



TALLINNA TEHNIKAÜLIKOOL
INSENERITEADUSKOND
Elektroenergeetika ja mehhatroonika instituut

**TÖÖSTUSLIKE ELEKTRISÕIDUKITE
LAADIMISÜHENDUSE LOOMINE
MASINNÄGEMISE ABIL**

**AUTONOMOUS CHARGING PORT IDENTIFYING FOR
ELECTRIC VEHICLES**

BAKALAUREUSETÖÖ

Üliõpilane: Martin Reinok

Üliõpilaskood 185586EAAB

Juhendaja: Heigo Mölder, PhD

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.

"....." 2021

Autor: Martin Reinok

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

"....." 2021

Juhendaja: Heigo Mölder

/ allkiri /

Kaitsmisele lubatud

"....." 2021

Kaitsmiskomisjoni esimees

/ nimi ja allkiri /

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina _____ (autori nimi)

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

(lõputöö pealkiri)

mille juhendaja on

(juhendaja nimi)

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.

_____ (kuupäev)

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

LÕPUTÖÖ LÜHIKOKKUVÕTE

Autor: Martin Reinok

Lõputöö liik: Bakalaureusetöö

Töö pealkiri: Tööstuslike elektrisõidukite laadimisühenduse loomine masinnägemise abil

Kuupäev:
18.05.2021

50 lk

Ülikool: Tallinna Tehnikaülikool

Teaduskond: Inseneriteaduskond

Instituut: Elektroenergeetika ja mehhatroonika instituut

Töö juhendaja(d): Heigo Mölder

Töö konsultant (konsultandid):

Sisu kirjeldus:

Antud lõputöö eesmärk on luua masinnägemise süsteem robotiseeritud laadimissüsteemide automatiseerimiseks. Süsteemi rakendamiseks kasutatakse LED tuvastuselementi, mis paigaldatakse pistikupesa lähedale ning kaamera peab leidma elemendi ja selle koordinaadid edastama automaatika kontrolleriile. LED tuvastuselement võimaldab kaamera säriaja muuta väga kiireks, eraldades elemendi taustast ning seeläbi on masinnägemise tarkvara erinevate algoritmide rakendamine kiire ja töökindel, ka erinevate ilmastikuoludega. Tarkvara kasutab ringi- ja mallituvastus algoritmi, mille koostöö tulemusena on süsteem võimeline leidma defineeritud tuvastuselementi.

Töös valitakse masinnägemiseks ilmastikukindel kaamera ning vastavalt süsteemi tuvastuskauguse nõuetele valitakse sobiv objektiiv. Kaamerale disainitakse korpus koos sisseehitatud soojendusega, et tuvastamine oleks võimalik ka karmides väli tingimustes. Süsteemi töökindluse analüüsimiseks koostati videofailide logisüsteem, mis võimaldab algoritmide ja kaamera tööd analüüsida üle interneti.

Märksõnad: Masinnägemine, automatiseerimine, malli sobitamine, automaatne laadimissüsteem.

ABSTRACT

<i>Author:</i> Martin Reinok	<i>Type of the work:</i> Bachelor/Master Thesis
<i>Title:</i> Autonomous charging port identifying for electric vehicles	
<i>Date:</i> 18.05.2021	<i>50 pages</i>
<i>University:</i> Tallinn University of Technology	
<i>School:</i> School of Engineering	
<i>Department:</i> Department of Electrical Power Engineering and Mechatronics	
<i>Supervisor(s) of the thesis:</i> Heigo Mõlder	
<i>Consultant(s):</i>	
<i>Abstract:</i> <p>The purpose of this thesis is to design a machine vision system which automates an electrical charging system. The system uses a LED marker light, which is mounted close to the charger port. The camera system must locate the LED and send the coordinates to a logic controller. LED marker light was chosen, because it allows the camera to have short exposure time, which separates the marker from the background, making the machine vision algorithms quick and reliable, even with difficult weather. The developed software implements Hough circle and template matching algorithms, because of which the system can find the marker.</p> <p>According to the detection distance requirements, a waterproof camera and lens are chosen. Additionally, a housing with built-in heating element for cold weather was designed for the camera. To analyze the reliability of the system, a log system was implemented, which uploads videos recorded by camera to a webserver with necessary data represented on the video.</p>	
<i>Keywords:</i> Machine vision, automatization, template matching, automatic charging system.	

LÕPUTÖÖ ÜLESANNE

Lõputöö teema:	Tööstuslike elektrisõidukite laadimisühenduse loomine masinnägemise abil
Lõputöö teema inglise keeles:	Autonomous charging port identifying for electric vehicles
Üliõpilane:	Martin Reinok, 185586EAAB
Eriala:	Elektroenergeetika ja mehhatroonika
Lõputöö liik:	bakalaureusetöö
Lõputöö juhendaja:	Heigo Mölder, PhD
Lõputöö kaasjuhendaja: (ettevõtte, amet ja kontakt)	
Lõputöö ülesande kehtivusaeg:	18.05.2023
Lõputöö esitamise tähtaeg:	18.05.2021

Üliõpilane (allkiri)

Juhendaja (allkiri)

Õppekava juht (allkiri)

Kaasjuhendaja (allkiri)

1. Teema põhjendus

Autonoomsed laadimisühendus süsteemid on olulised kõigi elektri jõul töötavatele sõidumasinadele. Kuna metallilise kontaktühendusega pistiklaadijad on oluliselt efektiivsemad kui juhtmevaba induktiivlaadimine, siis on mõistlik arendada metallilise kontaktühendusega pistikroboteid autonoomse laadimisühenduse loomiseks.

Autonoomsed laadimisühendus süsteemid vähendavad oluliselt käsitsi tehtavat tööd suuremate masinate puhul, näiteks laevad, kus muidu peab pistikuid ühendama käsitsi. Muuhulgas lisavad autonoomsed laadijad mugavust elektriautode ja muude elektrimasinate laadimiseks.

2. Töö eesmärk

Töö eesmärgiks on luua masinnägemise juhtsüsteem, mis leiab laadimispistiku üles mistahes masinal/tootel ning väljastab vastavad koordinaadid automaatika kontrolleri pistikühenduse loomiseks.

3. Lahendamisele kuuluvate küsimuste loetelu:

- Kuidas on konkurendid teostanud autonoomset pistikute tuvastamist
- Miks kasutada masinnägemist võrreldes teiste alternatiivsete süsteemidega?
- Kuidas leida masinnägemise abil üles pistikupesa erinevatel masinatel?
- Millist tarkvara kasutada ning milline kaamera valida masinnägemise süsteemiks?.

4. Lähteandmed

Lähteandemetena on plaanis kasutada erinevate elektrisõidukite pistikühendusi. Vaadelda erinevaid ühendusvõimalusi kasutades passiivset tuvastuselementi või LED valgusallikat. Tegemist on konkreetse robotsüsteemiga, millele soovitake masinnägemisega täpset juhtsüsteemi.

5. Uurimismeetodid

Vastavalt lähteandmetele ning soovitud süsteemi lõpptulemustele koostatakse automaatika süsteem ning valitakse välja kõik vajalikud komponendid skeemis. Seejärel arendatakse masinnägemise tarkvara, et tuvastussüsteem toimima saada. Masinnägemise metoodika põhineb suurel määral katsetamisel ja testimisel, et parameetrid saaks paika, kuid enne katsetamist tuleb valida sobiv kaamera ja sobiv tarkvara kirjutamise platvormi. Plaanis on kasutada masinnägemiseks kohandatud miniarvutit, OpenCV tarkvara platvormi, kirjutatud kasutades Python programmeerimiskeelt. Täpne struktuur ja valitud komponentide blokid selguvad lõputöö tegemise käigus.

6. Graafiline osa

- Erinevate kaamerate võrdlus
- Erinevate masinnägemise arvutite võrdlus
- Valitud kaamera olulisemad parameetrid, kaamera korpuse ja kinnituse joonised
- Automaatika skeem juhtsüsteemi toiteks ning suhtluseks
- Masinnägemise pildid, marker tuvastamise tulemused

7. Töö struktuur

Esiialgu on paika pandud lõputöö põhistruktuur.

Täpsemad alampealkirjad selguvad lõputöö tegemise käigus.

1. Lõputöö eesmärk

2. Ülevaade maailmas varem tehtud sarnastest juhtsüsteemidest ja rakenduste
3. Erinevate tehnoloogiate võrdlus – miks valida masinnägemine
4. Valitud riistvara, automaatika skeemi koostamine ning joonised
5. Tarkvara selgitamine, masinnägemise süsteemi filtrid, kommunikatsioon.
6. Testimine, tulemused, kokkuvõtte
7. Edaspidine arendustöö

8. Kasutatud kirjanduse allikad

- OpenCV (Open Source Computer Vision Library) kodulehekülg. Kasutatud 29.10.2020, <https://opencv.org/>
- Stack Overflow foorum tarkvarapõhiste küsimuste lahendamiseks. Kasutatud 29.10.2020, <https://stackoverflow.com/>
- GigE Vision standardi selgitus ja dokumentatsioon. Kasutatud 29.10.2020 <https://www.visiononline.org/vision-standards-details.cfm?type=5>
- NVIDIA masinnägemise arendussüsteem Jetson Nano. Kasutatud 29.10.2020 <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- GigE standardile vastava kaamerate tootja LUCID Vision Labs, Inc. Kasutatud 29.10.2020, <https://thinkLUCID.com/>

9. Lõputöö konsultandid

10. Töö etapid ja ajakava

- Masinnägemise kontseptsiooni ja automaatika süsteemi projekteerimine (15. september)
- Masinnägemise ja automaatika süsteemi katsetused (1. detsember – 15. jaanuar)
- Kirjanduse läbitöötamine ning teoreetilise osa kirjutamine (15. jaanuar)
- Analüüsi ja kokkuvõtte koostamine (1. märts)
- Töö esimene versioon valmis (1. aprill)
- Juhendajale läbilugemiseks saatmine (2. aprill)
- Paranduste sisseviimine (30. aprill)
- Juhendajale teiseks läbilugemiseks saatmine (1. mai)
- Töö lõplik versioon valmis (15. mai)

Kinnise kaitsmise ja/või lõputöö avalikustamise piirangu tingimused formuleeritakse pöördel.

SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE	3
LÕPUTÖÖ ÜLESANNE	6
EESSÕNA	11
Lühendite ja tähiste loetelu	12
SISSEJUHATUS	13
1. ÜLEVAADE SAADAVAL OLEVATEST PISTIKUPESA TUVASTAMISMEETODITEST	15
2. MASINNÄGEMINE	17
2.1 Masinnägemine	17
2.2 Hough operaator	17
2.3 Malli tuvastamine piltidelt	18
2.4 Valgustus ja kaamera säriaeg	20
3. RIISTVARA VALIK	22
3.1 Masinnägemise arvuti valik	22
3.2 Kaamera valik	23
3.2.1 Kaamera	24
3.2.2 GigE standard	25
3.3 Objektiivi valik	25
3.3.1 Objektiiv	27
3.4 Andmeside komponendid	28
3.4.1 PoE toitesead	28
3.4.2 Võrgujaotur	29
3.5 Kaamera küttesüsteem	29
3.6 Korpuse disain	31
3.7 Elektroonika skeem	32
4. TARKVARA	34
4.1 Ülevaade tarkvarast, konfigureerimisvõimalused	34
4.2 Kaamera kommunikatsioon	35
4.3 Omron FINS suhtlus	36
4.4 Põhifunktsionaalsus	37
4.5 Serverisse logifaili laadimine	39
5. TULEMUSED	41
KOKKUVÕTE	43
KASUTATUD KIRJANDUSE LOETELU	44
LISAD	48
Lisa 1 Kaamera seadete konfiguratsioon	49
Lisa 2 Lõik parameetrite seadistuse moodulist	50

EESSÕNA

Käesolev lõputöö on teostatud reaalse seadme väljatöötamiseks. Konfidentsiaalsuslepingu tõttu ei ole võimalik testimisperiodist konkreetseid näiteid tuua, kuid seadme põhimõte, komponendid ning seadistamise käik on töös esitatud.

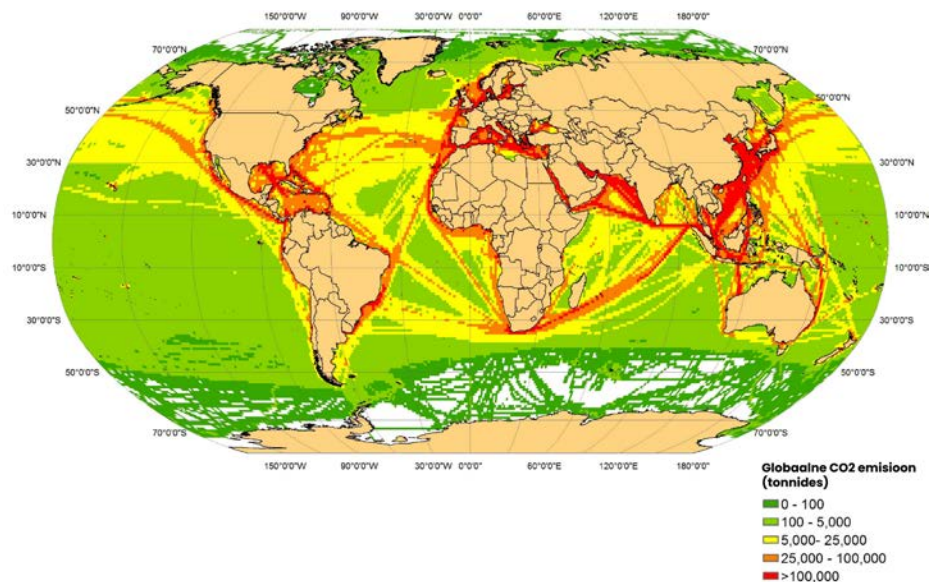
Täna juhendajat Heigo Mõlder abi ja juhendamise eest.

Lühendite ja tähiste loetelu

API	Tarkvaraseadistuskomplekt (<i>ingl k Application Programming Interface, API</i>)
CHT	Hough ringi operaator (<i>ingl k Circle Hough Transform, CHT</i>)
FINS	Kontrolleri kommunikatsioonimeetod (<i>ingl k Factory Interface Network Service</i>)
fps	Kaadrisagedus (<i>ingl k Frames per second, fps</i>)
HT	Hough operaator (<i>ingl k Hough Transform, HT</i>)
IDE	Tarkvaraarendus keskkond (<i>ingl k Integrated Development Environment, IDE</i>)
IP	Internetiaadress (<i>ingl k Internet Protocol address, IP</i>)
L4T	Linux operatsioonisüsteem Jetson seeria arvutitele (<i>ingl k Linux for Tegra, L4T</i>)
PLC	Tööstuskontroller (<i>ingl k Programmable Logic Controller, PLC</i>)
PoE	Toide üle Etherneti (<i>ingl k Power Over Ethernet, PoE</i>)
SDK	Tarkvaraarenduskomplekt (<i>ingl k Software Development Kit, SDK</i>)

SISSEJUHATUS

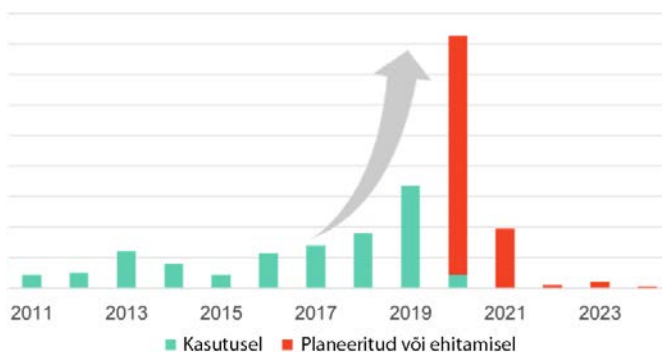
Toimuvast kaubavahetusest ca. 80% transporditakse mere kaudu erinevate riikide laevastiku abil [33]. Kuna suurem osa laevastikust tarbib diiselkütust, paiskavad laevad mitu miljonit tonni CO₂ kasvuhooonegaasi aastas ning aastal 2015 olid veeteedel CO₂ emissioonid ligi 932 miljonit tonni [38].



Joonis 0.1 Maailma veeteede CO₂ emissioonid aastal 2015 [38]

CO₂ emissioonide vähendamiseks on hakatud uurima hübriid ja täiselektriliste laevade ja praamide võimalust, mis Euroopas on õnnestunud mitmel juhul, näiteks Norra Ampere ning Eesti Tõll, eelkõige lühemate vahemaade tõttu kui mujal maailmas. Euroopas liikleb üle tuhande praami, millest enamus töötavad fossiilkütuste abil, kuna elektriakude energiatihedus on ligi 100 korda väiksem ning seetõttu väheneb ka läbitav distants. Täiselektriliste transpordivahendite alternatiivina saab ehitada hübriidi, mis oluliselt vähendab õhku paisatava CO₂ hulka ning vähendab ka fossiilkütuse kasutamise hulka [37]. Uute hübriid ja täiselektriliste laevade ehitus on märgatavalt tõusnud, visuaalselt kujutatud tabelis 0.1 [39].

Tabel 0.1 Hübriid ja täiselektriliste laevade ehituse planeerimine (2019) [39]



Täiselektriliste ja hübriid laevade laadimine toimub peamiselt kaldavõrgus, mis annab võimaluse kasutada lähedal olevate päikese- ja tuuleparkide energiat, vähendades veelgi fossiilikütuste tarbimist. Sellegipoolest ei saa elektrilaevad, või muud elektrilised sõidukid, tiheda graafiku ja lühikese sadamas viibimise aja tõttu laadida elektriakut täielikult täis. Alternatiivina laetakse päeval iga sadamas viibitud kaubalaadimise ajal niipalju kui võimalik ning pikemal pausil, harilikult öösel, laetakse aku täielikult täis. Lühikeste pauside tõttu on laadimiseks vajaolevate võimsuste ülekandmiseks vaja väga suuri kaableid ja pistikut. Lisaks sõidavad praamid päevas mitu korda edasi-tagasi, mis tähendab igal korral laadija pesasse ühendamist ja lahti ühendamist. Käsitsi pistiku ühendamine võtab palju aega ning on monotoonne töö. Selle asemel saab ehitada automatiseeritud laadimissüsteemi, mis autonoomselt leiab sõidukil oleva laadimispesa üles ning loob laadimisühenduse.

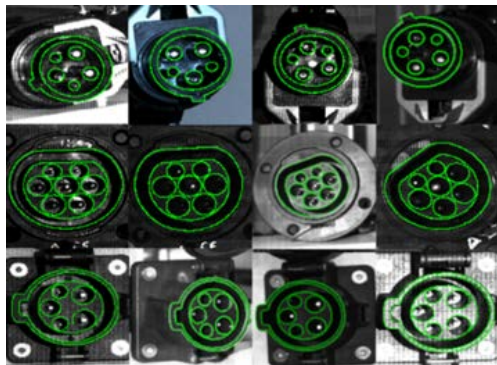
Antud Bakalaureuse töö eesmärk on luua masinnägemise süsteem, mis on võimeline leidma ja edastama pistikupesa asukohta tööstuslikule automaatika kontrolleri (PLC). Masinnägemise kaamera peab olema kinnitatav olemasolevate pistiku- ja kontrollersüsteemide külge, muutes kaamerasüsteemi universaalseks, mida on võimalik kasutada ka teistsuguste pistikusüsteemidega. Pistikupesa leidmiseks masinnägemise rakenduse abil kaalutakse olemasoleva tarkvaraga kaamerasüsteemide kasutamist, või isearendatud kaamerasüsteemi tarkvara. Sobiv lahendus peab olema paindlik ja töökindel ning omama ka edaspidist täiendamise ja kohandamise võimalust.

1. ÜLEVAADE SAADAVAL OLEVATEST PISTIKUPESA TUVASTAMISMEETODITEST

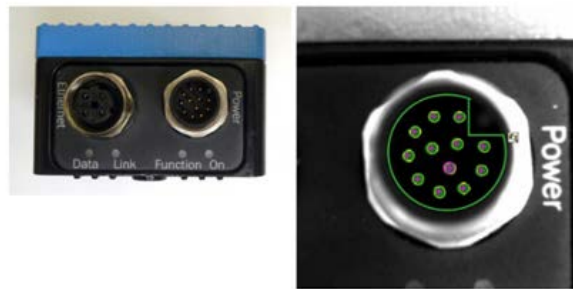
Suuremad elektrisõidukid nagu hübriid ja täiselektrilised praamid, ning tulevikus ka lennukid [2], nõuavad laadimiseks väga suuri võimsuseid, kuna laadimisaeg sadamates või jaamades on limiteeritud, mitmel juhul alla 10 minuti. Laadimiseks on eraldatud aeg, mis kulub kauba peale ja maha laadimiseks [34]. Turul olevate elektrilaeva laadijate laadimisvool on kuni 5760 A ning 1000 VDC [1]. Selliste võimsuste ülekandmiseks on vajalikud ka suured tööstuslikud laadijad, mille käsitsi ühendamine ei ole praktiline, sest laadimisühendust peab sooritama mitu korda päevas [34]

Laadimispistiku pesade tuvastamiseks saab üldiselt kasutada kas ise disainitud malli sobitamise algoritmi [3], turul kättesaadavaid malli sobitamise meetodeid [4] või kasutades lisaobjekte mis on masinnägemise algoritmile selgeks õpetatud [5].

a)



b)



Joonis 1.1 a) malli sobitamise algoritm [3] ning b) SICK Inspector kaamerasüsteem [4]

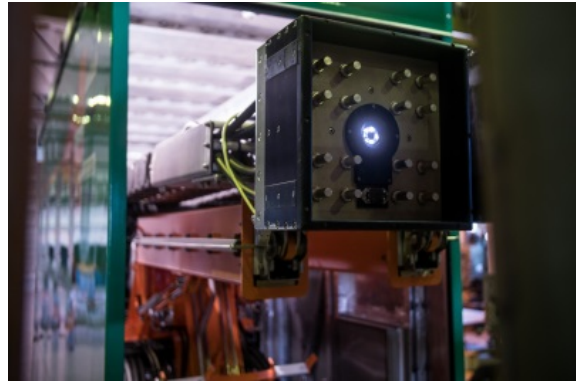
Uuritud turul kättesaadav tuvastussüsteem SICK Inspector on mõeldud rohkem staatiliseks objektituvastuseks ning kui sisse õpetatud objekti suurus muutub rohkem kui 20%, ei ole võimalik objekti enam tuvastada [4]. Seetõttu on mõistlikum masinnägemise tarkvara kirjutada ise, et töökindluse ja funktsionaalsuse üle oleks täielik kontroll.

Peamine takistus malli sobitamise algoritmide kasutamisel tööstuslike laadijatega on laadijapistiku pesa suurus [5]. Suuruse tõttu ei pruugi kogu pistikupesa pindala kaamerale nähtav olla kogu ühendamisprotsessi käigus. Alternatiivina saab kasutusele võtta lisaobjektid, näiteks passiivne või aktiivne suure kontrastiga täpp või LED valgustus, mis on paigaldatud pistikupesa külge (joonis 1.2). Sellisel juhul otsib kaamera vaid täppi ning edastab täpi asukoha laadimiskäpa controlleritele.

a)



b)



Joonis 1.2 a) Mobimar Nector pistikupesa ning b) Mobimar pistik [5]

Lisaobjektiga süsteemi lahenduseks saab kaamera paigaldada pistiku keskele, kuid selleks tuleb välja arendada uus pistik ning see võib minna kalliks maksma. Ise väljaarendatud kaamerasüsteemi lahenduse eesmärk oleks sobida ka standardsetele juba turul olevatele pistikutele ning mitte vähendada töökindlust või pistikupesa leidmise tõenäosust. Sellisel juhul saab kaamera monteerida kas pistiku alla, peale või külgedele ning tuvastusobjekti paigaldada samuti pistikupesast samavõrra kaugemale, et pistiku ühenduse puhul oleks kaamera ja tuvastusobjekt kohakuti.

Kaameraga pistikupesa tuvastusele on ka alternatiivseid meetodeid kasutusel, näiteks Norra täiselektrilise praami Ampere laadija FerryCHARGER (joonisel 1.3a). Kaamerat FerryCHARGER-i pistikul, ega pesal ei ole. Sarnast süsteemi FerryCHARGER-ile kasutab ka Cavotec-i automaatne ühendussüsteem (joonisel 1.3b), kuid mõlema lahenduse tehnilisele infole ligipääs puudub.

a)



b)



Joonis 1.3 a) *FerryCHARGER* laadija *Stemmann-Technik* tootjalt [35] ning b) *Cavotec*-i pistikupesa praamil [36]

2.MASINNÄGEMINE

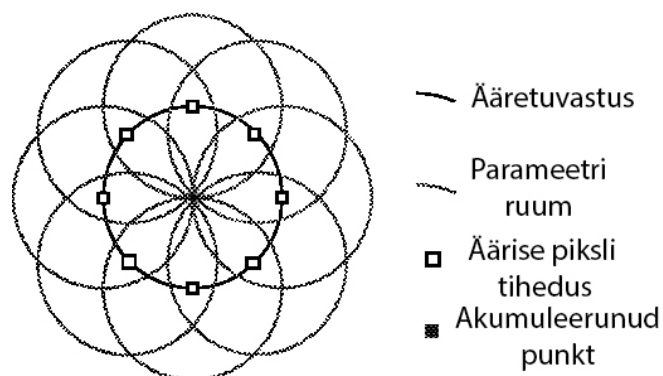
2.1 Masinnägemine

Masinnägemise eesmärk on üles ehitada arvutuslike mudeleid lähtuvalt inimeste nägemisest ning luua autonoomseid süsteeme, mis suudavad lahendada samu ülesandeid mis inimesedki, enamuses paremini ning jätkusuutlikumalt. Kaamerast saadud andme massiivide analüüsimisel saab eristada segmentatsiooni, ääriseid, nurki ja jooni [14]. Masinnägemise algoritmid on tehtud kasutatavaks ning kättesaadavaks avatud lähtekoodiga teekide nagu OpenCV [21], VLFeat [15], Dlib [16] jpm abil.

2.2 Hough operaator

Hough operaator (HT) on üks esimesi ja vanimaid masinnägemise algoritme, mis võimaldab tuvastada nii standardseid [18], kui ka ebastandardseid kujundeid [19]. HT põhiprintsiip on kaameralt saadud andmestruktuuride projekteerimine parameetruumis, mis esindab võimalikke kujundi asendeid kaamera pildil. Seejärel otsitakse akumulereunud punktikogumid, mis vastavad otsitava kujundi kõige tõenäolisemale asukohale kaamera pildil [18].

Hough' ringi operaator (CHT) on tuntud ja laialdaselt kasutusel olev ringituvastus algoritm [20], mida saab rakendada ka OpenCV teegi abil. CHT eesmärk on leida akumulereiva algoritmi abil kindla raadiusega ringikujulisi formatsioone kaameralt saadud pildil. Ringikujuliste objektide leidmiseks eristab CHT algoritm John Canny ääretuvastuse meetodi abil pildil olevad äärised [21], seejärel akumulereiv algoritm funktsioon joonistab kasutaja poolt määratud tihedusega (äärise piksli tihedus) äärele uusi ringe. Lahendatud akumuleraatori kõige tihedam punkt on ringi keskpunkt. Meetod on visuaalselt kujutatud joonisel 2.1.



Joonis 2.1 CHT akumulereiv algoritm tuvastab ringi keskpunkti [40]

Kui ringi suurus ei ole teada, lahendab CHT algoritm järk-järgult kasutaja ettemääratud parameetritega kõik võimalikud raadiused läbi ning väljastab suurima akumulaatori andmed või ei väljasta midagi, kui pildil ringe ei ole [21]. OpenCV teegis on CHT rakendamiseks kasutusel kuus kasutaja poolt seadistavat parameetrit, mis on esitatud tabelis 2.1

Tabel 2.1 OpenCV teeki rakendatud CHT parameetrid [21]

<i>dp</i>	Pöördvõrdeline akumulaatori resolutsiooni suhe
<i>minDist</i>	Minimaalne vahemaa iga uue ringi vahel
<i>lisa1</i>	J. Canny meetodi parameeter äärste tuvastamiseks
<i>lisa2</i>	Akumulaatori keskpunkti tuvastamise lävis
<i>minRadius</i>	Minimaalne tuvastatava ringi raadius
<i>maxRadius</i>	Maksimaalne tuvastatava ringi raadius

Dp parameeter määrab, missuguse resolutsiooniga sisestatakse lähtepilt CHT algoritmi, väärtus 1 puhul sisestatakse originaal resolutsioon [21]. Mida suurem on dp number, seda väiksemaks muutub algoritm käsitletav pilt, mis kiirendab algoritmi, kuid tulemused ei pruugi olla korrektsed.

MinDist parameeter määrab pikslites, kui suur vahe saab olla iga leitava ringi keskpunktide vahel. Kui kaks ringi satuvad lähemale, kui parameetri väärtus siis väljastatakse vaid suurem ring [21].

Lisa1 on OpenCV CHT rakenduse meetodi spetsiifiline parameeter, mis määrab J. Canny meetodi sisendid äärste tuvastamiseks [21].

Lisa2 on OpenCV CHT rakenduse meetodi spetsiifiline parameeter, mis määrab akumulaatori punkti lävise, millest määratakse kas tegemist on ringiga või mitte. Mida väiksem on väärtus, seda rohkem hajunud saab akumulaatori keskpunkt olla ning seda rohkem võib algoritm tuvastada ringe, mis ei ole tegelikult ringid. Joonisel 2.1 on ringi keskpunktis akumuleerinud funktsiooni tulemus.

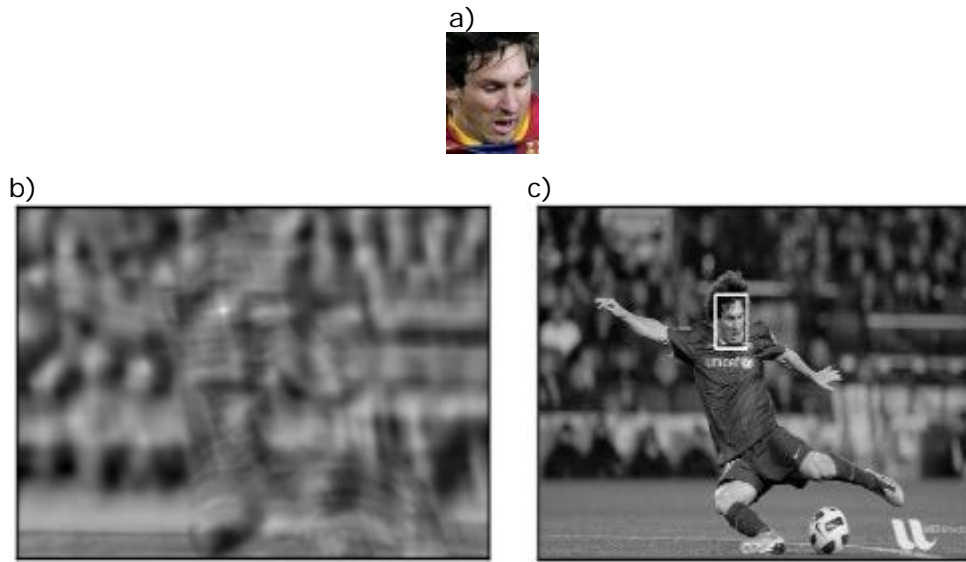
minRadius ja **maxRadius** määravad tuvastatava ringi suuruse pikslites. Algoritm lahendab järk-järgult kogu raadiusevahemiku läbi ning väljastab kõige suuremad akumuleerimistulemustega ringi koordinaadid ja raadiused pikslites.

Parameetrite määramine toimub testperioodi käigus, kus määratakse minimaalne ja maksimaalne LED lambi kaugus kaamerast ning sätestatakse lisaparameetreid, et saavutada soovitud tuvastamise kiirus ja täpsus.

2.3 Malli tuvastamine piltidelt

Malli tuvastamise meetod on laialdaselt kasutusel olev digitaalne masinnägemise algoritm, mille abil otsitakse pildi pealt eelnevalt defineeritud mustrit või kujundit.

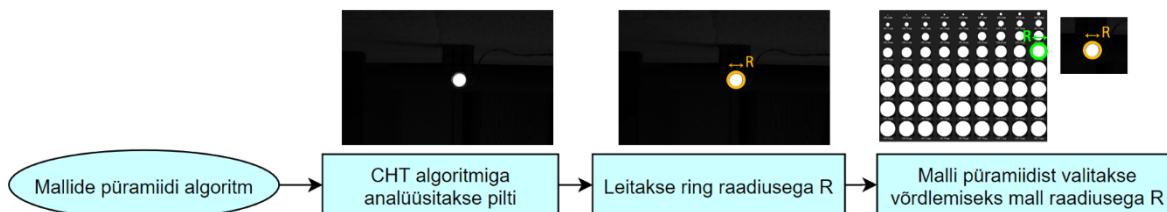
Algoritmi sisestatud pilti võrreldakse määratletud malliga ning pildi kohta väljastatakse maatriks, kus kõige kõrgem väärtus vastab võimalikule malli asukohale pildil. Malli tuvastamine eeldab, et malli dimensioonid, rotatsioon ning skaala on identne ka otsitava pildil. Mida vähem dimensioonid või skaala erinevad mallist, seda ebatõenäolisem on korrektse asukoha leidmine pildilt. Malli tuvastamise tulemused on kujutatud joonisel 2.2, kus jooniselt c) otsitakse eelnevalt defineeritud malli, mis on kujutatud joonisel a).



Joonis 2.2 a) Eeldefineeritud mall b) Malli tuvastuse tulemus kujutatud binaarselt c) Malli tuvastamise tulemus kujutatud algoritmi sisestatud pildil [21]

Sellegipoolest saab malli tuvastamise abil erinevaid malli skaalasid leida pildilt kasutades OpenCV dokumentatsioonis kirjeldatud piltide püramiidi [23], mille rakendamisel sisestatakse järjest kahaneva skaalaga pilte malli tuvastuse algoritmi ning kõikidele tehakse tuvastus. Kui originaalpilt on väiksem, kui mall, siis algoritm lõpetab töö. Algoritmi lõpetades väljastatakse iga analüüsitud pildi kohta tulemusmaatriks, mille alusel saab leida pildi, kuhu mall sobis kõige paremini. Selline tulemusmaatriks on kujutatud joonisel 2.2b. Sellise meetodiga ei ole skaala erinevus enam oluline, kuna algoritmi sisestatud pildi skaalat muudetakse vastavalt vajadusele. Sellegipoolest kui algoritmi sisestatud pildi skaala erineb kordades malli suuruselt, on algoritmi töö mahukas ja aeglane, kuna peab läbi töötama suurel hulgal erinevaid pildi suuruseid, millelt ei pruugi mallile vastet leida. Selline lahenduskaik ei sobi reaajas pilditöötamiseks, kuna piltide analüüsiks kuluv aeg ei ole regulaarne ning viidete ajal võib reaajas situatsioon olla juba muutunud. Piltide püramiidi asemel koostati antud töös nõ. mallide püramiidi, ehk iga võimaliku tuvastuselemendi suuruse kohta koostati mall kindla teatud raadiusega ning vastavalt CHT algoritmist tuvastatud ringi raadiusele valiti võrdlemiseks sama raadiusega mall. Niiviisi on tuvastus kiirem, kuna mallituvastust

tuleb teha vaid korra, ühe malliga. Koostatud malli püramiidi algoritmi etapid on kujutatud joonisel 2.3.



Joonis 2.3 Malli püramiidi algoritm

2.4 Valgustus ja kaamera säriaeg

Pistikupesa suurte dimensioonide tõttu ning töökindluse eesmärgil sooritatakse masinnägemine lisakujundi abil, mis on paigaldatud pesa lähedale. Kujundi abil saab kaamera arvutada relatiivset laadimiskäpa positsiooni pesast ning andmed edastada kontrollerile, mis sooritab ühenduse. Lisakujund saab olla kas passiivne või aktiivne ning nende võrdlus on esitatud tabelis 2.2.

Tabel 2.2 Passiivse ja aktiivse kujundi eelised ja negatiivid

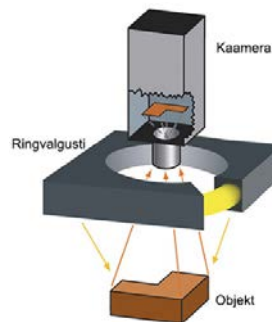
Kujundi tüüp	Näide	Plussid	Miinused
Passiivne	QR-kood, kleeps, joonistatud märgis	Lihtne paigaldada, tuvastamine malli sobitamise meetodiga	Ilmastikuolud segavad tuvastamist, kleeps ei pruugi vastu pidada, öösel ei ole nähtav ilma lisavalgustusega
Aktiivne	LED element	Kergesti tuvastatav, ilmastikuolud ei häiri tuvastamist, tuvastamine toimub kasutades lihtsaid kuid töökindlaid algoritme	Vajab toidet, elektroonika peab olema vastupidav väli tingimustele

Aktiivse elemendi eeliseks on suurem tausta eemaldamine, kuna kaamera säriaja saab viia väga kiireks, tehes tausta tumedaks ning aktiivne LED tuvastuselement jääb ereduse tõttu nähtavale. Sellisel viisil toimib süsteem nii öösel kui ka päeval ajal. Passiivse elemendi jaoks on vaja eraldi valgustust, eelistatult ringvalgustust ühtlase valgusjaotuse tõttu. Ringvalgustus sobib ideaalselt passiivse elemendi tuvastamiseks, kuna valgustab väga ühtlaselt üle kogu kaamera vaatevälja (joonis 2.4). Sellegipoolest ei ole ringvalgustus antud süsteemi asjakohane, kuna päeval ajal ei ole valgustusest abi ning ere päike võib tekitada varje, mis takistavad tuvastamist. Lisaks on keeruline tuvastada passiivset elementi lumiste või vihmaste ilmadega.

a)



b)



Joonis 2.4 a) Uuritud ringvalgustus *MibroBite* [13] ning b) ringvalgustuse töödiagramm [53]

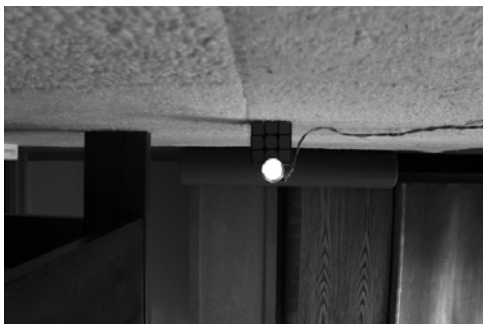
Kuna pistikupesale paigaldatud lisakujundi stabiilne leidmine, sõltumata valguse ja ilmastikuoludest, on peamise tähtsusega siis valiti lisakujundiks aktiivne LED tuvastuselement. LED elemendi tõttu ei pea süsteemi tarkvara arvestama väli tingimustega, kuna väga kiire kaamera säriaja puhul ei ole varje, lund või vihma näha. Valitud tuvastuselement (joonisel 2.5) on veekindel tasemega IP68 ning valgusvoog on 80 lm, mis on piisavalt ere, et tuvastuselementi on näha ka päikese käes. Elemendi diameeter on 16 mm, mis on piisavalt suur ka ühe meetri kauguselt tuvastamiseks.



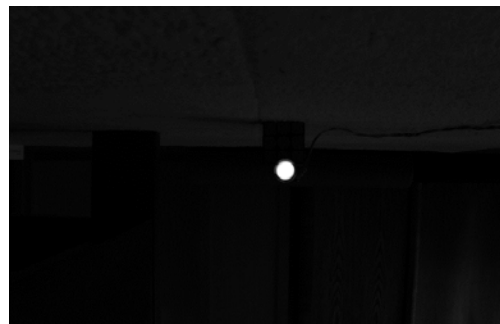
Joonis 2.5 Valitud aktiivne tuvastuselement, LED tuvastuselement [42]

Aktiivse elemendi peamiseks eeliseks on kaamera säriaja ja võimenduse abil tausta eemaldamine. Sellisel juhul saab algoritmi sisestada juba suuresti filtreeritud pildi, millelt tarkvara peab üles leidma korrektsete omadustega LED elemendi, eristades muid eredaid objekte nagu polti pea või päikese peegeldus, mis võivad pildile sattuda. Joonisel 2.6 on toodud näide säriaja abil tausta eemaldamisest, kus pildil 2.6a on säriaeg kümme korda aeglasem, kui joonisel 2.6b. Kiirema säriaja tõttu jõuab vähem valgust kaamera sensorile ning pilt on tumedam, kuid kuna LED element on väga ere, jääb see nähtavale.

a)



b)



Joonis 2.6 Kaamera säriaja abil tausta eemaldamine, a) säriaeg on 200 ms ning b) 20 ms

3. RIISTVARA VALIK

Riistvara seadmete valikul lähtuti süsteemi nõuetest ning võimalusest, et seadet võib olla vaja toota rohkem kui üks. Seetõttu on enamus valitud seadmed olemasolevad tooted, mille omadused ja parameetrid olid kõige sobilikumad lähteülesande sooritamiseks. Süsteemi nõuded on koostatud võimalikult konkureerivaks olemasolevate süsteemidega ning on esitatud tabelis 3.1

Tabel 3.1 Süsteemi nõuded

Nõue	Nõue väärtus	Selgitus
Tuvastuselemendi tuvastamise kaugus	(0,1...1) m	Kaamera peab pistikupesa elemendi tuvastama alates 1 m kauguselt ning kaamera kõige minimaalsem kaugus on 0,1 m elemendist
Tuvastuselemendi tuvastamise kiirus	< 1 s	Tuvastamine peab toimuma koheselt, kui selleks on käsk saadud kontrollerilt
Tuvastuselemendi koordinaatide edastamise sagedus	> 5 Hz	Süsteem peab koordinaate edastama minimaalselt 5 korda sekundis kontrollerile
Veekindlus	IP 65	Süsteem peab väli tingimustes vastu pidama
Töötamisrežiim	24/7	Süsteem peab olema töökindel nii öösel kui ka päeval, ilmastikuoludest sõltumata

3.1 Masinnägemise arvuti valik

Turul on saada mitu erineva võimekusastmega masinnägemise arvutit, nagu näiteks Raspberry PI, Jetson Nano ning Jetson TX2. Nvidia Jetson seeria arvutid on ühed populaarseimad süsteemid masinnägemise lahendusteks erinevate jõudlusastmete ning suure seeria valiku tõttu. Jetson seeriaga konkureerib Raspberry PI seeria, mille hiljutisem mudel 4B kasutab veidike kiiremat põhi-protsessorit, kui samas hinnavaheemikus olev Jetson Nano [6]. Kuid Jetson seeria suurimaks eeliseks on videokaart koos CUDA tehnoloogiaga, mida masinnägemise tarkvara OpenCV suudab rakendada [7]. Lisaks on kõikidele Jetson süsteemidele eelinstallitud JetPack tarkvaraarenduskomplekt mis sisaldab vajalikke moduleid ja dokumentatsiooni masinnägemise arendusprotsessiks, kaasa arvatud Python programmeerimiskeel koos Jetson seeriale optimeeritud OpenCV teegiga.

Jetson Nano valiti antud süsteemi masinnagemise arvutiks parema graafikakaardi ning CUDA tuumade võimaluste pärast. Lisaks ei ole Raspberry PI seeria arvutitel sisse ehitatud võimalust interneti paketi suuruse määramiseks [52], mis on antud töös valitud kaamera jaoks oluline. Peale Jetson Nano on Jetson seerias veel võimsamaid arvuteid [8], kuid kuna antud süsteemi masinnagemise vajadus ei ole nii nõudlik, siis piisab algmudelilist Nano. Jetson Nano paigaldati ka korpusesse, mis võimaldab arvuti monteerida DIN-liistule [12].



Joonis 3.1 Jetson Nano arendaja komplekt JETBOX korpuses [11][12]

3.2 Kaamera valik

Kaamera valik lähtub süsteemi parameetritest, tähtsad omadused OpenCV teekide rakendamiseks on resolutsioon, kaadrisagedus ning kaamera sensori tüüp. Antud süsteemis peab kaamerale olema ka hea signaali mürakindlus, veekindlus ning tarkvaraline töökindlus.

Resolutsioon on peamise tähtsusega süsteemi tuvastamise täpsusel, kuna tihedama resolutsiooniga saab täpsemalt tuvastada elemendi asukohta ja kuju. Sellegipoolest mida suurem on resolutsioon, seda aeglasemaks läheb ka algoritmide kiirus. Jetson Nano toetab pildisuuruseid kuni 24 miljoni pikslini, kuid kiiruse ja töökindluse tõttu peaks see olema madalam [41].

Kaadrisagedus määrab, kui tihti saab süsteem saata pistikupesa koordinaate kontrolleriile. Minimaalne kiirus selleks on määratud 10 Hz, seega kaamera kaadrisagedus peab olema suurem kui 10 kaadrit sekundis.

Sensori tüüp määrab kas kaameralt saadud pilt on värviline või must-valge. Värvisensorite andmetöötlus on aeglasem ja protsessorimahukam, kuna sensorilt tuleb 3 korda rohkem infot iga kaamera piksli kohta. Kuna antud süsteemis ei ole värvid olulised, siis andmetöötluse kiirendamiseks ja lihtsustamiseks valitakse mono tüüpi sensor.

Mürakindlus peab kaamerasüsteemil olema hea, kuna kaamera paigaldatakse väga lähedale kõrgepinge seadmetele, mille puhul ei tohi elektrihäiringud andmetesse sattuda. Sobiva mürakindluse saab kasutades varjestatud andmeside kaableid.

Veekindlus võimaldab kaamerasüsteemi paigaldada väli tingimustesse ning ka mere- või sadama äärde.

3.2.1 Kaamera

Turul on saada lai valik erinevaid kaameraid, nagu LUCID Vision Triton seeria, Allied Vision Mako seeria ning FLIR Grasshopper seeria kaamerad. Antud töö rakenduse jaoks sobib kõige paremini Triton seeria, kuna kaamera on IP67 veekindel ning võrreldes teiste samas hinnaklassis olevate kaameratega, on saadaval väga sisurohke dokumentatsiooniga tarkvaraarenduskomplekt, millel on eraldi seadistusmoodulid Jetson Nano-I oleva Ubuntu 18.04 operatsioonisüsteemile, koos Pythoni näidetega. Triton seeria kaameratest valiti TRIO23S, mis on 2,3 MP mono sensoriga kaamera, maksimaalse kaadrisagedusega 52 fps (joonisel 3.2). Koos LUCID firma poolt pakutavate lisaseadmetega tehti kaamera ka IP67 tolmu- ja veekindlaks [22]. Sellegipoolest, et kaamera on veekindel, on tootja poolt töötemperatuuri piirideks määratud -20 °C kuni 55 °C, mille hoidmiseks on kaamerale vajalik ka soojustus, et külmal ajal talveilmadel temperatuur alla sobiva ei langeks.

a)

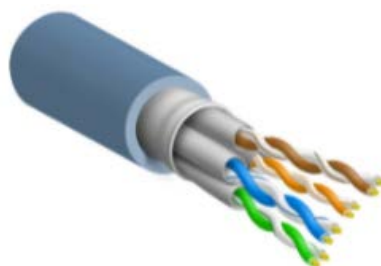


b)



Joonis 3.2 Triton 2.3 MP Mudel a) tagant vaade b) eest vaade ilma objektiivita [22]

Triton seeria kaameratel on võimalik kasutada nii toiteks, kui ka andmesideks sama kaablit, kasutades toide üle Etherneti (PoE) toiteseadet. Toitekaabel ja andmeside kaabel on Cat 6a varjestatud tüüpi (joonisel 3.3), mis lisaks sobib paigaldamiseks energiakettil suuremate automaatika süsteemide jaoks [43].



Cat 6a SFTP

Joonis 3.3 Cat 6a varjestatud kaabel [44]

LUCID kodulehel on ka kaamera mudelifailid saadaval, mille abil koostati kaamerale korpus. Kaamera edastab andmeid vastavalt GigE standardile ning kasutab 9000 baidiseid Etherneti pakette, mille edastamiseks on vajalik ka sama paketi mahtu toetav jaotur ning masinnägemise arvuti.

3.2.2 GigE standard

GigE Vision on liidese standard, mis võeti kasutusele 2006 aastal juhtivate kaamera – ja tarkvaratootjate poolt. Liides pakub raamistiku suuremahulise kaamera andmestiku edastamiseks üle võrguliidese. GigE eesmärk on lihtsustada kaamera ja kolmanda osapoole tarkvara ühildumist, mistõttu GigE kaameraid on kiire ja kerge arendussüsteemi sisse rakendada, kuna tootja poolt on suurem osa tarkvaraarendusest juba tehtud [44]. LUCID Vision Triton seeria, ning enamus teised LUCID Vision firma kaamerad, toetavad GigE standardit ning võimaldavad seetõttu väga mugavalt muuta kaamera sätteid, kaamera IP aadressi ning hallata andmetüüpe. GigE standard määrab ka mitme seadme ja tarkvara omavahelise sobivuse, näiteks kolmanda osapoole tarkvara, mis on üles ehitatud vastavalt GigE standardile, töötab kõikide GigE standardit toetavate kaameratega [44]. Lisaks on GigE standard väga lühikese viivitusajaga, kuna kasutab UDP protokollit TCP asemel. See võimaldab edastada suuri andmemahte reaajas pilditöötamiseks [44].

3.3 Objektiivi valik

Süsteemi tuvastamiskaugus on määratletud 1 m ... 0,1 m. 1 m kauguselt peab kaameral olema võimalikult lai vaateväli, et tuvastada LED elementi suuremalt alalt, samas elemendi suurus pikslites peab jääma võimalikult suureks, et tuvastamine oleks stabiilne. Lisaks ei tohi LED elemendi suurus pikslites ületada kaamera resolutsiooni väga lähedalt, kuna siis ei ole tuvastamine võimalik. Sobiva objektiivi valimiseks arvutati teoreetilised kaamera vaateväljad ning LED tuvastuselemendi suurused pildil erinevate objektiivi fookuskaugustega välja.

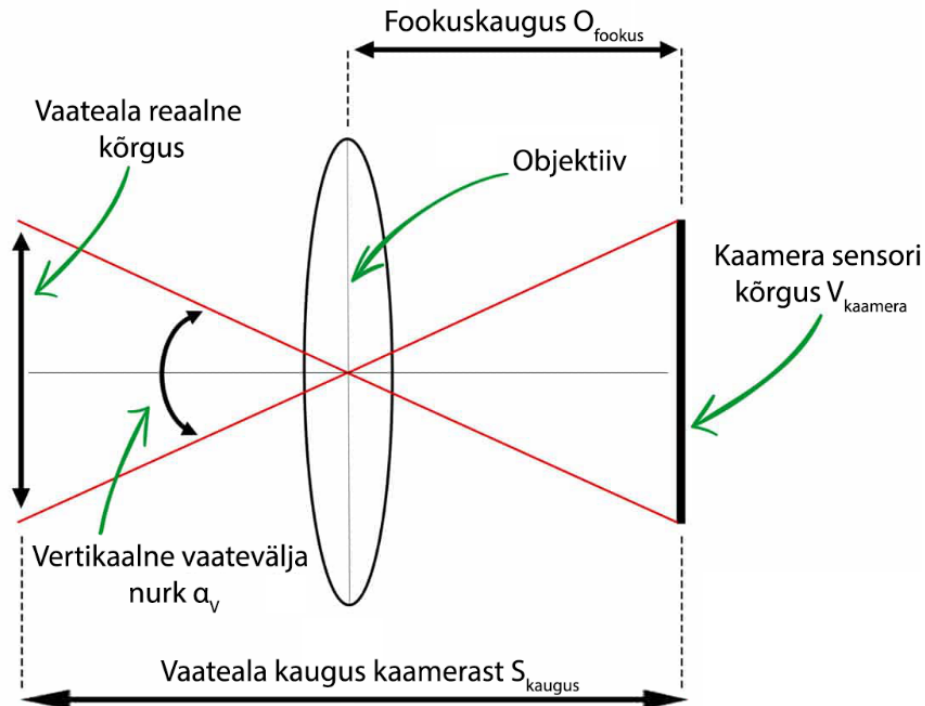
$$\text{Kaamera sensori laius } H_{\text{kaamera}} = 6,62 \text{ mm}$$

$$\text{Kaamera sensori kõrgus } V_{\text{kaamera}} = 4,14 \text{ mm}$$

$$\text{LED diameeter } D_{\text{led}} = 16 \text{ mm}$$

$$\text{Vaateala kaugus kaamerast } S_{\text{kaugus}} = 1000 \text{ mm}$$

Vertikaalse kaamera vaatevälja nurga arvutamiseks kasutati seost kaamera sensori suuruse ja objektiivi fookuskauguse vahel, mis on visuaalselt kujutatud joonisel 3.4 ning matemaatiliselt esitatud valemis 3.1 ja 3.2:



Joonis 3.4 Kaamera ja objektiivi seos vaatevälja nurga suhtes [45]

$$\alpha_V = 2 * \arctan \frac{V_{kaamera}}{2 * O_{fookus}}, \quad (3.1)[45]$$

kus α_V - kaamera vertikaalne vaatevälja nurk
 O_{fookus} - objektiivi fookuskaugus (tabel 3.2)

Valemit 3.1 kasutati ka kaamera horisontaalse vaatevälja nurga leidmiseks, kasutades arvutamiseks kaamera sensori laiust:

$$\alpha_H = 2 * \arctan \frac{H_{kaamera}}{2 * O_{fookus}}, \quad (3.2)[45]$$

Nii horisontaalse kui ka vertikaalse vaatevälja nurga saab seejärel teisendada kaamera nähtava pildi reaalseks pindalaks kindlal kaugusel. Pindala arvutused koostati, arvestades maksimaalset kaugust LED tuvastuselemendist, 1 m kauguselt sensorist.

$$Vaateala reaalne kõrgus = 2 * \tan \left(\frac{\alpha_V}{2} \right) * s_{kaugus}, \quad (3.3)$$

kus s_{kaugus} on kaamera sensori kaugus LED tuvastuselemendist.

Arvutati nii horisontaalne kui ka vertikaalne reaalne pikkus ning pikkuste korrutamisel saab kaamera nähtava reaalse pindala kaugusel s_{kaugus} , koos vastava pindala ühikuga.

$$Vaatevälja pindala(s_{kaugus}) = Vaateala reaalne kõrgus * Vaateala reaalne laius \quad (3.4)$$

Tabelis 3.2 on esitatud neli objektiivi erinevate fookuskaugustega ning kõikide objektiivide vaatevälja pindala 1 m kaugusel kaamerast on välja toodud. Lisaks on iga objektiivi kohta esitatud teoreetiline LED tuvastuselemendi minimaalne suurus ühe meetri kaugusel kaamerast, mille abil hinnatakse kas tuvastamine on stabiilne või mitte.

Tabel 3.2 Erinevate objektiivide vaatevälja ja vaatepindala võrdlus

Objektiivi Fookuskaugus f_{fookus} (mm)	LED Kaugus kaamerast (m)	Vaatevälja pindala (m ²)	LED lambi diameeter pikslites
4	1	1,71	18
6	1	0,761	28
8	1	0,428	37
12	1	0,190	55

Fookuskaugusega 4 mm objektiivi vaatevälja pindala on kõige märkimisväärses, kuid LED lambi suurus 1 m kauguselt on liiga väike, raadius on alla 10 piksli ning võib tekitada ebastabiilsust tuvastamisel, kuna tuvastuselement ei pruugi välja näha ringikujuline niivõrd madala resolutsiooniga. 8- ja 12 mm fookuskaugusega objektiividel on liiga suunatud vaateväli, mis ei ole otsimis- ja tuvastusfunktsiooniks sobiv. 6 mm fookuskaugusega objektiivi vaateväli on mõistliku suurusega ning LED lambi tuvastamine 1 m kauguselt on stabiilne ligi 30 piksli diameetri abil. 6 mm fookuskaugusega objektiiv jätab ka tagavara ruumi, juhuks, kui tuvastatav element satub kaugemale kui 1 m.

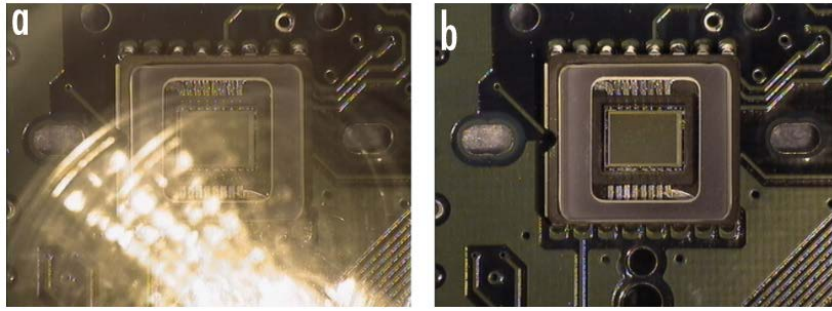
3.3.1 Objektiiv

Kaamerale sobivatest C-kinnitusega objektiividest valiti 6 mm fookuskaugusega objektiiv, mille vaateväli on antud ülesande jaoks piisav, kus kaamera näeb 1 m kauguselt 0.761 m² suurust ala. Suur nägemispindala tuleb kasuks LED lambi otsimisel. Objektiiv on paigaldatud kaamera külge ning tihendatud korpusega, mis kaitseb objektiivi IP67 piirangus vee ja tolmu eest.



Joonis 3.5 a) Edmund Optics UC Series 6 mm objektiiv, b) objektiivi veekindel korpus [26]

Objektiivile paigaldati ka polariseeriv filter, mis eemaldab potentsiaalselt häirivad polariseeritud peegeldused kaamera pildilt. Polariseeritud peegeldused tekivad, kui päikese või muu valgusallika kiired peegelduvad 90 kraadise nurga all objektidelt kaamera sensorisse. Joonisel 3.6 on toodud näide polariseeriva filtri rakendusest.



Joonis 3.6 joonisel (a) on tugev polariseeritud peegeldus, mis joonisel (b) on polariseeriva filtriga eemaldatud [30]

Polariseeriv filter neeldub kõik 90 kraadise nurga all olevad peegeldused ning peegeldustest vaba pilt jõuab kaamerasse [30].

3.4 Andmeside komponendid

Kaamera võimaldab andmevahetuseks ja toiteks kasutada eraldi interneti ja toitekaablit või PoE toiteseadet, mis edastab üle hariliku Ethernet kaabli ka vajaliku kaamera toite ning niiviisi pildiandmed, kui ka kaamera toide on sooritatud ühe kaabliga. PoE süsteem valiti kaamera toiteks, sest kahe kaabli asemel on vaid üks, mis on viisakam ja lihtsam üles seada.

3.4.1 PoE toiteseadede

Kaamerale on vajalik IEEE 802.3af standardiga PoE toiteseadet [22]. Standard määrab minimaalse toitepinge, 44 VDC, ning maksimaalse võimsuse, 15.4 W [27]. Lisaks peab PoE seade toetama 1000BASE-T kiirust, mis on kaamera andmete edastamiskiirus. Oluline oli ka DIN-liistule kinnitamise võimalus, kuhu kõik komponendid hiljem monteeritakse. Planet IPOE-162 (joonisel 3.7) oli koheselt kättesaadav ning kõigile parameetritele vastav toode [29]. Kirjutamise ajal on leitud ka soodsamaid alternatiive [28]. Valitud seadmel on kahekordne toitesisend, mida saab kasutada puhvertoiteallikaga (UPS), mis tagab töökindluse ka elektrikatkestuse puhul.



Joonis 3.7 Planet IPOE-162 valitud PoE toiteseadet [29]

Valitud PoE toiteseadmel on üks PoE pesa ning üks andmeside pesa, kaamera ühendatakse PoE pesasse ning andmetele saab ligi tavalisest Etherneti pesast, ühendades selle kas otse arvutisse, või läbi võrgujaoturi arvutisse.

3.4.2 Võrgujaotur

Antud süsteemi puhul tuleb kasutada võrgujaoturit, kuna mitu seadet soovivad omavahel andmeid jagada: PLC, kaamera, Jetson Nano ning lisaks ka 4G marsruuter. Mitme seadme suhtlust saab sooritada kasutades jaoturit, mis haldab seadmete omavahelist kommunikatsiooni. Kaamera andmemahu edastamiseks on vaja jaoturit, mis on kiirusega 1000BASE-T ning lisaks peab jaotur toetama 9000 baidist paketsuuruse edastamist. Palju 1000BASE-T kiirusega standardsed jaotureid toetavad vaid 1500 baidist paketi edastamist, mis on vaikumisi paketi suurus enamikus võrkudes [32]. Leitud WEIDMÜLLER IE-SW-BL05-4GT-1GS jaotur (joonisel 3.8) sobib ülesande täitmiseks; kuna jaotur vastab kiiruse ja paketi suuruse nõudlusele ning lisaks on korpusel DIN-liistu kinnitus ja samuti kahekordne toitesisend puhvertoiteallika jaoks.



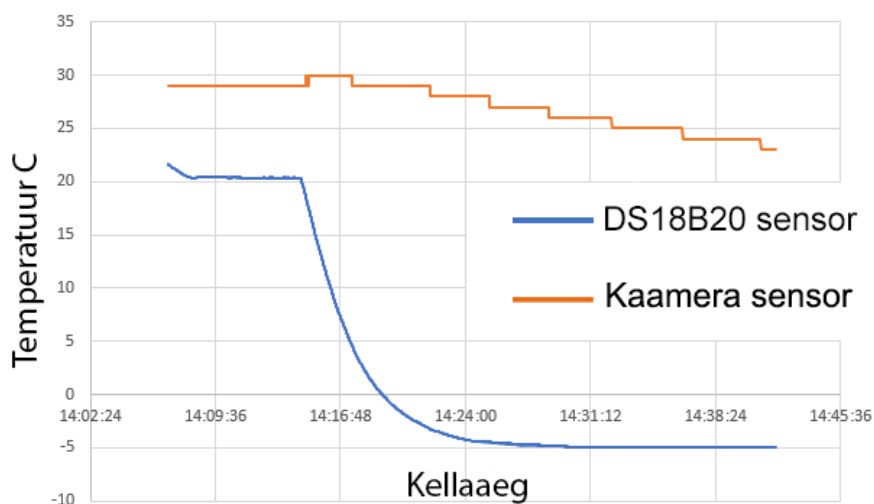
Joonis 3.8 WEIDMÜLLER valitud Etherneti jaotur [46]

Antud jaoturi seerias mitu erinevat tootevarianti, valiti 5-pesaline jaotur kiirusega 1000BASE-T, kuna rohkem pesasid teeb hinna kallimaks ning antud süsteemis on vaja vaid neli seadet omavahel ühendada.

3.5 Kaamera küttesüsteem

Külmematel talveilmadel võib väli temperatuur langeda alla $-20\text{ }^{\circ}\text{C}$, mis langeb alla valitud masinägemise kaamera nõutud töötemperatuuri. Lisaks võib suur temperatuuri kõikumine tekitada kondenseerumist kaamera objektiivil, mille vältimiseks on hea hoida kaamera temperatuur võimalikult ühtlasena. Temperatuuri hoidmiseks kaamera töötemperatuuri piirides on vaja teada väli temperatuuri ning selle alusel lülitada soojuselementi. Valitud masinägemise kaamerale on sisseehitatud temperatuuri sensor, kuid see mõõdab vaid sensori temperatuuri, mis ei vasta väli temperatuurile. Koostati ka 40 minuti vältel temperatuurisensori test, kus kasutati olemasolevat

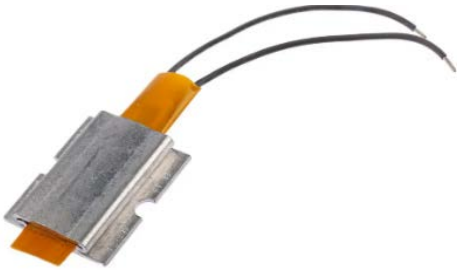
digitaalset termostaati DS18B20 [47] ning võrreldi tulemust kaamera sisese temperatuurisensoriga. Tulemused on toodud joonisel 3.9 ning selgus, et kaamera sensori temperatuuri väli temperatuur oluliselt ei mõjuta. Väli temperatuuri mõõtmiseks valiti olemasolev hinnakohane DS18B20, mis on veekindel digitaalne temperatuurisensor [47]. Sensorilt andmete lugemiseks saab kasutada Jetson Nano mitme otstarbelist sisendi/väljundi pinne, kuid need ei ole eriti töökindlad, kuna kaableid ei saa turvaliselt kinnitada, nagu kruviklemmiga saaks ning pinnid Jetson Nano plaadil ei ole mõeldud jootmiseks, mille eiramisel tühistatakse garantii [48].



Joonis 3.9 Kaamera temperatuurisensori ja DS18B20 temperatuuri sensori vahe

Alternatiivina otsustati kasutusele võtta Arduino Nano arendusplaat, mis on väga väike kuid töökindel minikontroller. Teek DS18B20 digitaalselt termostaadilt andmete lugemiseks oli kergesti kättesaadav ning rakendatav, mille tulemusena Arduino ja Jetson ühendatakse üle Serial kommunikatsiooni ning Arduino saadab pidevalt andmeid Jetson arvutisse, kus temperatuuri analüüsitakse ja temperatuuri langemisel alla $-5\text{ }^{\circ}\text{C}$ saadab Jetson signaali Arduino kontrollerrisse, et küte sisse lülitada. Kütteelemendiks valiti kompaktne ja isereguleeriv HP06 toode DBK tootjalt (joonisel 3.10a). Kütteelemendil on vastavalt temperatuurile tagasisidestatud voolu reguleeriv süsteem juba sisse ehitatud ning see hoiab konstantselt valitud $40\text{ }^{\circ}\text{C}$ temperatuuri, mis sobib hästi antud süsteemile. Väikse pindala tõttu mahub element mugavalt kaamera korpusesse ning elemendi alumiiniumist korpus kannab temperatuuri üle kaamerale, kui väli temperatuur langeb alla lubatud piiri. Kuna Arduino mikrokontroller ei võimalda väljastada 20 W võimsust, siis kütteelement saab vajaliku 24 V toite läbi releemooduli (joonisel 3.10b), millesse Arduino saadab signaali, kui temperatuur on piisavalt madal. Signaali saamisel lülitab releemoodul 24 V pinge toiteploki kütteelemendile.

a)



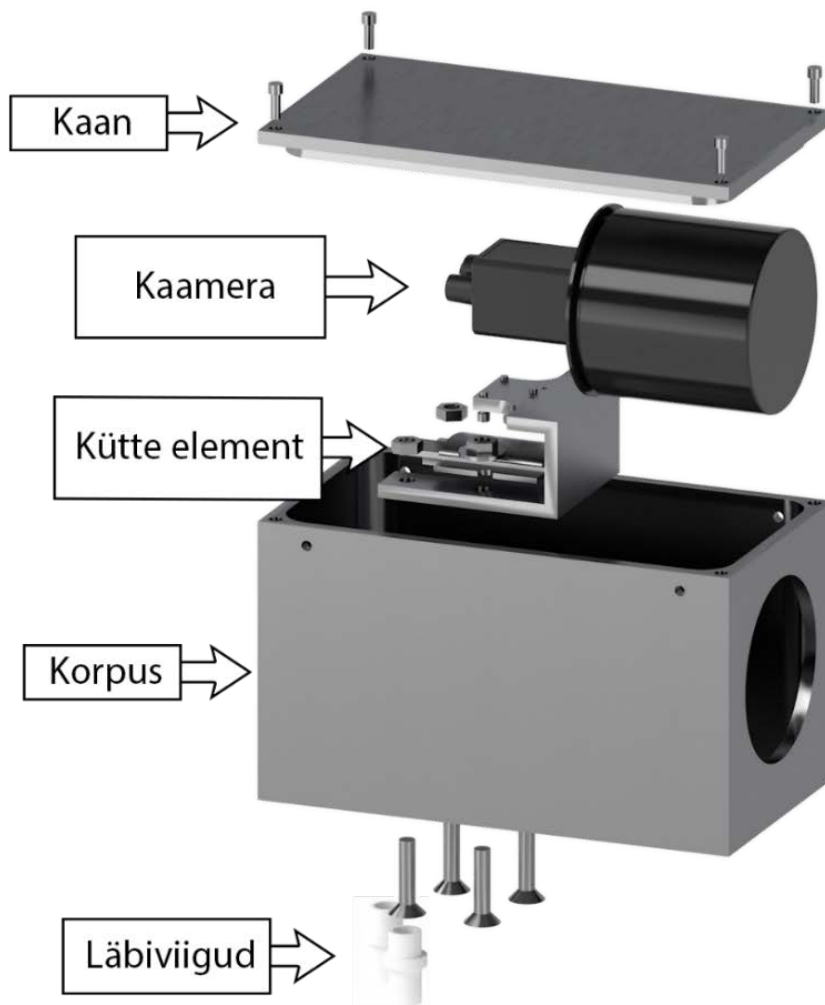
b)



Joonis 3.10 a) Isereguleeriv kütteelement HP06 [49] ning b) releemoodul kütte lülitamiseks [50]

3.6 Korpuse disain

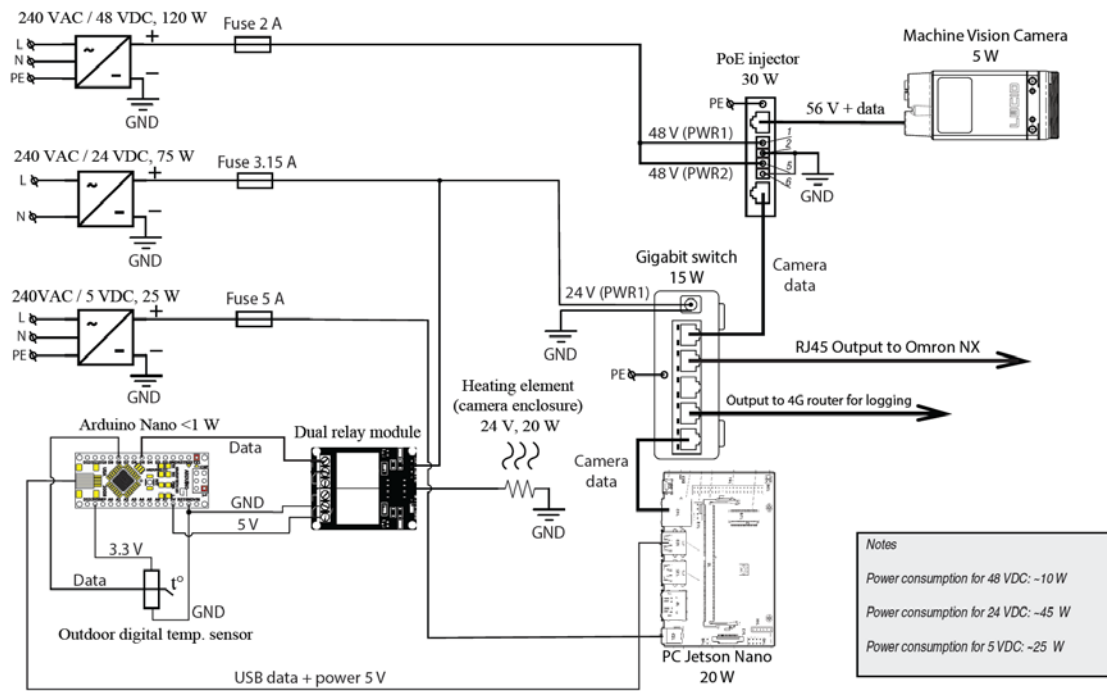
Kaamerasüsteemile modelleeriti eraldi korpus, mis kinnitatakse laadimiskäpa külge, suunaga pistikupesa poole. Korpus koosneb kolmest peamisest osast: kaan, kaamera kinnituselement ning korpus. Kaamera kinnituselement teeb kaamera monteerimise korpuse külge väga kiireks ning tõstab kaamera kõrgemale, et objektiivi tsepter oleks ka korpuse tsentris. Kaamera on kinnituselemendi külge kinnitatud kolme polti abil. Lisaks kinnitatakse ka soojenduselement kinnituselemendi külge, mis kannab soojuse edasi kaamerale. Objektiivi veekindla tuubi jaoks on lõigatud ava korpuse ette, millest kaamera välja näeb. Korpusele on alumistele ja külgmistele pindadele puuritud avad, et niiskuse või vee sattumisel korpusesse saaks see avadest väljuda. Kõik kolm korpuse osa on freesitud alumiiniumist ning kinnituseks kasutatud poldid, seibid ja mutrid on valitud A2 või A4 tasemega, ehk valmitatud korrosioonikindlast terasest. Kaamera ja soojenduselemendi kaablite jaoks on korpuse põhjas kaks ava läbiviikude jaoks, millega kaablid on kinnitatud ja tihendatud korpuse külge.



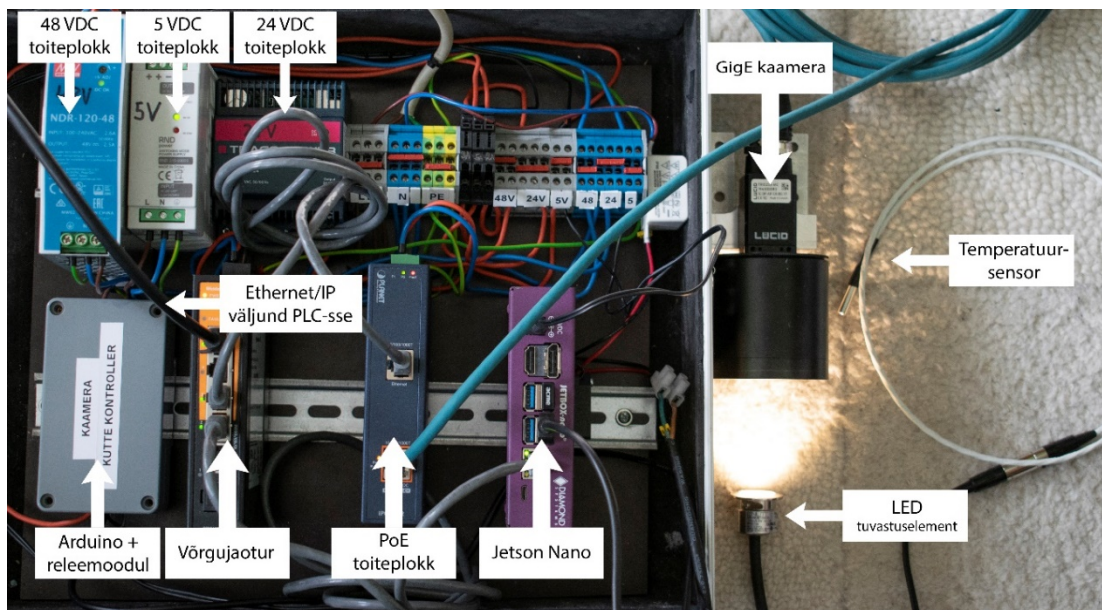
Joonis 3.11 Korpuse elementide vaade

3.7 Elektroonika skeem

Valitud elektroonika komponentidest koostati skeem ning komponentide voolutarbimise abil valiti vastava võimsusega toiteplokkid. Kõikidele toiteplokkidele väljundile paigaldati sulavkaitsmed, et avariikorras komponendid katki ei läheks. Toiteplokkid saavad toite 230 VAC pesast ning nii vahelduvvoolu kui ka alalisvoolu ühendused sooritati riviklemmide abil. LED tuvastuselement on skeemist eraldatud ning saab toite süsteemivälisest spetsiaalsest vooluallikast.



Joonis 3.12 Elektroonika põhimõtteskeem valitud komponentidega



Joonis 3.13 Koostatud süsteem

4.TARKVARA

Süsteemi masinnagemise tarkvara kirjutati kasutades Python programmeerimis keelt, kasutades OpenCV 4.4.1 ning Nvidia JetPack 4.5.1, mis oli eelinstallitud Jetson Nano arvutile. JetPack 4.5.1 sisaldab Nvidia Jetson seeriale optimeeritud versiooni Ubuntu operatsioonisüsteemist Linux Tegra süsteemile (L4T) ning enamusi vajalikke teeke tarkvara kirjutamiseks. Tarkvara arendus teostati autori isiklikul arvutil, kasutades PyCharm tarkvaraarendus keskkonda (IDE) ning koodi käivitamiseks, testimiseks ja parandamiseks kasutati Jetson Nano koos eraldi monitori ja klaviatuuriga, kuna antud süsteemis oli oluline näha kaameralt saadud tagasisidet monitori ekraanil.

4.1 Ülevaade tarkvarast, konfigureerimisvõimalused

Masinnagemise tarkvara käivitatakse käsurealt ning testimise lihtsustamiseks on käsureale lisatud vajalikud konfiguratsioonivalikud, millega saab muuta programmi tööülesandeid. Käsurea valikud võimaldavad välja lülitada automaatika kontrolleri suhtlus, salvestatud video üleslaadimise ja video salvestamise. Lisaks saab käsurea funktsiooniga muuta kiirelt süsteemi sihtpunkti, mille suhtes edastatakse koordinaate.

positional arguments:

```
run                Start program
```

optional arguments:

```
-h, --help          show this help message and exit
--no-plc            Disables PLC connectivity [default DISABLED]
--no-upload         Disables uploading of video [default DISABLED]
--no-record         Disables recording AND upload [default DISABLED]
--set-setpoint      Enters setpoint changing system [default DISABLED]
--show-video        Enables showing output video to monitor [default DISABLED]
```

Näide käivitamisest: >> python3 PLC_CONTROL.py run --no-upload --show-video

Kood.4.1 Programmi võimalikud konfiguratsioonivalikud

Lisaks käsurea konfiguratsioonile on kõik ülejäänud muudetavad parameetrid välja toodud eraldi spetsiaalses Python moodulis, mis on esitatud lisas 2. Selles moodulis saab muuta kõiki süsteemis kasutusel olevaid muutujaid, näiteks kaamera seadistust, CHT parameetrid jpm. ning tehtud muudatused hakkavad toimima peale programmi restarti. Süsteemi automaatseks käivitamiseks koostati Ubuntu systemd skript, mis jooksub programmi vaikimisi konfiguratsiooniga. Testimiseperioodil oli systemd skript eemaldatud ning kood käivitati käsurealt koos vajalike argumentidega.

Tarkvara koosneb viiest ühendatud Python moodulist:

FINS_UDP – Moodul Omron FINS-i protokollil abil suhtlemiseks

FTP – Moodul failide üleslaadimiseks ning lokaalselt kaustade ja failide haldamiseks

MODIFIERS – Süsteemi parameetrite haldusfail, kus on kõik muudetavad ja olulised parameetrid välja toodud

OPENCV – Moodul kaamera piltide analüüsimiseks ning tulemuste väljastamiseks

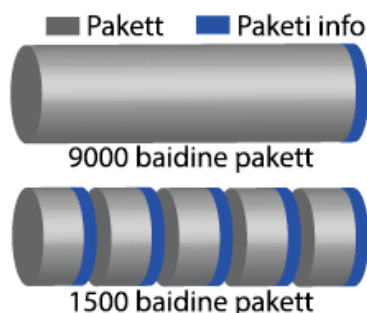
PLC_CONTROL – Moodul PLC käskude täitmiseks

Lisaks on iseseisev moodul, mis genereerib kõik vajalikud LED tuvastuselemendi mallid kaameraga tehtud pildist ning nimetab ja järjestab need vastavalt raadiusele, et tarkvara saaks neid kasutada vastavalt faili nimele.

Moodulid on kirjutatud võimalikult PEP8 stiili kohaselt, et kood oleks kergesti arusaadav ka teistele arendajatele ning oleks üldiselt korrektselt vormistatud [51].

4.2 Kaamera kommunikatsioon

Kaamera seadistamiseks ja suhtluseks kasutati LUCID Arena tarkvaraarenduskomplekti (SDK), mis on kättesaadav LUCID kodulehelt koos näidete ja õpetustega [24]. Järgnevalt süsteemi pingelinkasse ühendamist, initsialiseeritakse kaamera ning eraldi koodilõik seadistab kõik kaamera vajalikud sätted. Lisaks on kaamera IP-aadress seadistatud staatiliseks Arena SDK abil, et sellega saaks alati samalt aadressilt ligi. Kaamera Etherneti puhvri suurus baitides on seadistatud 9000, et kogu kaamera pildi saaks edastada vaid ühe Etherneti paketi. Standardiselt on puhvri suurus määratud 1500 baiti, mille puhul jaotatakse suurem sõnum tükkideks ning saadetakse eraldi väiksemate pakettidena. Pakettide jaotamine ning eraldi saatmine ja hiljem kokku monteerimine on aja- ja andmemahukas, kuna igal paketil peab lisaks pildi andmetele olema info, mis sisaldab paketi järjekorranumbrit ja muud vajalike metaandmeid [44]. Väikeste pakettide puhul on metaandmeid rohkem (joonis 4.1) ning seetõttu kasutatakse nii kaamerale, kui ka Jetson Nano arvutil 9000 baiti paketi puhvri suurus. Kuna Jetson Nano taastab peale restarti kõik algsätted, siis tuleb koodi alguses seada paketi suurus 9000 peale, et Nano saaks pakette korralikult vastu võtta.



Joonis 4.1 9000 ja 1500 baidiste pakettide erinevus [44]

Kaameralt piltide saamiseks kasutatakse tarkvaralist signaali, millele järgnevalt kaamera salvestab pildi puhvermällu, kus selle saab kopeerida. Kaameralt saadud pilt on üherealine maatriks, kus on kogu resolutsiooni väärtuses elemente. Resolutsiooni 1000x1200 puhul oleks maatriksi suurus 1x1200000. Tegeliku pildi suurus ning laius on seadistatud koodi poolt ning nende abil saab üherealise maatriksi muuta standardseks $n*d$ maatriksiks, kus n on pildi laius ning d on pildi pikkus.

4.3 Omron FINS suhtlus

Omron kontrolleri kommunikatsioonimeetod (FINS) on Omron firma poolt loodud kommunikatsiooni protokoll ühtlaseks ja stabiilseks suhtluseks PLC ja erinevate arvutite vahel. FINS protokoll saab rakendada üle Ethernet/IP, HostLink, ControllerLink, ToolBus ja teiste tööstuses kasutatavate protokollidega [25]. Kuna Jetson Nano arvutil on olemas Gigabitine Ethernet, siis kasutati tööstuskontrolleriga suhtlemiseks Ethernet/IP protokoll. Üle FINS-i suhtlemiseks on kasutatud avatud lähtekoodiga FINS_UDP Python moodulit, mida on modifitseeritud autori poolt. Moodulisse on lisatud digitaalne lukk, mis takistab erinevatel moodulitel korraga PLC kontrolleriga suhtlemist, et vältida andmete korruptsiooni. Lisatud on ka turvameetmeid, et kood ei seiskuks ühenduse kaotamise juhul. Kogu suhtlus üle FINS-i toimub baitides ja bittides, FINS_UDP moodul konverteerib vajalikud andmed baitideks ning kirjutab need määratud PLC mäluualasse, mis saadab andmete kättesaamisel kinnituseks ka vastuse. Kontrolleri mäluualade kontrollimiseks kasutati Omroni programmi Multiway V10, mis võimaldas kontrolleri mäluualade väärtuseid lugeda ja muuta. Joonisel 4.2 ja 4.3 on toodud näide Python mooduli ja PLC vahelisest suhtlusest. Joonisel 4.2 saadetakse Python mooduli abil väärtused 20 ja -20 kontrolleri mäluualadesse ning seejärel joonisel 4.3 Multiway programmi abil loetakse vastavaid mäluualasid. Tulemuseks annab Multiway HEX andmetüübis 0014 ning FFEC, mis vastavad väärtustele 20 ja -20.

```
xxx x=20
xxx y=-20
xxx Fins.FinsConnection.write_words(Fins.MEM_D_WORD, 2, (x, y), mode=INT)
b'\x82\x00\x02\x00\x00\x02\x00\x14\xff\xec'
```

Joonis 4.2 Python mooduli abil andmete saatmine kontrollerile

```
[12:17:45.961] <--:800003000B00000300100101820001000003
[12:17:45.965] -->:C00002000300000B00100101000000010014FFEC
```

Joonis 4.3 Kontrollerist andmete lugemine Multiway programmi abil

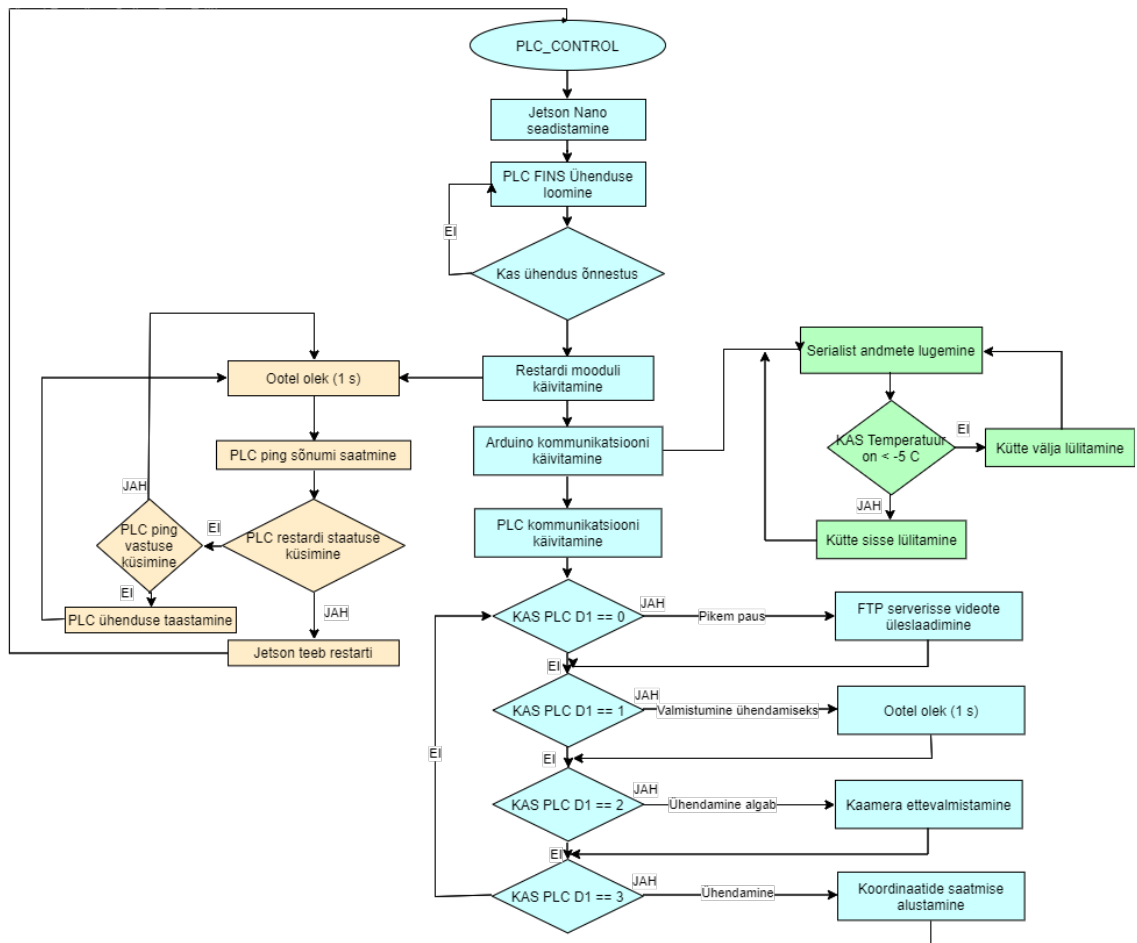
4.4 Põhifunktsionaalsus

Jetson Nano arvutil oleva PLC_CONTROL programmi käivitades seadistatakse esimesena Etherneti ja protsessori maksimaalsed kiirused. Seejärel alustab programm kontrolleri kommunikatsiooni loomist. Kui kontrolleriiga ei saanud ühendust, proovitakse uuesti, niikaua kuni lõpuks on ühendus saavutatud. Kuna IP aadressid on nii kontrolleriil kui ka Jetson Nano arvutil staatilised, siis ainuke põhjus, miks ühendust ei pruugita saavutada, on vigastatud või lahtine netikaabel või toite puudumisel kontrolleriil.

Kui FINS kommunikatsioon on saavutatud, käivitatakse Arduino suhtlust haldav moodul ning kontrolleri restarti käsklusi lugev moodul. Restarti käsklusi lugev moodul töötab iseseisvalt põhiprogrammist, et põhiprogrammi kokkujooksmisel oleks võimalik Jetson Nano arvutile teha restart läbi kontrolleri. Järgnevalt ootab põhiprogramm kontrolleri käskluseid, mida on koostatud neli ning nende saamiseks loeb Jetson kontrolleri D mäluala esimest väärtust:

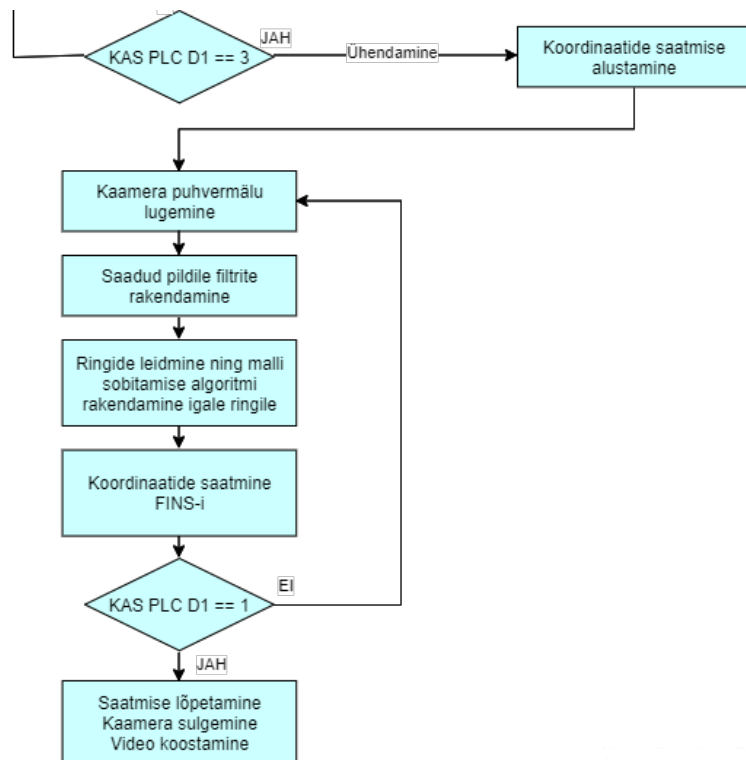
- D1 = 0 – Pikem paus süsteemis, seda aega kasutatakse video ja logi failide üleslaadimiseks serverisse. Kontrolleriil küsitakse staatust iga 5 sekundi tagant.
- D1 = 1 – Valmistumine ühenduse loomiseks. Kontrolleriil küsitakse staatust iga 1 sekundi tagant.
Lisaks kui ühendamine on hetkel aktiivne, siis väärtus 1 teatab ühendamise lõpetamisest, peatab koordinaatide saatmise ning sulgeb kaamera ja salvestab video.
- D1 = 2 – Alustatakse ühendamist. Süsteem käivitab ja valmistab kaamera järgmiseks etapiks. Kontrolleriil küsitakse staatust iga 0.5 sekundi tagant.
- D1 = 3 – Ühendamine, süsteem alustab koordinaatide saatmist vastavalt kaameralt saadud piltidele. Kontrolleriile saadetakse iga pildi analüüsimise järel LED tuvastuselemendi koordinaadid või kokkulepitud sümbol, kui tuvastuselementi ei ole näha.

Tarkvara algoritmil (joonisel 4.4 ja joonisel 4.5) on samuti kujutatud kõik eelnevalt välja toodud funktsionaalsus.



Joonis 4.4 Kommunikatsiooni haldav tarkvara osa

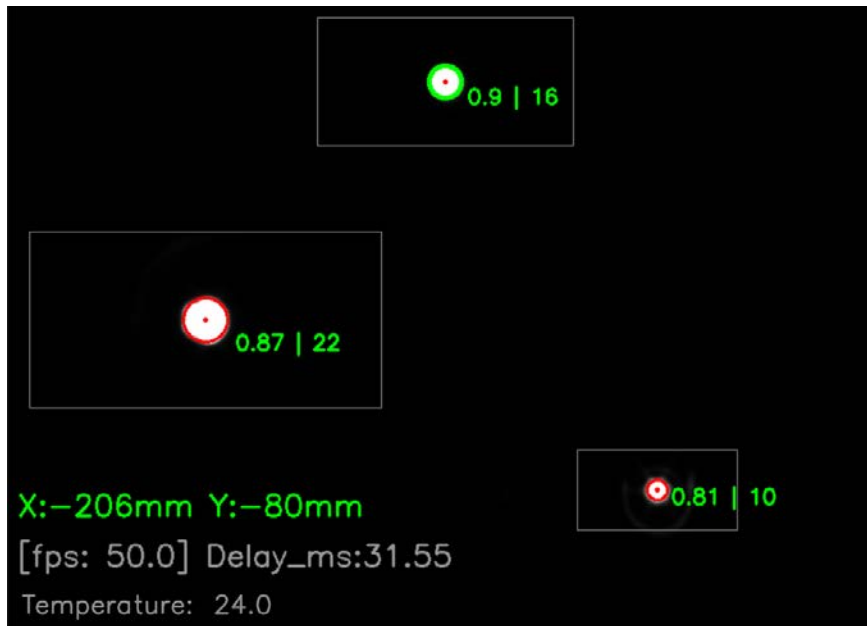
Kui kontrolleri annab signaali koordinaatide saatmiseks, läheb tarkvara pildianalüüsi etappi, mis on üles ehitatud suletud ringina mitme etapiga. Esiteks saadetakse kaamerale tarkvaraline signaal, et kaamera salvestaks pildi puhvermällu. Puhvermällust kopeeritakse pildi andmed, mis seejärel konverteeritakse $n \times d$ dimensioonidega maatriksiks, kus n ja d on vastavalt kaamera pildi laius ja kõrgus pikslites. Saadud korrektsete dimensioonidega pildile rakendatakse filtrid, mis tõstavad pildi kontrasti ning teevad kontuurid sujuvamaks, et järgmises etapis kasutatud CHT algoritmi tulemused oleksid töökindlamad. Pildile rakendatakse CHT algoritmi, mis leiab kõik ringikujulised elemendid ning nende raadiuse. Järgnevalt rakendatakse kõikidele leitud ringidele malli tuvastuse algoritmi, mis võrdleb tuvastatud ringi eeldefineeritud tuvastuselemendi malliga. Kui mallituvastuse algoritmi tulemus on piisav, saadetakse koordinaadid kontrolleriile ning kontrollitakse kas kontrolleri soovib jätkata. Kui kontrolleri soovib veel koordinaate, korratakse kogu ringi alates kaamerale tarkvaralise signaali saatmisest. Peale kontrolleriile koordinaatide saatmist salvestab tarkvara kaamera pildi videosse, mis ühendamise lõpetamisel saadetakse serverisse, et kaamera ja tarkvara tööd oleks võimalik jälgida ja vajadusel parandada. Joonisel 4.5 on esitatud masinõõnemise tarkvara osa algoritmi.



Joonis 4.5 Masinnägemise algoritme haldav tarkvara osa

4.5 Serverisse logifaili laadimine

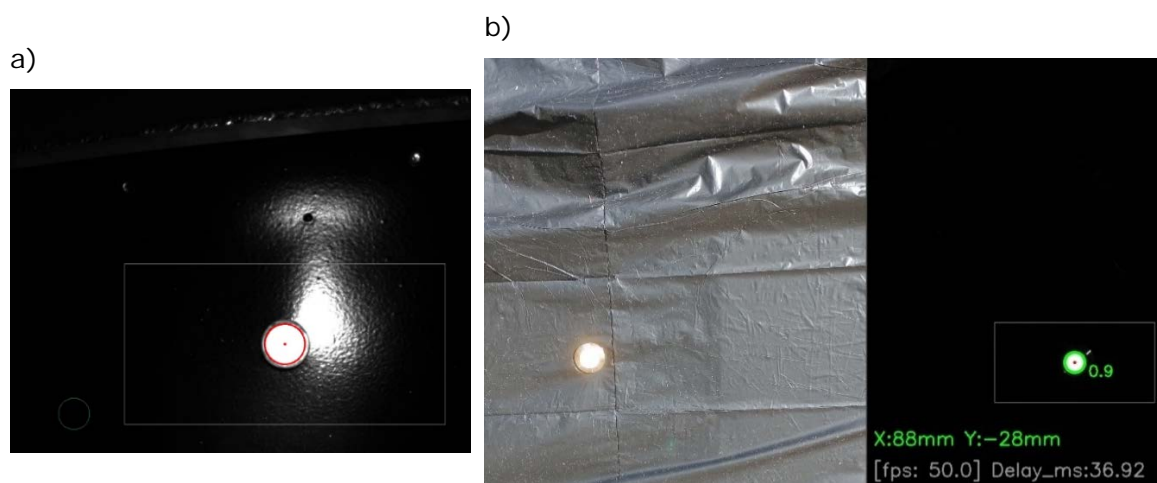
Kaamerasüsteemi tarkvarale on lisatud funktsionaalsus aktiivse kaamera piltide salvestamiseks koos oluliste andmetega, et peale kaamera töötamist oleks võimalik järgi vaadata, mida kaamera nägi ning milliseid koordinaate väljastas. Antud funktsionaalsust on võimalik ka välja lülitada, kui antud hetkel ei ole kaamera piltide salvestamine oluline. Kaamera salvestuse lõpetamisel koostab tarkvara piltidest video ning salvestab selle nii lokaalselt, kui ka laeb üle interneti serverisse, kuhu saab kõikjalt ligi. Videost on ekraanitõmmis välja toodud joonisel 4.6. Videos on ekraanile välja printitud tuvastuselemendi asukoht, koordinaadid ning malli tuvastuse tulemus roheliselt, vale kuju või heledusega häiringud on kujutatud punaselt ning pildi analüüsimiseks kulunud aeg ja väli temperatuurisensori andmed on printitud hallilt. Antud andmete abil on võimalik tuvastada potentsiaalseid tarkvara või riistvaralisi vigu ning neid parandada. Temperatuuriandmete abil saab tuvastada ka väli temperatuurist tekkinud häiringuid või analüüsida kütte lülitamisest tekkinuid muudatusi kaamera pildile. Videole printitud koordinaadi andmed on identsed kontrolleriile saadetud andmetega, mis võimaldab tuvastada potentsiaalseid kommunikatsiooni tõrkeid. Lisaks on tuvastatud elementide juurde printitud roheliselt malli sobitamise tulemus ning elemendi raadius, mille abil saab teada millise malliga tarkvara antud tuvastuselementi võrdles ning kuidas malli tuvastuse täpsus sõltub tuvastuselemendi lähenemisest või eemaldumisest.



Joonis 4.6 Logivideoost ekraanitömmis

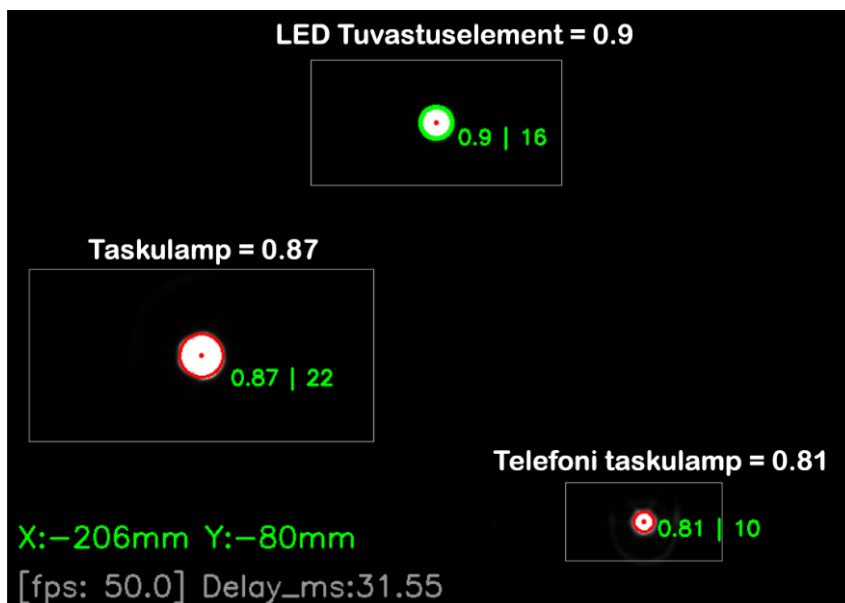
5. TULEMUSED

Masinnägemise süsteemi katsete läbiviimiseks ühendati Jetson Nano arvuti külge monitor, millele kuvati tarkvara abil kaamera pilt koos analüüsimiseks olulise infoga, mis on kirjeldatud peatükis 4.5. Kuvatud pildi abil (joonisel 4.6) oli võimalik seadistada kaamera sätteid, mis on esitatud ja selgitatud lisas 1 ning CHT parameetrid, mis on välja toodud peatükis 2.2. Seadistamisel selgus, et väli tingimustes peab tuvastuselement olema eredam kui päikese peegeldused. Vastaval juhul võivad päikese peegeldused, mis ei ole polariseeritud filtriga eemaldatud, tekitada häiringuid ja takistada tarkvaral korrektse elemendi leidmist. Antud situatsioon on kujutatud joonisel 5.1a ning sarnaste juhtumite vältimiseks võeti kasutusele võimsam tuvastuselement, mille valik on sooritatud peatükis 2.4. Võimsama elemendi abil sai kaamera säriaja muuta kõige lühemaks lubatud antud kaamerale ning lisaks eemaldati ka analoog võimendus kaamera sensorilt, mis enamgi tumendas pilti ning eemaldas peegeldused. Tulemus on esitatud joonisel 5.1b, kus peegeldused ning ere päike ei häiri süsteemi töökindlust.



Joonis 5.1 a) Nõrga tuvastuselemendiga häirisid väli tingimused tuvastamist ning b) Võimsama tuvastuselemendiga väli tingimused enam ei häiri töökindlust

Võimsama tuvastuselemendiga seadistati kaamera ja CHT parameetreid, kuni süsteem oli stabiilselt võimeline tuvastama tuvastuselementi nii ereda päikse käes, kui ka siseruumides. Seejärel, kui seaded olid paigas, rakendati süsteemile malli tuvastamise algoritm ning koostati vajalikud mallid tuvastuselemendist. Rakendatud malli tuvastamine võimaldab eristada sarnased objektid korrektsest tuvastuselemendist ning selle testimiseks viidi läbi katse, kus kaamera ette paigutati õige tuvastuselement, telefoni taskulamp ning suurem taskulamp. Joonisel 5.2 on välja toodud ekraanitõmmis antud katsetusest, kus malli tuvastamise algoritm edukalt eristas tuvastuselemendi teistest sarnastest objektidest.



Joonis 5.2 Edukas malli tuvastamise katsetus

Tarkvara tuvastas kõik kolm erineva suurusega elementi, kuid malli tuvastamise abil määrati igale elemendile sarnasuse väärtus, millest kõige suurem valiti välja, kuna see on kõige sarnasem eeldefineeritud mallile. Kui kaamerale oleks näha vaid üksainus tuvastuselement, siis on malli tuvastamise täpsus seadistatud lugema kujundit õigeks, kui see sarnaneb 60% väärtuses mallile. Malli tuvastamise abil eraldatud element on ääristatud rohelise ringiga ning ülejäänud elemendid, mille sarnasuse väärtus on madalam, on ääristatud punase joonega. Automaatika kontrolleriile edastatakse koordinaadid, mis on mõõdetud kõige kõrgema sarnasuse väärtusega elemendist, ehk roheliselt ääristatud ringi keskpunkti koordinaadid.

KOKKUVÕTE

Masinnägemise abil loodi pistikupesa leidmiseks süsteem, mida saab kasutada universaalselt erinevate robotiseeritud laadimissüsteemide automatiseerimiseks. Süsteem on suures osas koostatud valmiskomponentidest, mis on valitud põhirõhuga töökindlusel ning valitud kaamerasüsteem on veekindel kuni IP67, mis võimaldab paigaldamist ka karmidesse ilmastikuoludesse. Lisaks disainiti kaamerale soojendusega korpus, mis kaitseb otseste väli tingimuste eest ning hoiab soojust külmade temperatuuride juures. Vastavalt süsteemi tuvastuskauguse nõuetele valiti sobiva vaateväljaga objektiiv ning piisava resolutsiooniga kaamera. Pistikupesa leidmiseks valiti aktiivne tuvastuselement, mis paigaldatakse pesa lähendale ning elemendi leidmisel edastab süsteem koordinaadid automaatika kontrolleri (PLC). Kaaluti nii aktiivset kui ka passiivset tuvastuselementi, kuid aktiivne element võimaldab kaamera säriaja viia väga kiireks, mis eraldab aktiivse elemendi taustast ning teeb tuvastamise kiireks ja töökindlaks.

Süsteemi tarkvara koostati Python programmeerimiskeeles ning kasutab avatud lähtekoodiga masinnägemise teke algoritmide rakendamiseks. Tuvastuselemendi leidmiseks kasutab tarkvara Hough ringi operaatori ja malli tuvastamise algoritmi kombinatsiooni, mis võimaldab süsteemil edukalt eraldada eeldefineeritud tuvastuselemendi ka väga sarnaste objektide seast. Tarkvarale on lisatud konfiguratsioonivalikud, mille abil saab kiiresti lisada või eemaldada süsteemi tööülesandeid. Leitud tuvastuselemendi koordinaadid saadetakse üle Ethernet/IP kontrolleri ning tuvastusvigade ennetamiseks ja leidmiseks on koostatud videofailide logimise süsteem, mis laeb kaameralt saadud video serverisse ning võimaldab süsteemi töökindlust ja toimimist analüüsida üle interneti.

Töös jäid süvitsi uurimata mitmed aspektid ning täiendavalt saab uurida:

- Konkurentide ühendamissüsteemide tehnoloogiat. Kuna uuritud süsteemide tehnilisele infole ligipääs puudus töö kirjutamise ajal (peatükk 1), ei ole täpselt teada, kuidas välja toodud süsteemid toimivad,
- tarkvara ülesehitamist c++ programmeerimiskeeles ning selle eeliseid,
- värvi- või termokaamera kasutamist koos vastavate tuvastuselementidega,
- masinnägemise arvuti graafikaprotsessori efektiivsust ja selle suurendamist ning
- kaamerasüsteemi tuvastamise täpsust, korratavust ja määramatust.

KASUTATUD KIRJANDUSE LOETELU

- [1] Mobimar, Autonoomne laadimisüsteem praamidele [WWW] https://www.mobimar.com/application/files/2715/0460/5248/nector4000_brochure.pdf (05.05.2021)
- [2] NBCNEWS, Elektrilennuk sooritas esimese eduka lennu [WWW] <https://www.nbcnews.com/science/science-news/largest-electric-plane-yet-completed-its-first-flight-it-s-n1221401> (05.05.2021)
- [3] Justinas Miseikis, Matthias Ruther, Bernhard Walzel, Mario Hirz and Helmut Brunner, 3D Vision Guided Robotic Charging Station for Electric and Plug-in Hybrid Vehicles, 2017.
- [4] SICK, Inspector kaamera operaatori õpetus [WWW] https://cdn.sick.com/media/docs/6/36/836/Operating_instructions_Inspector_PIM60_ver_2.0_en_IM0048836.PDF (05.05.2021)
- [5] YouTube, Video Mobimari laadimissüsteemist [WWW] <https://www.youtube.com/watch?v=mYazeaclERM> (05.05.2021)
- [6] Medium, Jetson Nano ja Raspberry PI võrdlus [WWW] <https://medium.com/analytics-vidhya/jetson-nano-vs-raspberry-pi-e84642555adc> (05.05.2021)
- [7] OpenCV, CUDA tuumade selgitus ja näide [WWW] <https://opencv.org/platforms/cuda/> (05.05.2021)
- [8] NVIDIA, Erinevad arvutite moodulid [WWW] <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/> (05.05.2021)
- [9] SICK, Inspector masinnägemise kaamera [WWW] <https://www.sick.com/fi/en/machine-vision/2d-machine-vision/inspector/vspm-6f2113/p/p317945> (05.05.2021)
- [10] MOBIMAR, Autonoomne laadimissüsteem praamile [WWW] <https://www.mobimar.com/electric-propulsion-and-charging/charging-system> (05.05.2021)
- [11] NVIDIA, Jetson Nano arendaja komplekt [WWW] <https://developer.nvidia.com/embedded/jetson-nano-developer-kit> (05.05.2021)
- [12] Recab UK, Jetson Nano korpus [WWW] <https://shop.recabuk.com/diamond-systems-jetbox-nano-nvidia-jetson-enclosure-with-gpio-p-1841.html> (05.05.2021)
- [13] Advanced Illumination, Ringvalgusti kaamerale [WWW] <https://www.advancedillumination.com/products/df198-series/#> (05.05.2021)
- [14] T. S. Huang, Computer Vision: Evolution and Promise, 1996.
- [15] VLFeat, VLFeat avatud teegiga masinnägemise tarkvara koduleht [WWW] <https://www.vlfeat.org/> (05.05.2021)
- [16] Dlib, Dlib masinnägemise algoritmide tarkvara koduleht [WWW] <http://dlib.net/> (05.05.2021)
- [17] Richard O. Duda and Peter E. Hart. 1972. Use of the Hough transformation to detect lines and curves in Pictures, 1972.

- [18] Antoine Manzanera, Thanh Phuong Nguyen & Xiaolei Xu, Line and circle detection using dense one-to-one Hough transforms on greyscale images, 2016.
- [19] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, 1981.
- [20] T.J. Atherton, D.J. Kerbyson, Size invariant circle detection, 1999.
- [21] OpenCV, Dokumentatsiooni koduleht [WWW] <https://docs.opencv.org/3.4/> (05.05.2021)
- [22] ThinkLUCID, Triton kaamera seeria [WWW] <https://thinkLUCID.com/triton-gige-machine-vision/> (05.05.2021)
- [23] OpenCV, Pildipüramiidide teooria ning näide [WWW] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_pyramids/py_pyramids.html (05.05.2021)
- [24] ThinkLUCID, Tarkvara dokumentatsiooni ja allalaadimise leht [WWW] <https://thinkLUCID.com/downloads-hub/> (05.05.2021)
- [25] Kepware, Omron FINS http://www.kepware.com/Support_Center/SupportDocuments/Help/omron_fins_ethernet.pdf (05.05.2021)
- [26] Edmund Optics, UC seeria objektiivid [WWW] <https://www.edmundoptics.com/f/uc-series-fixed-focal-length-lenses/15027/> (05.05.2021)
- [27] IEEE, Toide üle Etherneti, IEEE 802.3af [WWW] https://standards.ieee.org/standard/802_3af-2003.html (05.05.2021)
- [28] Elfadistelec, Alternatiivne PoE toitesead [WWW] <https://www.elfadistelec.ee/et/poe-injector-for-gigabit-ieee-802-3af-2x-rj45-exsys-ex-6007poe/p/30162299?q=poe+injector&pos=1&origPos=1&origPageSize=50&track=true> (05.05.2021)
- [29] Elfadistelec, PoE toiteseadme andmeleht [WWW] https://www.elfadistelec.ee/Web/Downloads/_t/ds/IPOE-162_eng_tds.pdf (05.05.2021)
- [30] Edmund Optics, Masinnägemise fitrite tehnoloogia [WWW] <https://www.edmundoptics.eu/knowledge-center/application-notes/imaging/filtering-in-machine-vision/> (05.05.2021)
- [31] Farnell, Jaoturi andmeleht [WWW] <http://www.farnell.com/datasheets/2875246.pdf> (05.05.2021)
- [32] Networkworld, 9000-baidi suurused andmepaketid [WWW] <https://www.networkworld.com/article/2224722/jumbo-frames.html> (05.05.2021)
- [33] Riley E.J. Schnurr, Tony R. Walker, „Marine Transportation and Energy Use“
- [34] Ship-technology, Elektripraam Ampere [WWW] <https://www.ship-technology.com/projects/norled-zero-cat-electric-powered-ferry/> (05.05.2021)
- [35] Greenport, Esimene elektriline praam maailmas [WWW] <https://www.greenport.com/news101/energy-and-technology/making-the-transition-to-shore-power> (05.05.2021)

- [36] CAVOTEC, Automaatne pistikusüsteem [WWW]
<https://www.cavotec.com/en/your-applications/ports-maritime/crane-electrification/e-rtg/product-aps> (05.05.2021)
- [37] Praamid.ee, Esimene Eesti hübriidreisilaev Tõll [WWW]
<https://www.praamid.ee/wp/ts-laevade-parvlaev-toll-on-eesti-esimene-keskkonnasobralik-hubriidlaev/> (05.05.2021)
- [38] Icct, Kasvuhoonegaaside emisioonid laevandusest 2013-2015 [WWW]
https://theicct.org/sites/default/files/publications/Global-shipping-GHG-emissions-2013-2015_ICCT-Report_17102017_vF.pdf (05.05.2021)
- [39] IDTechEx, Elektriliste laevade prognoos 2021-2040 [WWW]
<https://www.idtechex.com/en/research-report/electric-leisure-and-sea-going-boats-and-ships-2021-2040/739> (05.05.2021)
- [40] Researchgate, CHT algoritmi kujutis [WWW]
https://www.researchgate.net/figure/Figure-3-Illustration-of-circular-Hough-transform_fig1_339189760 (05.05.2021)
- [41] NVIDIA, Jetson Nano andmeleht [WWW]
<https://developer.nvidia.com/embedded/downloads> (05.05.2021)
- [42] Aurora Lightning, LED tuvastuselement [WWW]
<https://www.ukelectricalsupplies.com/enlite-m-lite-pro-3w-3000k-ip68-led-marker-light.htm> (05.05.2021)
- [43] ThinkLUCID, M12 CAT6a kaabel [WWW] <https://thinkLUCID.com/product/m12-to-rj45-ip67-cat6a-cable-right-angle-down-15m-dark-green/> (05.05.2021)
- [44] ThinkLUCID, Gigabit ethernet ja Cat 6a kaabel [WWW]
<https://thinkLUCID.com/tech-briefs/gigabit-ethernet-the-efficient-and-reliable-interface/> (05.05.2021)
- [45] Shuttermuse, Kaamera vaatevälja arvutamine [WWW]
<https://shuttermuse.com/calculate-field-of-view-camera-lens/> (05.05.2021)
- [46] Weidmüller, Jaoturi andmeleht [WWW]
<https://www.tme.eu/Document/5e6b6785391115b971428ec0d31eedb4/243540000-EN.pdf> (05.05.2021)
- [47] Adafruit, Termostaadi andmeleht [WWW] <https://cdn-shop.adafruit.com/datasheets/DS18B20.pdf> (05.05.2021)
- [48] Nvidia, Garantii tagamine [WWW] <https://www.nvidia.com/en-us/support/warranty/> (05.05.2021)

- [49] DBK, Kütteelemendi andmeleht [WWW] <https://docs.rs-online.com/c3e0/0900766b8142252f.pdf> (05.05.2021)
- [50] ITT GROUP, Releemoodul [WWW] <https://robo-labor.ee/et/muundurid/862-releemoodul-1x10a.html> (05.05.2021)
- [51] Python, PEP 8 stiiljuhend Python programmeerimiskeelele [WWW] <https://www.python.org/dev/peps/pep-0008/> (05.05.2021)
- [52] ThinkLUCID, Raspberry PI Etherneti paketi määramine [WWW] <https://support.thinklucid.com/knowledgebase/jumbo-frames-on-raspberry-pi/> (05.05.2021)
- [53] U.S. Automation, Inc., Ringvalgusti diagramm [WWW] <https://usautocorp.com/product/smart-series-ring-illuminators/> (05.05.2021)

LISAD

Lisa 1 Kaamera seadete konfiguratsioon

Kaamera seadistati kasutades Arena API Python programmeerimiskeele moodulit. Koodilõigus on sinise kirjaga kujutatud eestikeelset kommentaari eelnevale koodireale.

```
nodemap['Width'].value = 1920
# Pildi laius pikslites
nodemap['Height'].value = 1200
# Pildi kõrgus pikslites
nodemap['OffsetX'].value = 0
# Pildi nihe kaamera sensori null-punkti suhtes, pikslites
nodemap['Gain'].value = 1.0
# Sensori analoog võimendus
nodemap['AcquisitionFrameRateEnable'].value = False
# Kaamera kasutab varieeruvat kaadrisagedust, vastavalt programmi kiirusele
nodemap['TriggerMode'].value = 'On'
# Kaamera teeb pilte peale signaali saamist
nodemap['TriggerSource'].value = 'Software'
# Signaali tüüp on digitaalne
nodemap['TriggerSelector'].value = 'FrameStart'
# Signaali vastuvõtmisel tehakse pilt
nodemap['TriggerActivation'].value = 'LevelHigh'
# Signaal aktiveeritakse kõrge loogilise väärtusega
nodemap['ExposureAuto'].value = 'Off'
# Kaamera on seatud manuaalsele säriajale
nodemap['ExposureTime'].value = 40.0
# Kaamera säriaja väärtus mikrosekundites
nodemap['DeviceStreamChannelPacketSize'].value = 9000
# Kaamera Ethernet paketi suurus
```

Kood 5.1 Koodilõik, mis seadistab kaamera

Lisa 2 Lõik parameetrite seadistuse moodulist

```
LED_DIAMETER_MM = 16

HOUGH_MIN_RADIUS_PX = 10
HOUGH_MAX_RADIUS_PX = 100
HOUGH_MIN_DISTANCE = 250
HOUGH_PARAM_1 = 250
HOUGH_PARAM_2 = 14

TEMPLATE_MATCH_THRESHOLD = 0.8

PROJECT_PATH = "/home/machine-vision-controller/machine-vision-controller/"
LOG_FOLDER_PATH_NOT_UPLOADED = "/home/machine-vision-controller/Desktop/mvc_video_log_not_uploaded/"
LOG_FOLDER_PATH_UPLOADED = "/home/machine-vision-controller/Desktop/mvc_video_log_uploaded/"
DELETE_OLD_LOGS = False
DELETE_OLD_LOGS_TIME = 30

ACQUISITION_FRAME_RATE = 50.0 # (maximum frame rate)
EXPOSURE_TIME = 40.0
GAIN = 0.0
FRAME_WIDTH = 1920
FRAME_HEIGHT = 1200
FRAME_OFFSET = 0

MEDIAN_BLUR = 5

GRAY_FILTER_UPPER = 255
GRAY_FILTER_LOWER = 230

RADIUS_FILTER_HEIGHT = 4
RADIUS_FILTER_WIDTH = 8
```

Kood.5.2 Lõik moodulist, mis haldab kõiki muudetavaid parameetreid