BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

IMAGE-BASED CLUSTERING OF MICROBIAL COLONIES

SHLUKOVÁNÍ MIKROBIÁLNÍCH KOLONIÍ NA ZÁKLADĚ OBRAZU

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. JAN LÁNCOŠ**
AUTOR PRÁCE

SUPERVISOR **Ing. KAREL BENEŠ**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Master's Thesis Assignment

144027

Institut:	Department of Computer Graphics and Multimedia (UPGM)
Student:	**Láncoš Jan, Bc.**
Programme:	Information Technology and Artificial Intelligence
Specialization:	Intelligent Systems
Title:	**Image-Based Clustering of Microbial Colonies**
Category:	Image Processing
Academic year:	2022/23

Assignment:

1. Get acquinted with searching colonies of interest in Petri dishes with agar
2. Design a model for segmentation of images of Petri dishes
3. Design a model for discrimination of different cultures in the dishes
4. Implement and train both of them
5. Evaluate their performance on a suitable dataset.

Literature:

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning. MIT Press, 2016.
- Jan, Jiří. Medical Image Processing, Reconstruction and Analysis. 1. Routledge, 2020. ISBN 113831028X.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Beneš Karel, Ing.**
Head of Department:	Černocký Jan, prof. Dr. Ing.
Beginning of work:	1.11.2022
Submission deadline:	17.5.2023
Approval date:	31.10.2022

Abstract

In-lab analysis of microbial colonies grown on Petri dishes is on the frontier of efforts for total laboratory automation. The core of this issue lies in precise localization of the colonies during image analysis. The state of the art solutions often employ machine learning models. However, these models tend to be heavily reliant on existence of quality labels which leads to a data scarcity problem. The proposed thesis addresses this issue by creation of a sample generator. The robustness of the proposed solution was corroborated by successfully applying the generator both in our segmentation and colony clustering efforts, significantly raising the F1 segmentation score from 0.518 to 0.729 and achieving a subsequent V-measure clustering score of 0.830. This approach to generating synthetic data brings us one step closer towards total laboratory automation.

Abstrakt

Laboratorní analýza mikrobiálních kolonií pěstovaných na Petriho miskách je v současné době předmětem intenzivního výzkumu ve snaze o dosažení totální laboratorní automatizace. Jádro tohoto problému spočívá v přesné lokalizaci kolonií při obrazové analýze. Současná řešení nejčastěji využívají strojové učení, které je však obecně závislé na kvalitních datech, kterých je v tomto odvětví k dispozici jen velmi málo. Abych adresoval tento problém, vytvořil jsem víceúčelový generátor syntetických dat. Úspěšně jsem jej aplikoval jak při segmentaci tak při shlukování kolonií. Výsledné segmentační F1 skóre se mi podařilo navýšit z 0,518 na 0,729 a při shlukování jsem s využitím této segmentace dosáhl V-measure skóre 0,830. Prací navržený přístup ke generování syntetických dat nás posouvá o krok blíže k plné laboratorní automatizaci.

Keywords

data augmentation, u-net, convolutional neural network, clustering, k-means, agar plate, petri dish, microbial colonies, laboratory automation, data generation, synthetic data, image analysis, segmentation, genetic algorithm, machine learning

Klíčová slova

augmentace dat, u-net, konvolutní neuronové sítě, shlukování, k-means, Petriho misky s agarem, mikrobiální kolonie, laboratorní automatizace, generace dat, syntetická data, analýza obrazu, segmentace, genetický algoritmus, strojové učení

Reference

LÁNCOŠ, Jan. *Image-Based Clustering of Microbial Colonies.* Brno, 2023. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Karel Beneš

Rozšířený abstrakt

Jedním z hlavních využití Petriho misek s agarem v biologických laboratořích je množení a následná analýza mikrobiálních kolonií. Přestože je tohle běžná a nezbytná praxe, dosud nebyla plně automatizována. Přístroje dnes sice umí naočkovat misku s agarem a analyzovat izolované vzorky, jsou to však stále laboratorní technici, kteří musí rozhodovat, které z možných stovek kolonií jsou hodné další, sofistikovanější analýzy. Automatizace tohoto kroku spočívá v přesné lokalizaci mikrobních kolonií na misce a jejich shlukování dle předpokládaných mikrobiálních druhů. Přístroj by následně mohl navzorkovat minimální počet těchto kolonií pokrývající rozmanitost celé misky, a to bez nutnosti lidské interakce.

Ačkoli lokalizace byla již vědeckou komunitou adresována, stále chybí obecně funkční řešení pokrývající širokou škálu různobarevných agarů a mikrobiálních druhů. o shlukování libovolného počtu mikrobiálních druhů se naproti tomu dle mého názoru zatím nikdo nepokusil. Oba tyto problémy se potýkají s nedostatkem kvalitních dat. Pro zapojení přístupů hloubkového učení, které jinak ve snahách o automatizaci v tomto poli převládá, jsou potřeba kvalitně pořízené obrázky kultivovaných Petriho misek spolu se segmentačními a shlukovacími maskami. Segmentační binární maskou jsou přesně definovány oblasti výskytu mikrobiálních kolonií. Shlukovací maska naproti tomu indikuje, které kolonie lze seskupit na základě svých vizuálních vlastností. Vytváření a značení obrázků kultivovaných Petriho misek je však drahé a časově náročné. Proto jsem navrhl způsob, kterým lze opětovně využít již existujících obrázků kultivovaných Petriho misek za účelem datové augmentace.

Nejdříve jsem z dostupných a binárně označených obrázků extrahoval jednotlivé kolonie spolu s odpovídající částí jejich segmentační masky a vytvořil jejich databázi. Jednotlivé kolonie jsem roztřídil dle jejich vizuální podobnosti. Následně jsem jednotlivým získaným výřezům kolonií pomocí techniky klíčování barev zvýšil průhlednost v místech, kde byla barva kolonií podobná okolnímu agaru. Části obrázku indikované segmentační maskou jako neobsahující kolonii jsem zprůhlednil úplně. Výsledkem je databáze semi-transparentních kolonií bez pozadí a s odpovídajícími segmentačními maskami. S využitím genetického algoritmu následně umisťuji tyto kolonie na obrázky Petriho misek s nekultivovaným agarem. Tyto prázdné misky jsem získal právě pro tento účel a obsahují proto velmi širokou škálu různobarevných agarů. Za účelem dosažení realismu nakonec simuluji jev pozorovatelný poblíž některých kolonií, kterým je pozměnění barvy agaru, například vlivem hemolýzy. Implementovaný generátor je schopen produkovat rozmanité misky ve velkém množství, a to rovnou s odpovídajícími maskami jak pro segmentaci, tak shlukování.

Abych zhodnotil praktickou využitelnost těchto dat, natrénoval jsem segmentační U-Net model na původně dostupné, reálné datové sadě. Dodatečně jsem natrénoval čtyři další modely stejné architektury, na čtyřech modifikovaných datových sadách. Abych otestoval, zda funguje klíčování i simulování poškození agaru, rozšířil jsem připravené datové sady o čtyři identicky generované syntetické datové sady. Zatímco vybrané kolonie, jejich modifikace, uspořádání i agar byly vždy skrze datové sady identické, lišily se v použití klíčování a simulaci poškození agaru. Po vyhodnocení všech pěti natrénovaných modelů na původní testovací datové sadě, která byla rozšířena o dříve neviděné reálné obrázky s různými barvami agarů, lze tvrdit, že tento přístup funguje a výrazně zlepšuje úspěšnost segmentace. Klíčování, stejně jako simulace poškození agaru, modely vylepšují, a to jak nezávisle na sobě, tak při souběžném použití. Celková úspěšnost segmentace vyjádřená pomocí metriky F1 skóre se zvýšila z 0,52 na 0,73.

Při adresování problému shlukování kolonií jsem se rozhodl využít algoritmu K-Means. Pro nalezení optimálního počtu shluků využívám metody Knee/Elbow Point Detection.

Z hlediska extrakce příznaků navrhuji tři přístupy: shlukování RGB hodnot jednotlivých pixelů a následné přiřazení shluku kolonii dle její pixelové kompozice, extrakci příznaků U-Net autoenkodéru opět pro jednotlivé pixely a nakonec ruční specifikaci příznaků pro každou izolovanou kolonii zvlášť. Autoenkodér, z něhož jsem extrahoval příznaky, byl trénovaný na původním datasetu reálných obrázků. Vlastnosti jsou extrahovány z výstupu poslední konkatenační vrstvy po finálním převzorkování v rámci sítě. Ručně extrahované příznaky třetího popsaného přístupu jsou jednoduché vizuální vlastnosti, jako je průměrná barva, velikost nebo tvar.

Všechny tyto přístupy dosáhly srovnatelných výsledků, nicméně nejlepšího výsledku dosáhl přístup zahrnující manuální extrakci příznaků pro jednotlivé kolonie. Přestože dosažené výsledky nelze srovnat s jinými výzkumem, V-measure je pevně vymezená metrika a dosažené skóre 0,91 je blízko teoretickému limitu ideálního shlukování, který činí 1.0.

Image-Based Clustering of Microbial Colonies

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Ing. Karel Beneš. Supplementary information was provided by Ing. Michal Čičatka and Ing. et Ing. Jan Karásek, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

. . . .
Jan Láncoš
May 16, 2023

Acknowledgements

I would like to thank the thesis supervisor Ing. Karel Beneš for his time, dedication and many valuable suggestions. Thanks are also due Ing. Michal Čičatka for his expert knowledge, comments and stimulating suggestions.

Contents

Chapter 1

Introduction

Agar-based media (or simply agar plates) are used throughout the world to study microorganisms in microbiology and medical diagnostics. Cultivating dishes with the purpose of further analysis is a daily occurrence in many laboratories, often requiring a highly qualified personnel to perform repetitive and time consuming tasks based on professional knowledge, such as inspecting each dish visually for signs of a specific microbe or manually selecting colonies of interest for the purpose of isolation on a dedicated agar plate. Eliminating the need for such tasks or at least reducing them to a minimum with the use of automation can help medical professionals and researchers around the world by reducing expenses and time spent performing mundane tasks manually.

Machine learning models are nowadays widely used for the purposes of image analysis. Agar plate in-lab analysis is no exception and deep learning is now being used to analyse high-resolution images of agar plates in many laboratory software solutions from various manufacturers. A common hurdle whenever machine learning is employed, however, is a lack of data to train on. Images of these plates are hard to obtain, since their successful cultivation and image acquisition takes up significant time and resources.

This problem is usually addressed by altering available data samples to augment the dataset. In case of agar plates, augmentation is a very viable and potent solution, as producing their largely original yet realistic images is a feasible task. Despite its potential in this matter, it is often underutilized by the scientific community. Generation of these new data samples is a key part of this thesis, as it directly improves any deep learning models employed in any form of image analysis of these agar plates.

This thesis deals primarily with the steps usually taken in between obtaining an already cultivated agar plate and a single microbial sample analysis. It proposes several methods of introducing automation or optimizing automation already in place within these steps, details the process of their implementation and in the end evaluates their effectiveness in comparison to other known existing approaches.

Chapter 2

Laboratory analysis of agar plates

When a sample presumed to contain microbes is obtained, it can rarely be analysed as is by the laboratory equipment. Regardless of its origin, the microbes are generally too diluted to be reliably retrieved or inspected. For this reason, cultivating media are used to multiply any microbes present in obtained samples to allow for easier isolation, retrieval and analysis.

2.1 Cultivation and analysis

Laboratories all over the world have diverging procedures and rules regarding producing and handling cultivated agar plates. They also employ different technologies in regards to their analysis and there is no objectively correct way to perform it. Despite this, several common steps can be generalized before the solutions start diverging significantly. Due to a lack of consensus on procedures intensified by a lack of relevant literature dealing with this issue, this section draws largely from a book summarizing the current science in regards to microorganisms [10].

There are multiple kinds and forms of media used to cultivate microbes in laboratory conditions. Usually these contain nutrients necessary for growth of any microbes deliberately placed in them. These media can vary in density, consistency and general make-up broadly. The most commonly used solid medium is an agar stored in Petri dishes. Such a dish, whether cultivated or not, is then often and will be further referred to as an *agar plate*. Many agars have selective properties and induce growth of only specific kinds of microbes. Some agars are chromogenic, meaning they change color when a certain type of microbe is present based on its biochemical activity. Generally, agar plates are used for determining the presence of particular microbes in samples of human origin, foods and other materials alike. Placing antibiotic capsules within an agar plate is a biological research technique also used for checking whether any given microbes have developed resistance for antibiotics.

Cultivating the plate entails moving a small amount of microbes onto the medium. The inoculation itself can be done manually or automatically, usually by spreading the microbes in thin long lines over the surface of the medium. If the line is sufficiently long, the microbial substrate is usually diluted enough towards its end to produce single microbes in sufficient distance from each other to form easily observable colonies originating from a single cell. An example of these “inoculation smears” can be observed in Figure 2.1.

After the plate is inoculated it is left in appropriate conditions usually for 12 – 48 hours. When retrieved, the grown microbes on the dish can be analysed. Often – prior to any

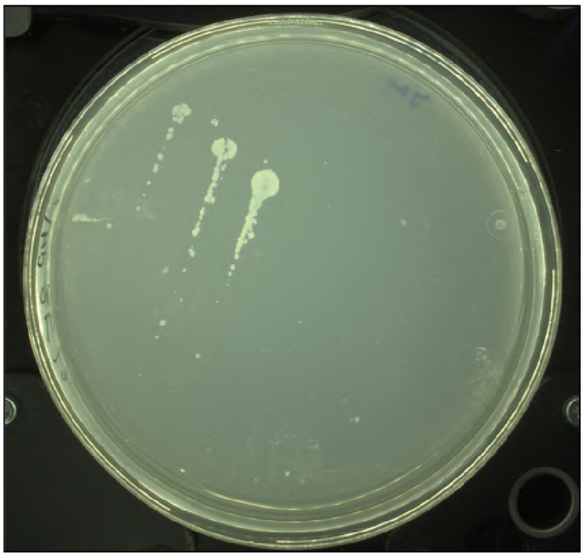


Figure 2.1: Three distinct microbial lines formed from material released from the inoculation swab. The thinning of the lines resulting in the formation of singular isolated colonies can be noted in each of the three cases.

further processing – a species needs to be isolated. A procedure called colony picking is then employed, which entails carefully obtaining very small amounts of microbes from what is perceived to be a single colony and cultivating it in a different agar plate meant to house only this particular microbe.

Microbial colonies (also called colony forming units – CFUs) can be defined as circular masses of microorganisms all originating from a single mother cell. An example of such colonies can be seen in Figure 2.2. Depending on the kind of analysis being performed, it can be done either manually or automatically with the use of dedicated instruments. The colonies can be analyzed based on their smell, they can be tested for their chemical and physical properties, or just simply observed visually. Visual observation is often a precursor to colonies of interest being analyzed further by other methods, usually after having been isolated, and is therefore often an integral part of the overall analysis.

Visually observable features are, e.g., size, shape, texture, opacity or color. Information can also be derived from the layout of the observed material on the dish, for example, when antibiotic capsules are placed upon the agar, as seen in Figure 2.3. The agar itself can also provide useful information, as is the case with the aforementioned selective agars or chromogenic agars. An agar plate can be sufficiently analyzed by a laboratory technician visually. Any ambiguity or uncertainty however often calls for an analysis by a dedicated instrument.

2.2 Current degree of automation

Machines are nowadays fully capable of cultivating dishes, picking colonies and reliably analysing microbes. Human interaction and direct input is, however, still required – as can be seen outlined in Figure 2.4 – to decide which particular colonies on a plate are of interest and worthy of further analysis, often expensive or time consuming on its own.

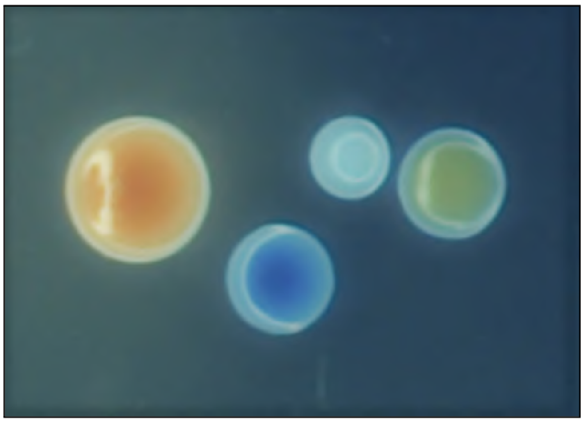


Figure 2.2: Four microbial colonies presumably originated from their respective mother cells. The diameters of these particular colonies range between 1 – 2 mm. What can be noted is the apparent roundness as well as color diversity. The apparent sickle-shaped gleams are reflections of the lights surrounding the dish when the image was taken.

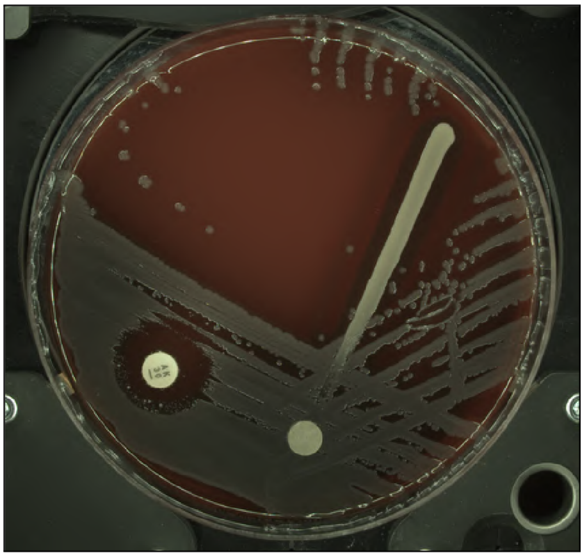


Figure 2.3: Image of a cultivated agar plate with two antibiotic capsules placed within. The capsule on the left is surrounded by an area untouched by the microbial material, implying the antibiotics successfully prevent growth of the tested microbe. The capsule on the right is either defective or the microbes are resistant, as the they seem to have no issue growing nearby.

The company Bruker Daltonics GmbH & Co. KG has developed an instrument used for accurately classifying microbial species called the MBT Biotyper [9]. This instrument uses a technique called Matrix Assisted Laser Desorption/Ionization, often abbreviated MALDI.

MALDI is a technology used to produce molecular ions from the analysed samples. These ions are created once the sample is hit by a laser pulse, which causes the sample's desorption. Mass spectrometers – usually also employing a technology called Time of Flight (TOF) – accelerate the created ions using a magnetic field in a tube of a known length. Heavier ions are detected later than lighter ones, as they are slower. The result

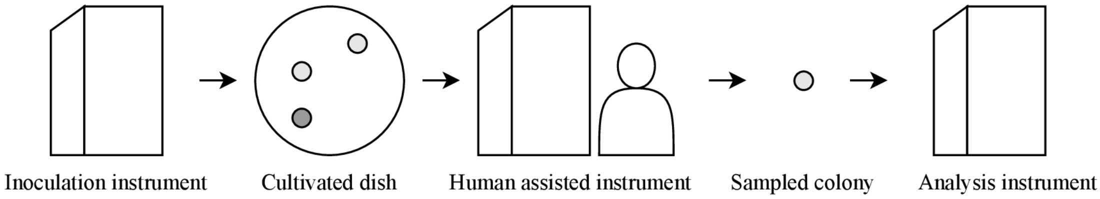


Figure 2.4: A sequence of steps usually taken when analysing a dish in a semi-automated laboratory environment. The dish is inoculated by a fully automatic instrument. A human using a semi-automated machine can then select and pick colonies of interest, and transport those to a machine doing the analysis.

of this analysis is a mass spectrum. Since each microbe has its own unique spectral “fingerprint”, it can be compared with a databank of known spectra and identified. [10]. The MBT Biotyper is pictured in Figure 2.5.

Figure 2.5: The MBT Biotyper – an instrument for classifying microbial species based on the MALDI-TOF technique. Samples for analysis are fed into the machine on prepared MALDI target plates.

To analyze and classify a sample with this instrument, said sample must first be properly transferred upon a MALDI target plate (MTP). The sample then has to be deposited with a special reagent crucial for the mass spectrometry analysis called MALDI matrix. Only then can it be inserted into the instrument.

The MBT Pathfinder is another instrument developed by the same company, which aims to automate the process of transferring samples of microbial colonies from agar plates to the previously mentioned MTPs. The instrument is at this point (May 2023) in the prototype stage and its task is to automate the process of colony picking. A crucial feature is a robust pre-selection algorithm of microbial colonies on the agar plate.

The MBT Pathfinder will usually come accompanied with a machine called Feeder – a rotating carousel with six columns, where stacks of agar plates intended for analysis can be placed, and a movable arm capable of handling single plates. Feeder can cooperate with

MBT Pathfinder, inserting single plates from its stacks into the latter. After the MBT Pathfinder is done with a plate, Feeder can again retrieve the plate and place it back into one of the stacks. These instruments can be seen pictured in Figure 2.6.

Figure 2.6: Pictured from left to right are Feeder, MBT Pathfinder and an MBT Pathfinder combined with Feeder.

When a dish is inside the MBT Pathfinder, the instrument takes images of it under six different illumination modes and presents them in the accompanying software running on a dedicated machine. The illumination modes consist of various combinations of differently placed light sources either being turned on or off. A demonstration of these modes can be seen in Figure 2.7. The segmentation model – which is a part of the accompanying software – then analyzes the resulting images and pre-selects several of the colonies detected in them. The operator can override the selection before authorizing it. The authorization step, which basically renders the MBT Pathfinder semi-automatic, is at this point necessary, mainly due to the strict regulations in the field of biomedicine which requires the final decision to be made by the operator. Once the selection is authorized, the instrument proceeds to pick the selected colonies and transfer the samples taken onto the prepared MTP. The MTP is then deposited with the reagent, placed into the MBT Biotyper and a proper analysis using the MALDI-TOF technique can proceed.

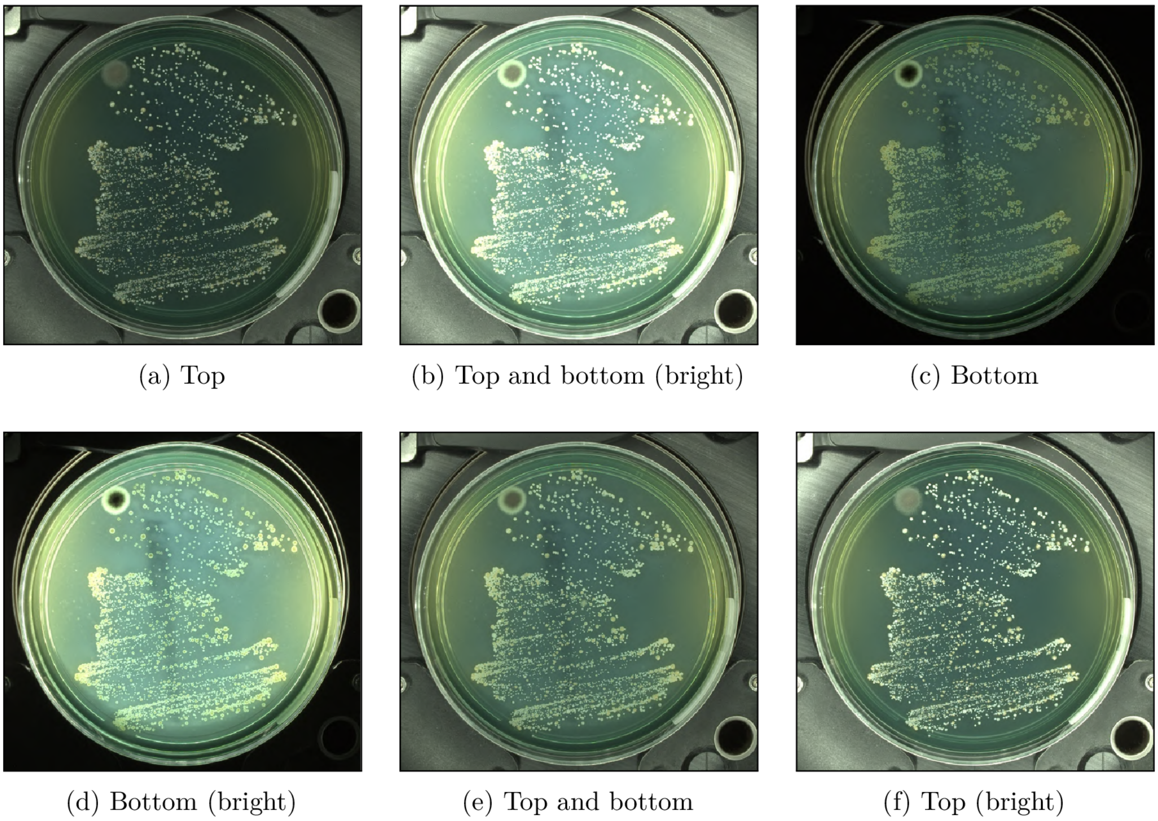
2.3 Possible improvements

The MBT Pathfinder is able to pre-select colonies fit for picking based on their successful detection and precise localization. Analysing every detected colony on a dish is however impractical and redundant. The pre-selection offered by the MBT Pathfinder could be more sophisticated and targeted.

Should there, e.g., be more microbial species present on the dish, it would potentially be very useful to be able to automatically pick every kind of microbial species for analysis precisely once, and to do so with high reliability.

This can be achieved by employing clustering methods. Once a colony gets detected, distinctive features and properties can afterwards get extracted from it. Based on these features, the colonies can be grouped optimally‘ balancing the need for minimizing the amount of unnecessary picking, while guaranteeing a maximal amount of variance among the resulting picked colonies.

Extracting features from the individual colonies, however, relies heavily on the instrument's ability to detect and segment them well. This problem should therefore be addressed



(a) Top

(b) Top and bottom (bright)

(c) Bottom

(d) Bottom (bright)

(e) Top and bottom

(f) Top (bright)

Figure 2.7: MBT Pathfinder's different modes of illumination demonstrated on a single dish. The relevance of the difference in illumination can best be seen upon inspecting what is probably mold in the top left part of the dish or the dark strip traversing the dish from top to bottom, which is its label written at the bottom.

before any attempts at clustering. A very well performing segmentation model combined with clustering of the detected colonies based on visual similarities could one day eliminate human input during this stage entirely, as outlined in Figure 2.8.

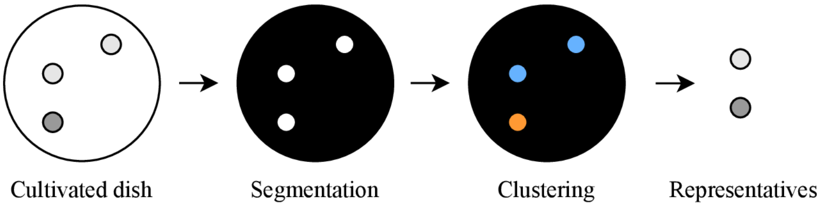


Figure 2.8: Steps to be potentially taken to render human input unnecessary in the picking stage indicated in Figure 2.4. Combining precise segmentation with precise clustering could lead to a successful sampling of all the species on the dish precisely once and without human supervision.

2.3.1 Segmentation

Precise localisation of colonies on an agar plate is a key step in any automatic processing of said plate, whether the task is colony counting, colony picking, image analysis or any other. A solid segmentation model is therefore imperative, as all the other possible procedures are directly dependant on it.

An interesting issue regarding agar plate image analysis is, that on one hand the image variance is fairly contained. Provided a standardized image format is guaranteed, coming from a specific instrument or company, one can usually define quite well the area of interest and expect similar lighting conditions. On the other hand, this advantage is partially offset by the fact that there is already a very large number of cultivating media and other such media can possibly be developed in the future. Said media can come in a variety of colors or degrees of transparency, and their coloring capabilities may differ. Antibiotic capsules may or may not be present, they can be scattered basically anywhere within the dish, be of any color or even shape. Also, since Petri dishes, as well as most agars, are transparent and often tagged with stickers or magic markers on the bottom, any algorithm must be able to differentiate what is on top of a dish, and what is on the bottom, and therefore irrelevant.

Many different approaches have already been experimented with in the matter of segmentation. One of the better known tools is the OpenCFU designed to detect circular colonies with the use of thresholding and particle filtering [6]. This was later outperformed by the AutoCellSeg which used the watershed segmentation [7]. The article however focuses more on each colony's detection, rather than its precise localization, as counting the colonies is also an important laboratory task yet to be fully and properly automated. Both these tools also require human interaction and do not utilize machine learning.

In a newer article, the researchers trained a convolutional deep belief network which acted as a feature extractor, while the segmentation itself was performed by a support vector machine [15]. This article however used a small unpublished custom made dataset, which makes any attempts at comparison complicated. Support vector machines were used in another article, along with convolutional neural networks, to detect and localize colonies in blood-based agars [5]. The team had achieved very good results and could even correctly detect and segment colony aggregates, but the focus on high-contrast, purely blood-based agars means the general applicability of their solution may be limited. The used dataset of 28 500 images has not been published.

A similar issue is addressed in a different article, in which the authors aimed for improvement in the foreground/background contrast by acquiring the image with near infrared lightning [20]. However, as the article is once again focused more on the task of colony counting, it lacks a proper evaluation of the segmentation's precision itself.

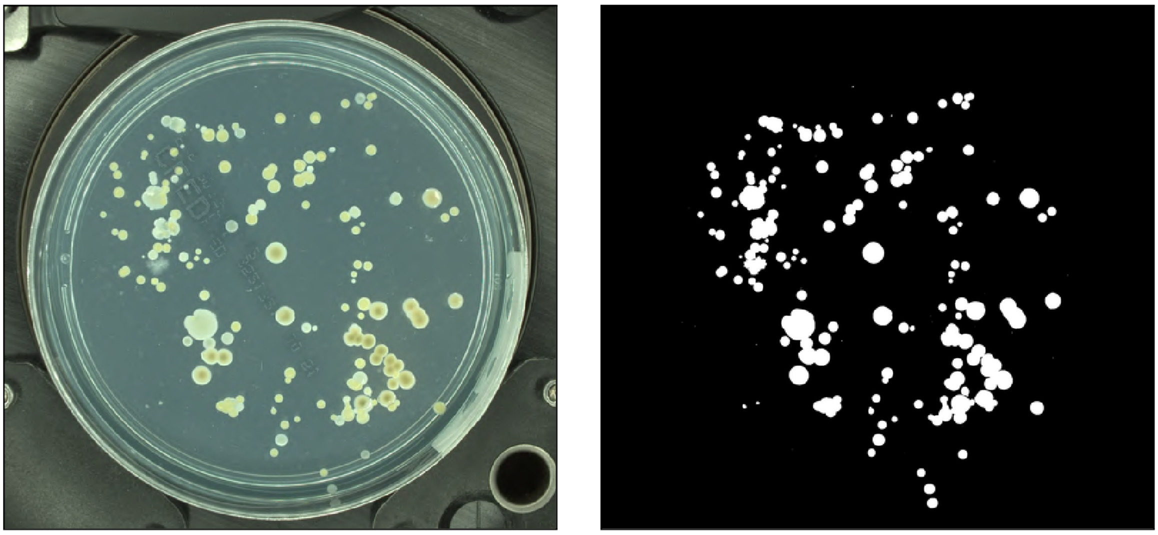
A more general approach is presented in the paper introducing CentroidNetV2 [4]. The presented algorithm is specifically designed to detect and localize many small and even overlapping objects. The team has achieved very good results, specifically in images of agar plates. It should be noted, however, that all the tested plates have contained an opaque black agar with a highly contrasting white colonies on top of it. The algorithm performs significantly worse when presented with lower contrast data, such as cell-nuclei images.

Lastly, another article applies semantic segmentation utilizing U-Nets trained on a custom unpublished dataset of 108 images [1]. The article was then extended by another article [13]. Both these articles focused on segmentation along with trying to classify the detected colonies into two groups based on their presumed virulence. While the articles achieved favourable scores in terms of precision and recall, the custom dataset used along with its limited size makes good comparison complicated.

The MBT Pathfinder's solution also uses deep learning to solve this issue. The model uses a modified ENet architecture with a tripled filter count and an additional convolutional block within the encoder. It was trained on the company's own BRUKERCOLONY dataset consisting of hand-labeled agar plate images taken by the MBT Pathfinder. Both the model and the dataset are described in a publicly available article [2]. Models, such as this one, are used frequently for segmentation of any sort.

Rather than trying to come up with a better performing alternative model or approach, a case could be made that all the machine learning attempts at segmentation could be helped at once if provided good labeled data. The deployed model has been observed to have trouble with agar plates of different colors, than of the ones it has seen prior. This suggests the problem is probably rooted within the dataset used for training. Similarly, the articles in the previous section often shared a common flaw – a small custom made dataset.

Obtaining cultivated agar plates to take images of is however no simple matter, especially in high numbers and high variance. Even assuming said plates are obtained; to be used for training and evaluating of the segmentation model, each image needs to have a corresponding binary mask indicating where the microbial material is. An example of such a mask can be seen in Figure 2.9. This mask is used as ground truth during the supervised learning, therefore it has to be as precise as possible. Cultivating the plates, acquiring the images and then creating these solutions is expensive, complicated and time consuming.



(a) Cultivated agar plate

(b) Segmentation mask

Figure 2.9: An image of a cultivated agar plate can be seen in (a) with a corresponding binary mask indicating the presence of biological material in (2.9b). This mask was produced manually and a similarly precise result would be desirable coming from the segmentation algorithm when given a similar image of a previously unseen agar plate.

To my best knowledge, aside of the BRUKERCOLONY, there is currently only one dataset of labeled cultivated agar plates publicly available, which is the AGAR dataset [11]. This one is however suitable mostly for colony counting applications, as it does include 18 000 images of plates, but it only labels each colony with a bounding box and not a precise pixel-wise binary mask. I have encountered several mentions of another dataset accross the more recent articles, called MicrobIA. I have not succeeded in finding it, nor its original article. There is also a fairly large dataset of good-resolution images of agar plates used in urine tests [3], this one is however not labeled neither for the purposes of colony segmentation, or even their detection.

An elegant solution to all of these problems possibly lies in synthetic data generation. Being able to fabricate images in line with the real data samples should in theory solve the issue of the dataset lacking variance, there would be no need for spending time cultivating the dishes, obtaining images, and most importantly, manually creating binary masks for them.

2.3.2 Colony clustering

Any kind of more complex or advanced image analysis of microbial colonies on agar plates beyond their simple detection and localization is to this day very rare.

One team of scientists in a previously mentioned article has managed to predict two distinct kinds of colony forming units [13], while another team has classified two species of bacteria based on their visual properties [14]. One article has focused on classifying two types of colonies based on their influence on purely blood-based agars [19]. To the best of my knowledge, however, nobody has yet attempted creating a general solution for clustering microbial material on agar plates into an arbitrary number of clusters based on their visual properties.

The issue with evaluating any hypothetical approaches to clustering, however, is once again the lack of good data. Samples from the BRUKERCOLONY dataset are not annotated for this purpose. The company does not possess any more images of cultivated agar plates, even less so ones where colonies can be grouped intuitively. To my knowledge, no publicly available datasets annotated for the purposes of clustering species exist. Ordering custom-made cultivated plates with a known number of microbes present, possibly even annotated, would be expensive and time consuming. Once again, the solution lies in synthetic data generation.

Chapter 3

Data augmentation using synthetic samples

The following section addresses the issues identified in the previous section by proposing a procedure for data augmentation. Taking into consideration the points made in the previous section, the following resources are available:

- the company's prototype colony picking robot MBT Pathfinder,
- MBT Pathfinder's current segmentation model, along with its original training and testing BRUKERCOLONY dataset,
- and the company's software tool for quickly producing binary masks of agar plate images.

For simplicity and convenience, the rest of the thesis will therefore limit its focus solely to data in-line with the MBT Pathfinder's specifications only. The data samples worked with will be 1900x1800 images of agar plates taken within the illumination modes provided by the MBT Pathfinder, as is the case with all of the data within the BRUKERCOLONY dataset. Access to the MBT Pathfinder's segmentation model offers a great opportunity to build upon it and also to compare any improvement of it on the same data.

3.1 Limitations

MBT Pathfinder's dataset in itself is severely lacking in both microbial and agar variance. While the microbes present in the images are unknown, the dataset does not seem to cover the known variety of colony shapes, textures and colors. And while the agars also come in many color and texture variations and new agars are still being developed, the dataset on the other hand contains only blood agars ranging from maroon to orange, clear agars and white-ish agars, as demonstrated in Figure 3.1.

It has been established before that getting new cultivated plates to produce images of is not a simple task. It is however not so complicated to obtain empty agar plates. Avoiding the need for cultivating them properly saves both time and money. A wide range of empty agar plates can be obtained in bulk, covering many different textures and colors. These empty plates can then be photographed by the MBT Pathfinder.

In terms of obtaining colony images – while the original dataset may be limited, it still contains imagery of some tens of thousands of colonies, all of them with a manually created

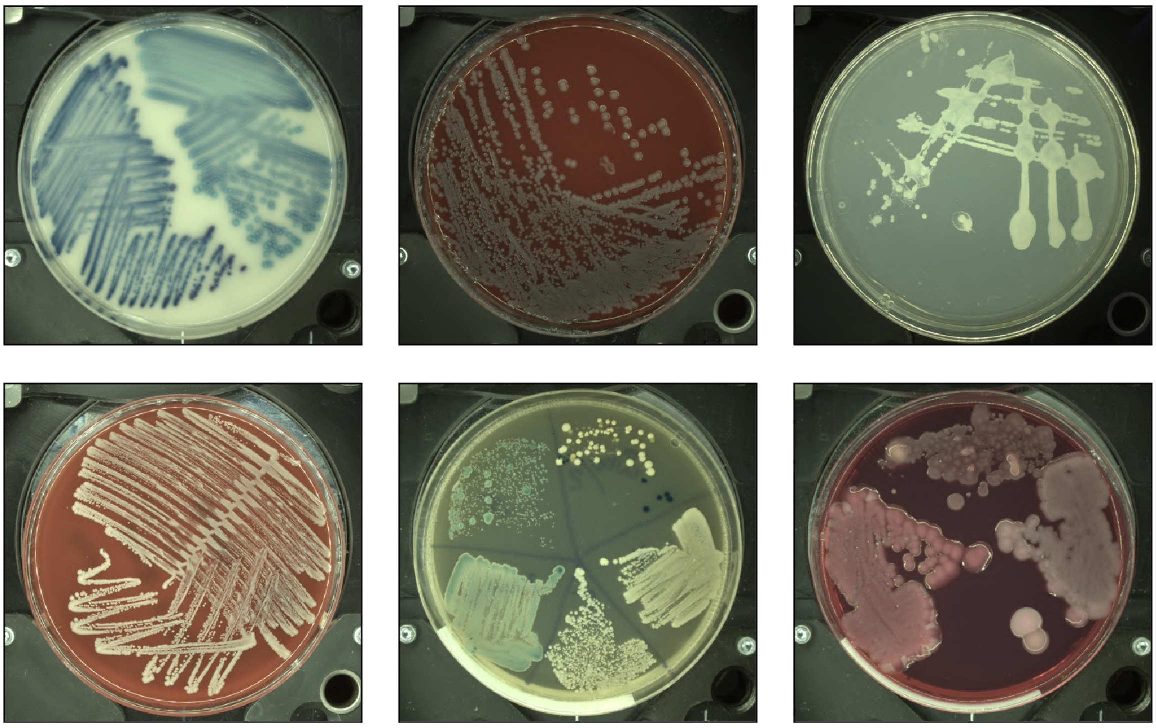


Figure 3.1: Several samples from the MBT Pathfinder's original BRUKERCOLONY dataset. The examples picked are among the most diverse of the whole dataset, yet only clear, white or blood-based agars can be seen. In most of these examples the distinct microbial smears produced by the operator during dish inoculation can be noted.

binary mask. These colonies can be extracted according to the mask and superimposed upon the image of the empty agar medium. Such an approach would significantly improve data augmentation capabilities and provide a posibility to generate samples tailored to a specific task.

While the dataset leans heavily towards what is presumably a single microbe species, many dishes contain a different looking material, in terms of color, shape, and texture. And while these colonies are usually low in numbers, possible alterations could mitigate this issue. The collection of said predominant microbe on different media however contains, apart from singular colonies, also its groups and fused lumps. Most importantly, it contains the distinct smears created by a laboratory technician or a machine when inoculating the dish. As these are often present in real images of cultivated dishes, they are invaluable to the purposes of synthetic data generation. An example of these smears can be seen in Figure 3.1

Simply placing a colony in the shape of its corresponding binary mask on an agar plate image however does not yield particularly impressive results. Since the binary mask used a threshold to obtain the sharp edges defining a colony contour, this threshold is then clearly visible once the colony has been placed on an empty agar plate image. The colonies are often semi-transparent, which means the original agar's color is visible underneath the colony. Since the original dataset had only red and white agars, this limits the use of this technique to only agars of said colors. Even if the colors are reasonably matched, however, the edge is usually still very prominent and the dish could not be passed for a real one by any means, as demonstrated by Figure 3.2.

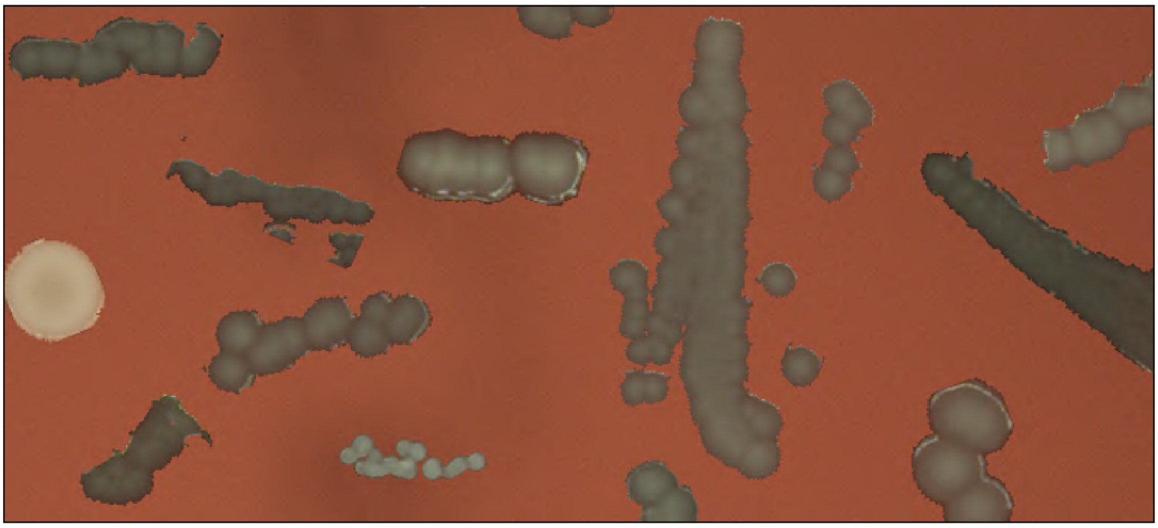


Figure 3.2: Even when colonies taken from a maroon agar plate are superimposed on another maroon agar plate, it is clear that even slight variations in shade result in visible edges. A deep learning algorithm could easily learn to search for these edges not present in real agar plate images.

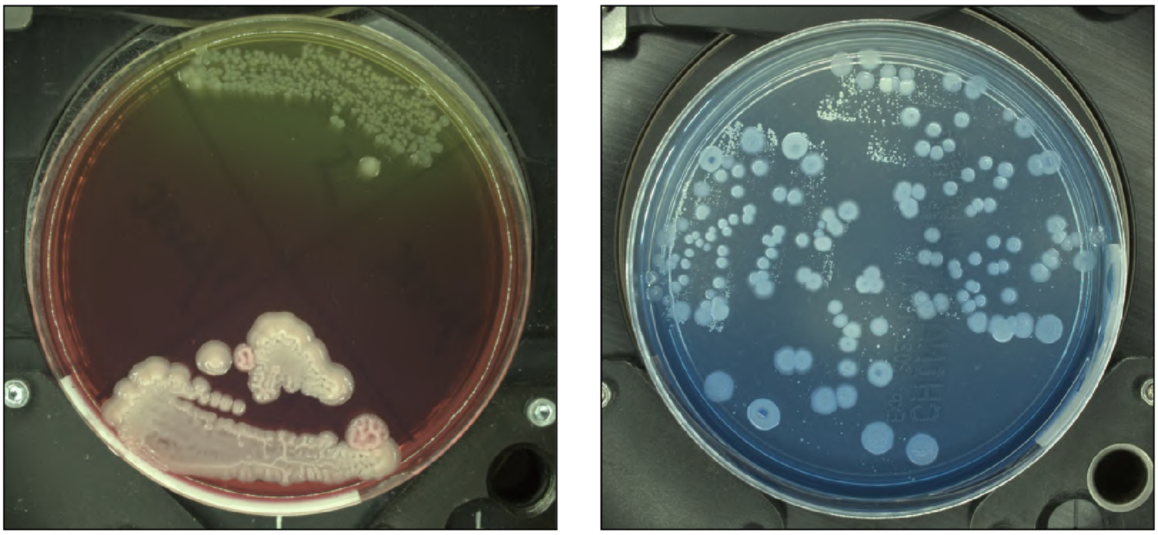
The issue can be partially mitigated by simply blurring the edges of all the colonies. This is however a destructive approach severely altering the original pixel composition. Blurring itself also does not remove the original agar's tint, which can still be prevalent throughout the colony. Since it destroys the information without mitigating the original problem, it can be ascertained that blurring is not a useful tool in this matter.

Lowering opacity around the edges of the colony is a comparably less destructive process, however it is not targeted at all and can lead to a decrease in relevant information being transferred. Some colonies do have sharp edges not influenced by the underlying agar color. Some colonies are also visibly transparent throughout, even in their center. Lowering the opacity throughout the entire colony however leads to very faint and faded results in comparison with real images.

Another problem is a phenomenon often observed when dealing with cultivated agar plates, which is the agar alteration near some of the colonies. This can be attributed to a biological process called haemolysis [10] when dealing with blood-based agars, though faded or otherwise visibly altered areas have been observed even on other types of agars as well. An example of both can be seen in Figure 3.3. Haemolysis is the rupturing of blood cells spilling out their content (haemo = “blood”, -lysis = “loosening”), which on a blood-based agar can look like a wide-spread gradient discoloration.

A related phenomenon occurs near some types of colonies, which is often described as a “halo”. Halos usually tightly surround a single colony's border and appear to be a part of it. The reason it is now being described as its own phenomenon is that it is sometimes observed on agars which do not contain any blood, so it in all likelihood is not haemolysis.

These phenomena are often a prominent part of agar plates and the segmentation algorithm should be exposed to them sufficiently during the learning phase. They are also important in regards to determining the species of a colony.



(a) Haemolysis in a blood-based agar

(b) Cultivated CBL agar

Figure 3.3: An example of agar being altered near some of the colonies. In its top part the image in (a) shows the effect of haemolysis on a blood-based agar. The image in (b) shows a varying degree of saturation in the China Blue Lactose agar in various parts of the dish. In this case it is unlikely to be haemolysis, as this agar does not contain blood. It serves to demonstrate that a degree of agar alteration can possibly be expected on agars of other composition too.

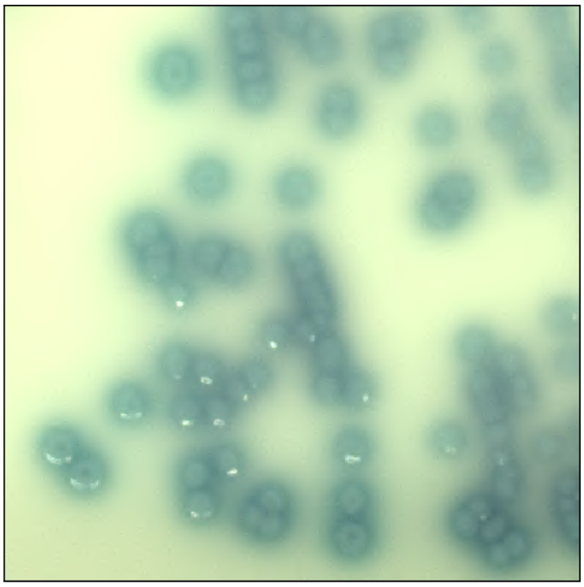


Figure 3.4: An example of "halos" forming near unknown colonies on an unknown agar.

3.2 Proposed solution

Taking into consideration all of the limitations described in the previous section, a general workflow can be defined:

1. Acquire diverse empty agar images.

2. Scrape isolated colony images from the original dataset and pair them with their binary mask counterparts.
3. Convert the resulting cutouts to their semi-transparent versions minimizing information loss.
4. Deploy the colonies over the dishes without having them overlap.
5. Simulate the agar reaction phenomena.

The basic idea of this workflow can be seen in Figure 3.5.

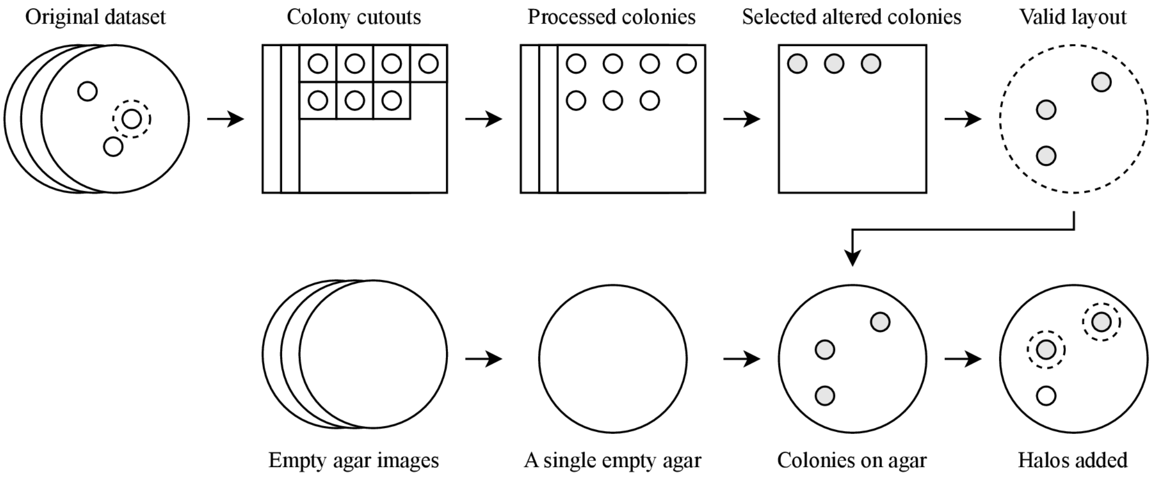


Figure 3.5: A sequence of steps taken towards acquiring artificially generated samples.

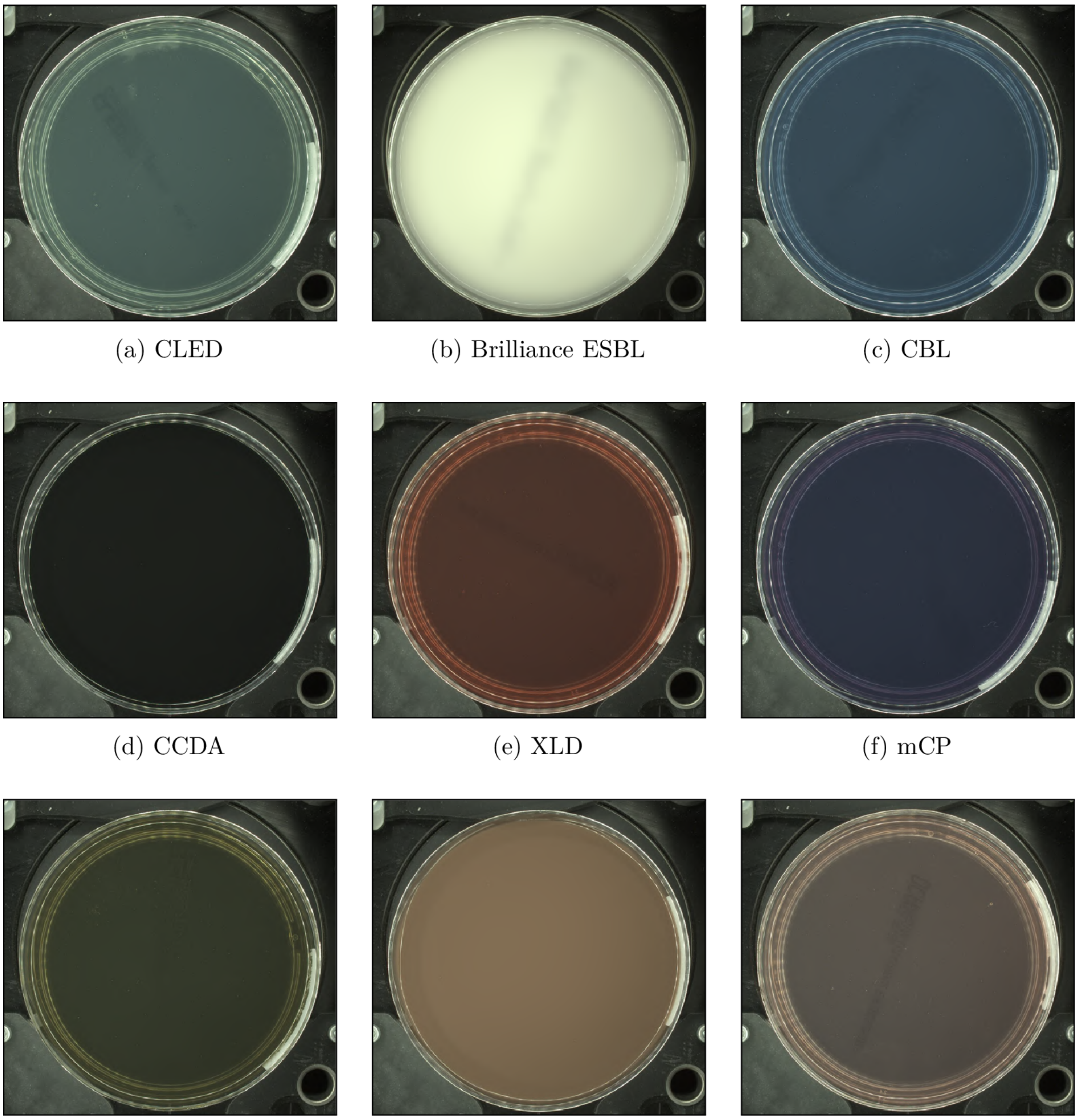
Data of empty agar plates that are to be used as base images for the image generator are not publicly available. I thus decided to create my own dataset. For the purposes of this thesis, 230 agar plates, 10 plates of each agar type were obtained in cooperation with the company Bruker Daltonics GmbH & Co. KG . The emphasis was on covering the broadest possible agar variety.

I then used the MBT Pathfinder prototype for acquiring the images of all the plates under the six predefined illumination modes. 10 plates have been damaged during storage, but in the end there is still 220 empty plates available. Images of these plates have been taken under the six available lighting conditions, resulting in 1 320 available images. An example of several empty plates can be seen in Figure 3.6. A summary of all the agar plates obtained can be seen in Table A.1.

3.2.1 Colony scraping

Based on the available hand-made binary masks of the original dataset, I have split each image into individual segments by isolating continuous shapes in the binary mask. The resulting cutouts have been paired with their corresponding masks of the same dimensions, as demonstrated by Figure 3.7. In total, I have extracted 21 144 colony-mask pairs from the original dataset.

Since the microbial species have not been labeled, I have manually categorized the colonies scraped from the available dataset's images based on their perceived visual similarity. A total of 40 distinct collections of colony-mask pairs have been prepared, with



(a) CLED

(b) Brilliance ESBL

(c) CBL

(d) CCDA

(e) XLD

(f) mCP

(g) HE

(h) Chocolate with vitox

(i) DCA

Figure 3.6: An example of several of the acquired empty agars. The aforementioned color variance can be noted, along with the varying degree of transparency. The full list of acquired agars can be seen in Table A.1. The images displayed have not necessarily been taken under the same lighting mode, as for example the creamy white agars reflect a lot of the light back when illuminated primarily from above.

volumes of images ranging from 5 to 7 293. Some of these collections contain the inoculation spread streaks. For simplicity, I will from now on refer to these visually distinct collections as species.

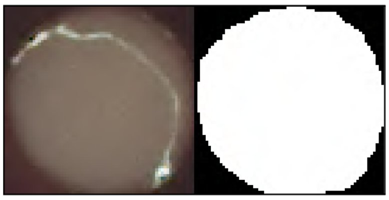


Figure 3.7: An example of an image containing a single colony cutout along with its binary mask. The white line cutting through the colony is a gleam caused by the illumination lights. A total of 21 144 pairs such as this have been created during the scraping phase.

3.2.2 Pre-processing the colonies

The most viable solution to the issue of removing agar tint from the colonies and smoothly softening their edges without losing information seems to be chroma keying.

Chroma keying is a technique used heavily by the film industry in the last century. A sufficiently contrasting color can be used as a background of any image and removed in post-production to than superimpose said image over a different background. While in movies this process is most often used on solid objects, which makes things simpler, it can be used with semi-transparent objects as well.

The colonies on agar plates pose a problem here, however, since nobody was deliberately choosing an agar color contrasting with the colonies. On the contrary – the colonies themselves can often be colorless naturally, and if they do display significant coloring, it can be caused by the underlying agar. To address this issue properly, I proposed the following set of rough rules:

- If a color is completely different to that of the agar, it will remain unchanged.
- If a color is similar to that of the agar, the RGB components will be balanced in such a way that the resulting color is more neutral depending on the degree of similarity.
- Depending on how much each pixel was manipulated in this manner, its opacity will be lowered accordingly.

This approach should lead to semi-transparent images of colonies which retain their unique colors completely, do not possess any more agar tint, and in places where they did previously display signs of agar tint, they are transparent enough to let the color of the new agar prevail.

To apply these rules, first the color of the surrounding agar must be obtained. Given that the agar's color can vary in different areas of the dish, a convenient way of obtaining the background color near the colony in question is just taking the single colony and mask image pair and averaging the RGB channels of all the image's pixels not marked as colony within the mask.

Once that is done, the similarity between the color of each pixel within the colony and the background is calculated as follows:

$$s = 255 - |(r_p - r_B)| + |(g_p - g_B)| + |(b_p - b_B)| \tag{3.1}$$

$$\Delta\alpha = \alpha - s \tag{3.2}$$

In these equations s denotes an arbitrary degree of similarity between two colors, r_p, g_p and b_p denote the RGB channels of the currently compared pixel respectively, r_B, g_B and

b_B denote the average value of the RGB channels of the background respectively, α denotes the alpha channel of the currently compared pixel and $\Delta\alpha$ denotes the altered alpha channel of the currently compared pixel.

This similarity s obtained in Equation 3.1 is then subtracted from the alpha channel a of the compared pixel, as seen in Equation 3.2. Note that while this equation can yield different results, the new value of the alpha channel a_n must be of values in range 0 – 255 and must therefore be trimmed, if necessary.

The opacity has now been reduced accordingly. A similar approach to keying out the background has been used in an article also dealing with generating synthetic agar plates [16]. I propose an additional improvement in attempting to also remove the background's color influence from the pixels that were faded but still remain possibly off-colored. The background color tint is faded in a similar manner. The following equation describes how a single RGB channel value is adjusted according to its similarity to the background:

$$\Delta c = c - \frac{r_B + g_B + b_B}{3} - c_B \cdot (1 - \frac{\alpha}{255}) \tag{3.3}$$

This equation is applied on every RGB channel of every processed pixel. Symbols r_B, g_B and b_B denote the RGB channels of the background color respectively, c_B denotes the value of the currently addressed RGB channel of the background, c denotes the unmodified value of the same RGB channel of the currently processed pixel, Δc denotes the new value and α denotes the currently processed pixel's alpha channel's value.

The basic idea behind this equation can be formulated as follows: How much and in which directions should the RGB channels of the background be shifted to achieve a gray color of roughly the same brightness? Such a color can be obtained by populating each of the RGB channels with the original background's channel average. To answer the question, for every color a set of channel shifts can be obtained which would result in this color. Applying these shifts to pixels within the colony to some degree now effectively fades the background color's influence. How much to apply these shifts has been calculated beforehand and is represented as the alpha value. The more a pixel is transparent, the more similar it is to the background, and therefore the more its respective channels should be shifted towards gray from the background color's perspective.

After applying these weighted alterations to all the pixels within the colony, we can see it yields comparatively good results even when used on a large variety of possible background colors, as demonstrated in Figure 3.8.

3.2.3 Colony augmentation

To maximize sample diversity, I modify the colonies superimposed on the agar plates in a variety of ways. These modifications are done during generation, so each keyed colony is stored paired with its segmentation mask only once, unaltered. Some modifications are done for each of the colonies separately, such as rotation, some can be done en masse for species, such as hue shifting.

Other modifications done are saturation shifting, value shifting, mirroring and scaling. Saturation and value shifting is done within ranges set by the colony image itself so that no overflow happens in any of the manipulated pixels. The scaling factor range has been set to 90 – 135 % of the original size. Rotation is done to a resolution of a single degree.

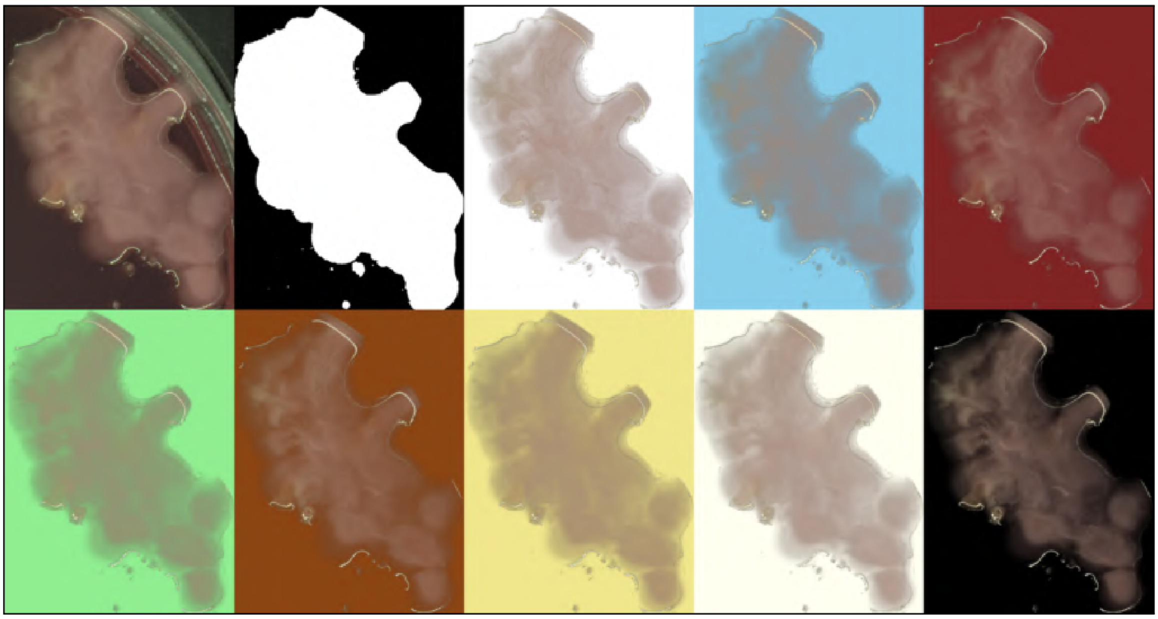


Figure 3.8: A demonstration of the chroma keying technique on a particularly large lump of biological material found on one of the dishes of the original dataset. The original cutout along with the corresponding binary mask can be seen in the top left. The other examples are this cutout keyed and superimposed over solid color backgrounds. It can be noted that the rough edges seen in the original superimposing attempt in Figure 3.2 can no longer be seen. The light reflections are left intact and the lump has mostly preserved its original color. At the same time, the color of the new background can clearly be seen prevailing in the less opaque regions of the lump with little to no traces of the original dark maroon agar.

3.2.4 Colony positioning

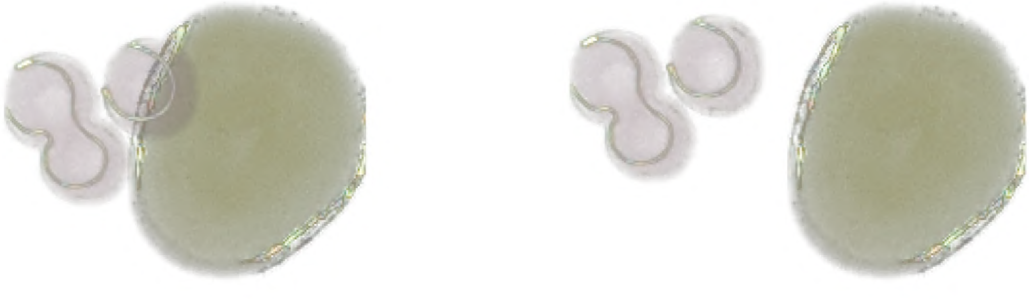
Having both the plates and the colonies prepared, a reliable way of superimposing them onto one another has to be created. The growth on the dish often covers more than a half of its total surface. At the same time, the keyed colonies are not fit to be placed over each other, as it would look unnatural.

I have decided to use a genetic algorithm to aid with the task of distributing the colonies in large numbers over the dish in such a way that they do not overlap. Genetic algorithms apply the process of natural selection to a population of solutions to a problem, given that it is possible to evaluate a single solution in terms of correctness. Such an evaluation is done by a fitness function [12].

The fitness function is very simple yet effective. Given a solution, which means all colonies have been in some way placed upon the dish, it simply counts all the pixels of all the colonies which do not overlap with colonies' pixels and lay within the dish area:

$$f = \frac{p_u}{p} \tag{3.4}$$

In this equation f denotes the resulting fitness, p_u denotes the sum of all of the the colonies' pixels which do not collide with another colonies' pixels and which lay within the dish are and p denotes the sum of all of the the colonies' pixels. The fitness function is visually demonstrated in 3.9.



(a) f = 0.93

(b) f = 1.0

Figure 3.9: To deploy the colonies within the plate there has to be no overlap. Any overlapping pixels as seen in (a) will therefore lead to a lower fitness function evaluation.

The problem itself is defined by a circular area in the shape of a dish and a list of colonies to be placed within said area. The total area of these colonies must be less then the area of the dish. It was however empirically discovered that it is already very hard to find solutions for lists of colonies with an area larger than 60 % of the dish, since many of the colony images take shape of complex smears.

In the beginning of a single generation run, a population of solutions is created, each of them evaluated by the fitness function. A solution is represented by a list of coordinates (integer pairs). Each pair signifies a position[1] of one colony on the dish. These integer pairs are in the context of this algorithm referred to as “genes”. All the positions are initially defined randomly, although in ranges defined by the dish dimensions. Should any solution be evaluated by the fitness function with the result of 1.0, the computation ends as the perfect solution has been found. Otherwise, a number of solutions is selected to enter the “mating pool”. I have opted for the easy to understand and predictable Steady-State Selection type, which means the parents are selected randomly [12], and set the number of parents to mate to 6.

Once these candidates enter the mating pool, they are paired. For each of these pairs, another pair of offsprings is created by combining the genes of their parents. This operation is called a “crossover”. Here I have decided to use a uniform approach, so for each of the genes there is a 50% chance it is inherited from either parent, as demonstrated in Figure 3.10.

After the crossover phase, a “mutation” can occur for each of the genes of every newly generated candidate. Here I have opted for a random type of mutation and set the probability of it happening to 4 %, as both notably lower or higher values have lead to a slower convergence during development. A random mutation means that a completely new pair of coordinates will be generated within the allowed range should a gene mutate. After mutation, all of the offspring is evaluated.

Then, since I have selected Steady-State Selection, the offspring are along with their parents all gradually compared with against the worst candidate within the original population in terms of fitness. Every time a candidate within the mating pool is found to be

[1] To the fitness function it does not matter whether these are the coordinates of the top left pixel of the colony or its center. Any colony which reaches beyond the borders of the dish results in a lower fitness.

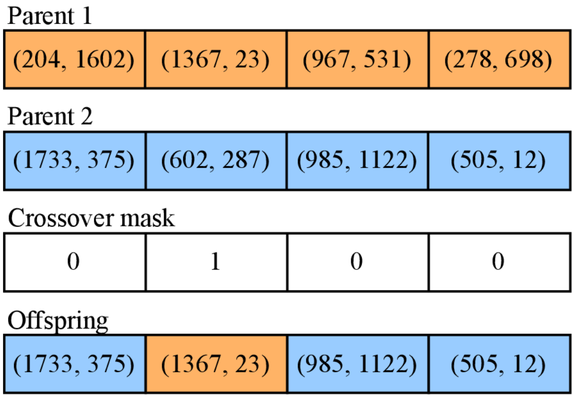


Figure 3.10: The crossover operation producing new solutions. A randomly generated binary crossover mask determines which gene is inherited from which parent.

more fit then the worst candidate in the original population, the worst candidate is replaced by the candidate from the mating pool. This ensures the population number is stable, hence the name Steady-State Selection [12].

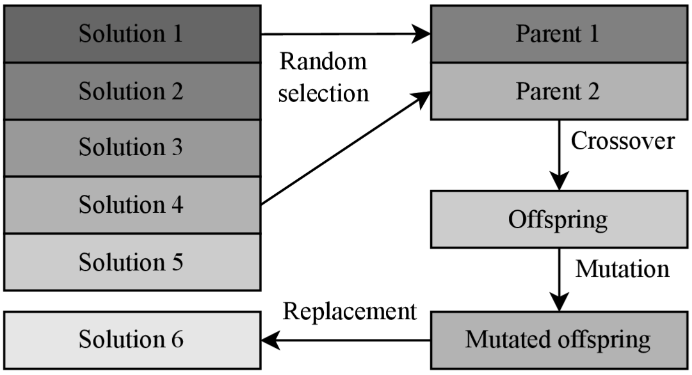


Figure 3.11: A visualisation of the genetic algorithm's cycle. Parents are randomly selected from the population, offspring is generated, mutated, and if better than the worst solution in the population, it replaces it. Otherwise it is discarded. The parents are always kept in the population.

This whole process is then repeated until a flawless solution is found. Depending on the number and nature of colonies selected to be placed, the algorithm can fail to find a fitting solution within the predefined amount of generations. It has been found empirically that letting the algorithm run for extended periods of time after it hit the 500 generations only very rarely led to actually finding a result. Any run which does not find a solution within 500 generations, which usually happens within a single minute [2], is therefore assumed to be stuck in a local maximum and cancelled and a new run is initiated with a different batch of colonies and a new initial population.

[2] The resulting script was ran on a machine with an NVIDIA Quadro T2000 GPU and an Intel Core i7-10850H CPU

3.2.5 Superimposing the colonies

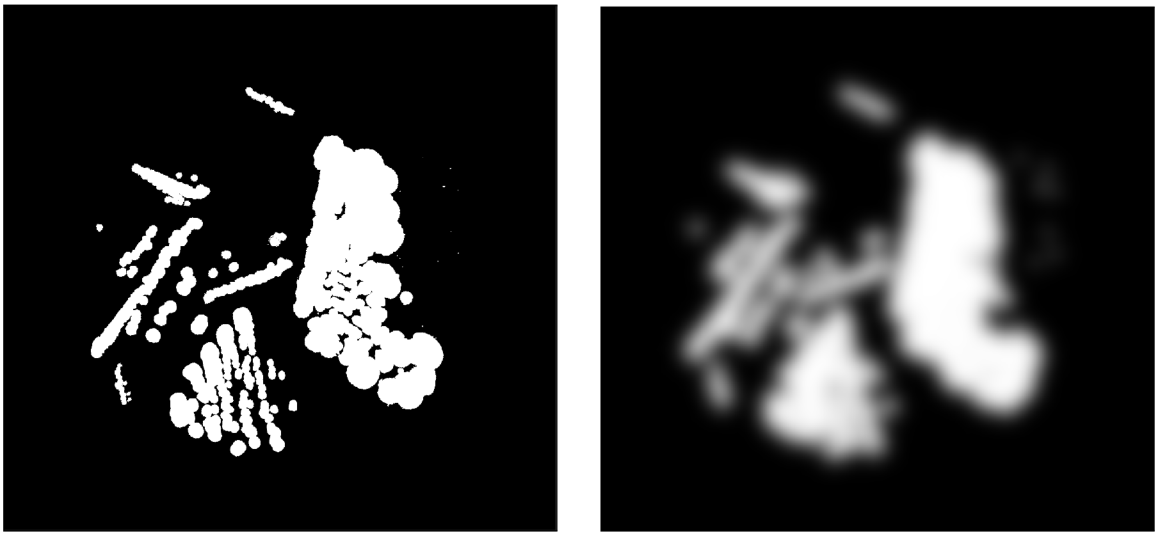
The semi-transparent colonies in the RGBA color format have now been distributed over the dish without collisions. Now they have to be superimposed upon the background image of an empty dish, which is in the RGB format. Once again, the resulting value of each pixel's RGB channel has to be weighed by the colony's alpha channel to determine its influence on the modified pixel:

$$\Delta c_B = c_B \cdot (1 - \alpha) + c_o \cdot \alpha \tag{3.5}$$

In this equation Δc_B denotes the value of a channel of the background pixel after being modified, c_B denotes the value of a channel of the background pixel prior to being modified, α denotes the value of an alpha channel of the overlay pixel and c_o denotes the value of a channel of the overlay pixel. The overlaying colonies now blend with the background naturally, as can be seen in Figure 3.8.

3.2.6 Simulating agar reaction

To add another level of accuracy, both the previously mentioned "halos" and the presumed haemolysis are simulated. The segmentation masks of colonies set to have halos or damage the agar by developing haemolysis are initially dilated and blurred, as demonstrated in Figure 3.12. The degree of dilation affects the reach. The resulting mask is then used to determine the degree of pixel alteration at all the affected points in the empty plate image.



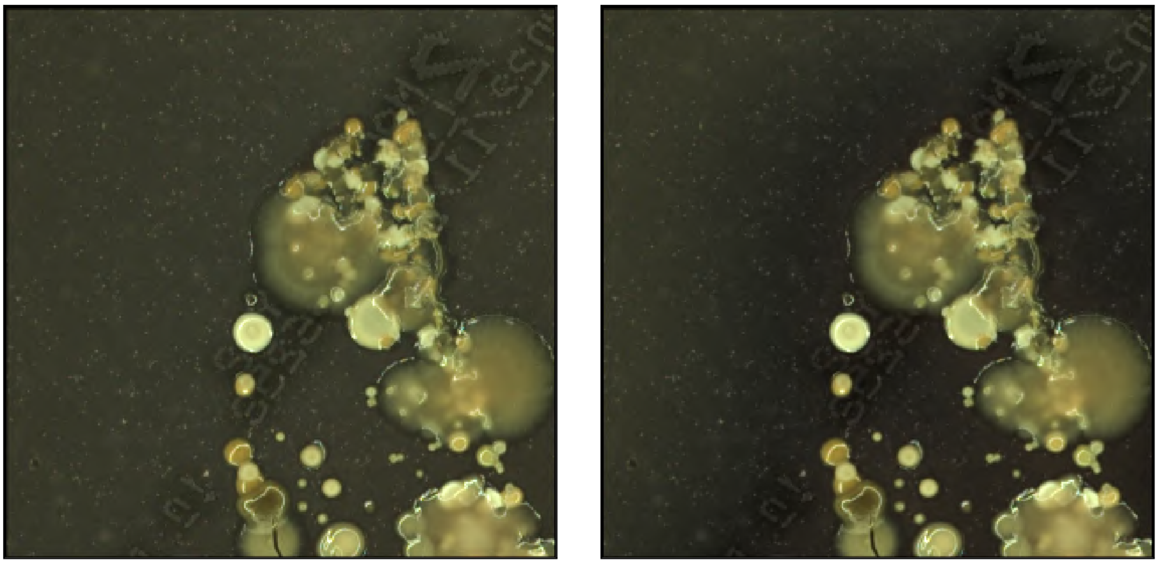
(a) Segmentation mask

(b) Dilated and blurred mask

Figure 3.12: To model the various kinds of agar reaction, a segmentation mask as seen in (a) indicating only colonies which have been set to develop one of the described phenomena is dilated and blurred. The resulting mask, as seen in (b) is used as a weight map to determine the degree of pixel alteration in various places of the dish.

When creating a halo, its color is defined beforehand, after that it is superimposed on the empty agar dish image using the same Equation 3.5 as with colonies. Only difference being the alpha value is now taken from the blurred mask to simulate the fading nature of the simulated phenomena.

The presumed haemolysis is simulated by lowering the Value and Saturation of the affected pixels in the HSV color model. The blurred binary mask serves the same purpose as with halos and sets the degree of alteration. An example of the effect of haemolysis being applied can be seen in Figure 3.13.



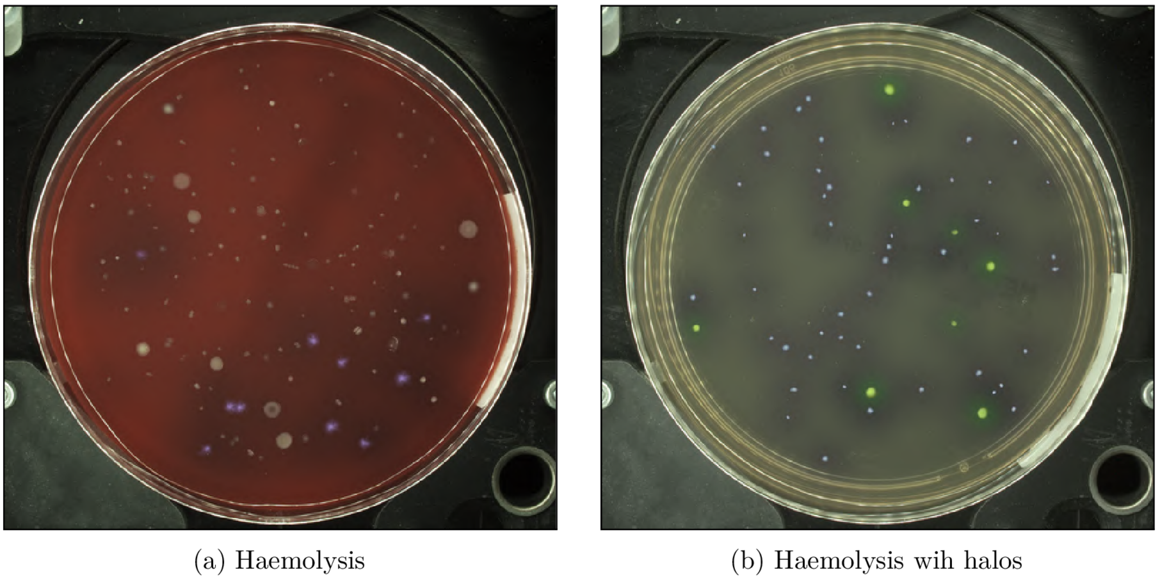
(a) No haemolysis

(b) Haemolysis

Figure 3.13: A detail of a single synthetic agar plate image. In (a), a cluster of colonies has been keyed and superimposed on the agar. In (b), the haemolysis effect has been applied according to the weight map created by dilating and blurring the segmentation mask of preselected colony species. Even though the MacConkey agar used in this example does not contain blood and is therefore not able to develop haemolysis, the ability to simulate this phenomenon on any dish is still useful and will aid the generalization of the deep learning model.

Changing the Hue was originally implemented as well, it turned out however that it is only limited to 179 distinguishable values, since the angular value wheel for hue must fit into a single `byte` in most Python libraries. This creates undesirable contours and leads to a terrace–like phenomenon when creating the smooth agar damage gradient, as the difference of 2° on a Hue spectrum is noticeable even to a human eye. A machine learning algorithm could easily learn to react to these edges, which would undesirably lead to overfitting.

It is also important with these alterations to watch out for any overflowing values. How much the Value and Saturation can be lowered throughout the picture must be checked beforehand. Fixing overflows simply by thresholding the values at zero could lead to visible edges. To prevent this, the lowest Saturation and Value values are found beforehand among all of the potentially affected pixels. These values then define the maximum degree of agar alteration.



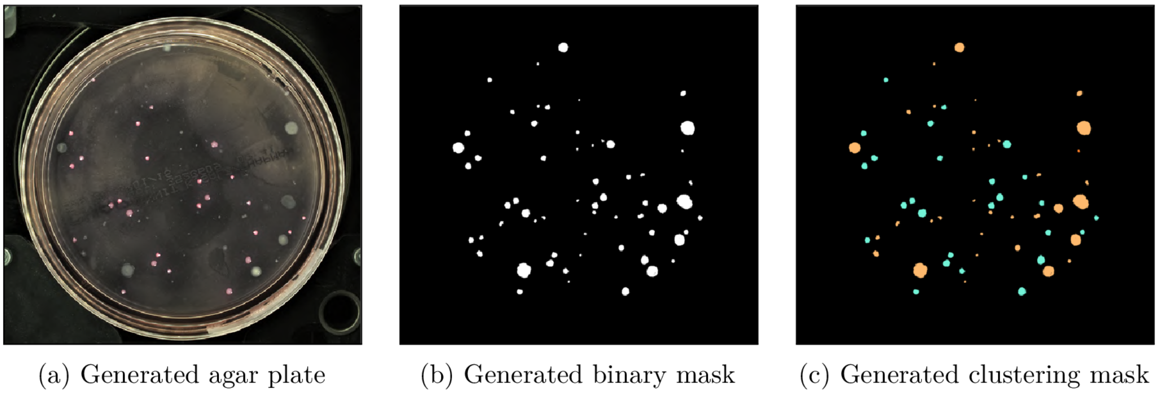
(a) Haemolysis

(b) Haemolysis wih halos

Figure 3.14: An example of the simulated agar alterations caused by the microbes and agar interacting. The plate in (a) has been generated to display signs of haemolysis around the species of purple colonies. The plate in (b) also contains faded parts and several colonies with prominent greenish "halos".

3.3 Agar plate generator

The approaches described in the previous sections were meant to result in realistic data samples capable of substituting real data in both segmentation and clustering attempts. The produced generating script was therefore designed in such a way, so that it not only produces the images paired with segmentation masks, but also with corresponding clustering masks, where each class is represented by a single arbitrary color, as demonstrated in Figure 3.15.



(a) Generated agar plate

(b) Generated binary mask

(c) Generated clustering mask

Figure 3.15: An example of the generator's output. Aside of the segmentation being generated, a clustering mask is generated as well labeling the colonies according to their shared visual properties and presumed species. The colors are arbitrary, each uniquely assigned to a single species.

The ability to specify the numbers and ratios of species was also implemented. The resulting `dish_generator.py` script's available parameters go as follows:

- `agar_directory` – Containing empty agar plate images
- `species_directory` – Containing subdirectories of keyed distinct colony-mask pairs
- `--cult_min` – Minimal number of species generated on a dish `[1-MAX]`
- `--cult_max` – Maximal number of species generated on a dish `[1-MAX]`
- `--halo_prob` – Probability that a single species develops a halo `[0-100]`
- `--haemo_prob` – Probability that a single species causes haemolysis `[0-100]`
- `--area_min` – Smallest possible percentage of a dish area to be covered `[0-100]`
- `--area_max` – Largest possible percentage of a dish area to be covered `[0-100]`
- `--balanced_prob` – Probability, that species will be represented in equal quantities (in percent) `[0-100]`
- `--iterations` – Number of GA iterations before discarding a dish and attempting a new layout `[1-X]`
- `--edit_hsv` – Creates more distinct species by manipulating their HSV channels en masse

For each dish attempted, the ratios and parameters are regenerated within the specified ranges. This way the script can generate datasets according to the current needs. Balanced dishes with many species are good for the purposes of teaching a model to generalize when training for segmentation, dishes with less species where some can be only present as inconspicuous outliers can be used to fine tune the upcoming clustering algorithms.

Since the colonies have to be keyed only once, the keying itself is not a part of the `dish_generator.py` script and has been implemented separately. The generator therefore works over a directory structure containing already keyed colony-mask pairs sorted according their presumed species. Each species is during generation assigned a unique color by which it is then marked on the output clustering mask pictured in Figure 3.15c.

Depending on the type of dishes desired, the generator script can produce fairly consistent, realistic looking and diverse results. An example of some of these can be seen in Figure 3.16.

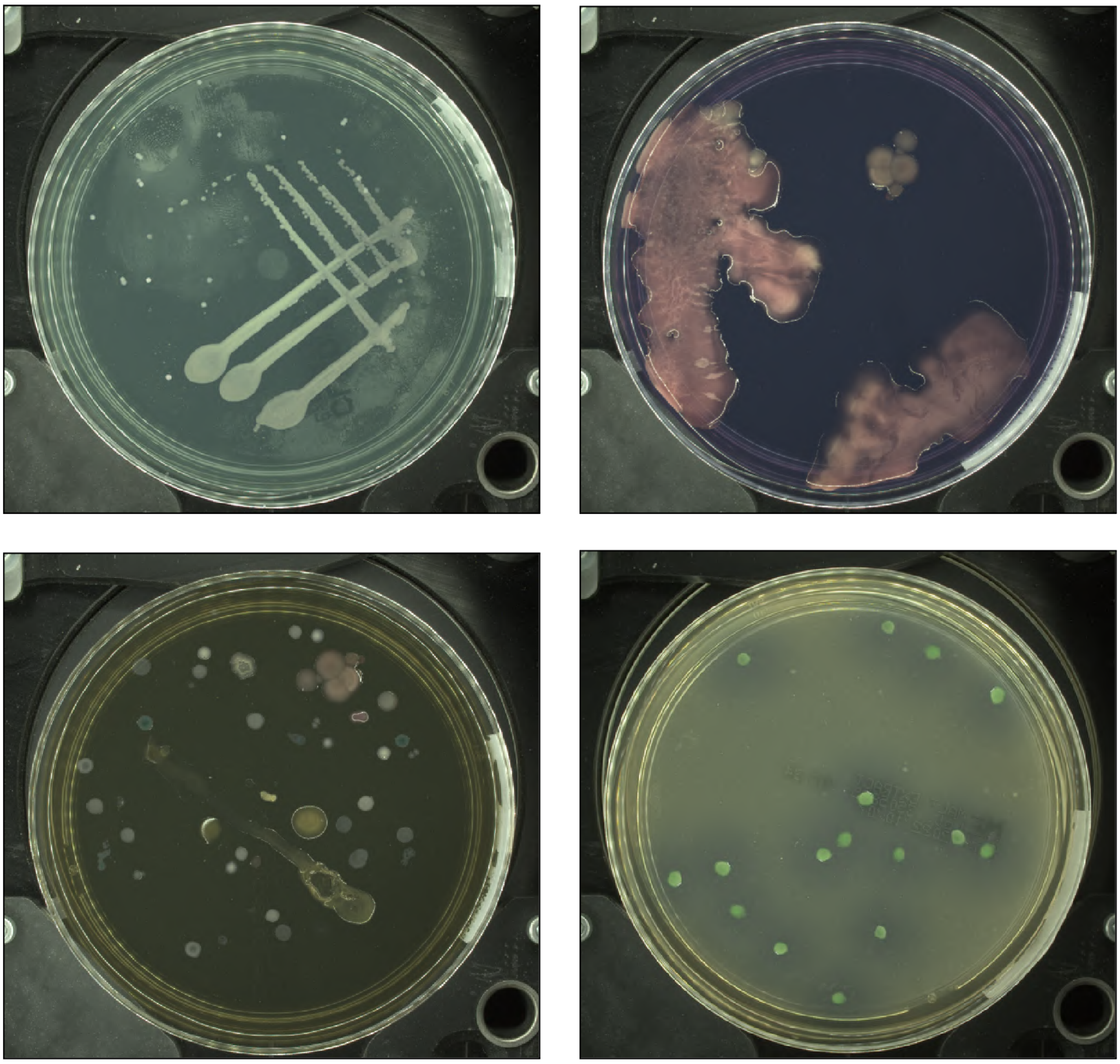


Figure 3.16: An example of several cherry-picked artificially generated plates demonstrating the generator's versatility.

Chapter 4

Segmentation using synthetic data

To evaluate the synthetic data for the purposes of data augmentation, the most direct method is to use it to train segmentation models and see if they improve when compared to the original one. It is also useful to evaluate the effectiveness of the individual realism improvements the generator offers, such as the background keying and the agar reaction simulation, as opposed to just a simple cutout placement.

The performance of segmentation models can be evaluated by several metrics. I decided to use the F1 score (also known as Sørensen–Dice coefficient or Dice similarity coefficient) along with the Jaccard index (Intersection over Union), since these metrics are frequently used to assess similar tasks. I also included Precision, Recall and Accuracy for the sake of completeness.

To explain these metrics, the terms “true positives”, “true negatives”, “false positives”, and “false negatives” (TP, TN, FP, FN) need to be defined. The terms “positive” and “negative” correspond to the actually retrieved result. The terms “true” and “false” express the results’ relationship with the ground truth. Using these terms the metrics can be defined as follows:

$$\mathit{Precision} = \frac{TP}{TP + FP} \tag{4.1}$$

$$\mathit{Recall} = \frac{TP}{TP + FN} \tag{4.2}$$

$$\mathit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

The F1 score is a known and widely used metric and the Jaccard index is often used to evaluate the effectiveness of 2D segmentation. An F1 score combines the precision and recall metrics:

$$\mathit{F1} = 2 \cdot \frac{\mathit{Precision} \cdot \mathit{Recall}}{\mathit{Precision} + \mathit{Recall}} \tag{4.4}$$

The Jaccard index is simply calculated as an intersection over union as demonstrated by Figure 4.1.

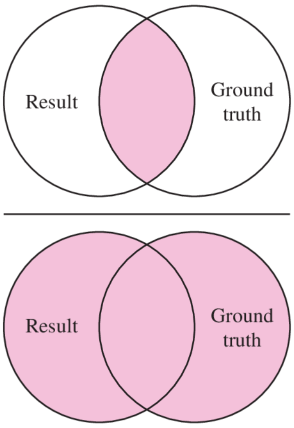


Figure 4.1: Intersection over union – also called the Jaccard index. The Ground truth set is all the pixels indicated as positive in the ground truth mask. The Result set is all the pixels indicated as positive in the output mask of our segmentation models.

4.1 Acquiring the datasets

The models to be trained on the partially synthetic data should respond better to images of agars other than white and red. The issue is the BRUKERCOLONY dataset contains no such agars and using synthetic data to validate the usage of synthetic data does not make sense. For a more general evaluation, I have decided to expand the testing subset of the BRUKERCOLONY dataset as well.

For this purpose I chose to use the previously obtained 240 empty agar plates. I inoculated all the plates with dust samples taken from various indoor surfaces and left the plates in room temperature for 48 hours.

Many of the plates have developed microbial growth. A large portion was overrun by mold, which often occurs in samples of microbial mixtures, and some agar kinds have proven completely resilient to everything. Despite that, a number of these plates had sufficient microbial growth to serve as real data samples. I have manually selected the successfully cultivated dishes, I sorted out those which also contained large masses of mold, and I have scanned all of these dishes in the MBT Pathfinder prototype. Examples of several of these dishes can be seen in Figure 4.2.

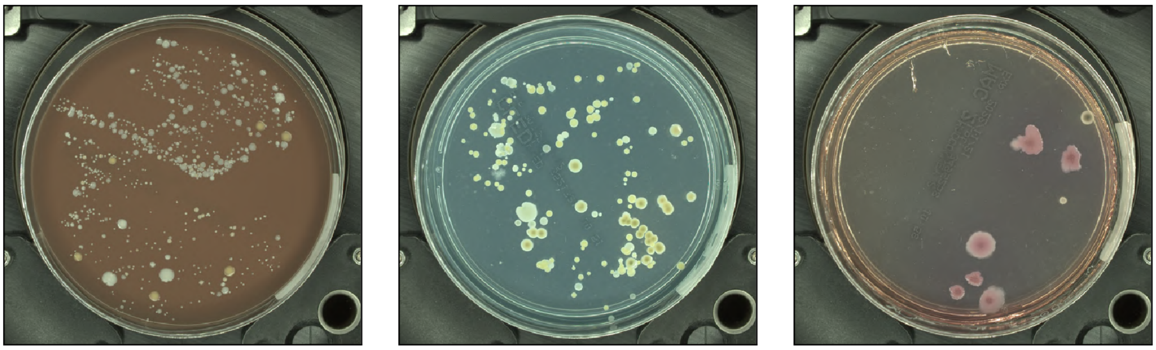


Figure 4.2: Several samples from the newly acquired real dishes. Which species are present on each of the dishes has not been determined. It is however important to note the agar color variance in comparison to the original BRUKERCOLONY dataset as seen in Figure 3.1.

For 43 of these images I have manually created labels in the form of corresponding segmentation masks. Since the synthetic data features these agar kinds, the models should overall respond better to real images of these agar plates.

To obtain training data, I have then utilized the generator to produce four identical datasets of 1 000 samples each. They differ solely in the use of keying and simulating agar reactions. The colony layouts, used agars and colorings are otherwise identical. This can be seen demonstrated in Figure 4.3. Along with the images themselves the corresponding binary segmentation masks were also generated, identical for all four variations. The generator's parameters were set as follows:

- 40 species,
- 25 % halo probability,
- 33 % haemolysis probability,
- 5-50 % of area to be covered by growth,
- 100 % probability of a balanced dish,
- HSV editing enabled.

To cover the broadest possible range of species/agar combinations, all generated dishes utilize all 40 of the distinctly looking microbes I have managed to isolate and sort. Both halos and haemolysis can develop near each species. Dishes are set to be populated both sparsely and densely. To ensure an equal representation of all the species, all dishes were set to be balanced. Even though it will possibly lead to unrealistically looking colonies, HSV manipulation was enabled. In my opinion this will help with the model's generalization, even if it learns to recognize colonies which in the real world are non-existent. The dishes themselves do not look realistic either. As the segmentation models will however processes the dish in parts, the broader context of the dish will not be as relevant.

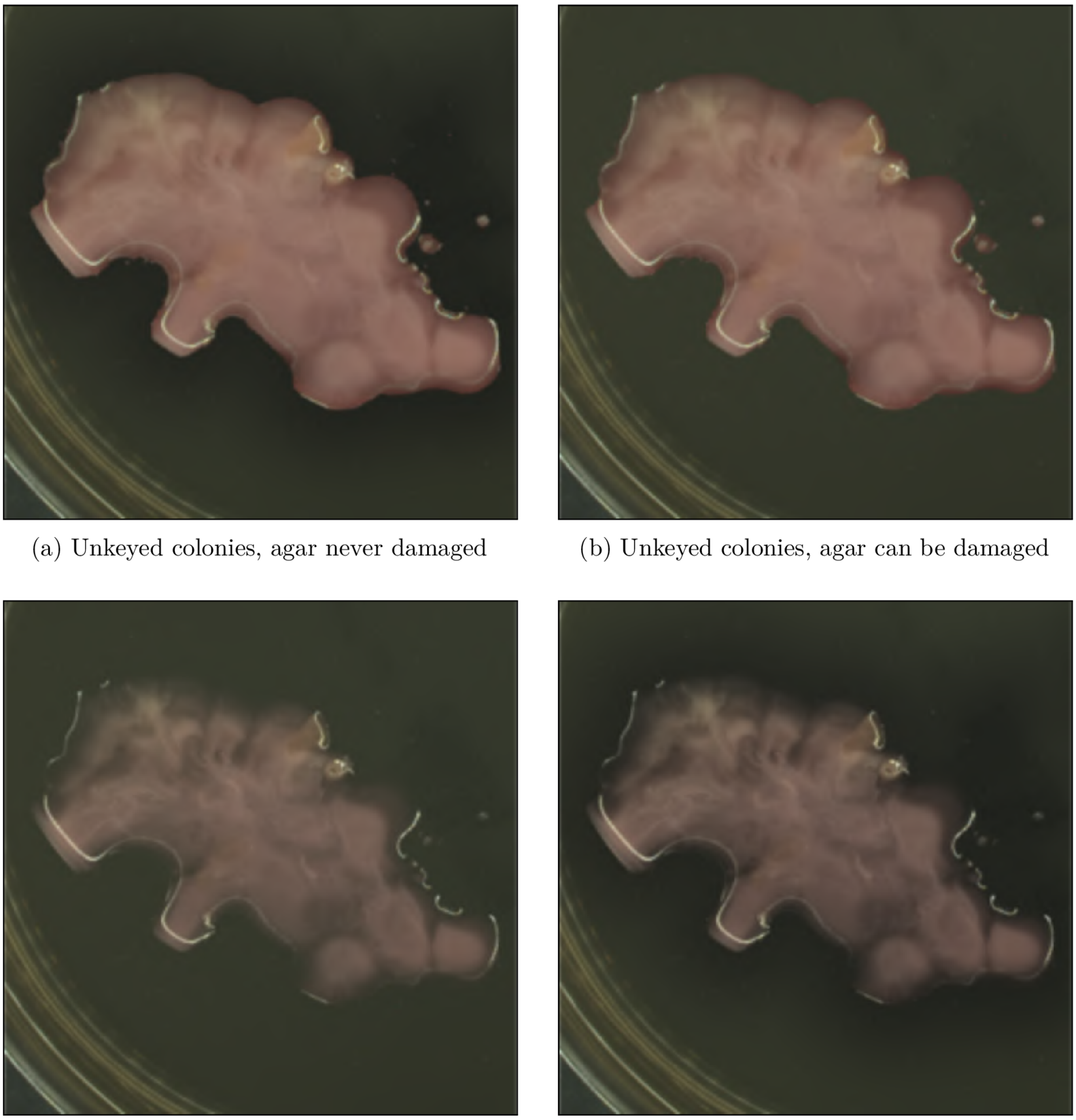
This data is to be used as an extension of the BRUKERCOLONY dataset to test its effectiveness when confronted with previously previously unseen real data.

4.2 Evaluating the synthetic data

Combining the available BRUKERCOLONY data, the newly generated synthetic data and the newly acquired and labeled real data, I have assembled five datasets for the segmentation experiment. The results will make it clear whether the synthetic dataset is a valid substitute for real data and if all the proposed features are relevant.

To eliminate the influence of MBT Pathfinder's model architecture, I have trained five slightly modified[1] U-Net segmentation models [17] on the four alternative partially synthetic datasets and the original BRUKERCOLONY. The architecture does not use residual layers. Batch normalization is utilized. Each sample was split into 16 images of size 512 to 512 pixels to fit more images into the training minibatches. Early stopping was employed with the patience of 10 epochs and an exponential decrease of the learning rate starting from the 15th epoch. A diagram of the experiment can be seen in 4.4. A summary of volumes of the final datasets on which the experiment was conducted can be seen in Table 4.1.

[1] Architecture taken from: https://github.com/pietz/unet-keras



(a) Unkeyed colonies, agar never damaged

(b) Unkeyed colonies, agar can be damaged

(c) Keyed colonies, agar never damaged

(d) Keyed colonies, agar can be damaged

Figure 4.3: A detail of a single dish generated in four different ways as part of four different datasets. The first two images show excerpt from datasets which do not use keying, the first and third do not simulate agar reaction phenomena. All four combinations of these options are therefore covered.

The results of the applicability experiment can be seen in Table 4.2. The first and most important thing to be noted is the fact that every single one of the models trained on the extended datasets achieved better results in every metric (aside of recall) than the one trained purely on the original BRUKERCOLONY dataset. A lower recall score means less actual microbial material was correctly detected, or in other words, more false negatives. An increase in all the other metrics on the other hand also means that there was a significant decrease in false detections.

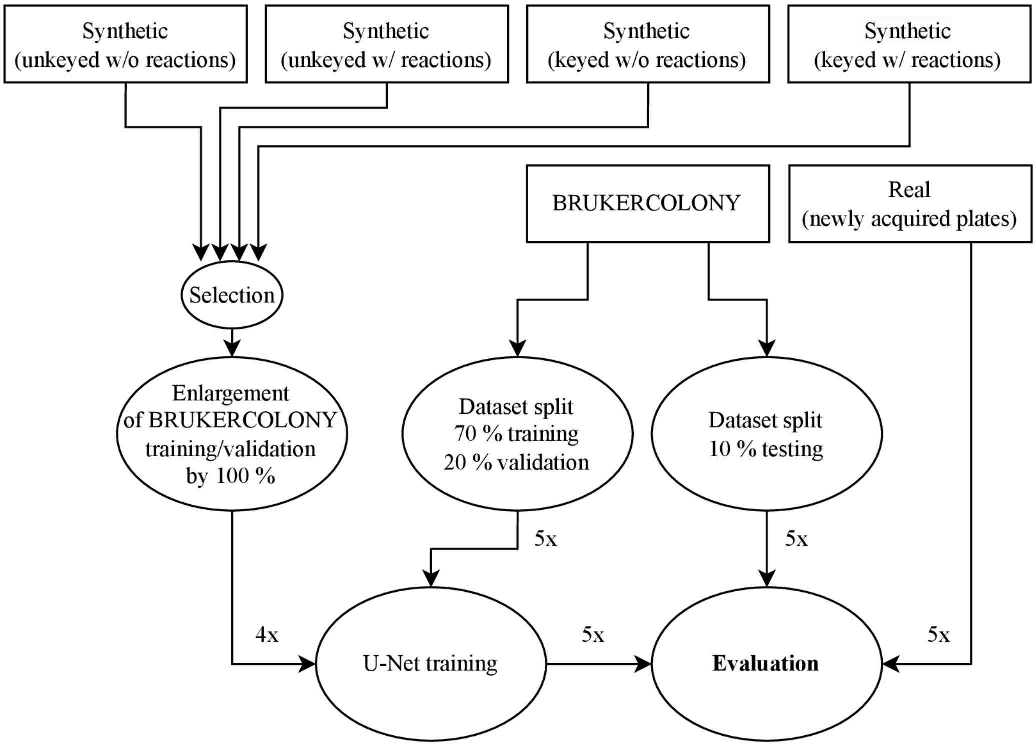

Synthetic (unkeyed w/o reactions)
Synthetic (unkeyed w/ reactions)
Synthetic (keyed w/o reactions)
Synthetic (keyed w/ reactions)
BRUKERCOLONY
Real (newly acquired plates)
Selection
Enlargement of BRUKERCOLONY training/validation by 100 %
Dataset split 70 % training 20 % validation
Dataset split 10 % testing
5x
5x
4x
U-Net training
5x
Evaluation
5x

Figure 4.4: An overall diagram of the segmentation on differently processed synthetic data experiment. One model was trained on the unmodified original BRUKERCOLONY dataset, four models were extended by synthetic data, each with slight variations, and all of the models were evaluated on BRUKERCOLONY testing data extended with some newly acquired real data featuring more unusual agars.

Table 4.1: All four datasets assembled to verify that the synthetic data helps segmentation in contrast to the original BRUKERCOLONY dataset. Four of them were extended by 100 % with identically laid-out synthetic data differing only in the used features. All the testing sets were extended by real images of newly acquired agar plates. These plates contain agars not included in the entire original BRUKERCOLONY dataset.

Dataset	train (real+synth.)	val (real+synth.)	test (real+new)
BRUKERCOLONY	383+0	114+0	51+43
BR. + Unkeyed w/o reactions	383+383	114+114	51+43
BR. + Unkeyed w/ reactions	383+383	114+114	51+43
BR. + Keyed w/o reactions	383+383	114+114	51+43
BR. + Keyed w/ reactions	383+383	114+114	51+43

It seems that introducing any synthetic data is immediately helpful to precision but detrimental to recall. However, it can also be noted that gradually introducing the realism features gradually increases the recall score to the point where it almost matches the original value. The dataset containing both the realism features has therefore a comparable recall, while still benefitting from the drastically improved precision.

When looking at the more complex metrics of F1 Score and Jaccard index, the trend is obvious. Introducing any synthetic data makes a big difference in itself, both the realism features themselves are more or less identical in their effectiveness and they work really well combined.

Table 4.2: Effectiveness of using keying, agar reaction phenomena simulation and synthetic data in general in segmentation models.

Training dataset	Accuracy	Precision	Recall	F1 Score	Jaccard index
BRUKERCOLONY	0.770	0.525	**0.781**	0.518	0.453
BR. + Unkeyed w/o reac.	0.950	0.743	0.697	0.671	0.609
BR. + Unkeyed w/ reac.	**0.958**	**0.758**	0.715	0.693	0.628
BR. + Keyed w/o reac.	0.947	0.719	0.747	0.697	0.628
BR. + Keyed w/ reac.	0.952	0.741	0.768	**0.721**	**0.648**

4.3 Improving the segmentation

I have conducted an additional experiment to determine the optimal ratio of real/synthetic data. The goal is to improve the performance of the current segmentation model used in the prototype of the MBT Pathfinder instrument. Five models of the same architecture as in the previous experiment were trained on the BRUKERCOLONY data again with synthetic data mixed in. This time all the synthetic data used both keying and simulated agar reactions. However, I have extended the BRUKERCOLONY dataset by 33 %, 66 %, 100 %, 150 % and 200 % respectively. A diagram of this experiment can be seen inFigure 4.5. A summary of volumes of the final datasets on which the experiment was conducted can be seen in Table 4.3.

Table 4.3: Effectiveness of extending the BRUKERCOLONY dataset by different volumes of synthetic data. The synthetic data is all generated keyed and utilizes agar reaction simulations. All the testing sets were extended by real images of newly acquired agar plates. These plates contain agars not included in the entire original BRUKERCOLONY dataset.

Dataset	train (real+synth.)	val (real+synth.)	test (real+new)
BRUKERCOLONY	383/0	114/0	51/43
BR. + 33 % Keyed w/ reac.	383+126	114+38	51+43
BR. + 66 % Keyed w/ reac.	383+253	114+75	51+43
BR. + 100 % Keyed w/ reac.	383+383	114+114	51+43
BR. + 150 % Keyed w/ reac.	383+575	114+171	51+43
BR. + 200 % Keyed w/ reac.	383+766	114+228	51+43

The results of the volume ratio experiment are summed up in Table 4.4. The differences in performance are less dramatic than in the previous experiment, however, a peak performance with a drop-off in both directions can still be observed near the dataset containing

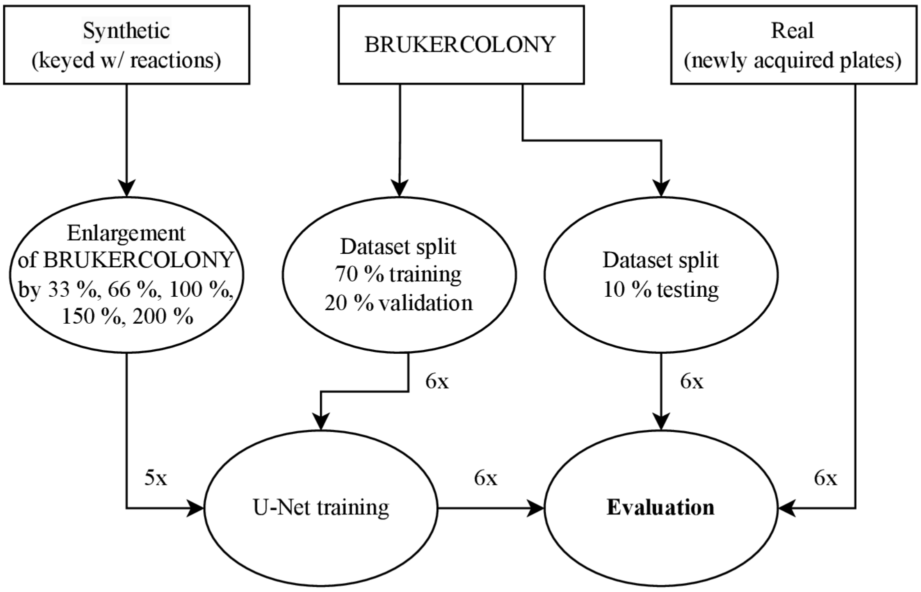


Figure 4.5: An overall diagram of the segmentation on different volumes of synthetic data experiment. One model was trained on the unmodified original BRUKERCOLONY dataset, five models were extended by synthetic data by different amounts. All of the models were evaluated on BRUKERCOLONY testing data extended with some newly acquired real data featuring more unusual agars.

150 % of synthetic data. The recall is even closer to the baseline than in the previous experiment and both the F1 and Jaccard scores have reached a better result too.

Ultimately, I have managed to improve the performance of the segmentation algorithm by 0.211 for F1 score and 0.203 for Jaccard index (a 40.73 % and 44.81 % increase respectively) by solely augmenting the dataset with artificially created images.

Table 4.4: Effectiveness of extending the dataset by different volumes of synthetic data.

Training dataset	Accuracy	Precision	Recall	F1 Score	Jaccard index
BRUKERCOLONY	0.770	0.525	0.781	0.518	0.453
BR. + 33 % Keyed w/ reac.	0.944	0.698	0.739	0.677	0.606
BR. + 66 % Keyed w/ reac.	0.952	0.743	0.768	0.719	0.647
BR. + 100 % Keyed w/ reac.	0.957	**0.748**	0.767	0.726	**0.656**
BR. + 150 % Keyed w/ reac.	0.955	**0.748**	**0.772**	**0.729**	**0.656**
BR. + 200 % Keyed w/ reac.	**0.960**	0.745	0.754	0.717	0.645

Chapter 5

Colony clustering

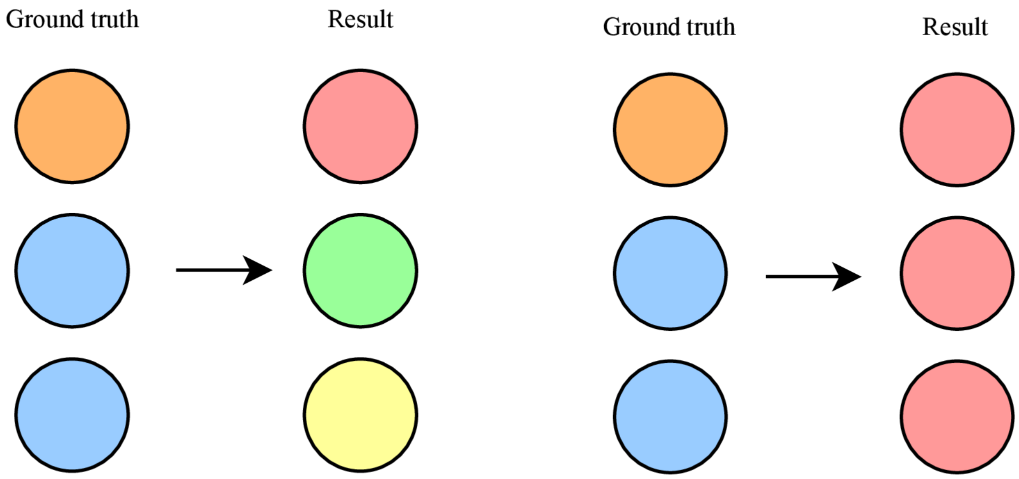
As has been stated in the previous chapters, the ability to group colonies by their visual properties is a necessity in any future solutions for automated in-laboratory analysis of cultivated agar plates. The ability to reliably sample every species on a plate automatically and without human interaction while minimizing error is a potential game changer for laboratory technicians, who spend many hours a day performing these activities manually.

In terms of evaluating the performance of any clustering attempts, it is unfortunately not clear whether it is more appropriate to penalize more a cluster count higher than necessary or lower than necessary. Since the actual cost of performing an unnecessary pick and analysis of a duplicate sample in comparison to entirely missing a species from a plate is unknown to us, I propose my own scenario by the optics of which the clustering experiments will be evaluated. I propose that the goal of the laboratory machine is to pick and analyze every single species present on a dish at least once. Therefore, redundancy is tolerated and preferable to missing (though it should still be noted and penalized in evaluation). For this reason, I will be using both the complete V-measure metric as well as its submetric, the homogeneity.

V-measure is a widely used metric for evaluating the performance of clustering algorithms [18]. Its score is calculated by balancing two submetrics – *Homogeneity* and *Completeness*. Homogeneity in general measures the algorithm's ability to produce clusters in which each data sample belongs to the same class label. Completeness on the other hand measures the algorithm's ability to map all data samples belonging to a single class label to a single cluster. Homogeneity is a metric accurately describing the degree of class labels overlapping in clusters. It however yields a high score even for solutions with an unnecessarily high number of clusters. In a trivial scenario, a single class label divided into however many clusters yields a homogeneity score of 1.0. Completeness is a counter-weight to this. It does not deal with clusters overlapping, but yields a lower score when a class label is distributed among more clusters. A trivial scenario in which several class labels are all mapped onto a single cluster, completeness yields a score of 1.0. The metrics are demonstrated in Figure 5.1.

5.1 Challenges

An issue is once again a lack of data. Images of real agar plates annotated for the purpose of clustering are basically non-existent. This problem is however in this case addressed by



(a) Homogeneity = 1.0

(b) Completeness = 1.0

Figure 5.1: In (a) we can see that even if the clusters do not align the Homogeneity can still be 1.0. This is true as long as no two objects with different class label map onto the same resulting cluster. Similarly, in (b) we can see that if there is only one resulting cluster, the Completeness will always be 1.0. This is true as long as no two objects with the same class label map onto different resulting clusters.

the purpose-built sample generator, which can produce culturally variant images of agar plates along with both the segmentation masks and corresponding class labels.

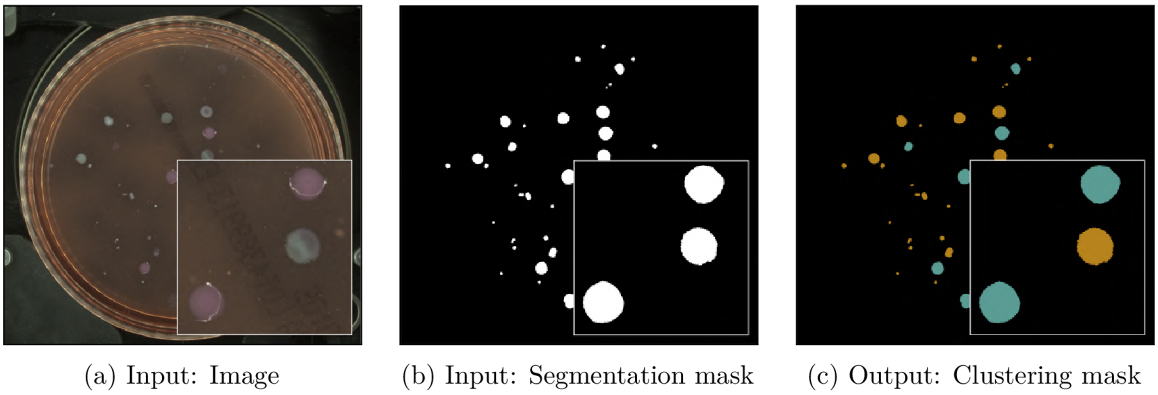
More problems stem from the fact, that the number of species to look for, especially in cases where we are trying to determine, whether a dish has been infested by an unexpected species, is not known beforehand. Any solution selected needs to reflect this issue and be flexible enough to accommodate a variable number of clusters.

As the issue of detection and segmentation has already been tackled and improved in the previous chapter, the clustering algorithms can be proposed relying on the segmentation binary mask as an input alongside the plate image itself. This pair will therefore be used to obtain a third image, a class label mask where each class is represented by a single arbitrary color, as can be seen in Figure 5.2.

5.2 Proposed approaches

To address the issue of an uncertain number of classes I decided to use the K-Means algorithm for all of the clustering itself and focus more on the possible ways of feature extraction.

The clustering step follows the segmentation step in the general workflow. For this reason, the clustering algorithm will be provided the segmentation mask produced by the segmentation model. The image can therefore either be processed in a pixel-wise manner, or each detected lump can be represented by a single data point and processed as a whole. This approach would allow for more features to be extracted, such as those pertaining to texture, since there is only very little information contained within a single pixel aside of its color composition. These features can be extracted manually, which is in line with how the agar plates are evaluated in real laboratories nowadays. A laboratory technician



(a) Input: Image (b) Input: Segmentation mask (c) Output: Clustering mask

Figure 5.2: A custom made artificially created agar plate with two distinct species – one gray, the other slightly reddish – with a precise binary segmentation mask. These will be used as hypothetical inputs for the proposed clustering algorithms. The third image represents the clustering ground truth. The colors indicate the class labels, but are otherwise arbitrary. All images show a small portion of itself enlarged for detail.

looks at the colonies and judges their color properties, their shape, texture or discoloration of the surroundings.

Alternatively, since the images have fixed resolutions, the pixel-wise approach also offers the opportunity to employ machine learning. A neural network asked to reconstruct an image of a cultivated agar plate extracts features from the image during the process of encoding. These extracted features can be used for clustering purposes too.

I decided to set the pixel-wise clustering of RGB values to be the baseline, as this is probably the most simple approach one can take. Then I decided to use a U-Net architecture trained as an autoencoder to provide its obtained features for each pixel in the image. Lastly, I propose a manual approach to extracting features from clearly separated lumps detected by the segmentation algorithm akin to what the laboratory technicians do.

5.2.1 Knee Point Detection for K-Means

The issue with using the K-Means algorithm in all of these approaches is the fact, that the number of clusters has to be specified beforehand. I however do not possess any information about the optimal cluster count prior to analysing any plate. This issue is usually addressed by employing a technique called Knee/Elbow Point Detection.

The K-Means algorithm is ran a specific number of times, each time with a different cluster count. Each run can then be evaluated either by Inertia scores or Distortion scores. Once these scores are projected onto a graph and interpolated, a knee/elbow point is the point with the biggest curvature, from which the numbers start decreasing in a linear fashion, as can be seen demonstrated in Figure 5.3. In my implementation I have used the Python library `kneed`[1], which encapsulates this task and provides the knee points if any are detected.

A Distortion score is an average of the squared euclidean distances from the cluster centers of the respective clusters. It is therefore a metric describing cluster togetherness. An Inertia is the sum of squared distances of samples to their closest cluster center.

[1] `kneed` library – `https://pypi.org/project/kneed/`

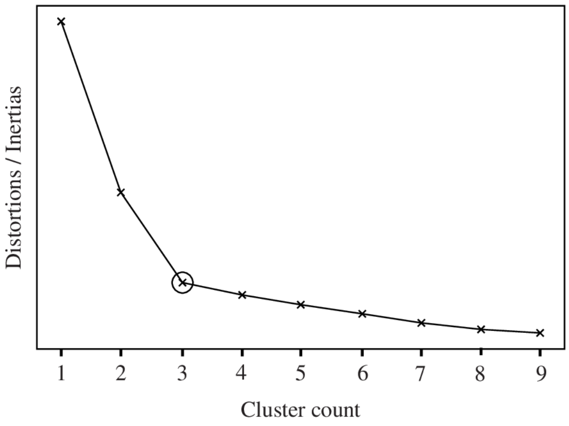


Figure 5.3: This figure demonstrates the location of the knee/elbow. Pictured are either Inertia or Distortion scores for various cluster counts. The values of the scores themselves do not matter. To locate the knee, one must locate the point of the biggest curvature, after which the values start descending in a more linear fashion. In this picture, the knee is visibly located at the three cluster mark. Is is otherwise possible that a knee can not be located by one or even both metrics.

In all the following clustering attempts the cluster counts examined will be in the range of 1 - 10. Therefore, each clustering on a single dish will have to be ran ten times to obtain a knee point. These numbers were selected arbitrarily, as no more than ten species on a single dish still seems plausible. Since I however generate my own data to experiment on, I can make sure there will never be more species on a single dish.

5.2.2 Pixel-wise RGB clustering

Possibly the most straightforward approach towards clustering the colonies on an image is to create a posterized version of said image, where the knee point detection helps with determining the cluster count and the feature vectors are just each pixel's RGB values normalized to a range of 0.0 - 1.0. Once the pixels are clustered ten times and a knee point is found, they are colored according to their cluster identity. The colors are arbitrary, they only have to be unique within the picture. This can be seen demonstrated in Figure 5.4b.

Once that is done, applying the binary mask on the resulting image produces something really close to the desired result, as demonstrated by Figure 5.4c.

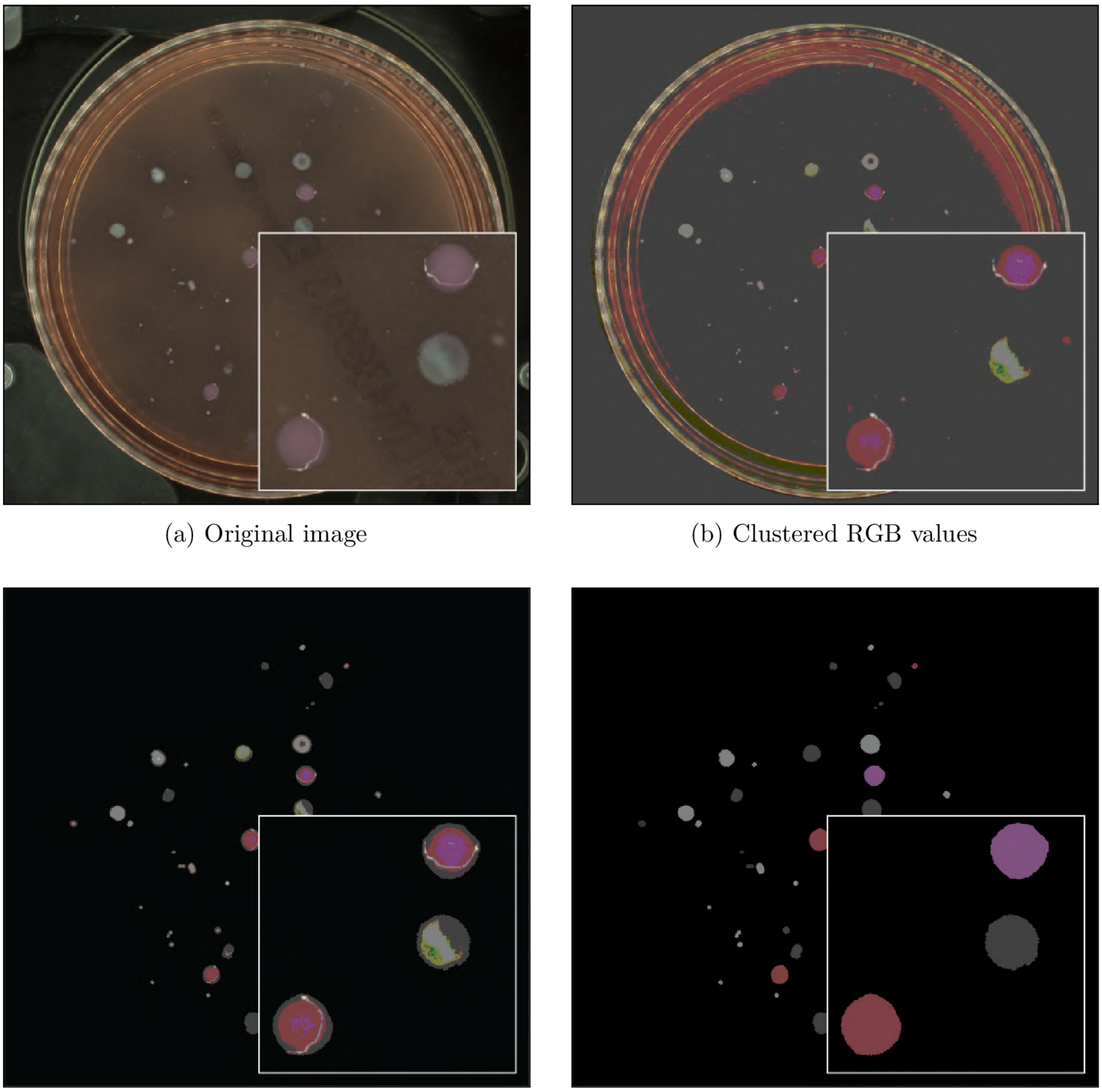
Each distinct lump can now be taken separately and analyzed in terms of its newly gained color composition. The simplest way of determining the cluster to which it belongs to is to simply take the most frequent color and fill the lump with it, as shown in Figure 5.4d.

5.2.3 Pixel-wise U-Net feature extraction

This approach is similar to the previous one. Instead of using the RGB channel values as features, it uses the features synthetised by a U-Net autoencoder for each of the pixels.

To train the model[2] a binary cross entropy loss function was employed along with a learning rate of 1e-4 and a batch size of 16 images with an ADAM optimizer[8]. The autoencoder was trained on the BRUKERCOLONY dataset of real agar plate images. To save time and

[2] Implementation taken from: https://github.com/zhixuhao/unet



(a) Original image

(b) Clustered RGB values

(c) Applied segmentation mask

(d) Dominant colors decide cluster

Figure 5.4: In (b) the result of clustering the original image's pixels based on their RGB values can be seen. The colors of the resulting clusters indicated are the mean values within said clusters. This creates the posterization effect, although for our purposes the colors could be completely arbitrary. The dishes surroundings has been cut to keep the cluster count relevant and minimal. In (c) the result after the application of the segmentation mask can be seen. The resulting multicolored lumps are then filled with the most dominant color within said lumps. This color then defines the cluster the lump belongs in. The resulting image is now in the correct format comparable to the ground truth seen in (d).

for easier manipulation, both the input and output of the U-Net were resized to 512x512, which significantly lowered the computational load and made up and down-sampling simpler. As this downscaling does not change the ratios of the clustered pixels, this resizing operation should have only minimal effect on the overall result.

The model's architecture is outlined in Figure 5.5. The features are extracted from the between the last block of convolutional layers after the last up-sampling is made within the net, as demonstrated by Figure 5.6. The resulting feature tensor has the shape 512x512x16. One vector of the sixteen values corresponding to a single pixel in the original image represents that pixel's feature vector.

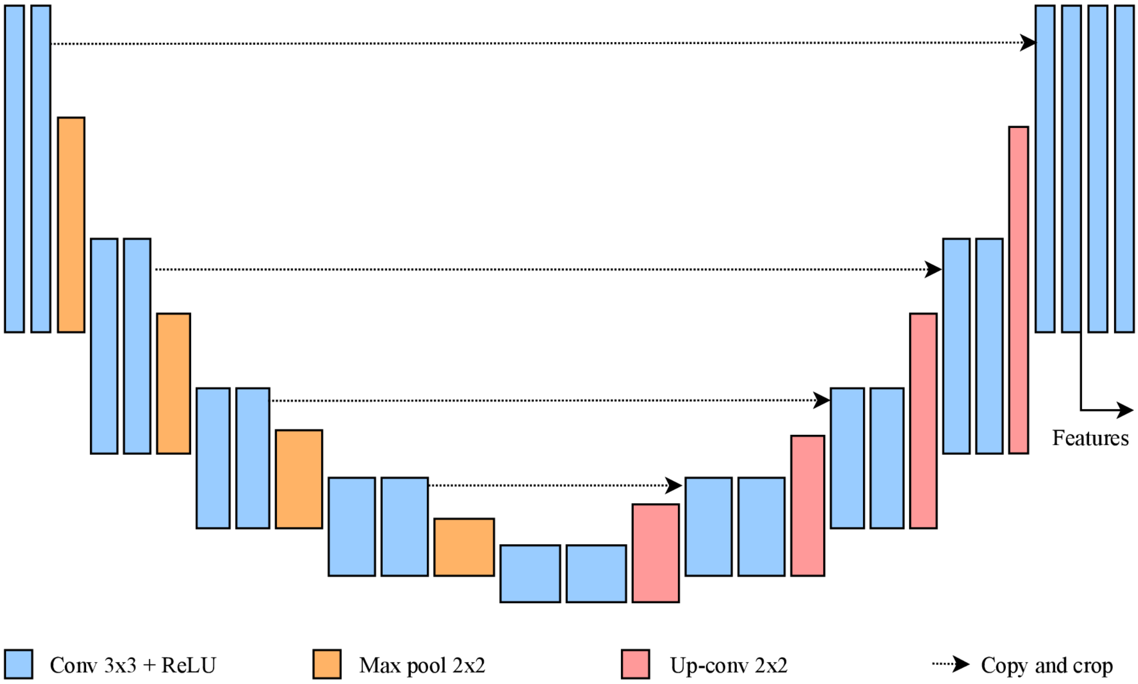


Figure 5.5: A simplified scheme of the U-Net autoencoder.

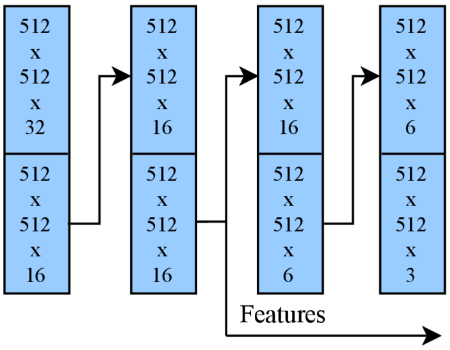
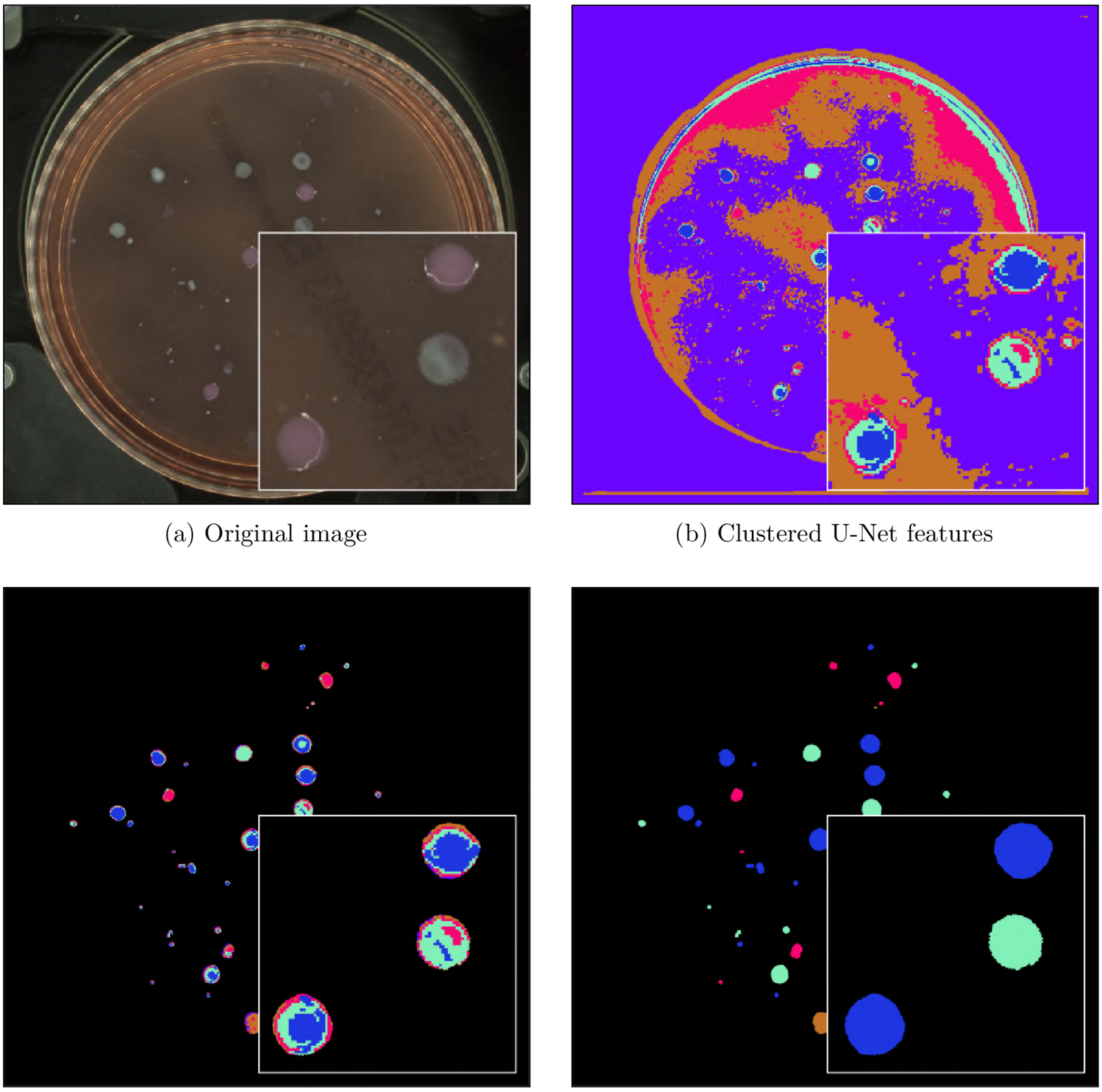


Figure 5.6: A closer look at the output part of the U-Nets topmost layer. The output of the second concatenation layer is taken and used as a feature tensor. The numbers in the upper half of the blocks describe the shape of the data as it enters the block. The lower half describes the data as it leaves the block. As this image describes the final section of the decoder, the resulting depth of the image is 3, which corresponds to the RGB channels.

Once the feature vectors are extracted, they are normalized to a range of 0.0 - 1.0, and then they are fed into the K-Means algorithm ten times to get the optimal cluster count by detecting the knee point. The result is then taken and the pixels are colored according to their cluster, as can be seen in Figure 5.7b. The colors are once again arbitrary, as long as each color is unique. It is at this point that the image is enlarged to the original image's size. As in the preceding approach, the binary mask is applied, creating multi-colored lumps of interest, seen in Figure 5.7c. Once the dominant color is determined, the final colors can be assigned to each lump, as seen in Figure 5.7d.



(a) Original image

(b) Clustered U-Net features

(c) Applied segmentation mask

(d) Dominant colors decide cluster

Figure 5.7: In (b) the result of clustering the pixel-wise features extracted from the U-Net autoencoder can be seen. The colors are arbitrary, each color representing a single cluster. Similarly as with the RGB clustering, the surroundings of the dish was cut off to keep the clusters relevant. In (c) the result after the application of the segmentation mask can be seen. The resulting multicolored lumps are then filled with the most dominant color within said lumps, which then defines the cluster the lump belongs in.

5.2.4 Manual colony feature extraction

The last approach is more akin to what currently happens in the laboratories when agar plates are analysed. Instead of clustering the pixels the algorithm clusters entire lumps of material based on their properties.

Initially, the binary mask is applied and each resulting isolated lump is taken separately. Then an average agar color of the dish is determined by simply taking all of the pixels within the dish not indicated by the binary mask and averaging their channel values. Once that is done, the RGB image can be converted to an RGBA image, where the alpha channel is obtained by applying Equation 3.2 to each of the pixels. The pixels can therefore be inversely weighted in terms of their similarity to the surrounding agar, which means the colonies can now effectively be worked with as if their background agar was keyed out. This allows for features to be extracted from the lumps and the surrounding agar separately. This is useful for getting information about colonies weighted by their relevance (pixels very similar to surrounding agar are probably not as relevant), as well as for getting information about the surrounding agar only, as it could be for example be discolored and provide information about the species present on the dish as well.

Once the alpha channel is available, the features extracted are:

- {avg, median}_{value, saturation, red, green, blue} – weighted averages and medians of the HSV and RGB components.
- stdev_{value, saturation, red, green, blue} – standard deviations of the HSV and RGB components.
- avg_alpha – an average of the previously calculated alpha values.
- area – the sum of the lump's pixels divided by the dish area.
- irregularity – length of the lump's contour divided by a square root of its area.
- avg_agar_{red, green, blue} – weighted by inverted alpha to put more emphasis on the surrounding agar.

All of the averages and medians are weighted by the pixels' alpha values and all values have been normalized in the range of 0.0 – 1.0. The hue has been omitted, as because of its cyclical nature it is not easily usable as a feature for the K-Means algorithm.

Once these features are extracted for each of the lumps, K-Means knee detection can be run similarly to the previous approaches to determine the optimal number of clusters. These clusters are then indicated in the cluster map as seen in Figure 5.8, again with a single arbitrary color representing a single cluster with the colors being unique across the picture.

5.3 Evaluating the clustering approaches

To accurately compare the proposed clustering approaches, several things have to be noted.

First – as has been mentioned before, no agar plate datasets annotated for the purposes of clustering exist. For this reason I will be making my own custom made dataset using the dish generator.

Next – all three of the described approaches rely heavily on good segmentation. Any mistake done by the segmentation algorithm would only get propagated forward and can not be mitigated by clustering. For this reason and for the purposes of fair comparison, a separate

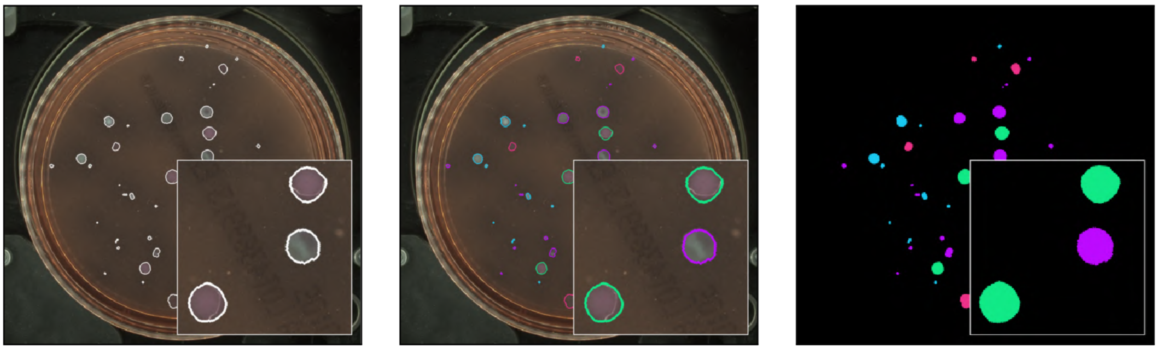

(a) Input: Image
(b) Input: Segmentation mask
(c) Output: Clustering mask

Figure 5.8: At first, the areas within the contours marked in white are analysed and features are extracted from them. Then, once they are clustered, the final clustering mask as seen on the right can be created.

experiment will be conducted in which all clustering algorithms will be provided the ground truth binary segmentation mask. This allows for a fair comparison among the clustering approaches. The resulting clustering masks are compared in a pixel-wise manner. Black pixels (those not indicating a presence of any species) represent their own cluster. They will be therefore irrelevant in the experiment in which the ground truth segmentation mask is provided, as they will map onto their class label counterpart perfectly.

And lastly – it is irrelevant how long the clustering of each plate using any of the algorithms takes, as the time complexity is not an issue in the considered scenario. Any actual real life implementation of these solutions would probably need to be written in a language other than Python and optimized. The performance of these algorithms will therefore not be measured.

5.3.1 Generating the testing dataset

For this experiment a dataset of 200 samples has been generated by the generator. The parameters were set as follows:

- 1-5 species,
- 25 % halo probability,
- 33 % haemolysis probability,
- 5-50 % of area to be covered by growth,
- 50 % probability of a balanced dish,
- HSV editing disabled.

Both the binary masks and the class labels were generated alongside the images.

5.3.2 Comparing the approaches

All three of the proposed clustering algorithms were given the generated testing dataset along with the ground truth binary segmentation masks. The algorithms were set to find a Knee point in both the projected Distortions and Inertias and then to prefer the higher value, since we are primarily trying to maximize homogeneity. The resulting cluster maps were compared against the generated ground truth class label images in a pixel-wise manner and evaluated by the V-measure and homogeneity metrics.

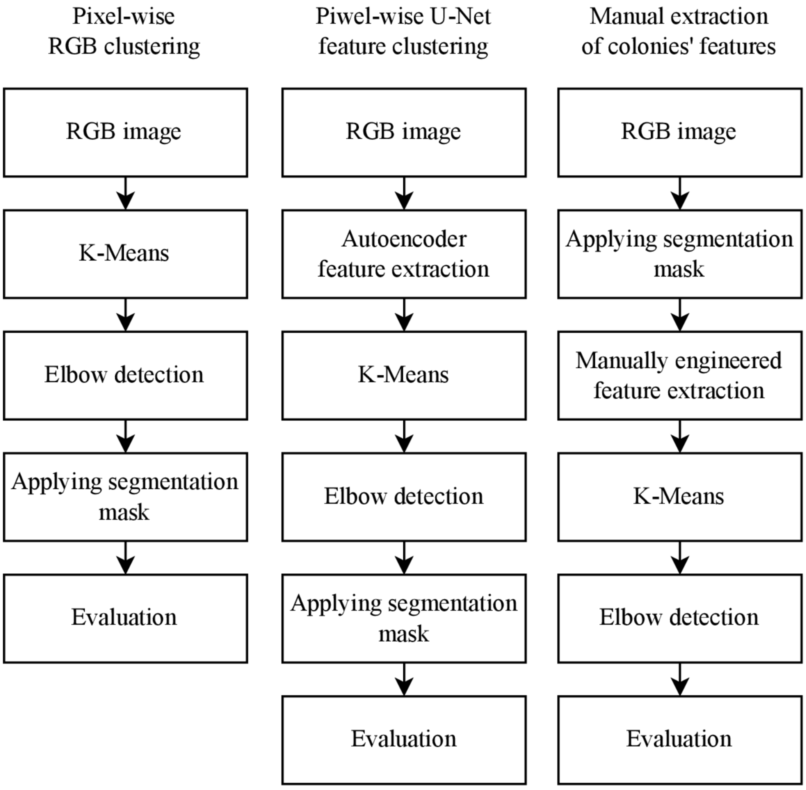


Figure 5.9: The three proposed approaches to clustering colonies summarized and compared.

The achieved scores of the experiment, in which the algorithms were provided the ground truth segmentation mask, can be seen in Table 5.1.

The autoencoder approach does not seem to yield particularly good results when compared to a simple clustering of the RGB values. It is at best on par, if not a little worse in both the V-measure and Homogeneity scores. The manual approach to feature extraction on the other hand yields better results. It surpasses the other two approaches clearly using either metric.

Table 5.1: A comparison of the three proposed clustering algorithms. While the V-measure describes the overall precision of the clustering attempt, homogeneity is more important in the hypothetical scenario outlined.

Clustering approach	V-measure	Homogeneity
RGB pixel-wise	0.908	0.926
Autoencoder pixel-wise	0.906	0.919
Manual features	**0.927**	**0.939**

The results are very similar when the segmentation model is used, as can be seen in Table 5.2. While there is a significant drop in the achieved scores across the whole table, it seems to be fairly uniform. The manual approach once again beats the other two approaches, which yield almost identical results.

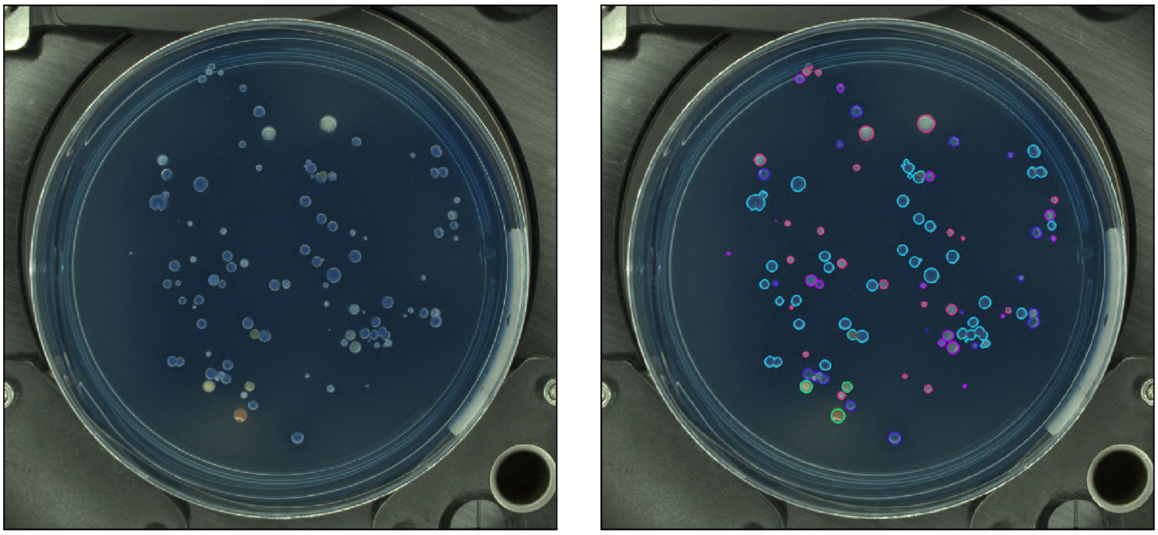
Table 5.2: A comparison of the of the same algorithms, but with the segmentation model producing the segmentation mask on the fly. No segmentation mask or other additional input was therefore provided, only the image of the agar plate. This table therefore in a way summarizes the entire proposed pipeline's performance.

Clustering approach	V-measure	Homogeneity
RGB pixel-wise	0.819	0.796
Autoencoder pixel-wise	0.818	0.789
Manual features	**0.830**	**0.813**

One can surmise from this, that the autoencoder feature extraction as described in this thesis is not a particularly effective approach, as a much simpler clustering of RGB values yields as good, if not better results. Manually extracting features from the detected colonies, however, seems to be a good way of approaching this issue.

Ultimately, it seems as though it is possible to perform clustering over different species on agar plates, and to even do so with reasonable efficiency. It is of course not real data upon which this experiment has been conducted, but similarly created data has been proven to be efficient in standing in for real data earlier in this thesis. When the entire pipeline as described in 2.8 is utilized, a score of 0.830 for V-measure can be achieved.

To better illustrate this performance, I also processed several images of the unlabeled real data acquired during the creation of this thesis. A cherry-picked example illustrating the results produced by the proposed pipeline on real data can be seen in Figure 5.10. While this result can be assessed only subjectively, the algorithm does seem to localize and group similarly colored colonies well.



(a) Real plate

(b) Fully processed real plate

Figure 5.10: In (a) one of the newly acquired real plates can be seen. Among the blue colonies, several pale and orange ones can also be seen. In (b) a very good performance of the segmentation model can be noted. Possibly because of this, also the clustering yielded good results, as the orange colonies were sorted into their own cluster, pale colonies likewise and the blue ones were sorted into several clusters according to their overall brightness, which is in line with the previously stated preference for homogeneity over completeness.

Chapter 6

Conclusion

This thesis addressed two current issues in laboratory analysis of agar plates: detection and localization of microbial colonies on agar plates, and grouping of said colonies according to visually perceived features. The lack of labeled data necessary for machine learning approaches to segmentation was overcome by synthetic data generation using a multi-purpose agar plate image generator. The generator utilizes images of empty dishes combined with pre-keyed and labeled colony cutouts gathered from available real data. Enlarging the training dataset with the generated synthetic data improved the segmentation model's performance by 41 % and 45 % for F1 score and Jaccard index respectively. The improved segmentation model will be deployed in a prototype of the MBT Pathfinder colony picking instrument for testing in a real-life laboratory environment.

To group the colonies, a K-Means alghorithm combined with the Elbow/Knee Point Detection was employed to determine an optimal cluster count. Three approaches towards feature extraction have been proposed and tested on synthetic data. The approach based on manually extracting features from segmented microbial colonies performed best, achieving a V-Measure score of 0.93 when tested on its own, and a score of 0.83 when combined with the segmentation model. While its performance on real data could not be quantified, upon subjectively assessing the results I believe the approach seems promising.

Any future work regarding this topic would make use of the MBT Pathfinder's various lighting modes, as there is a lot of information to be extracted from a single plate just by changing up the lighting conditions. I also believe that a well labeled and variable collection of colonies is imperative for the sample generator to produce good synthetic samples. These colonies can also be gathered while gradually exposed to the multiple lighting conditions, which opens up the possibility of generating differently lit versions of realistic agar plates images. The ability to generate colonies in more realistic layouts better reflecting the inoculation process could also be introduced.

I have presented this thesis at the Excel@FIT 2023 conference where it received an award for a practical and uncomplicated solution, which contributes to efforts for achieving total laboratory automation. Overall, the approaches described in this thesis offer a cheap, fast and very effective way of augmenting data for the purposes of segmentation and outline several promising ways of clustering microbial colonies on agar plates, which is the next hurdle to overcome in total laboratory automation of this field.

Bibliography

[1] Beznik, T., Smyth, P., Lannoy, G. de et al. *Deep Learning to Detect Bacterial Colonies for the Production of Vaccines.* arXiv, 2020. DOI: 10.48550/ARXIV.2009.00926.

[2] Cicatka, M., Burget, R. and Karasek, J. Machine-learning Approach to Microbial Colony Localisation. In: *2022 45th International Conference on Telecommunications and Signal Processing (TSP).* 2022, p. 206–211. DOI: 10.1109/TSP55681.2022.9851236.

[3] da Silva, G. R., Rosmaninho, I. B., Zancul, E. et al. Image dataset of urine test results on petri dishes for deep learning classification. *Data in Brief.* 2023, vol. 47, p. 109034. DOI: https://doi.org/10.1016/j.dib.2023.109034. ISSN 2352-3409.

[4] Dijkstra, K., van de Loosdrecht, J., Atsma, W. A., Schomaker, L. R. et al. CentroidNetV2: A hybrid deep neural network for small-object segmentation and counting. *Neurocomputing.* 2021, vol. 423, p. 490–505. DOI: https://doi.org/10.1016/j.neucom.2020.10.075. ISSN 0925-2312.

[5] Ferrari, A., Lombardi, S. and Signoroni, A. Bacterial colony counting with Convolutional Neural Networks in Digital Microbiology Imaging. *Pattern Recognition.* 2017, vol. 61, p. 629–640. DOI: https://doi.org/10.1016/j.patcog.2016.07.016. ISSN 0031-3203.

[6] Geissmann, Q. OpenCFU, a New Free and Open-Source Software to Count Cell Colonies and Other Circular Objects. *PloS one.* february 2013, vol. 8, p. e54072. DOI: 10.1371/journal.pone.0054072.

[7] Khan, A. u. M., Torelli, A., Wolf, I. et al. AutoCellSeg: robust automatic colony forming unit (CFU)/cell analysis using adaptive image segmentation and easy-to-use post-editing techniques. *Scientific Reports.* May 2018, vol. 8, no. 1, p. 7302. DOI: 10.1038/s41598-018-24916-9. ISSN 2045-2322.

[8] Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In: *3rd International Conference for Learning Representations.* 2014.

[9] Kostrzewa, M. Application of the MALDI Biotyper to clinical microbiology: progress and potential. *Expert Review of Proteomics.* Taylor & Francis. 2018, vol. 15, no. 3, p. 193–202. DOI: 10.1080/14789450.2018.1438193. PMID: 29411645.

[10] Madigan, M., Bender, K., Buckley, D. et al. *Brock Biology of Microorganisms.* 15th ed. Prentice Hall, 2017. ISBN 0-13-144329-1.

[11] Majchrowska, S., Pawłowski, J., Guła, G. et al. *AGAR a microbial colony dataset for deep learning detection.* 2021.

[12] Mitchell, M. *An Introduction to Genetic Algorithms.* The MIT Press, 1998. ISBN 9780262631853.

[13] Naets, T., Huijsmans, M., Smyth, P. et al. *A Mask R-CNN approach to counting bacterial colony forming units in pharmaceutical development.* arXiv, 2021. DOI: 10.48550/ARXIV.2103.05337.

[14] Nagro, S. A., Kutbi, M., Eid, W. M. et al. Automatic Identification of Single Bacterial Colonies Using Deep and Transfer Learning. *IEEE Access.* 2022, vol. 10, p. 120181–120190.

[15] Nie, D., Shank, E. A. and Jojic, V. A Deep Framework for Bacterial Image Segmentation and Classification. In: *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics.* New York, NY, USA: Association for Computing Machinery, 2015, p. 306–314. BCB '15. DOI: 10.1145/2808719.2808751. ISBN 9781450338530.

[16] Pawłowski, J., Majchrowska, S. and Golan, T. Generation of microbial colonies dataset with deep learning style transfer. *Scientific Reports.* Mar 2022, vol. 12, no. 1, p. 5212. DOI: 10.1038/s41598-022-09264-z. ISSN 2045-2322.

[17] Ronneberger, O., Fischer, P. and Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation.* 2015.

[18] Rosenberg, A. and Hirschberg, J. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In:. January 2007, p. 410–420.

[19] Savardi, M., Ferrari, A. and Signoroni, A. Automatic hemolysis identification on aligned dual-lighting images of cultured blood agar plates. *Computer Methods and Programs in Biomedicine.* 2018, vol. 156, p. 13–24. DOI: https://doi.org/10.1016/j.cmpb.2017.12.017. ISSN 0169-2607.

[20] Zhu, G., Yan, B., Xing, M. et al. Automated counting of bacterial colonies on agar plates based on images captured at near-infrared light. *Journal of Microbiological Methods.* 2018, vol. 153, p. 66–73. DOI: https://doi.org/10.1016/j.mimet.2018.09.004. ISSN 0167-7012.

Appendix A

Acquired empty agar plates

Table A.1: a table summarizing all of the agars acquired for the purposes of data augmentation – by first taking images when empty and then after amateur inoculation. All agars were bought from Thermo Fisher Scientific Brno s.r.o. and are traceable by their Reference number. Damaged dishes were stored near refrigerator walls, which froze and irreversibly damaged their content.

Reference	Name	Color	Transparency	Ok/Dmg
PO5163A	mCP Selective Agar	purple	transparent	10/0
PO5029A	MLCB Agar	blue	transparent	10/0
PO5060A	China Blue Lactose Agar	blue	transparent	10/0
PO5302A	Brilliance ESBL Agar	white	opaque	10/0
PO5146A	MacConkey Agar	red	transparent	10/0
PO5100A	Hektoen Enteric Agar	green	transparent	10/0
PO5090A	Chocolate Agar w/ Vitox	brown	opaque	10/0
PB5023A	Aesculin Blood Agar (Modified)	red	opaque	10/0
PO5048A	PEMBA	green	transparent	10/0
PO5091A	CCDA Selective Medium	black	opaque	10/0
PO5069A	Sorbitol MacConkey Agar	pink	transparent	10/0
PO5098A	Brilliance Salmonella Agar	white	opaque	10/0
PO5009A	C.L.E.D. Medium	green	transparent	7/3
PB5008A	Columbia Blood Agar w/ Sheep Blood	red	opaque	9/1
PO5057A	X.L.D. Medium	red	transparent	10/0
PO5110A	MacConkey Agar No. 2	maroon	transparent	6/4
PO5016A	Desoxycholate Citrate Agar	pink	transparent	10/0
PO5027A	Mannitol Salt Agar	pink	transparent	10/0
PO5055A	Malt Extract Agar	yellow	transparent	10/0
PO5014A	Baird Parker Agar	yellow	opaque	10/0
PO5017A	Slanetz and Bartley Medium	beige	transparent	10/0
PO5001A	Sabouraud Glucose Agar	yellow	transparent	8/2
PB5003A	Iso-Sensitest Agar w/ Sheep Blood	red	opaque	10/0

Appendix B

Autoencoder experiment

Working with the cluster maps of features extracted from the autoencoder gave me the idea of using them to aid segmentation. Many of these maps at a first glance already indicate the presence of colonies fairly well, as seen in Figure B.1.

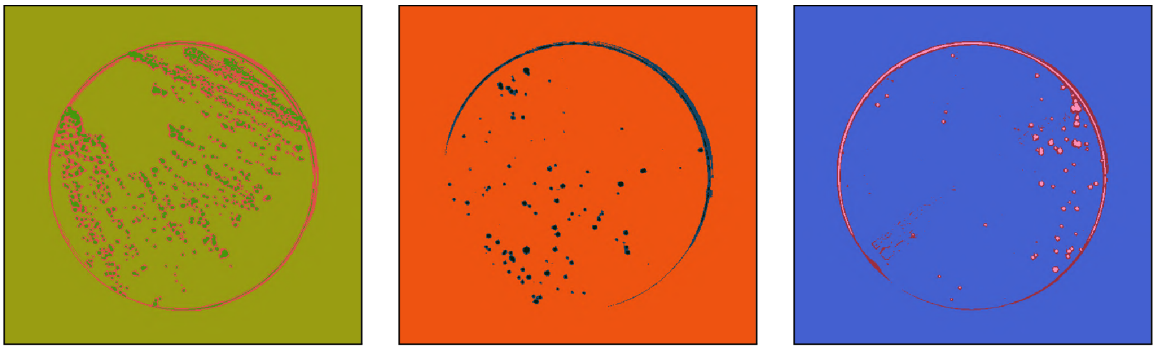


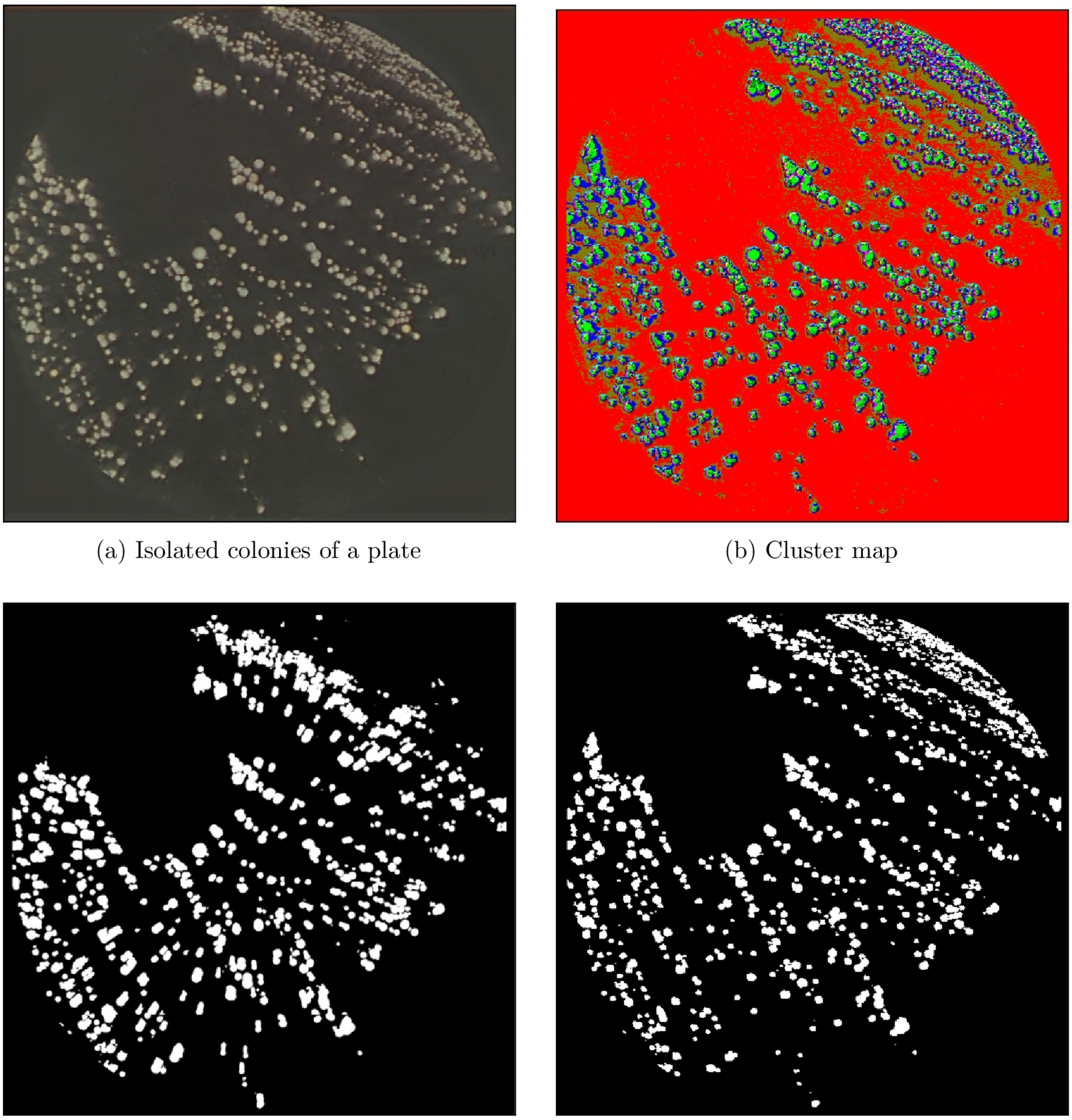
Figure B.1: Clustering features extracted from the autoencoder yields images fairly close to a reasonably good segmentation mask.

The issue is determining which clusters should be set to indicate the colonies and which clusters should indicate the various shades of the agar. Some dishes have growth covering more than 50 % of its surface and often the agar gets divided among several clusters. For this I thought the segmentation model – though not always precise – could still be used rather well. If the cluster overlaps more with where the mask indicates colonies are, the cluster corresponds to colonies, and vice versa.

This approach lead to very promising results with a single plate which I used for development. As can be seen in Figure B.2, the acquired segmentation mask corrected two of the errors the segmentation model made, and was generally more precise.

Based on these findings I have scaled up and attempted a larger experiment on more plates. Sadly, it turned out the initially selected dish was a lucky pick. While with some plates this approach did genuinely help, overall, it did more harm than good and I found no way of pre-determining which case any individual plate will be. If the segmentation model fails on a significant part of the dish, an entire cluster may then be falsely attributed to the wrong class and the error is aggravated, as seen in Figure B.3.

Experimenting with this approach showed, that the results are generally better when the cluster count is higher, as it seems to help with the finer details. In the end I worked with a fixed cluster count of 10, as the effect of even more clusters seemed negligible. Nev-



(a) Isolated colonies of a plate

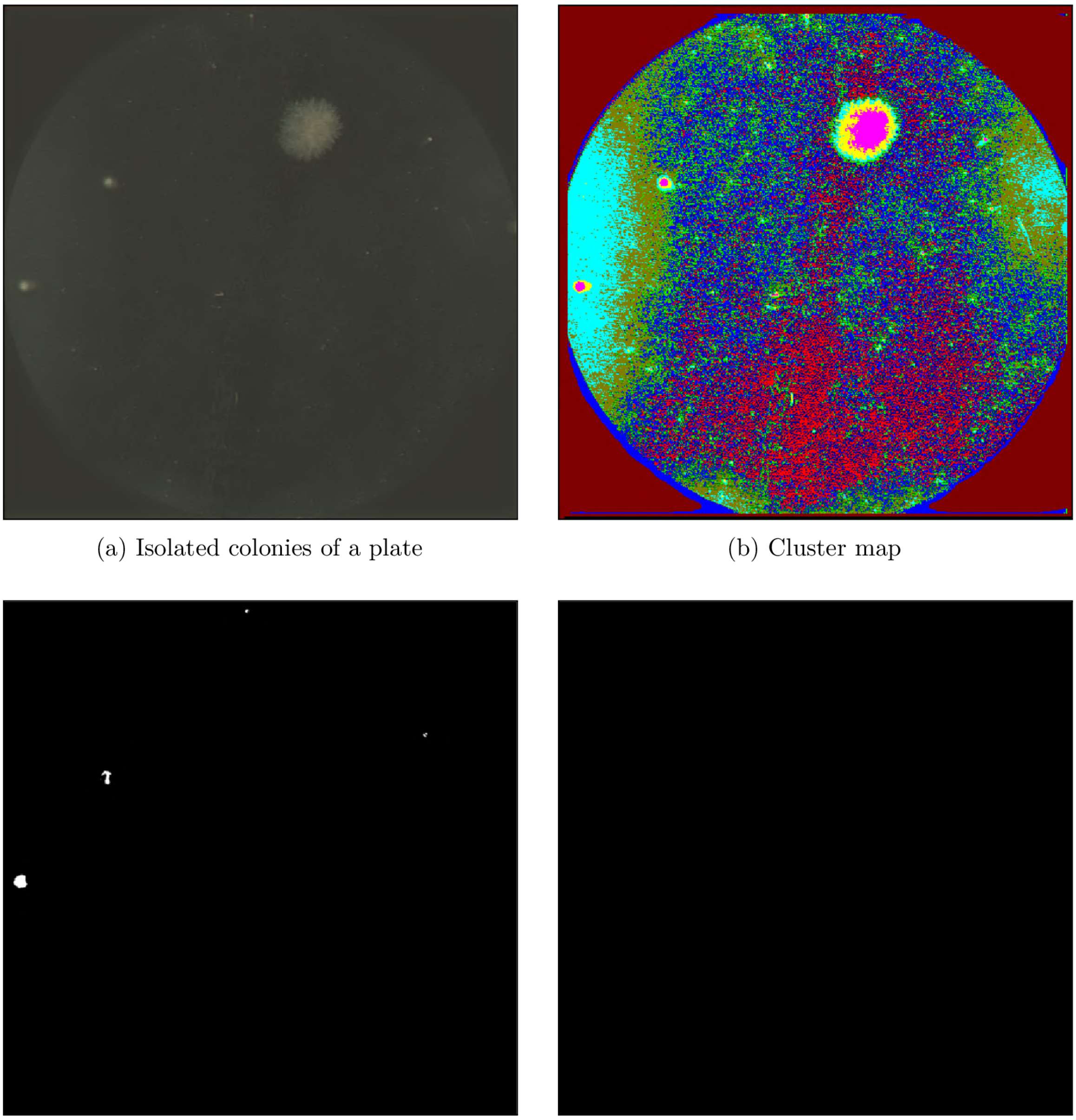
(b) Cluster map

(c) Conventional segmentation map

(d) Cluster based segmentation map

Figure B.2: The algorithm's input can be seen in (a). the surrounding of a dish is cropped and replaced with the agars average color, so that no unnecessary clusters are created. The resulting cluster map can be seen in (b). In (c), the output of the segmentation U-Net model can be seen. It missed a big chunk of colonies in the upper right, as well as confused a reflection of the colonies caused by lighting for actual matter, creating the resulting warping effect. In contrast, both errors have been remedied by the proposed approach, as can be seen in (d).

ertheless, working with 30 dishes hand-picked for their diversity, I never achieved an overall score better than when using just the segmentation model alone. This approach therefore seems only situational, as I have found no way of generalizing it well. I have therefore aban-



(a) Isolated colonies of a plate

(b) Cluster map

(c) Conventional segmentation map

(d) Cluster based segmentation map

Figure B.3: We can see in Figures B.3b and B.3c that the large piece of presumably mold was not detected. This has caused the pink cluster to be labeled an agar, which in return also labeled the other two correctly detected colonies an agar, aggravating the error.

doned any efforts to utilize the clustered autoencoder feature maps to aid segmentation in this manner.