastavalt, mis oli oluline aspekti juures, seda sai testitud. Alljärgnevalt on välja toodud erinevad metoodikad erinevate aspektide kohta:

- Esimeseks aspektiks oli, milliseid UID-sid suudab süsteem tuvastada. Erinevate kooditüüpide välja selgitamiseks printisin välja erinevaid kooditüüpe ning seejärel proovisin süsteemiga tuvastada erinevaid kooditüüpe.
- Teiseks aspektiks oli UID tuvastuskaugus. Süsteemi tuvastuskauguse tegeliku vahemiku kindlakstegemiseks valisin 1D- ja 2D-koodid, mille printisin paberile välja ja proovisin tuvastada erinevatel kaugustel.
- Kolmandaks aspektiks oli UID tuvastuskiirus. UID tuvastuse kiiruse leidmiseks sai programmi lisatud ajaarvestus, kui kaua läheb UID tuvastuseks ning dekodeerimiseks. Järgnevalt testisin erinevaid kooditüüpe erinevates keskkondades leidmaks UID tuvastus kiirust.
- Neljandaks aspektiks oli UID tuvastus nurga alt. Kui suure nurga alt suudab süsteem UID tuvastada. Selleks tegin teste järjest nurka suurendades ning proovides, kui suure nurga alt suudab süsteem UID tuvastada.
- Viiendaks aspektiks oli valgus. Valguse mõju uurimiseks sai tehtud rida teste hämaras ja valgusküllases keskkonnas.
- Kuuendaks aspektiks oli UID tuvastus läikival pinnal. UID tuvastuse testimiseks läikival pinnal proovisin süsteemiga tuvastada UID-sid erinevatel läikivatel pindadel. Läikivateks pindadeks on läikivale paberile või kleepsule prinditud kood.
- Seitsmendaks aspektiks oli juhtumid, kui UID ei ole tervelt näha. Erinevaid UID-sid kortsutasin või voltisin kokku, et ei oleks näha UID tervenisti. Järgnevalt proovisin erinevates keskkonna tingimustes UID tuvastada.

7.4 Testimise tulemused

Teste tegin erinevate aspektide kohta ning igat aspekti sai vaadeldud eraldi, et leida kuidas iga aspekt mõjutab UID tuvastus, kuid samas jälgisin ka erinevaid tegureid koos. Teste sai tehtud dahua IPC-HFW5442E-ZE-2712 IP kaameraga. Testides kasutasin 1D-koodi puhul Code-39, Code-128 ja EAN-13 tüüpi koode ning 2D-koodi puhul QR-kood. Testimisel oli igat koodi neljas erinevas suurus. Koodide suurused olid 5 mm, 10 mm, 20 mm ja 30 mm. Testide puhul oli oluline leida erinevaid aspekte ning neid analüüsida, et leida võimalusi, kuidas muuta süsteemi, et tagada parem tulemus.

- Süsteem suutis tuvastada järgnevaid kooditüüpe: Code-39, Code-128, EAN-8, EAN-13, Interleaved 2 of 5(I 2/5) ja QR-kood. Alustasin erinevate kooditüüpide proovimist vastavalt, millised on kasutusel Enics'is ning testisin veel erinevaid kooditüüpe arenguperspektiivi silmas pidades.
- Teste sai tehtud 2566x1440, 1920x1080 ja 1280x720 resolutsiooniga. Testidel selgus, et kaamera puhul on oluline, kui suur on pikslitihedus, kuna suurem pikslitihedus tagab kaugemalt UID tuvastuse, mis oli ka ootuspärane. Kauguse puhul on oluline jälgida pikslitihedust kaameral. Sellest tulenevalt peaks pikslitihedus olema suurem, kui 1920x1080 pikslit. Alla selle väheneb tuvastus kaugus küllaltki palju.
- Testimisel selgus, et testide tulemused sõltuvad ka erinevatest kasutatavatest kooditüüpidest. Kauguse puhul on oluline ka Code-39 ja Code-128 kooditüüpide puhul, kui pikk on kood. Mida rohkem on sümboleid koodis, seda tihedamaks läheb vöötkood ning väheneb ka kaugus, kuni süsteem ei suuda üldse tuvastada UID.
- Süsteem suudab tuvastada 2D-koode kaugemalt, kui 1D-koode. Testi tulemusena selgus, et 1 cm lai QR-koodi suudab süsteem tuvastada 3 kuni 50+ cm. Alla 1 cm laiuse QR-koodide tuvastuskauguse vahemik on väiksem ning tihtipeale ei suutnud süsteem üldse üles leida väga väikest QR-koodi. Code-39 ja Code-128 puhul 1 cm laiusega ning 4 tähemärgi UID tuvastuskauguste vahemik oli 5 kuni 45 cm. Koodi pikenemine iga tähemärgi võrra vähendab tuvastuskaugust. EAN-13 koodi suurus on alati 13 sümbolit, seega peab olema EAN-13 tüüpi koodid laiemad, kui 1 cm. 1 cm juures ei tuvastanud süsteem UID, sest vöötkoodi mustad ja valged triibud on liialt

tihedasti. Üheks oluliseks faktoriks kauguse puhul on valgus, mis tuli testimisel välja. Hämarates tingimustes on UID tuvastamine kehvem, kuna väheneb kontrastsus ja suurenevad mürad.

- Kiiruse puhul on samuti oluline pikslitihedus. Mida rohkem on piksleid kaadris, seda kauem kestab kaadri töötlemine. Testimisel selgus, et kaadri töötlemisaega saab vähendada, kui muuta kaadrid halltoonideks(grayscale). Muutes värvilise pildi halltoonideks on vaja iga piksli kohta anda vähem teavet. Testimisel selgus, et 2560x1440 resolutsiooni juures on UID tuvastus- ja dekodeerimiskiirus ~0,28 sekundit. Kuid 1920x1080 resolutsiooniga oli UID tuvastus ja dekodeerimis kiirus ~0,19 sekundit. Kui eelnevalt muuta kaadrid halltoonideks vähenes UID tuvastus ja dekodeerimis kiirus 0,03 kuni 0,05 sekundi võrra. Testide käigus selgus, et tähtis on ka mitu kaadrit sekundis edastatakse. Kui töötlemise aeg on suur ning süsteem ei suuda töödelda piisavalt kiirelt kaadreid, jääb edastus reaajast maha.
- Testide tulemuste põhjal saab öelda, et kuni 45 kraadise nurga all suudab süsteem küllaltki hästi tuvastada UID-sid. 2D-koode mõnevõrra paremini, kui 1D-koode.
- Nagu eelpool sai juba mainitud, et valgustus mängis kauguse puhul rolli. Soovitavalt peab valgus olema loomulik, sellisel juhul oli tulemus tuvastusel kõige parem. Kui liigne valgus, näiteks päike paistab otse UID-le peale või taskulambi valgus on otse UID-le peale, oli juhtumeid, kus süsteem ei suutnud tuvastada UID.
- Lääkival pinnal olev UID otseselt ei mõjuta UID tuvastust. Kuid kui UID asub lääkival pinnal ning otsene valgus paistab UID-le, siis tekivad juhud, kus süsteem ei tuvasta UID.
- Kui 1D-koodi ei ole tervikuna näha, siis süsteem ei suuda tuvastada koodi andmeid. 1D-koodi puhul peab olema näha koodi igat triipu. Kuid süsteem ei pea nägema igat triipu täis pikkuses. Kui pole tagatud iga triibu täispikkuses kuvamine suudab süsteem ikkagi koodi andmed tuvastada. 2D-koodi puhul on asi parem ning testides ilmses, et süsteem suudab tuvastada poolikuid 2D-koode. 2. peatükis sai välja toodud, et 2D-koodidel on sisse kodeeritud veaparandus, kui mingil juhul ei peaks olema terve kood nähtav. Sellest tulenevalt on 2D-koodide tuvastus võrreldes 1D-koodide tuvastusega turvalisem.

Kokkuvõtvalt 2D-koodide tuvastus on parem, kui 1D-koodide tuvastus. 2D-koode saab kaugemalt ja turvalisemalt tuvastada. Tuvastusel tuleb jälgida erinevaid aspekte, et leida omale vajalik kesktee, sest erinevad asjad mõjutavad üksteist ning lõpptulemust. Testide tulemuste põhjal on võimalik parandada süsteemi ning lahendada leitud murekohti, et muuta süsteemi paremaks.

KOKKUVÕTE

Käesoleva töö ülesanne tuli Enics'ist, et muuta liinitööd kiiremaks ja automaatsemaks. Lähteülesandeks oli muuta UID lugemine automaatseks kasutades kaamerat, varasema käsiskänneri kasutamise asemel. UID-ga märgistatud tooted liiguvad liinil ning kaamera peab nende UID-d automaatselt tuvastama ja salvestama.

Esmalt panin kirja vajalikud nõuded süsteemi loomiseks. Nõuded olid vajalikud, et tagada läbimõeldud süsteem. Nõuded tulenesid lähteülesandest, vastavalt sellest mida oli vaja loodava süsteemiga tagada. Nõuded lähtusid ka süsteemi asukohast, milleks oli lakiliin. Loodav süsteem ei tohtinud segada ülejäänud liini tööd.

Järgnevalt uurisin, kuidas on varem taolisi süsteeme tehtud. Lähtuvalt uuritud kirjandusest ja oma teadmistest valisin erinevad osad, et luua vajaminev programm süsteemi tööks. Programmi kirjutasin Pythoni programmeerimiskeeles. Pildituvastuseks kasutasin OpenCV teeki ja UID-de tuvastamiseks Zbar teeki.

Uurisin ka erinevaid kaameraid, et leida loodavale süsteemile sobiv kaamera. Oluline oli, et oleks võimalik tagada süsteemile vajalikud nõuded. Kaamera valikul lähtusin erinevate kaamerate eelistest ja parameetritest. Valituks osutus IP kaamera, kuna IP kaamera valik tundus kõige mõistlikum.

Peale programmi algversiooni valmimist ning kaamera valikut testisin, kuidas toimib kogu süsteem. Selleks testisin erinevaid eelnevalt määratud aspekte, et kontrollida süsteemi tööd. Süsteemi töö kontrollimine on tähtis, et leida süsteemi töös erinevaid probleemkohti ning neid lahendada, et muuta süsteemi paremaks. Testimine on oluline ka, et uurida, kas süsteem suudab lähteülesandes määratud nõuded tagada. Sellest tulenevalt saab testimise abiga leida puudujäägid süsteemi töös.

Süsteemi suutis lähteülesandest tulenevad nõuded üldjuhul täita, kuid jäid mõned murekohad. Murekohaks jäi mõnevõrra tuvastuskaugus. Süsteem ei suutnud tuvastada väga väikeste mõõtmetega UID-sid. Selle probleemi lahendamiseks tuleks kasutada suurema resolutsiooniga kaamerat või täiustada süsteemi. Süsteemi täiustamise juures tuleb programmi muuta, et programm otsib UID-sid otsitavast alast väiksemate alade kaupa. See tagaks parema UID tuvastuse, kuna UID-le oleks tagatud suurem eraldus.

Selle töö raames lõin raamistiku, milline peaks välja nägema automaatne UID tuvastussüsteem ja millele peaks süsteemi loomisel tähelepanu pöörama. Süsteemi loomine ja edasiarendamine toimub projekti järgmises faasis.

SUMMARY

The task for this thesis came from Enics to make line work faster and more automatic. The initial task was to make the UID reading automatic with the use of a camera, instead of previously using a handheld scanner. Products with UID labels move along the line, and a camera must automatically detect and record their UIDs.

First, I wrote down the necessary requirements for the design and implementation of the system. The requirements were necessary to ensure a well-thought-out system. The requirements were derived from the needs set by the project's terms of reference. The requirements were also based on the location of the system, which was a lacquer line. The system to be created should not interfere with the rest of the line.

Next, I researched how such systems have been created in the past. Based on the researched literature as well as my own knowledge, I chose different parts to create the necessary program for the system to work. I wrote the program in the Python programming language. I used the OpenCV library for image recognition and the Zbar library for UID detection.

I also researched different cameras to find a suitable option for the system I was creating. It was important to be able to meet the necessary requirements for the system. The choice of the camera was based on the advantages and technical parameters of various cameras. An IP camera was chosen due to the fact that it seemed to be the most reasonable option.

Once the initial version of the program was completed and the camera was selected, I tested how the system worked as a whole. To do this, I tested various predefined aspects to check how the system worked. Checking the operation of the system was important to find and solve various problems in the operation of the system in order to improve the system. Testing was also important to examine whether the system could meet the requirements set out in the terms of reference. As a result, deficiencies in the operation of the system could be identified with the help of testing.

The system was generally able to meet the requirements of the terms of reference, but some concerns remained. The detection distance remained somewhat of a concern. The system could not detect UID-s with very small dimensions. Using a camera with a higher resolution or improving the system could solve this problem. When improving the system, the program has to be changed so that the program is

searching for the UID in the searched area by smaller areas. This would provide better UID detection as there would be greater resolution for the UID.

This thesis establishes a framework for an automatic UID detection system, and investigates the intricacies of implementing such a system. Implementation and further development of the system will take place in the next phase of this project.

KASUTATUD KIRJANDUSE LOETELU

[1] The history of the barcode [WWW]

<https://www.smithsonianmag.com/innovation/history-bar-code-180956704/>
(23.09.2015)

[2] Barcodes and barcode scanners [WWW]

<https://www.explainthatstuff.com/barcodescanners.html> (02.03.2020)

[3] Connecting the Dots: An Introduction to 2D Barcodes [WWW]

<https://www.masabi.com/2011/03/04/connecting-the-dots-an-introduction-to-2d-barcodes-3/> (04.03.2011)

[4] Magic lives here – Statically vs Dynamically Typed Languages [WWW]

<https://android.jlelse.eu/magic-lies-here-statically-typed-vs-dynamically-typed-languages-d151c7f95e2b> (09.09.2018)

[5] Scikit-image vs OpenCV [WWW] <https://medium.com/@hashinclude/scikit-image-vs-opencv-a2ce6e9b23d1> (02.03.2018)

[6] OpenCV: About [WWW] <https://opencv.org/about/>

[7] Zbar bar code reader: About [WWW] <http://zbar.sourceforge.net/about.html>
(15.07.2011)

[8] How is an IP Camera Different than a Web Cam? [WWW] <http://blog.dlink.com/ip-cameras-vs-webcams/> (19.10.2010)

[9] Python vs Java: Which is best? Code examples and comparison for 2019 [WWW]
<https://raygun.com/blog/java-vs-python/> (13.12.2018)

[10] Java vs Python: What's the Difference? [WWW] <https://www.guru99.com/java-vs-python.html>

LISAD

LISA 1 – Dahua IPC-HFW5442E-ZE-2712 parameetrid

Technical Specification					
Camera					
Image Sensor	1/1.8" 4Megapixel progressive scan CMOS				
Effective Pixels	2688(H) x 1520(V)				
RAM/ROM	1GB/128MB				
Scanning System	Progressive				
Electronic Shutter Speed	Auto/Manual, 1/3~1/100000s				
Minimum Illumination	0.0024Lux/F1.8 (Color,1/3s,30IRE) 0.059Lux/F1.8 (Color,1/30s,30IRE) 0Lux/F1.8 (IR on)				
IR Distance	Distance up to 50m (164ft)				
IR On/Off Control	Auto/ Manual				
IR LEDs	4				
Lens					
Lens Type	Motorized				
Mount Type	Board-in				
Focal Length	2.7mm~12mm				
Max. Aperture	F1.8				
Angle of View	H: 114°~47°, V:62°~26°				
Optical Zoom	4.4x				
Aperture Type	Auto				
Close Focus Distance	1.2m(3.94ft)				
DORI Distance	Lens	Detect	Observe	Recognize	Identify
	W	56m(184ft)	20m(66ft)	9m(30ft)	4m(13ft)
	T	125m(410ft)	50m(164ft)	22m(72ft)	11m(36ft)
Pan/Tilt/Rotation					
Pan/Tilt/Rotation Range	Pan:0°~360°; Tilt:0°~90°; Rotation:0°~360°				
Artificial Intelligence					
Face Capture	Support face image capture				
Face Attributes	Support 6 kinds of face attributes extraction: Age, Gender, Expression(Jolly/Surprise/Normal/Rage/Sad/Disgusted/Confused/Frightened), Glasses, Mouth Mask, Moustache & Beard				
Perimeter Protection	Tripwire, Intrusion(Human & Vehicle classification)				
People Counting	Line crossing people counting, Region people counting				
General Intelligence					
Event Trigger	Motion detection, Video tampering, Scene changing, Network disconnection, IP address conflict, illegal Access, Storage anomaly				
General Intelligence	Object Abandoned/Missing				
Video					
Compression	H.265+/H.265/H.264+/H.264/H.264B/H.264H/MJPEG(sub stream)				
Streaming Capability	3 Streams				
Resolution	2688×1520/2304×1296/1080P(1920×1080)/1.3M(1280×960)/720P(1280×720)/D1(704×576/704×480)/VGA(640×480)/CIF(352×288/352×240)				
Frame Rate	Main Stream: 2688×1520(1~25/30fps) Sub Stream: D1(1~25/30fps) Third Stream: 1080P(1~11fps)				
Bit Rate Control	CBR/VBR				
Bit Rate	H.264: 32~8192Kbps H.265: 19~8192Kbps				
Day/Night	Auto(ICR) / Color / B/W				
BLC Mode	BLC / HLC / WDR(120dB)				
White Balance	Auto/Natural/Street Lamp/Outdoor/Manual				
Gain Control	Auto/Manual				
Noise Reduction	3D DNR				
Motion Detetion	Off / On (4 Zone, Rectangle)				
Region of Interest	Off / On (4 Zone)				
Electronic Image Stabilization (EIS)	Support				
Smart IR	Support				
Defog	Support				
Digital Zoom	N/A				
Flip	0°/90°/180°/270°				
Mirror	Off / On				
Privacy Masking	Off / On (4 Area, Rectangle)				
Audio					
Compression	G.711a/ G.711Mu/ AAC/ G.726/G.723				
Network					
Ethernet	RJ-45 (10/100Base-T)				
Protocol	HTTP; TCP; ARP; RTSP; RTP; UDP; RTCP;SMTP; FTP; DHCP; DNS; DDNS; PPPoE; IPv4/v6; SNMP;QoS; UPnP;NTP				
Interoperability	ONVIF(Profile S/Profile G), CGI,Milestone,Genetec,P2P				
Streaming Method	Unicast / Multicast				
Max. User Access	20 Users				
Edge Storage	NAS FTP Mirco SD card 256GB				
Web Viewer	IE(versions newer than IE8), Chrome, Firefox, Safari(versions newer than Safari 12)				
Management Software	Smart PSS, DSS, P2P				
Smart Phone	iOS, Android				
Certifications					
Certifications	CE-LVD:EN60950-1 CE-EMC:Electromagnetic Compatibility Directive 2014/30/EU FCC:47 CFR FCC Part 15, Subpart B UL/CUL: UL60950-1 CAN/CSA C22.2 No.60950-1-07				