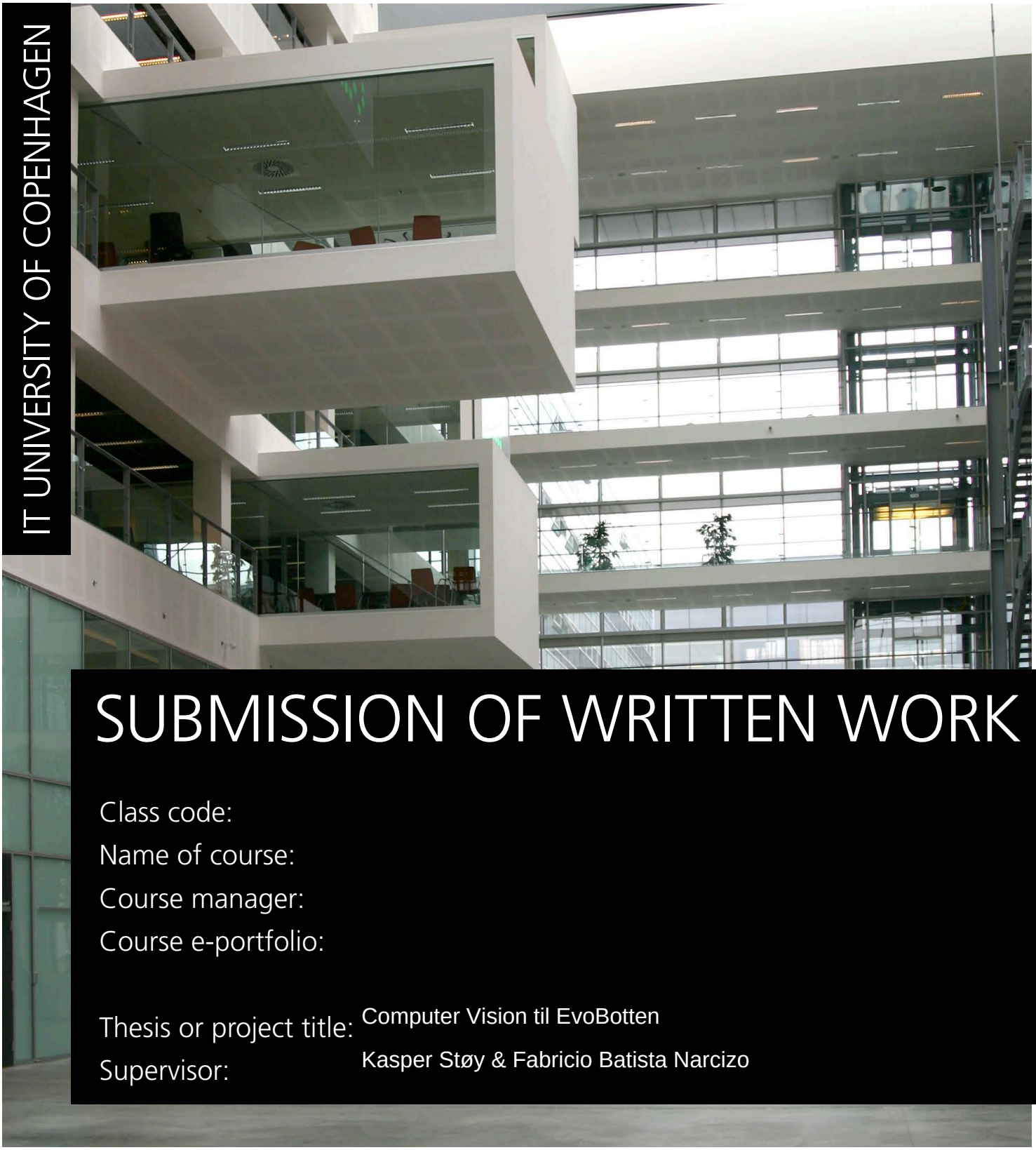
IT UNIVERSITY OF COPENHAGEN



SUBMISSION OF WRITTEN WORK

Class code:

Name of course:

Course manager:

Course e-portfolio:

Thesis or project title: Computer Vision til EvoBotten

Supervisor: Kasper Støy & Fabricio Batista Narcizo

	Full Name:	Birthdate (dd/mm-yyyy):	E-mail:
1.	Kasper Schnack	25/09-1986	ksch@itu.dk
2.	Sebastian Brerup	05/11-1987	sbre@itu.dk
3.			@itu.dk
4.			@itu.dk
5.			@itu.dk
6.			@itu.dk
7.			@itu.dk

Computer Vision til EvoBotten

Af Kasper Schnack & Sebastian Brerup
Vejledt af Kasper Støy & Fabricio Batista Narcizo

Københavns IT-Universitet
København, Danmark, 2016

Number of pages: 80,

Published at the it university of copenhagen *June 1, 2016*

Abstract

Automated production using robots play a significant role in chemistry, biotechnology and microbiology. Robots that are designed to perform a specific task are in the long run cheaper and much more efficient than humans. In research, however, most tasks are done by humans even though many tasks are cumbersome and repetitive. What complicates the introduction of robots in research are the small variations in the task sequences that are frequently introduced.

In this Master thesis we will contribute to a project called the EvoBot which seeks to make robots an integral part of research in order to lower costs and speed up the process of experimenting. We develop a proof-of-concept for a computer vision application for the EvoBot that enables it to find, locate and classify petri dishes and well plates. We present the design of the system implemented using the OpenCV framework along with physical modifications to the EvoBot. In addition to the vision system we develop a framework to test its precision and accuracy and lay ground work for future improvements. We evaluate different approaches based on accuracy and precision results of the detection methods that are experimented with. The evaluation indicates that detection without the use of tagging is feasible for use in the industry, with the introduction of some future improvements.

Indhold

Abstract **iii**

1 Introduktion **1**
1.1 Læsevejledning . . . 2
1.2 Problemstilling . . . 3
1.2.1 Problemfelt . . . 3
1.2.2 Problemformulering . . . 5
Arbejdsspørgsmål . . . 6
1.2.3 Overordnet afgrænsning . . . 6
1.3 Baggrund . . . 7
1.3.1 Motivation . . . 7
1.3.2 Væskehåndteringsrobotter . . . 8
EvoBot . . . 8
Andre væskehåndteringsrobotter . . . 9
1.3.3 OpenCV . . . 11

2 Computer Vision **12**
2.1 Forarbejdelse . . . 12
2.1.1 Farverum . . . 12
2.1.2 Sløring . . . 13
2.2 Segmentering . . . 14
2.2.1 Tærskling . . . 14
2.2.2 Canny Kantdetektor . . . 15
2.2.3 Sobel Edge Detection . . . 16
Tilbage til Canny . . . 17
2.2.4 Hysteresis Tærskling . . . 17
2.3 Repræsentation . . . 18
2.3.1 Hough Transform . . . 18
2.3.2 Hough Circle Transform . . . 21
2.3.3 Find Contours . . . 21

2.3.4 Polygon Approksimering . . . 23
2.4 Genkendelse . . . 24

3 Metode **25**
3.1 Computer Vision Script . . . 25
3.2 Fremstilling af datasættene . . . 26
3.2.1 Objekters placering . . . 27
3.2.2 Belysning . . . 29
Det Visuelle Spektrum . . . 29
Infrarød Belysning . . . 30
3.2.3 Opstilling af forsøget . . . 31
3.3 Ground Truth . . . 33
3.4 Evaluering af datasættene . . . 33
3.4.1 Robusthed . . . 34
3.4.2 Nøjagtighed & Præcision . . . 35
3.4.3 Ydeevne . . . 35
3.5 Fejlkilder . . . 35
3.5.1 Metodiske fejlkilder . . . 35
3.5.2 Svagheder ved vores system . . . 36
3.5.3 Menneskelige Faktorer . . . 36
3.5.4 Teori og praksis . . . 37

4 Implementering **38**
4.1 Klasser . . . 39
4.1.1 Image . . . 39
4.2 Computer Vision Systemet . . . 39
4.2.1 Processeringsværktøj . . . 39
4.2.2 Petriskåle . . . 40
4.2.3 Brøndplader . . . 40
4.3 Annoteringsværktøjet . . . 44
4.4 JSON Data . . . 44
4.5 Sammenligningsværktøjet . . . 45

5 Analyse **48**
5.1 Analysestrategi . . . 48
5.2 Overordnede resultater . . . 49
5.3 Datasæt 1: Det Visuelle Spektrum . . . 51

5.4 Datasæt 2: Infrarød belysning, Linux 55
5.5 Datasæt 3: Infrarød Belysning, Mac 61
5.6 Delkonklusion 67

6 Diskussion **68**
6.1 Datasæt 68
6.2 Præcision 72
6.3 Fejlkilder 74

7 Konklusion **76**

8 Perspektivering **79**

Bibliografi **81**

A Appendiks **84**
A.1 Samlet for alle datasæt 84
A.2 Datasæt 1 84
A.3 Datasæt 2 91
A.4 Datasæt 3 118

Figuroversigt

1.1 Brøndplade 4
1.2 Kalibrering af EvotBottens kamera i RealLab 4
1.3 EvoBotten i RealLab 8
1.4 EvoBottens tre lag 10
1.5 aBioBot 11

2.1 Udjævning 13
2.2 Bilateral sløring 14
2.3 'Salt og Peber'-støj 14
2.4 Canny Kantdetektion 15
2.5 Sobel Filtervindue 16
2.6 Resultat af Sobel Filter 16
2.7 Kantkandidater til Canny 17
2.8 Hysteresis Tærskling 18
2.9 Resultat af Canny 19
2.10 Hough Transform 19
2.11 Hough Line Transform 20
2.12 Table1 Suzuki paper 23
2.13 Ramer-Douglas-Peucker-algoritmen 24

3.1 Preliminær forsøgsopstilling 27
3.2 Endelig forsøgsopstilling 28
3.3 Forsøgsopstilling med spejling 28
3.4 Brøndpladerotationer 29
3.5 EvoBotten og den lystætte kasse 30
3.6 Farvespektrum 31
3.7 Datafremstilling, petriskåle 32

4.1 Systemoverblik 38
4.2 crcl_detect 41
4.3 Processeringdel af sqr_detect 41

4.4 Billedbearbejdelse - dilation 42
4.5 Findcontours i sqr_detect 42
4.6 Angle_cos i sqr_detect 43

5.1 Normalfordeling alle datasæt 50
5.2 Petriskål - 18 lysniveau 51
5.3 Petriskål - 38 lysniveau 52
5.4 Normalfordeling Datasæt 1 53
5.5 Datasæt 1, Billede 007 54
5.6 Datasæt 1, Billede 009 54
5.7 Datasæt 1, Billede 013 56
5.8 Normalfordeling Datasæt 2 57
5.9 Datasæt 2, Billede 112 59
5.10 Datasæt 2, Billede 082 60
5.11 Datasæt 3, Billede 027 63
5.12 Datasæt 3, Billede 045 63
5.13 Datasæt 3, Billede 142 64
5.14 Normalfordeling i datasæt 3 65
5.15 Datasæt 3, Billede 081 66

6.1 Støj i billedet efter at baggrunden er fratrukket 2 69
6.2 Støj i billedet efter at baggrunden er fratrukket 69
6.3 IR-LED kredsløb 70
6.4 Lysfiltre 70
6.5 Petriskål i infrarødt lys 71

8.1 aBioBots detektionssystem 80

Tabeloversigt

5.1 Nøjagtighed på tværs af alle datasæt. 49
5.2 Præcision på tværs af alle datasæt. 50
5.3 Nøjagtighed for datasæt 1. 51
5.4 Præcision for petriskåle i datasæt 1. 52
5.5 Præcision for billeder af interesse. 55
5.6 Nøjagtighed for datasæt 2. 56
5.7 Præcision for brøndplader i datasæt 2. 56
5.8 Gennemsnitlig afvigelse for brøndplader ud fra placering og rotation i datasæt 2. 58
5.9 Afvigelser for billedgruppen 111 til 120. 58
5.10 Afvigelser for billedserien 081 til 090. 60
5.11 Afvigelser for billedgruppen 051 til 060. 61
5.12 Nøjagtighed for datasæt 3. 62
5.13 Nøjagtighed for datasæt 3 (fordelt på afskærmning). 62
5.14 Præcision for brøndplader i datasæt 3. 64
5.15 Gennemsnitlig afvigelse for brøndplader ud fra placering og rotation i datasæt 3. 65

6.1 Areal som billederne dækker for alle datasæt. 73
6.2 px/cm horisontalt og vertikalt for alle datasæt. 73
6.3 Samlet px/cm for alle datasæt. 73
6.4 Afvigelser for alle datasæt i pixels. 74
6.5 Omregnede afvigelser for alle datasæt i pixels. 74

Kapitel 1

Introduktion

Computer Vision er et forskningsområde, der beskæftiger sig med at få computere til at kunne analysere og forstå information i billeder på samme abstraktionsniveau som mennesker. Forskningsfeltet har efterhånden mange praktiske anvendelsesmuligheder som bl.a. ansigtsgenkendelse i billeder, automatisk nummerpladegenkendelse på overvågningskameraer, selvkørende biler og gøre robotter i stand til at manipulere med objekter uden brug af fiksturer og faste placeringer. Det sidste eksempel er udgangspunktet for dette projekt. Helt specifikt er formålet med nærværende projekt at udvikle et 'proof-of-concept' af et detektionssystem til væskehåndteringsrobotten, EvoBot. EvoBot er et forskningsprojekt under ledelse af Kasper Støy[1], som har til formål at skabe den næste generation af automatiseret laboratorieudstyr. EvoBotten er stadig under udvikling og mangler en række funktionaliteter, før den kan tages i brug. En af funktionaliteterne er et detektionssystem til at spore objekter vha. Computer Vision, således at EvoBotten kan interagere med objekterne uden behov for manuelt input. Projektet bidrager desuden med en række anbefalinger til videreudvikling af systemet baseret på fund fra analysen. Ligeledes et framework til at teste nøjagtighed og præcision af detektionssystemet og dets fremtidige versioner.

Sourcefilerne til projektet findes på det åbne repository `https://bitbucket.org/speciale/evobotcv/`. Her findes datasættene ligeledes.

[1]Kasper Støy er Associate Professor på ITU, hvor han forsker i robotteknologi og kunstig intelligens(*Kasper Stoy - IT University of Copenhagen*).

1.1 Læsevejledning

I det første kapitel redegør vi for projektets problemstilling. Først beskriver vi den generelle problemstilling, der forsøges løst med EvoBotten. Dernæst giver vi en konkret og afgrænset beskrivelse af de problemstillinger, som vi arbejder med i specialet. Det leder til en beskrivelse af vores motivation og mål med specialet. Til sidst beskriver vi EvoBotten og OpenCV, som er det Computer Vision bibliotek, vi har anvendt i implementeringen af systemet.

I andet kapitel giver vi en mere detaljeret redegørelse for Computer Vision. Vi beskriver de metoder, som vi anvender i de forskellige faser af Computer Vision: forarbejdelse af billeder, segmentering, repræsentation og genkendelse. Vi diskuterer ligeledes løbende de tilgange, vi har fravalgt.

For at etablere et fundament for Computer Vision til EvoBotten og for at afdække kvaliteten af vores system, gennemfører vi en række forsøg med EvoBotten. I kapitel tre gennemgår vi de metodiske overvejelser, vi har gjort i designet af forsøgene og i udviklingen af detektionssystemet. Vi diskuterer og forklarer ligeledes, hvordan vi har udformet de datasæt, som vi tester vores system imod og hvilke parametre vi ser på.

I kapitel fire redegør vi for implementeringen af systemet. Herunder detektionssystemet, annoteringsværktøjet og sammenligningsværktøjet. Vi illustrerer ligeledes en grundlæggende struktur i vores system. I kapitel fem analyserer vi testresultaterne med fokus på nøjagtighed og præcision. Vi ser på, hvordan lysindstilling, placering og rotation af objekter påvirker resultaterne.

I kapitel seks diskuterer vi fejl, mangler, resultater og fund mht. implementeringen af detektionssystemet. Med baggrund i analysen og hele den eksperimentelle proces reflekterer vi over de valg, vi har foretaget og hvordan det påvirker resultaterne som helhed.

I kapitel syv vender vi tilbage til problemformuleringen og besvarer den ud fra den viden, vi har samlet.

I kapitel otte giver vi vores bud på, hvordan man i fremtiden kan videreudvikle detektionssystemet til EvoBotten.

God læselyst.

1.2 Problemstilling

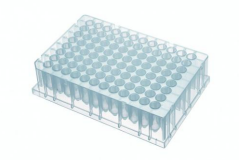
1.2.1 Problemfelt

Inden for kemi, bioteknologi, mikrobiologi og farmaci, spiller robotter en stadig større rolle i store dele af produktionen. Årsagen er, at robotter er hurtigere, mere præcise og langt mere effektive end mennesker, når de opererer med ensartede opgaver i et kontrolleret miljø.

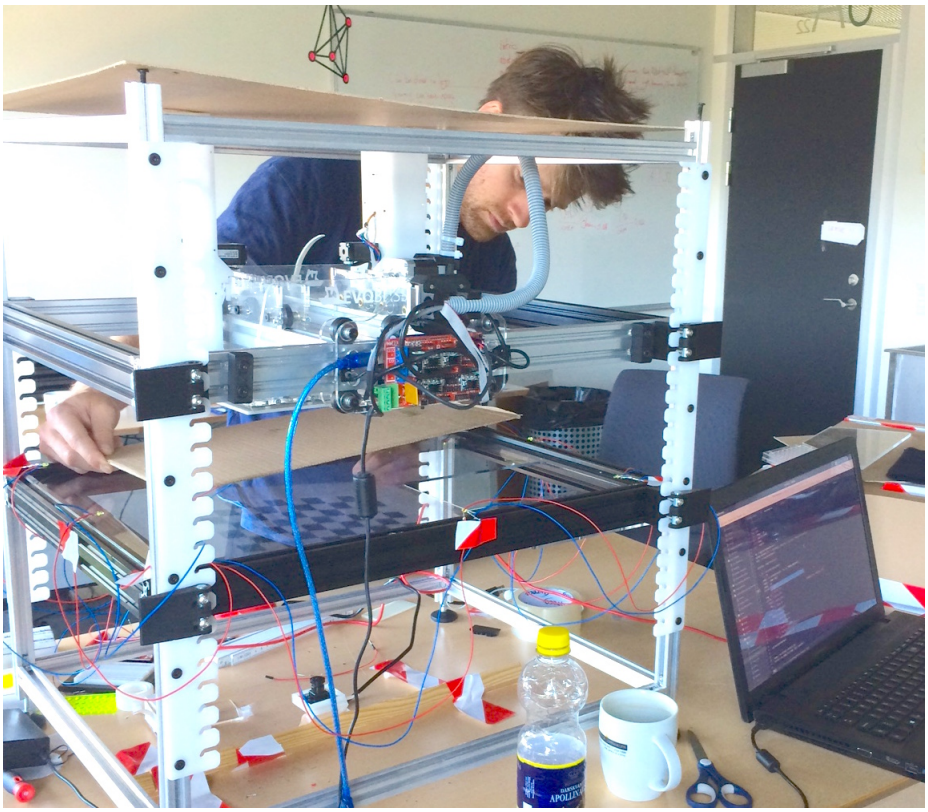
Ser man på den eksperimentelle del af industrien, anvendes robotter ikke i samme grad. Det skyldes, at opgaverne er langt mere varierende. Selvom der er mange repetitive opgaver forbundet med eksperimenter, er sekvensen og substansen af dem ofte skiftende. Variationen gør det vanskeligt at automatisere opgaverne. Et eksempel på en opgave kan være pipettering af forskellige væsker fra petriskåle til et større antal reagensglas. Hvis væskemængden og/eller væskesammensætningen varierer fra det ene eksperiment til det næste, vil det ved brug af en robot kræve omprogrammering, hvilket ofte er omkostningsfuld og tidskrævende. I et vist omfang bliver denne slags opgaver stadig udført af laboranter, trods de både er trivielle og indebærer risiko for fejl. Udfordringen er, at de robotter der anvendes i dag, er specialiserede og kun kan udføre et begrænset antal af de opgaver, der er behov for (Nejatimoharrami et al., 2016).

I projektet EvoBot ønskes dette forhold ændret. Robotten skal fungere som assistent i den eksperimentelle del af industrien ved at automatisere disse laboratorieopgaver. EvoBotten er dog ikke den eneste af sin slags. Der findes andre væskehåndteringsrobotter, som er i stand til at udføre lignende opgaver. I mange af de nuværende robotter er objekter som petriskåle og brøndplader fikseret, hvilket begrænser fleksibiliteten. Her kræves det ligeledes, at laboranten lære at anvende disse robotter. Ideen med EvoBot er, at interaktion i et laboratorie skal foregå i forlængelse af laborantens naturlige arbejdsform. EvoBotten eksisterer endnu kun i en tidlig prototype, men den er i stand til udfører bevægelser og pipetteringer baseret på koordinater. Dette foregår på en glasplade af størrelsen 40 x 40 cm hvor laboratorieobjekterne placeres (se figur 1.2 for et billede af EvoBotten). Ved at forsyne EvoBotten med Computer Vision, vil den være i stand til at identificere og lokalisere petriskåle og brøndplader (se figur 1.1), når de bliver placeret på pladen. Således er det muligt at udføre varierede forsøg uden at skulle rekonfigurere systemet mellem hvert forsøg.

For at tydeliggøre hvilken forskel et detektionssystem kan gøre, vil vi vise et tænkt eksempel på en manøvre som EvoBotten kunne udføre og hvordan den kunne forsimples



Figur 1.1: Brøndplade.



Figur 1.2: På billedet ses en igangværende kalibrering af EvoBottens kamera.

ved anvendelse af Computer Vision:

1. Indsæt petriskål med centrum i (x1, y1).
2. Indsæt reagensglas med centrum i (x2, y2).
3. Ryk til koordinat (x1, y1).
4. Sænk pipetten ned i petriskålen.
5. Sug fem milliliter væske op.
6. Løft pipetten op.
7. Ryk til koordinat (x2, y2).
8. Sænk pipetten ned i reagensglasset.
9. Udløs væsken i reagensglasset.

Dette skal så gentages X antal gange (instruktion 3-9). Det ønskede resultat kræver væsentligt færre input fra brugeren:

1. Indsæt petriskål.
2. Indsæt reagensglas.
3. Indtast X millimeter, der skal overføres imellem hvilke objekter.

Herefter skal robotten selv kunne genkende laboratorieobjekterne på den eksperimentelle overflade ved hjælp af Computer Vision og påbegynde pipettering. Den største udfordring ved at implementere et sådant system er, at der kræves høj præcision og nøjagtighed for at kunne tage det i betragtning som et reelt alternativ til eksisterende metoder. Vi vil derfor opstille en række forsøg og udvikle et system til at belyse hvornår detektionssystemet fejler.

1.2.2 Problemformulering

På baggrund af ovenstående når vi frem til følgende problemformulering:

Hvordan kan vi, ved hjælp af Computer Vision, skabe et fundament for at gøre EvoBotten i stand til at identificere og lokalisere petriskåle og brøndplader uden brug af tagging?

Arbejdsspørgsmål

Formålet med dette speciale er altså at bygge et fundament for detektionssystemet til EvoBotten. Projektet er tidsbegrænset og vi forventer ikke at kunne præsentere et færdigt Visionsystem klar til industriel brug. I stedet vil vi udvikle et proof-of-concept, som viser, at det kan lade sig gøre at finde petriskåle og brøndplader uden brug af tagging. Hertil udvikler vi et system, der kan anvendes til at måle nøjagtighed og præcision for dette og fremtidige Visionsystemer. Desuden vil vi bidrage med en række datasæt, som uden videre kan bruges til at teste softwareforbedringer. Det er vores vurdering at disse tre elementer vil danne et solidt grundlag for det videre arbejde. Konkrete arbejdsspørgsmål vi har i fokus, er således:

1. Hvilke Computer Vision metoder kan anvendes til at detektere petriskåle og brøndplader?
2. Hvordan kan vi teste Computer Vision systemet?
3. Hvordan påvirkes Visionsystemet af lys og objektplaceringer?

Disse tre spørgsmål er dækkende for vores tilgang til projektet.

1.2.3 Overordnet afgrænsning

Computer Vision er ikke som de klassiske naturvidenskaber, der bruger almengyldige lovmæssigheder som byggesten til at udlede generaliserbare teorier. Forskningen foregår i et kaotisk miljø med mange ubekendte faktorer. Man er samtidigt nødt til at arbejde på et højt abstraktionsniveau for at få resultater, der har praktisk anvendelse. Dette medfører, at feltet får et mere eksperimenterende og casebaseret islæt. Af samme årsag vil vi ikke forsøge at lave generelle bidrag til Computer Vision som videnskab, men nøjes med at drage konklusioner på baggrund af de observationer vi har gjort i det brugsscenarie, som er specifikt for EvoBotten.

Hvis der ses bort fra alle begrænsninger, ville det perfekte Computer Vision system for EvoBotten være:

1. I stand til at identificere og lokalisere alle laboratorieobjekter med 100% nøjagtighed og præcision (altså være robust).
2. I stand til at detektere objekter uden forsinkelse (have høj ydeevne).

3. Fejlfrit implementeret.

4. Fuldt integreret med EvoBotten.

Af disse fire områder har vi valgt at fokusere på to og afgrænse os fra de to andre:

Robustheden er det primære fokus i dette projekt. Robustheden dækker over begreberne nøjagtighed og præcision, og kan måles på forskellige parametre, som diskuteres i metoden, kapitel 3. Vi vil ligeledes gerne have et højt fokus på testning af systemet, så vi kan dokumentere systemets evne til at detektere objekter. Dette diskuteres også i metodeafsnittet.

En høj ydeevne betyder at genkendelse og lokalisering af objekterne sker indenfor acceptable tidsintervaller. Om end det er noget vi har haft for øje i selve udviklingen af systemet, har vi valgt at afgrænse os fra at teste dette. Da vi udelukkende fokuserer på Computer Vision, inddrager vi ikke potentielle brugere. Det er derfor vanskeligt at sige noget kvalificeret om hvilke tidsintervaller, der er acceptable i laboranters naturlige arbejdsgange.

Der eksisterer i forvejen et integrationsAPI til EvoBotten og da selve integrationen er uden for sigtet af dette projekt, vil vi nøjes med at referere til dette.

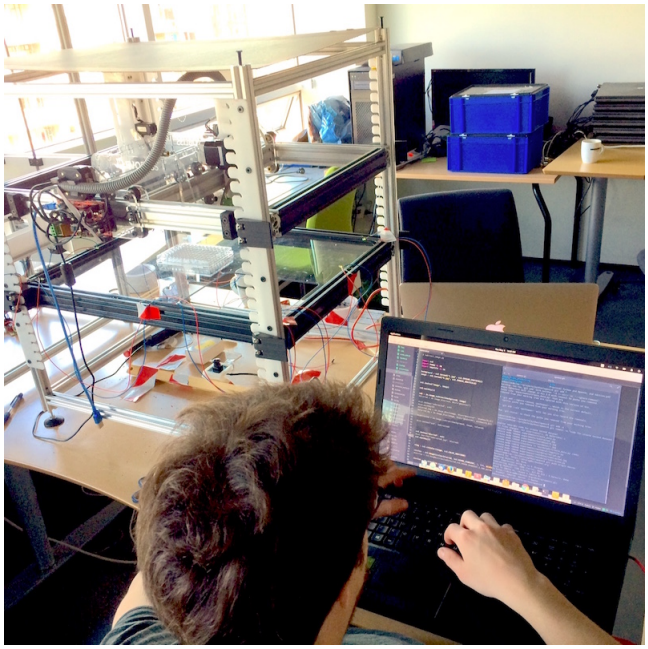
Foruden case-afgrænsning til EvoBotten og yderligere afgrænsning i forhold til Computer Vision delen af denne case, vil der forekomme flere afgrænsninger, som beskrives løbende.

1.3 Baggrund

1.3.1 Motivation

Ud over ønsket om at studere Computer Vision, bunder vores interesse for EvoBot-projektet i at skrive et industrielt speciale og arbejde med løsninger, som er praktisk applicérbare.

Inden vi besluttede os for at gå ind i projektet, havde ingen af os erfaring med Computer Vision. Men vi har begge en stor passion for softwareudvikling og den tekniske tilgang til videnskab. Vi er motiveret af at lære Computer Vision, som i de sidste årtier har gjort store landvindinger bl.a. båret frem af den teknologiske udvikling indenfor processorkraft og den øgede interesse for feltet i Tech-industrien.



FIGUR 1.3: EvoBotten i RealLab.

1.3.2 Væskehåndteringsrobotter

Væskehåndtering spiller en afgørende rolle i forskning indenfor biovidenskab(Kong et al., 2012). For at hjælpe processen og minimere fejltagelser under forsøg, eksisterer der en række robotter, der skal assistere disse forsøg (Kong et al., 2012).

EvoBot

EvoBotten er en væskehåndteringsrobot bygget af dele fra en 3d printer. Den skal bruges til at automatisere laboratorieforsøg inden for feltet kemisk kunstigt liv. I første omgang er sigtet de dele af forsøgene, som har at gøre med væskehåndtering. Kemisk kunstigt liv (herfra kaldet kunstigt liv) er en tværfaglig disciplin, som har til formål at skabe systemer, der ligner biologiske organismer eller som ud fra bestemte kriterier for liv, kan kaldes levende (*kunstigt liv | Gyldendal - Den Store Danske*).

Som nævnt findes der andre væskehåndteringsrobotter, hvoraf de fleste benytter sig af fiksturer til håndtering af laboratorieobjekter. Disse fiksturer omslutter oftest laboratorieobjekterne i uigennemsigtige materialer, som gør det vanskeligt at observere de kemiske processer i petriskålen. Dette er en nødvendighed, når man forsker i kunstigt liv og

ønsker at følge cellers udvikling over tid. For at opnå en automatisering her, skal robotten ikke kun kunne opsætte forsøget, men også udføre automatisk overvågning, logge interessante resultater og stoppe et forsøg, der går galt. Dette forsøges integreret med EvoBotten ved at lave alle eksperimenter på en glasplade uden fiksturer, så forsøgene kan overvåges fra alle sider.

EvoBotten forsøger desuden at løse, er den praktiske udfordring der opstår ved anvendelse af væskehåndteringsrobotter i eksperimenter indenfor kunstigt liv. Eftersom disse eksperimenter er meget dynamiske i sin karakter, kræver det en lang række løbende rekonfigurering og omprogrammering under et forsøg med almindelige væskehåndteringsrobotter. Der fire egenskaber som EvoBotten forsøger at inkorporere:

- Rekonfigurerbarhed
- Alsidighed
- Lav i pris
- Mulighed for udvidelser

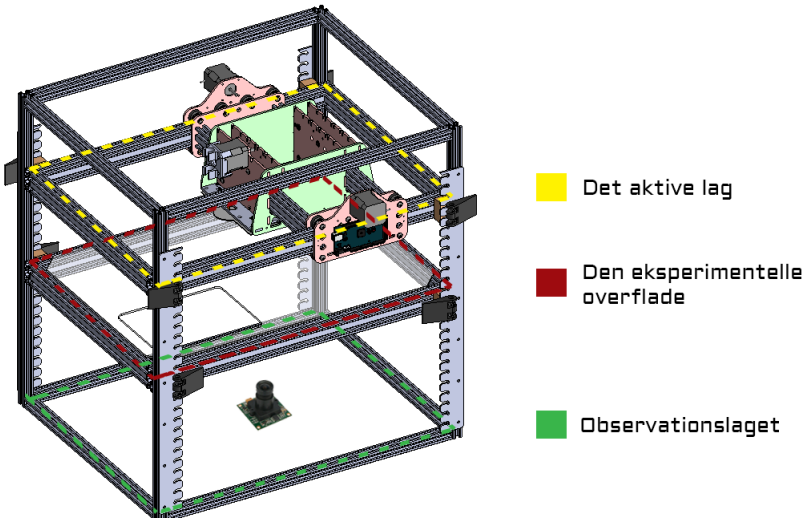
For at opnå dette udnyttes principper fra feltet Modular Robotics (Yim et al., 2007). Pointen med denne tilgang er, at man kan indkapsle kompleksiteten og samtidig have et simple plug-and-play interface.

EvoBotten er designet således, at den består af en strukturel ramme og tre lag. Opdelingen er følgende: Det øverste lag er det aktive lag. Her sidder pipetten og styretøjet. Det midterste lag er den eksperimentelle overflade, hvor væskerne og beholderne befinder sig. Det nederste lag er observationslaget - her er kameraet påsat Nejatimoharrami et al., 2016. Se figur 1.4.

EvoBottens kamera er placeret i bunden, hvilket er et simpel men afgørende designvalg, som adskilder EvoBotten fra andre væskehåndteringsrobotter. Dette giver fuldt udsyn til det eksperimentelle lag imens der udføres operationer fra det aktive lag.

Andre væskehåndteringsrobotter

Der findes en række væskehåndteringsrobotter i produktion som automatiserer operationer, som f.eks. pipettering. Her ser vi på nogle eksisterende løsninger: Equator GX8

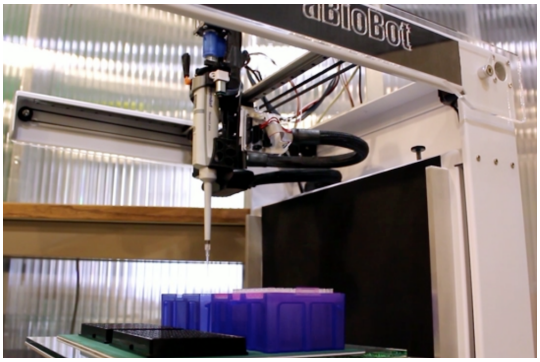


Figur 1.4: EvoBottens tre lag.

Dispenser (Deerac Fluidics, 2005), Thermo Scientific Multidrop Combi nL (Thermo Scientific, 2011), CO-RE 384 Probe Head (*384-Probe Head*) og Micro10x Robotic Benchtop 12-Channel Dispenser (*Robotic Dispenser Liquid Handling Robot - Micro10x*^{TM}).

Disse væskehåndteringsrobotter er alle designet til at kunne håndtere meget små mængder væske. Equator GX8 og Thermo Scientific Multidrop Combi nL kan håndtere helt ned til 0,05 mikroliter væske. Til forskel fra EvoBotten er de anvendte brøndplader fikseret og robotterne har ikke indbygget Visionsystemer. De er således bygget med det ene formål at håndtere væsker og ikke til at overvåge forsøg.

Der findes dog en robot under udvikling, som er designet til at løse nogle af de samme problemstillinger som EvoBotten, nemlig aBioBot (se figur 1.5). Præcis som det er tilfældet med EvoBotten, er der her fokus på at gøre væskehåndteringsrobotter mere brugervenlige og at billige (aBioBot, 2015a). Robotten er endnu ikke i produktion, men designet er færdigt og består af tre hoveddele: LabBench, Yan og 'The Bot'. LabBench er brugergrænsefladen, der er udviklet i HTML5, som logger alle eksperimenter i skyen. Yan er robottens Computer Vision system, der ligesom vores system, er udviklet i OpenCV. Den sidste del er selve robotten, som de selv beskriver som en modificeret 3D printer. Hele frameworket for robotten er bygget på open source hardware og software og LabBench og Yan vil blive tilgængelige via åbne API'er (aBioBot, 2015b). Til forskel fra EvoBotten er aBioBottens kamera placeret over forsøgene. Computer Vision systemet Yan er ikke tilgængeligt i skrivende stund, hvilket ellers kunne have bidraget til specialet.



Figur 1.5: Prototype af aBioBot.

1.3.3 OpenCV

OpenCV er et open source Computer Vision framework, som vi har anvendt til at implementere Computer Vision som en del af vores system. Vi har anvendt version 3.0.0. OpenCV kan anvendes i C++, C, Java og Python. Vi har skrevet vores implementering i Python.

Kapitel 2

Computer Vision

I litteraturen er der flere bud på, hvordan Computer Vision skal faseinddeles (Paulsen and Moeslund, 2012, p. 4). Vi læner os op ad Paulsens (2012) beskrivelse af de fem faser i Computer Vision: billedanskaffelse, forarbejdelse, segmentering, repræsentation og genkendelse. Forarbejdelse er den bearbejdelse, der sker globalt på billedet. Herunder sløring og gråskalakonvertering. Segmentering er bearbejdelse af lokale dele af billedet, der ønskes fremhævet, herunder tærskling og kantdetektion. Repræsentation er hvor information oversættes til objekter, der er fundet. Her bruges algoritmer som f.eks. Hough Circle Transform, der finder cirkler i billedet og repræsenterer dem som et centerkoordinat og en radius. Det sidste skridt er genkendelse, hvor der udføres abstrakt tolkning på data. I vores tilfælde genkendes cirkler som petriskåle og rektangler som brøndplader.

Billedanskaffelse foretages før både bearbejdelse og analyse. Vi har derfor valgt at behandle det separat i kapitel 3 (metode).

2.1 Forarbejdelse

2.1.1 Farverum

I Computer Vision anskues et gråtonebillede som et todimensionelt array af pixler, der hver især har en intensitet mellem 0 og 255 (fra sort til hvid) (Adrian, 2015, pp. 18-19). Der arbejdes desuden også med forskellige farverum. Et velkendt farverum er RGB som har tre kanaler med todimensionelle array af pixler (Red, Green, Blue). Almindelige farvebilleder bruger dette format. Et billede i dette farverum kræver altså tre gange så mange operationer. Derfor bliver billeder ofte konverteret til gråtoneskala, så der kun

Figur 2.1: På billedet ses effekten af udjævning påført i stigende grad fra venstre til højre.
(Adrian, 2015)

opereres på en kanal. De billeder vi analyserer vil derfor være konverteret fra RGB til gråtoner.

2.1.2 Sløring

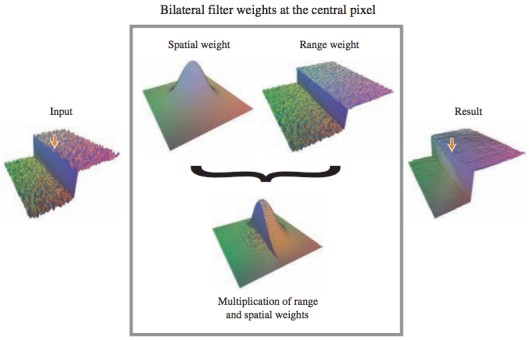
Sløring er en udtværing af billedet, som sammenblander nærliggende pixlers intensiteter. Sløring bruges til at fjerne støj i et billede og kendetegnes ved at detaljerne i billedet forsvinder (se figur 2.1). I de fleste tilfælde fungerer ting som kantdetektion og tærskling bedre efter sløring. Vi har taget fire former for sløring i betragtning: Udjævning, Gaussisk, Bilateral og Median.

Når man udfører Udjævning bruges der et $k \times k$ pixler vindue, som kaldes 'convolution kernel'. Her skal k være et ulige heltal (Adrian, 2015, p. 106). Vinduet køres horisontalt og vertikalt over billedet. Ved hver placering bliver den midterste pixels intensitet ændret til gennemsnittet af summen af intensiteter i vinduet.

Gaussisk sløring fungerer på samme måde, men i stedet for at at tage gennemsnittet, tages et vægtet gennemsnit. Pixler vægtes tungere jo tættere de befinder sig på pixlen i midten. På denne måde bevares kanterne bedre.

Bilateral sløring er langsommere end sine modstykker. Dog er den effektiv til at fjerne støj og samtidig bevare kanter. Den fungerer ved at bruge Gaussisk sløring og samtidig at vægte de intensiteter højest som ligger tæt på pixlens egen intensitet (Paris et al., 2009). Det ses illustreret i figur 2.2. Som vi skal se, benytter vi denne sløringsmetode til at detektere brøndplader.

Mediansløring fungerer som udjævning, men i stedet for gennemsnittet tages medianen af værdierne i det vindue man har valgt. Dette er særligt brugbart, hvis man vil fjerne



Figur 2.2: På figuren illustreres hvordan bilateral sløring bevarer kanter. (coldvision, 2016)

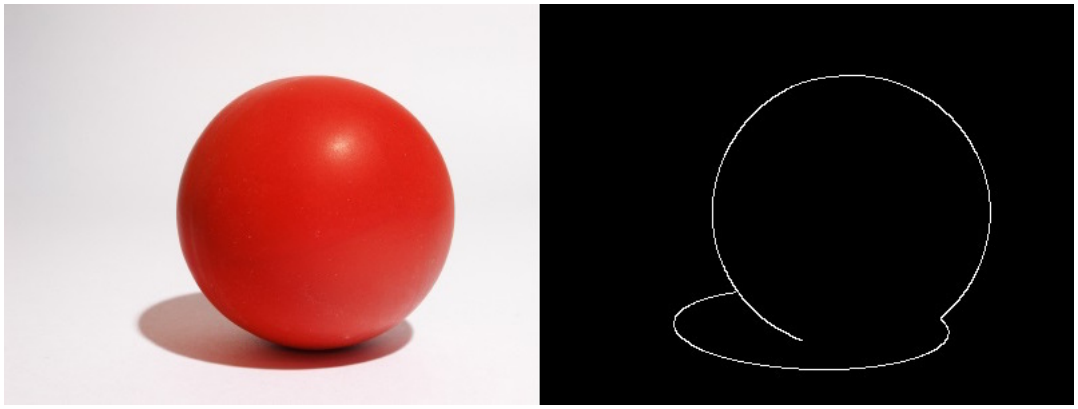
Figur 2.3: Øverst til højre ses det originale billede. Billedet bliver påført 'salt og pepper'-støj. På de to nederste billeder ses det hvordan Medianansløring fjerner støjen og bevarer billedet. (*'Salt and Pepper' noise reduction*)

'salt og peber'-støj fra billedet, da den midterste pixel udskiftes med intensiteten af en pixel, som allerede findes i vinduet. (Adrian, 2015, pp. 104-111). Et eksempel med 'salt og pepper'-støj ses i figur 2.3.

2.2 Segmentering

2.2.1 Tærskling

Tærskling er grundlæggende en binær konvertering af et billede. Konverteringen foregår ved, at hver pixel får en intensitet på enten 0 eller 255 (altså sort eller hvid), alt efter om



Figur 2.4: Canny påføres for at finde kanten for bolden og dens skygge.
(*java - Canny edge detector in OpenCV - Stack Overflow*)

pixlens intensitet er over eller under den angivne tærskel (Adrian, 2015, pp. 155-156).

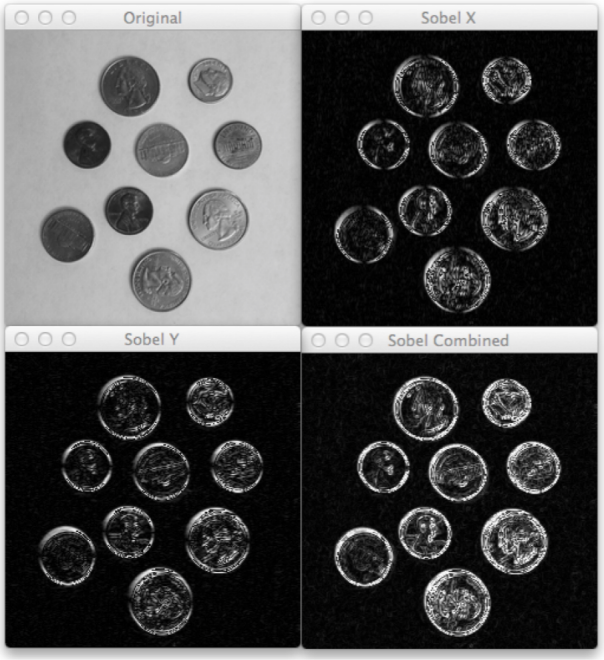
2.2.2 Canny Kantdetektor

Når man i Computer Vision behandler et billede, er det oftest for at kunne udtrække informationer, som kan repræsenteres symbolsk eller numerisk. F.eks. at tælle antallet af bolde i et billede. For at kunne detektere bolde, er man nødt til at kunne adskille objekterne i et billede fra hinanden. En effektiv metode til det, er at søge efter områder, hvor der er stor forskel i billedets intensitet. En stor forskel i intensitet er typisk en indikation på et område som adskiller to forskellige objekter, eller med andre ord, annoterer en kant. I figur 2.4 er der stor forskel i intensitet mellem den hvide baggrund og kanten langs bolden og dens skygge. I venstre side kan man se hvordan Canny adskiller baggrunden fra bolden - noteret ved den hvide linje. Hvordan det lader sig gøre kigger vi nærmere på herunder.

Canny Kantdetektor er en metode, som bliver benyttet som et af de sidste led i billedsegmentering, inden man begynder at identificere objekter i billedet (Bradski and Kaehler, 2008, p. 222). Canny bygger på kantfindermetoden Sobel, som derfor altid anvendes som led i Canny. For at give en bedre forståelse af Canny redegør vi derfor først for Sobel-metoden (Adrian, 2015, pp. 133-134).

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

Figur 2.5: Sobel Filtervindue.
(Adrian, 2015)

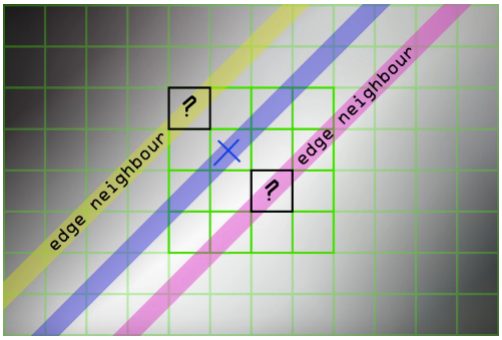


Figur 2.6: Resultat af Sobel Filter.
(Adrian, 2015)

2.2.3 Sobel Edge Detection

Sobel kantdetektor (også kaldet Sobel operatoren) bruges til at finde kanter i et billede. Kanter beskrives, i Computer Vision sammenhæng, som områder i et billede, hvor der sker en skarp ændring i intensitet eller farve. Men eftersom både Sobel og Canny bruger sort-hvid billeder som input, er det kun ændringer i intensiteter, som for os er interessante. Sobel fungerer ved at påføre et horisontalt og et vertikalt filter, som ses på figur 2.5 i den rækkefølge (Adrian, 2015).

Som det kan ses på figur 2.6 giver filtret en respons, når det støder på en kant, som er orienteret samme vej som filtret. Responsen kommer til udtryk vel høje pixelintensiteter, hvor der findes en kant. De steder hvor der ikke er kanter, vil højre og venstre/øvre og



Figur 2.7: Billedet illustrerer tre pixels på tværs af gradienten der bliver taget i betragtning som kantkandidat.
(Computerphile, 2015)

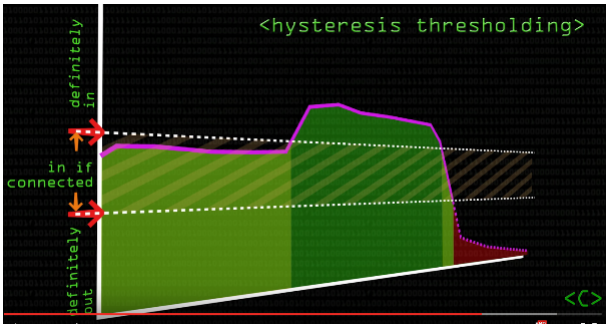
nedre del af sobelfiltret udligne hinanden og give en lav respons. Når vi lægger de to resulterende billeder sammen får vi summen af kanterne i billedet 'Sobel Combined' (Adrian, 2015, pp. 127-133).

Tilbage til Canny

Som det måske kan ses på figur 2.6, er kanterne i Sobel ikke annoteret med en enkelt streg, men derimod med et spektrum af intensiteter i glidende overgang, som alle repræsenterer kandidater til en kant. Det vil vi gerne have ændret, eftersom det er væsentligt nemmere at arbejde med en binær annotering af, hvad der er en kant og hvad der ikke er. Canny udvælger den stærkeste af kantkandidaterne for at kunne repræsentere kanten med en enkelt pixel og dermed kunne udlede objektets konturer (Bradski and Kaehler, 2008, p. 152). Den stærkeste kantkandidat er den pixel, der langs med kanten, har den højeste intensitet af sine kantnaboer. Et eksempel ses på figur 2.7, hvor algoritmen skal vælge imellem pixlen markeret med x eller de to naboer med spørgsmålstegn, alt efter hvilken har den højeste intensitet. Se figur 2.7.

2.2.4 Hysteresis Tærskling

Ud over at finde konturen, fjerner Canny også uønskede/uinteressante kanter, som f.eks. er genereret af støj. Dette gøres ved hjælp af Hysteresis Tærskling. Hysteresis Tærskling er en to-skridt-tærskling: først findes de dominerende kanter, som er kanter med høj pixelintensitet. Disse kanter vil, hvis de har intensitet over den øverste tærskel, blive bevaret. Uønskede kanter, som har lavere intensitet en den nedre tærskel, vil blive



Figur 2.8: Billedet illustrerer en 1D repræsentation af en kant på et billede, hvorpå der bliver udført hysteresis tærskling. Langs x-aksen ses en kant og på y-aksen ses kantens intensitet. Kanten er annoteret med en lilla streg og hysteresistærsklerne er annoteret med hvide stiplede linjer. Den del af kanten som ligger over tærsklen, bliver registreret som et kantstykke. Den del der ligger i det skraverede område imellem de to tærskler vil, fordi det ligger i forlængelse af et registreret kantstykke, også blive registreret som et kantstykke. Den sidste del af kanten, som ligger under tærsklen, vil ikke blive registreret som et kantstykke.
(Computerphile, 2015)

kasseret. Imellem de to tærskler vil kanterne kun blive bevaret, hvis de er direkte forbundet til dominerende kanter. Det undersøges ved at følge kanten fra dominerende pixels for at se om, der er forbundne kanter, som kunne være relevante at medtage (Bradski and Kaehler, 2008, p. 152). Metoden er illustreret i figur 2.8.

Canny kantdetektor finder altså konturerne i et billede, som adskiller områder med høj intensitetsskift - typisk områder, der adskiller forskellige objekter. I dette projekt bruger vi Canny, når vi skal finde brøndplader og vi bruger Sobel, når vi skal finde petriskåle.

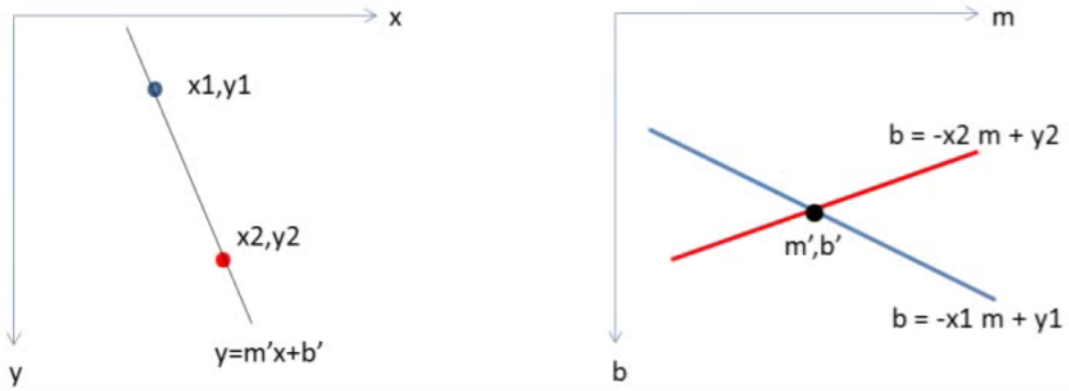
2.3 Repræsentation

2.3.1 Hough Transform

I dette afsnit beskriver vi den generelle metode Hough Transform og varianten Hough Circle Transform. Herunder ser vi på, hvad metoderne kan bruges til og hvordan de fungerer.

Hough Transform er en metode som anvendes bruges til at detektere simple geometriske objekter, eller objekter som kan parametriseres. Det første metoden gør er at transformere billedet til binær form. Et binært billede er et sort-hvid billede, som kun har to

Figur 2.9: Eksempel på resultat af Canny metoden.
(Bradski and Kaehler, 2008)

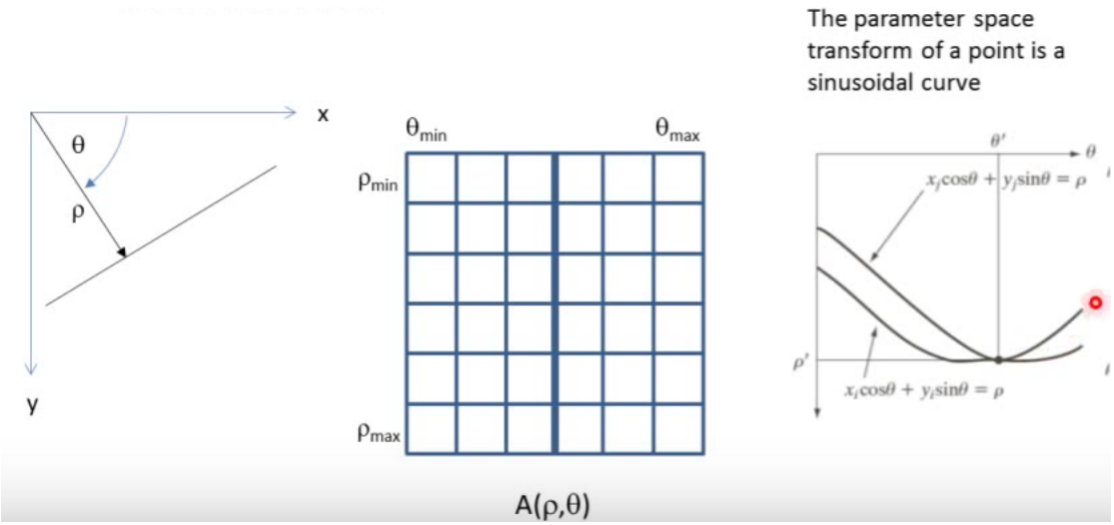


Figur 2.10: I ovenstående figur ses hvordan to punkter på en linje i xy-planet danner to linjer, som skærer hinanden i (a, b) i det nye plan.
(William Hoff, 2012)

intensiteter; typisk 0 og 255. Det binære billede vil typisk være et resultat af Canny.

Efter Canny metoden er blevet anvendt, kan Hough Transform lede efter geometriske former. Metoden fungerer ved at lade hver pixel stemme for tendenser i billedet. Det forklares bedst ved at se på metodens simpleste implementering, nemlig Hough Line Transform. Denne metode har til formål at finde lige linjer, der kan beskrives matematisk med den lineære funktion $y = ax + b$, hvor a er hældningskoefficienten og b er konstantleddet. For at finde tendenserne i billedet bytter vi rundt på konstanter og variabler og omskriver funktionen til $b = x_0a + y_0$.

Ved at lave denne omskrivning vil hver pixel i det oprindelige billede, blive repræsenteret ved en lige linje i det plan, som a og b udspænder. Linjerne i ab-planet beskriver alle de linjer, som punktet i xy-planet kan ligge på. Som det ses på figur 2.10 vil to pixler, der ligger på samme linje i xy-planet, danne linjer i ab-planet, som skærer hinanden i punktet (a, b). Her er a hældningskoefficienten til linjen og b er konstantleddet til den



Figur 2.11: Figuren viser implementeringen af Hough Line Transform. (William Hoff, 2012)

tilsvarende linje i xy-planet. Måden man finder linjerne i xy-planet er altså ved at kigge på hvor i ab-planet, der er skæringspunkter, hvori mange linjer skærer hinanden. Man siger, at linjerne i ab-planet stemmer om en linje i xy-planet. Hver skæring i et punkt giver én stemme for en linje. Mange stemmer er en indikator på, at der er mange pixels som ligger på en linje i xy-planet og altså danner en linje. Da hver enkelt pixel har en stemme, er metoden ret robust overfor brudte linjer og anden støj.

Denne implementering er praktisk til at give en forståelse for principperne bag Hough Transform. Men den har én ulempe, nemlig at hældningskoefficienten til en vertikal linje er uendelig stor. Derfor implementeres Hough Line Transform med en anden funktion for en ret linje:

$\rho = x\cos(\theta) + y\sin(\theta)$, hvor ρ er længden på linjen, $l1$, der dannes ortogonalt på linjen, $l2$, til (0,0). θ er vinklen fra x-aksen til $l1$. Dette illustreres i figur 2.11.

Eftersom vi ikke benytter os af denne metode, vil vi ikke gå dybere ind i forklaringen af denne. I stedet vil vi nu forklare Hough Circle Transform, som vi bruger til at detektere petriskålene (Bradski and Kaehler, 2008, pp. 153-158).

2.3.2 Hough Circle Transform

Hough Circle Transform, er en variant af Hough Transform, som bruges til at finde cirkler i et billede. Det er denne metode vi bruger til at finde petriskåle. De grundlæggende principper, som blev beskrevet i sidste afsnit, bliver også anvendt i denne metode. Var fremgangsmåden direkte overført, ville man være nødt til at plotte alle potentielle cirkler for hver pixel. Da en cirkel har tre parametre (et x og et y koordinat for cirklens centrum og en radius), ville det hurtigt kræve meget høj regnekapacitet at finde skæringspunkter i rummet og derefter finde maxima. Den mere effektive fremgangsmåde som bliver benyttet, er ved at opdele metoden i to. Således begrænses kompleksiteten til ét array i to dimensioner og ét array i én dimension. Det sker som følgende:

1. Først findes cirklens centrum. Her udnyttes det at man med Sobel kan få kantens gradient. Med gradienten er det muligt at tegne en ortogonal linje til hver kant i *ab*-planet. Nu kan vi benytte os af afstemning til at finde cirklernes centrum. I de punkter hvor mange af de ortogonale linjer skærer hinanden, er en indikation på et fælles centrum og dermed for en cirkel.

2. Når vi har cirklernes centrum, kan vi finde deres radier ved, i én dimension, at stemme for antallet af kanter indenfor et interval af mulige radier. Ved de radier hvor vi får en høj respons i det originale billede, er der stor sandsynlighed for at have en cirkel.

(Bradski and Kaehler, 2008, pp. 158-161) Ved at dele metoden op, kan man på denne måde trinvis finde, først cirklernes centre og dernæst deres radier.

2.3.3 Find Contours

Metoden vi har brugt til at finde brøndplader i vores datasæt, cv2.findContours, er en OpenCV-metode, der finder konturer i billeder. Modsat Hough Transform søger metoden ikke efter objekter med specifikke parametre. I stedet bruges en grænsefølgealgoritme til at kortlægge koordinater for objekter i et billede. Da algoritmen altså ikke diskriminerer og dermed inkluderer alle de kanter den finder, er den også meget følsom overfor støj i billedet. Dette stiller høje krav til forarbejdelse og billedets kvalitet. Metoden kan derfor ikke stå alene; eftersom vi udelukkende vil se på rektangulære objekter, er vi nødt til at efterbehandle konturerne, som bliver fundet af findContours. Dette kommer vi nærmere ind på i næste afsnit. I dette afsnit beskriver vi grænsefølgealgoritmen, som ligger til grund for cv2.findContours metoden. Algoritmen ser således ud:

For et binært inputbillede $F = \{f_{ij}\}$ hvor i er rækkerne i billedet og j er kolonnerne. NBD, som er det sekventielle nummer for den nuværende grænse, sættes til 1 til at starte med som også bliver nummeret rundt i kanten af billedet. I LNBD gemmes det sekventielle nummer for den sidste grænse vi er stødt på. Hver gang vi begynder at skanne en ny række sættes LNBD = 1. Nu udføres et 'raster scan', altså et skan hvor der itereres over hver pixel. Der skelnes mellem ydre og indre grænser. For de pixels for hvilke $f_{ij} \neq 0$ udføres følgende:

1. Vælg en af følgende:
   (a) Hvis $f_{ij} = 1$ og $f_{i,j-1} = 0$ besluttes det at pixlen (i, j) er startpunkt for en ydre grænse. $NBD \leftarrow NBD + 1$ og $(i_2, j_2) \leftarrow (i, j - 1)$.
   (b) Ellers hvis $f_{ij} \geq 1$ og $f_{i,j+1} = 0$ besluttes det at pixlen (i, j) er start punkt for en indre grænse.
   (c) Ellers gå til punkt (4).
2. Brug figur: 2.12 til at finde den rigtige forælder til den fundne grænse.
3. Fra startpunktet (i, j) følg grænsen ved at følge (3.1)-(3.5).
   3.1. Med udgangspunkt i (i_2, j_2), se med urets retning på de pixels, som støder op til (i, j). Den første pixel med intensitet over nul bliver kaldt (i_1, j_1). Bliver der ikke fundet nogen pixel med intensitet over nul sæt $f_{ij} \leftarrow -NBD$ og gå til (4).
   3.2. $(i_2, j_2) \leftarrow (i_1, j_1)$ og $(i_3, j_3) \leftarrow (i, j)$.
   3.3. Med udgangspunkt i pixlen efter (i_2, j_2), se mod urets retning på hver pixel der støder op til (i_3, j_3). Den første pixel med intensitet over nul kaldes for (i_4, j_4).
   3.4. Sæt værdien for f_{i_3,j_3} på følgende måde:
      i. Hvis pixlen $(i_3, j_3 + 1)$ er en 0-pixel og den blev undersøgt i (3.3): sæt $f_{i_3,j_3} \leftarrow -NBD$.
      ii. Ellers hvis $f_{i_3,j_3} = 1$: sæt $f_{i_3,j_3} \leftarrow NBD$.
      iii. Ellers lad f_{i_3,j_3} beholde den værdi den har i forvejen.

TABLE 1
Decision Rule for the Parent Border of the Newly Found Border B

Type of the border B' with the sequential number LNBD / Type of B	Outer border	Hole border
Outer border	The parent border of the border B'	The border B'
Hole border	The border B'	The parent border of the border B'

FIGUR 2.12: Tabellen bruges til at afgøre hvilken 'forælder' den nyligst fundne kontur har.
(Satoshi and Keiichi, 1985)

3.5. Hvis $(i_4, j_4) = (i, j)$ og $(i_3, j_3) = (i_1, j_1)$ er grænsen blevet fulgt hele vejen rundt. Hvis det er tilfældet: gå til (4). Ellers $(i_2, j_2) \leftarrow (i_3, j_3)$ og $(i_3, j_3) \leftarrow (i_4, j_4)$ og gå til (3.3).

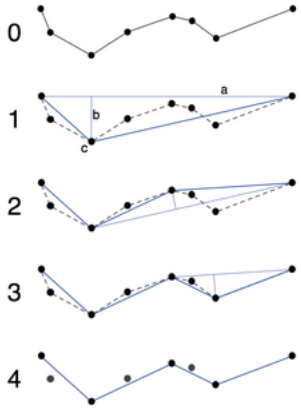
4. Hvis $f_{ij} \neq 1$: $LNBD \leftarrow |f_{ij}|$ og genoptag raster scan fra $(i, j + 1)$. Algoritmen afsluttes, når raster scan når til nederste højre hjørne af billedet.

(Satoshi and Keiichi, 1985)

2.3.4 Polygon Approksimering

cv2.findContours returnerer en liste med koordinater for alle hjørnerne i de konturer, den finder. For at finde ud af hvilket objekt den returnerer, bruger vi cv2.approxPolyDP. Denne metode bygger på Ramer–Douglas–Peucker algoritmen, som har til formål at forsimple et polygon ved at skære hjørner fra. I dette afsnit gives der en konceptuel forklaring af metoden. Input er en liste af koordinater, som repræsenterer et polygon. Epsilon er den brugerdefinerede tærskel som bestemmer hvilke koordinatsæt, som skal indgå i det endelige polygon.

1. Det første algoritmen gør, er at tage linjen imellem første koordinatsæt og sidste koordinatsættet i listen i betragtning. Disse punkter er automatisk en del af det nye polygon.



Figur 2.13: Figuren illustrerer Ramer–Douglas–Peucker algoritmen appliceret på et sæt af koordinater.
(*Polygon Simplification Algorithm*)

2. Herefter udregnes afstanden mellem linjen og de resterende koordinater i listen. Hvis afstanden til det fjerneste koordinat (x_1, y_1) er over epsilon vil dette punkt blive inkluderet i det nye polygon ved at forbinde startpunktet med (x_1, y_1) og (x_1, y_1) med slutpunktet.

3. Algoritmen kalder rekursivt sig selv, hver gang et nyt koordinatsæt bliver inkluderet. Hvis (x_1, y_1) bliver tilføjet kalder algoritmen sig selv, først med listen af koordinater, der udspændes af [startpunktet til (x_1, y_1)] og så med [(x_1, y_1) til slutpunktet].

Algoritmen er illustreret i figur 2.13.

Ved hjælp af cv2.findContours() og cv2.approxPolyDP() er vi under de rigtige forhold i stand til at finde brøndplader. Hvilke parametre vi bruger til metoderne, kommer vi ind på, når vi beskriver implementeringen i kapitel 4.

2.4 Genkendelse

Når vi får en repræsentation af objekterne foregår genkendelsesprocessen implicit. Dette skyldes at vi kan parametrisere de objekter, som vi ønsker at finde og at vi ikke på nuværende tidspunkt laver nogen unik identifikation/adskillelse af objekter som er ens. Med andre ord: Vi ved at systemet har fundet en petriskål når det detekterer en cirkel og at det har fundet en brøndplade når det detekterer et rektangel.

Kapitel 3

Metode

For at skabe et dokumenteret fundament for Computer Vision til EvoBotten, gennemfører vi en række forsøg. Forsøgene skal både vejlede processen for systemet, der skal implementeres, og til sidst udmønte sig i en række anbefalinger til det videre arbejde. For at uddybe denne tankegang, skal hele processen deles i fem stadier i sekvens:

1. Udvikling af Computer Vision Script
2. Fremstilling af data
3. Ground Truthing
4. Tests
5. Evaluering

Under arbejdet med Computer Vision er vi blevet klar over, at vi beskæftiger os i et felt med uendeligt mange ubekendte variabler. Det er altså ikke muligt at afdække alle mulige opstillinger. Vi har derfor været nødt til lave en afvejning af hvilke variabler, der med størst sandsynlighed vil have en betydelig påvirkning på udfaldet af forsøgene og prioriteret derefter. Der er ikke nogen eksakt model for at finde de rigtige parametre, og derfor er det nødvendigt at gøre nogle antagelser og have en eksperimentel tilgang. For at dokumentere processen bedst muligt, beskriver vi hvilke afgrænsninger, vi har foretaget, og diskuterer, hvordan vores valg kan påvirke det endelige resultat.

3.1 Computer Vision Script

I den første del af projektperioden har videnindsamlingen været udpræget eksplorativ og eksperimenterende. Den første prototype af visionsystemet er testet på et lille datasæt

med tilfældige objektplaceringer. Fokus i denne del af projektet har været at danne os et overblik over potentielt interessante Computer Vision metoder, som kan anvendes til at genkende petriskåle og brøndplader. En række vigtige valg blev foretaget i denne proces, herunder hvilke metoder vi anvender i de forskellige lag (forarbejdelse, segmentering og genkendelse). Det kan derfor ikke udelukkes, at der eksisterer andre metoder, der kan give anderledes resultater.

Vi forestillede os tidligt i forløbet, at en af de største udfordringer ville være at detektere glas på en gennemsigtig overflade. Det viste sig ikke at være et særlig stort problem, da petriskålens kant blev kraftigt markeret af måden lyset faldt på. Herudover kan petriskåle parametriseres som cirkler, hvilket gør det ligetil at finde dem på vores billeder. Dette skyldes at der ikke er andre objekter i billederne, som har samme form.

Brøndplader er en smule anderledes. Vi forestillede os ligeledes, at det ville være svært at finde dem da de også er gennemsigtige. Problemet viste sig i højere grad at hænge sammen med støj i billederne og parametriseringen af brøndpladerne.

I forlængelse af dette har vi udviklet et Visionsystem. Metoderne vi har udvalgt til systemet, er dem som har givet os de bedste resultater i den eksplorative fase af projektet. Systemet er udviklet i Python, og der gøres brug af OpenCV og numpy biblioteket og indeholder de nødvendige metoder.

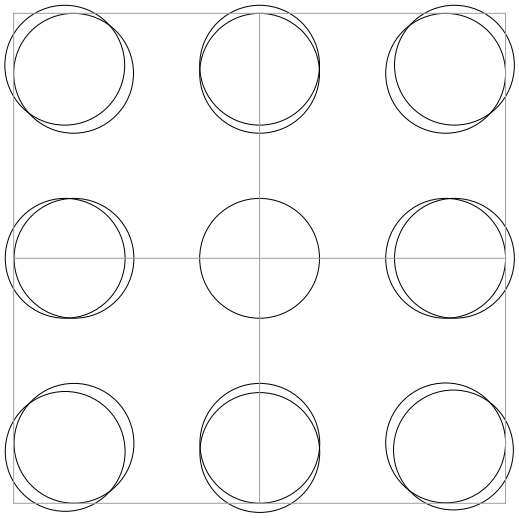
Visionsystemet indeholder to forskellige metoder: en metode til at finde brøndplader og en til at finde petriskåle. Hver metode er optimeret til hvert deres datasæt. Disse dele af systemet beskrives i kapitel 4, implementering.

3.2 Fremstilling af datasættene

Vi har udformet tre forskellige datasæt, der danner grundlag for testningen af Visionsystemet. Disse datasæt er forskellige i kombinationen af placering af objekter og belysning. Vigtigheden af belysningen i et Computer Vision system vurderes til at vægte $\frac{2}{3}$, mens selve systemdesignet og softwaren kun vægtes $\frac{1}{3}$ (Paulsen and Moeslund, 2012, p. 10). Inden vi har igangsat forsøget, har vi derfor fokuseret på hvordan EvoBotten skal skærmes/belyses. Billederne er af petriskåle eller brøndplader, placeret på den eksperimentelle overflade, set fra neden.

3.2.1 Objekters placering

Det første parameter i datasættene er placering af objekterne. Vi kigger på placeringen for at undersøge om den relative vinkel til kameraet kan påvirke genkendelsen. For hver placering vi vælger at teste, tilføjer vi én grad af kompleksitet til vores datasæt. Det er derfor en god idé at udvælge de placeringer, som giver mest mulig information. En oplagt opstilling kunne være en 3×3 matrix fordelt på hele den eksperimentelle overflade, da den ville give os information fra forskellige vinkelekstremaer. Opstillingen ses på figur 3.1. Denne opstillingen kan give os denne information, men vil forøge kompleksiteten af

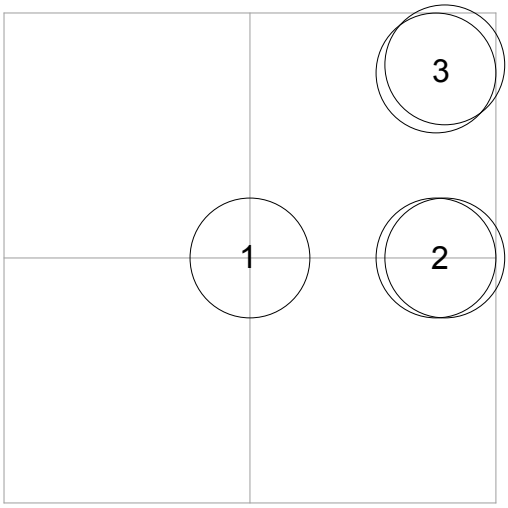


Figur 3.1: $3x3$ opstilling.

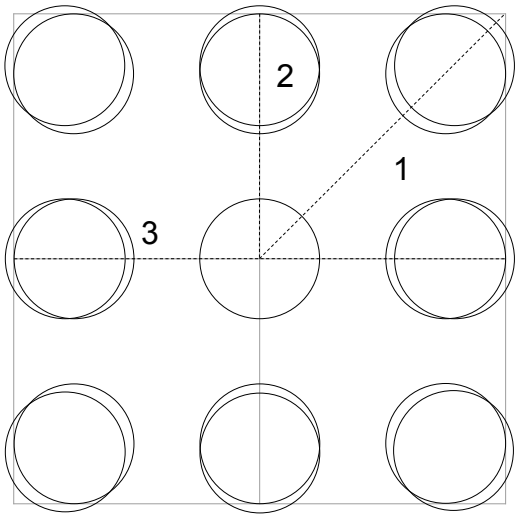
datasættet med ni. Ved at benytte spejling kan vi få samme information ved at se på blot tre placeringer, som det kan ses på 3.2. Selvom objekterne er illustreret som runde, er det disse placeringer vi anvender for både petriskåle og brøndplader. Spejlingerne i figur 3.3 viser hvorledes, placering 2 og 3 kan bruges til at sige noget om de ønskede placeringer.

Det skyldes, at det for algoritmen er den relative vinkel og ikke vinklens orientering, der forårsager en ændring i objektets udseende. Det er derfor tilstrækkeligt at inkludere disse tre placeringer i datasættet.

Eftersom petriskålene er runde, er deres egenrotation hverken synlig eller relevant. Brøndplader er derimod rektangulære. Deres egenrotation kan muligvis påvirke Vision-systemets resultater. Dette testes ved at have billeder af brøndplader med tre forskellige rotationer for hver placering i vores datasæt.



Figur 3.2: Endelig forsøgsopstilling.



Figur 3.3: Ovenstående figur viser hvordan den valgte forsøgsopstilling kan give samme information som den første ved brug af spejling.

I figur 3.2 ses de placeringer vi anvender i datasættene. I analysen vil placering 1 blive kaldt midten, placering 2 vil blive kaldt siden og placering 3 vil blive kaldt hjørnet. Dette gælder både for petriskåle og brøndplader. Herudover vil brøndpladerne også have en rotation som anskueliggøres i figur 3.4. I analysen vil rotation 1 refereres som 90 grader, rotation 2 som 0 grader og rotation 3 som 45 grader.

Figur 3.4: Ovenstående figur viser de tre vinkler hvoraf vi tager billeder af brøndpladen for hver placering.

3.2.2 Belysning

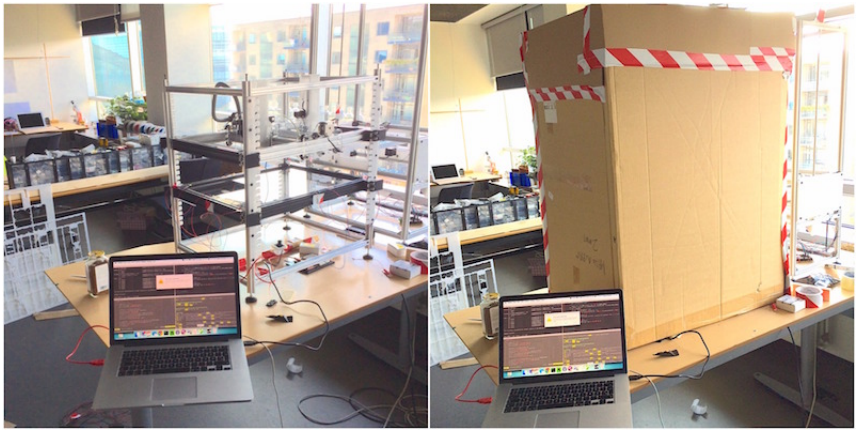
Kontrol af lysmiljøet er det andet parameter i vores datasæt. At teste lysegenskaber systematisk er en kompliceret opgave. Grunden til dette er, at der er utallige variabler, der påvirker lysmiljøet. Dette uddybes under fejlkilder.

Vi valgt at anvende to lysmiljøer i vores datasæt. I det første anvender vi et almindeligt webkamera, som opfanger lys fra det visuelle spektrum. Her har vi en forsimplet model for kontrol af lyset og tager kun højde for billedets gennemsnitlige lysniveau.

I det andet lysmiljø anvender vi et infrarødt filter i et kamera, som opfanger lysbølger ude for det visuelle spektrum. I dette lysmiljø oplyser vi den eksperimentelle overflade med infrarødt lys.

Det Visuelle Spektrum

Ved at konvertere et billede til HSL (hue, saturation, Luminance) farverummet, får man en kanal, der siger noget om lysniveauet (Luminance) på billedet (*HSL Colorspace*). Luminance er det lysniveau, som et menneske opfatter i et billede (Paulsen and Moeslund, 2012, p. 119). Begrebet vil blive refereret til som lysniveau fremadrettet. Hvis man tager gennemsnittet af alle intensiteterne for denne kanal, kan man få et groft udtryk for hvor kraftig lysniveauet er i billedet. Denne metode kan dog give nogle skævvridninger, hvis der f.eks. er en meget kraftig lyskilde, der kun dækker lidt af billedet. Dog finder vi metoden anvendelig i forhold til vores forsøg, da den giver os en konkret målestok i forhold til belysningen i billederne, vi tager i betragtning. Vi kan dermed anvende metoden til at beskrive det relative lysniveau på billederne i vores datasæt.



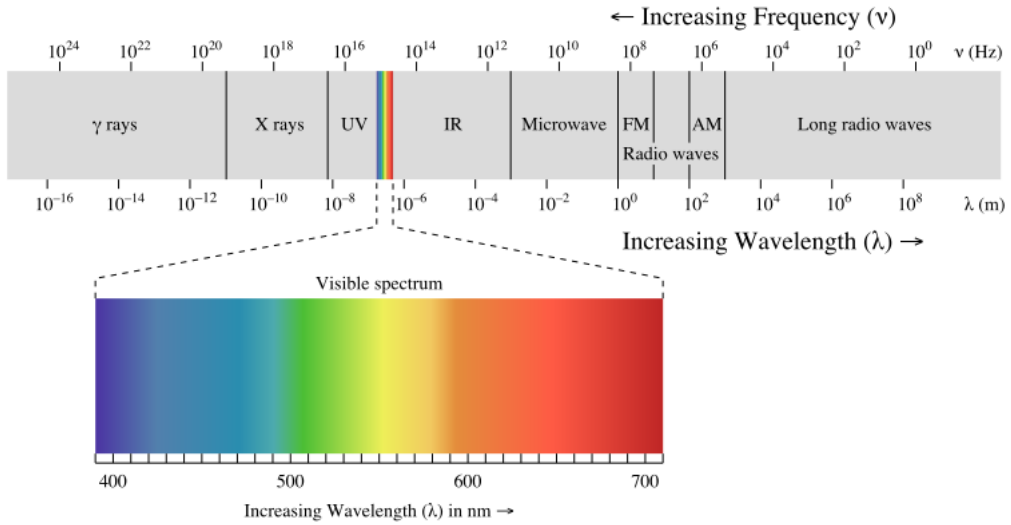
FIGUR 3.5: Ovenstående billede viser EvoBotten med og uden den lystætte kasse, som vi har bygget af pap og tape.

Vi styrer lysniveauet ved brug af en lystæt kasse, som helt eller delvist overdækker EvoBotten. Ved at justere afstanden fra bordet til kassens kant, kan vi justere mængden af lys, der bliver lukket ind. Dette kan selvfølgelig ændre sig alt efter, hvor meget lys der er i lokalet. Vi korrigerer for dette ved hele tiden at holde øje med den gennemsnitlige værdi for lysniveauet på skærmen, mens vi tager billederne.

Infrarød Belysning

Som nævnt bruger vi et infrarødt filter i kameraet og oplyser den eksperimentelle overflade med infrarødt lys I det andet lysmiljø. Bølgelængden som filtret opfanger, er på ca. 840 nanometer (nm) til 860 nm. Da det kun er termiske lysbølger, der kan ses i dette spektrum, forstyrrer udefrakommende lyskilder ikke i nær samme grad. Vi anvender denne metode til at fjerne støj i billedet, så vi med større præcision kan adskille objekter fra baggrunden på billederne i vores datasæt.

Som det kan ses i figur 3.6, ligger det infrarøde spektrum lige over det visuelle spektrum. De LED'er vi har anvendt, har en bølgelængde på $850nm$, deres strømstyrke er på $50mA$, spændingen er på 1,5 Volt og de lyser med en 60 graders vinkel (*Buy IR-LED 850 nm 3 mm (T1), Everlight Electronics, HIR 204/H0*).



Figur 3.6: Her ses en illustration af farvespektrummet.
(*Light: Electromagnetic waves, the electromagnetic spectrum and photons*)

3.2.3 Opstilling af forsøget

Vi har tre forskellige forsøgsopstillinger: I den første anvender det første lysmiljø, hvor vi ser på hele det visuelle spektrum. I den anden og tredje opstilling anvender det andet lysmiljø med infrarødt lys.

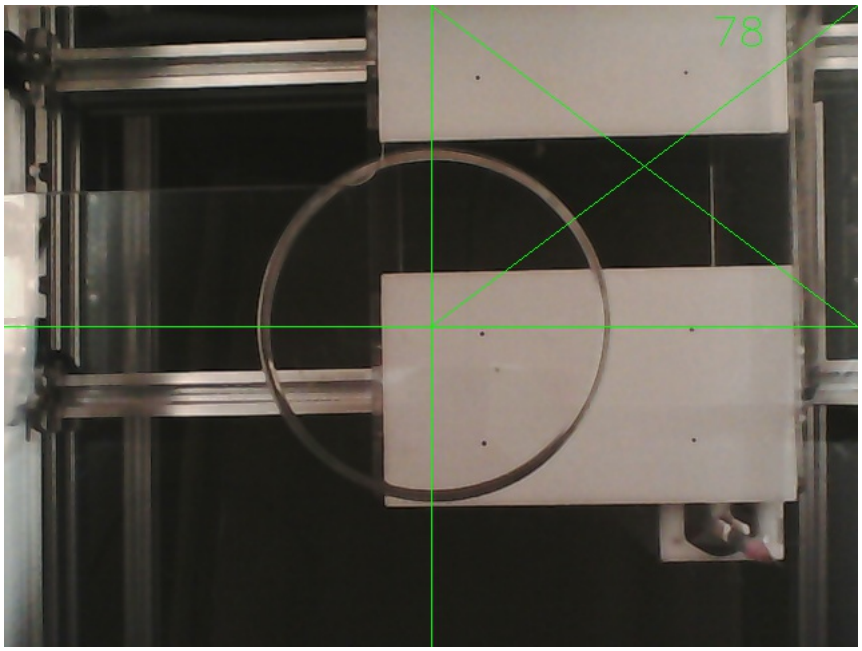
Den første opstilling kalder vi "Datasæt 1 - Det Visuelle Spektrum, Linux". I denne opstilling har vi brugt et kamera af modellen HP WebCam HD 2300. Der er ikke anvendt et infrarødt filter og vi har afgrænset os til to dimensioner for petriskålene og tre for brøndplader. For petriskålene er dimensionerne den gennemsnitlige lysintensitet på billedet og placering af petriskålen. For brøndpladerne er det den gennemsnitlige lysintensitet, placering og rotation af objektet. Billederne er taget på en Linux-pc (Ubuntu 16.04).

For begge objekter er lysindstillingerne følgende:

1. 18 Luminance
2. 48 Luminance
3. 78 Luminance
4. 108 Luminance

5. 138 Luminance

Skalaen starter på 18, da det er det mørkeste lysmiljø, vi har været i stand til at simulere. Til datasættet har vi taget tre billeder for hvert lysniveau, altså et for hver placering af objektet. For at sikre konsistente placeringer i vores datasæt har vi brugt målestreger i scriptet, der bruges til at tage billederne. Dette illustreres i figur3.7.



Figur 3.7: Her vises et eksempel på hvordan vi kunne placere objekterne med den rigtige placering og rotation.

Den anden opstilling kalder vi "Datasæt 2 - Infrarød Belysning, Linux". Her har vi brugt et kamera af modellen ELP-USBFHD01M-L36 med infrarødt filter. I denne opstilling ser vi ikke på det gennemsnitlige lysniveau. Vi bruger de samme objektplaceringer som i datasæt 1 samt en billedserie uden objekter på den eksperimentelle overflade. Vi har taget ti billeder for hver placering og for hver rotation. Billederne er taget på en Linux-pc (Ubuntu 16.04).

Den tredje og sidste opstilling kalder vi "Datasæt 3 - Infrarød Belysning, Mac". Der bruges samme kamera som i datasæt 2, tilsluttet en Mac (OS X - El Capitan 10.11.4). Vi har ikke medtaget billeder uden objekter i dette datasæt. Vi har yderligere gjort brug af den lystætte kasse. Der er for hver placering og rotation taget 10 billeder med den lystætte kasse og 10 billeder uden - dog med et låg på toppen af EvoBotten for at skærme for infrarøde strålinger fra loftslys.

3.3 Ground Truth

I Computer Vision er Ground Truth det begreb, der dækker over en grundlæggende tilgang til metrisk analyse af billeder. Ground Truthing er i generelle termer en måde at måle vores Computer Vision systems resultater i forhold til et datasæt, der er relativt bedre. Her kan relativt bedre være f.eks. i forhold til præcision og nøjagtighed, som er de parametre, vi ønsker at teste vores data i forhold til. Ground Truthing falder ind under følgende kategorier:

- Syntetisk produceret: Billeder, der bliver genereret fra computermodeller.
- Reelt produceret: Videoer eller billedsekvenser, der er taget med det formål at det skal være truth data.
- Reelt valgte billeder, fra allerede eksisterende kilder.
- Automatiseret annotering: Machine Learning metoder anvendes til at udtrække elementerne fra dataen.
- Manuel annotering: Her definerer vi hvor i billedet de elementer, der ønskes fundet, befinder sig.
- Kombination af disse.

(Krig, 2014, p. 285).

Ground Truth data et sæt af billeder med markeringer, der beskriver de egenskaber, man ønsker at måle et system imod. Disse egenskaber er i vores tilfælde objekter på den eksperimentelle overflade og deres placering. Markeringerne kan være automatisk genereret, men da vi forsøger at analysere vores automatiske markeringer, ville det være absurd. Da størrelsen på datasættene tillader det, har vi derfor valgt selv at lave markeringerne (Krig, 2014, p. 284).

3.4 Evaluering af datasættene

Som et led i test og evaluering bruger vi Ground Truth data. Det overordnede begreb vi bruger når vi vurderer vores tests er robusthed. Her redegør vi for begrebet.

3.4.1 Robusthed

Som diskuteret i (Krig, 2014, p. 194), skal robusthed afgrænses til et eller flere kriterier. Her ser vi på robustheden som en fællesbetegnelse for præcision og nøjagtighed, som vi definerer nedenfor. Følgende forhold kan have påvirkning på systemets robusthed:

- Belysning: Som nævnt af Krig (Krig, 2014, p. 196) er belysning et meget højt prioriteret område for billedanalyse. Ligeledes nævnes det af Paulsen (Paulsen and Moeslund, 2012, p. 10), at belysningen er enormt vigtig, hvilket gør robusthed i forhold til belysning til et godt udgangspunkt.
- Farver: Hvis man anvender farver i sin billedanalyse, er det relevant at have en særlig nøjagtighed i forhold til hvilke farver, der analyseres og hvilke farverum, der anvendes.
- Ufuldstændighed: Egenskaber kan ikke forventes altid at være ens for hver frame i en strøm af billeder. Herunder kan det f.eks. være at kameraet eller objektet har bevæget sig.
- Afstand: Afstanden fra objektet til kameraet kan have betydning for genkendelse af objektet. På EvoBotten er det f.eks. muligt at ændre på højden af den eksperimentelle overflade.
- Forvrængning af former: Når objekter bliver manipuleret på en måde som ændrer deres udseende. Robusthed i forhold til forvrængning er også en en prioritet, da vi ser det som en del af præcisionen af systemet.

I EvoBotten er det ikke alle faktorerne, der giver mening at sikre robusthed overfor. F.eks. er noget af det første vi gør, i forarbejdelsen af billederne, at konvertere vores billeder til gråtoneskala. Det er derfor ikke nødvendigt at inddrage farver i vores analyse. Afstanden fra kameraet til den eksperimentelle overflade kan på EvoBotten varieres for at gøre plads til større objekter. At variere denne højde kan potentielt set påvirke Visionsystemets evne til at detektere objekter. Dette har vi valgt at afgrænse os fra og vi tester kun med én indstilling for højden. Det er dog værd at nævne, at der kan forekomme ufuldstændige billeder i form af støj fra f.eks. spildte væsker eller andet på billedet. Hertil tester vi heller ikke robustheden i forhold til objekter, der ikke kan ses 100 % af kameraet. Dette er som sagt en afgrænsning vi har valgt, som eventuelt kan tages op til overvejelse i fremtidige iterationer.

3.4.2 Nøjagtighed & Præcision

Nøjagtighed og præcision er to begreber vi anvender som målestok for systemets robusthed. Nøjagtighed beskriver i hvor høj grad systemet finder de objekter, der er på billederne. For hver funden position for et objektet måles præcision ud fra afstanden til den reelle position, som er annoteret i Ground Truth dataen. I en bredere kontekst er både nøjagtighed og præcision vigtige elementer, når vi skal teste Visionsystemet. En fejl kan betyde at kemikalier bliver pipetteret ud på glaspladen eller ned i en forkert brønd, hvilket kan ødelægge forsøget.

De egenskaber der tages i betragtning, når vi måler præcisionen for petriskåle, er dets centrum og radius. For brøndplader finder vi de fire punkter, der udgør de fire hjørner i rektanglet.

3.4.3 Ydeevne

Ydeevne er i mange tilfælde et vigtigt element. Især hvis det kræves af systemet at man kan få hurtige opdateringer, når der sker ændringer i miljøet. Ydeevne er også et parameter, som er vigtigt for EvoBotten. Svar fra Visionsystemet må kunne afkræves indenfor et rimeligt tidsrum. Dog er der nogen fleksibilitet - vi kan godt leve med en svartid på ét sekund, hvis det forøger robustheden. Ydeevne er dog ikke en faktor, vi tester systemet for. Men vi har noteret, at den holder sig inde for rimelige svartider.

3.5 Fejlkilder

Når man har at gøre med Computer Vision eller andre forskningsfelter, hvor det kan være svært at kontrollere alle variabler, er det i særlig grad vigtigt at være sig bevidst om de faktorer, der kan påvirke de endelige resultater. Her beskriver vi potentielle fejlkilder, som kan have påvirkning på resultaterne af vores eksperimenter.

3.5.1 Metodiske fejlkilder

Lysmiljøet på et kontor er udpræget kaotisk set fra et Computer Vision-perspektiv. Lysmiljøet kan have en enorm påvirkning på det endelige resultat. Der er mange parametre, som har stor potentiel påvirkning på genstandsfeltet: lofts- og skrivebordslampers

placering ift. genstandsfeltet, deres strålingsretning, fluktuation, antal Watt og andre beskaffenheder, antal vinduer i rummet, deres størrelse, deres placering ift. solen, rummets generelle beskaffenhed og tidspunkt på dagen. Kun ved at lukke alt lys ude og bruge egne lamper, kan det lade sig gøre at kontrollere lysmiljøet i det visuelle spektrum. Men selv her kan man kun gøre sig antagelser om, hvad der virker bedst; direkte lys eller diffust lys, lysintensitet, antal lyskilder eller lyskildernes beskaffenhed. Ingen af disse parametre kan generaliseres. De skal derfor testes i det konkrete brugs-scenarie, for at de kan vurderes bedre eller dårligere end deres alternativer. Alternativer, hvilke der er uendeligt mange af. I stedet for at forsøge at kontrollere kaosset, har vi derfor valgt at lave en simplificeret model af lysmiljøet, som beskrevet i afsnit 3.2.3. Denne model er meget simplificeret, og kan derfor være kilde til fejlagtige observationer. Tydeligst af alle er, at vi ikke er i stand til at fratrække baggrunden fra de billeder, vi analyserer pga. manglende konsistens mellem baggrundsbilledet og forgrundsbilledet. Det kan forklare, hvorfor vi ikke kan detektere brøndplader i disse billeder, og er en af hovedårsagerne til at vi vælger at eksperimentere med infrarødt lys. At anvende det infrarøde filter medfører dog nogle andre problemer i forhold til Evobottens generelle brugsscenarier. Man er nødt til enten at have to kameraer monteret eller at finde en måde, hvorpå man kan slå det infrarøde filter til eller fra, da observation af forsøg ikke er muligt med et infrarødt filter. Dette kan muligvis gøres ved at justere på eksponeringen, hvilket dog synes kun at være muligt fra en Windows PC.

3.5.2 Svagheder ved vores system

Metoden som vi bruger til at finde afvigelsen mellem vores Ground Truth data og de algoritmiske annoteringer af brøndplader, har den svaghed, at kun små afvigelser kan håndteres. Den fungerer ved, for hvert hjørne på den ene annotering, at finde det hjørne på den anden, for hvilken afstanden er kortest. Hvis de to annoteringer ikke overlapper min 50 %, vil de forkerte hjørner blive sammenkoblet, hvilket så vil give os en forkert måling. Vi sikrer os mod denne fejlkilde ved at gennemgå alle billederne manuelt og checke, at deres præcision er høj nok til ikke at få koblinger til de forkerte hjørner.

3.5.3 Menneskelige Faktorer

Da vi har lavet vores Ground Truth med manuelle annoteringer, er der en risiko for at vi har været upræcise. Især på billeder hvor det kan være svært at se objektets konturer med det blotte øje, er der en risiko for, at vi har lavet fejlagtige annoteringer. Det kan

muligvis have påvirkning på vores præcisionsmålinger. Det er desværre både svært at korrigere for og at opdage.

En direkte fejlkilde vi har opdaget er, at der er en forskel på kameraets synsfelt i datasæt 2 og datasæt 3. Det betyder at billedet bliver forskudt, så det ikke er den samme del af den eksperimentelle overflade, der er i fokus i datasættene.

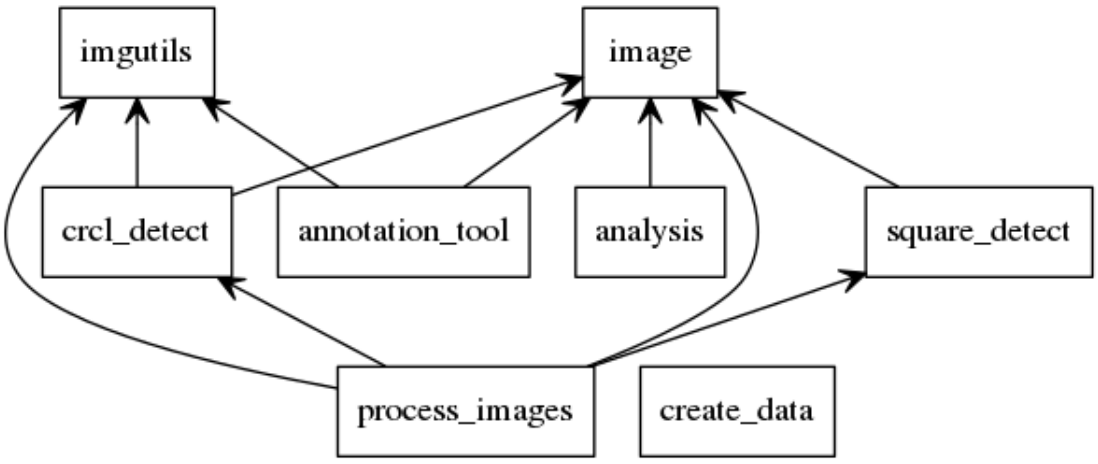
3.5.4 Teori og praksis

I vores datasæt har vi aldrig mere end ét objekt på hvert billede. Af konkrete brugsscenarier vil det være svært at finde et, hvor der kun er behov for at spore et enkelt objekt. Forsøg med væskehåndtering inkluderer nemlig oftest overførsel fra ét objekt til et andet. Ud fra vores datasæt kan vi ikke sige noget om robustheden, hvis der bliver introduceret flere objekter på ét billede. Desuden vil systemet skulle udbygges, hvis det skal være muligt at spore objekter, der bevæger sig samtidigt. Som det er nu, har vi kun en placering og en klassificering til at identificere de enkelte objekter. Hvis f.eks. to petriskåle, i et billede, bevæges samtidigt, vil det på nuværende tidspunkt ikke være muligt at skelne dem fra hinanden. For at kunne det kræves det at hvert objekt får en unik identifikator af en art.

Kapitel 4

Implementering

I dette kapitel vil vi præsentere de vigtigste dele af det Computer Vision-system, vi har udviklet. Vi har valgt at inkludere dette kapitel for at tydeliggøre den overordnede systemarkitektur, samt årsager bag brug af forskellige metoder fra OpenCV-biblioteket, som ellers kan være svære at gennemskue. Vores software består af implementering af de Computer Vision-metoder som vi bruger samt et annoteringsværktøj og sammenligningsværktøj. Alle scripts, som er udviklet i forbindelse med specialet, er skrevet i Python. OpenCV har interfaces til mange sprog herunder Python, Java og C++ (*Introduction to OpenCV-Python Tutorials — OpenCV 3.0.0-dev documentation*). Valget om at bruge Python til udviklingen af systemet var åbenlyst for os, da det generelle framework til EvoBotten er skrevet i Python.



FIGUR 4.1: Figuren viser et diagram over modulerne i Visionsystemet og deres indbyrdes kald.

4.1 Klasser

Vores software er ikke opbygget efter traditionel objektorienteret arkitektur. Dog har vi anvendt klasser til at konstruere vores egne datatyper, som vi anvender når vi laver 'Ground Truthing'. Klasserne er Image, Rectangle og Circle.

4.1.1 Image

Image-klassen er en type, der indeholder informationer om det billede, der tages i betragtning. Et Image-objekt har tre felter: *filename*, *avg_lightness* og *annotations*. Filename er navnet på billedfilen, *avg_lightness* er den lysintensitet, vi har målt på billedet og annotations er en liste, der indeholder Rectangle- og Circle-objekter, som repræsenterer de genstande, der er blevet fundet med genkendelse eller annotereret med annoteringsværktøjet.

Rectangle-klassen indeholder informationer om rektanglets hjørner, hvilket er angivet som punkter. Klassen indeholder således et felt, der er en liste med fire punkter.

Circle klassen har to felter, et for center og et for radius.

Foruden disse felter, har hvert annoteringsobjekt et felt, der beskriver hvad de er (rektangel eller circel). Grunden til dette er at, når alle objekterne er blevet instantieret, bliver der gemt to JSON-filer. En fil, der indeholder informationerne fra genkendelse fra Visionsystemet og en fil, der indeholder informationerne fra annoteringsværktøjet.

4.2 Computer Vision Systemet

4.2.1 Processeringsværktøj

process_images.py er 'point of access' til vores detektionssystem. Det er med andre ord dette script vi kalder, når vi skal lave automatiske annoteringer. Scriptet sørger for at kalde relevante metoder til både at udføre og gemme detektionerne. Scriptet tager en fil-sti som argument og bearbejder alle jpg-filer, der ligger i stien. Det kører circle_detect og sqr_detect, som finder petriskåle eller brøndplader, hvis der er nogen i billedet, påtegner dem og returnerer hhv. et Circle-objekt eller et Rectangle-objekt. For hvert billede vil der blive tilføjet et Image-objekt med de fundne objekter til en liste.

Når scriptet har itereret igennem alle billederne, vil listen af Image-objekter blive konverteret til JSON-format og gemt som *data.json*. Image-, Circle- og Rectangle-klasserne vil blive gennemgået i afsnittet 4.1. Først ser vi overordnet på hvordan circle_detect og sqr_detect fungerer.

4.2.2 Petriskåle

At detektere petriskåle kræver ikke meget kode, da vi kan bruge OpenCV's egen implementering af Hough Circle Transform. Den har vist sig at være meget robust selv med en minimal forarbejdelse af billederne. Det eneste vi har været nødt til, er at konvertere billedet til gråtoneskala, hvorefter der udføres Median sløring med et vindue på 5×5 pixler(px) for at mindske støj i billedet. Herefter foretager vi selve cirkeldetektionen med følgende parametre til *cv2.HoughCircles()*:

1. Billedet der skal bearbejdes.
2. Hough Gradient bearbejder først billedet med Canny og finder dernæst kantens gradient for at finde kandidater til cirklens centrum, som beskrevet i Kapitel 2 i afsnittet om Hough Circle Transform.
3. Her specificeres den faktor, som akkumulatorbilledet bliver forringet med i opløsning , i forhold til inputbilledet. I vores script er dette parameter sat til to. Akkumulatorbilledet, som er det billede hvor der stemmes om centre og radier, får således en opløsning, som er halvt så høj som på det originale billede.
4. Dette parameter specificerer den afstand, der mindst skal være cirklernes centre imellem.
5. De to sidste parametre, minRadius og maxRadius specificerer de intervaller, målt i pixels, hvori metoden vil lede efter cirkler.

Metoden returnerer koordinater for centrum af cirklen og en radius, som kan bruges til at annotere cirklen og returnere et Circle-objekt.

4.2.3 Brøndplader

At detektere brøndpladerne er en betragtelig større udfordring end at finde petriskålene. Det er dog lykkedes, ved en kombination af belysning, bearbejdelse og repræsentation, at få et godt resultat. Til forarbejdelsen anvender vi to billeder: et billede med objektet,

```python
import cv2
import numpy as np
import image as im
import imgutils as imu

def find_circles(img):
    cimg = img
    img = cv2.cvtColor(img.copy(),cv2.COLOR_BGR2GRAY)
    img = imu.image_blur(img, blur="median", window=(5,5))
    detections = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 2, 200, minRadius=120, maxRadius=140)
    if detections is None:
        return cimg, []
    else:
        detections = np.uint16(np.around(detections))
        circles = []
        for circle in detections[0,:]:
            # draw the circle on the image
            cv2.circle(cimg, (circle[0], circle[1]), circle[2], (0,255,0), 2)
            # draw the center of the circle
            cv2.circle(cimg, (circle[0], circle[1]), 2, (0,0,255),3)
            x, y, r = int(circle[0]), int(circle[1]), int(circle[2])
            # make the circle_object and add it to circles
            circle_object = im.Circle((x,y), r)
            circles.append(circle_object)
        return cimg, circles
```

FIGUR 4.2: På billedet ses scriptet bruges til at finde cirkler i billeder.

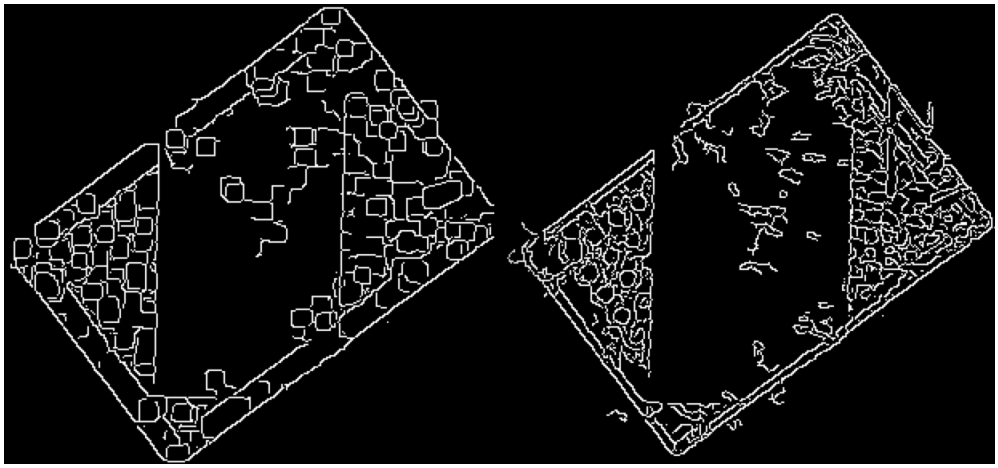
der ønskes fundet og et billede af baggrunden uden objekt. Begge billederne bliver konverteret til gråtoneskala, får fordoblet kontrasten, hvorefter baggrunden bliver trukket fra. Hvis der ikke er støj i billedet, fremstår objektet med sort baggrund. Herefter tredobler vi kontrasten i billedet. Da vi arbejder med relativt mørke billeder, er det

```python
def find_squares(img, bg):
    cimg = img
    rectangles = []
    img, bg = img*2, bg*2
    img = cv2.cvtColor(img.copy(),cv2.COLOR_BGR2GRAY)
    subtracted = cv2.subtract(img, bg)
    enhanced_contrast = subtracted*3
    eroded = cv2.erode(enhanced_contrast, np.ones((2, 2)))
    blurred = cv2.bilateralFilter(eroded,5,21,21)
    dilated = cv2.dilate(blurred, np.ones((9, 9)))
    canny = cv2.Canny(dilated, 16, 39)
    dilated = cv2.dilate(canny, np.ones((5, 5)))
```

FIGUR 4.3: På billedet ses den første del af sqr_detect scriptet.

nødvendigt at arbejde med kontrasten for at få brøndpladen til at fremtræde i billedet, selvom det også medfører en risiko for at tilføje støj. For at modvirke støjen, bruger vi herefter *cv2.erode()*, som skrumper strukturer i billedet. Det meste støj er på nuværende tidspunkt så minimalt, at det med denne metode forsvinder helt fra billedet. Herefter bruger vi bilateralt slørring til yderligere at begrænse støjen men bevare kanterne. Når dette er gjort, bruger vi *cv2.dilate()*, som får strukturerne i billedet til at

vokse igen. Dette gør vi både før og efter vi bruger Canny, da det får kanterne langs brøndpladen, som oftest bliver brudt, til at 'vokse sammen' igen. Dette ses i figur4.4. Dette er et kritisk skridt i bearbejdningen, som muliggør at *cv2.findContours()*, kan



FIGUR 4.4: På billedet ses processeringen før og efter dilation.

finde brøndpladens sande konturer. *cv2.findContours()* tager følgende parametre:

```
26      _, contours, _ = cv2.findContours(dilated, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    #second argument is mode, third is method
27      for cnt in contours:
28          cnt_len = cv2.arcLength(cnt, True) #second argument specifies whether curve is closed or
    not.
29          cnt = cv2.approxPolyDP(cnt, 0.07*cnt_len, True) #second argument specifies the epsilon
    and third argument whether the curve is closed or not
30          if len(cnt) == 4 and cv2.contourArea(cnt) > 25000 and cv2.contourArea(cnt) < 50000 and
    cv2.isContourConvex(cnt):
31              cnt = cnt.reshape(4,2)
32              max_cos = np.max([angle_cos(cnt[i], cnt[(i+1) % 4], cnt[(i+2) % 4]) for i in
    xrange(4)])
33              if max_cos < 0.1:
34                  rectangle_object = im.Rectangle(np.array(cnt).tolist())
35                  rectangles.append(rectangle_object)
36                  cv2.drawContours(cimg, [cnt], -1, (0, 255, 0), 2)
37      return cimg, rectangles
```

FIGUR 4.5: Her ses findcontoursdelen af sqr_detect.

1. Billedet, der skal behandles.

2. Mode: Dette parameter specificerer hvilke konturer, der skal returneres og med hvilken datastruktur. *cv2.RETR_EXTERNAL*, som vi anvender, returnerer kun det yderste lag af konturerne, hvor alle andre modes også returnerer indre konturer, enten som lister (RETR_list) eller som træer (f.eks. RETR_TREE). Så længe vi kan regne med at behandlingen fjerner baggrunden, er vi kun interesseret i den ydre kontur, som omslutter brøndpladen.

```
5 def angle_cos(p0, p1, p2):
6     d1, d2 = (p0-p1).astype('float'), (p2-p1).astype('float')
7     return abs(np.dot(d1,d2) / np.sqrt(np.dot(d1,d1)*np.dot(d2, d2)))
```

FIGUR 4.6: Her ses angle_cos metoden fra sqr_detect.

3. Method: De to mest almindeligt anvendte metoder er *cv2.CHAIN_APPROX_NONE* eller *cv2.CHAIN_APPROX_SIMPLE*. Den første metode returnerer en liste af samtlige punkter langs konturen, og den anden returnerer kun punkterne for hjørnerne i konturerne. Den sidste metode bruger mindst hukommelse, og giver os samtidig information omkring strukturen af det fundne objekt, som vi bruger i den næste del af scriptet.

Herefter itereres der over de fundne konturer. Først bliver konturens længde udregnet med *cv2.arcLength()*, som tager listen af kanter som parameter og en boolean der afgører om den skal lave udregningen for længden af en kurve eller betragte punkterne som kanter i en lukket kontur, hvor den medregner afstanden fra første til sidste punkt. Da et rektangel har en lukket kontur, sætter vi dette parameter til True. Herefter forsimples den fundne kontur med *cv2.approxPolyDP()*. Denne metode tager følgende parametre:

1. En liste af kanter.
2. Epsilon. De punkter som har en længde fra de etablerede punkter, som ligger over denne tærskel, vil blive taget i betragtning i det nye polygon. Vi har sat epsilon til 7% af den samlede længde for konturen.
3. Bool closed: Som med *cv2.arcLenght()* specificerer man om det er en lukket kontur eller en kurve.

På nuværende tidspunkt er bearbejdningen af konturen færdig og detektionen går i gang. Det checkes om der er 4 kanter i konturen, om arealet for konturen er imellem 25000px og 50000px, om konturen er konveks og dermed ikke har indre vinkler på over 180 grader. Hvis konturen opfylder disse kriterier, vil den blive taget i betragtning som en firkant. Men inden det endeligt kan afgøres, bliver listen omformet fra en liste i 3 dimensioner til en i 2. Listen indeholder ikke færre data, men bliver nemmere at arbejde i *angle_cos()*, hvor cosinus til de indre vinkler bliver udregnet ved hjælp af en hurtig omregning af et teorem for udregning dot-produktet for to vektorer:

$$A \cdot B = ||A||B|| \cos(\theta) \implies \cos(\theta) = \frac{A \cdot B}{||A||B||}$$

Den største værdi bliver så gemt i variablen *max_cos*. Hvis *max_cos* er under 0.1 svarende til at det største vinkeludsving er ca.±5% ($\cos^{-1}(0.1) \approx 84° \wedge \cos^{-1}(-0.1) \approx 95°$), bliver brøndpladen annoteret på billedet og returneres sammen med et Rectangle-objekt med de fire punkter, der udgør brøndpladens hjørner.

4.3 Annoteringsværktøjet

annotation_tool.py er 'point of access' for den manuelle annotering, altså scriptet som vi har brugt til at lave vores Ground Truth Data. Det tager en fil-sti som argument og vil så loade alle jpg-filer som er i den sti, så brugeren kan annotere om der er en petriskål eller et brøndplade i billedet. Annoteringerne bliver herefter gemt som Rectangle- eller Circle-objekter i et Image-objekt og tilføjet til en liste af objekter for hver jpg-fil. Når alle billederne er løbet igennem, konverteres listen til JSON-format og gemmes som *annotation_data.json*.

4.4 JSON Data

Både Visionsystemet og annoteringsværktøjet gemmer hver deres JSON-fil, der indeholder informationer om billederne og de respektive annoteringer. Den ene JSON-fil repræsenterer vores Ground Truth data og kommer fra annoteringsværktøjet. Den anden JSON-fil repræsenterer vores detektionsdata og kommer fra Visionsystemet.

Hver fil indeholder et JSON-array, bestående af JSON-image-objekter. Et eksempel, der viser strukturen er således:

```
{
  "image": {
    "annotations":
    [
      {
        "shape": "Rectangle"
        "box": [[290,348],[290,121],[635,121],[635,348]],
      },
      {
        "shape": "Circle",
        "radius": 134,
```

```
            "center": [504,130]
          }
      ],
      "avg_lightness": 108,
      "filename": "44.jpg"
    }
}
```

Dette objekt indeholder informationer om billedet, der har navnet 44.jpg. Billedet har en *avg_lightness* på 108. Det indeholder et array kaldet annotations, der indeholder to annoteringsobjekter: et Rectangle-objekt og et Circle-objekt. Rectangle-objektet har shape "Rectangle" og et array med koordinater til rektanglets fire hjørner. Circle-objektet har foruden dens form også en radius på 134 og en centerattribut med koordinater til cirklens centrum. Det skal nævnes, at dette JSON-objekt er et tænkt eksempel, der ikke findes i vores data.

Vi har valgt JSON til at gemme informationerne, da vi gerne ville have en strengbaseret lagringsmetode, der er læselig for mennesker. Python har en nem serialiseringsproces, hvis man repræsenterer data, som dictionaries, hvilket vi har gjort. På den måde har vi ikke selv skulle stå for parsing og serialisering. Vi kunne også have gemt informationerne som binær data som nemt kunne serialiseres eller 'pickles', men så ville dataen ikke længere være læselig for et menneske. Herudover kunne vi have gemt dataen i en SQL database, hvilket ville give noget mere struktur, men samtidigt gøre applikationen langt mere tung. Ved at gemme dataen som JSON, er det muligt at konvertere til en noSQL database som mongoDB.

4.5 Sammenligningsværktøjet

Sammenligningsværktøjet har en klassestruktur, der minder om den vi bruger for Image-, Rectangle- og Circle-klasserne. I stedet for at gemme informationer om hver annotering, indeholder sammenligningsværktøjets klasser informationer om afvigelserne på annoteringer ud fra to ens billeder med hver deres annoteringer.

Informationerne som værktøjet bearbejder kommer fra de føromtalte JSON-filer. Værktøjet indeholder fire klasser: Report, image_obj_data, circle_data og rectangle_data.

Et rectangle_data-objekt indeholder informationer om afvigelser for hvert hjørne i et rektangel. Ligeledes gør et circle_data-objekt for afvigelsen på radius og centerpunktet for en cirkel.

Et image_obj_data-objekt indeholder en liste med afvigelser for hver annotering ud fra to Image objekter. Herudover indeholder objektet informationer om billedet på samme måde som et Image objekt, altså gennemsnitlig lysintensitet og filnavn.

Report objektet er samlingen af alle informationerne, som bliver genereret ud fra de to datasæt. Et Report objekt har seks felter: *annotations*, *detected*, *annotations_dev_list*, *img_obj_dev_list*, *img_drawed_list* og *error_list*.

Feltet *annotations* indeholder et tal som angiver det antal annoteringer som findes i Ground Truth datasættet. Feltet *detected* er det antal annoteringer som Visionsystemet har fundet. Feltet *annotation_dev_list* er en liste med alle rectangle_data- og circle_data-objekterne. Ligeledes er *img_obj_dev_list* en liste, der indeholder hvert img_obj_data-objekt. Feltet *img_drawed_list* er en liste med billeder, hvor annoteringerne fra både Ground Truth datasættet og Visionsystemet er blevet påtegnet. Afslutningsvist har vi *error_list*, som indeholder en beskrivelse af de fejl, der kunne være medført ved rapportgenereringen. Dette kan forekomme, hvis værktøjet støder ind i to annoteringsobjekter, som forsøges sammenlignet, men som ikke har samme form eller hvis datasættene ikke indeholder lige mange Image objekter.

Værktøjet fungerer på den måde, at det tager to lister med Image objekter som input, som kommer fra dataen i JSON-filerne. Herefter bliver objekterne med det samme filnavn sammenlignet og afvigelserne på deres annoteringer bliver udregnet.

Måden afvigelsen bliver udregnet på, er forskellig for rektangler og cirkler. For cirkler ses der på centrum og radius. Afvigelsen for centrum udregnes ved at finde afstanden mellem to punkter ved at gøre brug af formlen $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. Implementeringen af denne kan findes i Pythons math bibliotek og kan kaldes ved at bruge metoden *math.hypot*$(x_1 - x_2, y_1 - y_2)$. Radiussen bliver udregnet med simpel subtraktion, som bliver afrundet og den absolutte værdi returneres.

Udregningen for rektanglerne er meget lignende. Her er der tale om fire forskellige afstande, der skal udregnes - en afstand for hvert hjørne i rektanglet. Metoden fungerer således: For hvert hjørne i det detekterede rektangel bliver afstanden til hvert hjørne i det annoterede rektangel udregnet. Herefter returneres den korteste afstand.

Der er nu dannet et overblik over, hvordan de forskellige værktøjer er fremstillet, arbejder sammen og hvordan afvigelser er udregnet. I næste kapitel ser vi nærmere på rapportdataen.

Kapitel 5

Analyse

I dette kapitel vil vi præsentere resultaterne fra de forsøg, vi har foretaget.

5.1 Analysestrategi

Som nævnt i metoden arbejder vi med tre forskellige datasæt. Vi opsummerer kort her:

1. Datasæt med fokus på den gennemsnitlige lysniveau i hvert billede. Lysniveauerne er hhv. 18, 48, 78, 108 og 138. Der er ikke blevet anvendt noget filter. Placeringerne af objekterne er som beskrevet i kapitel 3.

2. Datasæt med infrarødt filter i kameraet. Billederne er blevet taget på en Linux-pc (Ubuntu 16.04).

3. Datasæt 3 er ligeledes et datasæt med infrarød filtrering af lyset. Billederne er taget på en Mac (OS X - El Capitan 10.11.4).

Vi sammenligner afvigelsen i mellem Visionsystemets annoteringer og vores Ground Truth data. I stedet for at sammenligne alle datapunkter ser vi på den gennemsnitlige afvigelse for et objekt. Det vil sige, at for en petriskål, der har center og radius, vil den gennemsnitlige afvigelse være $\frac{centerafvigelse+radiusafvigelse}{2}$. For en brøndplade vil det så være afvigelsen for de fire punkter: $\frac{p1+p2+p3+p4}{4}$.

Vi ser desuden på standardafvigelsen, der også vil blive refereret til som spredningen. Standardafvigelsen er et udtryk for observationernes variation omkring middelværdien. Observationerne er afvigelserne for hvert objekt i datasættene. Standardafvigelsen er altså objektets gennemsnitlige afvigelses variation omkring middelværdien.

Vi har også lavet diagrammer, der viser normalfordelingen for afvigelser. Vi har lavet ét diagram for alle datasættene samlet og ét for hvert datasæt. Diagrammerne anvendes til at visualisere dataen og give overblik. De vil ikke blive brugt i selve analysen da vi har for få datapunkter. Dataen er udregnet med metoden *stat.norm.pdf()* fra python biblioteket SciPy. Metoden er en implementation af frekvensfunktionen, hvormed vi får frekvensen på y-aksen og den gennemsnitlige afvigelse på x-aksen i diagrammet.

5.2 Overordnede resultater

Når vi ser overordnet på alle datasættene, er der ét gennemgående mønster. Det er svært at finde én lysopsætning og konfiguration af Visionsystemet, som gør det muligt at finde både petriskåle og brøndplader. Faktisk er det ikke lykkes at detektere en eneste brøndplade i det første datasæt. Ligeledes har vi ikke kunne detektere en eneste petriskål i de to sidste datasæt. Dette resultat er dog ikke overraskende; Scriptet som detekterer brøndplader er nødt til at kunne fratrække baggrunden for at få gode resultater. Disse får vi ved anvendelse af infrarødt lys. Scriptet som detekterer petriskåle er ikke så følsomt overfor støj og kan sagtens detektere petriskålene i det første datasæt. Til gengæld falder lyset i datasættene med infrarødt lys på en måde, så petriskålene ikke bliver oplyst, men i stedet reflekterer lyset i form af små prikker, som scriptet ikke kan detektere, selv om baggrunden bliver effektivt fjernet.

Tabel 5.1: Nøjagtighed på tværs af alle datasæt.

	Total (antal)	Detekteret (antal)	Detekteret (procent)
Alle Objekter	420	215	51,19 %
Petriskåle	105	11	10,48 %
Brøndplader	315	204	64,76 %

Set på tværs af alle datasættene burde der blive detekteret 420 objekter. Visionsystemet har fundet 215 objekter, hvilket giver en nøjagtighed på 51,2%. 105 af de 420 objekter er petriskåle. Heraf er 11 i blevet detekteret. Dette giver en nøjagtighed på 10,48 %. De resterende 315 objekter er brøndplader. 204 af dem er blevet detekteret. Dette giver en nøjagtighed på 64,76 %.

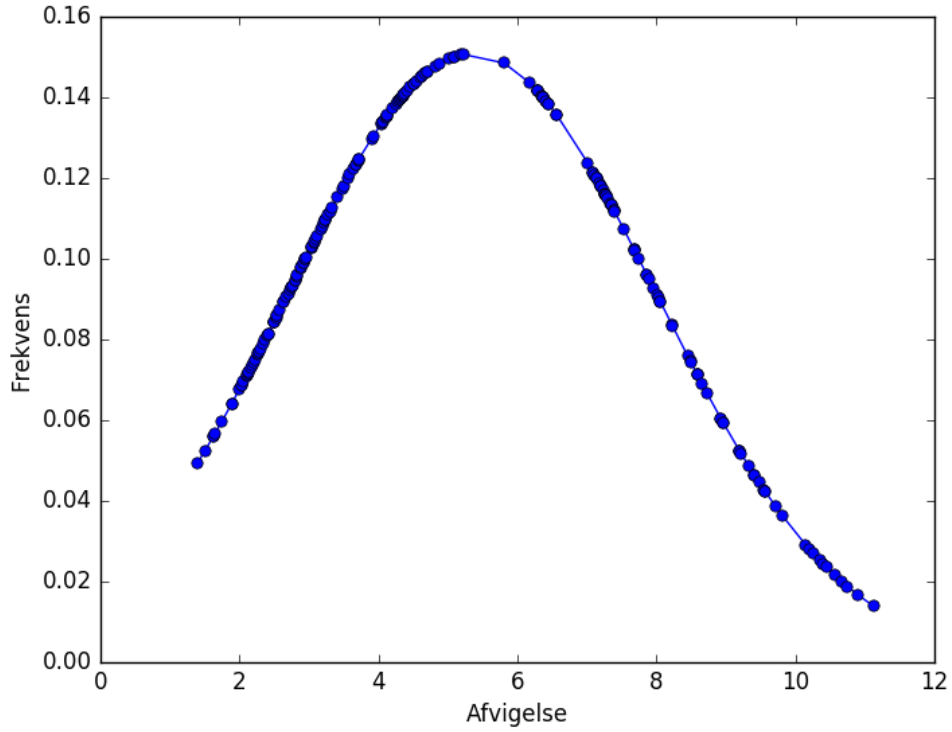
Som det ses i tabel 5.1 er nøjagtigheden overvejende lav, hvilket hænger sammen med, at scriptsne er designet hovedsageligt til hvert deres datasæt. Der er tre gange så mange billeder af brøndplader som af petriskåle. Det skyldes at der er inkluderet tre rotationer for hver placering af brøndpladerne. Årsagen til at så lav en procentdel af petriskålene

er blevet detekteret, er at de kun er blevet detekteret i datasæt 1 - et datasæt som er væsentlig mindre end de to andre.

Tabel 5.2: Præcision på tværs af alle datasæt.

	Gennemsnitlig afgivelse (px)	Standardafvigelse (px)
Alle Objekter	5,35	2,65
Petriskåle	3,95	2,04
Brøndplader	5,42	2,65

Som det ses i tabel 5.2 er den gennemsnitlige afvigelse forholdsvis lav. Anskues alle objekter samlet, er den gennemsnitlige afvigelse på 5,35 px. For petriskåle er 3,95 px og for brøndplader er den 5,42 px. Præcisionen for petriskålene er altså en smule bedre end for brøndpladerne. Standardafvigelsen for alle objekter er 2,65 px. For petriskåle er den 2,04 px og 2,65 px, for brøndplader. I figur 5.1 ses et diagram over normalfordelingen af alle detektionerne.



Figur 5.1: Normalfordeling på tværs af alle tre datasæt.

Disse resultater er meget overordnede. For at få et retvisende billede af nøjagtigheden og præcisionen, er vi derfor nødt til at betragte hvert datasæt for sig.

5.3 Datasæt 1: Det Visuelle Spektrum

Som det ses i tabel 5.3 har scriptet ikke været i stand til at finde en eneste brøndplade. Derfor vil vi se isoleret på billederne med petriskåle.

Tabel 5.3: Nøjagtighed for datasæt 1.

	Total (antal)	Detekteret (antal)	Detekteret (procent)
Alle Objekter	60	11	18,33 %
Petriskåle	15	11	73,33 %
Brøndplader	45	00	00,00 %

Datasættet består af 60 billeder, der hver især har én annotering. Ud af disse 60 billeder indeholder 15 en petriskål, hvoraf 11 af disse er blevet detekteret. Det svarer til 73,33%.

Dette resultat skal yderligere ses i forhold til det gennemsnitlige lysniveau. Ingen petriskåle er blevet detekteret ved det laveste gennemsnitlige lysniveau (18). Ved dette lysniveau kan petriskålen også kun lige kan ænses med det blotte øje (se figur 5.2). Bortset fra de tre billeder med et gennemsnitligt lysniveau på 18, er alle petriskåle på

Figur 5.2: Her ses et eksempel på et billede med et gennemsnitligt lysniveau på 18. Petriskålen i midten kan kun lige akkurat ænses med det blotte øje.

nær én blevet fundet. Som det ses i figur 5.3, er det også her svært at se petriskålen. Det gennemsnitlige lysniveau er nemlig 38, hvilket er lavere end tilsigtet. Dette kan være forklaringen på den manglende detektion.

Figur 5.3: Her ses den eneste petriskål, der ikke blev detekteret hvor det gennemsnitlige lysniveauet er over 18. Grunden til dette er sandsynligvis, at det gennemsnitlige lysniveau er 38 i stedet for 48, som den burde være.

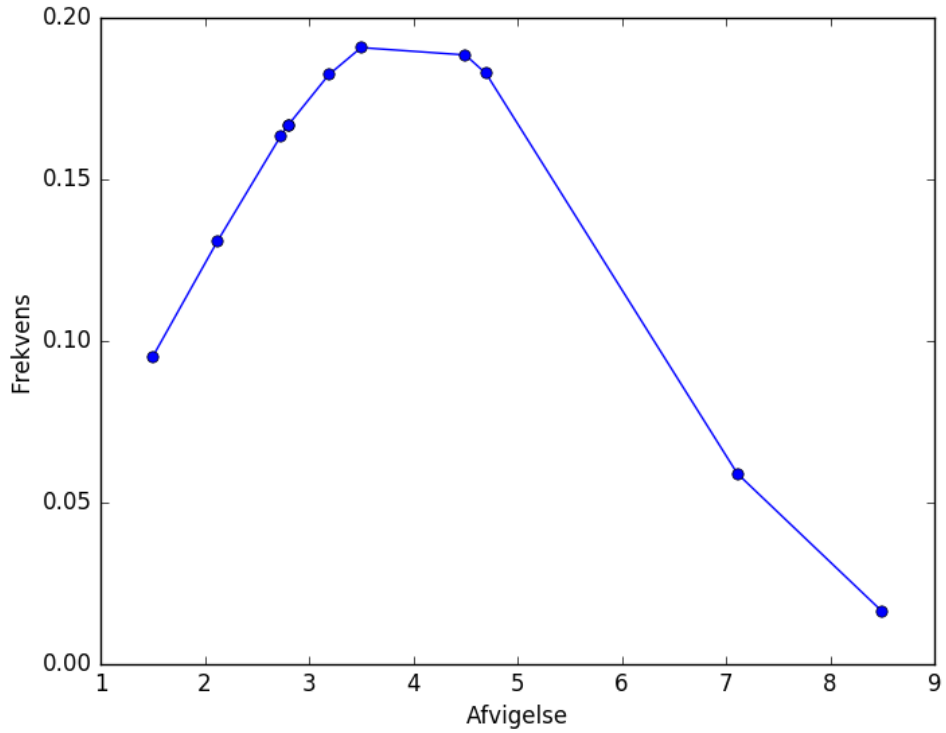
Det lader altså til at detektionssystemet for petriskåle fejler ved et lysniveau på under 48. Vi fortsætter til præcisionen for de detekterede petriskåle.

Tabel 5.4: Præcision for petriskåle i datasæt 1.

	Gennemsnitlig afgivelse (px)	Standardafvigelse (px)
Petriskåle	3,95	2,04

Som tabel 5.4 viser, er der en gennemsnitlig afvigelse på 3,95 px og en standardafvigelse på 2,04 px. Hvis man ser på afvigelserne for billederne hver for sig, er der nogle enkelte, som stikker ud. Billede 007 er det billede som har den højeste afvigelse. Afvigelsen på 8,5 px er gennemsnittet af afvigelsen fra centrum på 8,0 px og afvigelsen fra radius på 9,0 px. På figur 5.5 ses annoteringen påtegnet med grønt og detektionen med rødt. Billedet har et gennemsnitligt lysniveau på 48. Petriskålens position i dette billede er i siden. Her ses det, at den relative vinkel påvirker præcisionen, da scriptet tilsyneladende detekterer petriskålen langs dens øvre kant. Billede 013 er det billede som har den laveste afvigelse på 1,5 px.

En anden udstikker på afvigelsen er billede nummer 009. Den gennemsnitlige afvigelse for dette billede er på 7,11 px med 7,21 px fra centrum og 7,0 px fra radius. Billedet har

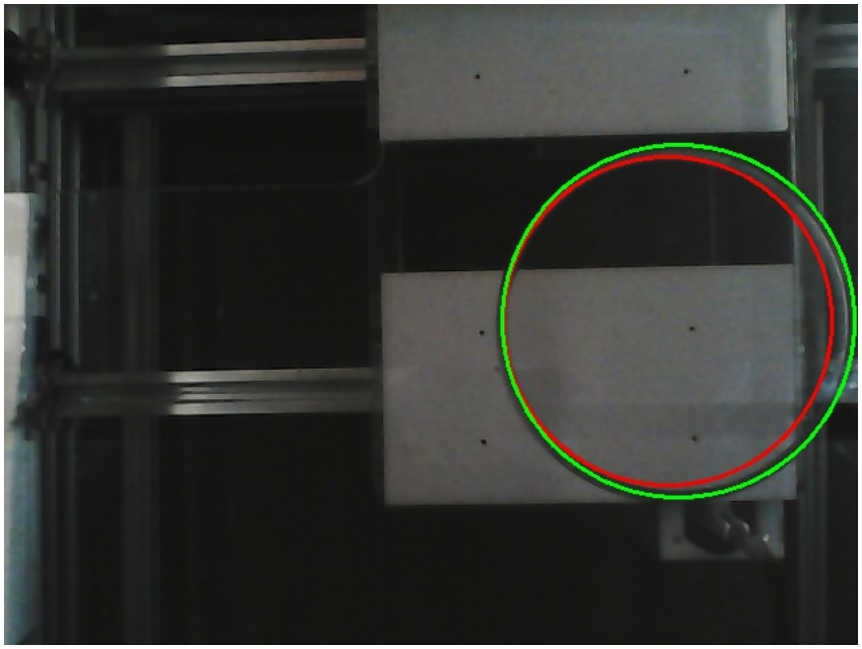


Figur 5.4: Her ses normalfordelingen af detektionerne i datasæt 1.

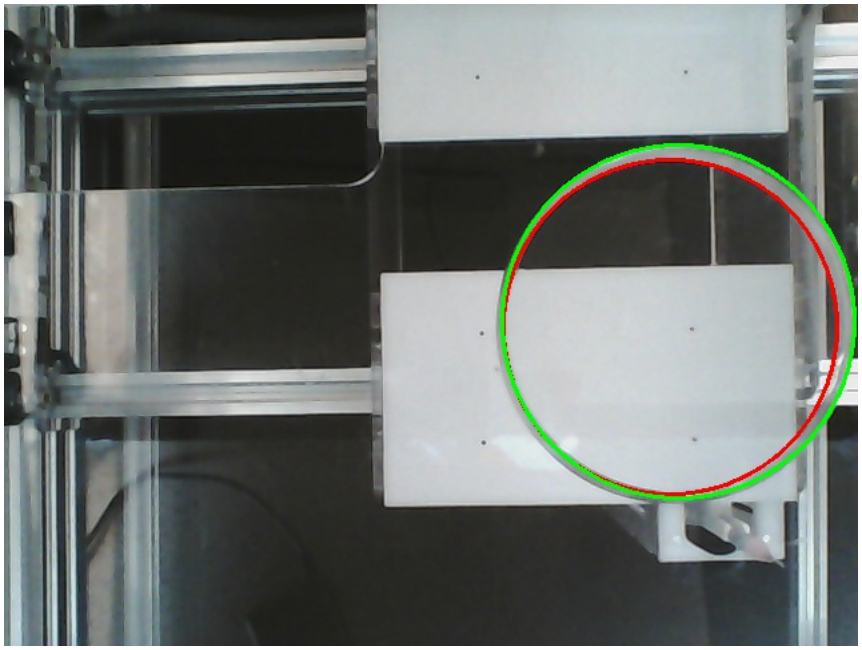
et gennemsnitligt lysniveau på 111. Det skulle have været 108. Ser man på figur 5.6 ses det, at der er sket det samme som på billede 007. Scriptet fanger igen den øvre kant af petriskålen.

Der tegner sig et billede af, at præcisionen falder ved denne vinkel. Det gælder også i mindre grad for billede 008. Den gennemsnitlige afvigelse er på 4,5 px. Centerafvigelsen er 6,0 px og radiusafvigelsen er 3,0 px. Problemet er også her, at systemet registrerer den øvre kant. Dette er dog ikke gældende for alle billeder. Et eksempel er billede 010. Dette billede har et gennemsnitligt lysniveau på 139 og petriskålen har den samme placering som 007, 008 og 009. Den gennemsnitlige afvigelse ligger på 2,73 px. Centerafvigelsen er 4,47 px, mens radiusafvigelsen er 1,0 px. Afvigelsen er lavere, da scriptet detekterer petriskålen korrekt.

At scriptet detekterer den øvre kant ses ikke udelukkende ved en høj vinkel mellem objektet og kameraet. På billede 003 sker dette også og her er petriskålen placeret i midten. Det giver en gennemsnitlig afvigelse på 4,7 px fordelt på 5,39 px på centerafvigelsen og



Figur 5.5: Som billedet viser, detekterer scriptet (den røde streg) den øvre kant af petriskålen, som fremtræder når objektet placeres i siden. Billedet er 007.jpg fra datasæt 1.



Figur 5.6: Her ses det at den øvre kant på petriskålen detekteres. Billedet er taget fra datasæt 1 og hedder 009.jpg.

4,0 px på radiusafvigelsen. Det er dog ikke et markant udslag i den gennemsnitlige afvigelse, da afstanden mellem den indre og den ydre cirkel ikke er særlig høj.

En opsummering af de omtalte billeder ses i tabel 5.5. For god ordens skyld ser vi også se på de billeder, der har den laveste afvigelse. Afvigelserne for disse findes i den nederste del af tabellen.

Tabel 5.5: Præcision for billeder af interesse.

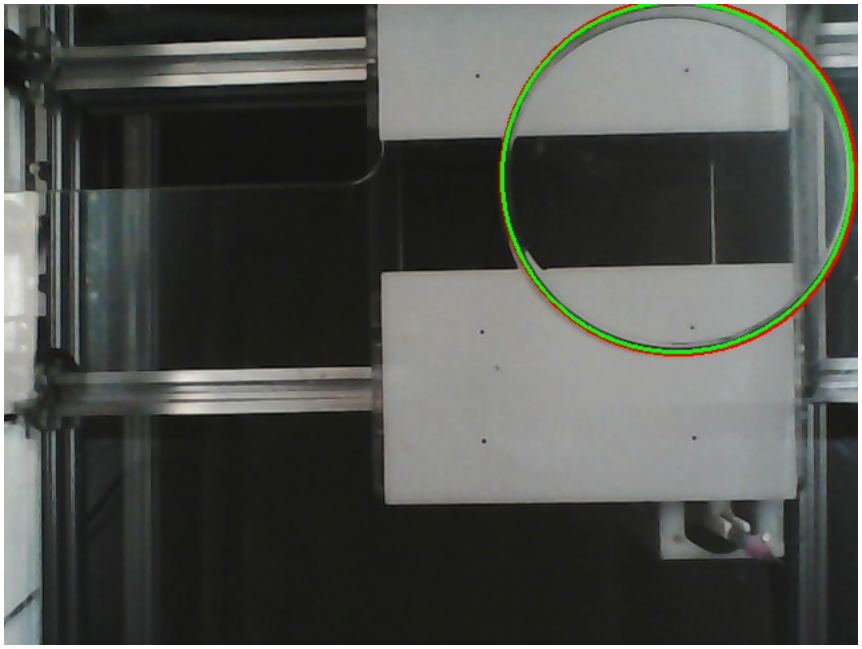
Billede	Gns. lysniveau	Center Afv.	Radius Afv.	Gns. Afv.
003.jpg	78	5,39	4,0	4,7
007.jpg	48	8,0	9,0	8,5
008.jpg	77 (78)	6,0	3,0	4,5
009.jpg	111 (108)	7,21	7,0	7,11
010.jpg	139 (138)	4,47	1,0	2,73
013.jpg	78	1,0	2,0	1,5
004.jpg	109 (108)	2,24	2,0	2,12

Placeringen af billede 013 er i hjørnet, som kan ses på figur 5.7. Det gennemsnitlige lysniveau er 78. Den gennemsnitlige afvigelse er 1,5 px Centerafvigelsen er 1,0 px og radiusafvigelsen er 2,0 px. Det billede med den næstlaveste gennemsnitlige afvigelse(2,12) er 004, hvor placeringen er i midten og det gennemsnitlige lysniveau er 109 (den skulle ligge på 108).

Opsummerende er der nogle udstikkere i forhold til standardafvigelsen. Dette kan forklares ved, at scriptet enkelte steder detekterer den øvre kant af petriskålen. Dette er mest tydeligt på de billeder, hvor petriskålen er placeret i siden, men kan også forekomme, hvor petriskålen er i midten. Det lader ikke til, at der er en sammenhæng mellem præcisionen og det gennemsnitlige lysniveau. Sammenhængen findes i forhold til nøjagtigheden, hvor vores datasæt har vist, at hvis det gennemsnitlige lysniveau er under 48, bliver petriskålen ikke detekteret. Med så lille et datasæt er det svært at sige om placeringen påvirker præcisionen. I siden præcisionen værst. Men i hjørnet, hvor det kunne antages at formen på petriskålen blev mest forvrænget, er præcisionen højest.

5.4 Datasæt 2: Infrarød belysning, Linux

Dette datasæt består af 140 billeder, hvoraf 120 af dem har én annotering hver. Ud af de 120 annoteringer er 90 af dem brøndplader, mens 30 af dem er petriskåle. Grunden til at dette datasæt er så markant større end det første, er, at der er blevet taget 10 billeder



Figur 5.7: Her ses billed 013.jpg med den laveste afvigelse. Billedet er taget fra datasæt 1.

for hver position af objektet. Som det kan ses i tabel 5.6, er der ikke blevet detekteret en eneste petriskål, hvorfor vi i dette datasæt kun tager brøndpladerne i betragtning.

Tabel 5.6: Nøjagtighed for datasæt 2.

	Total (antal)	Detekteret (antal)	Detekteret (procent)
Alle Objekter	120	90	75 %
Petriskåle	30	0	0 %
Brøndplader	90	90	100 %

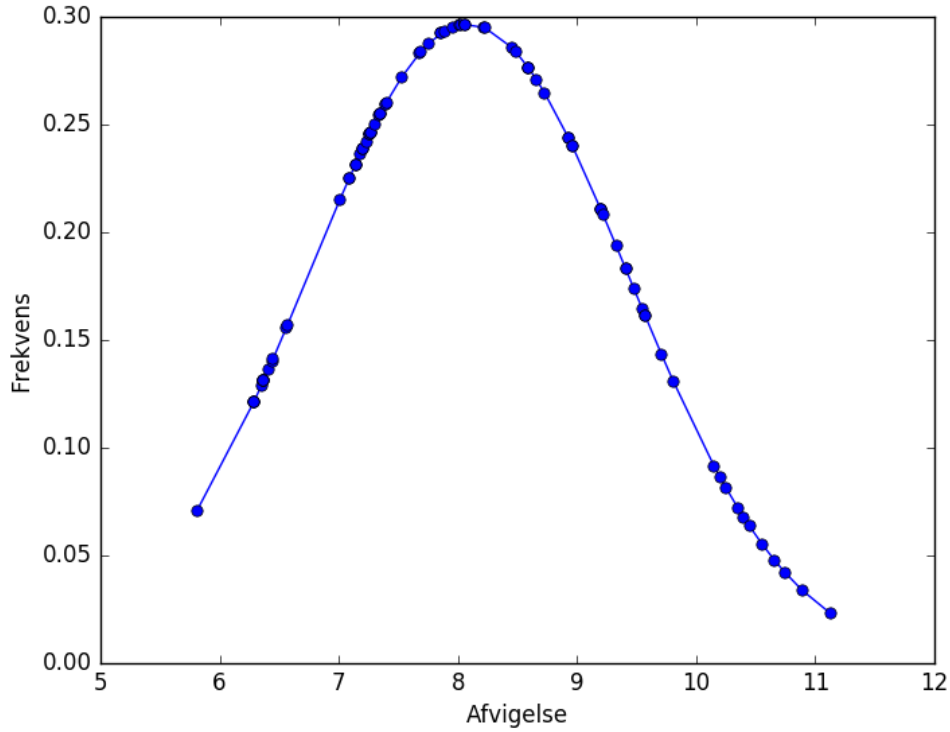
Som tabel 5.6 viser, er der blevet detekteret 90 ud af 120 objekter, hvilket svarer til 75 %. Samtlige brøndplader blevet detekteret, og der er ikke blevet detekteret nogen petriskåle. Eftersom alle brøndpladerne er fundet og ingen af petriskålene blev fundet, vil vi ikke kigge nærmere på nøjagtigheden for dette datasæt, men i stedet fokusere på præcisionen.

Tabel 5.7: Præcision for brøndplader i datasæt 2.

	Gennemsnitlig afgivelse (px)	Standardafvigelse (px)
Brøndplader	8,09	1,35

Vi måler præcisionen i dette datasæt ved at sammenligne koordinatsættene til de fire

hjørnepunkter på brøndpladerne. Vi anskuer afvigelsen som gennemsnittet af afvigelserne for hjørnerne. Det gør vi for at holde analysen på det abstraktionsniveau, hvorfra vi kan udtrække nogle meningsfulde betragtninger. Enkelte steder ses der på afvigelsen for hvert hjørne. Herudover grupperer vi billederne. For hver placering og rotation, har vi taget ti billeder. Disse ti billeder kalder vi for en billedserie. Gennemsnittet for en billedserie er således gennemsnittet af disse ti billeders gennemsnitlige afvigelse. Ligeledes illustreres normalfordelingen i figur 5.8.



Figur 5.8: Her ses et diagram over normalfordelingen af detektionerne i datasæt 2.

Som det ses i tabel 5.7, er den gennemsnitlige afvigelse på 8,06 px. Spredningen er 1,36 px. På nogle billedserier stikker den gennemsnitlige afvigelse ud. Hvis vi f.eks. kigger på de billeder, der er placeret i hjørnet med en rotation på 45 grader, er den gennemsnitlige afvigelse højere. I tabel 5.8 ses gennemsnittet af afvigelserne for hver placering og rotation, hvor tallet i parentesen er standardafvigelsen.

Som tabellen viser har billedserien med brøndplader, der er placeret i hjørnet med en rotation på 45 grader, en relativt høj afvigelse. Den gennemsnitlige afgivelse for alle

Tabel 5.8: Gennemsnitlig afvigelse for brøndplader ud fra placering og rotation i datasæt 2.

	0 grader	45 grader	90 grader
Midten	7,96 (0,21)	8,08 (0,53)	6,75 (0,36)
Siden	9,05 (0,25)	9,55 (0,31)	6,82 (0,16)
Hjørnet	7,25 (0,1)	10,56 (0,29)	7,31 (0,11)

objekter med denne placering og rotation er på 10,56 px og har en spredning på 0,29 px. Den lave spredning viser, at der er en høj afvigelse for alle billeder i serien. Den højeste afvigelse er 11,13 på billede 112. Billede 116 har den laveste afvigelse på 10,15 i denne billedserie.

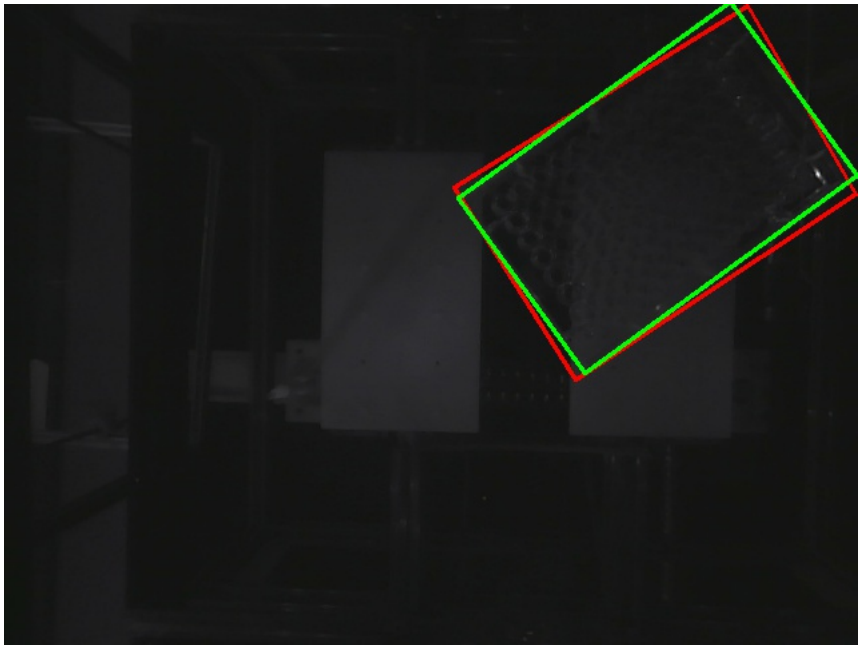
Tabel 5.9: Afvigelser for billedgruppen 111 til 120.

Billede	Hjørne 1 afv.	Hjørne 2 afv.	Hjørne 3 afv.	Hjørne 4 afv.	Gns. afv.
111	13,34	7,21	8,06	14,04	10,66
112	13,34	8,54	8,6	14,04	11,13
113	13,34	6,4	7,62	14,04	10,35
114	12,37	8,54	8,6	14,04	10,89
115	12,37	6,4	9,43	14,04	10,56
116	12,37	6,4	7,81	14,04	10,15
117	12,37	8,54	8,06	14,04	10,75
118	13,34	7,21	7,21	14,04	10,45
119	13,34	6,4	7,81	14,04	10,4
120	13,34	6,4	7,21	14,04	10,25

Tabel 5.9 viser, at der er en tydeligt højere afvigelse for hjørne 1 og hjørne 4 end de to resterende. Ser man på billede 112 i figur 5.9 er detektionen skæv. Den røde streg, der illustrerer detektionen, er roteret. Dette kan hænge sammen med at brøndpladens to yderste hjørner er placeret udenfor billedet. Dette er gældende for samtlige billeder i denne billedgruppe.

Billedserien hvor brøndpladen er placeret i siden med en rotation på 45 grader stikker også ud ift. den gennemsnitlige afvigelse. Billederne, der er tale om, er nummer 081 til 090. Billedserien har en gennemsnitlig afvigelse på 9,55 px og en lav spredning på 0,31 px. Den lave spredning viser, at der ikke er stor forskel på afvigelsen i de enkelte billeder i serien.

Ud fra tabel 5.10 ses det at alle brøndplader i billedserien har en markant højere afvigelse på hjørne 4. Dette udslag kan hænge sammen med, at hjørne 4 ligger lige udenfor



Figur 5.9: Som billedet viser, er detektionen en anelse roteret i forhold til, hvor objektet egentlig er placeret. Billedet er taget fra datasæt 2 og hedder 112.jpg.

billedet. Som det ses i figur 5.10, er detektionen igen roteret (som ved figur 5.9). Det tyder på, at præcisionen falder, hvis et eller flere hjørner er ude af kameraets vinkel.

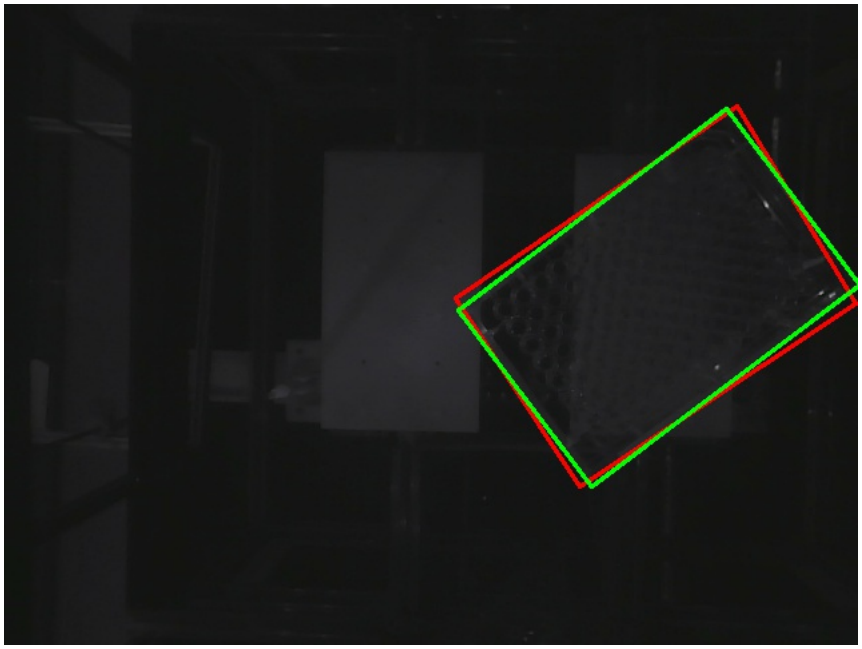
Nu har vi anskuet to steder, hvor der har været en særlig høj afvigelse. Begge tilfælde har været brøndplader med en rotation på 45 grader. Begge steder har ligeledes været påvirket af, at brøndpladerne har været placeret lidt uden for kameraets synsfelt. Derfor ser vi nu på den sidste placering, hvor brøndpladen også har en 45 graders rotation. Placeringen er i midten og afvigelsen for billedserien er relativt høj med en gennemsnitlig afvigelse på 8,08 px.

Som dataen i tabel 5.11 viser er der en markant højere afvigelse på hjørne 4. Detektionen er ligeledes roteret i denne billedserie. Altså er det måske den 45 graders rotation og ikke kameraets synsfelt, der forsager afvigelsen. Præcisionen er generelt lavest i billedserierne med 45 grader rotation. Her vi finder de to højeste afvigelser og den fjerdehøjeste. Den tredjehøjeste afvigelse er billedserien med brøndplader, der er placeret i siden med 0 graders rotation.

Den laveste gennemsnitlige afvigelse findes de i billedserierne, hvor brøndpladerne er roteret 90 grader. For både midten, siden og hjørnet, er afvigelsen ca. 7 px. Dette

Tabel 5.10: Afvigelser for billedserien 081 til 090.

Billede	Hjørne 1 afv.	Hjørne 2 afv.	Hjørne 3 afv.	Hjørne 4 afv.	Gns. afv.
081	6,71	7,62	7,07	14,32	8,93
082	8,25	9,22	9,0	14,32	10,2
083	6,71	9,22	9,0	14,32	9,81
084	6,71	7,62	9,0	14,32	9,41
085	6,71	8,25	9,0	14,32	9,57
086	6,71	8,25	9,0	14,32	9,57
087	6,71	7,62	9,0	14,32	9,41
088	8,25	9,22	7,07	14,32	9,71
089	6,71	9,22	7,07	14,32	9,33
090	7,28	7,62	9,0	14,32	9,55



Figur 5.10: Som billedet viser, er detektionen roteret i forhold til hvor brøndpladen egentlig er placeret. Billedet er taget fra datasæt 2 og hedder 082.jpg.

illustreres i tabel 5.8, hvor det kan ses, at midten har en afvigelse på 6,75 px. Siden har en afvigelse på 6,82 px mens hjørnet har en afvigelse på 7,31 px. Den laveste afvigelse for i et billede over hele datasættet er imidlertid billed 076 med en afvigelse på 5,81 px.

Ser man overordnet på alle billedserierne, er der en afvigelse på alle billeder på over 6,75 px. Et andet fænomen der kan observeres ud fra datasættet, er, at alle brøndpladerne, der er blevet detekteret, er en smule større end dem vi har annoteret. Det eneste sted

Tabel 5.11: Afvigelser for billedgruppen 051 til 060.

Billede	Hjørne 1 afv. (px)	Hjørne 2 afv. (px)	Hjørne 3 afv. (px)	Hjørne 4 afv. (px)	Gns. af
051	6,4	7,62	5,83	9,22	7,2
052	6,4	9,49	5,83	12,65	8,5
053	6,4	5,83	5,83	12,65	7,6
054	6,4	9,49	5,83	11,18	8,2
055	6,4	9,49	5,83	12,08	8,4
056	5,66	8,54	5,83	12,08	8,0
057	6,4	5,83	5,83	10,3	7,0
058	6,4	8,54	5,83	12,08	8,2
059	6,4	9,49	5,83	12,65	8,5
060	6,4	10,2	5,83	12,21	8,6

hvor dette ikke er gældende, er i de billedserier, hvor brøndpladen har en rotation på 45 grader.

Opsummerende har vi ikke detekteret nogen petriskåle i dette datasæt, men derimod er samtlige brøndplader blevet detekteret. For de detekterede brøndplader ses der en gennemsnitlig afvigelse på 8,09 px med en standardafvigelse på 1,36 px. Når man går ned i dataen ses det, at brøndpladerne med en rotation på 45 grader generelt har en højere afvigelse end resten af rotationerne. Alt i alt ligger præcisionen over 6,75 px for alle placeringer og rotationer. Blandt de brøndplader der er blevet detekteret, er der en overvejende sammenhæng mellem årsagen til afvigelserne og den rotation som brøndpladen har. Hvis brøndpladen har en rotation på 45 grader, bliver det detekterede rektangel forskudt i rotationen, mens resten primært bliver forstørret.

5.5 Datasæt 3: Infrarød Belysning, Mac

Det tredje og sidste datasæt indeholder 240 billeder, hvoraf alle billederne indeholder én annotering hver. Ud af de 240 billeder er 180 af dem brøndplader, mens 60 af dem er petriskåle. Dette datasæt er dobbelt så stort som det forhenværende, hvis man ser bort fra de billeder, der ikke indeholder et objekt i datasæt 2. I dette datasæt har vi altså ikke inkluderet billeder uden objekter. Grunden til at datasættet er større, er, at vi har udbygget datasættet til også at indeholde billeder, hvor den eksperimentelle overflade er helt dækket til af den lystætte kasse. På de billeder hvor den ikke er fuldt overdækket, har vi brugt en minimal afskærmning på toppen af robotten. Ligesom i

datasæt 2 er der ikke blevet detekteret nogen petriskåle. Vi vælger derfor udelukkende at se på brøndpladerne.

Tabel 5.12: Nøjagtighed for datasæt 3.

	Total (antal)	Detekteret (antal)	Detekteret (procent)
Alle Objekter	240	114	47,5 %
Petriskåle	30	0	0 %
Brøndplader	180	114	63,33 %

Som tabel 5.12 viser, er der detekteret 114 objekter ud af samtlige 240, hvilket svarer til 47,5 %. Samtlige 114 objekter er brøndplader, hvilket svarer til, at 63,33 % af brøndpladerne er blevet detekteret. Denne nøjagtighed er ikke nær så høj som i datasæt 2, hvor den var på 100%. Forskellen på datasættene er, at dette er lavet på en Mac med OS X, hvor datasæt 2 er lavet på en computer med Linux (Ubuntu). Herudover har robotten været helt dækket til på halvdelen af billederne og den anden halvdel har været dækket til med et låg på toppen. I datasæt 2 brugte vi ingen afdækning.

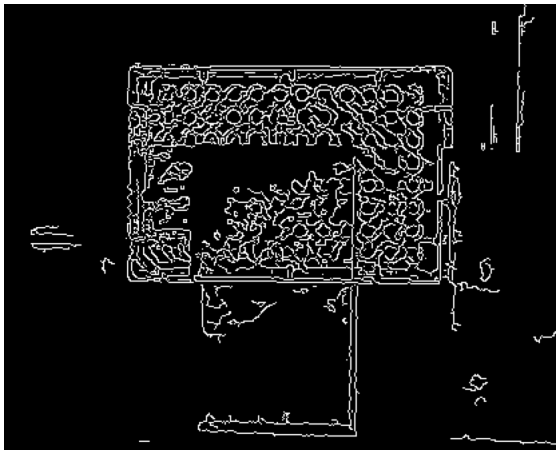
Som med datasæt 2 har vi lavet billedserier. For hver position og rotation er der først blevet taget ti billeder med låg på og derefter ti billeder, hvor den lystætte kasse har overdækket hele EvoBotten. En billedserie består således af 20 billeder. Vi blander billeder, hvor robotten er dækket helt til og billeder, hvor den ikke er, da vi ikke har fundet en sammenhæng mellem tildækning og nøjagtighed eller præcision.

Tabel 5.13: Nøjagtighed for datasæt 3 (fordelt på afskærmning).

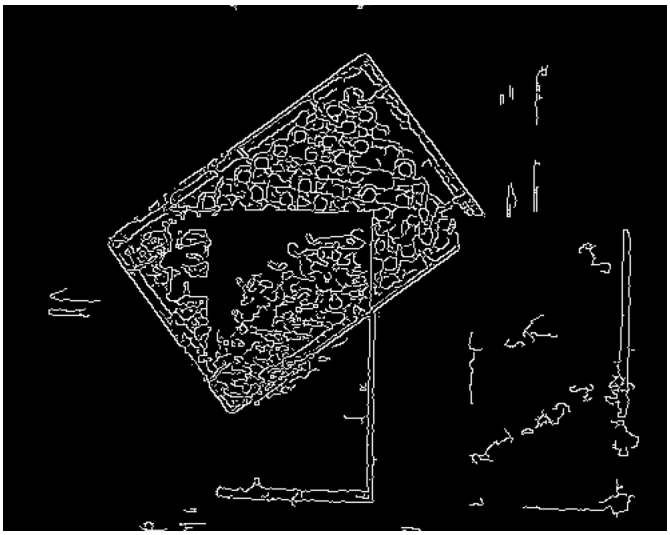
Billedserie	Med låg	Fuld afskærming
Billedserie 1 (001-020)	8	7
Billedserie 2 (021-040)	0	0
Billedserie 3 (041-060)	0	0
Billedserie 4 (061-080)	6	9
Billedserie 5 (081-100)	10	10
Billedserie 6 (101-120)	9	9
Billedserie 7 (121-140)	10	9
Billedserie 8 (141-160)	2	5
Billedserie 9 (161-180)	10	10
I alt antal	55	59
i alt procent	61,11 %	65,56 %

Som tabel 5.13 viser, er der ingen sammenhæng mellem afskærmning og andelen af detekterede brøndplader. Der er detekteret fire objekter mere ved brug af den lystætte

kasse end uden. Men på enkelte af billedserierne er der detekteret flere med låg (billedserie 1 og 7). Det interessante er, at der ikke er fundet en eneste brøndplade i billedserie 2 og 3. Det vil sige, at der for brøndplader placeret i midten med en rotation på hhv. 45 grader og 90 grader ikke er detekteret en eneste. Herudover er der kun fundet 7 ud af 20 brøndplader i billedserie 8. Dette kan muligvis skyldes, at de hvide plader på EvoBottens aktive lag forstyrrer detektionen (se figur 5.11 og 5.12).



Figur 5.11: De hvide plader genererer støj, der gør at brøndpladerne ikke kan approksimeres til et rektangel.



Figur 5.12: De hvide plader genererer støj, der gør at brøndpladerne ikke kan approksimeres til et rektangel.

Scriptet leder efter hjørnepunkter og når fire hjørner er fundet, anses dette som en firkant såfremt de indre vinkler er inden for acceptable margener. Som både figur 5.11

og 5.12 viser, er det svært at se et af hjørnerne på grund af den hvide plade. Ligeledes er det svært for scriptet at udregne, at der er en firkant. Dette gør, at den midterste placering klart er den placering med lavest nøjagtighed. Dog er der blevet detekteret femten brøndplader ved den midterste placering, hvor rotationen er 0 grader.

Det andet sted hvor nøjagtigheden er overvejende lav i forhold til resten, er i billedserie 8. Billedserien indeholder brøndpladerne, der er placeret i hjørnet med en rotation på 90 grader. Her er der kun detekteret syv ud af tyve brøndplader. Det kan hænge sammen med, at brøndpladen ikke er så oplyst som den er i andre billedserier. Det påvirker således scriptets evne til at detektere rektanglet. Dette anskueliggøres i figur 5.13.

Figur 5.13: Som billedet viser, er der kun en begrænset oplysning af brøndpladen. Billedet er taget fra datasæt 3 og hedder 142.jpg.

Generelt kan man aflæse et lavere gennemsnitligt lysniveau i datasæt 3 i forhold til datasæt 2. Det kan hænge sammen med at datasættene er blevet lavet på hvert deres styresystem, som muligvis kan have indflydelse på hvordan input fra kameraet læses. Et andet interessant punkt er at præcisionen er højere end i datasæt 2.

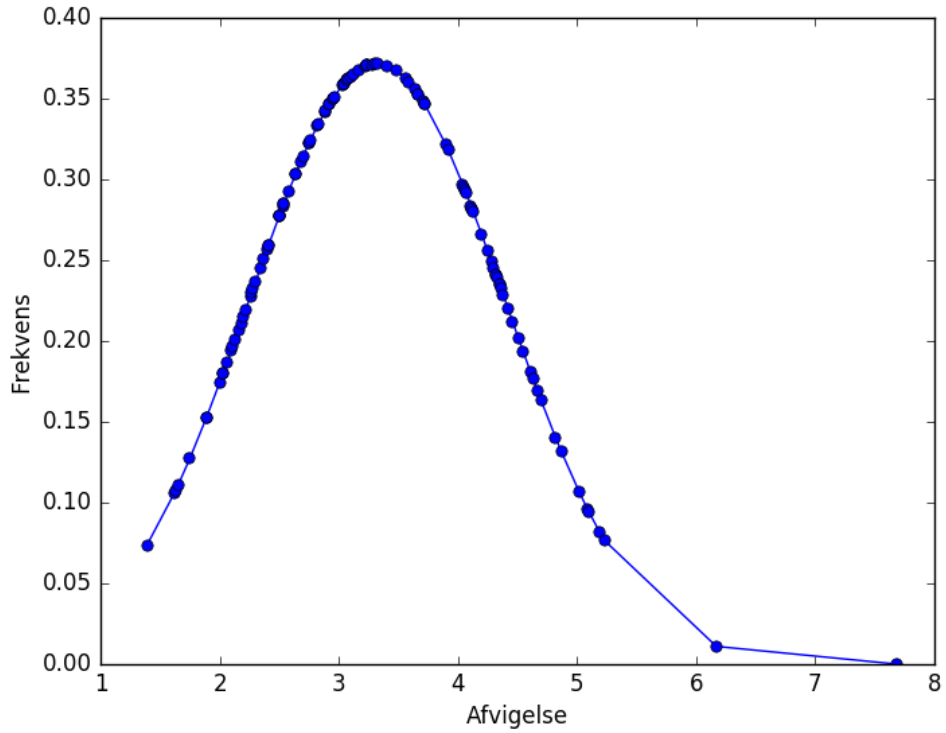
Tabel 5.14: Præcision for brøndplader i datasæt 3.

	Gennemsnitlig afvigelse (px)	Standardafvigelse (px)
Brøndplader	3,32	1,07

Som tabel 5.14 viser, er den gennemsnitlige afvigelse på 3,32 px med en spredning på 1,07 px. Dette er det hidtil laveste niveau for både gennemsnitlig afvigelse og standardafvigelse. Der er dog stadig nogle udstikkere, som vi ser nærmere på. Ligeledes vises normalfordelingen af brøndpladernes afvigelser i figur 5.14.

Tabel 5.15: Gennemsnitlig afvigelse for brøndplader ud fra placering og rotation i datasæt 3.

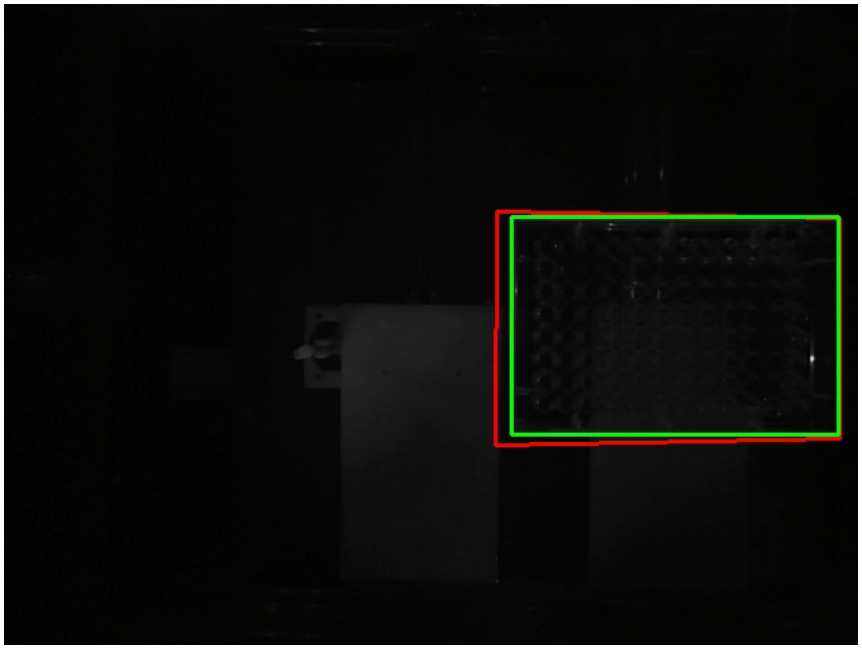
	0 grader	45 grader	90 grader
Midten	2,78 (0,88)	- (-)	- (-)
Siden	3,06 (0,5)	3,83 (0,43)	2,96 (1,49)
Hjørnet	2,78 (0,87)	4,48 (0,36)	2,92 (0,27)



Figur 5.14: Billedet illustrer normalfordelingen af de detekterede brøndplader i datasæt 3.

De billedserier der afviger mest i forhold til præcisionen, er serien med objekter i hjørnet, som er roteret 45 grader. Denne serie har en gennemsnitlig afvigelse på 4,48 px, som er den højeste afvigelse i dette datasæt. Herudover har billedserien med brøndplader til

siden og en rotation på 90 grader en forholdvis høj standardafvigelse på 1,49 px. Denne billedserie indeholder billedet med den højeste gennemsnitlige afvigelse (081).



Figur 5.15: Som billedet viser, går den detekterede firkant ud over brøndpladen og flugter med den hvide plade). Billedet er taget fra datasæt 3 og hedder 081.jpg.

Hvis man ser på figur 5.15, tydeliggøres det at det detekterede rektangel flugter med den hvide plade i baggrunden i stedet for brøndpladens kant. Det samme gør sig gældende i flere billeder i denne billedserie (081-100).

Opsummerende er der grundlæggende en lavere nøjagtighed i dette datasæt end i datasæt 2. Der er ikke en sammenhæng mellem nøjagtigheden af detektionen og tildækningen af robotten. Ligeledes er der i billedgruppe 8 (hjørnet, 90 graders rotation) svag belysning, der gør at kun 7 brøndplader er detekteret. Den gennemsnitlige afvigelse og standardafvigelsen er det hidtil laveste - både i forhold til datasæt 1 og datasæt 2. De steder hvor den gennemsnitlige afvigelse stikker meget ud, kan hænge sammen med at de hvide plader forstyrrer scriptets evne til at finde præcis, hvor kanterne i rektanglerne er forbundet. Brøndplader placeret i midten med en rotation på 0 og 45 grader, er ikke blevet detekteret, hvilket kan hænge sammen med, at det gennemsnitlige lysniveau er lavere i dette datasæt. Et lavere lysniveau betyder færre informationer i billedet, hvilket kan være årsagen til, at baggrunden ikke bliver ordentligt fratrukket. Dermed kommer

de hvide plader i baggrunden til at forstyrre detektionen. Forskellen på lysniveauet i datasæt 2 og 3 kan skyldes, at de er lavet på forskellige styresystemer.

5.6 Delkonklusion

Helt grundlæggende ses det ud fra dataen, at detektionen af hhv. petriskåle og brøndplader bliver kraftigt påvirket af filtreringen af lyset. Med infrarød belysning og et infrarødt filter i kameraet, har vi kun fundet brøndplader. Med et almindeligt kamera har vi kun fundet petriskåle. Det faktum gør, at det ikke giver mening at se overordnet på dataen, og det giver misvisende værdier særligt for nøjagtigheden.

I datasæt 1 testes nøjagtigheden og præcisionen i forhold til det gennemsnitlige lysniveau og placeringen af objektet. Her er udelukkende blevet detekteret petriskåle og ingen brøndplader. Herudover viser dataen, at systemet fejler, hvis det gennemsnitlige lysniveau kommer under 48. Yderligere bliver petriskålens øverste kant detekteret på nogle af billederne, hvilket påvirker præcisionen. Det er værd at bemærke, at denne fejldetektion ikke er sket på billederne med det højeste lysniveau.

I datasæt 2 er der udelukkende blevet detekteret brøndplader og ingen petriskåle. Samtlige brøndplader er blevet detekteret. I forhold til præcisionen af detektionen, har brøndplader med en rotation på 45 grader generelt en højere gennemsnitlig afvigelse. Der er en sammenhæng mellem hvordan afvigelsen opstår og hvordan brøndpladen er roteret. Ved 45 graders rotation er detektionens egenrotation skæv. Ved 0 og 90 grader bliver den detekterede petriskål forstørret.

I det tredje og sidste datasæt ses der en grundlæggende lavere nøjagtighed end i datasæt 2. Der ses en markant forbedring i præcision, da både den gennemsnitlige afvigelse og standardafvigelsen er lavere end i datasæt 2. Der er ligesom i datasæt 2 udelukkende blevet detekteret brøndplader. I billedserierne med brøndplader i midten med en rotation på 0 og 45 grader, er der ingen detektioner. Disse resultater kan hænge sammen med, at de to hvide plader, der er en del af robotten, forstyrrer detektionen.

Disse resultater diskuteres i det følgende kapitel. Herudover ser vi også på, hvad resultaterne betyder og hvordan det kan bidrage i en bredere optik.

Kapitel 6

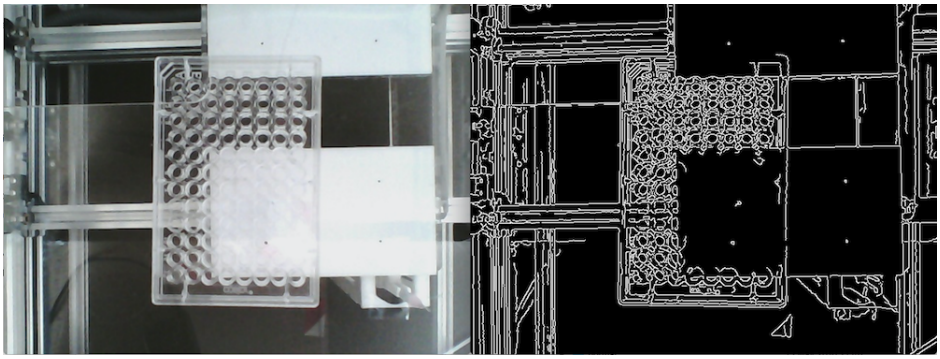
Diskussion

I dette kapitel vil vi diskutere resultaterne fra analysen på et højere abstraktionsniveau og koble de underliggende tendenser med vores grundlæggende problemstillinger og arbejdsspørgsmål. Vi vil træde et skridt tilbage og diskutere mulige forklaringer på disse tendenser på baggrund af observationer, vi har gjort igennem projektet. Da dette projekt er industrielt og formålet er at lægge et fundament for Computer Vision til EvoBotten, mener vi, at det er relevant at fremlægge både det, der fungerede og det der ikke gjorde.

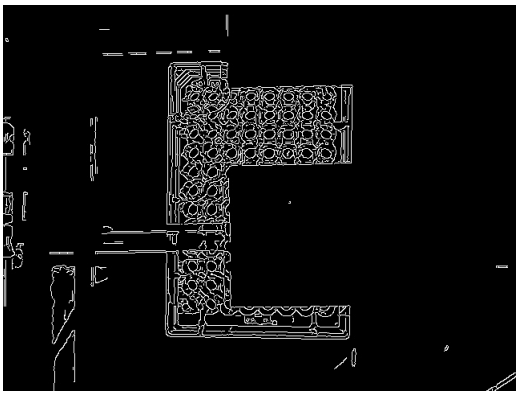
Et af de første fund vi har gjort i projektet, er, at det er svært at detektere brøndplader. Som nævnt i metoden, kapitel 3, er det gået op for os allerede da vi arbejdede på de tidlige prøvebilleder. En sandsynlig forklaring på dette har at gøre med de karakteristika, vi leder efter. Petriskålen parametriserer vi som en cirkel og da der ikke er andre objekter på billedet, som har form som en cirkel, er de ikke vanskelige at finde. Brøndpladerne parametriserer vi til et rektangel sammensat af 4 krydsende linjer med indre vinkler på 90 grader +- 5%. Som det kan ses på figur 6.1, danner baggrunden efter forarbejdelse mange rektangler, men ingen af dem indikerer korrekt hvor brøndpladen er. De to hvide plader, i det aktive lag, danner her i høj grad støj, så dem vender vi tilbage til. Først vil vi diskutere opbygningen af vores datasæt.

6.1 Datasæt

Der er forskel på måden, de tre datasæt er opbygget. I datasæt 1 forsøger vi at kontrollere lys i det visuelle spektrum. I datasæt 2 og 3 går vi over til at bruge infrarødt lys. Som nævnt i metoden, er det vanskeligt at lave en model for lysmiljøet i det visuelle spektrum, som giver et målestoksforhold uden at blive for simpel. Vores model, med en lystæt kasse, giver os en måde at måle det gennemsnitlige lysniveau for hvert billede. Men da



Figur 6.1: Ovenstående billede viser den støj der bliver introduceret efter billedmanipulation.

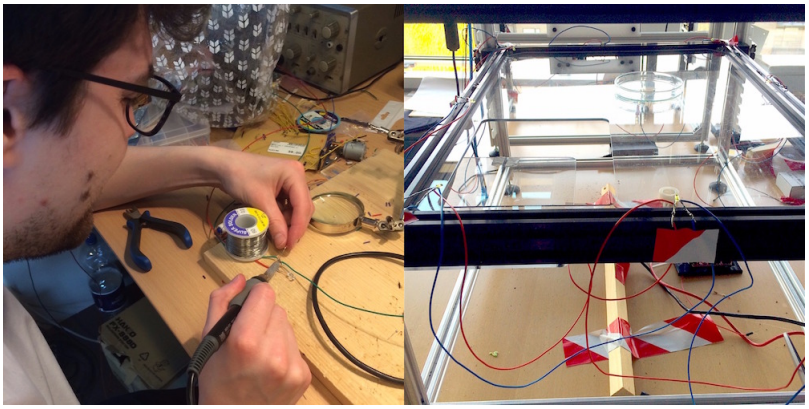


Figur 6.2: Ovenstående billede viser den støj, der bliver introduceret, efter baggrunden er fratrukket og billedmanipulation er blevet udført. De to områder der mangler af brøndpladen på billedet, kommer efter fratrækning af baggrunden og skyldes støj fra de hvide plader.

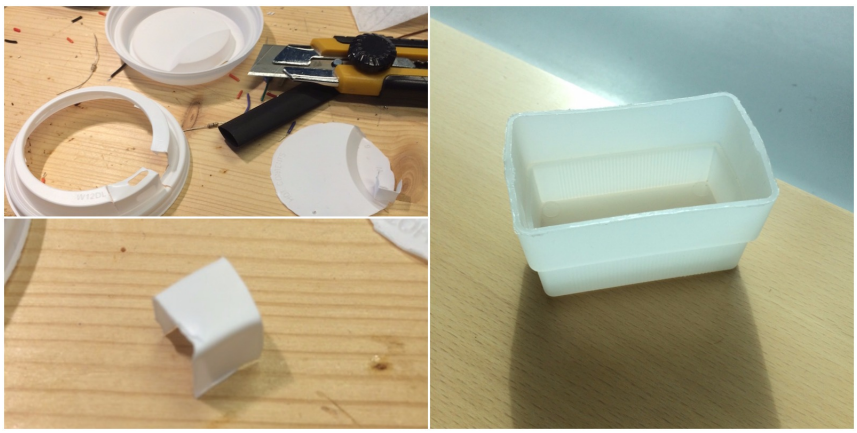
målestokken netop er gennemsnittet af lysniveauet, er det ikke garanteret at lyset fordeles konsistent over billedet. Det har, som vi kan se af resultaterne fra første datasæt, gjort det svært at fjerne baggrunden på en hensigtsmæssig måde, hvilket illustreres i figur 6.2.

Som vores tidlige eksperimenter indikerede bekræftes det i datasæt 1, at det er ukompliceret at detektere petriskåle. Såfremt der er et gennemsnitligt lysniveau på minimum 48 i billedet detekteres alle petriskålene.

På baggrund af de resultater vi har fået for brøndplader i datasæt 1 (0 ud af 45 blev detekteret), har vi forsøgt i datasæt 2 at skabe et lysmiljø, som er mindre støjende. Til det har vi monteret otte IR-dioder langs kanten af den eksperimentelle overflade (figur 6.3) og anvendt et filter, som fjerner lysbølger fra det visuelle spektrum. Vi kunne også have forsøgt at optimere algoritmen, men vi valgte at gå i den anden retning.



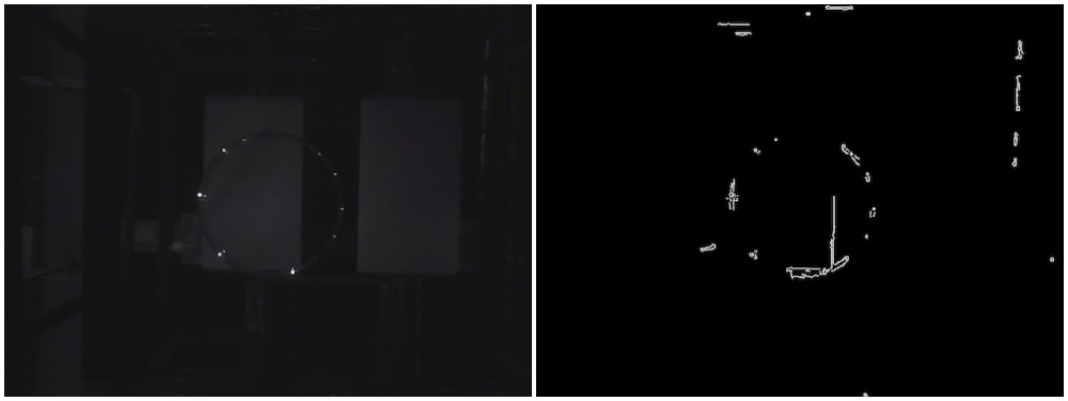
FIGUR 6.3: Kredsløb med IR-dioder under udvikling.



FIGUR 6.4: Lysfiltre vi har forsøgt os med.

I det nye lysmiljø ser vi en markant støjreduktion, hvilket gør det muligt at fratrække baggrunden med stor nøjagtighed og vi formår at detektere samtlige brøndplader. At bruge infrarødt lys er altså en fordel, når brøndpladerne skal detekteres. Overraskende nok kan vi nu ikke detektere petriskålene. Som det ses på figur 6.5 bliver petriskålene ikke tilstrækkeligt oplyst til, at Canny kan udtrække formen. Problemet her er, at diodens lys er meget koncentreret, hvilket resulterer i små kraftigt oplyste områder langs kanten af petriskålen. For at sprede lyset har vi forsøgt med forskellige mælkehvide filtre i plastik og i gummi. Det ene har vi selv lavet af et kaffelåg. Det andet er beregnet til blitz på kameraer. Andre filtre blev overvejet, men det gik op for os, at det ikke ville gøre nogen mærkbar forskel. Vi besluttede i stedet at optimere præcisionen og teste grænserne for Visionsystemet.

Inden vi går videre til at se på datasæt 3, vil vi kort diskutere årsagen til forskellen



Figur 6.5: Systemet kan ikke detektere petriskålen, da cirklen er ufuldstændig.

i oplysningen af brøndpladerne og petriskålene i datasæt 2. Vores forklaring er, at brøndpladernes komplekse struktur gør det muligt for lyset at reflektere flere gange hvilket spreder lyset og oplyser brøndpladen. Dette er ikke i samme grad muligt på i de relativt få flader i petriskålen. Som det kan ses til højre på figur 6.5, er cirklen dog ikke langt fra at tage form. En mulig måde at slutte cirklen og dermed gøre det muligt at bruge én metode til både at detektere petriskåle og brøndplader, kunne være ved at øge det infrarøde lysniveau. Det kunne man gøre ved at montere dioder i en ring hele vejen rundt langs den eksperimentelle overflade.

Om end systemet ikke fandt nogen petriskåle, fandt det samtlige brøndplader i datasæt 2. Vi har altså vist, at systemet er i stand til finde brøndplader og petriskåle uden brug af tagging.

Vi er også interesseret i at kende systemets begrænsninger, så vi kan sige noget om, hvilke forhold der er nødvendige for, at systemet fungerer som det skal. Af denne årsag, har vi lavet datasæt 3, som tester flere aspekter af systemet. Datasættet er større og testet både med og uden den lystætte kasse.

Vi har opdaget en markant forskel på det gennemsnitlige lysniveau fra datasæt 2 til datasæt 3, selvom de fysiske forhold er uændrede. Vi har anvendt samme kamera, filter og belysning, men kan alligevel dokumentere en ændring i det gennemsnitlige lysniveau fra ca. 22 i datasæt 2 til ca. 8 i datasæt 3. En forskel, som vi ikke overvejede da vi lavede testsne, er, at billederne i datasæt 3 er lavet med en Mac og i datasæt 2 er lavet med en Linux-pc. Vi mener, at de forskellige styresystemer kan være årsag til forskellen i lysniveauet. Dette har vi ikke kunne finde belæg for.

Datasæt 3 har til formål, at finde de forhold som får Visionsystemet til at bryde sammen. Vi har forsøgt os med og uden den lystætte kasse, for at se om det har kunne påvirke systemet. Det har vist sig ikke at have betydning. Det viser derimod at systemet er robust overfor lys fra det visuelle spektrum. Vi kunne også have testet systemets evne til at håndtere overeksponering af lys. Det ville med stor sandsynlighed kunne have påvirket systemet. Men uden forsøgene er det svært at sige hvordan. En vigtig pointe er, at der kræves test, med flere forsøgsopstillinger, for at kunne afdække Visionsystemets robusthed.

Et andet vigtigt fund i analysen er, at de hvide plader i det aktive lag, har stor betydning for resultaterne. Påvirkningen er dog forskellig, da kameraet, i de to datasæt, er placeret to forskellige steder. Derfor dækker de hvide plader forskellige dele af objekterne i de to datasæt. Det er svært at sige, om det er det, der har medført at datasæt 3 påvirkes mere af de hvide plader. Men en ting er klart: en højere konsistens i baggrunden ville bidrage positivt til detektionen af brøndplader.

Opsummerende er der stor forskel på vores tre datasæt. Denne forskel er et udtryk for selve processen i vores arbejde og et ønske om, først at kunne forbedre resultaterne, for derefter at kunne udfordre dem igen.

6.2 Præcision

Præcision er et vigtigt omdrejningpunkt for hele analysen. Vi har dog valgt at fokusere udelukkende på præcision som en relativ størrelse i pixler. Vores sigte i dette projekt har været at finde det system, som har kunne give os den bedste præcision. Men for at kunne diskutere præcisionen af vores system, er det interessant at se, hvordan den relaterer sig til den fysiske verden. Vi har dog ikke taget stilling til, hvad en acceptabel præcision er. Som vi var inde på i kapitel 3, kan et forsøg blive ødelagt, hvis Visionsystemet detekterer tilstrækkeligt forkert. Med nøjagtighed er det enten eller. Enten detekteres objektet, ellers gør det ikke. I sidste tilfælde vil det ødelægge forsøget. Når man ser på præcision, er grænsen lidt mere flydende. Her afhænger det af tolerancen for de objekter Visionsystemet bruges på i det enkelte forsøg. Som eksempel har de petriskåle, vi bruger i vores datasæt en radius, og dermed en tolerance, på $4,5cm$. Skal det garanteres, at der ikke bliver pipetteret væske ved siden af petriskålen, må afvigelsen altså højst være under $4,5cm$. Tolerancen for brøndplader er væsentlig mindre, da brøndenes radius er på kun $0,35cm$. Vi kan vurdere Visionsystemets præcision ved, at se på hvilke tolerancer,

det skal kunne overholde. For at gøre det er vi først nødt til at finde forholdet mellem pixler og cm. Hvis man skulle finde det præcise forhold, ville det kræve brug af to metoder, 3-punktskalibrering og Affine Transformation. Men da vi har afgrænset os fra at oversætte billedkoordinater til koordinater, som EvoBotten forstår, har vi også afgrænset os fra at lave denne kalibrering. Hvordan det gøres kommer vi ind på i kapitel 8. I stedet vil vi bruge en udregningsmetode, som ikke er helt præcis, men den kan give os et skøn. Udregningen gøres mulig ved at tage billeder af en lineal på den eksperimentelle overflade, både horisontalt og vertikalt. Overraskende er der også her en forskel imellem styresystemerne.

Tabel 6.1: Areal som billederne dækker for alle datasæt.

	Mål
Datasæt 1(Linux)	$24,2cm \times 18,5cm$
Datasæt 2(Linux)	$35cm \times 30cm$
Datasæt 3(Mac)	$37cm \times 29cm$

Alle billeder har en opløsning på $640px \times 480px$. Det giver følgende udregning for datasætne:

Tabel 6.2: px/cm horisontalt og vertikalt for alle datasæt.

	Horisontalt	Vertikalt
Datasæt 1(Linux)	$\frac{640px}{24,2cm} = 26,45px/cm$	$\frac{480px}{18,5cm} = 25,95px/cm$
Datasæt 2(Linux)	$\frac{640px}{35cm} = 18,29px/cm$	$\frac{480px}{30cm} = 16px/cm$
Datasæt 3(Mac)	$\frac{640px}{37cm} = 17,30px/cm$	$\frac{480px}{29cm} = 16,55px/cm$

Som det ses i tabel 6.2 er de vertikale og horisontale forhold ikke ens. Men vi forholder os ikke til retningen for afvigelsen i analysen. Vi er derfor nødt til at bruge én samlet målestok for px/cm for at kunne lave en konvertering.

Tabel 6.3: Samlet px/cm for alle datasæt.

	Horisontalt
Datasæt 1(Linux)	$\frac{640px+480px}{24,2cm+18,5cm} = 26,23px/cm$
Datasæt 2(Linux)	$\frac{640px+480px}{35cm+30cm} = 17,23px/cm$
Datasæt 3(Mac)	$\frac{640px+480px}{37cm+29cm} = 16,97px/cm$

På trods af at denne omregning kun kan give os et estimat, er forskellen på forholdene, vertikalt og horisontalt, lille. Tallet kan derfor bruges til at udregne et skøn for, hvilke tolerancer det nuværende Visionsystem kan understøtte. Dette ses i tabel 6.5.

Tabel 6.4: Afvigelser for alle datasæt i pixels.

	Gns. afv. (px)	Højeste afv.(px)	Laveste afv.(px)
Datasæt 1(Linux)	$3,95px$	$8,5px$	$1,5px$
Datasæt 2(Linux)	$8,09px$	$11,13px$	$5,81px$
Datasæt 3(Mac)	$3,32px$	$7,68px$	$1,39px$

Tabel 6.5: Omregnede afvigelser for alle datasæt i pixels.

	Gns. afv. (cm)	Højeste afv.(cm)	Laveste afv.(cm)
Datasæt 1(Linux)	$\frac{3,95px}{26,23px/cm} = 0,15cm$	$\frac{8,5px}{26,23px/cm} = 0,32cm$	$\frac{1,5px}{26,23px/cm} = 0,06cm$
Datasæt 2(Linux)	$\frac{8,09px}{17,23px/cm} = 0,47cm$	$\frac{11,13px}{17,23px/cm} = 0,65cm$	$\frac{5,81px}{17,23px/cm} = 0,34cm$
Datasæt 3(Mac)	$\frac{3,32px}{16,97px/cm} = 0,20cm$	$\frac{7,68px}{16,97px/cm} = 0,45cm$	$\frac{1,39px}{16,97px/cm} = 0,08cm$

Selv med forbehold for en upræcis omregningsmetode kan det ses af i tabel 6.5 for datasæt 1, at præcisionen for detektion af petriskålene holder sig inde for tolerancen, med en stor margen, i alle tilfælde. Præcisionen i datasæt 2 er for lav til at kunne holde sig inden for tolerancen på de $0,35cm$ for brøndplader, da kun en enkelt af detektionerne har en afvigelse, som er lavere. Datasæt 3 har derimod for alle detektioner en præcision, som er under $0,35cm \cdot 16,97px/cm = 5,94px$ pånær de to med den højeste afvigelse. For datasæt 3 giver det, med forbehold for at tallet kan være misvisende, en succesrate på 98,25%.

Opsummerende har vi vist at Visionsystemet lader til at have en præcision, som gør det i stand til at håndtere petriskåle. Ligeledes er det tæt på at kunne håndtere brøndpladerne, set bort fra nøjagtigheden. Afslutningsvis vil vi diskutere fejlkilderne.

6.3 Fejlkilder

Som beskrevet i metodekapitlet 3 har vores metodetilgang påvirket vores resultater. Dette er særligt gældende for datasæt 1, hvor metoden, til at kontrollere det gennemsnitlige lysniveau, kan kritiseres. Modellen tager nemlig ikke højde for, hvilken retning lyset kommer fra og om lyset er centreret eller fordelt over hele billedet. Dette vanskeliggør fratrækningen af baggrunden. Set i et større perspektiv er valg af metode og model for lysmiljøet afgørende for systemet, hvis man vil sikre konsistens.

Vores resultater viser at brugen af infrarødt lys gør det muligt at skabe et mere støjfrit testmiljø. Datasæt 2 og 3 er dog vanskelige at generalisere yderligere, da der både er forskel i lysniveauet og på kameraplaceringerne.

Mange af de problemstillinger som blevet afdækket i analysen, baserer sig på fysiske forhold. Vi har igennem test og analyse opbygget en forståelse for, hvordan disse forhold påvirker vores system. Det giver mening også at se på, om selve systemet kan forbedres. Visionsystemet har en indlejret logik, der bygger på at parametrisere objekterne som enten cirkler eller rektangler. Men som vi så på i indledningen, eksisterer der også andre metoder. Her forklarede vi om væskehåndteringsrobotten, aBioBot. Med inspiration herfra kunne man måske med fordel forsøge at detektere brøndplader med en mere detaljeret parametrisering end den vi har brugt. Hvorfor dette kunne gavne, kommer vi ind på i perspektiveringen.

Kapitel 7

Konklusion

Vi har nu præsenteret de dele, der tilsammen udgør specialet. Vores bidrag er:

- Et proof of concept Visionsystem, der er i stand til detektere petriskåle og brøndplader, ved at parametrisere dem som simple former.
- Et modul, der anvendes til at annotere Ground Truth Data på eksisterende datasæt.
- Et modul, som kan sammenligne data fra Visionsystemet med Ground Truth Data.
- Tre datasæt som vi har testet systemet med, hvilke også kan anvendes til at udbygge og teste fremtidige versioner af Visionsystemet.
- En analyse og diskussion af Visionsystemet, der samler vores erfaringer og beskriver hvordan systemet og fysiske forhold kan forbedres.

Visionsystemet knytter sig til vores første arbejdsspørgsmål: Hvilke Computer Vision metoder kan anvendes til at detektere petriskåle og brøndplader?

En petriskål parametriseres som en cirkel med et centerpunkt (x, y) og en radius. Vi har været i stand til at genkende dem med metoden Hough Circle Transform. Vi har anvendt OpenCV-bibliotekets implementation af metoden *cv2.HoughCircles()*. Af forarbejdelse har vi anvendt gråtoneskala konvertering og mediansløring.

En brøndplade parametriseres som et rektangel. Det er nødvendigt at forarbejde disse billeder mere end ved petriskålene. Der anvendes et billede med brøndplade og et uden, så baggrunden kan trækkes fra det originale billede. Begge billeder er konverteret til gråtoneskala, kontrasten bliver øget og der tilføjes bilateral sløring. Der anvendes også erodering og dilatering, der sammen med Canny og formdeskriptoren *cv2.approxPolyDP()* bearbejder konturerne i billedet. Hvis konturerne danner et rektangel, bliver brøndpladen detekteret.

Visionsystemet kan sandsynligvis forbedres. Her vil det gavne at se på aBioBot, der leder efter en mere detaljeret skabelon for en brøndplade end et simpelt rektangel. Vi mener, at en kombination af vores system med IR-dioder og denne tilgang kunne gavne EvoBotten.

Det bringer os videre til næste arbejdsspørgsmål: Hvordan kan vi teste Visionsystemet?

Ved at opstille klare regler og afgrænse hvad der ønskes testet, har vi systematisk udført en række forsøg. Vi har afgrænset os til at teste systemets robusthed overfor belysning og objektplacering. I den forlængelse har vi udformet tre forsøgsopstillinger. Det har givet tre forskellige datasæt, der dog deler de samme objektplaceringer:

- Datasæt 1 - Det visuelle spektrum, hvor det gennemsnitlige lysniveau forsøges kontrolleret med en lystæt kasse.
- Datasæt 2 - Infrarød (Linux, Ubuntu 16.04), den eksperimentelle overflade oplyses med infrarøde dioder og kameraet indeholder et infrarødt filter. Billederne er taget på en Linux-PC.
- Datasæt 3 - Infrarød (Mac, OS X 10.11), ligeledes med infrarødt filter og belysning, men billederne er taget på en Mac. Alle placeringer er taget med og uden den lystætte kasse.

Med vores annoteringsværktøj har vi udformet Ground Truth Data. Vi kan således med sammenligningsværktøjet udregne afvigelserne mellem Visionsystemets detektion og Ground Truth annoteringerne.

Disse afvigelser er grundlaget for vores analyse, hvilket tager os videre til det sidste arbejdsspørgsmål: Hvordan påvirkes Visionsystemet af lys og objektplaceringer?

Visionsystemet bliver i høj grad påvirket af filtreringen af lyset. Det ses i analysen af datasættene. Med anvendelse af infrarød belysning er systemet i stand til at detektere brøndplader mens det uden infrarød belysning, kan detektere petriskåle.

I datasæt 1 kan systemet kun detektere petriskåle. Eftersom der ikke er andre cirkulære objekter i billederne, kan petriskålene nemt afgrænses. Hertil viser analysen at systemets evne til at detektere petriskåle, påvirkes af det gennemsnitlige lysniveau i billedet. Ved et gennemsnitligt lysniveau på under 48, blev ingen petriskåle detekteret. Vores metode til at kontrollere lysniveauet kan dog kritiseres, da vi ikke forholder os til lysets retning og fordeling på billederne.

Eftersom vi ikke detekterede nogen brøndplader, lavede vi de to datasæt med infrarød belysning. Metoden fungerer til at detektere brøndplader fordi støjen formindskes i billederne. Den fungerer dog ikke til petriskålene, da lyset fra de infrarøde dioder ikke oplyser petriskålene tilstrækkeligt. Systemets præcision synes at være påvirket af rotationen på brøndpladen. Der ses i datasæt 2 en generel lavere præcision for brøndplader med en rotation på 45 grader.

Systemet har højst nøjagtighed i datasæt 2 og højst præcision i datasæt 3. Der ses et højere gennemsnitligt lysniveau i Datasæt 2 end datasæt 3. Dette kan hænge sammen med forskellen i det anvendte styresystem. Kameraet blev placeret forskelligt i de to datasæt. Det har medført, at de to hvide plader i baggrunden dækker et andet område af billederne, hvilket også kan påvirke nøjagtigheden.

Den præcision vi har arbejdet med i projektet er relativ. Men for at få en idé om hvilke tolerancer systemet kan håndtere, har vi lavet en løs estimering af px/cm. Tolerancen for petriskåle er $4,5cm$ og tolerancen for brøndplader er $0,35cm$. For petriskåle holder systemets sig inden for tolerancen i alle tilfælde. Brøndpladerne i datasæt 2 gør ikke, men i datasæt 3 holder den sig indenfor tolerancen i 112 i af de 114 detektioner. Det svarer til $98,25\%$.

Det er vores vurdering at en optimering af belysningen og konsistens i baggrunden vil gavne systemets nøjagtighed og præcision. Infrarød belysning og filtrering af lyset en klar fordel. Med den infrarøde tilgang har vi vist, at systemet er robust over for det visuelle spektrum og gør det muligt at fratrække baggrunden med minimal støj.

På baggrund af ovenstående kan vi nu besvare problemformuleringen som lyder,

Hvordan kan vi, ved hjælp af Computer Vision, skabe et fundament for at gøre EvoBotten i stand til at identificere og lokalisere petriskåle og brøndplader uden brug af tagging?

Vi kan skabe fundamentet ved at lave et proof-of-concept Visionsystem, der genkender objekter ved at parametrisere dem som simple former. Vi har vist, at man med datasættene, vores annoterings- og sammenligningsværktøj og analyse kan teste og forbedre systemets robusthed overfor lys og placering af objekter.

Kapitel 8

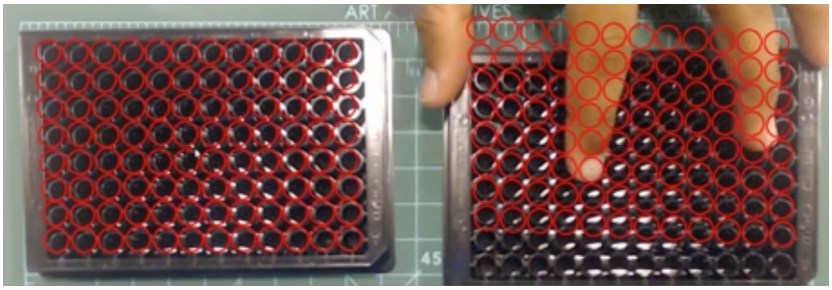
Perspektivering

I dette sidste kapitel opstiller vi en række anbefalinger til fremtidige forbedringer af Visionsystemet. På den korte bane er der to forbedringer som er nødt til at være på plads før at Visionsystemet kan integreres i EvoBottens framework:

1. Første skridt er at kunne oversætte koordinaterne fra webcameraet til koordinater som EvoBotten kan forstå. Det grundlæggende arbejde for dette er gjort og yderligere dokumentation kan findes her:
   `https://bitbucket.org/afaina/evobliss-software/wiki/Calibrate%20the%20EvoCam`

2. Andet skridt er, at teste i hvilket omfang det eksisterende system er i stand til at håndtere flere objekter i sin synsvinkel på samme tid. Vi mener, at denne funktionalitet er absolut nødvendig for kunne tage EvoBotten i brug med Visionsystemet. Disse overvejelser har vi haft med helt fra starten og systemet er også bygget til at håndtere flere objekter. Men eftersom vi ikke har testet for det, kan vi ikke sige noget om dets evne til at håndtere flere objekter på en gang.

På længere sigt skal Visionsystemet udvikles inkrementelt til at blive mere robust. Til det har vi gjort os nogle overvejelser, som kan bruges til inspiration for det videre arbejde med detektionssystemet. Måden vi selv har arbejdet med metoder og parametre for at få den bedste detektion har været udpræget eksplorativ. Man kunne godt forestille sig at arbejde videre på denne måde og forsøge sig med andre visionmetoder, som man så kunne sammenligne med de metoder, vi har brugt. F.eks. bruger aBioBot, som vi beskrev i kapitel 1, en tilgang til at finde brøndplader som kunne være interessant at se nærmere på. Ud fra deres videoer ser det ud som om de har har parametriseret de cirkulære brønde i brøndpladen og leder efter dem i stedet for den omsluttende firkant som brøndpladen danner. På denne måde kan de muligvis undgå at bruge infrarødt lys,



Figur 8.1: aBioBots detektionssystem (aBioBot, 2015a).

da cirkler er nemme at identificere. I de forsøg vi har lavet med at detektere cirkler i den størrelse, har systemet lavet mange fejldetektioner. Som det ses i figur 8.1, leder de efter en matrice af cirkler af en bestemt størrelse og form. Det kan lade sig gøre, hvis de på forhånd kalibrerer for kameraets afstand til brøndpladerne.

En anden mulighed kunne være at forbedre den eksisterende metode. Det er vores vurdering at man vil kunne detektere både petriskåle og brøndplader ved brug af et infrarødt kamera og med langt højere præcision ved at installere flere IR-dioder. På baggrund af vores forsøg er det vores vurdering at denne tilgang vil være effektiv til at styre lysmiljøet og dermed gøre systemet robust overfor støj.

En kombination af disse to tilgange kunne man også forestille sig.

Bibliografi

aBioBot (2015a). *aBioBot*. URL: https://abiobot.org/ (visited on 05/23/2016).

— (2015b). *Platform*. URL: https://abiobot.org/platform/ (visited on 05/23/2016).

Adrian, Rosebrock (2015). *Practical Python and OpenCV*. 2nd. PyImageSearch.

Bradski, G. and A. Kaehler (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media. ISBN: 978-0-596-55404-0. URL: https://books.google.dk/books?id=seAgiOfu2EIC.

CHAOSPRO. *HSL Colorspace*. URL: http://www.chaospro.de/documentation/html/paletteeditor/colorspace_hsl.htm (visited on 05/24/2016).

coldvision (2016). *Bilateral Filtering with CUDA and OpenCV 3.x| Machine Learning | Software Development*. URL: http://www.coldvision.io/2016/01/21/bilateral-filtering-with-cuda-and-opencv-3-x/ (visited on 05/31/2016).

Computerphile (2015). *Canny Edge Detector - Computerphile*. URL: https://www.youtube.com/watch?v=sRFM5IEqR2w (visited on 05/22/2016).

Deerac Fluidics (2005). *Equator GXSeries brochure*. URL: http://eng.techtum.se/Gismo/Bildarkiv/org/1111/equator_gx_brochurea4.pdf (visited on 05/23/2016).

ELFA DISTRELEC. *Buy IR-LED 850 nm 3 mm (T1), Everlight Electronics, HIR 204/H0*. URL: http://www.elfadistrelec.dk/en/ir-led-850-nm-mm-t1-everlight-electronics-hir-204-h0/p/17520349?queryFromSuggest=true (visited on 05/23/2016).

Gyldendal. *kunstigt liv | Gyldendal - Den Store Danske*. URL: http://denstoredanske.dk/Natur_og_miljÃÿ/Biologi_generelt/Begreber_m.m./kunstigt_liv (visited on 05/23/2016).

Hamilton Company. *384-Probe Head*. URL: http://www.hamiltoncompany.com/products/automated-liquid-handling/consumables/pipetting-options/384-probe-head (visited on 05/23/2016).

Hudson Robotics. *Robotic Dispenser Liquid Handling Robot - Micro10x^{TM}*. URL: http://hudsonrobotics.com/products/liquid-handling/micro10x/ (visited on 05/23/2016).

IT University of Copenhagen. *Kasper Stoy - IT University of Copenhagen*. URL: https://pure.itu.dk/portal/en/persons/kasper-stoey(2a9c0934-98ca-4101-b829-b88d77f6487e).html (visited on 05/22/2016).

java - Canny edge detector in OpenCV - Stack Overflow. URL: http://stackoverflow.com/questions/30426388/canny-edge-detector-in-opencv (visited on 05/31/2016).

Khan Academy. *Light: Electromagnetic waves, the electromagnetic spectrum and photons*. URL: http://da.khanacademy.org (visited on 05/31/2016).

Kong, Fanwei, Liang Yuan, Yuan F Zheng, and Weidong Chen (2012). "Automatic liquid handling for life science a critical review of the current state of the art". In: *Journal of laboratory automation* 17.3, pp. 169–185.

Krig, Scott (2014). *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. Apress.

Nejatimoharrami, F., A. Faina, J. Cejkova, M.M. Hanczyc, and K. Støy (2016). "EvoBot: An Open-Source, Modular Liquid Handling Robot for Nurturing Microbial Fuel Cells". In: *The Fifteenth International Conference on the Synthesis and Simulation of Living Systems (ALIFE XV)*.

OpenCV. *Introduction to OpenCV-Python Tutorials — OpenCV 3.0.0-dev documentation*. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_setup/py_intro/py_intro.html (visited on 05/27/2016).

Paris, Sylvain, Pierre Kornprobst, Jack Tumblin, Frédo Durand, and others (2009). "Foundations and Trends® in Computer Graphics and Vision". In: *Foundations and Trends® in Computer Graphics and Vision* 4.1, pp. 1–73.

Paulsen, R.R. and T.B. Moeslund (2012). *Introduction to Medical Image Analysis*. DTU Compute. ISBN: 978-87-643-0949-2. URL: https://books.google.dk/books?id=RTx3rgEACAAJ.

Qvist, Ian. *The Ramer-Douglas-Peucker Polygon Simplification Algorithm*. URL: http://ianqvist.blogspot.com/2010/05/ramer-douglas-peucker-polygon.html (visited on 05/31/2016).

'Salt and Pepper' noise reduction. URL: https://s-media-cache-ak0.pinimg.com/236x/83/7e/87/837e87b19ae77cc1b0578fe7db9bd03b.jpg (visited on 05/31/2016).

Satoshi, Suzuki and Abe Keiichi (1985). "Topological structural analysis of digital binary images by border following". In: *Computer Vision, Graphics, and Image Processing* 30.1, pp. 32–46. DOI: doi:10.1016/0734-189X(85)90016-7.

Thermo Scientific (2011). *Multidrop_Combi_nL.pdf*. URL: http://www.ninolab.se/fileadmin/Ninolab/pdf/thermolabsystems/Multidrop_Combi_nL.pdf (visited on 05/23/2016).

William Hoff (2012). *EGGN 512 - Lecture 13-1 Hough*. URL: `https://www.youtube.com/watch?v=uDB2qGqnQ1g` (visited on 05/22/2016).

Yim, Mark, Wei-Min Shen, Behnam Salemi, Daniela Rus, Mark Moll, Hod Lipson, Eric Klavins, and Gregory S Chirikjian (2007). "Modular self-reconfigurable robot systems [grand challenges of robotics]". In: *Robotics & Automation Magazine, IEEE* 14.1, pp. 43–52.

Appendiks A

Appendiks

A.1 Samlet for alle datasæt

```
avg deviation all:5.34860465116
Standard Dev all:2.64586063845
```

```
Circles avg dev: 3.95
Circles std dev: 2.04089997973
```

```
Rectangles avg dev: 5.42401960784
Rectangles std dev: 2.65373313071
```

A.2 Datasæt 1

```
Total annotations:60
Detected annotations:11
Overall avg dev:3.95
Standard dev:2.04089997973

Images:

 Filename: 001.jpg
 Lightness: 17
```

```
 Filename: 002.jpg
 Lightness: 48
```

```
  Circle
       Deviation center: 5.39
       Devation radius: 1.0
       Avg Deviation: 3.19
```

```
Filename: 003.jpg
Lightness: 78
  Circle
       Deviation center: 5.39
       Devation radius: 4.0
       Avg Deviation: 4.7
```

```
Filename: 004.jpg
Lightness: 109
  Circle
       Deviation center: 2.24
       Devation radius: 2.0
       Avg Deviation: 2.12
```

```
Filename: 005.jpg
Lightness: 138
  Circle
       Deviation center: 3.0
       Devation radius: 4.0
       Avg Deviation: 3.5
```

```
Filename: 006.jpg
Lightness: 17
```

```
Filename: 007.jpg
Lightness: 48
  Circle
       Deviation center: 8.0
       Devation radius: 9.0
       Avg Deviation: 8.5
```

```
Filename: 008.jpg
Lightness: 77
  Circle
       Deviation center: 6.0
       Devation radius: 3.0
       Avg Deviation: 4.5
```

```
Filename: 009.jpg
Lightness: 111
  Circle
       Deviation center: 7.21
       Devation radius: 7.0
       Avg Deviation: 7.11
```

```
Filename: 010.jpg
Lightness: 139
  Circle
       Deviation center: 4.47
       Devation radius: 1.0
       Avg Deviation: 2.73
```

```
Filename: 011.jpg
Lightness: 17
```

```
Filename: 012.jpg
Lightness: 38
```

```
Filename: 013.jpg
Lightness: 78
  Circle
       Deviation center: 1.0
       Devation radius: 2.0
       Avg Deviation: 1.5
```

```
Filename: 014.jpg
Lightness: 108
```

```
  Circle
      Deviation center: 3.61
      Devation radius: 2.0
      Avg Deviation: 2.8
```

```
Filename: 015.jpg
Lightness: 137
  Circle
      Deviation center: 3.61
      Devation radius: 2.0
      Avg Deviation: 2.8
```

```
Filename: 016.jpg
Lightness: 18
```

```
Filename: 017.jpg
Lightness: 48
```

```
Filename: 018.jpg
Lightness: 78
```

```
Filename: 019.jpg
Lightness: 108
```

```
Filename: 020.jpg
Lightness: 138
```

```
Filename: 021.jpg
Lightness: 18
```

```
Filename: 022.jpg
Lightness: 48
```

```
Filename: 023.jpg
Lightness: 78
```

```
Filename: 024.jpg
Lightness: 108
```

```
Filename: 025.jpg
Lightness: 138
```

```
Filename: 026.jpg
Lightness: 18
```

```
Filename: 027.jpg
Lightness: 48
```

```
Filename: 028.jpg
Lightness: 78
```

```
Filename: 029.jpg
Lightness: 108
```

```
Filename: 030.jpg
Lightness: 138
```

```
Filename: 031.jpg
Lightness: 18
```

```
Filename: 032.jpg
Lightness: 48
```

```
Filename: 033.jpg
Lightness: 78
```

```
Filename: 034.jpg
Lightness: 108
```

```
Filename: 035.jpg
Lightness: 138
```

```
Filename: 036.jpg
Lightness: 18
```

```
Filename: 037.jpg
Lightness: 47
```

```
Filename: 038.jpg
Lightness: 78
```

```
Filename: 039.jpg
Lightness: 108
```

```
Filename: 040.jpg
Lightness: 138
```

```
Filename: 041.jpg
Lightness: 18
```

```
Filename: 042.jpg
Lightness: 48
```

```
Filename: 043.jpg
Lightness: 78
```

```
Filename: 044.jpg
Lightness: 108
```

```
Filename: 045.jpg
Lightness: 138
```

```
Filename: 046.jpg
Lightness: 17
```

```
Filename: 047.jpg
Lightness: 47
```

```
Filename: 048.jpg
Lightness: 77
```

```
Filename: 049.jpg
Lightness: 108
```

```
Filename: 050.jpg
Lightness: 138
```

```
Filename: 051.jpg
Lightness: 17
```

```
Filename: 052.jpg
Lightness: 48
```

```
Filename: 053.jpg
Lightness: 77
```

```
Filename: 054.jpg
Lightness: 109
```

```
Filename: 055.jpg
Lightness: 139
```

```
Filename: 056.jpg
Lightness: 18
```

```
Filename: 057.jpg
Lightness: 47
```

```
Filename: 058.jpg
Lightness: 78
```

```
Filename: 059.jpg
Lightness: 108
```

```
 Filename: 060.jpg
 Lightness: 138
```

```
Errors during calculation:
[]
```

A.3 Datasæt 2

```
Total annotations:120
Detected annotations:90
Overall avg dev:8.08711111111
Standard dev:1.34568548781

Images:

 Filename: 001.jpg
 Lightness: 22
```

```
 Filename: 002.jpg
 Lightness: 22
```

```
 Filename: 003.jpg
 Lightness: 22
```

```
 Filename: 004.jpg
 Lightness: 22
```

```
 Filename: 005.jpg
 Lightness: 22
```

```
 Filename: 006.jpg
 Lightness: 22
```

```
 Filename: 007.jpg
 Lightness: 22
```

```
Filename: 008.jpg
Lightness: 22
```

```
Filename: 009.jpg
Lightness: 22
```

```
Filename: 010.jpg
Lightness: 22
```

```
Filename: 011.jpg
Lightness: 23
```

```
Filename: 012.jpg
Lightness: 23
```

```
Filename: 013.jpg
Lightness: 23
```

```
Filename: 014.jpg
Lightness: 23
```

```
Filename: 015.jpg
Lightness: 23
```

```
Filename: 016.jpg
Lightness: 23
```

```
Filename: 017.jpg
Lightness: 23
```

```
Filename: 018.jpg
Lightness: 23
```

```
Filename: 019.jpg
Lightness: 23
```

```
Filename: 020.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 021.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 022.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 023.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 024.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 025.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 026.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 027.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 028.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 029.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 030.jpg
Lightness: 23
-----------------------------------------------------------------
Filename: 031.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 8.94
        Deviation point 4: 8.49
        Avg Deviation: 8.49
```

```
Filename: 032.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 7.21
        Deviation point 4: 7.81
        Avg Deviation: 7.89
```

```
Filename: 033.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 6.4
        Deviation point 4: 8.06
        Avg Deviation: 7.75
```

```
Filename: 034.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 6.4
        Deviation point 4: 8.49
        Avg Deviation: 7.86
```

```
Filename: 035.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 6.4
        Deviation point 4: 7.81
        Avg Deviation: 7.69
```

```
Filename: 036.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 7.21
        Deviation point 4: 8.49
        Avg Deviation: 8.06
```

```
Filename: 037.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 7.21
        Deviation point 4: 8.06
        Avg Deviation: 7.96
```

```
Filename: 038.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 7.21
        Deviation point 3: 8.06
        Deviation point 4: 8.94
        Avg Deviation: 8.01
```

```
Filename: 039.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 6.4
        Deviation point 4: 8.49
        Avg Deviation: 7.86
```

```
Filename: 040.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.49
        Deviation point 2: 8.06
        Deviation point 3: 7.21
        Deviation point 4: 8.49
        Avg Deviation: 8.06
```

```
Filename: 041.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.32
        Deviation point 2: 5.66
        Deviation point 3: 8.6
        Deviation point 4: 5.66
        Avg Deviation: 6.56
```

```
Filename: 042.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.32
        Deviation point 2: 5.83
        Deviation point 3: 8.49
        Deviation point 4: 8.06
        Avg Deviation: 7.18
```

```
Filename: 043.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.49
        Deviation point 4: 9.06
        Avg Deviation: 7.26
```

```
Filename: 044.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.49
        Deviation point 4: 8.06
        Avg Deviation: 7.01
```

```
Filename: 045.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.49
        Deviation point 4: 5.66
        Avg Deviation: 6.41
```

```
Filename: 046.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.6
        Deviation point 4: 5.66
        Avg Deviation: 6.44
```

```
Filename: 047.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 6.32
        Deviation point 2: 5.66
        Deviation point 3: 8.49
        Deviation point 4: 5.83
        Avg Deviation: 6.57
```

```
Filename: 048.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.49
        Deviation point 4: 5.83
        Avg Deviation: 6.45
```

```
Filename: 049.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.06
        Deviation point 4: 5.83
        Avg Deviation: 6.35
```

```
Filename: 050.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 8.49
        Deviation point 4: 9.06
        Avg Deviation: 7.26
```

```
Filename: 051.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 6.4
        Deviation point 2: 7.62
        Deviation point 3: 5.83
        Deviation point 4: 9.22
        Avg Deviation: 7.27
```

```
Filename: 052.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.4
        Deviation point 2: 9.49
        Deviation point 3: 5.83
        Deviation point 4: 12.65
        Avg Deviation: 8.59
```

```
Filename: 053.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 6.4
        Deviation point 2: 5.83
        Deviation point 3: 5.83
        Deviation point 4: 12.65
        Avg Deviation: 7.68
```

```
Filename: 054.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.4
        Deviation point 2: 9.49
        Deviation point 3: 5.83
        Deviation point 4: 11.18
        Avg Deviation: 8.23
```

```
Filename: 055.jpg
Lightness: 25
  Rectangle
```

```
        Deviation point 1: 6.4
        Deviation point 2: 9.49
        Deviation point 3: 5.83
        Deviation point 4: 12.08
        Avg Deviation: 8.45
```

```
Filename: 056.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 5.66
        Deviation point 2: 8.54
        Deviation point 3: 5.83
        Deviation point 4: 12.08
        Avg Deviation: 8.03
```

```
Filename: 057.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 6.4
        Deviation point 2: 5.83
        Deviation point 3: 5.83
        Deviation point 4: 10.3
        Avg Deviation: 7.09
```

```
Filename: 058.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.4
        Deviation point 2: 8.54
        Deviation point 3: 5.83
        Deviation point 4: 12.08
        Avg Deviation: 8.22
```

```
Filename: 059.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 6.4
        Deviation point 2: 9.49
        Deviation point 3: 5.83
        Deviation point 4: 12.65
        Avg Deviation: 8.59
```

```
Filename: 060.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 6.4
        Deviation point 2: 10.2
        Deviation point 3: 5.83
        Deviation point 4: 12.21
        Avg Deviation: 8.66
```

```
Filename: 061.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 11.18
        Deviation point 3: 9.49
        Deviation point 4: 9.43
        Avg Deviation: 9.48
```

```
Filename: 062.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 7.62
        Deviation point 3: 9.49
        Deviation point 4: 9.43
        Avg Deviation: 8.59
```

```
Filename: 063.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 7.81
        Deviation point 2: 10.05
        Deviation point 3: 9.49
        Deviation point 4: 9.43
        Avg Deviation: 9.2
```

```
Filename: 064.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 9.0
        Deviation point 3: 9.49
        Deviation point 4: 9.43
        Avg Deviation: 8.93
```

```
Filename: 065.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 8.25
        Deviation point 3: 9.49
        Deviation point 4: 10.3
        Avg Deviation: 8.96
```

```
Filename: 066.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 8.25
        Deviation point 3: 9.43
        Deviation point 4: 9.43
        Avg Deviation: 8.73
```

```
Filename: 067.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 7.81
        Deviation point 2: 10.05
        Deviation point 3: 9.49
        Deviation point 4: 9.43
        Avg Deviation: 9.2
```

```
Filename: 068.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 10.05
        Deviation point 3: 9.49
        Deviation point 4: 9.43
        Avg Deviation: 9.2
```

```
Filename: 069.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 8.25
        Deviation point 3: 9.49
        Deviation point 4: 10.3
        Avg Deviation: 8.96
```

```
Filename: 070.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.81
        Deviation point 2: 10.2
        Deviation point 3: 9.43
        Deviation point 4: 9.43
        Avg Deviation: 9.22
```

```
Filename: 071.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 7.0
        Deviation point 2: 6.71
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.37
```

```
Filename: 072.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.0
        Deviation point 2: 6.4
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.29
```

```
Filename: 073.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.0
        Deviation point 2: 6.71
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.37
```

```
Filename: 074.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.0
        Deviation point 2: 6.71
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.37
```

```
Filename: 075.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 7.0
        Deviation point 2: 6.4
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.29
```

```
Filename: 076.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.28
        Deviation point 2: 5.0
        Deviation point 3: 6.71
        Deviation point 4: 4.24
        Avg Deviation: 5.81
```

```
Filename: 077.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.0
        Deviation point 2: 6.4
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.29
```

```
Filename: 078.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.0
        Deviation point 2: 6.4
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.29
```

```
Filename: 079.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 7.0
        Deviation point 2: 6.71
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.37
```

```
Filename: 080.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.0
        Deviation point 2: 6.71
        Deviation point 3: 4.47
        Deviation point 4: 7.28
        Avg Deviation: 6.37
```

```
Filename: 081.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 7.62
        Deviation point 3: 7.07
        Deviation point 4: 14.32
        Avg Deviation: 8.93
```

```
Filename: 082.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.25
        Deviation point 2: 9.22
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 10.2
```

```
Filename: 083.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 6.71
        Deviation point 2: 9.22
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 9.81
```

```
Filename: 084.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 7.62
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 9.41
```

```
Filename: 085.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 8.25
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 9.57
```

```
Filename: 086.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 8.25
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 9.57
```

```
Filename: 087.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 6.71
        Deviation point 2: 7.62
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 9.41
```

```
Filename: 088.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.25
        Deviation point 2: 9.22
        Deviation point 3: 7.07
        Deviation point 4: 14.32
        Avg Deviation: 9.71
```

```
Filename: 089.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 9.22
        Deviation point 3: 7.07
        Deviation point 4: 14.32
        Avg Deviation: 9.33
```

```
Filename: 090.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.28
        Deviation point 2: 7.62
        Deviation point 3: 9.0
        Deviation point 4: 14.32
        Avg Deviation: 9.55
```

```
Filename: 091.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 8.06
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.35
```

```
Filename: 092.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.21
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.14
```

```
Filename: 093.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.06
        Deviation point 2: 7.21
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.2
```

```
Filename: 094.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.06
        Deviation point 2: 7.21
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.2
```

```
Filename: 095.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 8.06
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.35
```

```
Filename: 096.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.21
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.14
```

```
Filename: 097.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.25
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.4
```

```
Filename: 098.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 8.06
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.35
```

```
Filename: 099.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 7.21
        Deviation point 2: 7.81
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.14
```

```
Filename: 100.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 7.21
        Deviation point 2: 8.06
        Deviation point 3: 5.0
        Deviation point 4: 8.54
        Avg Deviation: 7.2
```

```
Filename: 101.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.49
        Deviation point 4: 6.71
        Avg Deviation: 7.27
```

```
Filename: 102.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.06
        Deviation point 4: 7.07
        Avg Deviation: 7.26
```

```
Filename: 103.jpg
Lightness: 25
  Rectangle
```

```
        Deviation point 1: 9.43
        Deviation point 2: 5.39
        Deviation point 3: 8.6
        Deviation point 4: 6.71
        Avg Deviation: 7.53
```

```
Filename: 104.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.49
        Deviation point 4: 6.71
        Avg Deviation: 7.27
```

```
Filename: 105.jpg
Lightness: 25
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.6
        Deviation point 4: 6.4
        Avg Deviation: 7.23
```

```
Filename: 106.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.6
        Deviation point 4: 6.71
        Avg Deviation: 7.3
```

```
Filename: 107.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.6
        Deviation point 4: 7.07
        Avg Deviation: 7.39
```

```
Filename: 108.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.06
        Deviation point 4: 6.4
        Avg Deviation: 7.09
```

```
Filename: 109.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.47
        Deviation point 3: 8.6
        Deviation point 4: 7.07
        Avg Deviation: 7.39
```

```
Filename: 110.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 9.43
        Deviation point 2: 4.24
        Deviation point 3: 8.6
        Deviation point 4: 7.07
        Avg Deviation: 7.34
```

```
Filename: 111.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 13.34
        Deviation point 2: 7.21
        Deviation point 3: 8.06
        Deviation point 4: 14.04
        Avg Deviation: 10.66
```

```
Filename: 112.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 13.34
        Deviation point 2: 8.54
        Deviation point 3: 8.6
        Deviation point 4: 14.04
        Avg Deviation: 11.13
```

```
Filename: 113.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 13.34
        Deviation point 2: 6.4
        Deviation point 3: 7.62
        Deviation point 4: 14.04
        Avg Deviation: 10.35
```

```
Filename: 114.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 12.37
        Deviation point 2: 8.54
        Deviation point 3: 8.6
        Deviation point 4: 14.04
        Avg Deviation: 10.89
```

```
Filename: 115.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 12.37
        Deviation point 2: 6.4
        Deviation point 3: 9.43
        Deviation point 4: 14.04
        Avg Deviation: 10.56
```

```
Filename: 116.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 12.37
        Deviation point 2: 6.4
        Deviation point 3: 7.81
        Deviation point 4: 14.04
        Avg Deviation: 10.15
```

```
Filename: 117.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 12.37
        Deviation point 2: 8.54
        Deviation point 3: 8.06
        Deviation point 4: 14.04
        Avg Deviation: 10.75
```

```
Filename: 118.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 13.34
        Deviation point 2: 7.21
        Deviation point 3: 7.21
        Deviation point 4: 14.04
        Avg Deviation: 10.45
```

```
Filename: 119.jpg
Lightness: 24
  Rectangle
```

```
        Deviation point 1: 13.34
        Deviation point 2: 6.4
        Deviation point 3: 7.81
        Deviation point 4: 14.04
        Avg Deviation: 10.4
```

```
Filename: 120.jpg
Lightness: 24
  Rectangle
        Deviation point 1: 13.34
        Deviation point 2: 6.4
        Deviation point 3: 7.21
        Deviation point 4: 14.04
        Avg Deviation: 10.25
```

```
Filename: 121.jpg
Lightness: 23
```

```
Filename: 122.jpg
Lightness: 23
```

```
Filename: 123.jpg
Lightness: 23
```

```
Filename: 124.jpg
Lightness: 23
```

```
Filename: 125.jpg
Lightness: 23
```

```
Filename: 126.jpg
Lightness: 23
```

```
Filename: 127.jpg
Lightness: 23
```

```
Filename: 128.jpg
Lightness: 23
```

```
Filename: 129.jpg
Lightness: 23
```

```
Filename: 130.jpg
Lightness: 23
```

```
Filename: 131.jpg
Lightness: 21
```

```
Filename: 132.jpg
Lightness: 21
```

```
Filename: 133.jpg
Lightness: 21
```

```
Filename: 134.jpg
Lightness: 21
```

```
Filename: 135.jpg
Lightness: 21
```

```
Filename: 136.jpg
Lightness: 21
```

```
Filename: 137.jpg
Lightness: 21
```

```
Filename: 138.jpg
Lightness: 21
```

```
Filename: 139.jpg
Lightness: 22
```

```
 Filename: 140.jpg
 Lightness: 21
```

```
Errors during calculation:
[]
```

A.4 Datasæt 3

```
Total annotations:240
Detected annotations:114
Overall avg dev:3.32157894737
Standard dev:1.07383766533

Images:

 Filename: 001.jpg
 Lightness: 7
   Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 4.24
        Deviation point 3: 2.0
        Deviation point 4: 1.0
        Avg Deviation: 2.16
```

```
 Filename: 002.jpg
 Lightness: 7
   Rectangle
        Deviation point 1: 4.0
        Deviation point 2: 3.16
        Deviation point 3: 3.16
        Deviation point 4: 1.0
        Avg Deviation: 2.83
```

```
 Filename: 003.jpg
 Lightness: 7
```

```
Filename: 004.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 3.16
        Deviation point 3: 14.21
        Deviation point 4: 6.32
        Avg Deviation: 6.17
```

```
Filename: 005.jpg
Lightness: 7
```

```
Filename: 006.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 3.16
        Deviation point 3: 2.0
        Deviation point 4: 1.0
        Avg Deviation: 1.89
```

```
Filename: 007.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 3.16
        Deviation point 3: 2.0
        Deviation point 4: 2.24
        Avg Deviation: 2.41
```

```
Filename: 008.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 3.16
```

```
        Deviation point 3: 3.0
        Deviation point 4: 2.83
        Avg Deviation: 3.92
-------------------------------------------------------------

Filename: 009.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 3.16
        Deviation point 3: 2.0
        Deviation point 4: 1.0
        Avg Deviation: 1.89
-------------------------------------------------------------

Filename: 010.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 5.39
        Deviation point 3: 2.24
        Deviation point 4: 2.24
        Avg Deviation: 2.82
-------------------------------------------------------------

Filename: 011.jpg
Lightness: 8
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 4.47
        Deviation point 3: 2.0
        Deviation point 4: 2.24
        Avg Deviation: 2.53
-------------------------------------------------------------

Filename: 012.jpg
Lightness: 8
-------------------------------------------------------------

Filename: 013.jpg
Lightness: 8
```

```
Filename: 014.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 3.0
       Deviation point 2: 5.0
       Deviation point 3: 2.24
       Deviation point 4: 2.24
       Avg Deviation: 3.12
```

```
Filename: 015.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 2.24
       Deviation point 2: 3.16
       Deviation point 3: 2.0
       Deviation point 4: 1.0
       Avg Deviation: 2.1
```

```
Filename: 016.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 1.41
       Deviation point 2: 3.61
       Deviation point 3: 2.24
       Deviation point 4: 2.24
       Avg Deviation: 2.37
```

```
Filename: 017.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 1.41
       Deviation point 2: 3.16
       Deviation point 3: 2.0
       Deviation point 4: 2.24
       Avg Deviation: 2.2
```

```
Filename: 018.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 1.41
       Deviation point 2: 3.16
       Deviation point 3: 2.0
       Deviation point 4: 4.24
       Avg Deviation: 2.7
```

```
Filename: 019.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 1.41
       Deviation point 2: 5.39
       Deviation point 3: 2.0
       Deviation point 4: 2.24
       Avg Deviation: 2.76
```

```
Filename: 020.jpg
Lightness: 8
```

```
Filename: 021.jpg
Lightness: 8
```

```
Filename: 022.jpg
Lightness: 8
```

```
Filename: 023.jpg
Lightness: 8
```

```
Filename: 024.jpg
Lightness: 8
```

```
Filename: 025.jpg
Lightness: 8
```

```
Filename: 026.jpg
Lightness: 8
```

```
Filename: 027.jpg
Lightness: 8
```

```
Filename: 028.jpg
Lightness: 8
```

```
Filename: 029.jpg
Lightness: 8
```

```
Filename: 030.jpg
Lightness: 8
```

```
Filename: 031.jpg
Lightness: 8
```

```
Filename: 032.jpg
Lightness: 8
```

```
Filename: 033.jpg
Lightness: 8
```

```
Filename: 034.jpg
Lightness: 8
```

```
Filename: 035.jpg
Lightness: 8
```

```
Filename: 036.jpg
Lightness: 8
```

```
Filename: 037.jpg
Lightness: 8
```

```
Filename: 038.jpg
Lightness: 8
```

```
Filename: 039.jpg
Lightness: 8
```

```
Filename: 040.jpg
Lightness: 9
```

```
Filename: 041.jpg
Lightness: 8
```

```
Filename: 042.jpg
Lightness: 8
```

```
Filename: 043.jpg
Lightness: 8
```

```
Filename: 044.jpg
Lightness: 8
```

```
Filename: 045.jpg
Lightness: 8
```

```
Filename: 046.jpg
Lightness: 8
```

```
Filename: 047.jpg
Lightness: 8
```

```
Filename: 048.jpg
Lightness: 8
```

```
Filename: 049.jpg
Lightness: 8
```

```
Filename: 050.jpg
Lightness: 8
```

```
Filename: 051.jpg
Lightness: 8
```

```
Filename: 052.jpg
Lightness: 8
```

```
Filename: 053.jpg
Lightness: 8
```

```
Filename: 054.jpg
Lightness: 8
```

```
Filename: 055.jpg
Lightness: 8
```

```
Filename: 056.jpg
Lightness: 8
```

```
Filename: 057.jpg
Lightness: 8
```

```
Filename: 058.jpg
Lightness: 8
```

```
Filename: 059.jpg
Lightness: 8
```

```
Filename: 060.jpg
Lightness: 8
```

```
Filename: 061.jpg
Lightness: 7
```

```
Filename: 062.jpg
Lightness: 7
```

```
Filename: 063.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.0
        Deviation point 3: 4.0
        Deviation point 4: 2.0
        Avg Deviation: 2.5
```

```
Filename: 064.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.41
        Deviation point 3: 3.0
        Deviation point 4: 3.16
        Avg Deviation: 2.64
```

```
Filename: 065.jpg
Lightness: 7
```

```
Filename: 066.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 2.24
        Deviation point 3: 4.12
        Deviation point 4: 3.61
        Avg Deviation: 3.24
```

```
Filename: 067.jpg
Lightness: 7
```

```
Filename: 068.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.41
        Deviation point 3: 4.12
        Deviation point 4: 3.16
        Avg Deviation: 2.92
```

```
Filename: 069.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 2.0
        Deviation point 3: 3.16
        Deviation point 4: 2.0
        Avg Deviation: 2.54
```

```
Filename: 070.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.41
        Deviation point 3: 4.12
        Deviation point 4: 3.16
        Avg Deviation: 2.92
```

```
Filename: 071.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.41
        Deviation point 3: 3.0
        Deviation point 4: 3.61
        Avg Deviation: 2.75
```

```
Filename: 072.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 2.0
        Deviation point 3: 2.83
        Deviation point 4: 3.16
        Avg Deviation: 2.75
```

```
Filename: 073.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.41
        Deviation point 3: 4.12
        Deviation point 4: 3.61
        Avg Deviation: 3.04
```

```
Filename: 074.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 5.39
        Deviation point 3: 3.0
        Deviation point 4: 2.24
        Avg Deviation: 3.41
```

```
Filename: 075.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 2.0
        Deviation point 3: 4.12
        Deviation point 4: 3.16
        Avg Deviation: 3.07
```

```
Filename: 076.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 2.24
        Deviation point 3: 4.0
        Deviation point 4: 3.16
        Avg Deviation: 3.1
```

```
Filename: 077.jpg
Lightness: 7
```

```
Filename: 078.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 2.24
        Deviation point 3: 4.12
        Deviation point 4: 3.61
        Avg Deviation: 3.24
```

```
Filename: 079.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 1.41
        Deviation point 3: 4.12
        Deviation point 4: 3.61
        Avg Deviation: 3.04
```

```
Filename: 080.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.0
        Deviation point 2: 8.06
```

```
        Deviation point 3: 4.0
        Deviation point 4: 3.61
        Avg Deviation: 4.67
```

```
Filename: 081.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 11.7
        Deviation point 2: 14.42
        Deviation point 3: 3.16
        Deviation point 4: 1.41
        Avg Deviation: 7.68
```

```
Filename: 082.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 12.21
        Deviation point 2: 1.0
        Deviation point 3: 4.0
        Deviation point 4: 0.0
        Avg Deviation: 4.3
```

```
Filename: 083.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 2.83
        Deviation point 3: 2.24
        Deviation point 4: 0.0
        Avg Deviation: 1.62
```

```
Filename: 084.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 15.0
```

```
        Deviation point 3: 2.24
        Deviation point 4: 1.41
        Avg Deviation: 5.02
```

```
Filename: 085.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 1.0
        Deviation point 3: 3.16
        Deviation point 4: 3.61
        Avg Deviation: 2.3
```

```
Filename: 086.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 14.42
        Deviation point 3: 3.16
        Deviation point 4: 1.41
        Avg Deviation: 5.1
```

```
Filename: 087.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 2.83
        Deviation point 3: 4.0
        Deviation point 4: 1.41
        Avg Deviation: 2.41
```

```
Filename: 088.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 1.0
```

```
        Deviation point 3: 4.0
        Deviation point 4: 1.0
        Avg Deviation: 2.06
-------------------------------------------------------------

Filename: 089.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 1.0
        Deviation point 3: 4.12
        Deviation point 4: 0.0
        Avg Deviation: 1.63
-------------------------------------------------------------

Filename: 090.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 1.0
        Deviation point 3: 3.16
        Deviation point 4: 1.41
        Avg Deviation: 1.75
-------------------------------------------------------------

Filename: 091.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 2.24
        Deviation point 3: 2.83
        Deviation point 4: 1.41
        Avg Deviation: 2.18
-------------------------------------------------------------

Filename: 092.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 1.0
```

```
        Deviation point 3: 3.16
        Deviation point 4: 0.0
        Avg Deviation: 1.39
```

```
Filename: 093.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 2.24
        Deviation point 3: 3.61
        Deviation point 4: 1.0
        Avg Deviation: 2.27
```

```
Filename: 094.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 2.83
        Deviation point 3: 5.83
        Deviation point 4: 2.24
        Avg Deviation: 3.08
```

```
Filename: 095.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 1.0
        Deviation point 3: 3.16
        Deviation point 4: 3.61
        Avg Deviation: 2.5
```

```
Filename: 096.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 8.06
```

```
        Deviation point 3: 2.83
        Deviation point 4: 1.41
        Avg Deviation: 3.64
-------------------------------------------------------------

Filename: 097.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 2.24
        Deviation point 3: 3.16
        Deviation point 4: 1.41
        Avg Deviation: 2.26
-------------------------------------------------------------

Filename: 098.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 2.24
        Deviation point 3: 5.1
        Deviation point 4: 2.24
        Avg Deviation: 2.95
-------------------------------------------------------------

Filename: 099.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 2.24
        Deviation point 3: 4.0
        Deviation point 4: 2.24
        Avg Deviation: 2.68
-------------------------------------------------------------

Filename: 100.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 1.0
```

```
        Deviation point 3: 4.12
        Deviation point 4: 2.24
        Avg Deviation: 2.4
```

```
Filename: 101.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 5.1
        Deviation point 3: 3.16
        Deviation point 4: 5.0
        Avg Deviation: 3.67
```

```
Filename: 102.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 3.16
        Deviation point 3: 4.24
        Deviation point 4: 4.12
        Avg Deviation: 3.24
```

```
Filename: 103.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 5.1
        Deviation point 3: 4.24
        Deviation point 4: 4.12
        Avg Deviation: 3.72
```

```
Filename: 104.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 6.32
```

```
        Deviation point 3: 3.61
        Deviation point 4: 4.12
        Avg Deviation: 4.63
```

```
Filename: 105.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 3.16
        Deviation point 3: 4.24
        Deviation point 4: 5.1
        Avg Deviation: 3.48
```

```
Filename: 106.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 4.12
        Deviation point 3: 3.16
        Deviation point 4: 4.0
        Avg Deviation: 3.17
```

```
Filename: 107.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 6.08
        Deviation point 3: 3.61
        Deviation point 4: 5.1
        Avg Deviation: 4.05
```

```
Filename: 108.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 6.32
```

```
        Deviation point 3: 3.61
        Deviation point 4: 5.1
        Avg Deviation: 4.32
```

```
Filename: 109.jpg
Lightness: 7
```

```
Filename: 110.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 5.1
        Deviation point 3: 3.61
        Deviation point 4: 4.12
        Avg Deviation: 4.11
```

```
Filename: 111.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 4.12
        Deviation point 3: 3.61
        Deviation point 4: 5.1
        Avg Deviation: 4.32
```

```
Filename: 112.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 5.1
        Deviation point 3: 3.61
        Deviation point 4: 4.12
        Avg Deviation: 3.56
```

```
Filename: 113.jpg
Lightness: 7
```

```
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 5.1
        Deviation point 3: 3.61
        Deviation point 4: 5.1
        Avg Deviation: 4.35
```

```
Filename: 114.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 5.1
        Deviation point 3: 2.0
        Deviation point 4: 5.1
        Avg Deviation: 3.3
```

```
Filename: 115.jpg
Lightness: 7
```

```
Filename: 116.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 6.71
        Deviation point 3: 4.24
        Deviation point 4: 4.12
        Avg Deviation: 4.12
```

```
Filename: 117.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 5.1
        Deviation point 3: 4.24
        Deviation point 4: 3.0
        Avg Deviation: 4.2
```

```
Filename: 118.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.24
        Deviation point 2: 5.1
        Deviation point 3: 2.24
        Deviation point 4: 5.1
        Avg Deviation: 3.67
```

```
Filename: 119.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 5.1
        Deviation point 3: 4.24
        Deviation point 4: 4.12
        Avg Deviation: 3.72
```

```
Filename: 120.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 5.1
        Deviation point 3: 3.61
        Deviation point 4: 3.0
        Avg Deviation: 3.28
```

```
Filename: 121.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.0
        Deviation point 2: 4.12
        Deviation point 3: 2.24
        Deviation point 4: 1.0
        Avg Deviation: 2.34
```

```
Filename: 122.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.0
        Deviation point 2: 4.12
        Deviation point 3: 1.0
        Deviation point 4: 1.0
        Avg Deviation: 2.03
```

```
Filename: 123.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.0
        Deviation point 2: 3.61
        Deviation point 3: 1.0
        Deviation point 4: 1.41
        Avg Deviation: 2.0
```

```
Filename: 124.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 9.85
        Deviation point 2: 3.61
        Deviation point 3: 2.24
        Deviation point 4: 1.41
        Avg Deviation: 4.28
```

```
Filename: 125.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.0
        Deviation point 2: 5.1
        Deviation point 3: 2.24
        Deviation point 4: 1.0
        Avg Deviation: 2.58
```

```
Filename: 126.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 5.1
        Deviation point 3: 1.41
        Deviation point 4: 1.0
        Avg Deviation: 2.13
```

```
Filename: 127.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 5.0
        Deviation point 2: 6.0
        Deviation point 3: 2.83
        Deviation point 4: 1.0
        Avg Deviation: 3.71
```

```
Filename: 128.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 3.61
        Deviation point 3: 1.0
        Deviation point 4: 1.0
        Avg Deviation: 1.65
```

```
Filename: 129.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 4.12
        Deviation point 3: 1.0
        Deviation point 4: 2.0
        Avg Deviation: 2.03
```

```
Filename: 130.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 2.0
        Deviation point 2: 4.47
        Deviation point 3: 1.0
        Deviation point 4: 1.41
        Avg Deviation: 2.22
```

```
Filename: 131.jpg
Lightness: 8
  Rectangle
        Deviation point 1: 9.06
        Deviation point 2: 4.12
        Deviation point 3: 1.0
        Deviation point 4: 1.41
        Avg Deviation: 3.9
```

```
Filename: 132.jpg
Lightness: 8
  Rectangle
        Deviation point 1: 4.12
        Deviation point 2: 2.83
        Deviation point 3: 2.24
        Deviation point 4: 10.3
        Avg Deviation: 4.87
```

```
Filename: 133.jpg
Lightness: 8
  Rectangle
        Deviation point 1: 2.0
        Deviation point 2: 5.1
        Deviation point 3: 3.61
        Deviation point 4: 1.41
        Avg Deviation: 3.03
```

```
Filename: 134.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 2.0
       Deviation point 2: 4.12
       Deviation point 3: 2.0
       Deviation point 4: 1.0
       Avg Deviation: 2.28
```

```
Filename: 135.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 2.0
       Deviation point 2: 6.0
       Deviation point 3: 1.0
       Deviation point 4: 1.0
       Avg Deviation: 2.5
```

```
Filename: 136.jpg
Lightness: 8
```

```
Filename: 137.jpg
Lightness: 8
  Rectangle
       Deviation point 1: 1.0
       Deviation point 2: 3.61
       Deviation point 3: 2.24
       Deviation point 4: 5.0
       Avg Deviation: 2.96
```

```
Filename: 138.jpg
Lightness: 7
  Rectangle
       Deviation point 1: 2.0
       Deviation point 2: 3.61
```

```
        Deviation point 3: 1.41
        Deviation point 4: 1.41
        Avg Deviation: 2.11
```

```
Filename: 139.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 8.25
        Deviation point 2: 4.12
        Deviation point 3: 1.0
        Deviation point 4: 1.0
        Avg Deviation: 3.59
```

```
Filename: 140.jpg
Lightness: 8
  Rectangle
        Deviation point 1: 1.41
        Deviation point 2: 3.61
        Deviation point 3: 1.0
        Deviation point 4: 4.12
        Avg Deviation: 2.54
```

```
Filename: 141.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 2.24
        Deviation point 3: 3.61
        Deviation point 4: 3.16
        Avg Deviation: 2.5
```

```
Filename: 142.jpg
Lightness: 7
```

```
Filename: 143.jpg
Lightness: 7
```

```
Filename: 144.jpg
Lightness: 7
```

```
Filename: 145.jpg
Lightness: 7
```

```
Filename: 146.jpg
Lightness: 7
```

```
Filename: 147.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 1.41
        Deviation point 3: 4.0
        Deviation point 4: 5.39
        Avg Deviation: 2.95
```

```
Filename: 148.jpg
Lightness: 7
```

```
Filename: 149.jpg
Lightness: 7
```

```
Filename: 150.jpg
Lightness: 7
```

```
Filename: 151.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 0.0
        Deviation point 2: 2.83
        Deviation point 3: 3.61
        Deviation point 4: 5.1
        Avg Deviation: 2.88
```

```
Filename: 152.jpg
Lightness: 7
  Rectangle
       Deviation point 1: 0.0
       Deviation point 2: 2.83
       Deviation point 3: 3.61
       Deviation point 4: 4.12
       Avg Deviation: 2.64
```

```
Filename: 153.jpg
Lightness: 7
```

```
Filename: 154.jpg
Lightness: 7
```

```
Filename: 155.jpg
Lightness: 7
```

```
Filename: 156.jpg
Lightness: 7
```

```
Filename: 157.jpg
Lightness: 7
  Rectangle
       Deviation point 1: 1.41
       Deviation point 2: 1.0
       Deviation point 3: 4.12
       Deviation point 4: 6.32
       Avg Deviation: 3.22
```

```
Filename: 158.jpg
Lightness: 7
  Rectangle
       Deviation point 1: 1.0
       Deviation point 2: 2.83
```

```
        Deviation point 3: 3.61
        Deviation point 4: 4.12
        Avg Deviation: 2.89
```

```
Filename: 159.jpg
Lightness: 8
```

```
Filename: 160.jpg
Lightness: 8
  Rectangle
        Deviation point 1: 1.0
        Deviation point 2: 2.83
        Deviation point 3: 4.12
        Deviation point 4: 5.39
        Avg Deviation: 3.33
```

```
Filename: 161.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 4.12
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.04
```

```
Filename: 162.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 5.66
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.42
```

```
Filename: 163.jpg
Lightness: 7
```

```
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 1.0
        Deviation point 3: 1.41
        Deviation point 4: 8.0
        Avg Deviation: 4.0
```

```
Filename: 164.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 1.0
        Deviation point 3: 2.83
        Deviation point 4: 7.0
        Avg Deviation: 4.38
```

```
Filename: 165.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 5.0
        Deviation point 2: 3.61
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.25
```

```
Filename: 166.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 3.61
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.12
```

```
Filename: 167.jpg
Lightness: 7
```

```
  Rectangle
        Deviation point 1: 5.39
        Deviation point 2: 5.0
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.7
```

```
Filename: 168.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 5.66
        Deviation point 3: 1.41
        Deviation point 4: 8.0
        Avg Deviation: 5.23
```

```
Filename: 169.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 5.83
        Deviation point 2: 3.61
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.46
```

```
Filename: 170.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 5.39
        Deviation point 2: 3.61
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.35
```

```
Filename: 171.jpg
Lightness: 7
```

```
  Rectangle
        Deviation point 1: 6.71
        Deviation point 2: 5.66
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 5.19
```

```
Filename: 172.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 6.4
        Deviation point 3: 1.41
        Deviation point 4: 6.0
        Avg Deviation: 4.36
```

```
Filename: 173.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 5.0
        Deviation point 2: 3.61
        Deviation point 3: 2.83
        Deviation point 4: 7.0
        Avg Deviation: 4.61
```

```
Filename: 174.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 6.4
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.82
```

```
Filename: 175.jpg
Lightness: 7
```

```
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 4.24
        Deviation point 3: 1.41
        Deviation point 4: 8.06
        Avg Deviation: 4.55
```

```
Filename: 176.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 4.24
        Deviation point 3: 1.41
        Deviation point 4: 7.0
        Avg Deviation: 4.07
```

```
Filename: 177.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 4.47
        Deviation point 2: 4.24
        Deviation point 3: 2.24
        Deviation point 4: 7.07
        Avg Deviation: 4.51
```

```
Filename: 178.jpg
Lightness: 7
  Rectangle
        Deviation point 1: 3.61
        Deviation point 2: 3.61
        Deviation point 3: 2.24
        Deviation point 4: 7.07
        Avg Deviation: 4.13
```

```
Filename: 179.jpg
Lightness: 7
```

```
  Rectangle
      Deviation point 1: 3.61
      Deviation point 2: 4.24
      Deviation point 3: 1.41
      Deviation point 4: 8.06
      Avg Deviation: 4.33
```

```
Filename: 180.jpg
Lightness: 7
  Rectangle
      Deviation point 1: 4.47
      Deviation point 2: 6.4
      Deviation point 3: 1.41
      Deviation point 4: 8.06
      Avg Deviation: 5.09
```

```
Filename: 181.jpg
Lightness: 6
```

```
Filename: 182.jpg
Lightness: 6
```

```
Filename: 183.jpg
Lightness: 6
```

```
Filename: 184.jpg
Lightness: 6
```

```
Filename: 185.jpg
Lightness: 6
```

```
Filename: 186.jpg
Lightness: 6
```

```
Filename: 187.jpg
Lightness: 6
```

Filename: 188.jpg
Lightness: 6

Filename: 189.jpg
Lightness: 6

Filename: 190.jpg
Lightness: 6

Filename: 191.jpg
Lightness: 7

Filename: 192.jpg
Lightness: 7

Filename: 193.jpg
Lightness: 6

Filename: 194.jpg
Lightness: 7

Filename: 195.jpg
Lightness: 6

Filename: 196.jpg
Lightness: 6

Filename: 197.jpg
Lightness: 6

Filename: 198.jpg
Lightness: 7

Filename: 199.jpg
Lightness: 6

```
Filename: 200.jpg
Lightness: 6
```

```
Filename: 201.jpg
Lightness: 6
```

```
Filename: 202.jpg
Lightness: 6
```

```
Filename: 203.jpg
Lightness: 6
```

```
Filename: 204.jpg
Lightness: 6
```

```
Filename: 205.jpg
Lightness: 6
```

```
Filename: 206.jpg
Lightness: 6
```

```
Filename: 207.jpg
Lightness: 6
```

```
Filename: 208.jpg
Lightness: 6
```

```
Filename: 209.jpg
Lightness: 6
```

```
Filename: 210.jpg
Lightness: 6
```

```
Filename: 211.jpg
Lightness: 7
```

```
Filename: 212.jpg
Lightness: 6
```

```
Filename: 213.jpg
Lightness: 7
```

```
Filename: 214.jpg
Lightness: 6
```

```
Filename: 215.jpg
Lightness: 6
```

```
Filename: 216.jpg
Lightness: 7
```

```
Filename: 217.jpg
Lightness: 7
```

```
Filename: 218.jpg
Lightness: 7
```

```
Filename: 219.jpg
Lightness: 7
```

```
Filename: 220.jpg
Lightness: 6
```

```
Filename: 221.jpg
Lightness: 6
```

```
Filename: 222.jpg
Lightness: 6
```

```
Filename: 223.jpg
Lightness: 6
```

```
Filename: 224.jpg
Lightness: 6
```

```
Filename: 225.jpg
Lightness: 6
```

```
Filename: 226.jpg
Lightness: 6
```

```
Filename: 227.jpg
Lightness: 6
```

```
Filename: 228.jpg
Lightness: 6
```

```
Filename: 229.jpg
Lightness: 6
```

```
Filename: 230.jpg
Lightness: 6
```

```
Filename: 231.jpg
Lightness: 6
```

```
Filename: 232.jpg
Lightness: 6
```

```
Filename: 233.jpg
Lightness: 6
```

```
Filename: 234.jpg
Lightness: 7
```

```
Filename: 235.jpg
Lightness: 7
```

```
Filename: 236.jpg
Lightness: 6
```

```
Filename: 237.jpg
Lightness: 6
```

```
Filename: 238.jpg
Lightness: 6
```

```
Filename: 239.jpg
Lightness: 6
```

```
Filename: 240.jpg
Lightness: 7
```

```
Errors during calculation:
[]
```