

Linköping University | Department of Computer and Information Science
Bachelor's thesis, 16 ECTS Computer Science
2020 | LIU-IDA/LITH-EX-G--20/030--SE

Image Processing for Improved Bacteria Classification

Emil Bråkenhielm (emibr678)
Peder Leijonhufvud (pedle936)

Supervisor : Adrian Horga
Examiner : Martin Sjölund

External supervisor : Erika Andeskär, Martin Johansson

Upphovsrätt

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid den senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>.

Copyright

The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances.

The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: <http://www.ep.liu.se/>.

Abstract

Mastitis is a common disease among cows in dairy farms. Diagnosis of the infection is today done manually, by analyzing bacteria growth on agar plates. However, classifiers are being developed for automated diagnostics using images of agar plates. Input images need to be of reasonable quality and consistent in terms of scale, positioning, perspective, and rotation for accurate classification. Therefore, this thesis investigates if a combination of image processing techniques can be used to match each input image to a pre-defined reference model. A method was proposed to identify important key points needed to register the input image to the reference model. The key points were defined by identifying the agar plate, its compartments, and its rotation within the image.

The results showed that image registration with the correct key points was sufficient enough to match images of agar plates to a reference model despite any varieties in scale, position, perspective, or rotation. However, the accuracy depended on the identification of the salient features of the agar plate. Ultimately, the work proposes an approach using image registration to transform images of agar plates based on a pre-defined reference model, rather than a reference image.

Acknowledgements

Firstly, we would like to thank our external supervisor Adrian Horga for the constructive comments, help, and support throughout this thesis. His insightful comments have been greatly appreciated. Secondly, we would like to thank our examiner Martin Sjölund for his final feedback.

We would also like to express our gratitude towards Agricam AB for the opportunity to do this thesis. A special thanks for the input and support from our supervisors at the company, Erika Andeskär, and Martin Johansson.

Contents

Abstract	iii
Acknowledgments	v
Contents	vii
List of Figures	ix
1 Introduction	1
1.1 Motivation	1
1.2 Aim of the work	2
1.3 Problem description	2
1.4 Limitations	2
2 Theory	3
2.1 Agar Plate	3
2.2 Image Processing	4
2.2.1 Color correction	4
2.2.2 Edge detection	4
2.2.3 Masking	10
2.2.4 Color Space Segmentation	10
2.2.5 Image registration	11
2.2.6 Image accuracy validation	14
3 Related Work	15
3.1 Edge detection	15
3.2 Shape detection	15
3.3 Image Registration	16
4 Method	17
4.1 Implementation	17
4.1.1 Identify and mask the agar plate	18
4.1.2 Identify compartment edges	19
4.1.3 Identify agar plate orientation	19
4.1.4 Image registration	20
4.2 Evaluation	21
5 Results	23
5.1 Implementation	23
5.1.1 Identify and mask the agar plate	23
5.1.2 Identify compartment edges	27
5.1.3 Identify agar plate orientation	28
5.1.4 Image registration	29

5.2 Evaluation	29
6 Discussion	31
6.1 Results	31
6.2 Method	32
6.3 The work in a wider context	33
7 Conclusion	35
7.1 Future work	35
Bibliography	37

List of Figures

2.1	a) Empty Petri dish with four compartments. b) Petri dish with agar, i.e., an agar plate.	3
2.2	a) Before applying Gaussian Blur. b) After applying Gaussian Blur	5
2.3	The Sobel y-kernel checks for the gradient on the y-axis	5
2.4	Non-maximum suppression illustration. Values in red are dismissed, and green kept.	6
2.5	a) Before Hysteresis, b) after Hysteresis.	6
2.6	Dilation adds pixels to the object contour, while erosion removes pixels.	7
2.7	The right image shows the output of a skeletonized binary image.	7
2.8	a) Illustrates S_2 as a 1-component. b) Surroundness among components. c) Surroundness among borders (c)	8
2.9	a) A data set without outliers and inliers distinguished ¹ . b) The RANSAC algorithm applied to fit a line. Outliers in red are dismissed in the result ²	9
2.10	a) Rho and theta parameters of a straight line [Duda]. b) Three different colored curves obtained by <i>Equation 2.2</i> . The intersecting point indicates a line with the parameters p and θ .	10
2.11	Figure illustrates an image with and without masking of irrelevant data using points given by Circle Hough Transform	11
2.12	a) Example of an input image with a distinctive red-colored compartment. b) An HSV-space 3D scatter plot of the input. c) An RGB-space 3D scatter plot of the input.	11
2.13	a) Will be adjusted to mimic the reference image in b) as close as possible, based on the reference points	12
4.1	Workflow of the implementation structure	17
4.2	Flowchart to identify and mask the agar plate.	18
4.3	Workflow to identify the compartment edges and the center point of the agar plate.	19
4.4	a) illustrates the new ROI boundary. b) show the edges and intersection (i.e., center point) of the compartment edges.	20
4.5	Flow chart to identify the orientation of the agar plate.	20
4.6	Illustration of a structure based on pre-defined reference key points.	21
4.7	a) Flat ellipse. b) Foreshortened circle (spatial illusion).	21
4.8	Illustrates an output with a reference template overlay.	22
5.1	a) Input image. b) The outer contour of the agar plate identified. c) Compartment edges identified. d) Final output.	24
5.2	a) Input image of an agar plate under dark conditions. b) Image processed with SCB. c) Image processed with SCB with prior brightness and contrast balance	25
5.3	a) Input image. b) image processed with Canny Edge Detection. c) Image processed with Canny Edge Detection but, with extra Gaussian Blur.	25
5.4	a) Closing applied to fix broken canny edges. b) Border following of the outermost contour. c) RANSAC to fit an ellipse on the identified outer edge. d) The final contour of the agar plate identified.	26

5.5	a) Agar plate with detected outer contour. b) Pixels outside the outer contour masked	26
5.6	a) Masked binary Canny image. b) Morphological Closing to form one thick line. c) Skeletonize of the thick line, giving a pixel-wide centerline.	27
5.7	a) Lines found by Hough Lines over skeletonized output. b) The same lines are drawn over the input image.	28
5.8	a) Intersection points between the outer contour and the identified compartment edges. b) Segmented red compartment using color space segmentation. c) Sorted intersection points.	28
5.9	a) Identified key points for the input image. b) Reference key points.	29
5.10	a) Image constructed with extreme perspective distortion before registration. b) The result after one iteration. c) The result after two iterations.	29
5.11	Validation of final output from <i>Figure 5.1 d)</i>	30



1 Introduction

Mastitis is the most common disease among cows in dairy farms in Sweden¹. Aside from great suffering for the animals, mastitis is very costly for the agriculture and contributes to the overuse of antibiotics. As an example, the cost per mastitis case in Sweden approximates to be over 3300 SEK due to treatment cost and reduced milk production. Furthermore, there are approximately 380 000 milk cows in Sweden, and 75 000 of these cows are treated annually with antibiotics due to mastitis. Mastitis is caused by, in most cases, a bacterial infection, and it is therefore important to get a diagnosis as soon as possible to know what kind of bacteria it is.

Today, most cases of mastitis are diagnosed by licensed veterinarians or specially trained laboratory technicians² who analyze bacteria growth on agar plates. This is both time-consuming and expensive for the farmer.

Bacteria classifiers are currently in development to automate the whole process by analyzing images of agar plates. The classification today is dependent on the conditions and quality of the image. A too large difference between the images may result in faulty diagnostics. The input images, therefore, need to be properly pre-processed using image processing to address any variety in scale, positioning, perspective, illumination, and noisy backgrounds.

1.1 Motivation

Agricam is an IT-company located in Linköping who develops products and services adapted for dairy farmers. The company aims to streamline and digitalize animal health work. They are currently developing a bacteria classifier to automatically identify mastitis bacteria in milk samples from Swedish dairy farms.

When growing bacteria, a milk sample is applied to four different compartments of an agar plate. Each compartment contains a different growth medium to maintain a higher diagnos-

¹<http://intervacc.se/forskning-utveckling/projekt/novel-antigen/om-s-aureus/>

²<http://www.juverportalen.se/media/1176/mastitdiagnostik-naer-var-hur.pdf>

tic accuracy.

The rotation of the agar plate is thus of great importance to not confuse any sample with the wrong compartment. Each growth medium has different colors, which helps to distinguish the compartments. Which color corresponds to what growth medium, however, are not identified by the classifier.

Therefore, each sample image needs to be pre-processed to ensure correct rotation. Variety in scale, position, perspective, and illumination may also affect the diagnosis. So by defining a model in terms of previously mentioned factors, each image can be processed to fit the model, consequentially leading to correct rotation and higher diagnostic accuracy.

1.2 Aim of the work

This thesis aims to investigate if a combination of image processing techniques can pre-process images of agar plates to be positioned correctly despite variations in scale, positioning, lighting, perspective and noisy backgrounds.

1.3 Problem description

1. Investigate if a combination of image processing techniques can be used for scale, rotation, and perspective invariant image registration of agar plates, based on the criteria below.

Criteria:

1. The outer edge of the agar plate, as well as the compartment edges shall be identified.
2. Pixels outside the agar plate should be masked to remove background noise.
3. Depending on the angle, rotation, position, and scale, the image should be adjusted to match a reference picture.

1.4 Limitations

The following limitations have been taken during the work:

1. Images used should be with high enough resolution and quality to distinguish each bacteria cluster. Correction of image deviations may also be limited to the extent that the image quality needs to be good enough for the classifier. Too extreme variations may, therefore, be dismissed.

2 Theory

The following chapter describes the necessary information needed to solve the problem description and to help design an appropriate method. The theoretical part will first explain the concept of an agar plate, followed by various existing approaches to image processing and edge identification.

2.1 Agar Plate

Agar plates are Petri dishes containing agar, which is a gelatinous substance derived from red algae used as a bacteria growth medium. Agar plates are often used in laboratory settings to analyze if certain bacteria exist within a sample. Depending on the situation, the agar plate could be divided into multiple compartments, as shown in *Figure 2.1*. Each compartment can be used to test different samples or the same sample with different bacteria.



Figure 2.1: a) Empty Petri dish with four compartments. b) Petri dish with agar, i.e., an agar plate.

2.2 Image Processing

Digital image processing is about manipulating pixels. For example, improving, analyzing, or in some way, extracting information from an image. This section will focus on this type of algorithms and techniques suitable for the aim of the work.

2.2.1 Color correction

Altering the color of an image is an essential task in image processing and often means removing dominant or undesirable colors in a picture. Objects may have different colors on the images compared to reality.

[1] proposes a color balance algorithm named Simplest Color Balance (SCB). The algorithm works by stretching red, green, and blue values within a [0, 255] range. Pixels with high values are assumed to contain high levels of the color in question and vice versa.

The same idea can be applied to balance brightness, where grayscale-values instead of RGB-values are used. An image with low brightness, for example, may not have any pixels close to 255. By stretching the scale [0, 255], the brightness can be either increased or decreased.

2.2.2 Edge detection

Edges (i.e., an abrupt change of gray or color intensity) need to be distinguished to extract specific data of objects in images. By minimizing the number of false edges, actual edges could easier be identified. If the goal is to find the most intense edge, as the edge of an agar plate, blur filters and morphological operations could be used as an extension. The Hough Transform algorithm (see *Section 2.2.2.6*) is designed to find simple shapes within an image and to extract the coordinates or the area of an object.

2.2.2.1 Canny Edge Detection

The Canny operator [2] is based on a multi-stage algorithm initially developed to detect edges in images. Canny is divided into five different stages explained below.

Gaussian Blur

The first stage of Canny is to apply a Gaussian blur filter to reduce any image noise occurring, as shown in *Figure 2.2*. Otherwise, pixel noise could be falsely detected as an edge. The algorithm uses a kernel, which is a convolution matrix of $n * n$ pixels, to search all pixels in an image. Each pixel is evaluated, and its pixel value is replaced with a weighted mean of its neighboring pixels. A two-dimensional Gaussian function is generally used to remove image noise (see *Equation 2.1*).

$$G(x, y) = \exp[-(x^2 + y^2 / 2\sigma^2) / 2\pi\sigma^2] \quad (2.1)$$

Where σ (sigma) is the parameter that controls the smoothing extent of an image.

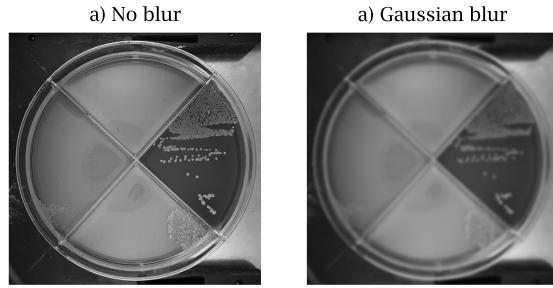


Figure 2.2: a) Before applying Gaussian Blur. b) After applying Gaussian Blur

Sobel Operator

After the applied blur, a Sobel kernel is used to identify all intensity gradients. With its kernel, Sobel looks for areas in the image that have a high gradient, i.e., a distinct and sudden change in color (in RGB) or brightness (in grayscale). *Figure 2.3* shows how the method works in the x-axis as well as the y-axis. Each value within the kernel is multiplied by the value at the corresponding location in the image. The sum of the matrix gives a gradient value. A higher value means a more defined edge. The derivative of the respective edges is then calculated to determine the angle of the edge.

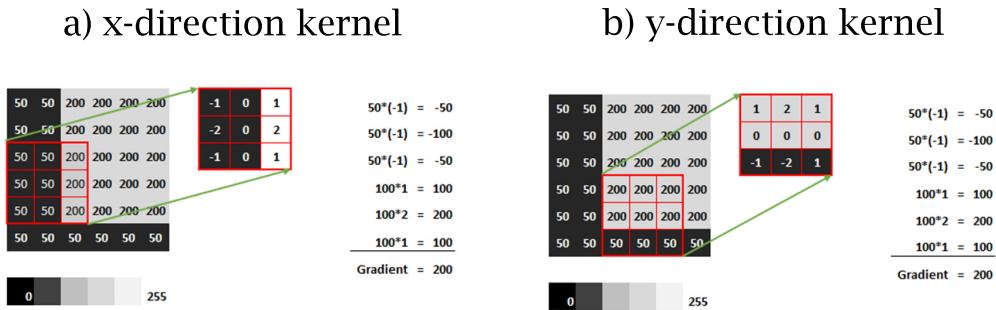


Figure 2.3: The Sobel y-kernel checks for the gradient on the y-axis

Non-maximum suppression

While the Sobel filter finds most edges, usually, only the most prominent are interesting. The output may have both thick and thin edges, and a non-maximum suppression can be used to sort out any false edges.

By using the gradient intensity matrix determined in the previous step, the algorithm seeks through the image, following the direction of the edge. For each pixel, the algorithm checks neighboring pixels to the left and right of the edge. If a neighboring pixel in the same direction is more intense than the current pixel, the most intense pixel is kept. *Figure 2.4* shows an example where the pixels with intensity values 170 and 115 would be dismissed and 255 kept.

Double threshold

The non-maximum suppression will keep the most intense pixels identified in each edge, but there can still be intensity variations between these pixels. A lower and upper threshold is applied to strengthen the edge, which checks for pixels that are strong, weak, or

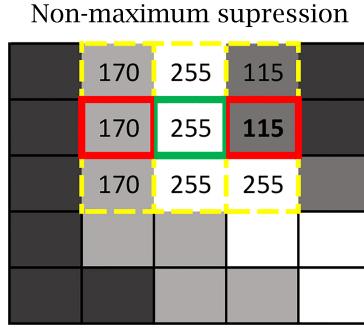


Figure 2.4: Non-maximum suppression illustration. Values in red are dismissed, and green kept.

non-relevant. The lower threshold is used to identify non-relevant pixels, while the upper threshold searches for strong pixels. Any pixels between the lower and upper threshold are marked as weak and will later be processed in the Hysteresis mechanism.

Hysteresis

The last stage of the Canny process is to decide which edges are true or not. With user-defined min and max parameters of a new threshold, previously detected weak pixels are transformed to strong pixels only if their pixel value is within the defined threshold range. Figure 2.5 b) illustrates this procedure by discarding non-relevant pixels to value 0. The identified pixel also needs to be connected to a known edge, or else it will be discarded.

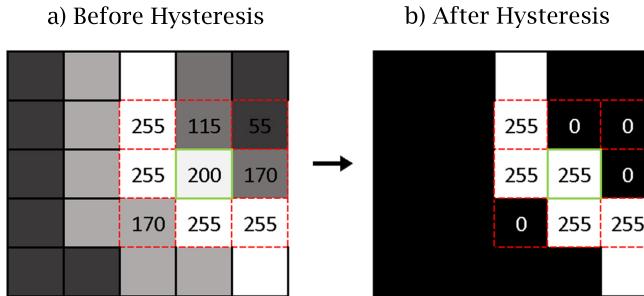


Figure 2.5: a) Before Hysteresis, b) after Hysteresis.

2.2.2.2 Morphological Operations

Morphological operations are used to process images based on geometrical structures. Different operations can be used to fix broken edges or areas within or outside the given boundaries of an object. The most common morphological operations are *Dilation* and *Erosion* [3], which either adds or removes pixels from an object contour. Dilation, followed by erosion, is called *Closing*, which can be used to repair disconnected edges. Its counterpart *Opening* are instead erosion followed by dilation can instead be used to reduce image noise

A structuring element (a matrix of user-defined size), searches the image checking each pixel with its corresponding neighbor (see Figure 2.6). The value of each pixel within the structur-

ing element is then decided based on different rules. For the dilation, each pixel is set to the most intense value found within the structuring element, while erosion sets each pixel values to the least intense.

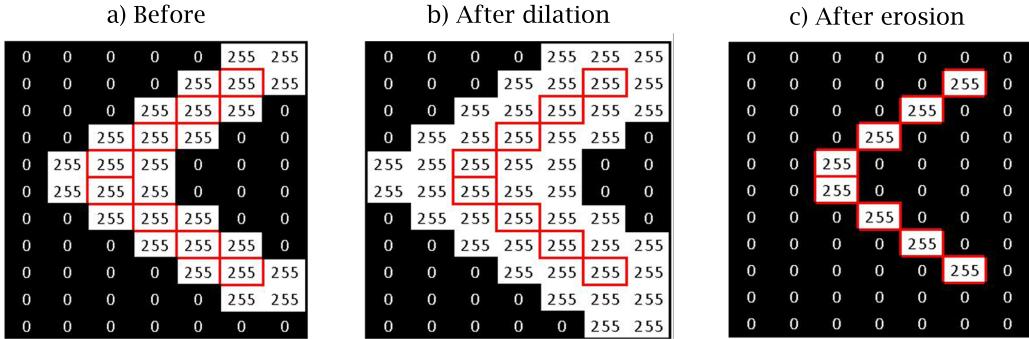


Figure 2.6: Dilation adds pixels to the object contour, while erosion removes pixels.

2.2.2.3 Skeletonize

Skeletonization [4] is a technique that modifies a binary image using Dilation and Erosion to reduce foreground regions into a skeletal remnant, i.e., the innermost contour of a binary object (see *Figure 2.7*). In an ideal case, the result should give a 1-pixel-wide representation. By skeletonizing an image, one can achieve improved computation speed in image processing with only a light skeleton to process, as well as increased accuracy of finding the actual position of a line or edge.

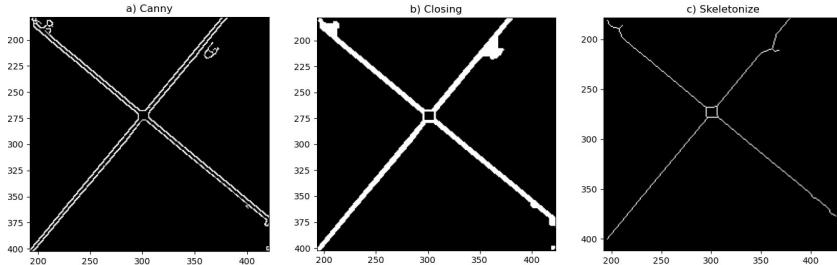


Figure 2.7: The right image shows the output of a skeletonized binary image.

Opening is used to reveal the skeleton of an object. The process is repeated as long as the connectivity of the binary object is not broken. When broken, the last successful iteration is saved as output.

2.2.2.4 Border following

Extracting the contour of an object can be useful to analyze shapes and detect or recognize different objects. Border following [5] is a type of algorithm that makes it possible to extract contours of given objects. Contours are generally defined by a closed curve joining the continuous points of an object boundary. For a contour to be identified, the object boundary should consist of pixels with the same color or intensity. For binary images, contours are divided into 1- and 0-components, which represent areas of their respectively binary image value. In *Figure 2.8*, S_1 , which is the background 0-component, is interpreted as a hole between the

frame border and the B_1 border. The algorithm checks images for 1-component and defines its borders as the transition between any 1-components neighboring 0-component.

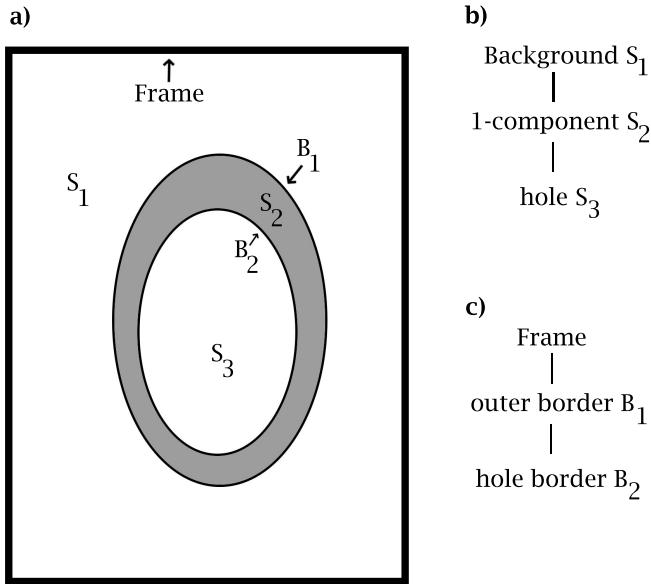


Figure 2.8: a) Illustrates S_2 as a 1-component. b) Surroundness among components. c) Surroundness among borders (c)

Each contour found can be sorted into a different hierarchy based on length or occurrence moving outwards or inwards. A child- and parent contour of a 1-component would occur in the same hierarchy. If a hierarchy level only consists of single contours, i.e., contours without a corresponding child or parent contour, that hierarchy level would return a negative value. The external contour object of a single-object image could then be defined as being at the top of the hierarchy.

2.2.2.5 Random Sample Consensus

Random Sample Consensus (RANSAC) [6] is an algorithm used to estimate parameters of a given model. It is often used in Computer Vision to solve correspondence problems when defining what part of an image corresponds to in another image, i.e., feature matching. The main idea is to cope with outliers occurring in input data, which is done by random sampling of observed data. The RANSAC algorithm can be configured to match specific shape models such as circles or ellipses, which can be useful to give a more accurate approximation of, e.g., a found contour.

Figure 2.9 illustrates a simple least square method and a RANSAC approach for fitting a two-dimensional line. Both sets of observation contain inliers, i.e., points which can be approximated fitted to a line, and outliers which cannot be fitted. However, the least square method does not distinguish between outlier or inliers, and would, in this case, give a poor result.

By using RANSAC, a random subset, entirely consisting of inliers, would define the fitted line. Since the algorithm is based on the probability that a particular group of inliers should represent the fitted line, the accuracy of RANSAC is typically improved by more iterations.

¹https://commons.wikimedia.org/wiki/File:Line_with_outliers.svg

²https://commons.wikimedia.org/wiki/File:Fitted_line.svg.

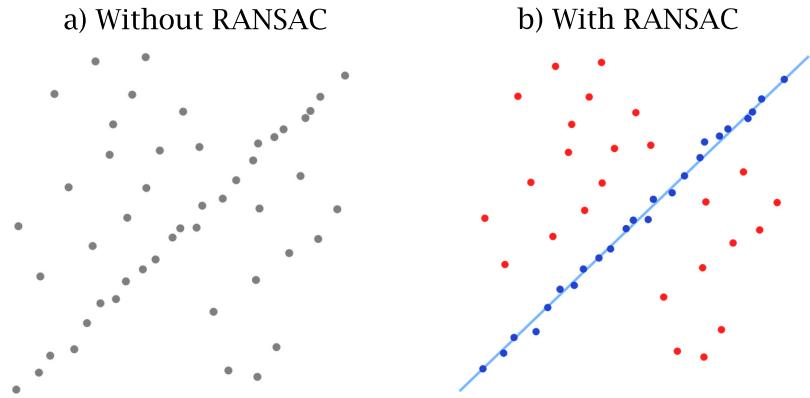


Figure 2.9: a) A data set without outliers and inliers distinguished¹. b) The RANSAC algorithm applied to fit a line. Outliers in red are dismissed in the result².

Except when overfitting the model, which can cause the algorithm to find a local minimum, giving poor results.

2.2.2.6 Hough Transform

The Hough Transform [7] is a technique used to isolate simple shapes within an image. Initially, Hough Transform was created with the intent to find lines but has later been extended to identify positions of arbitrary shapes, most commonly circles and ellipses. By using a voting procedure, the purpose of Hough Transform is to identify imperfect instances of objects inside a certain class. A simple shape is defined as a structure that can be represented by only a few parameters. For example, a line can be defined with two parameters: slope and intercept. A circle can be defined with three parameters: the coordinates of the center (x, y) and the radius (r).

The standard Hough Transform is used for detecting lines and achieves this by defining a line in the Hesse normal form:

$$p = x\cos(\theta) + y\sin(\theta) \quad (2.2)$$

where p is the perpendicular distance, in pixels, from the origin to the line. θ represents the angle measured in radians, which is the orientation of p as shown in *Figure 2.10 a)*.

Each line can now be represented in the (p, θ) form, which implies that any value (p, θ) corresponds to a line in a two-dimensional parameter space. Therefore, every (x, y) value in an image can now be represented as a curve, according to *Equation 2.2*. The (p, θ) -plane could be imagined as the hough-space where every (x, y) value in an image are accumulated into this space.

The result consists of several sinusoidal curves that intersect where lines occur, as shown in *Figure 2.10 b)*. A voting procedure is then used to tell where lines are in an image. For every point on a line, the corresponding accumulator cell is increased. Finding the cells with the most peaks in the accumulator array tells where the lines are located in the corresponding image. To better find the peaks in the accumulator array, a threshold could be applied to Hough Transform. However, different operations might yield different results depending on the image.

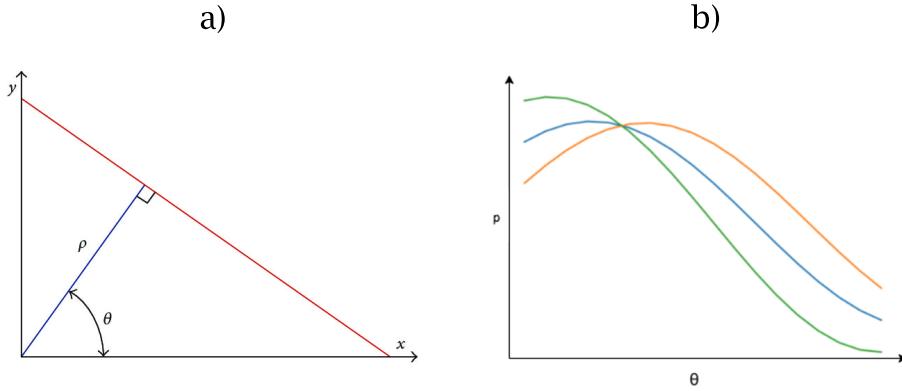


Figure 2.10: a) Rho and theta parameters of a straight line [7]. b) Three different colored curves obtained by *Equation 2.2*. The intersecting point indicates a line with the parameters p and θ .

2.2.2.7 Modified Hough Transform

Using modified versions of Hough Transform enables the technique to find shapes like circles and ellipses. To find circles in imperfect images, one can use Circle Hough Transform (CHT). The technique is similar to the standard Hough Transform, with the difference being that the parameter space is different. The same procedure can then be done in CHT to detect other features with other analytical descriptions. In general, a circle can be defined by:

$$(x - a)^2 + (y - b)^2 = r \quad (2.3)$$

where (r) is the radius and (a, b) is the center of a circle.

For every fixed (x, y) value in an image, every parameter in *Equation 2.3* can be found. The hough-space is then defined as the (a, b, r) -plane. However, because the (a, b, r) -plane defines a three-dimensional space, the parameters can be identified in two stages. The first stage is to fix the radius to find the optimal center point in the two-dimensional parameter space. After that, the optimal radius can be derived in one-dimensional parameter space. The computational complexity of the algorithm increases if more parameters exist in the parameter space.

2.2.3 Masking

Region-of-interest (ROI) processing is an approach aiming to extract that information from a specific subregion of an image. The ROI is usually defined by points, lines, circles, or other shapes, which is then used to apply a binary mask to the image. The binary mask is created with the same size as the input image. Pixels within the ROI is set to 1 and the remaining pixels to 0. *Figure 2.11* illustrates before and after masking, where the ROI is the agar plate.

2.2.4 Color Space Segmentation

In circular shaped objects, it can sometimes be challenging to determine the rotation. With the human eye, one can recognize a specific shape, color, or feature within an object that only occurs in a particular direction, thus determine the rotation. This rotational landmark can be extracted using *Image Segmentation* [8], which simply divides groups of pixels together based on given criteria.

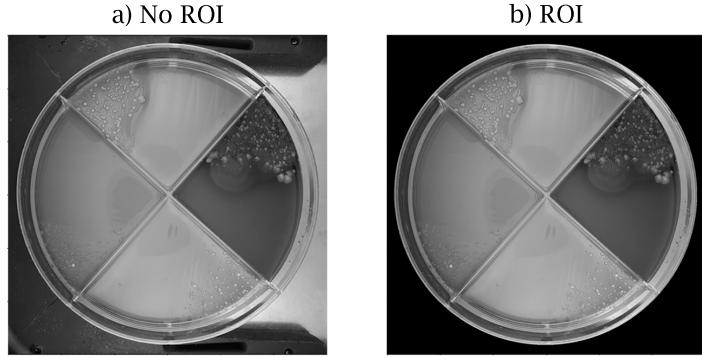


Figure 2.11: Figure illustrates an image with and without masking of irrelevant data using points given by Circle Hough Transform

In *Figure 2.12 a)*, a compartment in distinctive red can be distinguished in the right half of the image. To extract the orientation data of this compartment, its pixels need to be segmented. To segment areas based on color, criteria can be defined by Color Spaces, which are colors represented of their different components, e.g., RGB (Red, Green, Blue) or HSV (Hue, Saturation, Value). For example, in *Figure 2.12 b)*, the red tones in the HSV-space are, in this case, generally more localized and visually separable. The RGB-space in *Figure 2.12 c)*, on the other hand, shows that the red tones have a larger span across both the green and blue axis. By adjusting color space values, different information can be extracted from the image. For example, changing the Red in RGB to 0 would remove all red tones in a picture, leaving only a spectrum of green and blue.

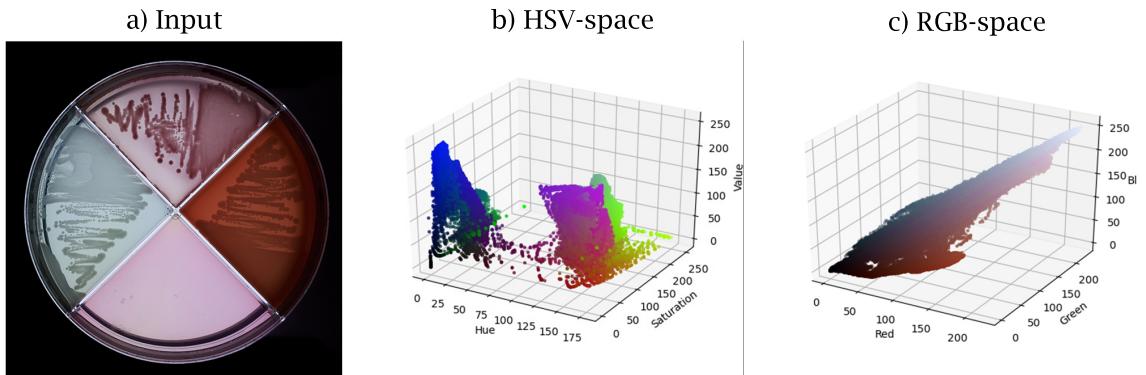


Figure 2.12: a) Example of an input image with a distinctive red-colored compartment. b) An HSV-space 3D scatter plot of the input. c) An RGB-space 3D scatter plot of the input.

2.2.5 Image registration

Misalignment in images is a common issue and can happen for a variety of reasons. The most common reason is due to images being captured under variable conditions, such as changes in camera perspective or the scene content. Image registration [9] is a technique used in medical sciences, remote sensing, and computer vision that corrects misaligned images. The fundamental idea is to align two or more images of the same scene with one image designated as the reference image. Thus, getting an image matching the reference image (see *Figure 2.13*). There exist multiple image registration techniques, but the majority of them can

be broken down into the following steps described below:

1. *Feature detection*: The detection process of different features in an image. This process can be manual or automatic, but an automatic detection is preferable. For example, this process looks for edges, contours, closed-boundary regions, line intersections, and corners. Different image registration techniques can be divided into intensity-based or feature-based. Briefly, intensity-based methods look for distinctive edges in an image whereas, feature-based looks for specific objects which are more prevalent in computer vision.
2. *Feature matching*: The process of matching the different features detected in the reference image and those detected in the non-aligned pictures (sensed images).
3. *Transform model assessment*: The parameters of the mapping functions are estimated by aligning the sensed image with the reference image.
4. *Image transformation*: The sensed image is transformed by employing the mapping functions.

The tasks described above have typical problems, and problems might occur if specific requirements are not obtained [10]. The detection of features should be distinctive objects or objects spread over the image, which are easily detectable. The feature matching between the sensed image and the reference image needs to have enough common features to achieve proper registration. Especially, on occasions when the images do not cover the same scene, object collision occurs or other unexpected changes.

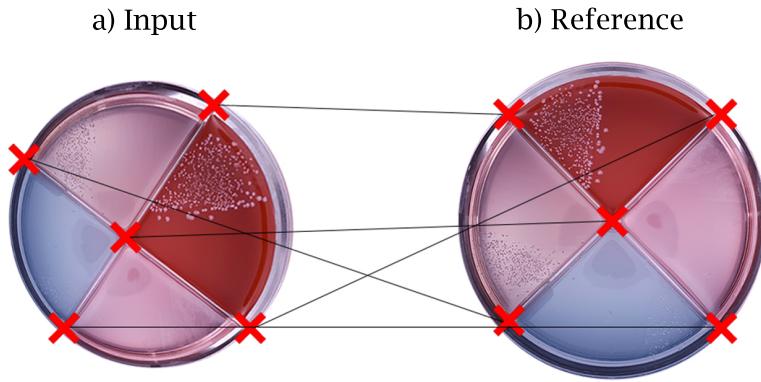


Figure 2.13: a) Will be adjusted to mimic the reference image in b) as close as possible, based on the reference points

2.2.5.1 Different transformation models

Different image registration techniques [9] can also be classified depending on what transformation models they use. The most commonly used technique is linear transformations, followed by projective transformations. Non-linear transformations exists as well, but it is beyond the scope of this work. Linear transformations include transformation on rotation, translation, scaling, and shearing in images, and are collectively known as affine transforms. The transformation in a two-dimensional space is done by determining the transformation matrix shown below:

1. *Rotation:* object can be rotated given an angle θ proportionate to its origin.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (2.4)$$

(x_1, y_1) is the new point, (x_2, y_2) is the old point, and θ is the rotation parameter.

2. *Scaling:* Scaling is done to resize an image or to match images with different sizes.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} + \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \quad (2.5)$$

Where (x_1, y_1) is the new point, (x_2, y_2) is the old point, and (s_x, s_y) is the scaling parameters. For example, if $s_x = 2$ and $s_y = 2$ the old point (x_2, y_2) is scaled by two in every direction.

3. *Translation:* If a point x, y is to be translated by t units then, the transformation matrix is:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \quad (2.6)$$

where (x_1, y_1) is the new point, (x_2, y_2) is the old point, and (t_1, t_2) is the translation value.

4. *Shearing:* Shearing slants the shape of an object. There exist two sheering transformations, x-shear, which changes the values of the coordinates and y-shear, which changes the values of the y coordinates.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \quad (2.7)$$

where (x_1, y_1) is the new point, (x_2, y_2) is the old point, and $(a_{11}, a_{12}, a_{21}, a_{22}, a_{13}, a_{23})$ is the sheering parameters.

5. *Reflection:* Reflection is done by mirroring an image of the original object. Simply, reflection does not change the size of the object.

The difference between linear transformations and projective transformations is that linear transformations are global by nature. Therefore, the geometrical difference between images cannot be modeled. However, projective transformations allow for warping a target image to match a reference image, which is achieved by a homography. A homography is a transformation matrix that maps points from the target image to the corresponding point in the reference image as follows:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \quad (2.8)$$

Where H is the homography matrix.

2.2.6 Image accuracy validation

When working with large datasets, manual validation of each output may be too time-consuming and a waste of resources. By automating the validation process for each output, higher accuracy can be achieved as well as saving time otherwise spent on debugging.

For shape oriented image processing, different types of *Image moment* [11],[12] algorithms can be used. Image moments are values of the weighted average of pixel intensities in images, which can then be used to compare two images. The moment of a single-channel binary image, for example, is defined as *Equation 2.9*, with $I(x, y)$ being the intensity of a pixel on a given x, y coordinate.

$$M = \sum_x \sum_y I(x, y) \quad (2.9)$$

The formula can be extended to consider both pixel intensity, as well as its position in the image using *Equation 2.10*. However, this method is not entirely transformational invariant.

$$M_{ij} = \sum_x \sum_y x_i y_j I(x, y) \quad (2.10)$$

As an extension to image moments, *Central moments* can be used to address translation and scale-varieties. The central moment is defined using *Equation 2.11*. Where \bar{x} and \bar{y} are the weighted averages of pixels constituting different shapes (or blobs) found in an image.

$$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j I(x, y) \quad (2.11)$$

For the moments to be scale-invariant, the *Normalized Central Moments* are calculated as follows in *Equation 2.12*.

$$\eta_{ij} = \frac{\mu_{ij}}{(i + j)/2 + 1} \quad (2.12)$$

Even though the normalized central moments are translation and scale-invariant, rotation is still something to take into consideration. An algorithm called Hu Moments addresses this problem using a set of 7 values. Each value is calculated using Normalized Central Moments, where the first six moments address translation, scale, rotation, and reflection, and the last moment is skew invariant, which can identify mirror images.



3 Related Work

This chapter addresses previous work in image processing that tries to solve similar problems this thesis does. There exist several methods on how to identify different shapes and varieties in angle, rotation, position, and scale in images. However, no study found solves a similar problem this thesis does. Therefore, this chapter focuses on related work using similar techniques written about in the theory chapter.

3.1 Edge detection

Many recent papers have focused on Canny Edge Detection and similar techniques to detect edges in images. [13] did a study of different edge detection methods for various image processing applications. The same edge detection algorithm can not be applied for all types of images since edge detection methods are problem-oriented. Because it is challenging to perform edge detection in noisy images, the authors compare different edge detection methods with their advantages and limitations. Canny edge detection produced excellent results, especially under noisy conditions.

Moreover, many recent papers have focused on improving Canny Edge Detection to detect edges in noisy images better. [14] proposes an improvement by using a modified median filter instead of Gaussian smoothing. The algorithm could successfully remove, with optimal threshold values on the canny operator, noise from an ultrasound image of a kidney. [15] suggests an adaptive threshold selection method for the Canny Edge Detection algorithm to preserve more useful edges and more robust noise.

3.2 Shape detection

In work for pupil identification, [16] suggests using a combination of the two techniques Canny Edge Detection and Hough Transform to detect pupils in images. The canny operator is used to identify the edge of a pupil, while Hough Transform finds the exact position. The proposed algorithm managed to accurately fit circles of different eye images under different lighting conditions. Additionally, [17] presents an algorithm that can detect and outline the outer edge of the pupils in human eyes using Hough Transform and Canny Edge detection.

The algorithm was tested on 100 human eyes and produced a 95% successful result.

RANSAC is a popular method of choice for model fitting to find ellipses in images. [18] suggests an algorithm using the RANSAC procedure for pupil detection. Using RANSAC provides high accuracy with low running time in normal and noisy conditions and for variable illumination. [19] proposes an ellipse detection method using RANSAC that achieves high accuracy and computation cost in detecting multiple ellipses in images. The algorithm works in two steps. Firstly, region segmentation and contour detection are applied. Secondly, with each contour segment found, a modified RANSAC is applied to five randomly selected pixels to form an accurate ellipse.

3.3 Image Registration

There are numerous image registration methods, which are frequently used in medical imaging, automatic target recognition, and computer vision. [20] provides knowledge of different image registration methods and their use in various application areas by discussing their advantages and disadvantages. The techniques are divided into two groups, Area-based and feature-based. Feature-based techniques find correspondence in salient features in images such as lines, points, and contours. Area-based methods, however, emphasize on feature matching without the detection of salient objects.

4 Method

This chapter will summarize and describe the methods and approaches used to answer the problems stated in *Section 1.3*. The first section will describe the workflow proposed to generate the final results. The second section describes how the results will be evaluated.

4.1 Implementation

The work will be implemented in steps described in *Figure 4.1*. Each step is dependent on the output of its predecessor. The first step is to identify and mask the agar plate. With the irrelevant data masked, the second step is to identify the compartment edges in the agar plate. With the agar plate and its compartment edges identified, the orientation will be calculated based on the output of the previous steps. Lastly, based on gathered key points, the images will be processed using image registration.

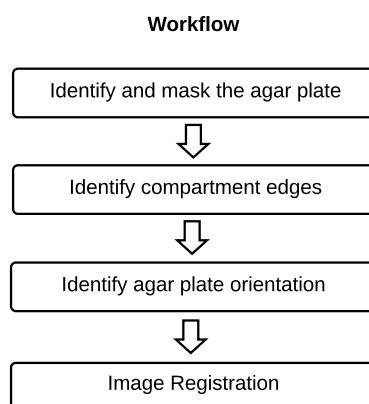


Figure 4.1: Workflow of the implementation structure

4.1.1 Identify and mask the agar plate

The first step is to identify the agar plate, which will provide an elliptical contour of points that defines the ROI. The workflow will be structured, as shown in *Figure 4.2*, and described throughout the section.

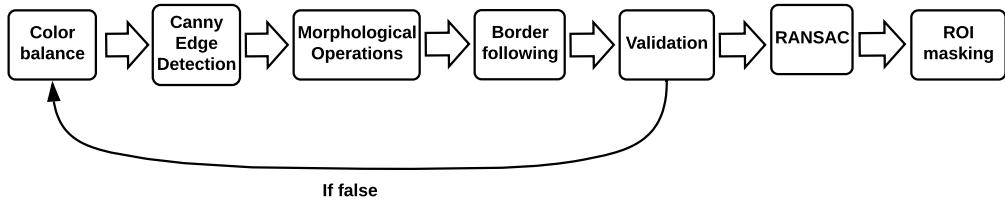


Figure 4.2: Flowchart to identify and mask the agar plate.

Different images might not have the same conditions regarding the illumination or intensity of RGB values, which means that each image might not get the same result when being processed. Therefore, it is essential to even out these variables. Images that are too dark or too bright may not take full advantage of color balancing; so they may also need to balance brightness and intensity beforehand. Each input will then be processed in terms of balancing brightness and contrast, followed by color balancing using SCB before further processing. The goal of this step is to improve the accuracy of the final output.

With contrast, brightness, and color all balanced, the next step is to process the image using Canny to intensify edges and reducing any image noise. Before applying Canny, an additional Gaussian blur will firstly be applied to reduce any bacteria within each compartment. Configuring Canny to the desired level with a minimal number of bacteria clusters, however, might cause disconnectivity in the outer contour. Morphological operations such as Closing will then be used to repair any broken lines.

The next step is to define any contours. CHT could be used to identify a perfectly circular contour, but due to variations in perspective, the agar plate will always be a bit elliptical. Using CHT, in this case, would give inaccurate results. Therefore, the contour will be found using Border Following. The contour found in the highest hierarchy should represent the external contour of the agar plate.

In some cases, segments of the contour may be imperfect due to, e.g., bacteria clusters growing close to the edge. To address this problem, each image will be validated before proceeding to the next step of this section. The contour defined by Border Following will be compared to an approximated contour found using CHT. The comparison will be made using Hu Moments. Since the Border Following contour never will be completely circular, as opposed to the contour from CHT, a margin of error will be considered. If the Hu Moment returns a value exceeding the margin of error, the whole workflow process will be repeated. Invalid matches may be the result of bacteria clusters covering segments of the agar plate edges in shadowed areas. The problem can be solved by increasing or decreasing the initial brightness or contrast. The image will be reprocessed until a valid match is found.

The validated contour will then be fitted to an elliptical model using RANSAC, which should give a better approximation of the external contour, providing the desired set of key-points.

Lastly, with the ROI defined by the contour key-points from previous steps, a binary mask will be applied to the image, removing the unwanted background.

4.1.2 Identify compartment edges

The second process is to identify the compartment edges and the center point on the agar plate. Each compartment divider can be segmented as two straight lines i.e., compartment edges. Therefore, by using Hough Transform, these shapes can be identified. Some pre-processing may be needed to identify the edges correctly and reduce irrelevant noise that might occur. *Figure 4.3* illustrates how the workflow is structured.

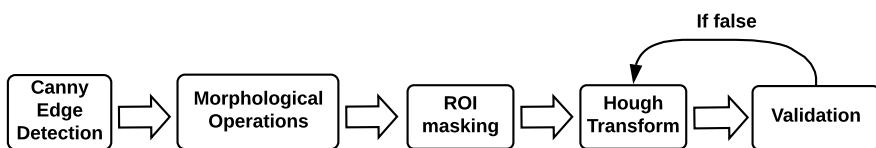


Figure 4.3: Workflow to identify the compartment edges and the center point of the agar plate.

Once again, Canny, with an extra Gaussian blur, is applied to identify only the most distinct edges. The difference in this case, compared to finding the outer contour, is that only the centermost line of the compartment edges is of interest. As seen in *Figure 4.4 b)*, each compartment divider consists of two edges, one on each side. Finding the centermost line is important since the intersection between the two crossing compartment edges constitutes the actual center point of the agar plate. The edges of each side of each compartment division can be merged to form one thick line using Morphological Closing. By using Skeletonize, the line can then be reduced to represent a pixel-wide centerline of each divider.

The mask boundaries from the previous section will be reused. The mask will be reduced in size, leaving a new ROI. As shown in *Figure 4.4 a)*, the area outside the drawn boundary is not needed to identify the positioning and angle of the compartment edges.

With a binary image representing the skeleton of the compartment edges, Hough Transform is used to identify the skeleton as lines. Lines will be sorted and validated in pairs based on values of rho and theta to make sure that only one line in each direction of the dividers is found.

4.1.3 Identify agar plate orientation

This part will focus on how to detect the rotation of the agar plate using segmentation. As shown in *Figure 4.5*, color space segmentation will be used to define the agar plate orientation, and key-points extracted from the previous sections can lastly be sorted considering rotation. In the image dataset, the red compartment found in each image stands out. Since any compartments within an agar plate typically need to be distinguishable to the human eye, color-based landmarks could be generally assumed here. The location of the segmented compartment will then be used to calculate the orientation.

Selecting the most prominent color should result in the most accurate output. The first step is to analyze both the HSV- and RGB-space of the image, generating corresponding 3D scatter

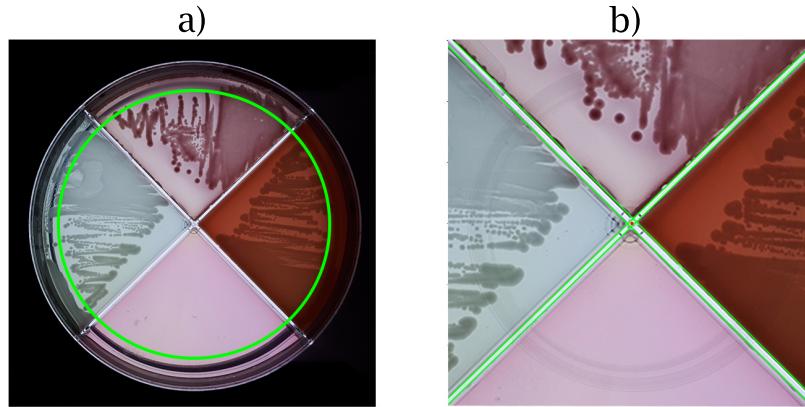


Figure 4.4: a) illustrates the new ROI boundary. b) show the edges and intersection (i.e., center point) of the compartment edges.



Figure 4.5: Flow chart to identify the orientation of the agar plate.

plots of their color space. The color space providing the most visually separable and localized tones of red should be chosen. From the color space chosen, values from the scatter plot can be read to apply approximated values to the image.

With the selected compartment segmented, the intersection points between the outer contour in *Section 4.1.1* and the lines from *Section 4.1.2* are calculated as a first step to find the landmark points. Pixels on a line between each intersection point will be checked clockwise, forming a rectangular bounding box. A mean value of the pixel intensity on each line will be calculated. The line with the highest mean value should be the one crossing the segmented compartment, and the start- and end coordinate of the line will then define the first two landmark points. Remaining two intersection points will then be sorted clockwise in ascending order. From the sorted intersection point, all key-points from the previous two sections will lastly be sorted considering the rotation.

4.1.4 Image registration

The final touch is to transform the input image to match a reference using image registration. Since each input image will be different, key points from the previous sections will be sorted automatically and matched to a reference. The reference will be defined by key points representing, in this case, a completely symmetrical agar plate structure, as shown in *Figure 4.6*. Based on the sorted landmark points from *Section 4.1.3*, all key points gathered so far will be sorted considering the rotation.

With the key points and its corresponding matches, the homography can now be calculated to finally warp the image to the pre-defined reference points, thus completing the image registration.

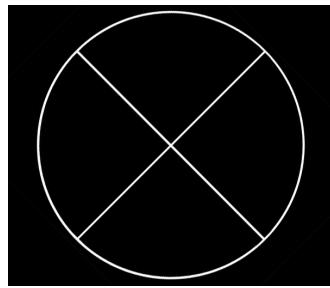


Figure 4.6: Illustration of a structure based on pre-defined reference key points.

Lastly, the elliptical projection of the agar plate, as illustrated in *Figure 4.7*, needs to be considered to improve the perspective accuracy in the more extreme cases. Since the goal is to produce a flat projection of the agar plate, additional iterations of the whole process will be applied to cope with any remaining perspective distortion.

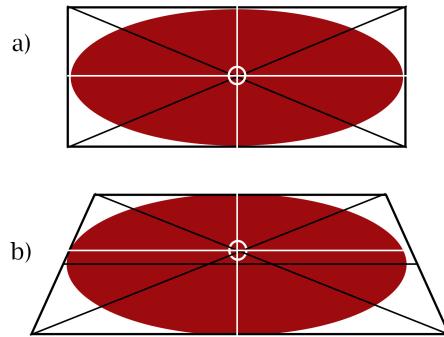


Figure 4.7: a) Flat ellipse. b) Foreshortened circle (spatial illusion).

4.2 Evaluation

The approach of this thesis is to evaluate if a combination of image processing techniques can be used to solve the problem description. The thesis problem is assumed to be solved if the criteria in *Section 1.3* are fulfilled. Therefore, the basis of evaluation for this method is to examine if the workflow, shown in *Figure 4.1*, can produce results that fulfill the criteria.

Each processed image will be evaluated manually to check if they fulfill the criteria. The evaluation will be done by comparing the output to a template overlay, as seen in *Figure 4.8*. By checking that the compartment edges follow the red cross, representing the center of the image, perspective and positioning are assumed to be valid. The red circle validates the masking and scale of the agar plate. The rotation of the agar plate will lastly be validated based on the orientation of the red compartment in relation to the green arrow.

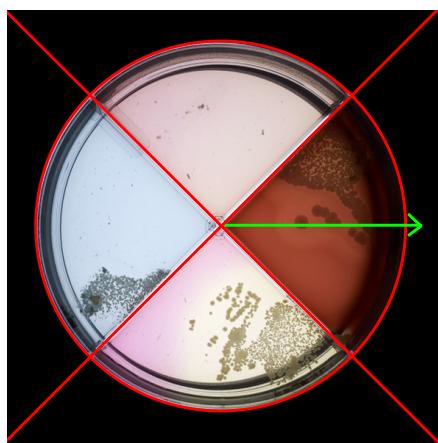


Figure 4.8: Illustrates an output with a reference template overlay.



5 Results

This chapter will present the results of choosing and implementing the different techniques explained in the method. Moreover, images on agar plates before and after using the different techniques will be shown.

5.1 Implementation

Using a dataset of 300 images, *Figure 5.1* is a general representation of the five most distinctive types of input images of the dataset in terms of rotation, perspective, bacterial growth, lightning, and positioning. Figures throughout the implementation are based on outputs visually and pedagogical enough to prove the method of their respective step. Minor pixel-level variations throughout the dataset could be found, still giving reasonable final results.

The method was implemented using Python 3.8 and the open library OpenCV 4.2. Results were produced on the following hardware: Intel I7-5600 2.59GHz, 8GB DDR3L 1600MHz, and 256GB SSD.

5.1.1 Identify and mask the agar plate

There is a significant positive relationship with using SCB to intensify the primary colors, brightness, and contrast of images. This result is consistent throughout the image data set, which had varying color and lighting conditions. SCB managed to intensify both brightness and the RGB-colors to give a clearer image and visible agar plate. For pictures with very low brightness (first row in *Figure 5.2*), the prior balance of brightness and contrast gave some improvement in the color balance. However, in most cases, there was not any difference, and SCB provided good results on its own (see the second row in *Figure 5.2*).

5. RESULTS

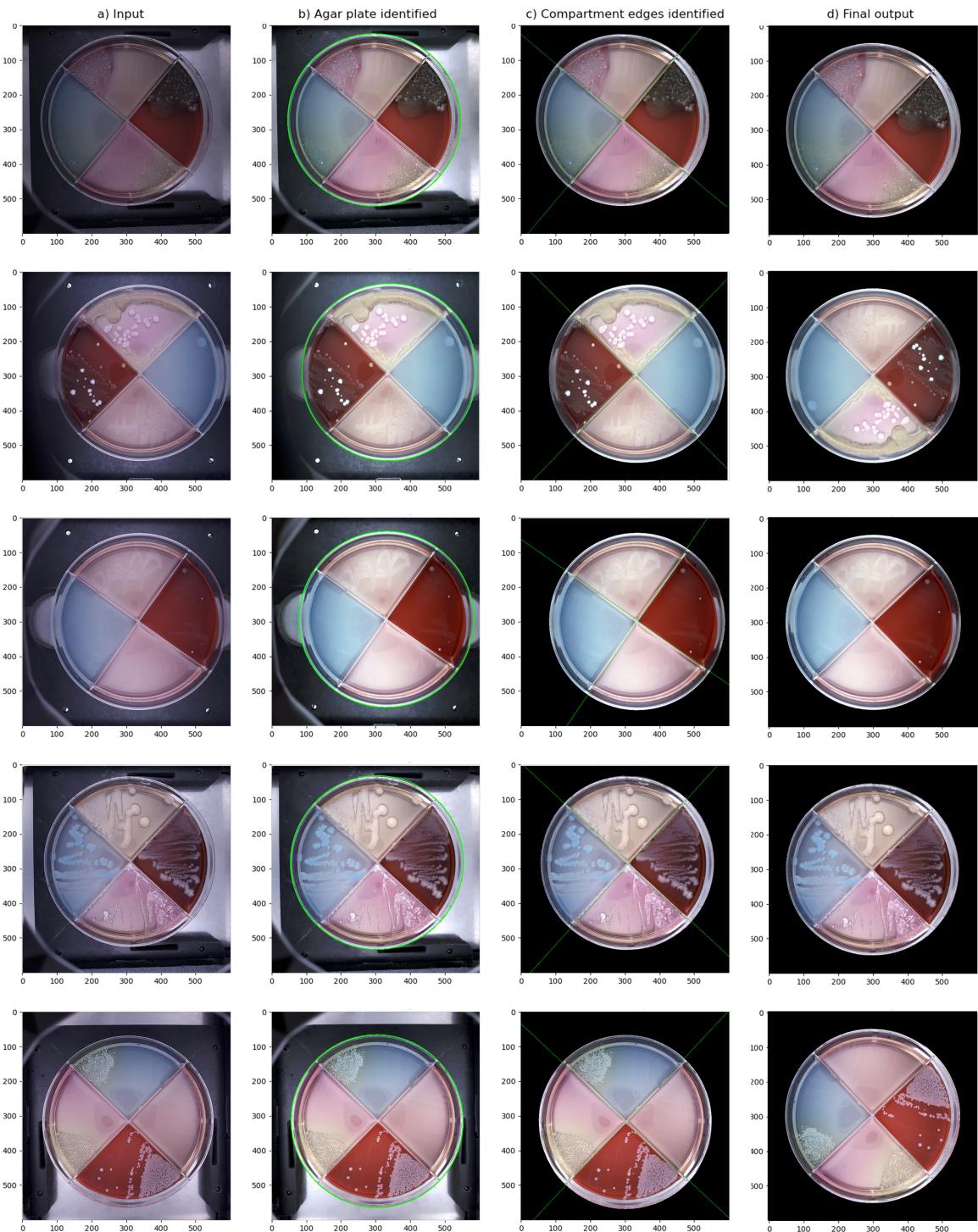


Figure 5.1: a) Input image. b) The outer contour of the agar plate identified. c) Compartment edges identified. d) Final output.

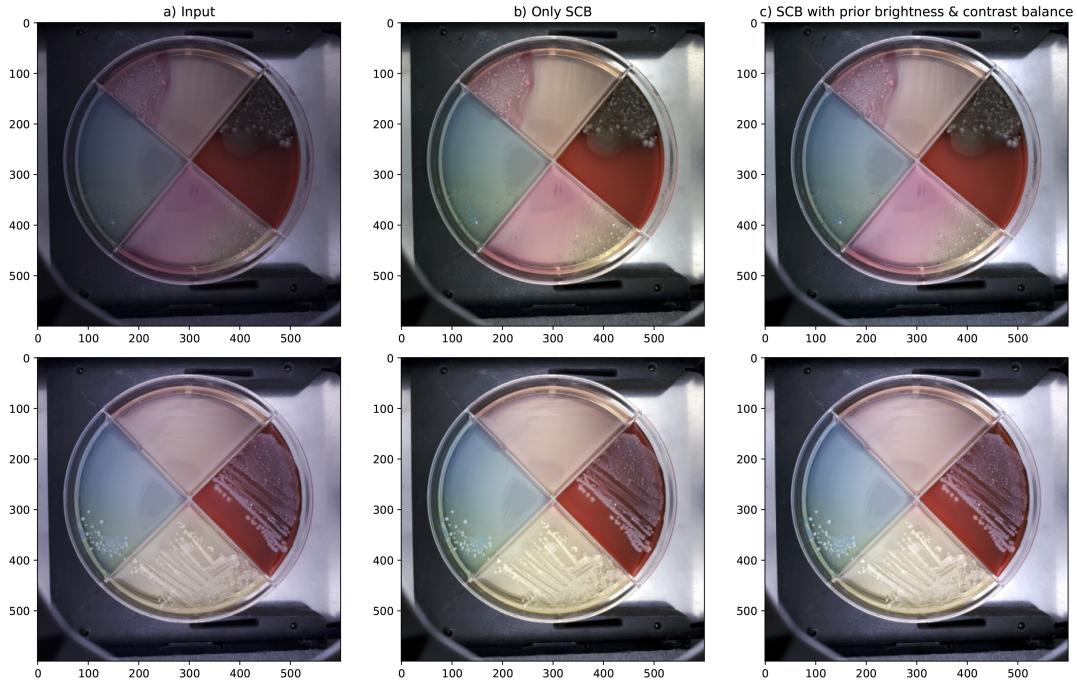


Figure 5.2: a) Input image of an agar plate under dark conditions. b) Image processed with SCB. c) Image processed with SCB with prior brightness and contrast balance

As seen in *Figure 5.3*, Canny proves to be a useful tool for initial edge detection. Adding another iteration of Gaussian blur before Canny, gave a significant improvement in terms of removing image noise such as bacteria clusters and highlighting the actual structure of the agar plate. The result throughout the dataset was mainly dependent on the extent of bacterial growth. Some more extreme cases may have caused the agar plate contour to be connected to features in the background. These deviations were later addressed using RANSAC at the end of this section. For this work, the lower and upper threshold values for Canny were set 0, respectively 255, to keep only the most intense pixels. The Gaussian blur sigma value was set to 0.8.

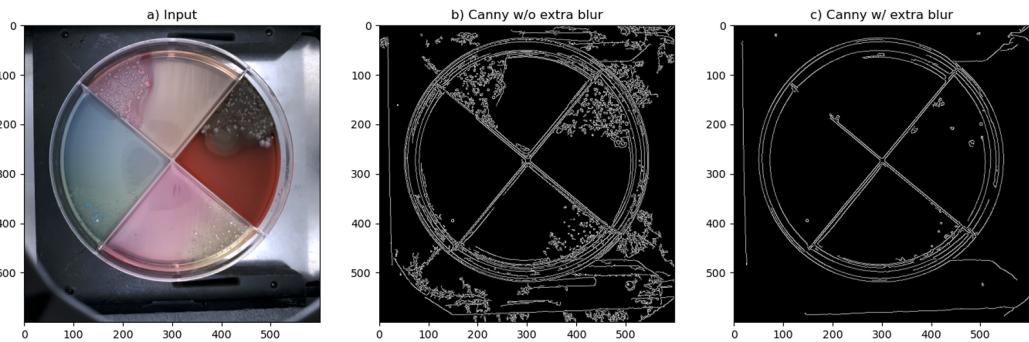


Figure 5.3: a) Input image. b) image processed with Canny Edge Detection. c) Image processed with Canny Edge Detection but, with extra Gaussian Blur.

5. RESULTS

Figure 5.4 shows that RANSAC effectively improves the approximation of the outer contour in deviances occurring during border following. *Figure 5.4 b)* shows the deviances occurring with Border Following. Closing was made with five iterations of dilation, followed by six iterations of erosion, giving a consistent output for the RANSAC to correct. The sixth iteration of erosion proved useful to remove additional background noise.

Using closing on the Canny output in *Figure 5.3 c)*, the outermost edges could be merged, forming one thick edge, thus providing a more distinctive outer edge, as seen in *Figure 5.4 a)*.

The found contour was validated with an 80% matching rate to their corresponding CHT shape. The matching rate was set a bit lower to cope with the more extreme cases. Images with a matching rate of over 85%, in this case, proved in general to provide the most accurate results.

RANSAC was set to 100 iterations, which gave accurate results with providing the elliptical contour, as seen in *Figure 5.4 c)*. However, it was quite time-consuming with a computation time of up to 50 seconds.

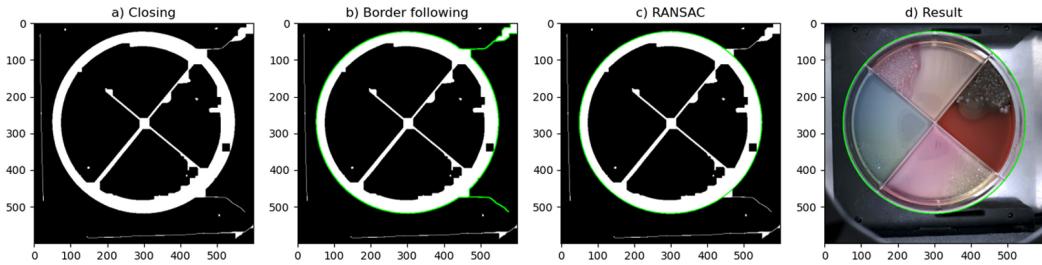


Figure 5.4: a) Closing applied to fix broken canny edges. b) Border following of the outermost contour. c) RANSAC to fit an ellipse on the identified outer edge. d) The final contour of the agar plate identified.

Lastly the ROI masking was made using the elliptical contour extracted from the RANSAC, which gave good results as seen in *Figure 5.5*

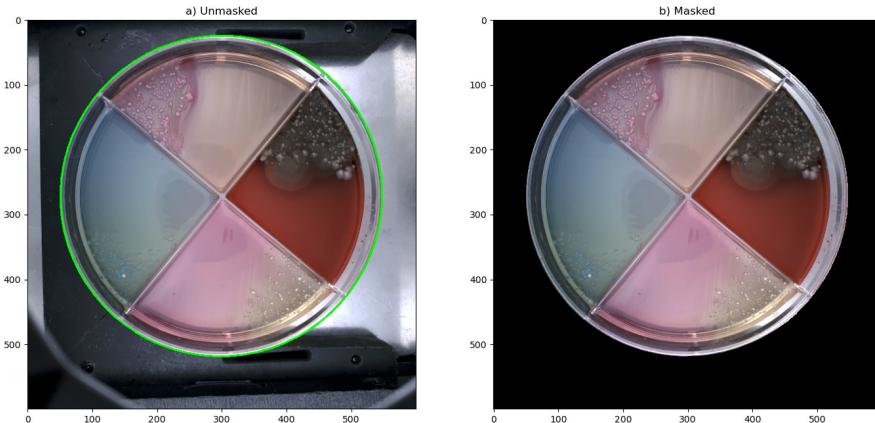


Figure 5.5: a) Agar plate with detected outer contour. b) Pixels outside the outer contour masked .

5.1.2 Identify compartment edges

Canny, together with the additional Gaussian blur, proved to be an efficient method for identifying the compartment edges. The Gaussian blur sigma value was set to 1.2, a bit higher as opposed to the value in *Section 5.1.1*. Canny still operated with the same lower and upper threshold values as before. As a consequence of bacteria clusters growing close to the compartment edges, some edges could not be identified properly. This resulted in some deviations, as seen in the second row of *Figure 5.6 a)*.

Morphological Closing gave accurate results in merging any lines, as seen in *Figure 5.6 b)*. However, bacteria clusters close to the compartment edges did sometimes interfere with the edge, as in the second row of *Figure 5.6 b)*.

Lastly, the first row in *Figure 5.6 c)* shows that Skeletonization produced an accurate representation of the centermost edge in *Figure 5.6 b)*, while the input in the second row was affected by bacteria growing close to the edges. Fairly accurate final results of finding the compartment edges were still produced in both cases in *Figure 5.7*, due to the validation.

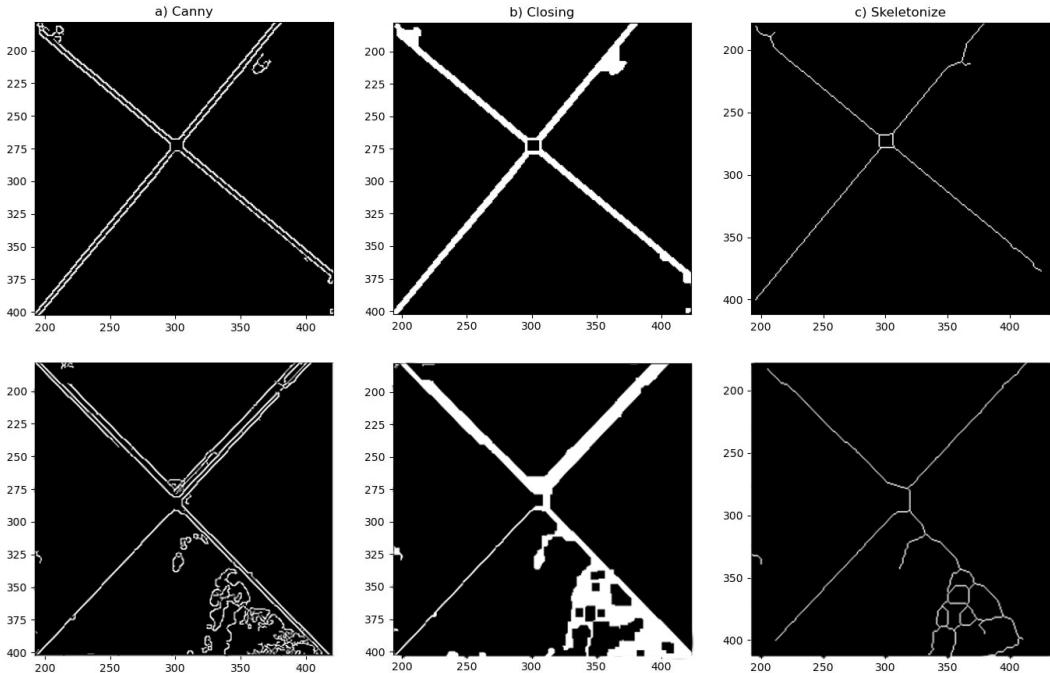


Figure 5.6: a) Masked binary Canny image. b) Morphological Closing to form one thick line. c) Skeletonize of the thick line, giving a pixel-wide centerline.

5. RESULTS

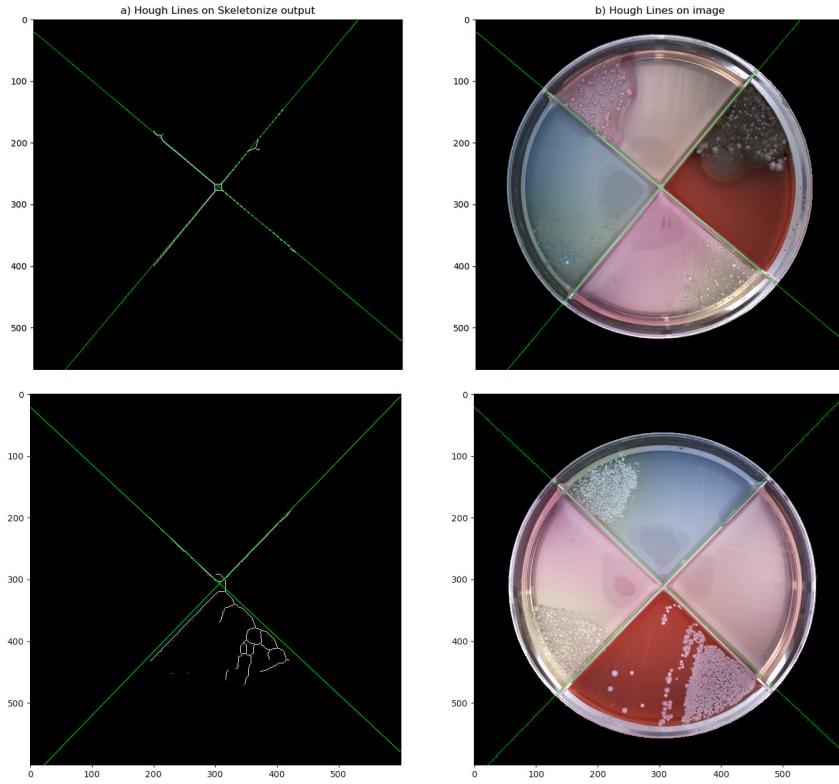


Figure 5.7: a) Lines found by Hough Lines over skeletonized output. b) The same lines are drawn over the input image.

5.1.3 Identify agar plate orientation

Changing the values within the HSV-space gave the most distinctive segmentation of any red pixels. HSV-values were selected to work for the whole dataset, which gave a lower threshold of $[H = 0, S = 130, V = 0]$ and a upper threshold of $[H = 179, S = 255, V = 255]$ to segment the red compartment (see *Figure 5.8 b*). Calculating the red RGB mean value of each line to identify the orientation proved to give consistent results. In many cases, it even worked without the segmentation, but the segmentation was still needed to give accurate results on the entire dataset. With the segmentation of the red compartment, the orientation could easily be identified, as seen in *Figure 5.8 c*.

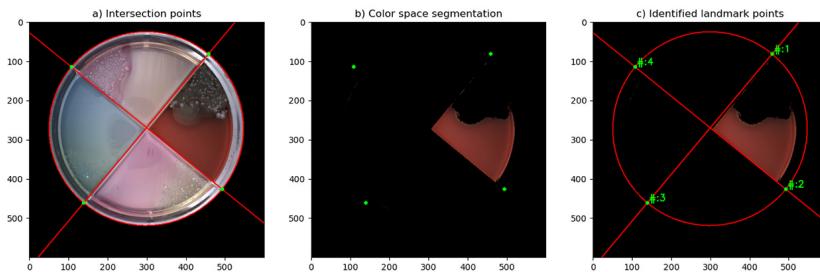


Figure 5.8: a) Intersection points between the outer contour and the identified compartment edges. b) Segmented red compartment using color space segmentation. c) Sorted intersection points.

5.1.4 Image registration

As seen in *Figure 5.9 a)*, key points were able to be matched to their corresponding reference points in *Figure 5.9 b)*. The figure only shows a total of 37 key points for illustrative purposes. The final outputs in *Figure 5.1 d)* were processed with a total of 10 005 key points, divided into 2 500 key points per line, 5 000 for the ellipse, and 5 for the center and landmark points.

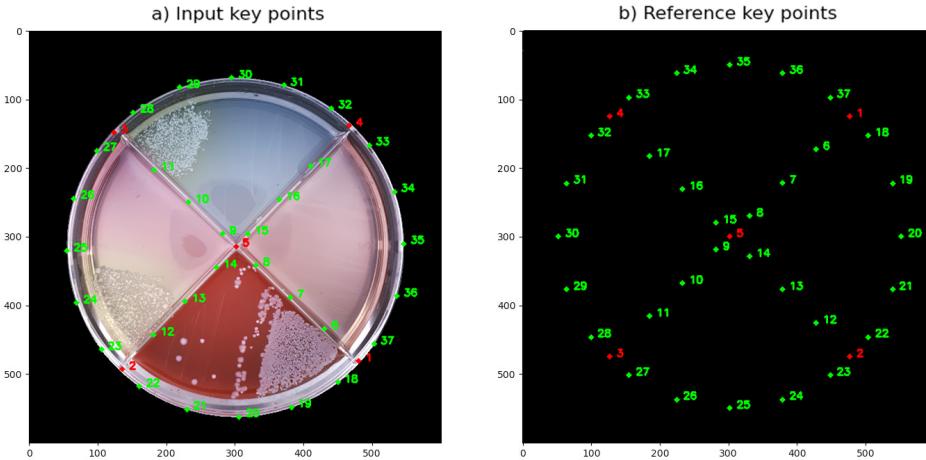


Figure 5.9: a) Identified key points for the input image. b) Reference key points.

Finally, *Figure 5.10* shows very promising results for any perspective distortion still occurring after the first full iteration. A second iteration provided a near-perfect, circular projection of the agar plate.

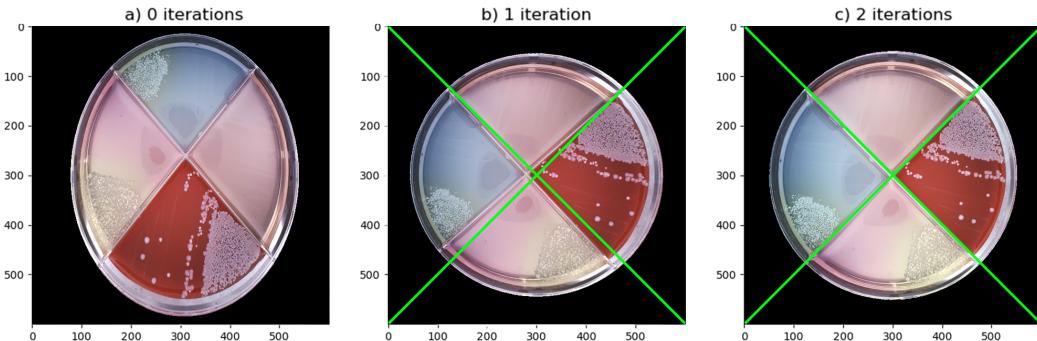


Figure 5.10: a) Image constructed with extreme perspective distortion before registration. b) The result after one iteration. c) The result after two iterations.

5.2 Evaluation

Evaluating the final output proved that the outputs were valid considering a small margin of error. *Figure 5.11 a)* shows a near-perfect transformation and rotation, while the remaining outputs show slight variations compared to the overlay template. However, the variations

5. RESULTS

were minimal, and the red compartment was in all cases oriented to the right according to the green arrow throughout the data set.

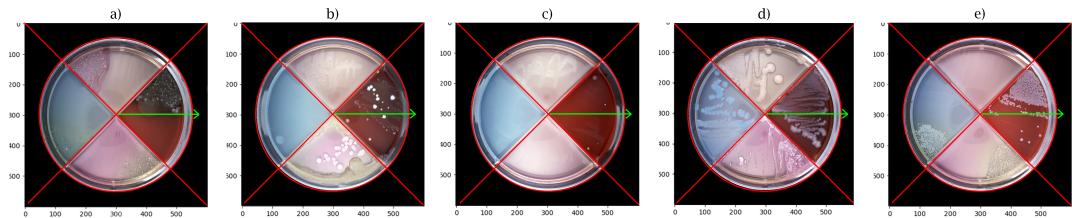


Figure 5.11: Validation of final output from *Figure 5.1 d)*



6 Discussion

This chapter discusses and analyzes the results and methods derived from the previous chapters. Additionally, the work will be put in a wider context.

6.1 Results

Overall the results look promising, with only some minor variations in perspective and rotation (see examples in *Figure 5.1 d*). The biggest reason for any output variations seems to be deviations when finding the compartment edges, essentially due to the extent of bacteria occurring, as well as their positioning. While *Figure 5.1 b*) shows nearly identical results, *Figure 5.1 c*) shows slight variations in the positioning and angle of the identified edges.

Using Canny with the additional blur seems to be a promising way to cope with any noise from bacteria growth. Very accurate final results were possible to produce when slightly adjusting the parameters of Dilate, Erosion, Canny, and Gaussian blur individually for each image. However, parameters were set to work with the whole dataset to provide a more generally applicable solution to the thesis problem.

The fact that the agar plate is circular, and not dependent on the background made it much more difficult to identify its orientation. Usually, some landmark points in corners, shapes, or the background can be found, but here a color segmentation was needed as an addition to defining the rotation.

Figure 5.1 in each step shows that the criteria in *Section 1.3* are met. Therefore, the thesis problem description is assumed to be solved. How the implementation fulfills the criteria can be shown in *Figure 5.1* as follows:

- *The outer edge of the agar plate shall be identified. The compartment edges shall be identified as well.*

Figure 5.1 b) shows that Canny, Morphological Closing, Border Following, and RANSAC were successful in providing an elliptical shape around the outer contour of an agar plate. *Figure 5.1 c*) shows that the compartment edges can be identified using Canny, Morphological Closing, Skeletonize, and Hough Transform.

- *Pixels outside the agar plate should be masked to remove background noise.*

As seen in *Figure 5.1 c)*, masking proved to be consistent and any relevant noise is masked if the outer contour of the agar plate is identified correctly. *Figure 5.1 c).*

- *Depending on the angle, position, and scale, the image should be adjusted to match a reference image.*

Figure 5.1 d) shows the input image in *Figure 5.1 a)* is adjusted in scale, position, perspective, and rotation based on a reference image.

6.2 Method

The method proved to work, thus leading to good results, but could mainly be improved in some aspects.

For example, additional validation of the Canny output in *Figure 4.3* could be implemented to validate that the compartment edges are identified, as shown in *Figure 4.4 b)*. Either the contrast and brightness could be adjusted as during the validation in *Figure 4.2*; alternatively the Gaussian blur sigma adjusted depending on the situation. If this were to be successful in all cases, the Closing and Skeletonize would not be needed. Instead, the centermost part of the compartment edges could be simply calculated from the four edges identified.

Even though using RANSAC increased the accuracy of identifying the agar plate, it proved to be relatively computationally expensive. The number of RANSAC iterations was a consequence of using parameters suitable enough for the whole dataset and could be lowered if the pre-processing was optimized. However, processing speed may only be of importance when processing larger data sets. In most cases, only a few images will need to be processed per occasion.

For application areas where processing speed is of the essence, the overall processing speed of larger data sets could be decreased using multi-threading, processing multiple images at once. Additionally, CHT alone could be used to replace the whole Section 5.1.1., still producing good-enough results in cases with little to no perspective distortion.

Instead of processing each output multiple times, any perspective distortion could be directly calculated with the major and minor axis of the circular projection of the agar plate. The major and minor axis should then correspond to the compartment edges and the contour found in Section 4.1.1. Ideally, this operation would reduce the need to only a single processing iteration. Alternatively, a validation of the compartment edges positioning could be done. If there still was an offset between the compartment edges of the input and the reference model, the second iteration could be initiated.

Lastly, the manual validation using an overlay reference template could be automated by calculating the Hu Moments. A representation of the outer contour, the compartment edges, and the rotation of each output could be constructed. The representation could then be used as a comparison to the reference template. The moment value of each representation could then be validated based on a percentage with a reasonable margin of error.

6.3 The work in a wider context

Imagining the social and ethical aspects of image processing is difficult. However, using image processing to pre-process images before being used by a bacteria classifier improves the possibility of correctly identifying the right type of bacteria. Therefore, this work could be a base to improve the input used for all types of automated bacteria classifiers using agar plates. Improving the results of automated bacteria classifiers can not only for mastitis, improve the time-consuming and lengthy diagnosis process.



7 Conclusion

In conclusion, a combination of image processing algorithms and techniques were found to answer the problem description. The solution proved to be invariant to scale, perspective, and rotation, which consequently opens up to a wider area of application.

During the research for the thesis, few papers were found with a similar problem description. A substantial amount of previous work has been done by others on edge detection and object identification, which made it possible to construct a good base for this work.

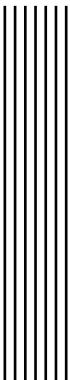
Papers found on image registration were based on the registration of images of the same object under varying conditions. However, the problem description of this thesis addressed the registration of different objects, but of the same type (an agar plate). Therefore, the characteristics of the problem made it necessary to find a more generally applicable approach.

The collection of techniques described in this thesis should, besides processing an agar plate, apply to other types of objects as well. Hopefully, it could add some improvement when used in, e.g., bacteria classification or other areas.

7.1 Future work

Future work for this thesis could be as follows:

1. Improving Canny, or finding other methods, that could mask bacteria clusters would improve the accuracy of identifying the compartment edges.
2. An implementation to cope with the elliptical projection distortion can be done, so that no more than one iteration of the whole process is needed.
3. Improve the accuracy of Canny and Border Following when detecting the outer edge of an agar plate. Improving the accuracy will remove the need for RANSAC thus, improving the computational speed.



Bibliography

- [1] Nicolas Limare, Jose-Luis Lisani, Jean-Michel Morel, Ana Belén Petro, and Catalina Sbert. "Simplest Color Balance". In: *Image Processing On Line* 1 (2011), pp. 297–315. DOI: [10.5201/ipol.2011.llmps-scb](https://doi.org/10.5201/ipol.2011.llmps-scb).
- [2] Li Xuan and Zhang Hong. "An improved canny edge detection algorithm". In: *2017 8th IEEE international conference on software engineering and service science (ICSESS)*. IEEE. 2017, pp. 275–278. DOI: [10.1109/ICSESS.2017.8342913](https://doi.org/10.1109/ICSESS.2017.8342913).
- [3] AM Raid, WM Khedr, MA El-Dosuky, and Mona Aoud. "Image restoration based on morphological operations". In: vol. 4. 3. 2014, pp. 9–21.
- [4] Gupta Rakesh and Kaur Rajpreet. "Skeletonization algorithm for numeral patterns". In: vol. 1. 1. 2008, pp. 63–72.
- [5] Satoshi Suzuki and KeiichiA be. "Topological structural analysis of digitized binary images by border following". In: *Computer Vision, Graphics, and Image Processing* 30.1 (1985), pp. 32–46. ISSN: 0734-189X. DOI: [https://doi.org/10.1016/0734-189X\(85\)90016-7](https://doi.org/10.1016/0734-189X(85)90016-7). URL: <http://www.sciencedirect.com/science/article/pii/0734189X85900167>.
- [6] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: vol. 24. 6. New York, NY, USA: Association for Computing Machinery, 1981. DOI: [10.1145/358669.358692](https://doi.org/10.1145/358669.358692). URL: <https://doi.org/10.1145/358669.358692>.
- [7] Richard O Duda and Peter E Hart. "Use of the Hough transformation to detect lines and curves in pictures". In: vol. 15. 1. ACM New York, NY, USA, 1972, pp. 11–15.
- [8] Romi Fadillah Rahmat, Tengku Chairunnisa, Dani Gunawan, and Opim Salim Sitompul. "Skin color segmentation using multi-color space threshold". In: *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*. IEEE. 2016, pp. 391–396. DOI: [10.1109/ICCOINS.2016.7783247](https://doi.org/10.1109/ICCOINS.2016.7783247).
- [9] Sayan Nag. "Image Registration Techniques: A Survey". In: Center for Open Science, 2017. DOI: [10.31224/osf.io/rv65c](https://doi.org/10.31224/osf.io/rv65c).
- [10] Barbara Zitova and Jan Flusser. "Image registration methods: a survey". In: vol. 21. 11. Elsevier, 2003, pp. 977–1000. DOI: [https://doi.org/10.1016/S0262-8856\(03\)00137-9](https://doi.org/10.1016/S0262-8856(03)00137-9).

- [11] Zhihu Huang and J. Leng. "Analysis of Hu's moment invariants on image scaling and rotation". In: vol. 7. May 2010, pp. V7–476. DOI: 10.1109/ICCET.2010.5485542.
- [12] François Chaumette. "Image moments: a general and useful set of features for visual servoing". In: vol. 20. 4. IEEE, 2004, pp. 713–723. DOI: 10.1109/TRO.2004.829463.
- [13] P Ganesan and G Sajiv. "A comprehensive study of edge detection for image processing applications". In: *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE. 2017, pp. 1–6. DOI: 10.1109/ICIIECS.2017.8275968.
- [14] Marina Nikolic, Eva Tuba, and Milan Tuba. "Edge detection in medical ultrasound images using adjusted Canny edge detection algorithm". In: *2016 24th Telecommunications Forum (TELFOR)*. IEEE. 2016, pp. 1–4. DOI: 10.1109/TELFOR.2016.7818878.
- [15] Weibin Rong, Zhanjing Li, Wei Zhang, and Lining Sun. "An improved CANNY edge detection algorithm". In: *2014 IEEE International Conference on Mechatronics and Automation*. IEEE. 2014, pp. 577–582. DOI: 10.1109/ICMA.2014.6885761.
- [16] Milad Soltani, Saeid Zadeh, and Hamid Pourreza. "Fast and Accurate Pupil Positioning Algorithm using Circular Hough Transform and Gray Projection". In: 2011, pp. 556–561.
- [17] S Divya and AD Dhivya. "Human Eye Pupil Detection Technique Using Circular Hough Transform". In: *International Journal* 7.2 (2019), pp. 116–118.
- [18] Radu Gabriel Bozomitu, Alexandru Păsărică, Robert Gabriel Lupu, Cristian Rotariu, and Eugen Coca. "Pupil detection algorithm based on RANSAC procedure". In: *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*. IEEE. 2017, pp. 1–4. DOI: 10.1109/ISSCS.2017.8034891.
- [19] Yingdi Xie and Jun Ohya. "Efficient detection of ellipses from an image by a guided modified ransac". In: *Image Processing: Algorithms and Systems VII*. Vol. 7245. International Society for Optics and Photonics. 2009, 72450W. DOI: <https://doi.org/10.1117/12.805891>.
- [20] Siddharth Saxena and Rajeev Kumar Singh. "A survey of recent and classical image registration methods". In: *International journal of signal processing, image processing and pattern recognition* 7.4 (2014), pp. 167–176. DOI: <http://dx.doi.org/10.14257/ijcip.2014.7.4.16>.