BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

ENHANCING RELIABILITY AND BENCHMARKING PERFORMANCE OF AGAR PLATE HANDLING ALGORITHMS FOR LABORATORY AUTOMATION ROBOTS

ZVYŠOVÁNÍ A MĚŘENÍ SPOLEHLIVOSTI ALGORITMŮ PRO MANIPULACI S AGAROVÝMI MISKAMI V LABORATORNÍ AUTOMATIZACI

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTOR
AUTOR PRÁCE

Tereza Kalivodová

ADVISOR
VEDOUCÍ PRÁCE

Ing. Martin Mézl, Ph.D.

BRNO 2024

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Bachelor's Thesis

Bachelor's study program **Biomedical Technology and Bioinformatics**

Department of Biomedical Engineering

Student: Tereza Kalivodová ***ID:*** 240516

Year of study: 3 ***Academic year:*** 2023/24

TITLE OF THESIS:

Enhancing Reliability and Benchmarking Performance of Agar Plate Handling Algorithms for Laboratory Automation Robots

INSTRUCTION:

1) Study the topic of picking and transferring microbial colonies cultured on Petri dishes, focusing on robotic solutions. 2) Familiarize yourself with the prototype of an automated colony picking robot, the MBT Pathfinder, and its use in microbiological laboratories. Present the fundamental algorithms applied to image data from the instrument’s cameras during the processing and analysis of agar plates. 3) Acquire appropriate datasets using the above-mentioned instrument. Utilize the obtained data to measure the reliability of selected algorithms applied to the data during the run of the instrument. 4) Modify configurations of the algorithms to enhance reliability or propose new solutions without altering the acquisition of image data. 5) Repeat reliability measurements with the proposed modifications. 6)Interpret and discuss the achieved results. Compare them, if possible, with the original implementations.

RECOMMENDED LITERATURE:

[1]M. Čičatka, R. Burget and J. Karásek, "Machine-learning Approach to Microbial Colony Localisation," 2022 45th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 2022, pp. 206-211, doi: 10.1109/TSP55681.2022.9851236.

[2] Andreini, P., Bonechi, S., Bianchini, M., Mecocci, A., & Scarselli, F. (2020). Image generation by GAN and style transfer for agar plate image segmentation. In Computer Methods and Programs in Biomedicine (Vol. 184, p. 105268). Elsevier BV. https://doi.org/10.1016/j.cmpb.2019.105268

Date of project specification: 5.2.2024 ***Deadline for submission:*** 29.5.2024

Supervisor: Ing. Martin Mézl, Ph.D.
Consultant: Ing. Michal Čičatka

doc. Ing. Jana Kolářová, Ph.D.
Chair of study program board

WARNING:

The author of the Bachelor's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRACT

This bachelor thesis investigates the issue of sample preparation in the field of microbiology and medical diagnostics with an emphasis on the automated robotic system MBT Pathfinder, developed by Bruker Daltonics GmbH & Co. KG. Using digital imaging techniques and convolutional neural networks, the thesis focuses on improving the algorithm for identifying the position of microbial colonies in the MBT Pathfinder system. The practical part of the thesis presents innovative approaches to optimize critical sample preparation steps to eliminate errors and increase process efficiency. The results of this work can enhance the reliability of microbiological analyses in medical diagnostics and microbiological research.

KEYWORDS

petri dish, agar plate, microbial colonies, maldi target, colony picking, image processing, convolutional neural network, laboratory automation

ABSTRAKT

Tato bakalářská práce zkoumá problematiku vzorkové přípravy v oblasti mikrobiologie a lékařské diagnostiky s důrazem na automatizovaný robotický systém MBT Pathfinder, vyvinutý firmou Bruker Daltonics GmbH & Co. KG. S využitím digitálních obrazových technik a konvolučních neuronových sítí se práce zaměřuje na zdokonalení algoritmu pro identifikaci pozice mikrobiálních kolonií v systému MBT Pathfinder. Praktická část práce prezentuje inovativní přístupy k optimalizaci kritických kroků vzorkové přípravy s cílem eliminovat chyby a zvýšit efektivitu procesu. Výsledky této práce mohou posílit spolehlivost mikrobiologických analýz v oblasti lékařské diagnostiky a mikrobiologického výzkumu.

KLÍČOVÁ SLOVA

petriho miska, agarová destička, mikrobiální kolonie, maldi terčík, výběr kolonií, zpracování obrazu, konvoluční neuronová síť, laboratorní automatizace

Typeset by the thesis package, version 4.07; http://latex.feec.vutbr.cz

ROZŠÍŘENÝ ABSTRAKT

Tato bakalářská práce se zabývá problematikou automatizace procesu náběru mikrobiálních kolonií. V oblasti mikrobiologie a lékařské diagnostiky je přesnost a spolehlivost přípravy vzorků klíčová. Historicky byl tento proces závislý na manuálních postupech vyžadujících vysoce kvalifikovaný personál. Výběr kolonií mikroorganismů a jejich příprava pro další analýzu byly časově náročné a vzhledem k opakovatelnosti procesu často náchylné k lidské chybě. V reakci na tuto potřebu byly vyvinuty automatizované robotické přístroje, které minimalizují potřebu kvalifikovaného personálu.

Veškeré experimenty v této práci jsou prováděny na zařízení MBT Pathfinder společnosti Bruker Daltonics GmbH & Co. KG. Tento přístroj představuje robotizovaný systém, který zajišťuje automatický přenos mikrobiálních kolonií na MALDI destičku a přípravu vzorku pro následnou analýzu pomocí hmotnostní spektrometrie.

Pracovní postup tohoto zařízení začíná vložením Petriho misky do přístroje. To může být provedeno manuálně, ale častěji je využíváno zařízení Feeder pro automatické vkládání misek. Miska je následně vyfocena a vrácena zpět do přístroje Feeder. Následně si uživatel na pořízených snímcích misek vybere kolonie, které chce nabrat. Kolonie lze vybrat buď manuálně, nebo je možný předvýběr pomocí algoritmu využívajícího zpracování obrazu a segmentace. Misky jsou poté vráceny do zařízení a daná kolonie je nabrána pomocí přenosové smyčky a nanesena na MALDI destičku.

Je zde několik kritických bodů, kde může dojít k selhání přístroje. Prvním z nich je algoritmus pro detekci misky. Pokud miska chybí, ale algoritmus vyhodnotí její přítomnost, může dojít ke zničení přenosové smyčky. V opačném případě, pokud by do zařízení byla vložena miska, ale zařízení by vyhodnotilo, že tam miska není, analýza misky by nebyla možná. Dalším problémem je skutečnost, že při opětovném vložení misky do přístroje může dojít k rotaci misky – miska nebude do zařízení vložena pod stejným úhlem jako poprvé. To může vést k nabrání a přenosu nesprávné kolonie. Posledním kritickým bodem pracovního postupu přístroje je výpočet správné pozice vybrané kolonie. V praktické části této práce jsou jednotlivé algoritmy upravovány a vylepšovány.

Nejprve jsem si vytvořila vlastní testovací datasety, jeden pro hodnocení přesnosti detekce misek a druhý pro hodnocení rotace misek a výkonu vyhledávání polohy kolonií. Testovací soubor dat pro detekci misek se skládá ze snímků pořízených pomocí přístroje MBT Pathfinder. Obsahuje 30 snímků bez misek a 26 snímků s miskami. Tyto výběry byly pečlivě vybrány z celkem 2759 vzorků, čímž bylo zajištěno zastoupení různých typů agarů, variací mikroorganismů, misek bez mikroorganismů a misek zachycených za různých podmínek osvětlení.Testovací dataset pro

hodnocení rotace misek a výkonu vyhledávání polohy kolonií byl vytvořen výběrem 40 vzorků z 3654 dostupných. Tyto vzorky představují jak standardní vzorky zpracovávané přístrojem, tak několik vzorků, které dříve představovaly problém. Tento soubor zkušebních dat byl navržen tak, aby zahrnoval širokou škálu chování, které se může vyskytnout při zpracování vzorků přístrojem MBT Pathfinder. Každý vzorek obsahoval 3 obrázky – **"Before" obrázek** = obrázek pořízený po prvním vložení misky do zařízení, **"After" obrázek** = obrázek pořízený po opětovném vsunutí misky do zařízení a **"Centered" obrázek** = **after obrázek** s potenciálně správnou kolonií uprostřed – z důvodu distorze kamery rybím okem může být posun misky nepřesný, tedy vybraná kolonie nemusí být zcela ve středu.

Detekce misky byla původně řešena pomocí složité konvoluční neuronové sítě. Zjednodušení architektury a úprava parametrů CNN, zmenšení a ořezání vstupních obrázků a rozšíření datasetu o misky s různými barvami agarů a s různým osvitem vedly ke zlepšení přesnosti z **0.982** na **1.0** a ke zkrácení času detekce z **94.82 ms** na **54.69 ms**.

Algoritmus pro výpočet úhlu rotace misky byl řešen pomocí **before** a **after obrázků**. Tyto snímky byly převedeny na šedotónové a následně na binární s využitím Canny detektoru. Poté byl pomocí metody template matching nalezen úhel rotace **before obrázku**, pro který měly **after obrázek** a orotovaný **before obrázek** největší korelaci. Tento úhel byl považován za úhel rotace. K mírnému zlepšení přesnosti došlo doplněním algoritmu o podmínky pro úpravu parametrů funkce Canny na základě jasu obrazu. Při opětovném vložení misky do zařízení totiž nedochází pouze k rotačním změnám, ale i k mírným translacím. Také se zde projevuje efekt rybího oka kamery. Z tohoto důvodu jsem se rozhodla použít metodu registrace obrazu. S využitím techniky RANSAC a detektoru ORB jsem dosáhla značného zlepšení přesnosti z **0.75** na **0.975**.

Nalezení správné pozice vybrané kolonie využívá výpočtu nových vycentrovaných souřadnic technikou template matching. Pro zlepšení tohoto algoritmu jsem přidala další krok pro zpřesnění orotovaných souřadnic. Před samotným výpočtem posunu mezi **after** a **centered obrázkem** jsou vypočítány nové orotované souřadnice. Využívá se opět metody template matching. K dalšímu vylepšení došlo po změně velikosti masek. Zvětšení masek vedlo ke zvýšení množství informace v obrazu, ale zároveň ke zvýšení výpočetní náročnosti. Optimální velikost masek jsem zvolila jako kompromis mezi těmito dvěma faktory. To vedlo ke zlepšení přesnosti z **0.725** na **0.95**. Je však důležité zmínit, že testovací dataset obsahoval 2 vzorky, které byly nesprávně analyzovány, pravděpodobně v důsledku selhání původního algoritmu. Po odstranění těchto vzorků byla konečná přesnost algoritmu na testovacím souboru dat **1.0**.

Následně jsem provedla experiment, který ověřuje, zda může algoritmus pro nalezení správné pozice vybrané kolonie kompenzovat chybu předchozího algoritmu pro výpočet rotace. Umělé zavedení chyby -1° tuto hypotézu potvrdilo.

Celkově tato práce představuje pokrok směrem ke zlepšení spolehlivosti a účinnosti přístroje MBT Pathfinder při mikrobiologické analýze. Optimalizace a zdokonalování algoritmů významně ovlivňují celkovou výkonnost systému, a věřím, že předložená řešení mohou přinést výhody a pokrok v praktických aplikacích.

Author's Declaration

Author:	Tereza Kalivodová
Author's ID:	240516
Paper type:	Bachelor's Thesis
Academic year:	2023/24
Topic:	Enhancing Reliability and Benchmarking Performance of Agar Plate Handling Algorithms for Laboratory Automation Robots

I declare that I have written this paper independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the paper and listed in the comprehensive bibliography at the end of the paper.

As the author, I furthermore declare that, with respect to the creation of this paper, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll. of the Czech Republic, Section 2, Head VI, Part 4.

Brno . . . .

author's signature*

*The author signs only in the printed version.

ACKNOWLEDGEMENT

I would like to express my gratitude to my thesis supervisor, Ing. Martin Mézl, Ph.D. for his time, consultations, and stimulating response to my thesis. I would also like to thank Ing. Michal Čičatka for his valuable comments, suggestions, patience, and constant support. In addition, I extend my thanks to Ing. Jan Láncoš for constructive feedback and helpful suggestions.

Contents

List of Figures

List of Tables

Introduction

In the field of microbiology and medical diagnostics, the accuracy and reliability of sample preparation is crucial. Historically, this process has been dependent on manual procedures requiring highly skilled personnel. The selection of colonies of microorganisms and their preparation for further analysis has been time-consuming and often prone to human error due to the repetitiveness of the process. In response to this need, automated robotic instruments have been developed to eliminate or at least minimize the need for skilled personnel.

The MBT Pathfinder is an automated robot developed and manufactured by Bruker Daltonics GmbH & Co. KG. It is specialized in preparing microbial colonies for subsequent analysis by mass spectrometry. A key feature of this device is its ability to precisely identify the positions of microbial colonies on the Petri dishes (agar plate) for optimal selection and transfer to MALDI target plates.

This thesis aims to improve the robustness and reliability of the algorithm responsible for determining the correct microbial colony positions in the agar plate images acquired by MBT Pathfinder. It focuses on eliminating errors during the act of colony selection and presents innovative approaches to address each critical step in sample preparation.

The introductory part of the thesis focuses on the issue of image digitization, including basic image types and image processing methods. The discussion further develops the topic of the basics of convolutional neural networks, which represent a key component in the solution of one of the instrument's tasks. The theoretical part also covers an analysis of microbial colonies, including the processes of cultivation, propagation, and growth. Automation of the colony selection method is another important topic discussed here. For a better understanding of the practical part of the work, critical areas of the MBT Pathfinder instrument are thoroughly described.

The practical part of the thesis discusses different approaches to improve robustness during the critical steps in sample preparation, highlighting innovative solutions and optimizations. Finally, individual improvements, results, and their comparison with the original solution are presented.

1 Relevant algorithms and techniques

This chapter delves into relevant algorithms and techniques crucial for addressing the objectives of my thesis. It explores fundamental concepts in digital image processing as well as advanced methodologies such as convolutional neural networks (CNNs).

1.1 Digital image processing

Digital image processing is a technology involving the use of a digital computer and algorithms to manipulate digital images. It is carried out to extract additional information present in the image data.

Image pre-processing is a fundamental stage in digital image processing, aimed at optimizing and enhancing the information extracted from the image before further analysis. This process involves various operations such as noise removal, highlighting key features, or contrast adjustments [18, 19, 23].

1.1.1 Digital image

Digital images are 2D or 3D matrices, where each value in the matrix represents an amplitude that determines the light or color properties of the corresponding pixels. Such images are defined as a finite, discrete representation of the original continuous image [21].

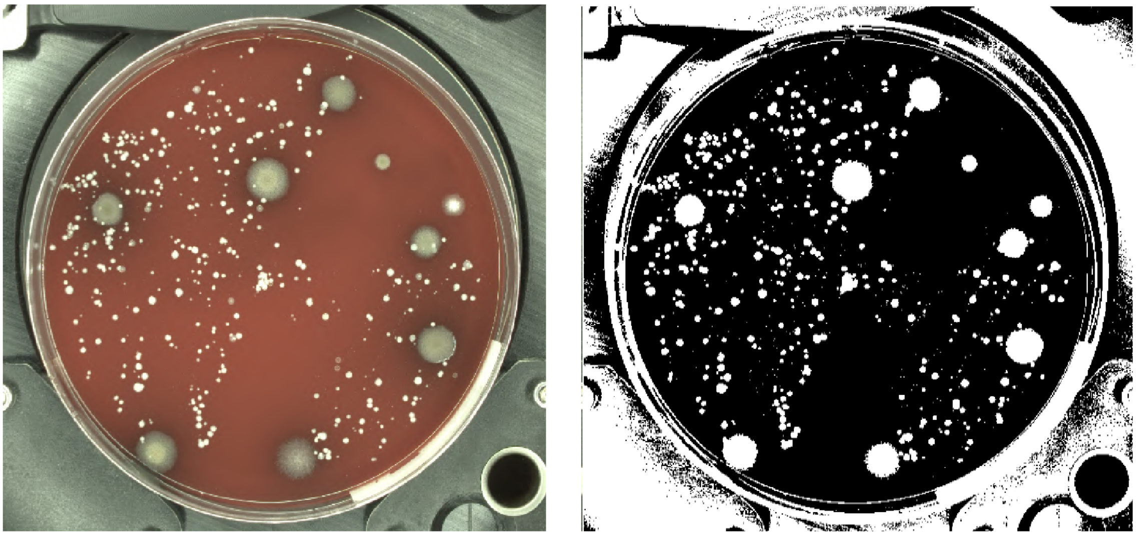
There are many types of digital images. The ones relevant to this thesis are:

- **Binary images**
- **Gray-scale images**
- **Color images**

Binary images

A binary image consists of only two pixel values, 0 (black) and 1 (white). Such images are termed 1-bit images because one pixel is represented by only one binary digit.

The binary image is mainly used to determine the shape, outline, and edges of an image, commonly used in image e.g. image segmentation or registration. An example of a binary image is illustrated below 1.1 [22, 21].



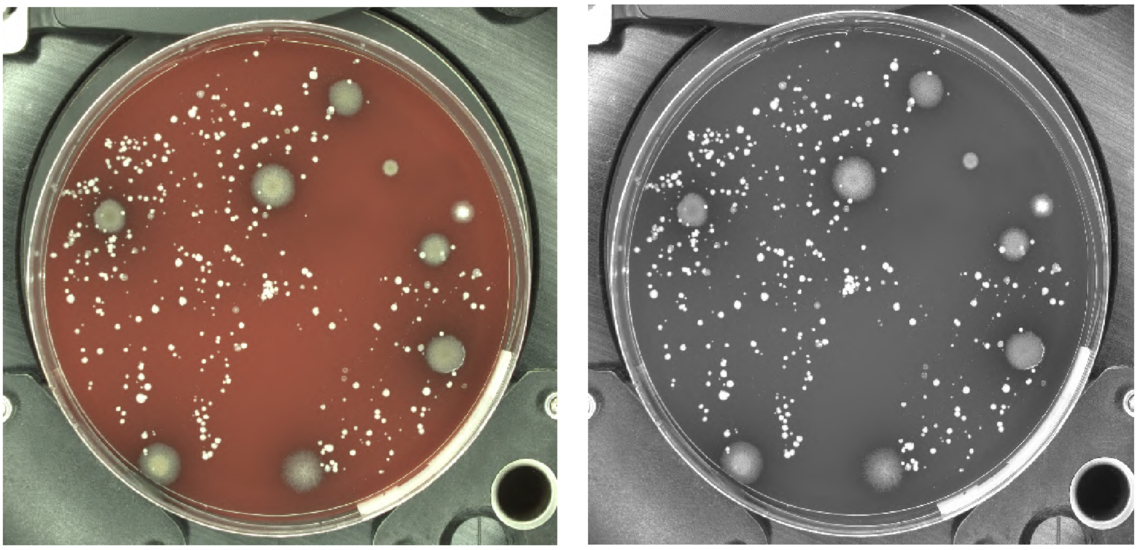
(a)

(b)

Fig. 1.1: An example of a regular color image in Subfigure a) and its corresponding binary image in Subfigure b).

Gray-scale images

A gray-scale image is an image consisting only of shades of gray, containing no color information. Typically, such images consist of 8-bit pixel values, thereby providing 256 distinct levels of gray per pixel. Higher values correspond to brighter pixels. Various applications may however require 12 or 16 bits, permitting more gray shades and increased accuracy in capturing detail. An example of the gray-scale image can be seen in Figure 1.2.



(a) (b)

Fig. 1.2: An example of a regular color image in Subfigure a) and its corresponding grayscale image in Subfigure b).

Color images

Digital images can be represented in various color models. This section discusses several well known types of color spaces.

RGB images

Unlike the previous single-channel images mentioned above, the RGB image consists of three channels – R (red), G (green), and B (blue). The addition of these channel values results in the creation of various colors.

By using 8 bits per pixel for individual color channels, we obtain a 24-bit color model, known as "True Color" which allows a wide range of colors to be represented in digital images. This model can be seen in Fig. 1.3 [20, 22].

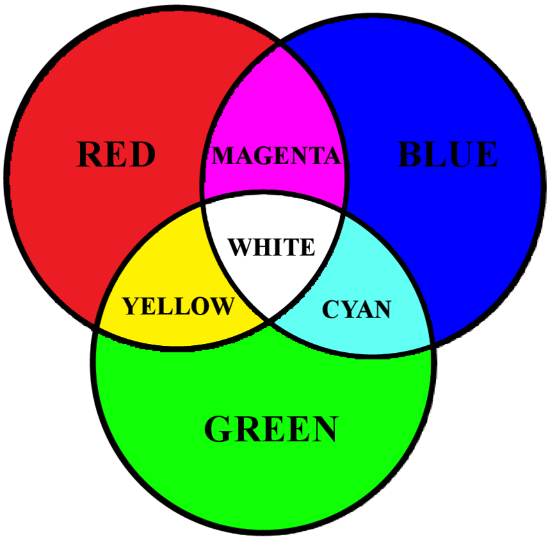


Fig. 1.3: RGB color model.

Other color models

In addition to the RGB model, several other color models are used in various areas of digital imaging:

- **CMYK color model (Fig. 1.4)** – Used primarily in the printing and graphic arts industries, combines four colors to reproduce a wide range of colors – cyan, magenta, yellow, and key (black).
- **HSV color model (Hue, Saturation, Value)** – Hue determines the base color, saturation determines the purity of that color, and value affects brightness and intensity.
- **HSL color model (Hue, Saturation, Lightness)** – Works with hue and saturation, but instead of value, it uses lightness to determine the brightness and tone of a color.
- **YIQ color model** – Used in television broadcasting, it works with luminance (Y) and two chrominance components (I and Q) to represent colors.

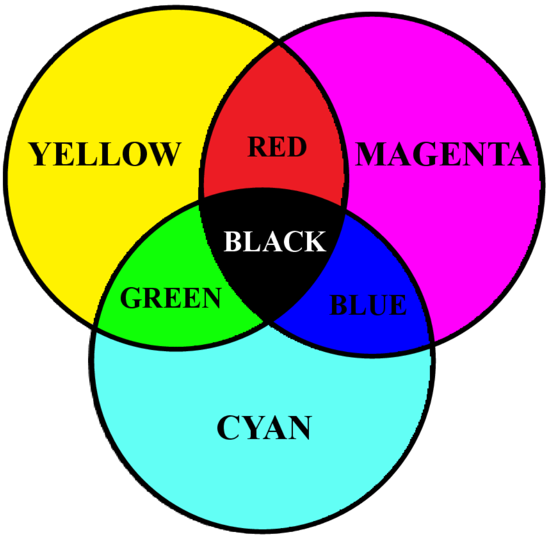


Fig. 1.4: CMYK color model.

1.1.2 Image processing techniques

Digital image processing is a vast and complex field that provides a variety of algorithms and techniques to achieve different results, such as image classification, image restoration, object detection, image registration, image segmentation, image compression, and more. This section focuses on specific digital image processing techniques and their applications that are fundamental to the context of the thesis. These include image classification, object detection, image registration, and image segmentation.

Image classification

Image classification is a basic computer vision task that assigns a label or class to the entire image. Each image is assumed to have only one class. Image classification models take an image as input and return a prediction of which class the image belongs to. This process usually involves three main stages: image preprocessing, image feature extraction, and classification [38].

Various types of image classification models exist, which can be broadly categorized into unsupervised and supervised methods. Unsupervised classification methods are automated processes that operate without utilizing labeled training data; instead, they aim to identify underlying patterns or clusters within the image data. Examples of unsupervised methods include cluster analysis techniques. In contrast, supervised classification methods rely on labeled training datasets to train models to make predictions. This category encompasses models such as those based on convolutional neural networks (CNNs) [39].

One of the experiments conducted in the practical part of this thesis involves image classification utilizing convolutional neural networks, with a detailed exposition provided in Section 1.2.

Object detection

Object detection is an image processing technique that locates objects in images. There are various algorithms available for object detection, but the state-of-the-art approaches utilize Machine Learning (ML) – specifically Deep Learning (DL). Both of these terms fall under the umbrella of Artificial Intelligence (AI). For clarity, the illustration in Figure 1.5 shows the hierarchical relationship of these concepts.

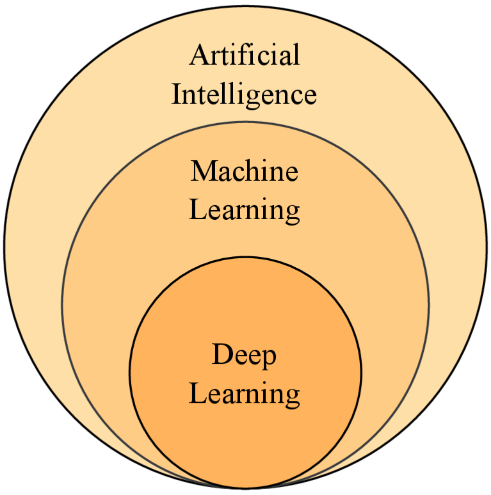


Fig. 1.5: Relationship between AI, ML and DL.

Deep learning techniques necessitate a substantial number of labeled training images and frequently demand the hardware optimization of graphics processing units (GPUs) due to the long training time. It is possible to train your own object detector or use a pre-trained object detector utilizing transfer learning, which can then be tuned for the application. Such techniques leverage convolution neural networks, which are more precisely described in Section 1.2.

For other machine learning techniques for object detection, it is also possible to either start with a pre-trained model or create your object detector to suit your application. Typical machine learning methods encompass aggregate channel features (ACF), SVM classification using histograms of oriented gradient (HOG) features, or the Viola-Jones algorithm for human face or upper body detection. Compared to a deep learning-based workflow where the feature selection is automatic, using machine learning requires features to be manually selected for object identification [24].

Image registration

Image registration is one of the main steps of image analysis. It is a process of discovering matching points between two or more images and aligning them into one integrated image.

The principle of the method is overlaying and aligning images of the same scene taken at different times, from different viewpoints, by different modalities (PET/MRI, PET/CT, SPECT/CT), or matching the scene to a model. This can be used for example for monitoring the tumor evolution, shape recovery, or comparison of target template and real-time image. More precisely, it is finding an optimal transformation of one image to resemble the structure of interest of another image. The result is a fusion of the reference image and the sensed images [8, 9].

The image registration techniques find widespread application in medical and satellite imaging, among other fields.

In the context of image registration, the RANSAC (The Random Sample Consensus) method is often used to find transformation parameters between two images. This transformation can include translation, rotation, scaling or fish eye distortion, and is used to align one image with another, which is important in many computer vision and image processing applications such as image fusion, motion capture, and object tracking. It is a robust algorithm commonly used in computer vision to reject outliers in parameter estimation. It works by iteratively selecting random subsets of data points to fit a model and then evaluating the quality of that model based on the entire data set [40, 41].

The main problem with the RANSAC method is its sensitivity to parameter choices, such as the number of key points in images, the matching rate, and the execution time required for each algorithm. This can lead to problems in accurately identifying outliers, especially in scenarios with high noise or complex data distributions.

In the area of feature detection, the important methods used for feature extraction from images are the Oriented Rapid and Rotated (ORB) method, the Scale-Invariant Feature Transform (SIFT) method, and the Speeded Up Robust Features (SURF) method.

ORB combines the FAST keypoint detector with the BRIEF descriptor and offers a computationally efficient alternative to SIFT and SURF. SIFT is known for its robustness to changes in scale and orientation, but can be computationally intensive. SURF, on the other hand, provides speed advantages over SIFT because it uses integral images to extract features. Each of these detectors has its own strengths and weaknesses, making them suitable for different applications based on factors such as computational resources and the nature of the image data [40].

Image segmentation

Image segmentation is the process of dividing an image into multiple segments, where each segment represents a different object in the image. This process is often used as a pre-processing step for object detection [18, 35].

Numerous algorithms are available for performing image segmentation. One of the simplest approaches can be considered thresholding. The thresholding process is defined by the formula:

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) > T \\ 0 & \text{if} f(x,y) \leq T \end{cases}, \tag{1.1}$$

where $g(x,y)$ is the output binary image, $f(x,y)$ represents the pixel intensity of the input image at coordinates (x,y) and T is the threshold value.

Another frequently used method is multi-level thresholding, which uses multiple thresholds and divides the image into several distinct regions according to their brightness levels. The multi-level thresholding process is defined by the formula:

$$g(x,y) = \begin{cases} a & \text{if } f(x,y) > T_2 \\ b & \text{if } T_1 \leq f(x,y) \leq T_2 \\ c & \text{if } f(x,y) \leq T_k \end{cases}, \tag{1.2}$$

where $T1, T2, ..., Tk$ are specific threshold values.

Contemporary methods employ automated image segmentation algorithms based on deep learning for addressing both binary and multi-label segmentation challenges.

Another important technique for image segmentation is edge detection. One widely used method is the Canny edge detection filter, which identifies edges in an image by detecting sudden changes in intensity.

The Canny function has three general parameters:

- image
- threshold 1
- threshold 2

The input parameter image is the input image to be processed, threshold 1 and threshold 2 are the thresholds that work so that if the gradient value of an edge is greater than threshold 2, it will be considered an edge. If it is less than threshold 1, it will be considered a non-edge and if the value is between these two thresholds, it will be considered an edge only if it is connected to another edge. The Canny function returns a binary image where the white pixels represent detected edges and the black pixels are non-edge regions [42].

1.2 Convolutional neural network

This section explains the concept of convolutional neural networks (CNN) and introduces the principle of how they work.

Convolutional neural networks are a form of a deep neural network. They are commonly used to process visual data, including images and video. Convolution is leveraged to extract features and hierarchical learning is employed for pattern recognition, rendering them highly suitable for tasks such as image classification, object detection, and segmentation [36, 37].

A convolutional neural network (CNN) is part of deep learning (DL), which is a subset of machine learning (ML). All of this then falls under Artificial Intelligence (AI). The hierarchical relationship, including the incorporation of CNN, is demonstrated in Figure 1.6.

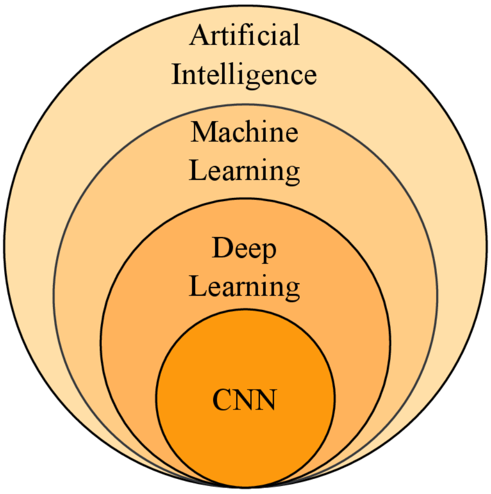


Fig. 1.6: Relationship between AI, ML, DL and CNN.

AI is the simulation of human-like intelligence and decision-making processes in computer systems. ML refers to algorithms that learn without being explicitly programmed, and DL represents a set of algorithms based on artificial neural networks that mimic the processes of the human brain [26].

1.2.1 CNN architecture

The CNN architecture is composed of multiple layers, such as convolutional, pooling, and fully connected layers. By repeating and combining these layers in various ways, the network can extract different levels of features and information from input data. The architecture's complexity allows for sophisticated analysis and processing [25].

The CNN architecture model is illustrated in Figure 1.7.

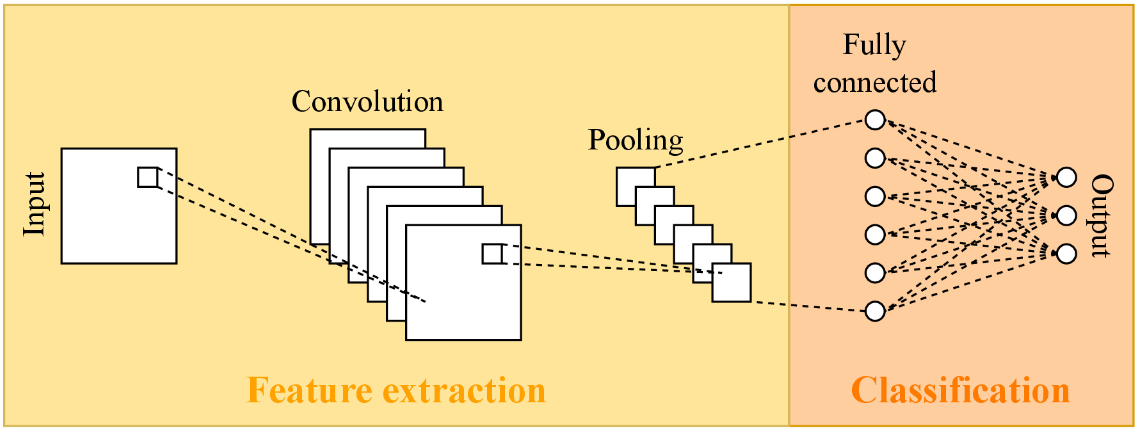


Fig. 1.7: CNN architecture featuring convolutional layer for feature extraction, pooling layer for down-sampling, and fully connected layer for classification.

Convolution layer

The convolution layer extracts symptoms through a blend of linear and non-linear procedures. Said procedures consist of a convolution operation and an activation function, which will be better explained in Section **Activation functions**.

Convolution is a linear operation that involves the displacement of a small mask, called a kernel, across the input image. At each location of the tensor, the product of the kernel's elements and the corresponding elements of the input tensor are calculated and then summed, yielding the output value at the corresponding location of the output tensor. The result is a reduced image, specifically a feature map. This procedure is demonstrated in Figure 1.8. To prevent the reduction of image dimensions, zero padding is utilized to add zeroes to both rows and columns of the input tensor. This process plays a critical role in convolutional neural networks as it ensures that the output dimensions of one layer correspond to the desired input dimensions of the next layer, which is imperative for the proper operation of the CNN [25].

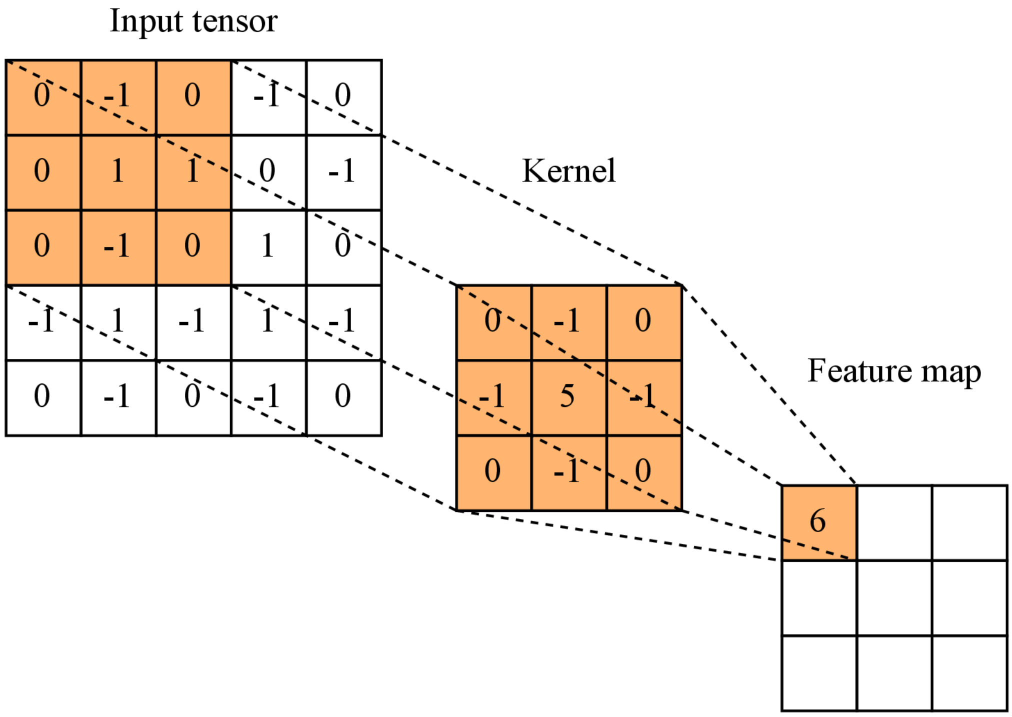


Fig. 1.8: The principle of convolution: the Kernel sliding across an input image, multiplying its elements with corresponding input tensor elements at each location, and summing the results to produce a feature map.

The process of training a CNN model for the convolution layer involves identifying the most efficient kernels for a given task, based on a training dataset. Kernels are the sole parameters learned automatically during the convolutional layer training process. Meanwhile, kernel size, number of kernels, padding, and stride[1] are hyperparameters that require configuration before the commencement of training.

Pooling layer

The pooling layer facilitates image subsampling and extraction of important information. Generally, a kernel of a certain size is used and its output for a certain area is either the average (local/ global average pooling) or the highest value (max pooling) of the local and non-overlapping parts of the image. These processes are illustrated in Figure 1.9. As a result, the spatial dimensionality of the data and the computational complexity in the subsequent layers is reduced [27, 25].

[1] The stride parameter defines the amount of filter shift across the input image.

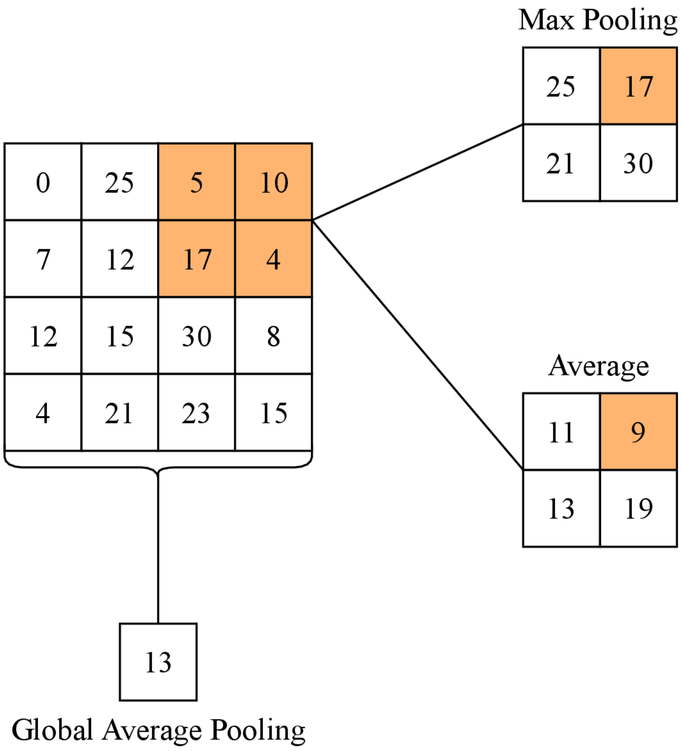


Fig. 1.9: Types of pooling operations: Max Pooling, Average, and Global Average Pooling.

Fully connected layer

The fully connected layer is typically the last layer of any CNN architecture for classification tasks. Its name derives from the fact that each neuron in this layer is linked to all neurons in the previous layer by a learnable weight. The input to this layer is commonly flattened, which results in a one-dimensional numerical array, a transformation carried out by the predecessor pooling or convolution layer.

The fully connected layer usually has the same number of output nodes as the number of classes. The outputs of a fully connected layer depend on the network task. In classification and segmentation neural networks, the final output represents the probabilities for each class, while in regression problems they can represent continuous values such as e.g. temperature or price predictions [27, 25].

Activation functions

The activation function operates as a transfer function in neural networks, influencing output and mapping the resulting values across the range of real numbers. They can be based on either non-linear or linear operations.

Linear activation functions are simple functions that produce a linear output based on an input value. They are rarely used in neural networks because they do not introduce nonlinearity into the model. However, linear activation functions can be advantageous in specific situations, such as in the output layer of a regression model where a linear output is required [32, 25].

Among the nonlinear activation functions, the Rectified Linear Unit (ReLU) stands out as one of the most widely used. While ReLU is prevalent in current neural network architectures, the Sigmoid and Tanh functions were more extensively utilized in earlier years. This preference was attributed to their mathematical properties, which closely resemble the behavior of biological neurons. These functions are depicted in Figure 1.10).

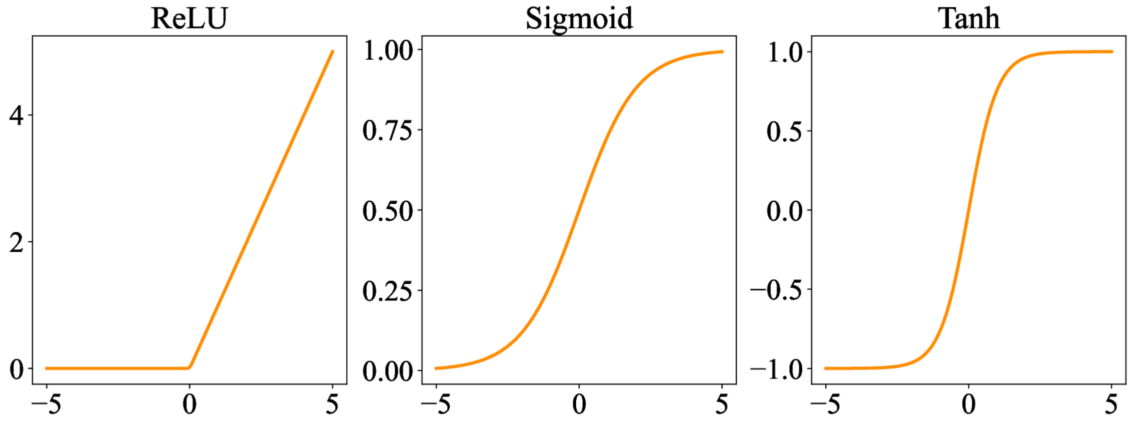


Fig. 1.10: Activation functions – ReLU, Sigmoid, Tanh.

1.2.2 CNN training

Network training is a fundamental process aimed at discovering meaningful representations in the convolutional layers and optimal weights in the fully connected layers. This is achieved by minimizing the differences between predicted outputs and ground truth values on the training dataset through the iterative adjustment of various network parameters. This process involves two essential components: the loss function and the gradient descent optimization algorithm [25].

A central element in CNN training is backpropagation, a mechanism that involves computing the gradients of the loss function concerning the network's parameters and using these gradients to update the parameters in the opposite direction of the gradient. This iterative optimization process plays a central role in fine-tuning the network's weights, enabling it to learn and adapt to the underlying patterns in the data.

Loss function

The loss function quantifies how well the model's output predictions match the true labels during forward propagation. Regression tasks typically use mean squared error (1.3), while cross-entropy is the most common loss function for multiclass classification (Binary cross-entropy loss function 1.4, Categorical cross-entropy loss function 1.5). The selection of the loss function hinges on the particular issue under examination and constitutes a hyperparameter [25, 33, 34].

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{1.3}$$

where N is the number of objects, y_i is the true label of the i-th object and $\hat{y}_i$ is the predicted value for the i-th object.

$$H(y, \hat{y}) = -\left(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})\right), \tag{1.4}$$

where y is the true label (either 0 or 1) and $\hat{y}_i$ is the predicted probability that the instance belongs to class 1.

$$H(y, \hat{y}) = -\sum_{i=1}^{C} y_i \cdot \log(\hat{y}_i), \tag{1.5}$$

where C is the number of classes, y_i is the true probability distribution for class i and $\hat{y}_i$ is the predicted probability distribution for class i.

Gradient descent

Gradient descent serves as the primary optimization algorithm during training. It iteratively updates the trainable parameters, such as kernels and weights, to minimize loss. The gradient of the loss function indicates the direction of the steepest increase, and each parameter is adjusted in the opposite direction of the gradient, with the step size determined by a hyperparameter called the learning rate [25, 34].

Mathematically, the update of a parameter (w) is computed as follows:

$$w := w - \alpha \cdot \frac{\delta L}{\delta w}, \tag{1.6}$$

where α is the learning rate and $\frac{\delta L}{\delta w}$ denotes the gradient of the loss function (L) with respect to the parameter w.

2 Colony picking and MBT Pathfinder image analysis

This chapter deals with the issue of microbial colonies, their growth and multiplication processes, and explains the concept of colony picking and the advantages of automating this process. The chapter also introduces the MBT Pathfinder device on which all experiments were performed.

2.1 Colony picking

Colony picking is a fundamental technique for isolating and studying individual microbial species used in various fields of biology, such as microbiology, molecular biology, and genetics. It involves the selection, retrieval, and transfer of microbial colonies, typically bacteria and yeasts, grown on a solid medium.

2.1.1 Microbial colonies and their cultivation

Microbial colonies are visible aggregates of microorganisms, such as bacteria, yeasts, or fungi, that have grown and multiplied from a single mother cell on a solid nutrient medium. They can vary in size, shape, color, and texture, depending on the type of microorganism, growth medium, and cultivation conditions [12].

Cultivation of microbial colonies involves the transfer of an inoculum, a small number of microbes, to a nutrient medium that contains essential nutrients to promote their growth [13].

Nutrient media can be classified into 3 forms:

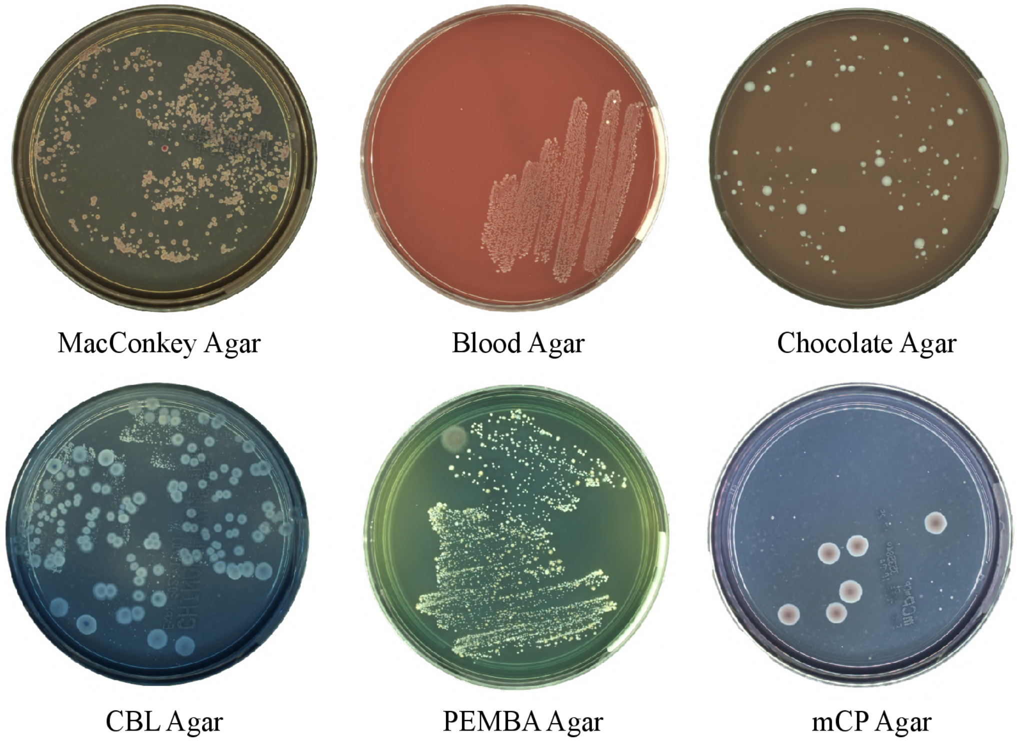
1. **Liquid media** – They are commonly used for growing microorganisms in suspension, such as for fermentation studies or maintaining cultures [29].
2. **Semisolid media** – They contain a lower concentration of agar than solid media (0.5 % or less) and produce a gel-like consistency that allows for the detection of bacterial motility.
3. **Solid media** – They provide a solid surface for microorganisms to grow on. This enables the formation of visible aggregates of microbes, i.e. colonies.

The most commonly used solid media are agar plates prepared by pouring agar into Petri dishes. Agar is a mixture of polysaccharides composed of galactose monomers, consisting of two primary components: agarose, a linear polysaccharide that forms a gel, and agaropectin, a branched non-gelling component. Agar comes in various types with different additives to suit specific microbial growth requirements. Discussion of these types can be found in Section **Types of agar** [1, 2].

Types of agar

Due to the differing growth requirements of various microorganisms, it is impossible to culture all microbial organisms using a single culture medium. Therefore, there are many different types of agars [1]. Some of the commonly used ones are listed below and displayed in Figure 2.1:

- **MacConkey Agar** – It isolates and differentiates lactose (pink colonies) and non-lactose (pale or translucent colonies) fermenting gram-negative enteric bacilli.
- **Blood Agar** – It is used for cultivating fastidious microorganisms and isolating and differentiating bacteria based on the types of hemolysis present.
- **Chocolate Agar** – It is enriched with heat-treated blood for cultivating fastidious microorganisms, especially Neisseria and Haemophilus species.
- **CBL Agar (China-blue lactose agar)** – It differentiates lactose-positive and lactose-negative microorganisms. Thus, lactose-positive microorganisms are blue, and lactose-negative are translucent.
- **PEMBA Agar (Bacillus cereus Selective Agar)** – It is used for the isolation and enumeration of Bacillus cereus in food samples. Bacillus cereus will appear blue with egg yolk precipitate of the same color around it.
- **mCP Agar (Membrane-Clostridium-Perfringens-Agar)** – This agar is applied for the detection and enumeration of Clostridium perfringens in water. It causes the formation of characteristic non-translucent golden colonies. The other colonies appear to be purple or blue-green.



MacConkey Agar

Blood Agar

Chocolate Agar

CBL Agar

PEMBA Agar

mCP Agar

Fig. 2.1: Several types of culture media.

Multiplication and growth of microorganisms

The growth of microorganisms is a situation in which all the chemical components of a cell increase in an orderly manner.

There are various mechanisms of cell multiplication [12, 13]:

- **Binary fission** – After reaching a certain size and weight, the parent cell divides into two identical daughter cells as demonstrated in Figure 2.2. This mechanism is most common in bacteria.
- **Budding** – A new organism arises from an outgrowth or bud at one particular location of the mother cell. It remains attached to the parent cell during its growth, detaching after maturation and leaving a scar. This mechanism is most common in yeasts.
- **Fragmentation** – An organism is split into fragments, which then individually mature. Fully developed organisms are clones of the original organism. This mechanism is most common in cyanobacteria.

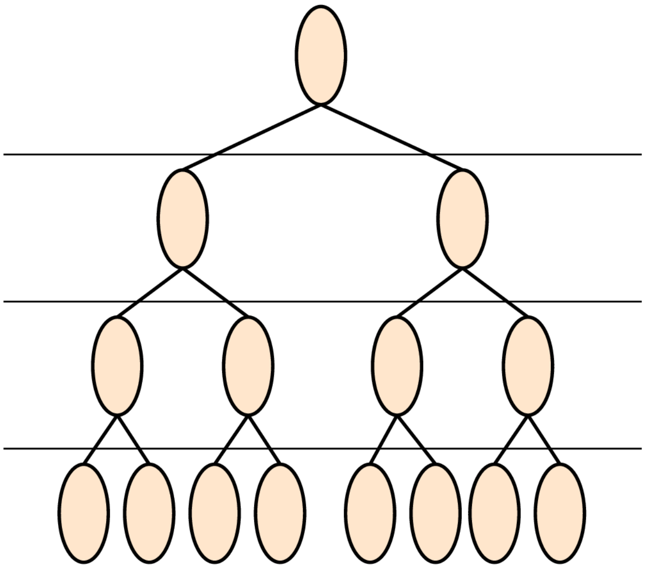


Fig. 2.2: Cell multiplication.

These processes lead to an increase in the number of cells and the formation of a population or culture [7].
It depends on:

- Oxygen
- Carbon dioxide
- Temperature
- pH
- Light
- Osmotic Pressure

The typical progression of microorganisms' multiplication and growth on the medium in a closed system is expressed by the growth curve (Figure 2.3).
The growth curve is divided into 4 characteristic phases [13, 5, 6]:

1. **Lag Phase** – The bacteria don't multiply, they only adjust their metabolism to the new environment.
2. **Log Phase** – The cells begin to divide in a regular exponential manner and some of them die, in the process, metabolites are formed.
3. **Stationary Phase** – The maximum number of bacteria is reached, as a result of nutrient depletion and high metabolite concentration, the number of cells equals the number of dead cells, so the growth rate is zero.
4. **Death Phase** – The number of dead cells exceeds the number of living cells as the bacteria lose the ability to divide. It may return to the lag phase and repeat the cycle.

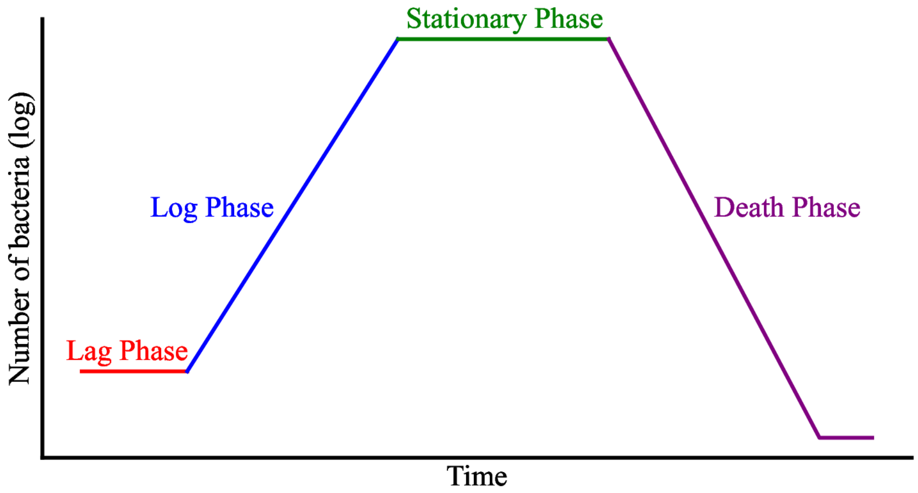


Fig. 2.3: Phases of the bacterial growth curve.

2.1.2 Manual colony picking method

Colony picking is a crucial procedure in microbiology and biotechnology research. The manual colony picking method relies on trained personnel employing aseptic techniques and specialized instruments to manually select individual colonies. Although this method allows accurate control over colony selection, it is tedious, time-consuming, and often inaccurate. This also makes it less practical for handling large numbers of samples [30, 31].

In the past, manual colony picking was the only available method. However, with the invention of automated colony picking instruments, the process has become much more streamlined and accurate.

2.1.3 Automation of manual colony picking method

Automating the colony picking methods has reduced the risk of sample contamination and improved their overall consistency. These instruments not only collect colonies from Petri dishes but can also inoculate them and in cases where the collection needles are multi-use, the instrument can additionally clean, sterilize, and dry the picking needle. As a result, automatic colony picking instruments are becoming more and more frequent in microbiological laboratories due to their increased efficiency and accuracy [3, 4].

2.2 MBT Pathfinder image analysis

This section will focus on the prototype of an automated picking robot called the MBT Pathfinder, developed by Bruker Daltonics GmbH & Co. KG.

2.2.1 Introducing MBT Pathfinder

The MBT Pathfinder is a robot designed to automate the process of transferring microbial colonies from agar plates to a MALDI target and preparing this target for subsequent classification of microbial species by mass spectrometry.

The main components of the device include a sample tray for positioning the targets and a Petri dish, reagents, and two transfer loops.

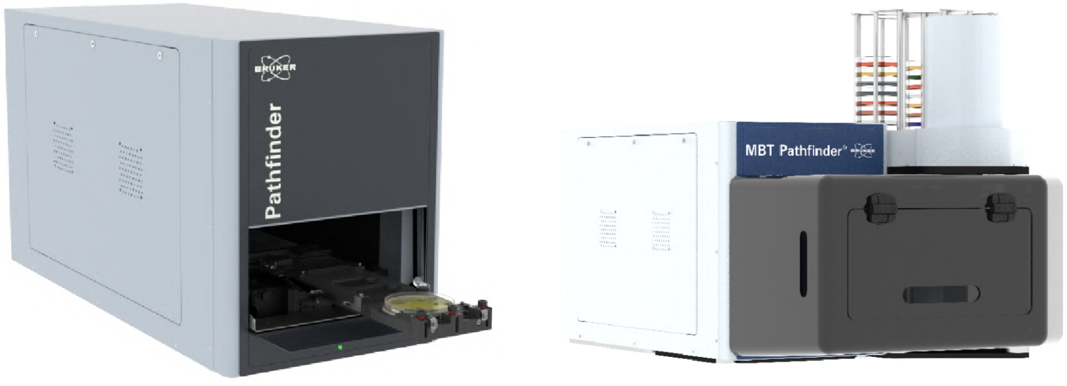
In sample preparation, the initial step involves the sequential insertion of individual agar plates. This can be done manually, but more often the MBT Pathfinder is combined with the Feeder, an automated robot that is used to feed the dishes into the MBT Pathfinder . Both configurations of the instrument, with and without the Feeder attachment, are depicted in Figure 2.4.

Following the insertion of the dish, images of the dish are captured from a top view under six different light configurations, after which the dish is returned to the Feeder. Subsequently, another dish is inserted and the process is repeated until all prepared dishes are processed. The algorithm then uses image processing and segmentation to automatically pre-select microbial colonies on each Petri dish. The user can then confirm the pre-selection or manually select additional colonies for transfer to the MALDI target.

The next step is applying the formic acid to the given positions of the MALDI target. It aims to disrupt the cell wall, allowing better extraction of the cell contents.

The Petri dishes are reinserted into the MBT Pathfinder , and the selected colonies are picked using a transfer loop and smeared onto the positions of the MALDI target containing a droplet of formic acid. The transfer loop is sterilized by heating it to a high temperature in a special annealing chamber to prevent cross-contamination. It is essential to locate the previously selected colony accurately at this stage. If the dish is not inserted into the instrument at the same angle as when the image was captured, it can significantly affect the preparation of the sample. The resulting problem will be comprehensively discussed and addressed in the following section. 2.2.2.

After transferring the colony to the target, the drying process is initiated to speed up sample preparation. The HCCA MALDI matrix is then applied to the given positions and the drying process is repeated. After this process, the target is ready to use for subsequent analysis.



(a)

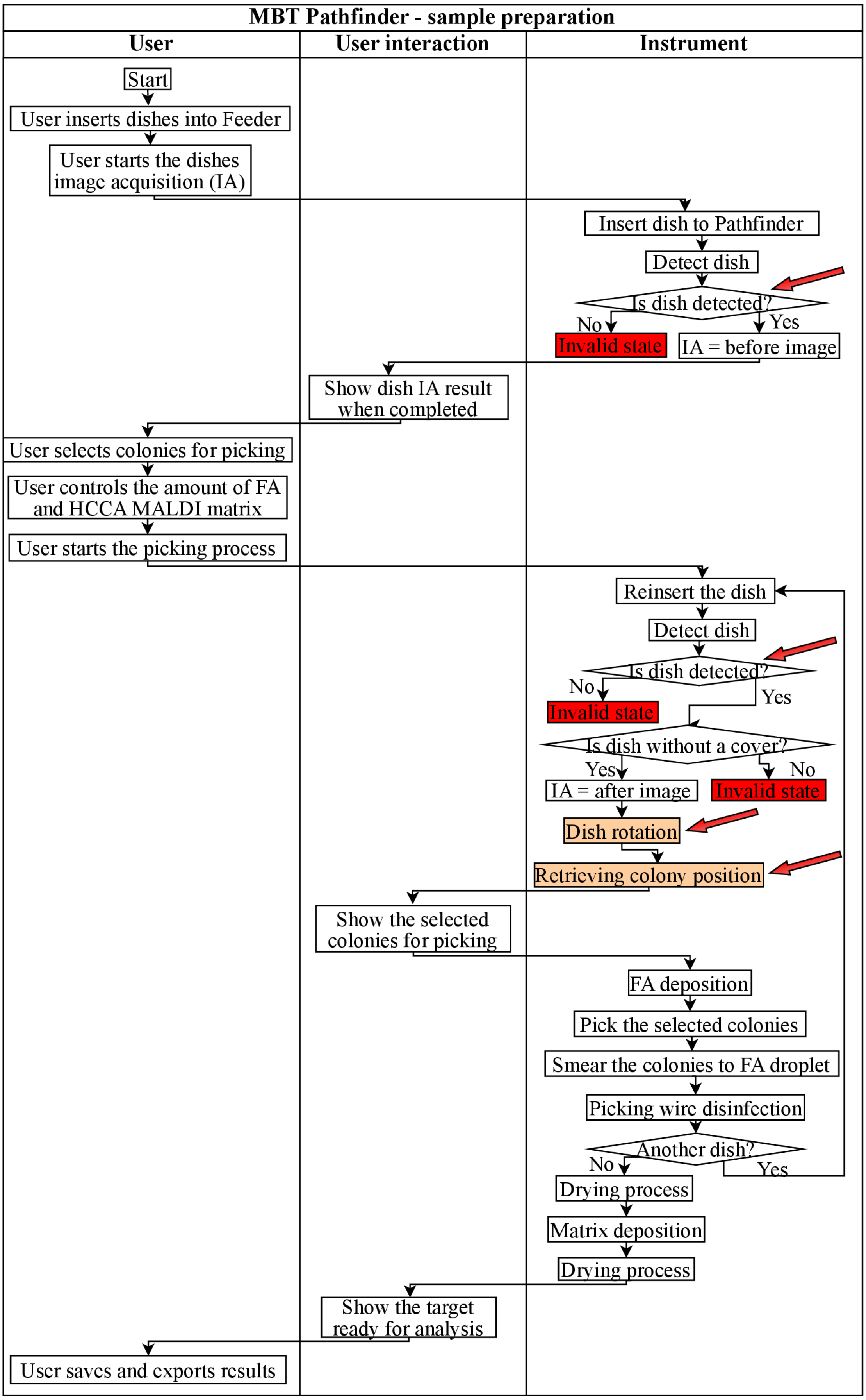
(b)

Fig. 2.4: MBT Pathfinder without Feeder in Subfigure a) and MBT Pathfinder combined with the Feeder in Subfigure b).

2.2.2 Crucial workflow steps

After a thorough investigation, I have determined three critical steps in the instrument's workflow. Failure of any of them could have a fatal impact on the prepared sample quality, potentially leading to incorrect diagnoses or inaccurate scientific results. It is crucial to thoroughly examine these essential steps to ensure their reliability and accuracy and make systematic improvements where necessary.

The crucial steps, namely dish detection, dish rotation, and colony position retrieval, are indicated by red arrows in the diagram 2.5. All those steps will be described in more detail in the following subsections.



MBT Pathfinder - sample preparation
User
User interaction
Instrument
Start
User inserts dishes into Feeder
User starts the dishes image acquisition (IA)
Insert dish to Pathfinder
Detect dish
Is dish detected?
No
Yes
Invalid state
IA = before image
Show dish IA result when completed
User selects colonies for picking
User controls the amount of FA and HCCA MALDI matrix
User starts the picking process
Reinsert the dish
Detect dish
Is dish detected?
No
Yes
Invalid state
Is dish without a cover?
Yes
No
IA = after image
Invalid state
Dish rotation
Retrieving colony position
Show the selected colonies for picking
FA deposition
Pick the selected colonies
Smear the colonies to FA droplet
Picking wire disinfection
Another dish?
No
Yes
Drying process
Matrix deposition
Drying process
Show the target ready for analysis
User saves and exports results

Fig. 2.5: MBT Pathfinder – sample preparation workflow.

Dish detection

The first crucial workflow step is dish detection. It aims to ensure whether an agar plate has been correctly placed onto the tray in the MBT Pathfinder or not.

If the dish is missing, but the algorithm evaluates it, the transfer loop may be destroyed trying to pick up a colony on a dish that is not there.

On the other hand, if a dish is inserted into the instrument, but the instrument does not evaluate it, it may not be possible to pick up the required colony.

Currently, this workflow step utilizes a convolutional neural network and a random forest model. To enhance clarity, refer to the diagram 2.6. The description and parameters of the CNN model are described in the diagram 2.7.

This architecture is quite computationally intensive and complicated. In addition, there are frequent classification errors, especially when processing a dish with a less common type of agar. The imperfections of the current neural network primarily resulted from a lack of variable data. Therefore, several experiments were performed for the purpose of improvement. These are described and presented in subsection 4.1.

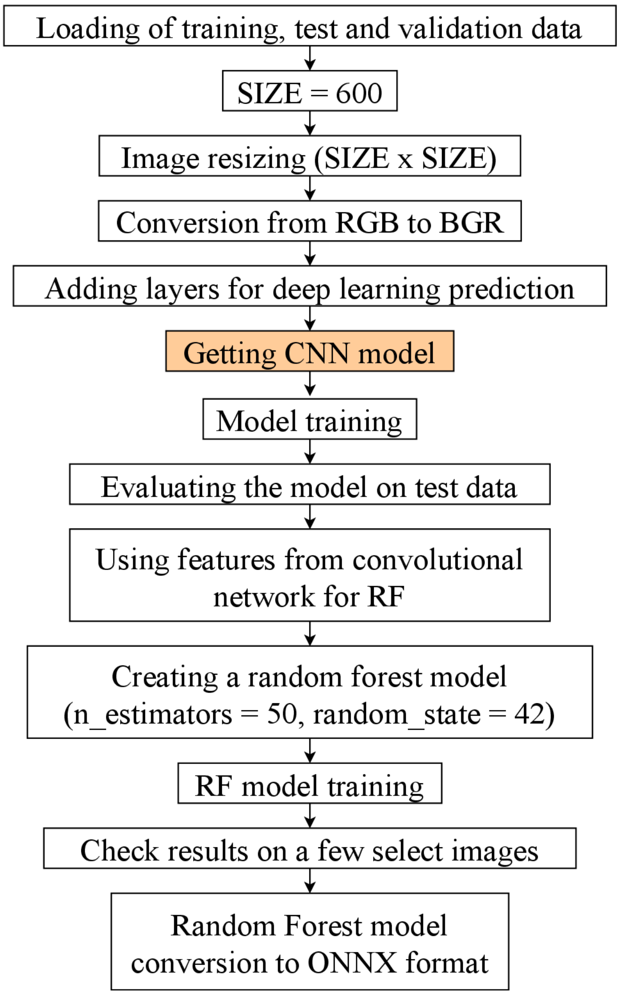

Loading of training, test and validation data
SIZE = 600
Image resizing (SIZE x SIZE)
Conversion from RGB to BGR
Adding layers for deep learning prediction
Getting CNN model
Model training
Evaluating the model on test data
Using features from convolutional network for RF
Creating a random forest model (n_estimators = 50, random_state = 42)
RF model training
Check results on a few select images
Random Forest model conversion to ONNX format

Fig. 2.6: Original architecture for dish detection.

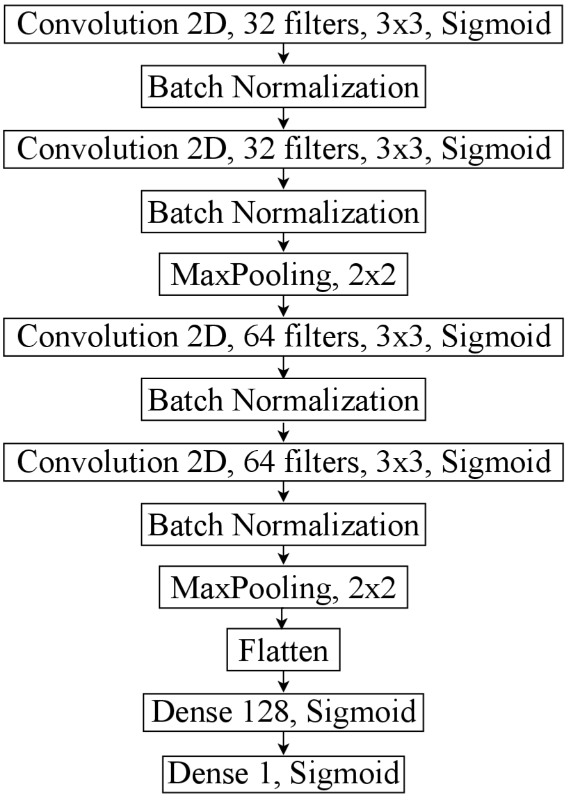
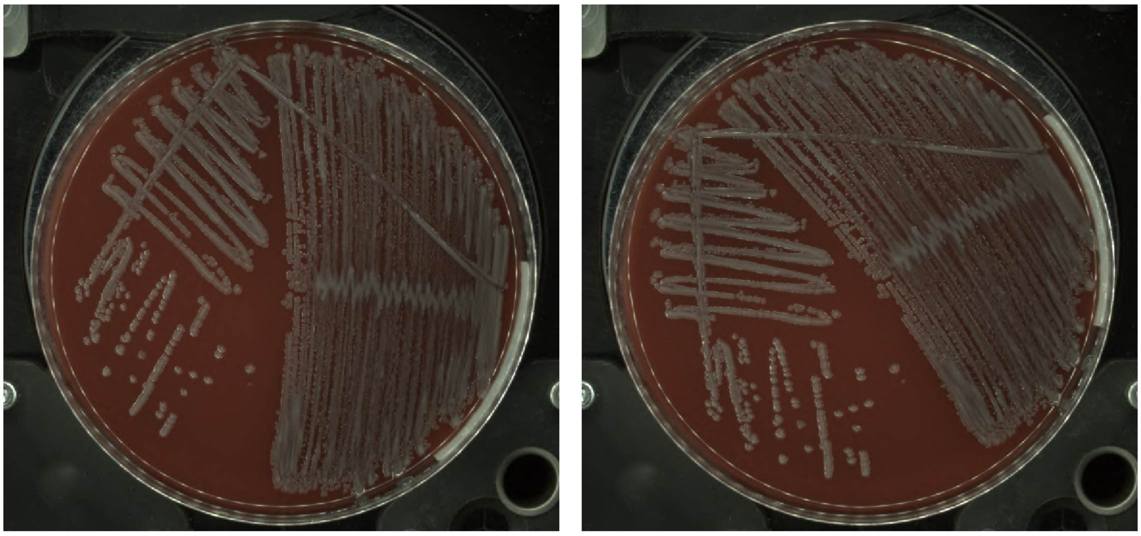

Convolution 2D, 32 filters, 3x3, Sigmoid
Batch Normalization
Convolution 2D, 32 filters, 3x3, Sigmoid
Batch Normalization
MaxPooling, 2x2
Convolution 2D, 64 filters, 3x3, Sigmoid
Batch Normalization
Convolution 2D, 64 filters, 3x3, Sigmoid
Batch Normalization
MaxPooling, 2x2
Flatten
Dense 128, Sigmoid
Dense 1, Sigmoid

Fig. 2.7: **The original model** comprises four convolutional layers (32, 32, 64, 64) with sigmoid activation functions for each layer. Batch Normalization is applied within each convolutional layer to normalize the outputs. The model incorporates Max Pooling with a 2x2 window. Additionally, the model includes two fully connected layers. The first layer consists of 128 neurons, and the second layer consists of a single neuron, both activated by a sigmoid activation function.

Dish rotation

Another problematic step in the workflow is retrieving a dish rotation angle. This step is necessary for the subsequent retrieval of the previously selected colony's position. For a better overview, the workflow diagram 2.10 is shown below.

Reinsertion of the agar plate may result in the plate having a slightly different rotation when compared to the image used to select the colonies to transfer. Each time the dish is inserted into the device, an image is obtained. For further description and explanation, I will give the following working names for these images: *Before image* and *After image*. These images are illustrated in Figure 2.8.



(a) Before image

(b) After image

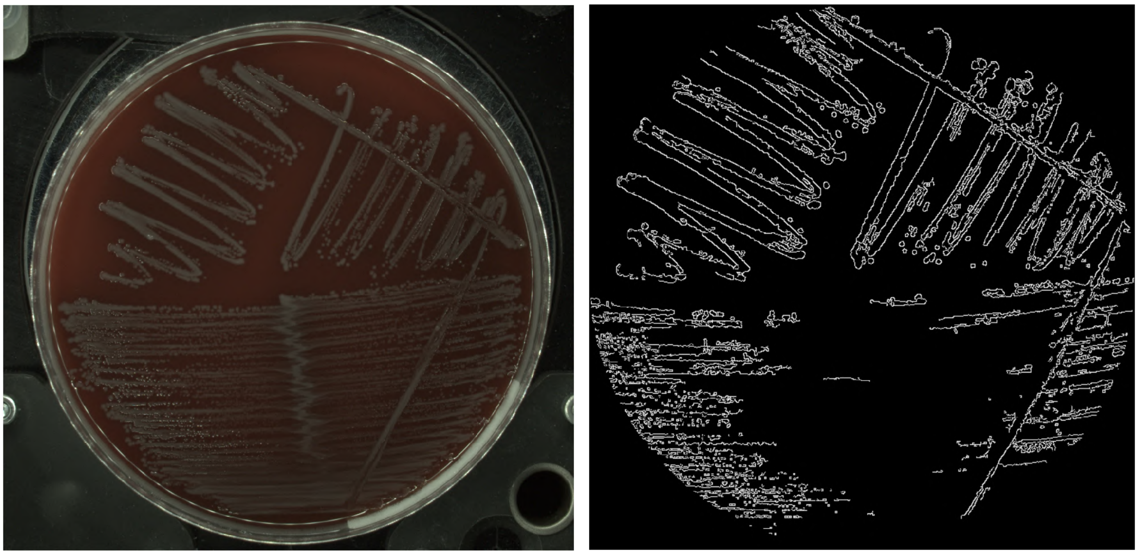
Fig. 2.8: An example of a *Before image* in Subfigure a) and an *After image* in Subfigure b).

In this step, the algorithm detects the difference in angles between the *Before* and *After images.* Subsequently, it rotates the *Before image* by this calculated angle.

The algorithm for finding the rotation angle first pre-processes the images – transforms images to grayscale, and then uses a Canny detection method that highlights the edges of the image. The result of using Canny detection is illustrated in Figure 2.9. Template matching is then utilized, which is a method relying on sliding a template, a small image with characteristic features, over the input image, a large image, where these characteristic features need to be found. To compare the template with the input image and locate the characteristic features, a matching process based on the calculation of normalized correlation coefficients is used. Normalization is performed to make the output independent of the intensity and contrast of the image, making it easier to compare. The equation for calculating the normalized correlation coefficient is illustrated below (2.1) [15].

$$R(x,y) = \frac{\sum_{x',y'}(T'(x',y')I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}(T'(x',y')^2 \sum_{x',y'} I'(x+x',y+y')^2}}, \tag{2.1}$$

where $T'(x',y')$ denotes the value of the template image at coordinates (x',y') representing pixel intensity in the template and $I'(x+x',y+y')$ is the value of the image at coordinates $(x+x',y+y')$, representing pixel intensity at that location in the image.



(a)

(b)

Fig. 2.9: An example of a regular image of the dish in Subfigure a) and its corresponding Canny edge detector output in Subfigure b).

I applied the algorithm to the test dataset, and based on the results detailed in Table A.1 in the Appendix, the algorithm's robustness comes into question. Due to the incorrect calculation of several rotation angles, a number of experiments were conducted to improve the algorithm's performance. These are described in subsection 4.2.

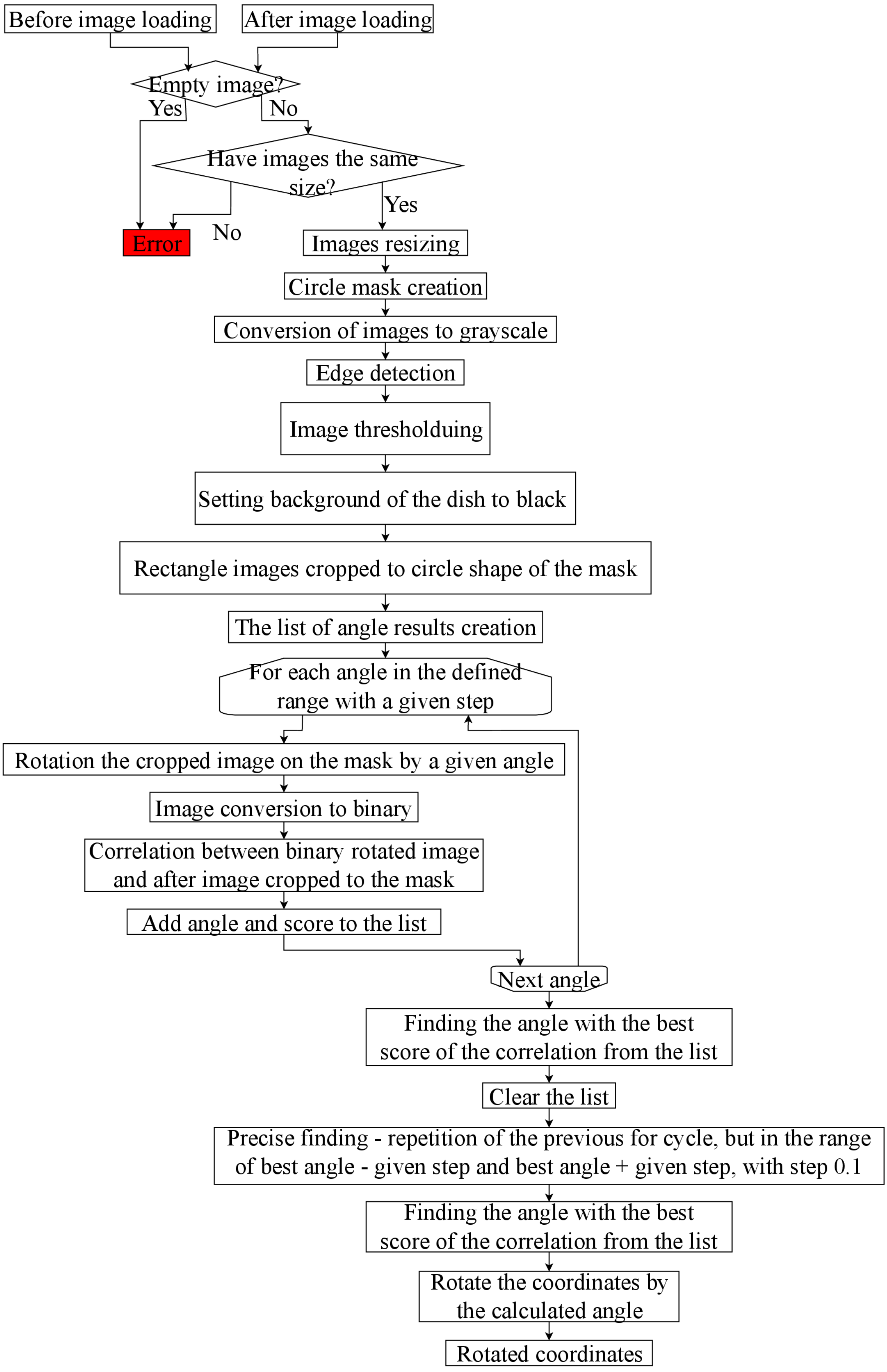
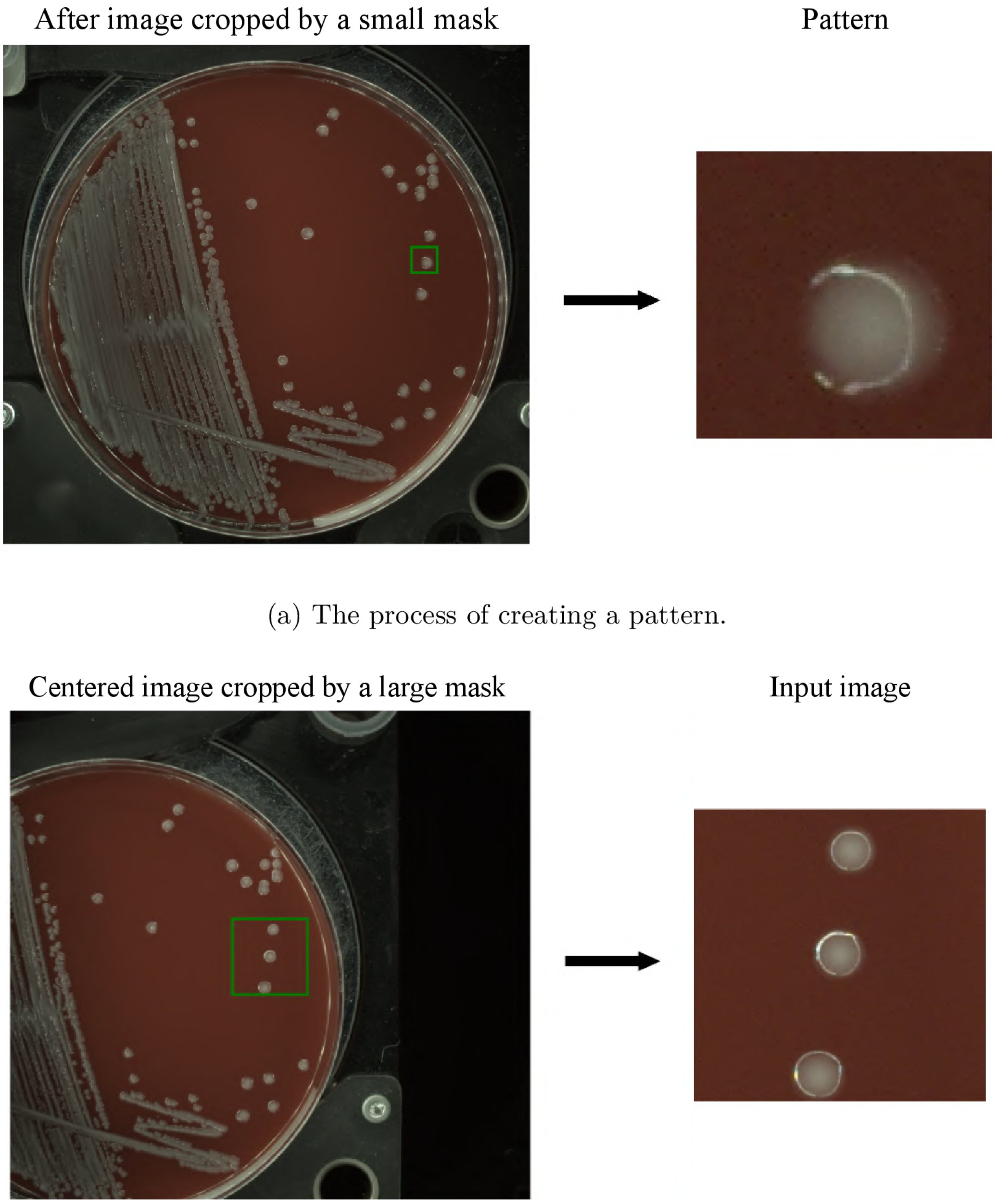

Before image loading
After image loading
Empty image?
Yes
No
Have images the same size?
Yes
No
Error
Images resizing
Circle mask creation
Conversion of images to grayscale
Edge detection
Image thresholduing
Setting background of the dish to black
Rectangle images cropped to circle shape of the mask
The list of angle results creation
For each angle in the defined range with a given step
Rotation the cropped image on the mask by a given angle
Image conversion to binary
Correlation between binary rotated image and after image cropped to the mask
Add angle and score to the list
Next angle
Finding the angle with the best score of the correlation from the list
Clear the list
Precise finding - repetition of the previous for cycle, but in the range of best angle - given step and best angle + given step, with step 0.1
Finding the angle with the best score of the correlation from the list
Rotate the coordinates by the calculated angle
Rotated coordinates

Fig. 2.10: Dish rotation workflow diagram.

Retrieving colony position

The most crucial step is retrieving the colony position because the accuracy of the entire sample preparation process is heavily influenced by the accuracy of this step. In this step, the algorithm uses image processing to find the selected colony by the user on the reinserted dish. For a clearer view, see the diagram 2.12.

This step is based on the rotation of the *Before image* and the before coordinates by the calculated angle, the output is then a *Rotated image* and *Rotated coordinates*. Then the pattern and the input images are created. The pattern is an *After image* cropped by a small mask with the center in *Rotated coordinates*. The input image is a *Centered image* cropped by a large mask with the center in the center of the image. These processes are depicted in Figure 2.11.


After image cropped by a small mask
Pattern
Centered image cropped by a large mask
Input image

(a) The process of creating a pattern.

(b) The process of creating an input image.

Fig. 2.11: The process of creating a pattern and an input image.

The subsequent step includes utilizing the template matching method to calculate the shift between the input image and the pattern. Thereafter, the new centered x and y coordinates are calculated.

The tray of the MBT Pathfinder then moves the dish to these new centered coordinates and in the center of the dish picks the colony, which is transferred to a MALDI target.

The pattern might get disrupted due to a potential error in determining the rotation angle. This can be observed in Subfigure 2.11a, where the selected colony should ideally be positioned right in the middle. However, it is shifted due to the inaccurate calculation of the angle of rotation. If the angle calculation were entirely incorrect, the cropping would occur elsewhere, resulting in the selected colony not being present in the pattern. This could lead to incorrectly identifying the position of the chosen colony, which, in turn, could result in selecting the wrong one. The outcomes of applying this algorithm to the test dataset are displayed in the Appendix (A.1).

The inaccurate determination of colony positions poses a critical challenge, necessitating improvements to the algorithm. The details of the experiments conducted to enhance the original algorithm are discussed in subsection 4.3.

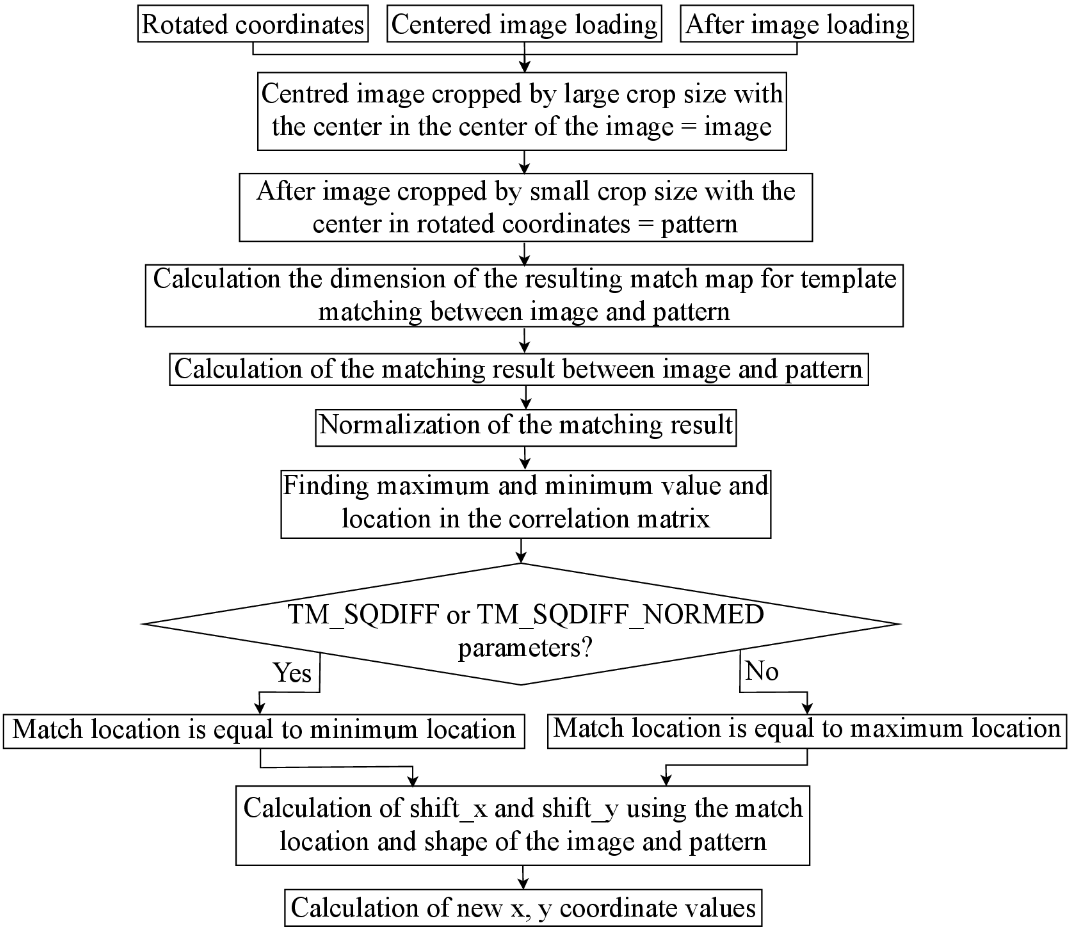

Rotated coordinates
Centered image loading
After image loading
Centred image cropped by large crop size with the center in the center of the image = image
After image cropped by small crop size with the center in rotated coordinates = pattern
Calculation the dimension of the resulting match map for template matching between image and pattern
Calculation of the matching result between image and pattern
Normalization of the matching result
Finding maximum and minimum value and location in the correlation matrix
TM_SQDIFF or TM_SQDIFF_NORMED parameters?
Yes
No
Match location is equal to minimum location
Match location is equal to maximum location
Calculation of shift_x and shift_y using the match location and shape of the image and pattern
Calculation of new x, y coordinate values

Fig. 2.12: Retrieving colony position workflow diagram.

2.2.3 Benchmarking current state

This section will discuss the results of the original unimproved algorithms used on the test dataset.

I created two distinct test datasets; one for evaluating dish detection accuracy and another for assessing dish rotation and colony position retrieval performance.

The test dataset for dish detection consists of images captured using the MBT Pathfinder . It includes 30 images without dishes and 26 images featuring dishes. These selections were meticulously chosen from a total of 2759 samples, ensuring the representation of diverse agar types, microorganism variations, dishes without microorganisms, and dishes captured under various illumination conditions. Examples of these samples are depicted in Figure 2.13. The examples of instances, where the dishes are not present are in Figure 2.14.

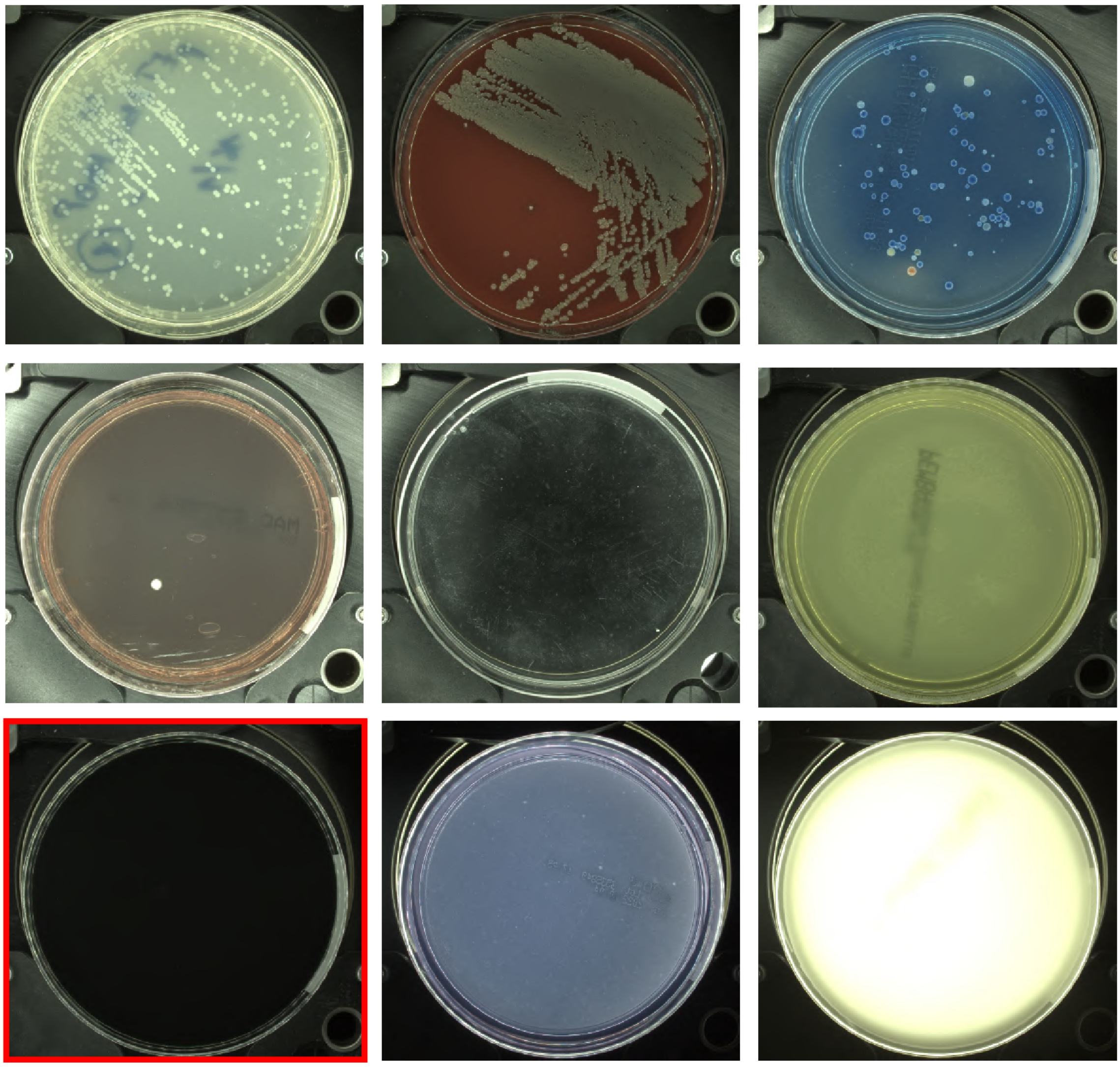


Fig. 2.13: Examples of diverse dishes from the test dataset used to evaluate dish detection performance, highlighting the variety of samples analyzed in the experiments. In the lower-left corner is an example of a dish with CCDA agar, which could be problematic for detection due to its resemblance to an empty dish.

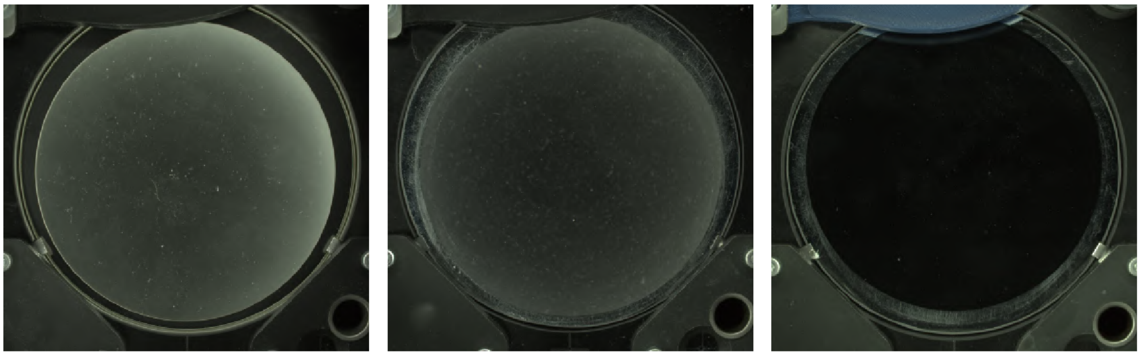


Fig. 2.14: Examples of various instances of dish absence in the dish detection test dataset.

The test dataset for evaluating dish rotation and colony position retrieval performance was established by selecting 40 samples out of the 3654 available. These samples represent both standard ones processed with the instrument and a few that had previously presented challenges. This test dataset was designed to include a wide variety of behaviors that can occur during sample processing by the MBT Pathfinder instrument. Examples of dishes from the test dataset are illustrated in Figure 2.15.

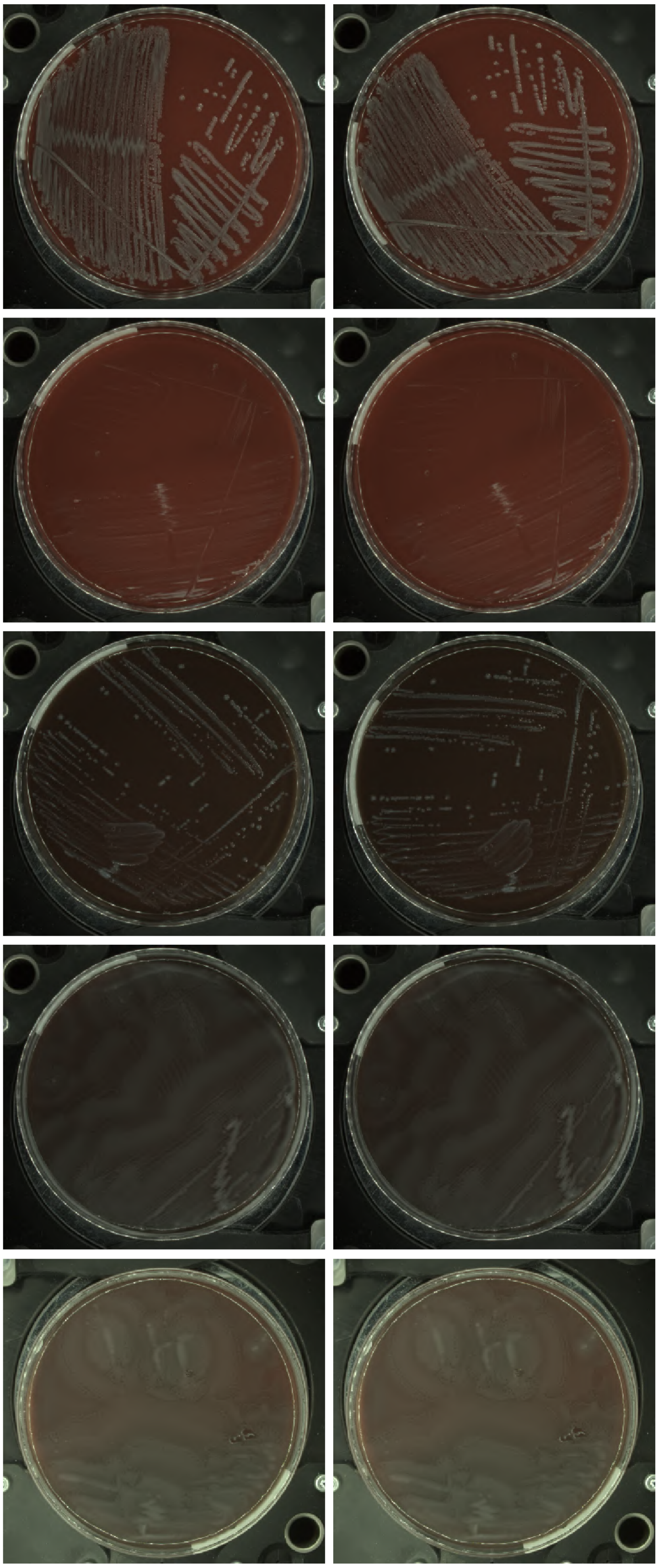
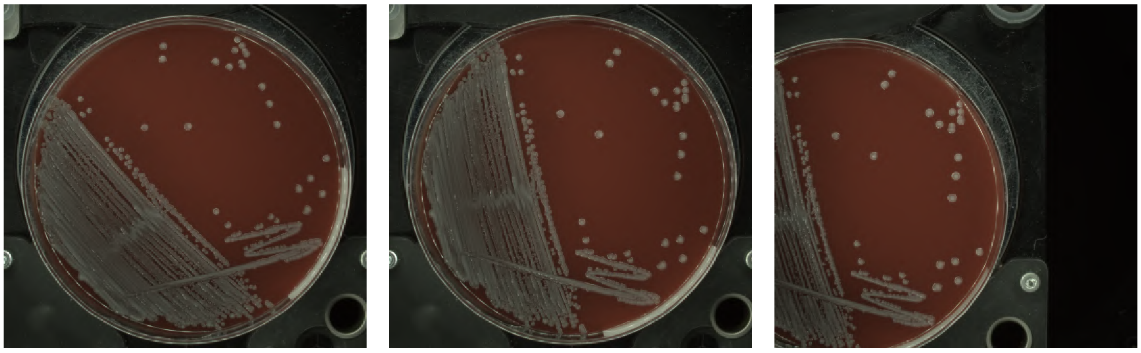


Fig. 2.15: Examples of a variety of dishes in the test dataset, illustrating the diversity of samples analyzed in the experiments.

Each sample contains 3 images:

- *Before image* – This image is captured after initially inserting the dish into the device. It serves as a preview for the user, showing the colonies present on the dish. The user can select the colonies from this image and confirm the selection. It's crucial to note that this step is the only hands-on user interaction with the device. After this, the process continues autonomously.
- *After image* = This image is taken after reinserting the dish into the device. At this stage, dishes are placed back into the device for colony picking. The image is rotated due to the design of the positioning system.
- *Centered image* = This image is the *After image* with the potentially correct colony in the center – due to camera distortion and imperfect hardware positioning system, the dish displacement may be inaccurate, i.e. the selected colony may not be completely centered. The potentially correct colony is centered for easier positioning of the picking wire.

These images are depicted in Figure 2.16.



(a) Before image (b) After image (c) Centered image

Fig. 2.16: Examples of a variety of dishes in the test dataset, illustrating the diversity of samples analyzed in the experiments.

One of the main contributions of this thesis is identifying and measuring undefined instrument behavior that can cause errors. For this purpose, I have developed and performed several experiments. The Section 4 provides detailed descriptions of each experiment.

Dish detection

The original model underwent training using images of microbial colonies exclusively on red, white, and transparent agars. Furthermore, these images went through resizing to a dimension of 600x600, due to speeding up the model training and subsequent reduction of the prediction time. The accuracy of this model applied to the test dataset is **0.982**, and the prediction time of the model takes **94.8 ms**. The results of applying this model to the test dataset are presented in Table 2.1.

There is a space for enhancement, not only in accuracy but also in prediction time. In the future, the processing of some key algorithms will be moved from the driver desktop to the firmware SBC (single board computer) embedded in the instrument. Thus, shrinking this model would greatly ease the migration and make the device more futureproof. Possible improvements and refinements of this problem are covered in Chapter 3.

Tab. 2.1: Table of results containing basic information about the original model.

Model	**Description**	**Prediction time (ms)**	**Accuracy**
1	Model trained on red, white, and transparent agars, and on empty dishes, resized to 600x600	94.8	0.982

Dish rotation

The results of the original algorithm for calculating dish rotation can be presented in Table A.1. There are 40 measurements, of which **10** dishes exhibit an angle deviation greater than 0.4 degrees. The angle 0.4 is a maximum deviation allowed by the company Bruker Daltonics GmbH & Co. KG, that stems from thorough validation experiments and risk analysis. The accuracy of the original algorithm is therefore **0.75**. Table 2.2 summarizes the statistical metrics for analyzing the differences between the measured and actual values of dish rotation.

The actual values of the dish rotation were obtained using an algorithm to create a fusion of the *After image* and the *Before image* rotated by a specified angle. Initially, I selected the calculated angle for the rotation. If the merged image was blurry or misaligned in any way, I adjusted the angle value until a sharper image was produced. The resulting angle was then considered as the actual value. An illustrative example of image fusion is provided in Figure 2.17.

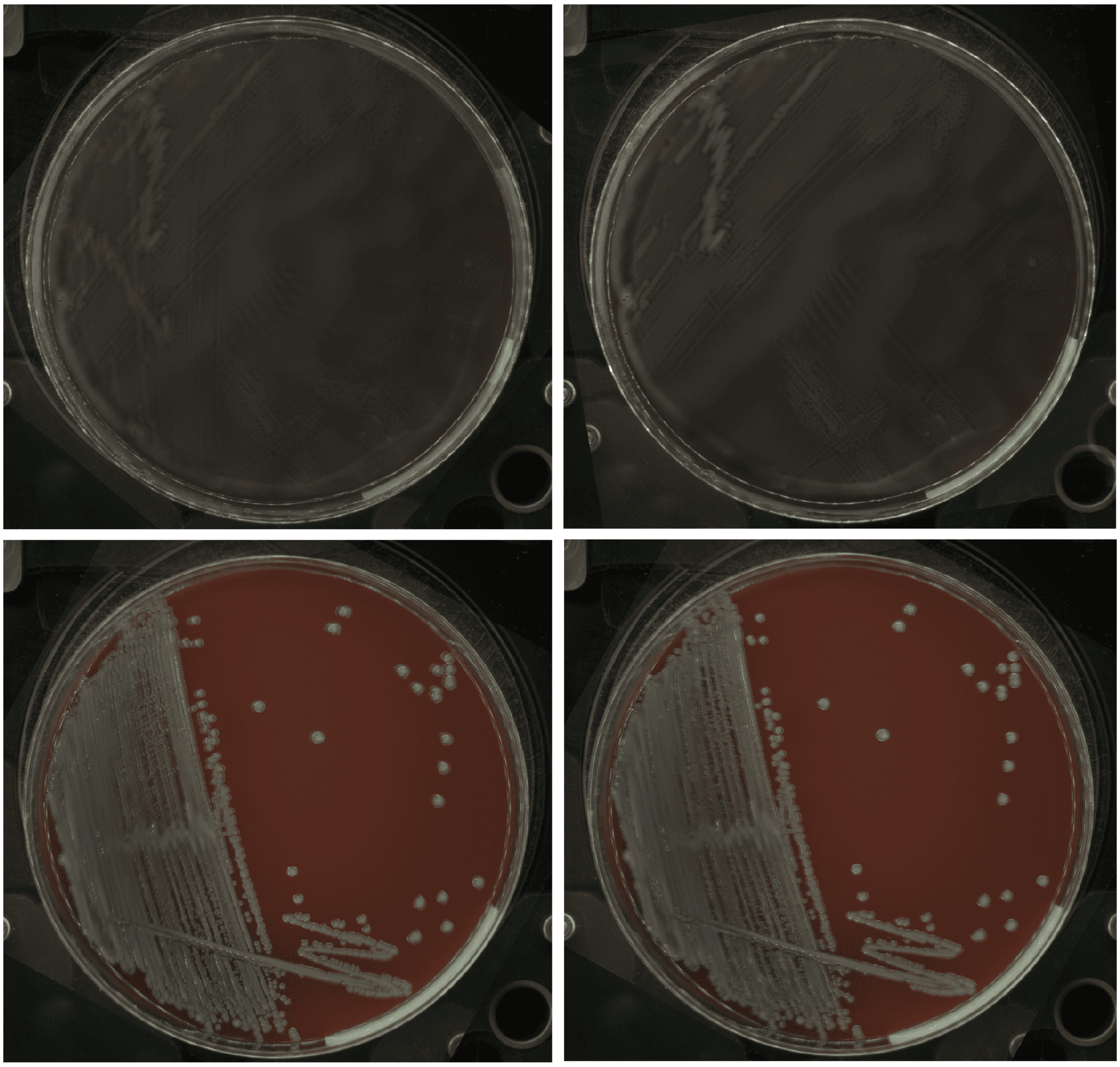


Fig. 2.17: Fusion of images: On the left is the fusion of images with the wrong angle, on the right with the true angle.

As we can see, the merging of images with the true angle is not as sharp as expected due to the dish being inserted into the machine not only at the wrong angle but also partially displaced. The fisheye effect from the camera also contributes to the distortion. This results in both rotational errors and translation errors. For this reason, there is room for improvement. Various possible approaches to this problem are discussed in Chapter 3.

In Table 2.2, the minimum value of the difference between the actual angle and the calculated rotation angle is **-0.6°**, and the maximum value is **40.8°**. To better understand the variability of these differences, I calculated the standard deviation, which is **6.36°**. Standard deviation measures the distribution of differences around the mean value of the differences. The Mean Squared Error (MSE) is **1.16°**, and the final metric mentioned herein is the mean absolute error (MAE), which has a value of **1.22°**. The MAE corresponds to the average of the absolutes of these differences.

It is important to note that the above statistical metrics are significantly affected by the entirely incorrect determination of the rotation angle of the dish "30-04-2021_2_UB_1011026280_D5". This dish showed significantly higher differences between measured and actual rotation values, compared to the others, by tens of degrees.

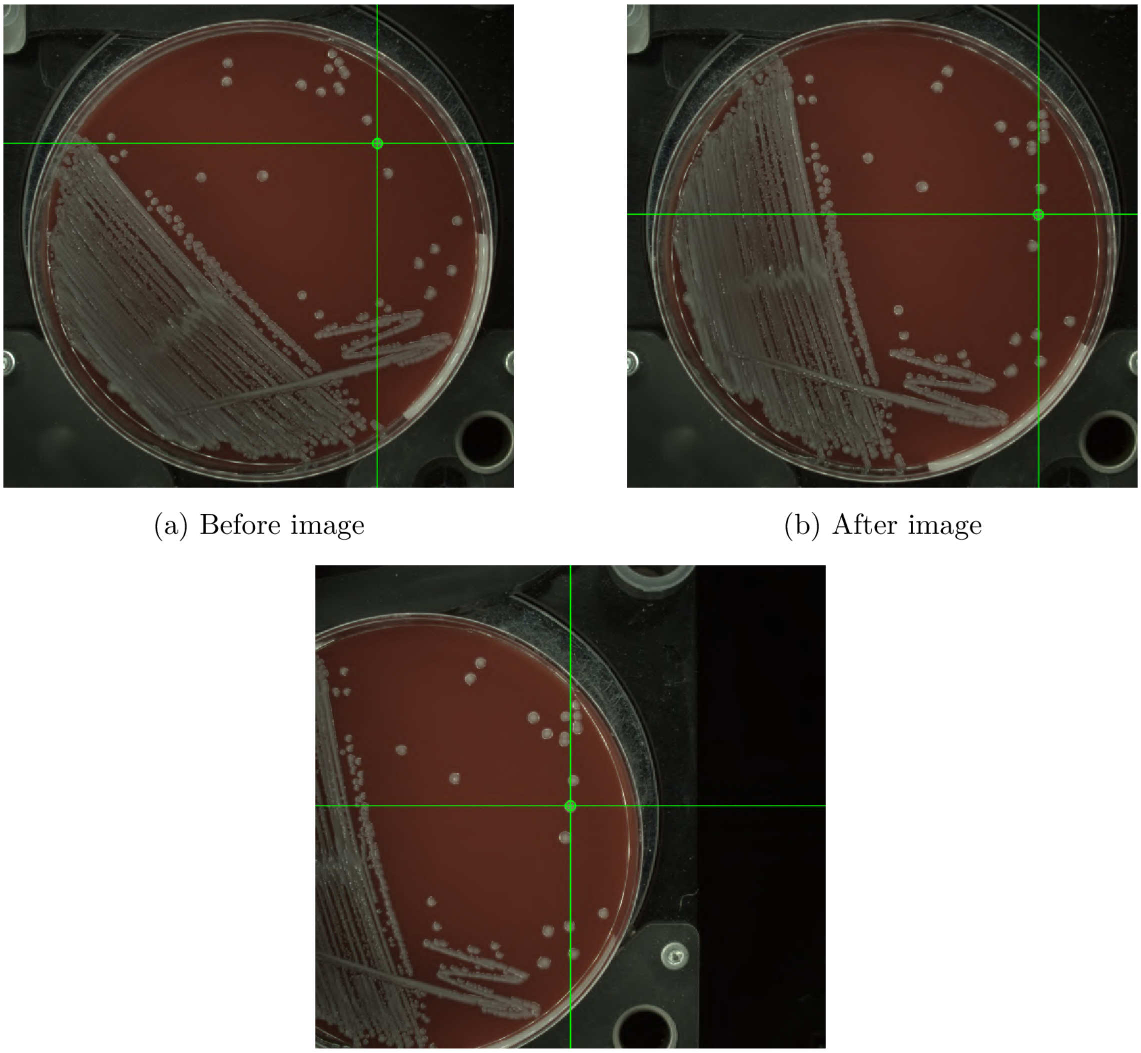
Possible improvements of this algorithm are discussed in Chapter 3.

Tab. 2.2: The table containing various metrics describing the difference between measured and true values of dish rotation.

Metrics	[°]
Min Δ	-0.6
Max Δ	41.40
Standard deviation ↓	6.36
MSE ↓	1.16
MAE ↓	1.22

Retrieving colony position

The results of the original algorithm for retrieving the colony position are shown in Table A.1. The accuracy of this algorithm is **0.725**. Concretely, for **8** dishes the algorithm fails both in the angle of rotation detection and in colony position retrieving, and for **3** dishes only the determination of the position was inaccurate. Examples showcasing both accurate and erroneous determination of colony positions are depicted in Figure 2.18 and Figure 2.19.

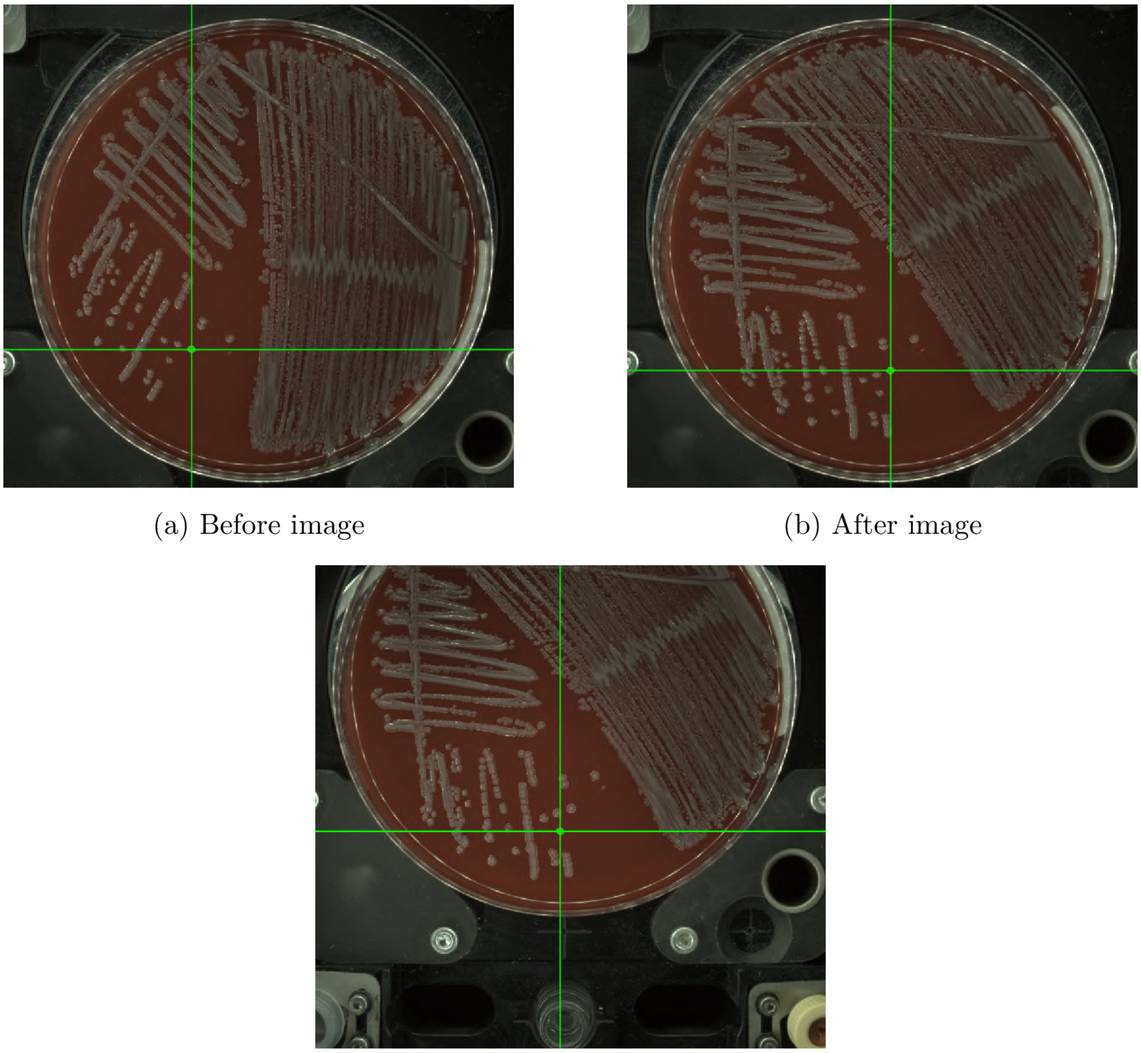


(a) Before image

(b) After image

(c) Centered image

Fig. 2.18: Example of an accurate colony position determination. In Subfigure c) it can be seen that the same colony was selected as the user selected in Subfigure a).



(a) Before image

(b) After image

(c) Centered image

Fig. 2.19: Example of an inaccurate colony position determination. In Subfigure c) it is evident that a different colony was picked than the one selected by the user in Subfigure a)

Retrieving the position of a colony is the most critical step in preparing samples. Therefore, it is crucial to improve the algorithm. Various approaches for enhancement are described in Chapter 3.

3 Refinement proposals and their rationale

This chapter delves into diverse strategies aimed at enhancing the efficacy of the original algorithms through a meticulous analysis of their respective advantages and drawbacks, accompanied by proposed solutions to potential shortcomings. Subsequent solutions to the individual approaches are described in Chapter 4.

3.1 Dish detection

As a first step, I plan to remove the application of random forest and change the architecture of the original model to be simpler. This could potentially improve the computational time.

Another approach could involve changing the input image size from the original 600x600 to 512x512. Reducing the input image could lead to increased computational efficiency. In addition, 512 is a power of 2, mainly to avoid rounding errors in the pooling operation that could lead to edge effects.

However, I am concerned about the possible loss of information with this resizing. Thus, an alternative strategy might be to refrain from simply shrinking the image, but rather consider cropping it or combining cropping with the resizing technique. I plan to crop the image at the point where the transition of the tray to the dish is most clearly visible. This could lead to preserving essential details while increasing the computational time.

Since the original model was only trained on red, white, and transparent agar and on empty dishes, the subsequent step I intend to pursue involves expanding the training, testing, and validation dataset to include dishes with different colors of agar. Figure 3.1 shows the low diversity of the original training dataset. I believe that this could contribute to increasing the model's capacity for generalization and thus its robustness.

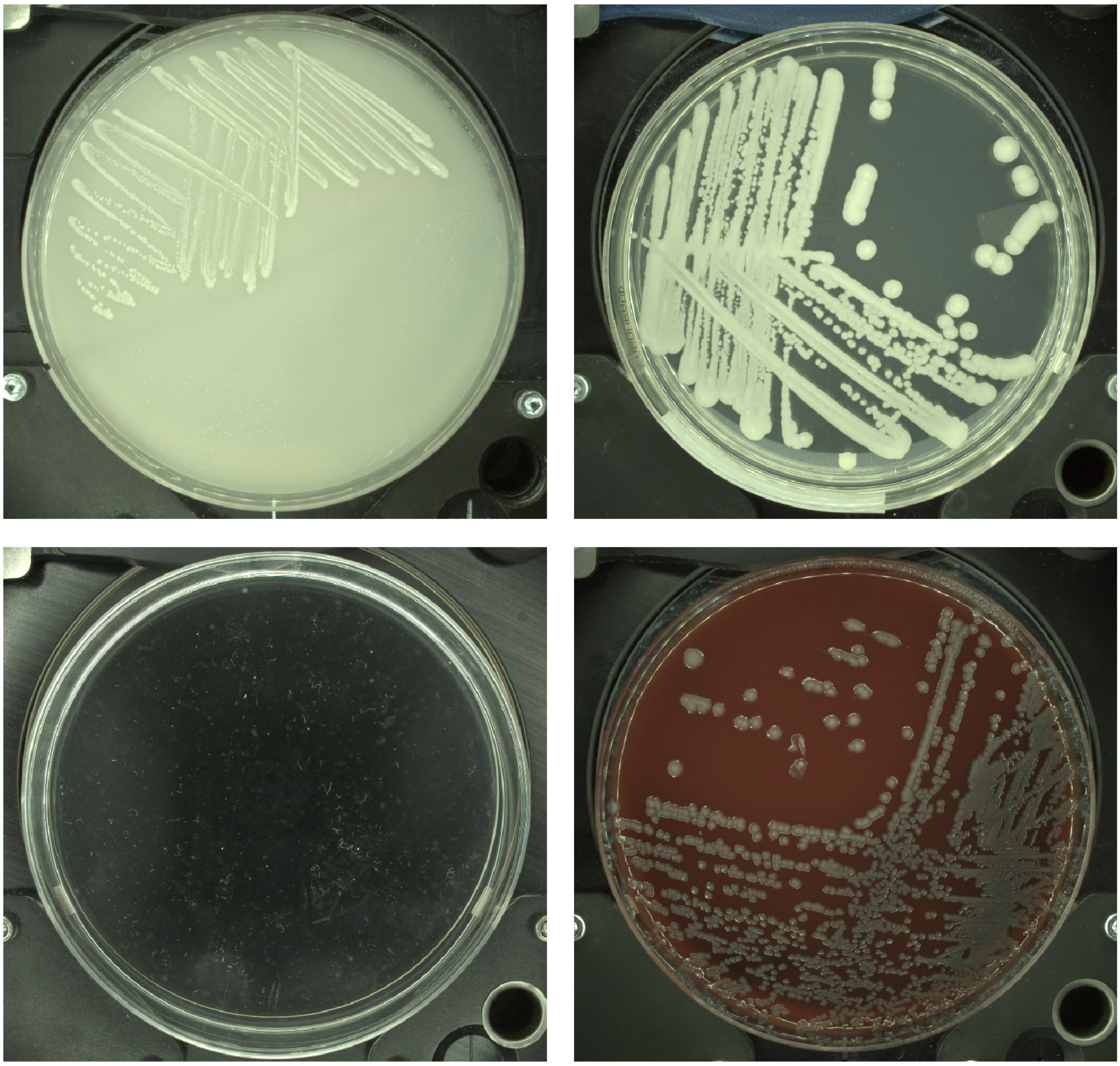


Fig. 3.1: Example of original train dataset for dish detection with minimal diversity.

3.2 Dish rotation

Since the algorithm for calculating the rotation of the dish uses a Canny detector to create masks, which are then compared using template matching, I decided to look at the outputs of this function, i.e. the created masks. I found that since the individual images are differently bright, the output mask sometimes contains an excessive number of detected edges or too few in certain cases. Thus, I think it might help to add conditions to change the parameters of the Canny function based on the number of white pixels found. This could solve the problem of inconsistent edge detection in output masks and thus improve the overall result of the following template matching.

If this approach does not lead to a significant improvement of the algorithm, it is possible to use some of the image registration methods. Since there is not only rotation of the dish, but also fish eye distortion caused by the camera, I believe that the optimal approach would be using the ORB detector to find key points in the image and then using the RANSAC method, which is often used for fish eye distortions, to estimate the transform based on the found key points and their corresponding matchings. This could lead to a more accurate determination of the rotation of the dishes.

3.3 Retrieving colony position

In order to increase the accuracy in calculating *Rotated coordinates*, I have decided to modify the algorithm. Instead of directly calculating the shift between the *After image*, cropped by a small mask with the center at the *Rotated coordinates*, and the *Centered image*, cropped by a large mask with its center at the image's center to obtain new *Centered coordinates*, I intend to introduce an additional step to enhance the accuracy of the calculated *Rotated coordinates*. This could be achieved by calculating the shift between the *Rotated image*, cropped by a small mask with its center at the *Rotated coordinates*, and the *After image* cropped by a large mask, also with its center at the *Rotated coordinates*. This process is visually represented for clearer understanding in Section 4.3.1.

Another approach that could help to improve the accuracy of retrieving the correct colony position is to apply a sharpening mask to the image and pattern before performing template matching. This could enhance the contrast and edges within the images, potentially leading to more precise template-matching results.

A further improvement in the accuracy of colony positioning could be achieved by increasing the size of the cropping masks. Thus, much more information about the surroundings of the selected colony could be obtained, and template matching could be much more accurate. However, increasing the cropping sizes also prolongs the computation time, so it is not ideal to greatly increase these values. A good tradeoff must be found between computational expansiveness and accuracy.

If the colony search algorithm proves to be successful, I intend to perform an additional experiment. This experiment will entail deliberately introducing a 1° error. 1° covers the vast majority of possible error conditions, virtually eliminating them. Manually subtracting this value from the calculated angle could confirm my theory of compensating for errors in dish rotation calculation through the colony position retrieval step. This would serve as a valuable backup plan in case the algorithm for calculating dish rotation fails.

4 Algorithm improvements

The following sections describe the implementations and improvements of the individual algorithms. All experimental results are discussed and summarized in the individual subsections.

4.1 Dish detection

This section addresses potential enhancements to the dish detection algorithm. Detailed descriptions of the individual experiments that contributed to the algorithm's enhancement are provided in the following subsections. Interim results are presented in the Table 4.1.

4.1.1 Experiment 1 – changing the architecture of CNN model

The first step that led to the improvement of the original algorithm for dish detection was to change the architecture of the model[1]. The new architecture is demonstrated in Figure 4.1. This resulted in an improvement in accuracy from **0.982** to **1** but a degradation in prediction time from **94.82 ms** to **96.61 ms**. This model holds the position designated as number 1 in Table 4.1.

[1] The model was taken from a web article [16].

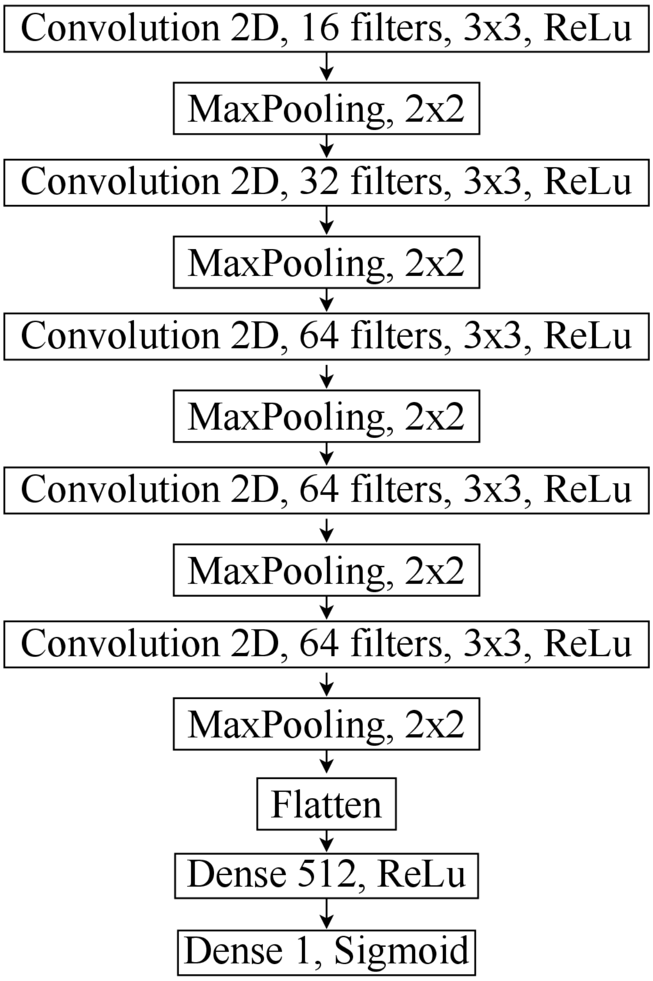

Convolution 2D, 16 filters, 3x3, ReLu
MaxPooling, 2x2
Convolution 2D, 32 filters, 3x3, ReLu
MaxPooling, 2x2
Convolution 2D, 64 filters, 3x3, ReLu
MaxPooling, 2x2
Convolution 2D, 64 filters, 3x3, ReLu
MaxPooling, 2x2
Convolution 2D, 64 filters, 3x3, ReLu
MaxPooling, 2x2
Flatten
Dense 512, ReLu
Dense 1, Sigmoid

Fig. 4.1: **The improved model** compared to the original one (Fig. 2.7) contains five convolutional layers (16, 32, 64, 64, 64) and uses the ReLU activation function. Unlike the original model, no explicit normalization layer is incorporated. Similar to the original model, the improved model employs Max Pooling with a 2x2 window. Additionally, the improved model includes two fully connected layers. The first layer is expanded to encompass 512 neurons, contrasting with the original model's 128 neurons, and is activated by the Rectified Linear Unit (ReLU) function. The second layer comprises a single neuron activated by a sigmoid function, maintaining consistency with the original model's architecture.

4.1.2 Experiment 2 – resizing the images to 512 x 512

In order to reduce the prediction time, the image size was changed from 600x600 to 512x512. The prediction time was thus reduced from **96.61** to **82.36 ms**, but at the expense of accuracy, which was decreased from **1.0** to **0.911**. The likely cause is the loss of necessary information about the surroundings of the selected colony. This is a problem because accuracy is much more important than reducing the prediction time. This model is assigned number 2 in Table 4.1.

4.1.3 Experiment 3 – cropping the image to 256 x 256

Since the resizing of the image lost the information needed for proper detection of the dishes, I decided to crop the image to a 256x256 window instead of resizing it. I decided to crop the image at the point where the transition of the tray to the dish is most clearly visible as shown in Figure 4.2. This resulted in an increase in accuracy from **0.911** to **0.964**, along with a significant improvement in prediction time, which decreased from **82.36** to **60.16 ms**. The model can be found in Table 4.1 under number 3.

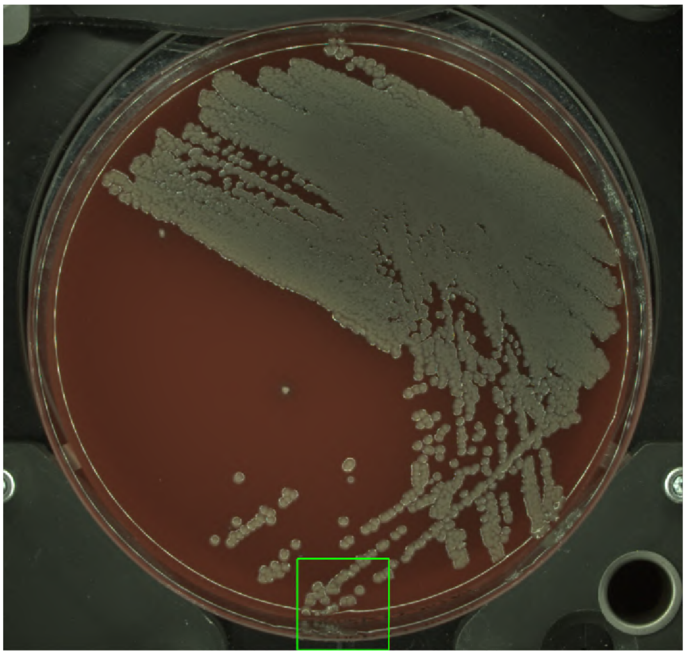


Fig. 4.2: Location of the crop window.

4.1.4 Experiment 4 – an extension of the dataset to include more color variations of agars

For the purpose of accuracy improvement, I created new train, test, and validation datasets. These datasets contain more color variations of dishes as can be seen in Figure 4.3. There was an improvement in prediction time from **60.16 ms** to **57.36 ms**, but no improvement in accuracy with such a trained model. This can be partially attributed to the previous practice of cropping images to 256 x 256 pixels. This problem is addressed in the following experiment.

Despite this, I believe that the expansion of the dataset to include new agar color variants has contributed to the generalizability of the model and its ability to perform better in practice, where different agar shades may occur that were not originally included in the training dataset. This model is listed in Table 4.1 under number 4.

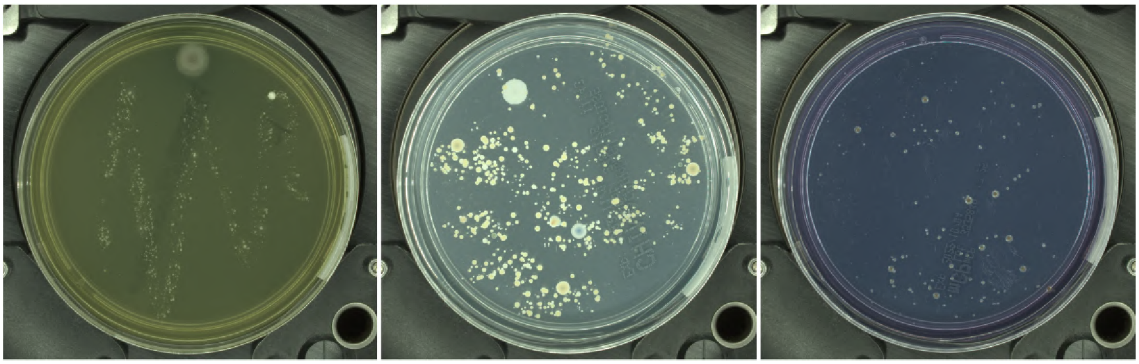


Fig. 4.3: Examples of different colors of agar.

4.1.5 Experimet 5 – resizing the image to 512x512 and cropping it to 256x256

The subsequent approach involved training the model on the extended dataset, initially reducing the input image to 512x512 and subsequently cropping it to 256x256. This procedure yielded an increase in accuracy from **0.964** to **0.982**. However, there was a minor degradation in prediction time from **57.36** to **57.92 ms**. This model is referenced as number 5 in the accompanying Table 4.1.

4.1.6 Experiment 6 – the extension of the dataset to include more dishes with bottom illumination

The final strategy was to expand the training, testing, and validation dataset by more dishes with bottom illumination. This decision was driven precisely by algorithm failures on these dishes. This resulted in an improvement in accuracy from **0.982** to **1.0**, and a reduction in prediction time from **57.92** to **54.69**. This model is denoted by 6 in Table 4.1.

Overall, experiment 6 led to an improvement in accuracy from **0.982** to **1.0** and to a reduction in prediction time from **94.82 ms** to **54.69 ms** compared to the original model. This final improvement is illustrated in the diagram 4.4.

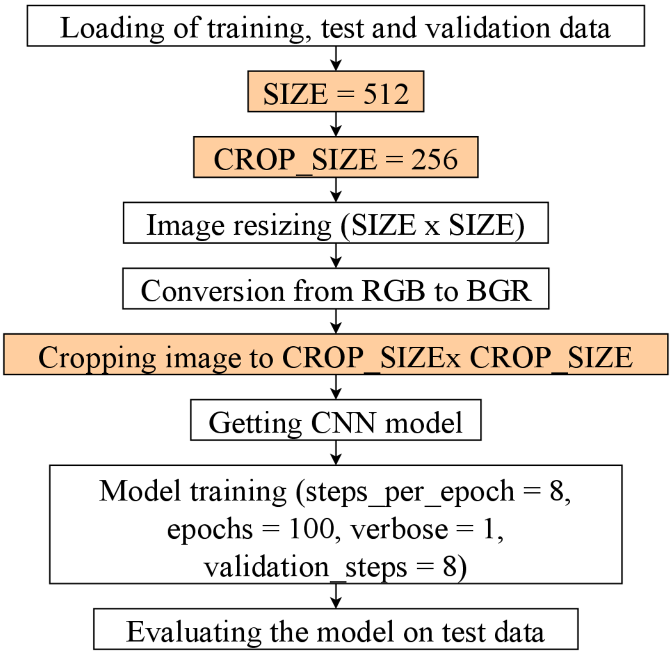

Loading of training, test and validation data
SIZE = 512
CROP_SIZE = 256
Image resizing (SIZE x SIZE)
Conversion from RGB to BGR
Cropping image to CROP_SIZEx CROP_SIZE
Getting CNN model
Model training (steps_per_epoch = 8, epochs = 100, verbose = 1, validation_steps = 8)
Evaluating the model on test data

Fig. 4.4: Improved CNN architecture: In comparison to the original architecture are images now resized to 512x512 instead of 600x600, and they are also cropped to 256x256. These changes are highlighted in the diagram.

Tab. 4.1: Table of results showing important parameters of individual models tested on the test dataset.

Model	**Description**	**Accuracy** ↑	**Prediction time (ms)** ↓
Base line	Model trained on red, white, and transparent agars, and on empty dishes, resized to 600x600	0.982	94.82
1	Model defined with different parameters = "New Model", trained on red, white, and transparent agars, and on empty dishes, resized to 600x600.	1.000	96.61
2	New Model trained on red, white, and transparent agars, and on empty dishes, resized to 512x512.	0.911	82.36
3	New Model trained on red, white, and transparent agars, and on empty dishes, cropped to 256x256.	0.964	60.16
4	New Model trained on more colors of agars, and on empty dishes cropped to 256x256.	0.964	57.36
5	New Model trained on more colors of agars, resized to 512x512, cropped to 256x256.	0.982	57.92
6	New Model trained on more colors of agars and more dishes with bottom illumination, resized to 512x512, cropped to 256x256.	**1.000**	**54.69**

4.2 Dish rotation

This subsection explores the implementation and testing of potential improvements to the dish rotation algorithm, previously discussed in Chapter 3. The results are summarized in Table A.2 in the Appendix.

The current improvement of the algorithm for sample preparation is shown in Figure 4.5.

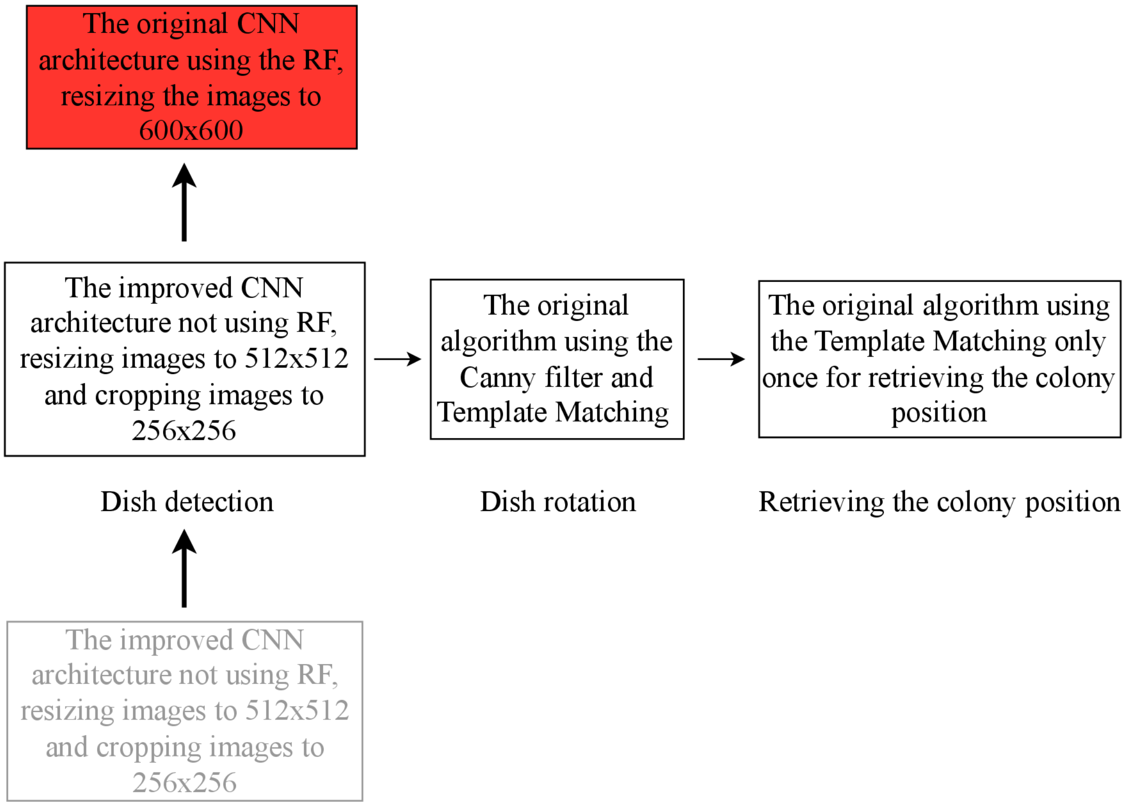

The original CNN architecture using the RF, resizing the images to 600x600
The improved CNN architecture not using RF, resizing images to 512x512 and cropping images to 256x256
The original algorithm using the Canny filter and Template Matching
The original algorithm using the Template Matching only once for retrieving the colony position
Dish detection
Dish rotation
Retrieving the colony position
The improved CNN architecture not using RF, resizing images to 512x512 and cropping images to 256x256

Fig. 4.5: Replacement of the original dish detection algorithm with a new improved one.

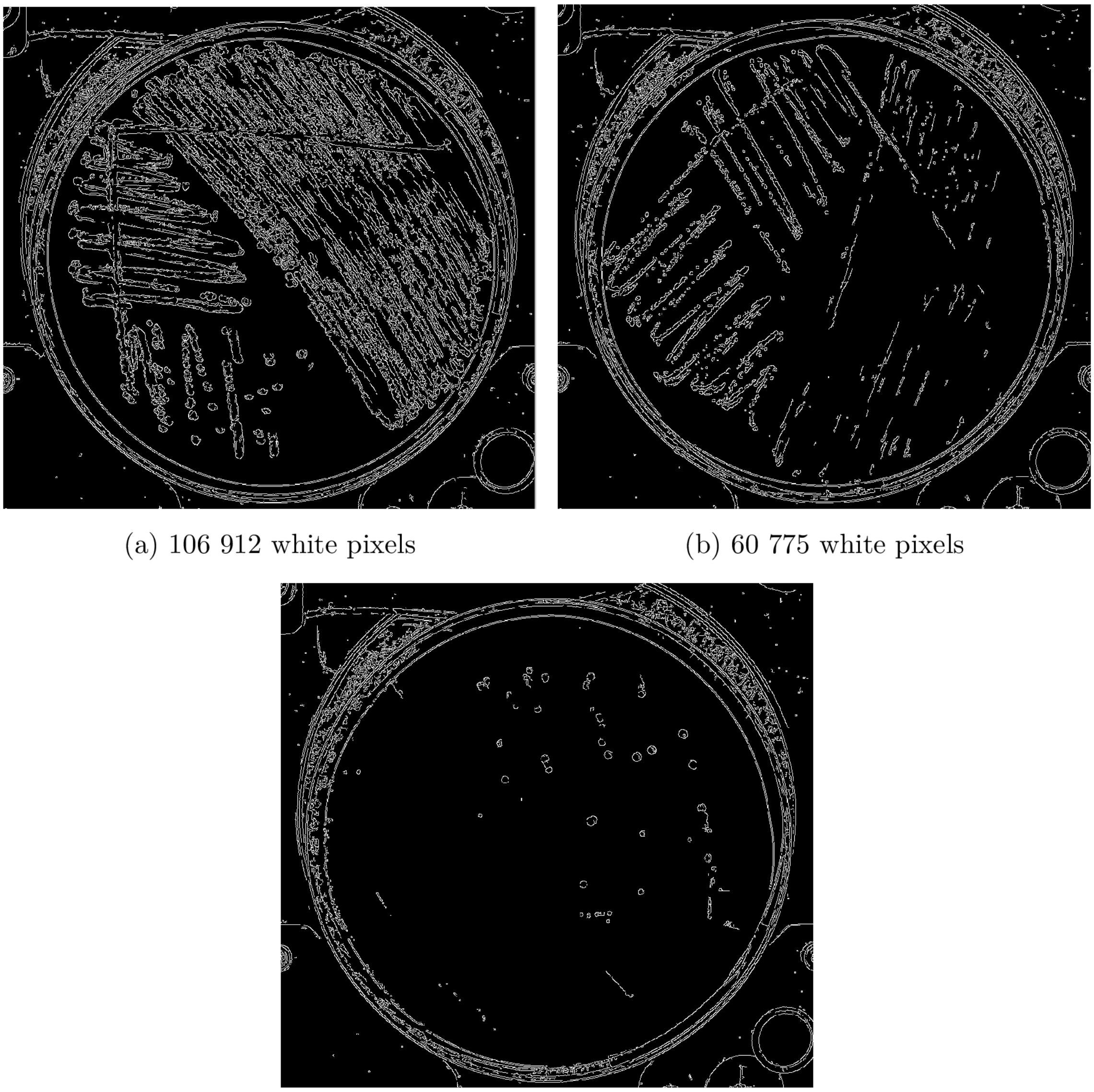
4.2.1 Experiment 1 – adding conditions for adjusting Canny function parameters based on image brightness

To detect the angle of rotation of the dish, the algorithm uses the **Canny** edge detection algorithm, which is generally used to detect edges in the image.

The problem occurs when the image is too dark, so the edges cannot be defined on it, or when the image is too bright and too many edges are detected on it. This is due to the fact that the dishes have different colors of agars and the MBT Pathfinder uses different types of illumination. Both of these scenarios can be seen in Figure 4.6. This is mainly due to the fact that the dishes have various agar colors and the MBT Pathfinder uses different types of illumination.

Therefore, a condition was added to the algorithm that decreases the lower and upper threshold limits if the number of white pixels is less than 43 000 and increases the threshold parameters if the number of white pixels is more than 90 000. The values of 43 000 and 90 000 have been empirically tested. The detailed workflow enhanced by these steps can be seen below 4.7.

This experiment led to an improvement in accuracy from **0.75** to **0.8**. The extended results are shown in Table A.2 in the Appendix.



(a) 106 912 white pixels

(b) 60 775 white pixels

(c) 30 047 white pixels

Fig. 4.6: Differences in the number of white pixels after edges detection. Image a) exhibits excessive brightness, leading to an abundance of detected edges. Conversely, image c) is too dark, so the edges cannot be defined exactly. These problems may consequently hinder the accurate computation of dish rotation.

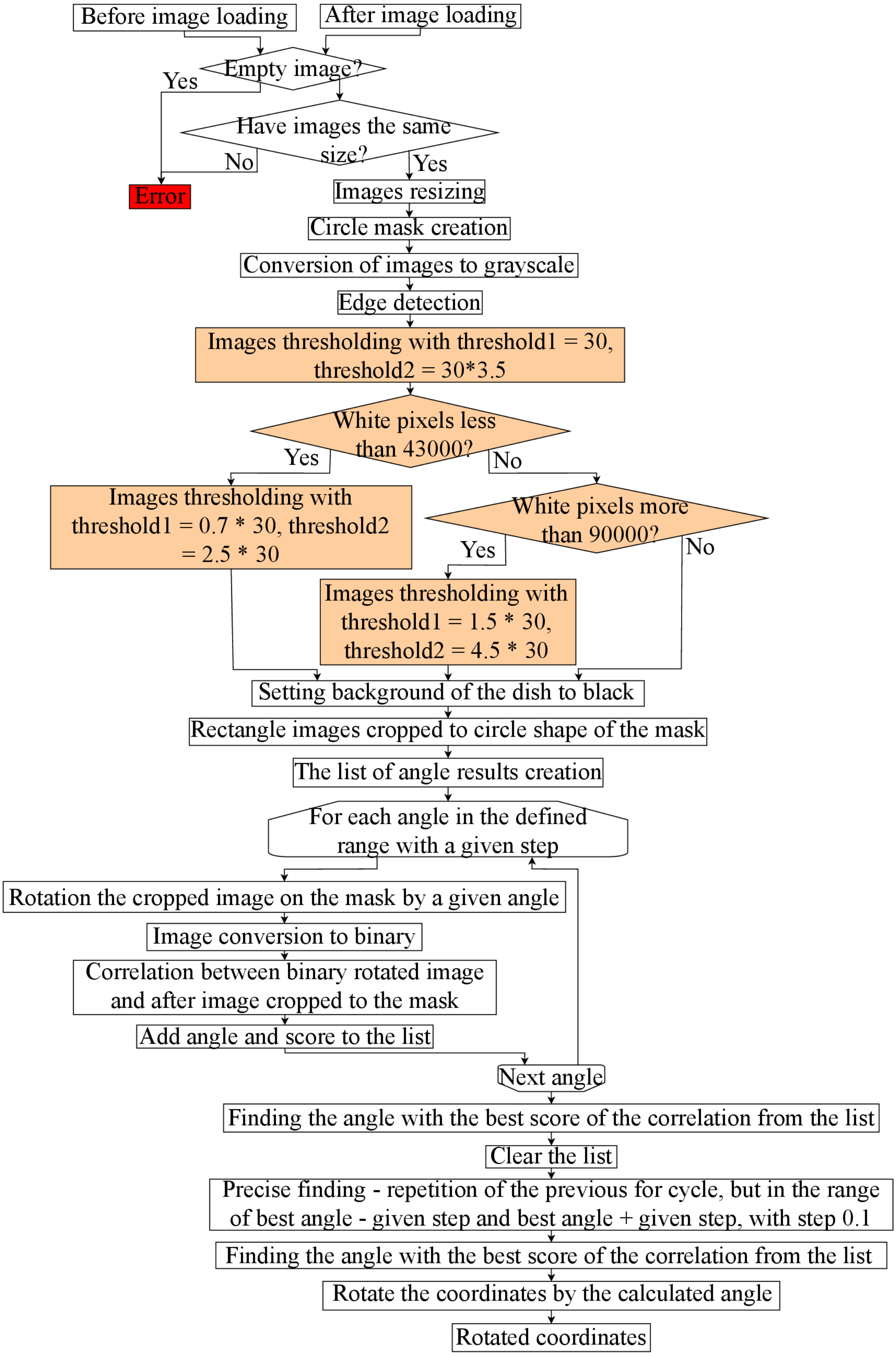

Before image loading
After image loading
Empty image?
Yes
Have images the same size?
No
Yes
Error
Images resizing
Circle mask creation
Conversion of images to grayscale
Edge detection
Images thresholding with threshold1 = 30, threshold2 = 30*3.5
White pixels less than 43000?
Yes
No
Images thresholding with threshold1 = 0.7 * 30, threshold2 = 2.5 * 30
White pixels more than 90000?
Yes
No
Images thresholding with threshold1 = 1.5 * 30, threshold2 = 4.5 * 30
Setting background of the dish to black
Rectangle images cropped to circle shape of the mask
The list of angle results creation
For each angle in the defined range with a given step
Rotation the cropped image on the mask by a given angle
Image conversion to binary
Correlation between binary rotated image and after image cropped to the mask
Add angle and score to the list
Next angle
Finding the angle with the best score of the correlation from the list
Clear the list
Precise finding - repetition of the previous for cycle, but in the range of best angle - given step and best angle + given step, with step 0.1
Finding the angle with the best score of the correlation from the list
Rotate the coordinates by the calculated angle
Rotated coordinates

Fig. 4.7: Dish rotation workflow diagram – improvement 1: This diagram highlights the additional steps adjusting the Canny detector threshold.

4.2.2 Experiment 2 – image registration

While there has been a slight improvement in the dish rotation algorithm, further enhancements are necessary to achieve the desired level of robustness. Therefore, I decided to use the methods of image registration and subsequent calculation of the rotation angle. For a better understanding, the workflow diagram of this algorithm can be seen in Figure 4.9.

This approach is based on searching for key points and descriptors using an ORB (Oriented FAST and Rotated BRIEF) detector. First, the images are converted to gray-scale and the ORB detector is initialized. The next step is to detect keypoints and compute descriptors using ORB. Comparator objects are created to compare the descriptors and find the corresponding matches between keypoints in the images. This can be seen in Figure 4.8. Subsequently, homography is used to transform one image to match the geometry of the other. The RANSAC method is applied to extract locations with reliable matches, filtering out outliers. Finally, using the transformation matrix obtained from homography and the arcus tangent function, the rotation angle between the images is calculated.

The extended results of this experiment can be seen in Appendix Table A.3. There has been a significant improvement in accuracy from **0.8** to **0.975**. The last 2 columns of the Table represent the calculated angles when using the SIFT (Scale Invariant Feature Transform) detector instead of the ORB. I also wanted to use the SURF detector, but this was not possible due to licensing restrictions. The choice of the ORB was correct since using SIFT did not lead to any improvement.

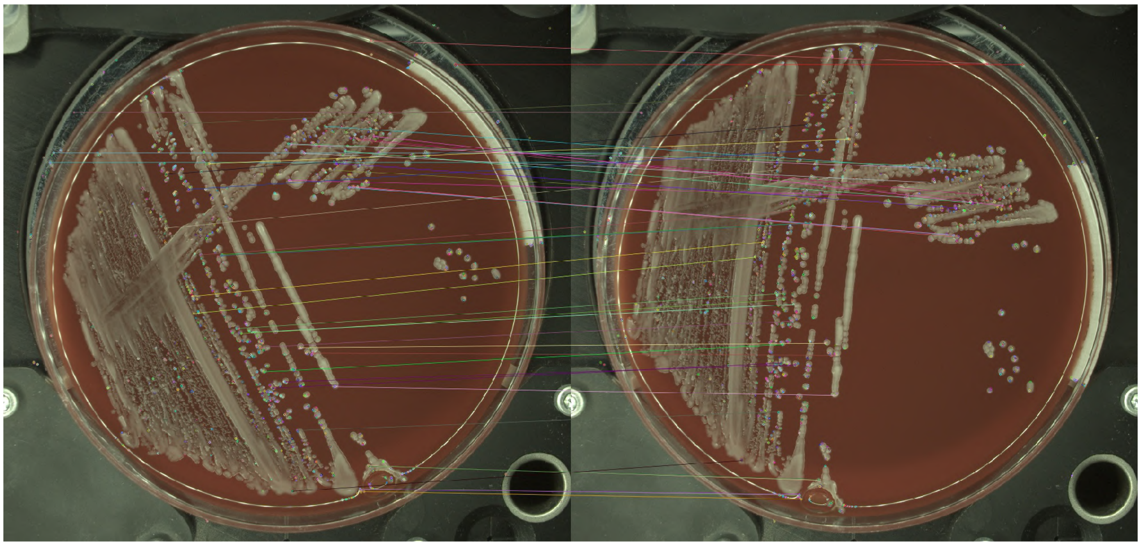


Fig. 4.8: Visualization of matched keypoints using ORB Detector.

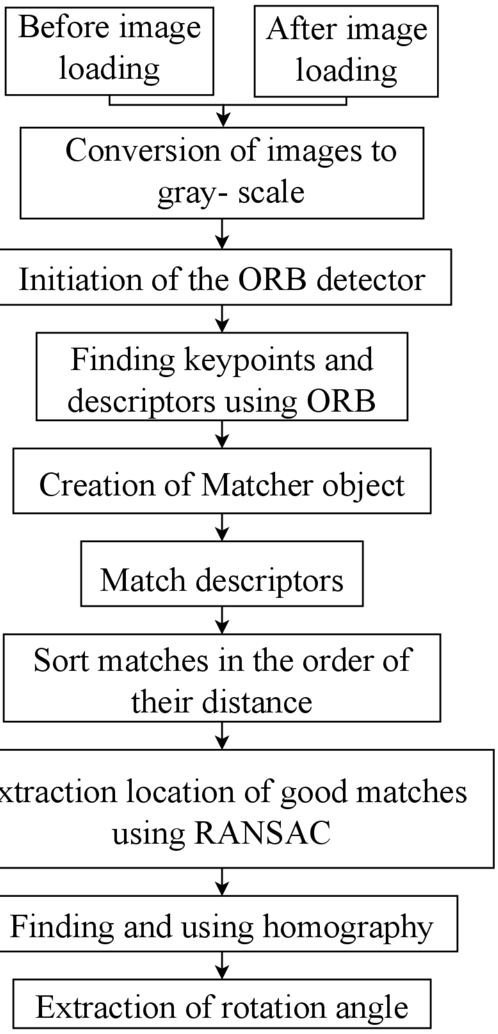

Before image loading
After image loading
Conversion of images to gray- scale
Initiation of the ORB detector
Finding keypoints and descriptors using ORB
Creation of Matcher object
Match descriptors
Sort matches in the order of their distance
Extraction location of good matches using RANSAC
Finding and using homography
Extraction of rotation angle

Fig. 4.9: Workflow diagram for calculating the rotation angle using image registration.

This experimental approach yielded a notable enhancement. The minimum value of the difference between the actual angle and the calculated rotation angle improved from **-0.6°** to **-0.2°**, and the maximum value significantly reduced from **40.8°** to **3.1°**. The standard deviation improved from **6.36°** to **0.49°**, the Mean Squared Error (MSE) decreased from **1.16°** to **0.08°**, and the Mean Absolute Error (MAE) decreased from **1.22°** to **0.12°**. The differences in these metrics are shown in Table 4.2.

Tab. 4.2: Improvement of various metrics describing the difference between measured and true values of dish rotation.

Metrics	**Original** [°]	**Improved** [°]	**Improvement** [°]
Min Δ	-0.6	-0.2	**-0.4**
Max Δ	40.8	3.1	**-37.7**
Standard deviation ↓	6.36	0.49	**-5.87**
MSE ↓	1.16	0.08	**-1.08**
MAE ↓	1.22	0.12	**-1.1**

4.3 Retrieving colony position

This section discusses several different approaches to improving the algorithm for retrieving colony position. The workflow diagram, supplemented by the steps discussed in the next chapter, can be seen below 4.16.

The results of the implementation of the original colony retrieving algorithm together with the improved algorithm for finding the rotation angle using image registration are shown in Table A.4. We can see that the improvement of the algorithm for finding the rotation angle has also led to a noticeable improvement in the accuracy of the algorithm for retrieving the correct colony position, from **0.725** to **0.85**.

The current improvement of the algorithm for sample preparation is shown in Figure 4.10.

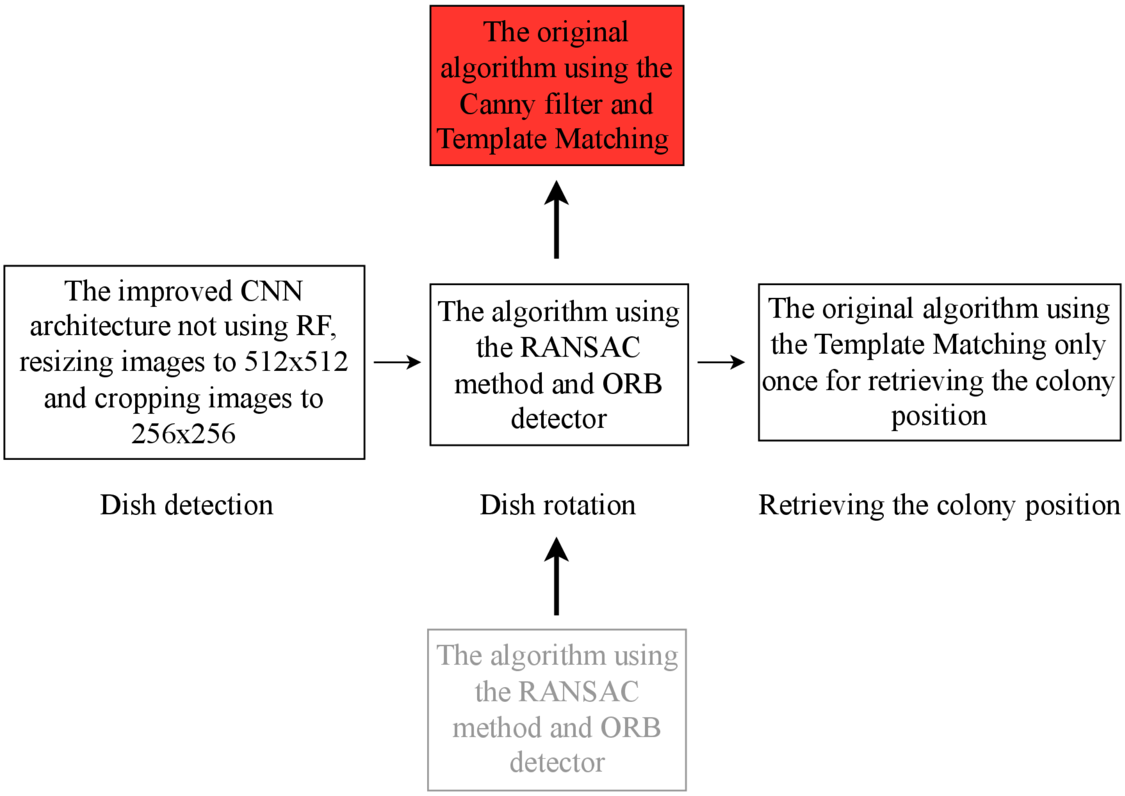

The original algorithm using the Canny filter and Template Matching
The improved CNN architecture not using RF, resizing images to 512x512 and cropping images to 256x256
The algorithm using the RANSAC method and ORB detector
The original algorithm using the Template Matching only once for retrieving the colony position
Dish detection
Dish rotation
Retrieving the colony position
The algorithm using the RANSAC method and ORB detector

Fig. 4.10: Replacement of the original dish rotation solution with a new improved one.

4.3.1 Experiment 1 – introducing additional correlation to increase the accuracy of Rotated coordinates

The first modification to the algorithm is adding a preliminary step that refines the position of the *Rotated coordinates* before proceeding to the final calculation of the coordinates of the selected colony.

This additional step involves calculating the offset between the pattern and image (search area) using the template matching method. The pattern is a *Rotated image* cropped by a small mask with the center in the *Rotated coordinates*. The image is the *After image* cropped by a large mask, also with the center in the *Rotated coordinates*. The resulting offset is then added to the *Rotated coordinates* to obtain the *New rotated coordinates*. These are then used to calculate the offset to find the correct coordinates of the selected colony.

To calculate the correct coordinates of the selected colony, the *After image* cropped by a small mask with the center in the *New rotated coordinates* is used as the pattern and the *Centered image* cropped by a large mask with the center in the *New rotated coordinates* is used as the image.

This process and the resulting improvement in the value of the *Rotated coordinates* is illustrated in Figure 4.11.

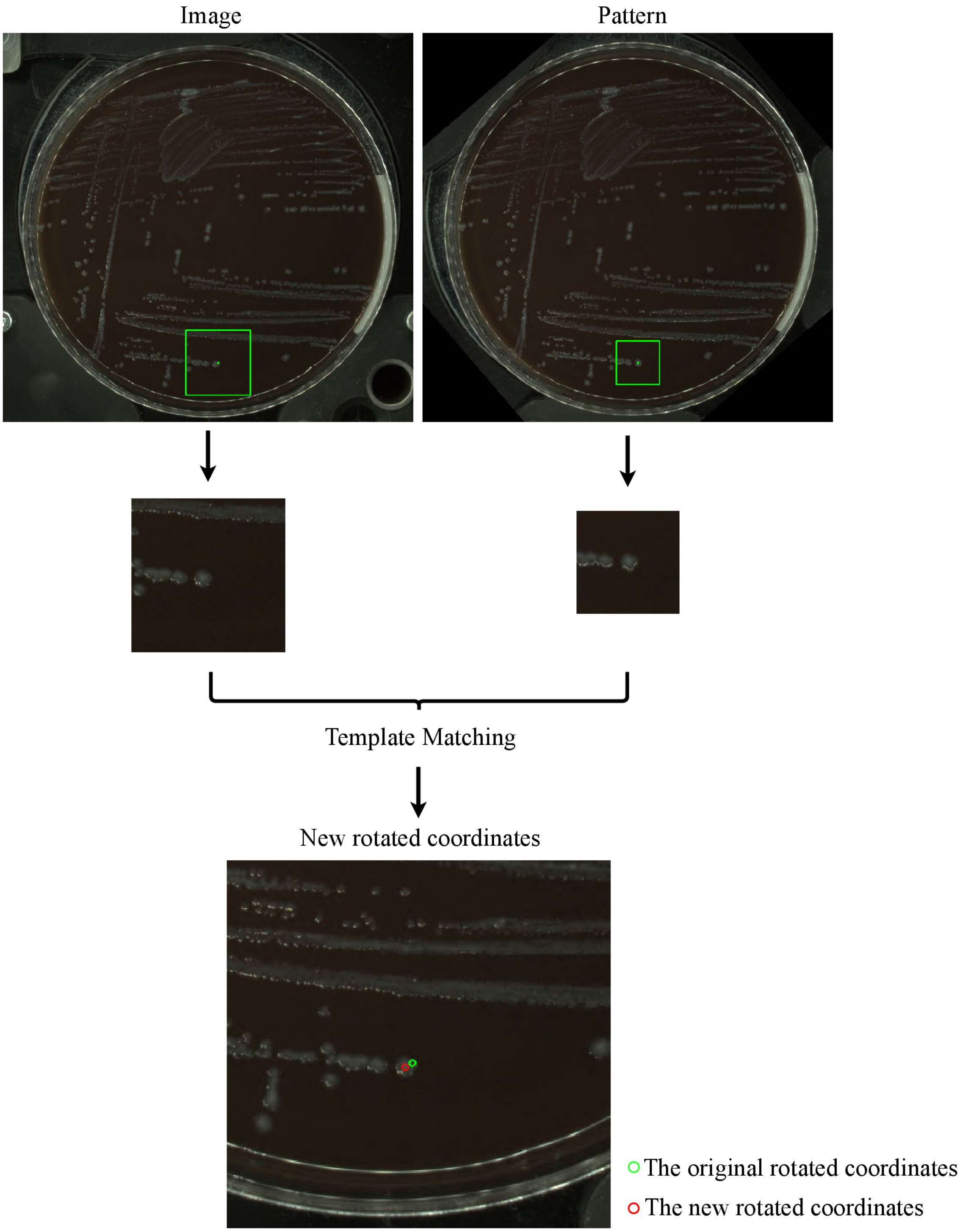


Image
Pattern
Template Matching
New rotated coordinates
○ The original rotated coordinates
○ The new rotated coordinates

Fig. 4.11: Improvement in the value of the *Rotated coordinates* by calculating the shift between the *Rotated image* cropped by a small mask with the center in the center of the image (Pattern) and the *After image* cropped by a large mask with the center also in the center of the image (Image).

This approach increases the precision of the calculated *Rotated coordinates* and thus the overall success rate of obtaining the correct colony position. Compared to the original algorithm without the improved calculation of the dish rotation, the improvement in accuracy is from **0.725** to **0.875**, and compared to the algorithm with the improved calculation of the dish rotation, the improvement is from **0.85** to **0.875**. The results are shown in Table A.5.

4.3.2 Experiment 2 – using a Kernel mask

The next approach is using the Kernel mask[2], specifically a sharpening mask (Fig. 4.12).

0	-1	0
-1	5	-1
0	-1	0

Fig. 4.12: Sharpening Kernel mask.

This mask is used on the image and the pattern before the shift is calculated. The result was a decrease in accuracy from **0.85** to **0.825**, which is likely due to the introduction of noise and making it more difficult to locate the pattern in the image accurately. The results of this experiment can be found in Table A.6.

4.3.3 Experiment 3 – changing the size of masks for template matching

Another experiment was changing the size of the masks used for template matching. Crop size small was changed from 92 pixels to 200 and crop size large was increased from 276 to 300 pixels. The difference in results of using the larger cropping mask instead of the smaller cropping mask is demonstrated in Figure 4.13. The results of this change can be seen in Table A.7.

[2] The convolution matrix used for sharpening, edge detection, embossing, and more [17].

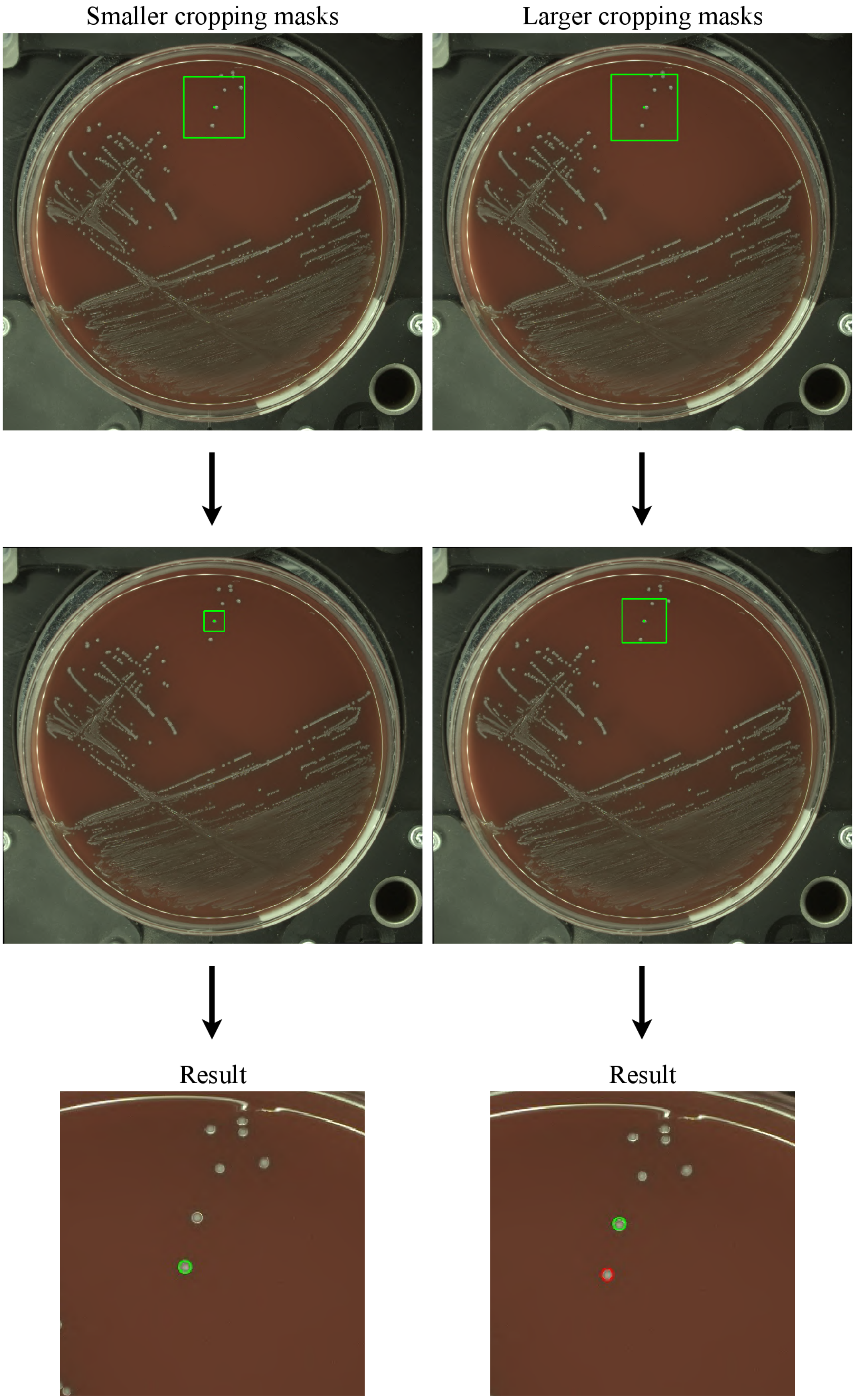


Fig. 4.13: The difference in results of using the larger cropping mask instead of the smaller cropping mask. The **green** mark represents the colony determined by the algorithm and the **red** mark represents the incorrectly selected colony by the algorithm using the smaller cropping mask.

This experiment resulted in the resolution of the majority of incorrect predictions. This is due to the fact that by increasing the cropping sizes the algorithm gained more information about the surroundings of the selected colony and could calculate the shift more accurately.

Increasing the cropping sizes prolongs the computation time from **2.30 ms** to **2.38 ms**, which is not so critical according to the improvement in accuracy in retrieving the colony position algorithm from **0.875** to **0.95**. The Table 4.3 shows the comparison of the computation times per image for different mask sizes.

Tab. 4.3: Table describing the comparison of computation times per image for different mask sizes.

Masks	**92, 276**	**200, 300**
Computation time per 1 image [ms]	2.30	2.38

The two erroneous forecasts that remain in the results are caused by the fact that the dataset used for this experiment consisted of images of dishes acquired by the device MBT Pathfinder before picking, after picking, and when the picked colony is centered. Since the previous algorithm most likely failed and thus incorrectly found the user-selected colony, and at a significant distance, the image for following template matching was incorrectly cropped and therefore does not contain the user-selected colony. This means that the subsequent correlation between the pattern and this image could not be performed correctly because these images are taken from completely different locations in the dish. This error is illustrated in the Figure 4.14. Due to this fact, I decided to exclude these 2 samples from the test dataset, so the overall improvement in accuracy is from **0.711** to **1.0**.

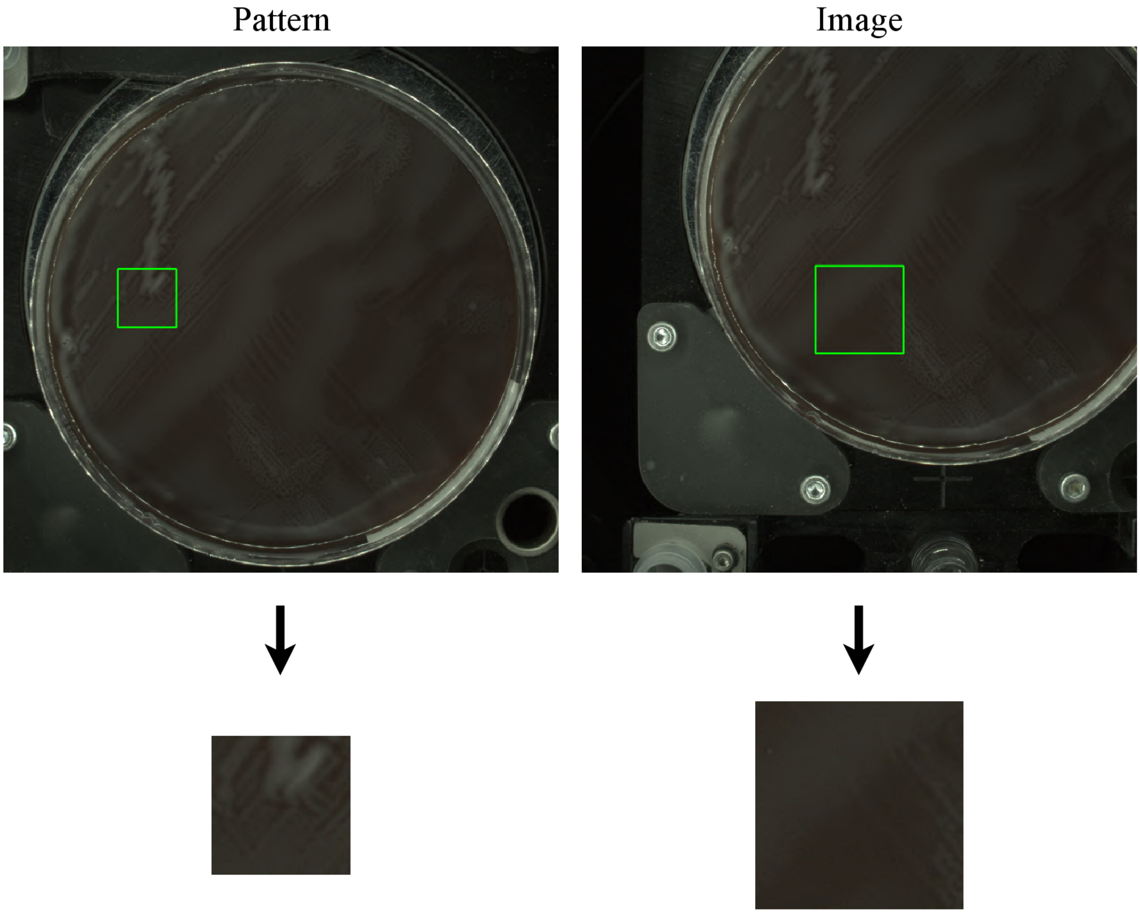


Fig. 4.14: Colony position retriever errors: The template matching between the pattern and the image cannot be performed correctly because these images are taken from completely different locations in the dish. This is due to a failure of the original algorithm.

4.3.4 Experiment 4 – artificial introduction of 1° error

Since the function for determining the rotation of the dish is still not guaranteed to work completely correctly, an experiment was performed to show that this inherent error can be compensated by an algorithm for obtaining the position of the colony.

In this experiment, I artificially introduced a 1-degree error. 1° covers the vast majority of possible error conditions, virtually eliminating them. First, we subtracted 1 degree from the calculated angle manually, and then the colony position was retrieved. The results of this experiment are identical to the results of the colony position retrieval algorithm without the added angle. After removing the 2 samples that were obtained incorrectly due to the failure of the MBT Pathfinder device, the accuracy is equal to **1.0**. These results are shown in the appendix in Table A.8.

We can therefore claim that the improvement of this algorithm was successful both in terms of refinement of the found colony position and in terms of improvement of the compensation of the error that arose during the search for the angle of rotation of the dish. An example of retrieving colony location improvement is illustrated in Figure 4.15.

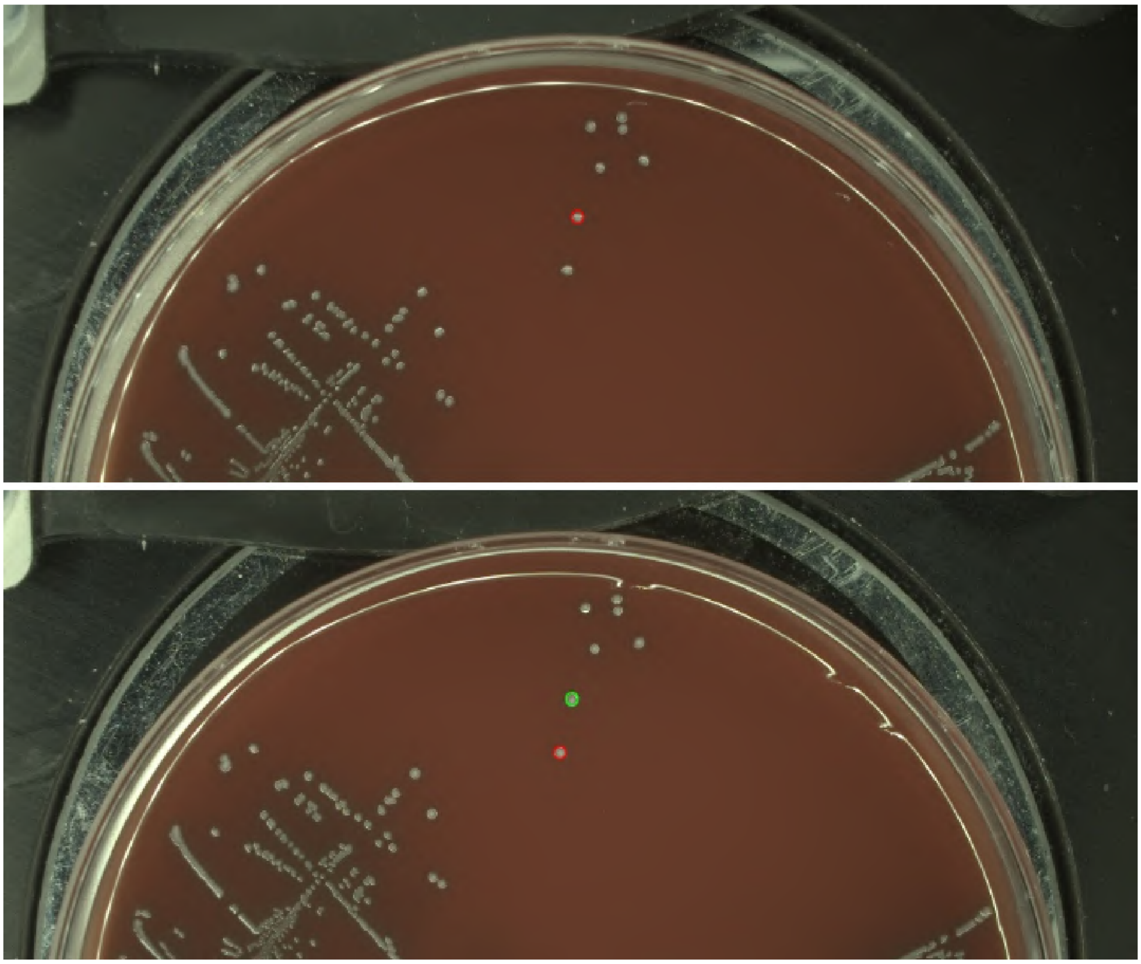
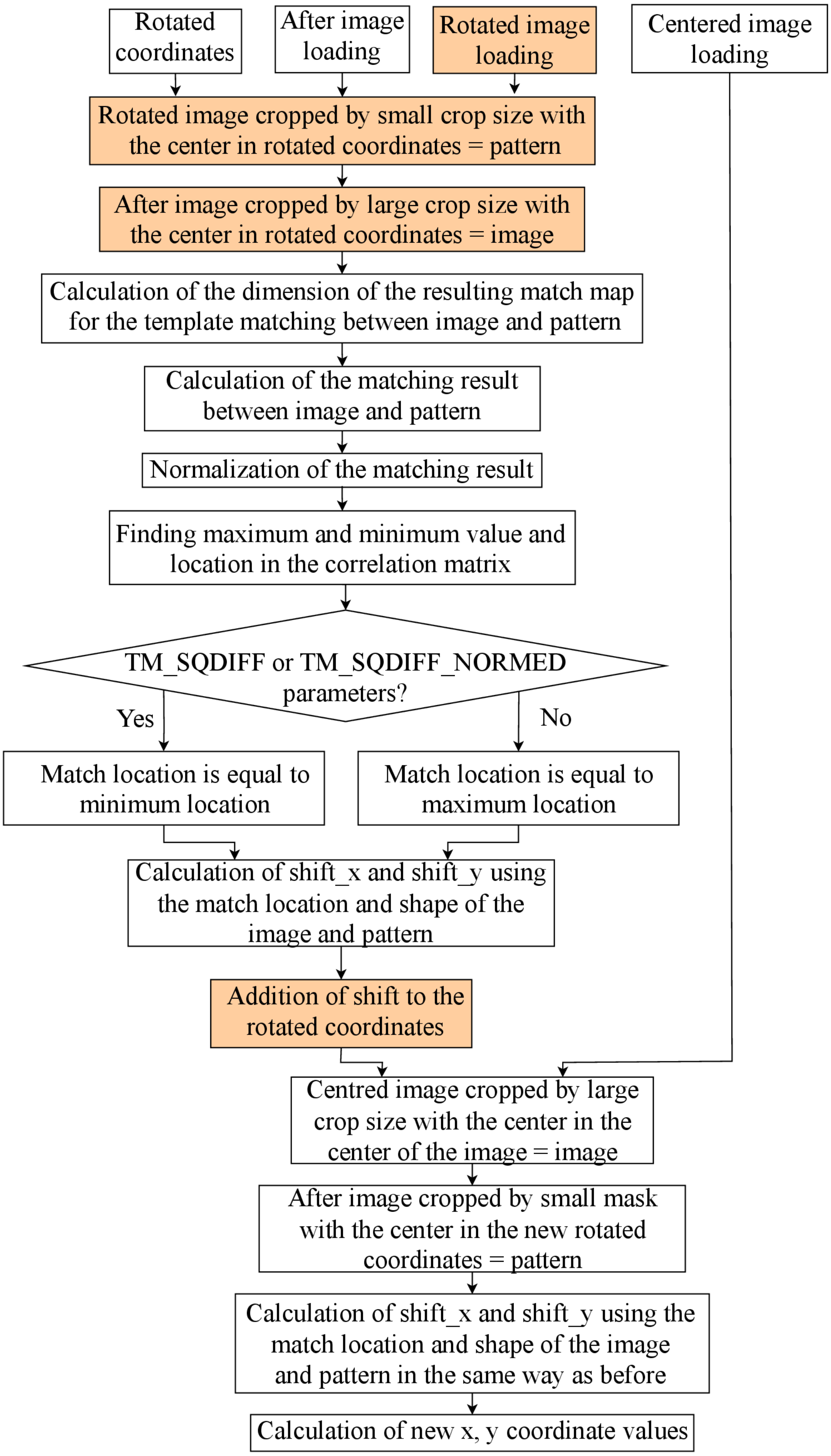


Fig. 4.15: Retrieving colony position: In the top Subfigure, the **red** marker indicates the colony selected from the label, which represents the colony selected by the user. In the bottom Subfigure, the **green** marker represents the colony found by the improved algorithm and the **red** marker represents the colony that was incorrectly determined by the original algorithm. It is evident that, compared to the original algorithm, the new algorithm finds the correct colony even when an artificial error of 1° is introduced.



Rotated coordinates
After image loading
Rotated image loading
Centered image loading
Rotated image cropped by small crop size with the center in rotated coordinates = pattern
After image cropped by large crop size with the center in rotated coordinates = image
Calculation of the dimension of the resulting match map for the template matching between image and pattern
Calculation of the matching result between image and pattern
Normalization of the matching result
Finding maximum and minimum value and location in the correlation matrix
TM_SQDIFF or TM_SQDIFF_NORMED parameters?
Yes
No
Match location is equal to minimum location
Match location is equal to maximum location
Calculation of shift_x and shift_y using the match location and shape of the image and pattern
Addition of shift to the rotated coordinates
Centred image cropped by large crop size with the center in the center of the image = image
After image cropped by small mask with the center in the new rotated coordinates = pattern
Calculation of shift_x and shift_y using the match location and shape of the image and pattern in the same way as before
Calculation of new x, y coordinate values

Fig. 4.16: Workflow diagram with highlighted additional steps.

Conclusion

This thesis represents a significant step towards more efficient and reliable operation of the MBT Pathfinder in microbiological analysis. The main objective was to increase the reliability and benchmarking performance of agar plate handling algorithms for laboratory automation robots.

The first chapter of my bachelor's thesis introduces the key algorithms and techniques necessary to tackle the problems of this thesis. It discusses topics such as digital image processing, provides a detailed overview of fundamental types of digital images, and elucidates basic image processing techniques such as object detection, image registration, and image segmentation. This chapter further explores Convolutional neural networks (CNNs), including a thorough description of their architecture, individual layers, and the CNN training process, including key concepts such as loss function and gradient descent. This chapter provides a solid foundation for understanding the image processing techniques and the complex workings of CNNs.

Chapter 2 addresses the microbial colonies, encompassing their cultivation, various types of nutrient media, classifications of agar, and the intricate processes associated with the propagation and growth of microorganisms. Subsequently, a comprehensive examination of the complexities associated with colony picking is conducted, with a detailed rationale highlighting the advantages of automation over manual methods. This chapter also provides the introduction and comprehensive description of the MBT Pathfinder instrument developed by Bruker Daltonics GmbH & Co. KG. A meticulous analysis of the instrument's entire workflow is conducted, with special attention devoted to elucidating the crucial steps within this workflow. Subsequently, the chapter delves into the creation of a test dataset and presents the results derived from implementing the original algorithms on this dataset. Simultaneously, the text addresses potential challenges and complexities that may arise in the application of these algorithms. It emphasizes the paramount importance of ensuring the seamless functionality of each step in the workflow, elucidating the reasons behind the necessity for flawless execution.

Chapter 3 discusses various strategies to improve the efficiency of the original algorithms. This is done through a detailed analysis of their advantages and disadvantages, complemented by suggestions to address any shortcomings. It offers a detailed description of my approach and explains the reasons behind each step I took.

Chapter 4 focuses on the implementation of the different approaches described in the previous chapter. All experiments that led to noticeable improvements are described in detail. The accuracy of the dish detection algorithm improved from **0.982** to **1.0** and the prediction time improved from **94.82 ms** to **54.69 ms**. For the dish rotation algorithm, the accuracy improved significantly from **0.750** to **0.975**. Finally, for the algorithm to obtain the correct colony position, the accuracy improved from **0.725** to **0.950**. However, it is important to mention that the dataset contained 2 samples that were incorrectly analyzed, probably due to the original algorithm failure. After removing these samples, the final accuracy of the algorithm on the test dataset was **1.0**. Finally, an experiment was performed in this chapter to confirm my theory that the angular rotation error can be compensated for by the colony position retrieval algorithm.

Overall, this work represents an advance toward improving the efficiency and reliability of the MBT Pathfinder instrument in microbiological analysis. The optimization and improvement of algorithms significantly affect the overall system performance and I believe that the solutions presented can bring benefits and progress in practical applications.

Bibliography

[1] ACHARYA Tankeshwar: *Bacterial Culture Media: Classification, Types, Uses.* [online]. Microbeonline. Last Updated: November 5, 2022. Retrieved from URL: `https://microbeonline.com/types-of-bacteriological-culture-medium/`.

[2] ACHARYA Tankeshwar: *Bacteriological Agar: Properties and Uses* [online]. Microbeonline. Last Updated: November 5, 2022. Retrieved from URL: `https://microbeonline.com/agar-properties-uses/`.

[3] ZHANG, Yilang; XING, Jian; CHEN, Guohua. Picking Path Optimization of Colony Picking Robot. In *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC).* [online]. 2021, 5(1), pp. 1109–1113. ISBN 978-1-7281-8535-4. ISSN 2045-2322. Retrieved from URL: `http://doi.org/10.1109/IMCEC51613.2021.9482259` [paywall].

[4] HUANG, Longlong; JIN, Zhenlin; ZHAO, Wenliang; ZHU, Wei; HE, Kai. A robot for quickly picking clone colony. In *3rd International Conference on Mechatronics, Robotics and Automation (ICMRA).* [online]. 2020, 5(1), pp. 94-98. ISBN 978-1-7281-9032-7. ISSN 2045-2322. Retrieved from URL: `http://doi.org/10.1109/ICMRA51221.2020.9398367`.

[5] DARPENTIGNY, Clementine; JEAN, Bruno. Wound causing agents. In: *Nanocellulose based-material for wound dressing* [online]. GlycoPedia. Retrieved from URL: `https://www.glycopedia.eu/e-chapters/nanocellulose-based-material-for-wound-dressing/article/wound-causing-agents`.

[6] WANG, Liyun; FAN, Daming; CHEN, Wei; TERENTJEV, Eugene M. Bacterial growth, detachment and cell size control on polyethylene terephthalate surfaces: a survey. *Scientific Reports*[online]. October 14, 2015, 5(1), 977-1000. ISSN 2045-2322. Retrieved from URL: `https://doi.org/10.1038/srep15159`. [cit. 2023-10-20].

[7] BrainKart.com. Growth and Multiplication of Bacteria In: *Microbiology and Immunology* [online]. Last Updated: October 14, 2015. Retrieved from URL: `https://www.brainkart.com/article/Growth-and-Multiplication-of-Bacteria_17831/`.

[8] MEHTA, Sourabh. What is image registration and how does it work? In: *Mystery Vault* [online]. Last Updated: May 7, 2022. Analyticsindiamag. Retrieved from URL: `https://analyticsindiamag.com/what-is-image-registration-and-how-does-it-work/`.

[9] ZITOVÁ, Barbara; FLUSSER, Jan. Image registration methods: a survey In: *Image and Vision Computing*, Volume 21, Issue 11, October 2003, pp. 977-1000, ISSN 0262-8856. Retrieved from URL: `https://doi.org/10.1016/S0262-8856(03)00137-9`. [paywall].

[10] TONDEWAD, Ms. Priyanka S. a DALE, Ms. Manisha P. Remote Sensing Image Registration Methodology: Review and Discussion In: *Procedia Computer Science*. 2020, pp. 2390-2399, Volume 171. ISSN 1877-0509. Retrieved from URL: `https://doi.org/10.1016/j.procs.2020.04.259`. [paywall].

[11] XIAOLONG DAI; KHORRAM, S. A feature-based image registration algorithm using improved chain-code representation combined with invariant moments In: *IEEE Transactions on Geoscience and Remote Sensing*. September 1999, pp. 2351-2362, vol. 37, no. 5. Retrieved from URL: `https://doi.org/10.1109/36.789634`. [paywall].

[12] MURRAY, Patrick R.; ROSENTHAL, Ken S.; PFALLER, Michael A. *Medical microbiology (8th ed.)*. Philadelphia: Elsevier, 2016. ISBN 978-0-323-29956-5.

[13] MADIGAN, M. T., MARTINKO, J. M., STAHL, D. A., CLARK, D. P. *Brock Biology of microorganisms (13th ed.)*. USA: PEARSON Education, 2010. ISBN 978-0-321-64963-8.

[14] JAN, Jiří. *Medical Image Processing, Reconstruction and Analysis: Concepts and Methods* Signal processing and communications. Boca Raton: CRC Press, 2005, ISBN 0-8247-5849-8.

[15] Doxygen *Template Matching* [online]. OpenCV.org Last updated: August 9, 2023. Retrieved from URL: `https://docs.opencv.org/3.4/de/da9/tutorial_template_matching.html`.

[16] BINH Phan. *10 Minutes to Building a CNN Binary Image Classifier in TensorFlow* [online]. https://towardsdatascience.com/ Last updated: July 7, 2020. Retrieved from URL: `https://towardsdatascience.com/10-minutes-to-building-a-cnn-binary-image-classifier-in-tensorflow-4e216b2034aa`.

[17] POWELL Victor. *Image Kernels* [online]. Setosa.io Retrieved from URL: `https://setosa.io/ev/image-kernels/`.

[18] GONZALEZ, Rafael C.; WOODS, Richard E. What is digital image processing. In: *Digital Image Processing fourth edition.* NY:Pearson Education Limited, 2018. ISBN 978-1-292-22304-9

[19] Simplilearn *What Is Image Processing : Overview, Applications, Benefits, and More* [online]. Last updated: August 23, 2023. Simplilearn.com. Retrieved from URL: `https://www.simplilearn.com/image-processing-article`.

[20] Ihritik *MATLAB | RGB image representation* [online]. Last updated: April 12, 2023. geeksforgeeks.org. Retrieved from URL: `https://www.geeksforgeeks.org/matlab-rgb-image-representation/`.

[21] KUNDU, Rohit. *Image Processing: Techniques, Types, & Applications [2023]* [online]. Last updated: August 3, 2022. v7labs.com. Retrieved from URL: `https://www.v7labs.com/blog/image-processing-guide`.

[22] University of Technology, Iraq Types of Digital Images In: *Image Processing* [online]. Last updated: August 3, 2022. uotechnology.edu.iq Retrieved from URL: `https://www.v7labs.com/blog/image-processing-guide`.

[23] MathWorks *What Is Digital Image Processing?* [online]. mathworks.com Retrieved from URL: `https://www.mathworks.com/discovery/digital-image-processing.html`.

[24] MathWorks *What Is Object Detection?* [online]. mathworks.com Retrieved from URL: `https://www.mathworks.com/discovery/object-detection.html`.

[25] YAMASHITA, Rikiya; NISHIO, Mizuho; DO, Richard Kinh Gian; TOGASHI, Kaori.(2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging* [online], 9(4), 611-629. ISSN 1869-4101. Retrieved from URL: `https://doi.org/10.1007/s13244-018-0639-9`.[paywall] [2023-10-05].

[26] NUZZI, Raffaele; BOSCIA, Giacomo; MAROLO, Paola; RICARDI, Federico. The Impact of Artificial Intelligence and Deep Learning in Eye Diseases: A Review. *Frontiers in Medicine* [online], 8. Last updated: August 30, 2021. Retrieved from URL: `https://doi.org/10.3389/fmed.2021.710329`. [2023-10-05].

[27] ALZUBAIDI, Laith; ZHANG, Jinglan; HUMAIDI, Amjad J.; AL-DUJAILI, Ayad; DUAN, Ye et al.. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* [online]. 2021, 8(1). ISSN 2196-1115. Retrieved from URL: `https://doi.org/10.1186/s40537-021-00444-8`. [cit. 2023-10-05].

[28] RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. Online. *Medical Image Computing and Computer-Assisted Intervention — MICCAI 2015*. Lecture Notes in Computer Science. 2015, 234-241. ISBN 978-3-319-24573-7. Retrieved from URL: `https://doi.org/10.1007/978-3-319-24574-4_28`. [cit. 2023-10-06].

[29] SCHAR-ZAMMARETTI, Prisca; DILLMANN, Marie-Lise; D'AMICO, Nicola; AFFOLTER, Michael; UBBINK, Job. Influence of Fermentation Medium Composition on Physicochemical Surface Properties of Lactobacillus acidophilus: Convolutional Networks for Biomedical Image Segmentation. *Applied and Environmental Microbiology*. Lecture Notes in Computer Science. [online] 2005, 71(12), 8165-8173. ISBN 978-3-319-24573-7. ISSN 0099-2240. Retrieved from URL: `https://doi.org/10.1128/AEM.71.12.8165-8173.2005`. [cit. 2023-10-06].

[30] Hudson Robotics. *Your Guide to Colony Picking Tools.* [online]. hudsonrobotics.com Last updated: June 20, 2023. Retrieved from URL: `https://hudsonrobotics.com/your-guide-to-colony-picking-tools/`. [cit. 2023-10-12].

[31] SHRESTHA, Asma: *Colony Picker: Principle, Protocol, and Advantages* [online]. Microbeonline. Last Updated: October 28, 2022. Retrieved from URL: `https://microbeonline.com/automated-colony-picker-principle-and-advantages/`.

[32] BAHETI, Pragati. *Activation Functions in Neural Networks [12 Types & Use Cases]* [online] v7labs.com Last Updated: May 27, 2021 Retrieved from URL: `https://www.v7labs.com/blog/neural-networks-activation-functions`. [cit. 2023-10-20].

[33] RITWICK, Roy. *Neural Networks: Forward pass and Backpropagation* [online] towardsdatascience.com Last updated: June 13, 2022 Retrieved from URL: `https://towardsdatascience.com/neural-networks-forward-pass-and-backpropagation-be3b75a1cfcc`. [cit. 2023-10-20].

[34] DO, Synho; SONG, Kyoung Doo; CHUNG, Joo Won. Basics of Deep Learning: A Radiologist's Guide to Understanding Published Radiology Articles on Deep Learning. Online. *Korean Journal of Radiology.* [online] 2020, 21(1). ISSN 1229-6929. Retrieved from URL: `https://doi.org/10.3348/kjr.2019.0312`. [cit. 2023-10-20].

[35] Srimandutta. *Thresholding-Based Image Segmentation* [online]. Geeksforgeeks. Last Updated: January 16, 2023. Retrieved from URL: `https://www.geeksforgeeks.org/thresholding-based-image-segmentation/`.

[36] FUKUSHIMA, Kunihiko. Neocognitron: A hierarchical neural network capable of visual pattern recognition. [Online] *Neural Networks*. 1988, s. 119-130. ISSN 08936080. Retrieved from URL: `https://doi.org/10.1016/0893-6080(88)90014-7`. [cit. 2023-12-01].

[37] AWATI, Rahul. *Convolutional neural network (CNN)* [online]. TachTarget. Last Updated: April, 2023. Retrieved from URL: `https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network`.

[38] XIN, Mingyuan and WANG, Yong. Research on image classification model based on deep convolution neural network. *EURASIP Journal on Image and Video Processing.* [online]. 2019. Retrieved from URL: `https://doi.org/10.1186/s13640-019-0417-8`. [cit. 2024-01-17].

[39] SHINOZUKA, M and MANSOURI, B. Synthetic aperture radar and remote sensing technologies for structural health monitoring of civil infrastructure systems. *Structural Health Monitoring of Civil Infrastructure Systems*[online]. 2009. Retrieved from URL: `https://doi.org/10.1533/9781845696825.1.114`. [cit. 2024-01-17].

[40] KARAMI, Ebrahim; PRASAD, Siva; SHEHATA, Mohamed. *Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images.* [online]. 2015. Retrieved from URL: `https://arxiv.org/ftp/arxiv/papers/1710/1710.02726.pdf` [cit. 2024-04-05].

[41] VINAY A.; RAO, Avani S.; SHEKHAR, Vinay S.; C., Akshay Kumar; MURTHY, K N Balasubramanya et al. *Feature Extractionusing ORB-RANSAC for Face Recognition.* [online]. 2015. Retrieved from URL: `https://doi.org/10.1016/j.procs.2015.10.068`. [cit. 2024-04-05].

[42] DING, Lijun and GOSHTASBY, Ardeshir. On the Canny edge detector. *Pattern Recognition.* [online]. 2001. Vol. 34, no. 3, pp. 721-725. Retrieved from URL: `https://doi.org/10.1016/S0031-3203(00)00023-6`. [cit. 2024-04-18].

Symbols and abbreviations

GPUs	Graphics Processing Units
ACF	Aggregate Channel Feature
HOG	Histograms of Oriented Gradien
CNN	Convolutional Neural Network
DL	Deep Learning
ML	Machine Learning
AI	Artificial Intelligence
ReLU	Rectified Linear Unit
MALDI	Matrix-Assisted Laser Desorption/Ionization
MAE	Mean Absolute Error
ORB	Oriented FAST and Rotated BRIEF
RANSAC	RANdom SAmple Consensus
SIFT	Scale Invariant Feature Transform

A Tables of results

A.1 Original algorithms

Tab. A.1: Table of results demonstrating the correctness of the rotation determination and the correctness of the colony detection.

Name	Angle [°]	True angle [°]	Δ [°]	Angle result	Detection result
01-05-2021 09_15_23_KM_1011021661_A1	-0.2	0	0.2	✓	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.8	-2.5	0.3	✓	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38.9	-37.8	1.1	✗	✗
01-05-2021 09_15_23_KM_1011021661_C6	-14.2	-14	0.2	✓	✓
01-05-2021 09_15_23_KM_1011021661_E8	-13.1	-12.7	0.4	✗	✗
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	✗
02-05-2021_UB_1011019967_A11	1	1	0	✓	✓
02-05-2021_UB_1011019967_B1	34.1	34.5	0.4	✗	✗
04-05-2021 11_29_44_KM_1011021661_A3	-1.6	-1.5	0.1	✓	✓
04-05-2021 11_29_44_KM_1011021661_A5	-4.5	-4.4	0.1	✓	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.8	-8.5	0.3	✓	✓
04-05-2021 11_29_44_KM_1011021661_C8	-7.6	-8	-0.4	✗	✓
04-05-2021 11_29_44_KM_1011021661_D6	-39.5	-40.1	-0.6	✗	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-33.3	-32.8	0.5	✗	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.8	1.7	-0.1	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.2	0.2	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-1	-0.8	0.2	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1.1	-1	0.1	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-43.2	-42.3	0.9	✗	✗
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	3	3.1	0.1	✓	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.8	1.9	0.1	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.2	-0.1	0.1	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3	3	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.6	0.6	0	✓	✗
18-05-2021 07_55_29_KM_1011021661_A8	1.4	1.4	0	✓	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.9	-0.9	0	✓	✗
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.7	1.6	-0.1	✓	✗
27-05-2021 07_55_54_KM_0021331037_D12	28.5	29.3	0.8	✗	✗
30-04-2021_2_UB_1011026280_D5	-50.5	-9.7	40.8	✗	✗
30-04-2021_2_UB_1011026280_F3	26.6	27.4	0.8	✗	✗
			Accuracy	0.75	0.725

A.2 Improved algorithms

Tab. A.2: Table of results of the algorithm determining the rotation of the dish using the template matching method. The values in the column True angle [°] were obtained using an algorithm to create a fusion of the **After image** and **Before image** rotated by a specified Angle [°]. The exact procedure is described in 2.2.3.

Name	Angle [°]	True angle [°]	Δ [°]	Angle result	Angle [°] after refactoring	Δ [°]	Angle result
01-05-2021 09_15_23_KM_1011021661_A1	-0.2	0	0.2	✓	-0.2	0.2	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.8	-2.5	0.3	✓	-2.8	0.3	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38.9	-37.8	1.1	✗	-38.9	1.1	✗
01-05-2021 09_15_23_KM_1011021661_C6	-14.2	-14	0.2	✓	-14.2	0.2	✓
01-05-2021 09_15_23_KM_1011021661_E8	-13.1	-12.7	0.4	✗	-13	0.3	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	2.3	0.0	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	0.2	-0.1	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	-16.7	0.0	✓
02-05-2021_UB_1011019967_A11	1	1	0	✓	1	0.0	✓
02-05-2021_UB_1011019967_B1	34.1	34.5	0.4	✗	34	0.5	✗
04-05-2021 11_29_44_KM_1011021661_A3	-1.6	-1.5	0.1	✓	-1.6	0.1	✓
04-05-2021 11_29_44_KM_1011021661_A5	-4.5	-4.4	0.1	✓	-4.5	0.1	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.8	-8.5	0.3	✓	-8.8	0.3	✓
04-05-2021 11_29_44_KM_1011021661_C8	-7.6	-8	-0.4	✗	-7.6	-0.4	✗
04-05-2021 11_29_44_KM_1011021661_D6	-39.5	-40.1	-0.6	✗	-39.5	-0.6	✗
05-03-2021 10_11_02_polished_cl_1011021701_A1	-33.3	-32.8	0.5	✗	-33.3	0.5	✗
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.8	1.7	-0.1	✓	1.8	-0.1	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.2	0.2	0	✓	0.2	0.0	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-1	-0.8	0.2	✓	-1	0.2	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	-0.2	-0.1	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1.1	-1	0.1	✓	-1.1	0.1	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-43.2	-42.3	0.9	✗	-43.2	0.9	✗
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	0.2	0.0	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	-1.8	0.0	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	-1.2	0.0	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	-0.2	0.0	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	3.2	0.0	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	3	3.1	0.1	✓	3	0.1	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	-3.6	0.0	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.8	1.9	0.1	✓	1.8	0.1	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.2	-0.1	0.1	✓	-0.2	0.1	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3	3	0	✓	3	0.0	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	0.2	0.0	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.6	0.6	0	✓	0.6	0.0	✓
18-05-2021 07_55_29_KM_1011021661_A8	1.4	1.4	0	✓	1.4	0.0	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.9	-0.9	0	✓	-0.9	0.0	✓
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.7	1.6	-0.1	✓	1.7	-0.1	✓
27-05-2021 27-05-2021 07_55_54_KM_0021331037_D12	28.5	29.3	0.8	✗	28.5	0.8	✗
30-04-2021_2_UB_1011026280_D5	-50.5	-9.7	40.8	✗	-9.6	-0.1	✓
30-04-2021_2_UB_1011026280_F3	26.6	27.4	0.8	✗	26.6	0.8	✗
						Accuracy	0.8

Tab. A.3: Table of results of the algorithm determining the rotation of the dish using the image registration.

Name	**Angle [°]**	**True angle [°]**	**△ [°]**	**Angle result**	**ORB Angle [°]**	**△ [°]**	**Angle result**	**SIFT Angle [°]**	**△ [°]**	**Angle result**
01-05-2021 09_15_23_KM_1011021661_A1	-0.2	0	0.2	✓	0.1	-0.1	✓	0	0	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.8	-2.5	0.3	✓	-2.6	0.1	✓	-2.6	0.1	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38.9	-37.8	1.1	✗	-38	0.2	✓	-37.8	0	✓
01-05-2021 09_15_23_KM_1011021661_C6	-14.2	-14	0.2	✓	-14	0	✓	-14	0	✓
01-05-2021 09_15_23_KM_1011021661_E8	-13.1	-12.7	0.4	✗	-12.8	0.1	✓	-12.8	0.1	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	2.3	0	✓	0	2.3	✗
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	0.1	0	✓	0.1	0	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	-16.7	0	✓	0	-16.7	✗
02-05-2021_UB_1011019967_A11	1	1	0	✓	1	0	✓	1	0	✓
02-05-2021_UB_1011019967_B1	34.1	34.5	0.4	✗	34.7	-0.2	✓	0	34.5	✗
04-05-2021 11_29_44_KM_1011021661_A3	-1.6	-1.5	0.1	✓	-1.5	0	✓	0	-1.5	✗
04-05-2021 11_29_44_KM_1011021661_A5	-4.5	-4.4	0.1	✓	-4.4	0	✓	-4.4	0	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.8	-8.5	0.3	✓	-8.5	0	✓	-8.5	0	✓
04-05-2021 11_29_44_KM_1011021661_C8	-7.6	-8	-0.4	✗	-8	0	✓	-8	0	✓
04-05-2021 11_29_44_KM_1011021661_D6	-39.5	-40.1	-0.6	✗	-40.1	0	✓	-40.2	0.1	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-33.3	-32.8	0.5	✗	-32.8	0	✓	-32.8	0	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.8	1.7	-0.1	✓	1.7	0	✓	1.7	0	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.2	0.2	0	✓	0.1	0.1	✓	0.1	0.1	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-1	-0.8	0.2	✓	-0.8	0	✓	-0.4	-0.4	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	-0.2	-0.1	✓	0	-0.3	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1.1	-1	0.1	✓	-1	0	✓	-1	0	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-43.2	-42.3	0.9	✗	-42.3	0	✓	0	-42.3	✗
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	0.2	0	✓	0	0.2	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	-1.8	0	✓	-1.8	0	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	-1.2	0	✓	-1.2	0	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	-0.2	0	✓	-0.1	-0.1	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	3.2	0	✓	3.2	0	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	3	3.1	0.1	✓	0	3.1	✗	0	3.1	✗
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	-3.6	0	✓	-3.6	0	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.8	1.9	0.1	✓	1.9	0	✓	2	-0.1	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.2	-0.1	0.1	✓	-0.1	0	✓	-0.1	0	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3	3	0	✓	3.1	-0.1	✓	3.1	-0.1	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	0.2	0	✓	0.1	0.1	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.6	0.6	0	✓	0.5	0.1	✓	0.2	0.4	✓
18-05-2021 07_55_29_KM_1011021661_A8	1.4	1.4	0	✓	1.5	-0.1	✓	1.4	0	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.9	-0.9	0	✓	-0.8	-0.1	✓	0.2	-1.1	✗
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.7	1.6	-0.1	✓	1.5	0.1	✓	1.5	0.1	✓
27-05-2021 07_55_54_KM_0021331037_D12	28.5	29.3	0.8	✗	29.3	0	✓	29.4	-0.1	✓
30-04-2021_2_UB_1011026280_D5	-50.5	-9.7	40.8	✗	-9.9	0.2	✓	0	-9.7	✗
30-04-2021_2_UB_1011026280_F3	0	27.4	27.4	✗	27.5	-0.1	✓	27.6	-0.2	✓
			Accuracy	**0.75**		**Accuracy**	**0.975**		**Accuracy**	**0.8**

Tab. A.4: Table of results demonstrating the implementation of the original colony retrieving algorithm together with the algorithm for finding the rotation Angle [°] using image registration.

Name	Angle [°]	True angle [°]	Δ [°]	Angle result	Detection result
01-05-2021 09_15_23_KM_1011021661_A1	0.1	0	-0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.6	-2.5	0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38	-37.8	0.2	✓	✗
01-05-2021 09_15_23_KM_1011021661_C6	-14	-14	0	✓	✓
01-05-2021 09_15_23_KM_1011021661_E8	-12.8	-12.7	0.1	✓	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	✓
02-05-2021_UB_1011019967_A11	1	1	0	✓	✓
02-05-2021_UB_1011019967_B1	34.7	34.5	-0.2	✓	✓
04-05-2021 11_29_44_KM_1011021661_A3	-1.5	-1.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_A5	-4.4	-4.4	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.5	-8.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_C8	-8	-8	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_D6	-40.1	-40.1	0	✓	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-32.8	-32.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.7	1.7	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.1	0.2	0.1	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-0.8	-0.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1	-1	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-42.3	-42.3	0	✓	✗
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	0	3.1	3.1	✗	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.9	1.9	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.1	-0.1	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3.1	3	-0.1	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.5	0.6	0.1	✓	✗
18-05-2021 07_55_29_KM_1011021661_A8	1.5	1.4	-0.1	✓	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.8	-0.9	-0.1	✓	✗
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.5	1.6	0.1	✓	✗
27-05-2021 07_55_54_KM_0021331037_D12	29.3	29.3	0	✓	✓
30-04-2021_2_UB_1011026280_D5	-9.9	-9.7	0.2	✓	✗
30-04-2021_2_UB_1011026280_F3	27.5	27.4	-0.1	✓	✓
			Accuracy	**0.975**	**0.85**

Tab. A.5: Table of results after rotated coordinates correction.

Correction					
Name	Angle [°]	True angle [°]	Δ [°]	Angle result	Detection result
01-05-2021 09_15_23_KM_1011021661_A1	0.1	0	-0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.6	-2.5	0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38	-37.8	0.2	✓	✗
01-05-2021 09_15_23_KM_1011021661_C6	-14	-14	0	✓	✓
01-05-2021 09_15_23_KM_1011021661_E8	-12.8	-12.7	0.1	✓	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	✓
02-05-2021_UB_1011019967_A11	1	1	0	✓	✓
02-05-2021_UB_1011019967_B1	34.7	34.5	-0.2	✓	✓
04-05-2021 11_29_44_KM_1011021661_A3	-1.5	-1.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_A5	-4.4	-4.4	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.5	-8.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_C8	-8	-8	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_D6	-40.1	-40.1	0	✓	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-32.8	-32.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.7	1.7	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.1	0.2	0.1	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-0.8	-0.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1	-1	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-42.3	-42.3	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	0	3.1	3.1	✗	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.9	1.9	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.1	-0.1	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3.1	3	-0.1	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.5	0.6	0.1	✓	✗
18-05-2021 07_55_29_KM_1011021661_A8	1.5	1.4	-0.1	✓	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.8	-0.9	-0.1	✓	✗
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.5	1.6	0.1	✓	✗
27-05-2021 07_55_54_KM_0021331037_D12	29.3	29.3	0	✓	✓
30-04-2021_2_UB_1011026280_D5	-9.9	-9.7	0.2	✓	✗
30-04-2021_2_UB_1011026280_F3	27.5	27.4	-0.1	✓	✓
			Accuracy	0.975	0.875

Tab. A.6: Table of results after Kernel mask application.

Kernel					
Name	**Angle [°]**	**True angle [°]**	**Δ [°]**	**Angle result**	**Detection result**
01-05-2021 09_15_23_KM_1011021661_A1	0.1	0	-0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.6	-2.5	0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38	-37.8	0.2	✓	✓
01-05-2021 09_15_23_KM_1011021661_C6	-14	-14	0	✓	✓
01-05-2021 09_15_23_KM_1011021661_E8	-12.8	-12.7	0.1	✓	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	✓
02-05-2021_UB_1011019967_A11	1	1	0	✓	✓
02-05-2021_UB_1011019967_B1	34.7	34.5	-0.2	✓	✓
04-05-2021 11_29_44_KM_1011021661_A3	-1.5	-1.5	0	✓	✗
04-05-2021 11_29_44_KM_1011021661_A5	-4.4	-4.4	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.5	-8.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_C8	-8	-8	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_D6	-40.1	-40.1	0	✓	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-32.8	-32.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.7	1.7	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.1	0.2	0.1	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-0.8	-0.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1	-1	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-42.3	-42.3	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	0	3.1	3.1	✗	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	✗
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.9	1.9	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.1	-0.1	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3.1	3	-0.1	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.5	0.6	0.1	✓	✗
18-05-2021 07_55_29_KM_1011021661_A8	1.5	1.4	-0.1	✓	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.8	-0.9	-0.1	✓	✗
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.5	1.6	0.1	✓	✗
27-05-2021 07_55_54_KM_0021331037_D12	29.3	29.3	0	✓	✓
30-04-2021_2_UB_1011026280_D5	-9.9	-9.7	0.2	✓	✗
30-04-2021_2_UB_1011026280_F3	27.5	27.4	-0.1	✓	✗
			Accuracy	**0.975**	**0.825**

Tab. A.7: Table of results after cropping masks resizing.

Mask resizing					
Name	Angle [°]	True angle [°]	Δ [°]	Angle result	Detection result
01-05-2021 09_15_23_KM_1011021661_A1	0.1	0	-0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.6	-2.5	0.1	✓	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38	-37.8	0.2	✓	✓
01-05-2021 09_15_23_KM_1011021661_C6	-14	-14	0	✓	✓
01-05-2021 09_15_23_KM_1011021661_E8	-12.8	-12.7	0.1	✓	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0	✓	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0	✓	✓
02-05-2021_UB_1011019967_A11	1	1	0	✓	✓
02-05-2021_UB_1011019967_B1	34.7	34.5	-0.2	✓	✓
04-05-2021 11_29_44_KM_1011021661_A3	-1.5	-1.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_A5	-4.4	-4.4	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.5	-8.5	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_C8	-8	-8	0	✓	✓
04-05-2021 11_29_44_KM_1011021661_D6	-40.1	-40.1	0	✓	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-32.8	-32.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.7	1.7	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.1	0.2	0.1	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-0.8	-0.8	0	✓	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1	-1	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-42.3	-42.3	0	✓	✓
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0	✓	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0	✓	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0	✓	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0	✓	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	0	3.1	3.1	✗	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.9	1.9	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.1	-0.1	0	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3.1	3	-0.1	✓	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0	✓	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.5	0.6	0.1	✓	✓
18-05-2021 07_55_29_KM_1011021661_A8	1.5	1.4	-0.1	✓	✓
20-05-2021_Biotarget_UB_0021331024_E12	-0.8	-0.9	-0.1	✓	✗
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.5	1.6	0.1	✓	✓
27-05-2021 07_55_54_KM_0021331037_D12	29.3	29.3	0	✓	✓
30-04-2021_2_UB_1011026280_D5	-9.9	-9.7	0.2	✓	✗
30-04-2021_2_UB_1011026280_F3	27.5	27.4	-0.1	✓	✓
			Accuracy	**0.975**	**0.95**

Tab. A.8: Table of results with artificially introduced error - 1°.

Mask resizing + artificial introduction of 1° error					
Name	Angle [°]	True angle [°]	Δ [°]	Angle result	Detection result
01-05-2021 09_15_23_KM_1011021661_A1	0.1	0	-0.1 + 1	✗	✓
01-05-2021 09_15_23_KM_1011021661_A7	-2.6	-2.5	0.1 + 1	✗	✓
01-05-2021 09_15_23_KM_1011021661_C1	-38	-37.8	0.2 + 1	✗	✓
01-05-2021 09_15_23_KM_1011021661_C6	-14	-14	0 + 1	✗	✓
01-05-2021 09_15_23_KM_1011021661_E8	-12.8	-12.7	0.1 + 1	✗	✓
01-05-2021-2-UB_1011019967_B9	2.3	2.3	0 + 1	✗	✓
02-05-2021 09_06_08_KM_1011021661_A1	0.1	0.1	0 + 1	✗	✓
02-05-2021 09_06_08_KM_1011021661_D2	-16.7	-16.7	0 + 1	✗	✓
02-05-2021_UB_1011019967_A11	1	1	0 + 1	✗	✓
02-05-2021_UB_1011019967_B1	34.7	34.5	-0.2 + 1	✗	✓
04-05-2021 11_29_44_KM_1011021661_A3	-1.5	-1.5	0 + 1	✗	✓
04-05-2021 11_29_44_KM_1011021661_A5	-4.4	-4.4	0 + 1	✗	✓
04-05-2021 11_29_44_KM_1011021661_B5	-8.5	-8.5	0 + 1	✗	✓
04-05-2021 11_29_44_KM_1011021661_C8	-8	-8	0 + 1	✗	✓
04-05-2021 11_29_44_KM_1011021661_D6	-40.1	-40.1	0 + 1	✗	✓
05-03-2021 10_11_02_polished_cl_1011021701_A1	-32.8	-32.8	0 + 1	✗	✓
09-02-2021 09_49_16_bio_sb_0021329273_A3	1.7	1.7	0 + 1	✗	✓
09-02-2021 09_49_16_bio_sb_0021329273_B5	0.1	0.2	0.1 + 1	✗	✓
09-02-2021 09_49_16_bio_sb_0021329273_D4	-0.8	-0.8	0 + 1	✗	✓
09-02-2021 09_49_16_bio_sb_0021329273_D7	-0.2	-0.3	-0.1 + 1	✗	✓
09-02-2021 11_55_04 bio_cl_0021329271_B6	-1	-1	0 + 1	✗	✓
09-02-2021 11_55_04 bio_cl_0021329271_B9	-42.3	-42.3	0 + 1	✗	✓
09-02-2021 11_55_04 bio_cl_0021329271_D8	0.2	0.2	0 + 1	✗	✓
10-02-2021 08_13_25 bio_cl_0021331197_G7	-1.8	-1.8	0 + 1	✗	✓
10-02-2021 08_13_25 bio_cl_0021331197_H9	-1.2	-1.2	0 + 1	✗	✓
10-08-2021_KM_1011026323_A10	-0.2	-0.2	0 + 1	✗	✓
10-08-2021_steeltarget_UB_1011019967_A5	3.2	3.2	0 + 1	✗	✓
11-02-2021 08_11_15_bio_sb_0021331184_B9	0	3.1	3.1 + 1	✗	✓
11-05-2021 08_16_03_KM_1011021661_D9	-3.6	-3.6	0 + 1	✗	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_C6	1.9	1.9	0 + 1	✗	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D1	-0.1	-0.1	0 + 1	✗	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D12	3.1	3	-0.1 + 1	✗	✓
15-07-2021_Biotarget_pathfinder_UB_0021328247_D7	0.2	0.2	0 + 1	✗	✓
17-02-2021 09_01_43_polished_cl_1011021740_C9	0.5	0.6	0.1 + 1	✗	✓
18-05-2021 07_55_29_KM_1011021661_A8	1.5	1.4	-0.1 + 1	✗	✓
25-02-2021 08_14_04_polished_cl_1011026119_C8	1.5	1.6	0.1 + 1	✗	✓
27-05-2021 07_55_54_KM_0021331037_D12	29.3	29.3	0 + 1	✗	✓
30-04-2021_2_UB_1011026280_F3	27.5	27.4	-0.1 + 1	✗	✓
			Accuracy	0	1

B List of attachments

Original_solution.py – The original solution of the dish detection.
New_model.py – The "New" model for dish detection with new parameters.
The_best_solution.py – The model for dish detection with the best results.
Rotator.py – The original solution for calculating dish rotation and obtaining colony position.
Experiments.py – The script with all experiments of dish rotator and colony position retriever.