FACULDADE
DE ECONOMIA
UNIVERSIDADE
DO PORTO

Deep Neural Networks in Medical Microbiology

Bacterial Colonies Classification

José Duarte Pinho Pereira

Dissertation

Master in Modelling, Data Analysis and Decision Support Systems

Supervised by:

PhD João Manuel Portela da Gama

PhD Bruno Miguel Delindro Veloso

2023

Biographical note

I hold a master's degree in Medicine from the Faculty of Medicine at the University of Coimbra, which was completed in the year 2018.

In the following year 2019, I undertook a General Training Internship in Centro Hospitalar de Santo António, in Porto.

Currently, I am in the final year of my residency program in the medical specialty of Clinical Pathology, which was initiated in 2020 at Instituto Português de Oncologia do Porto.

Acknowledgments

First, I would like to express my heartfelt gratitude to Inês for her unwavering support, understanding, patience, and constant belief in me throughout this journey. Without her, none of this would have been possible.

I extend my sincere thanks to my family for always believing in me and supporting my decisions.

I wish to convey my appreciation to my residency supervisor, Dr. Nuno Gonçalves, for supporting my pursuit of this master's degree. His guidance and knowledge have been invaluable to my medical career.

I would like to acknowledge and thank my Head of Department, Dra. Gabriela Martins, for recognizing the scientific merit of this project and wholeheartedly supporting it as a valuable contribution to my career.

My gratitude goes out to all my colleagues at work. Their support and teamwork have been essential in achieving our collective goals.

Last but certainly not least, I would like to thank my supervisor, Prof. Doutor João Gama, and my co-supervisor, Prof. Doutor Bruno Veloso, for their support, valuable suggestions, and reviewing of my dissertation.

Abstract

Automation has become important in medical laboratories, especially in clinical chemistry and hematology. However, microbiology laboratories still heavily rely on manual processes. One particularly time-consuming step in microbiology labs is the culture of samples on agar plates, followed by their manual review for microorganism identification and antibiotic profile analysis. This dissertation addresses this issue by employing deep-learning methods for detecting microbial colonies.

Divided into two parts, the first part of this dissertation focuses on training and enhancing the accuracy of four models using the Annotated Germs for Automated Recognition (AGAR) dataset. The models include Faster R-CNN and RetinaNet with two backbones (ResNet 50 and ResNet 101). Transfer learning and ensemble methods are employed for performance improvement.

The second part involves creating a new dataset with 165 images of agar plates with colonies of *S. aureus*, *P. aeruginosa*, and *E. coli* on various types of culture media. The same models are trained on this dataset, and similar techniques are applied to improve performance. Transfer learning is tested using the weights from the models trained on the AGAR dataset.

The initial training achieves a mean Average Precision (mAP) of 62 % on the AGAR dataset, with Faster R-CNN ResNet 101 delivering the best performance. A value of 66.40 % is achieved with the application of the ensemble method – Weight Boxes Fusion, surpassing the results from the creators of the AGAR dataset. On the new dataset, RetinaNet ResNet 50 achieves a value of 52.40 %, improved to 56.30 % using an ensemble of the results.

In summary, this dissertation successfully enhances the performance of the AGAR dataset, introduces a new dataset, and employs various techniques to improve performance. It represents a significant step forward in applying deep learning methods to laboratory medicine.

Keywords: Microbiology; Deep Learning; Object Detection; Bacterial Colonies.

Resumo

A automatização tem ganho importância nos laboratórios médicos, especialmente nos setores de química clínica e hematologia. No entanto, os laboratórios de microbiologia ainda dependem muito de processos manuais. Um passo particularmente moroso é a cultura de amostras em placas de agar, seguida da sua revisão manual para posterior identificação dos microrganismos e análise do perfil de antibióticos. Esta dissertação aborda esta questão através da aplicação de métodos de *deep learning* para a classificação de colónias microbianas.

Dividida em duas partes, a primeira parte do projeto foca-se no treino e na melhoria da performance de quatro modelos na base de imagens anotadas *Annotated Germs for Automated Recognition* (AGAR). Os modelos são Faster R-CNN e RetinaNet, com duas arquiteturas distintas (ResNet 50 e ResNet 101). Métodos de *transfer leaning* e *ensemble* são aplicados para melhoria da performance.

A segunda parte envolve a criação de uma nova base de imagens com 165 imagens anotadas de placas de agar com vários tipos de meios de cultura, com colónias de *S. aureus*, *P. aeruginosa* e *E. coli*. Os mesmos modelos são treinados e são aplicadas técnicas semelhantes para melhoria do desempenho. O *transfer leaning* é aplicado usando os pesos dos modelos treinados nas imagens AGAR.

O treino inicial atinge uma precisão média (mAP) de 62 % na base de dados AGAR, com Faster R-CNN ResNet 101 a apresentar o melhor desempenho. Um valor de mAP de 66,40 % é alcançado com a aplicação do método de *ensemble Weight Boxes Fusion*, superando os resultados relatados pelos autores da base de dados AGAR. Na nova base de imagens, o modelo RetinaNet ResNet 50 alcança um desempenho inicial de 52,40 %, que aumenta para 56,30 % através da aplicação do método de *ensemble*.

Em resumo, esta dissertação melhora com sucesso o desempenho na base de dados AGAR, introduz um novo conjunto de imagens e emprega várias técnicas para melhoria do desempenho. Este trabalho representa, assim, um passo em frente na aplicação de métodos de *deep learning* no domínio da medicina laboratorial.

Palavras-chave: Microbiologia; Deep Learning; Deteção de Objetos; Colónias Bacterianas.

Contents

List of Figures

List of Tables

Acronyms

AGAR Annotated Germs for Automated Recognition

AI Artificial Intelligence

AP Average Precision

CNN Convolutional Neural Networks

COCO Common Objects in Context

DETR End-to-End Detection Transformer

DIBaS Digital Images of Bacteria Species

DL Deep Learning

IoU Intersection over Union

mAP Mean Average Precision

mAR Mean Average Recall

ML Machine Learning

SVM Support Vector Machines

TSA Trypticase Soy Agar

WBF Weighted Boxes Fusion

YOLO You Only Look Once

Chapter 1

Introduction

This section will provide a brief introductory note regarding the dissertation. The motivation behind selecting this theme, the exposition of the problem's description, and the main objectives will be covered in this section as a review of the existing literature on this subject.

1.1 Motivation

1.1.1 Artificial Intelligence in the Medical Field

The constant technological development that the world has been undergoing in the last few years has affected, in some way, every aspect of our lives. Science has been evolving fast, and the field of medicine is no different. Due to its high impact on people's lives, developing better medical technologies has always been a focus. When it comes to Artificial Intelligence (AI) and Machine Learning (ML), the potential opportunities and applications are groundbreaking (Darcy, Louie, & Roberts, 2016).

ML has been capturing the interest of medical researchers and practitioners in predictive methods within health science and medicine. The number of articles, publications, and overall practical applications in this area has exponentially grown in the last ten years (Cabitza & Banfi, 2018).

In clinical laboratory medicine, it is also expected that ML methods will become more extensively used since the laboratory is the leading supplier of quantitative, structured, and codified data (Cabitza & Banfi, 2018).

The advancements in technology that have been occurring in laboratories have made it possible to incorporate expert system capabilities and software applications, such as auto-analysers and modules of laboratory information systems. Furthermore, incorporating ML methods in medical laboratories should be supported as they can lead to better laboratory organisation and expand laboratory professionals' core skills on a more significant path to change and innovation (Cabitza & Banfi, 2018).

There are multiple examples of the possible applications of ML for data from medical laboratories. ML has been studied for the prediction of diagnosis, risk factors, outcomes, survival prognosis (Camaggi et al., 2010; Salah, Muhsen, Salama, Owaidah, & Hashmi, 2019), for cancer screening (Deist et al., 2018; Ronzio, Cabitza, Barbaro, & Banfi, 2021; Surinova et al., 2015; H.-Y. Wang et al., 2016), and also for the study of the potential tumour and genetic markers (Kourou, Exarchos, Exarchos, Karamouzis, & Fotiadis, 2014), as well as pharmacological targets (Dezső & Ceccarelli, 2020).

Deep Learning (DL) has also been studied and applied in different areas of medicine. The most common and first applied areas were in radiology (McBee et al., 2018), and later also in microscopy histologic images (S. Wang, Yang, Rong, Zhan, & Xiao, 2019).

As for DL in laboratory medicine, DL is, nowadays, a very explored tool in several studies on proteomics (Wen et al., 2020), genomics (Zou et al., 2019), and image classification of, for example, blood cells in hematology (Khouani, El Habib Daho, Mahmoudi, Chikh, & Benzineb, 2020).

Microbiology, as a branch of laboratory medicine, is devoted to analysing samples through various tests to identify and characterise the causal agents responsible for infectious diseases.

Extensive research has been dedicated to exploring the integration of DL within microbiology. Pritt (2020a) have conducted a review of potential applications of AI in the microbiology setting, emphasising its significant role as a powerful tool for the future of this area. The authors allude to several approaches, such as the classification of chromogenic media, the automated microscopic detection of mycobacteria in sputum samples, and the automated detection of malaria parasites in blood smears. Moreover, there is a notable current trend in investigation in exploring the application of DL to study genomic information sourced from isolated bacteria, analyse metagenomic microbial findings from primary specimens, and interpret mass spectra acquired from cultured bacterial isolates (Pritt, 2020a).

Regarding the domain of computer vision within this field, numerous instances also exist. Ferrari, Lombardi, and Signoroni (2017) applied Convolutional Neural Networks (CNN) for a quantitative approach, to count bacterial colonies.

Talo (2019) and Zieliński et al. (2017) proposed an approach employing CNN for the categorisation of bacteria into distinct classes based on digital microscopy images. Another study by Nie, Shank, and Jojic (2015) applied DL techniques for bacterial classification. However, instead of microscopy images, their approach employed a collection of images derived from bacterial cultures grown on agar.

Savardi, Ferrari, and Signoroni (2018) have created a dataset encompassing various bacterial cultures and subsequently applied DL techniques to classify and detect hemolysis, a phenomenon exhibited by certain bacteria when cultivated on blood agar cultures.

Majchrowska, Pawłowski, et al. (2021) has developed an extensive dataset encompassing five distinct bacterial species and yeast. This dataset includes comprehensive photographic documentation and annotations of various bacterial culture plates. The authors then proposed the application of DL for bacteria and yeast classification. Additionally, they developed a framework for microbial objects counting (Graczyk, Pawłowski, Majchrowska, & Golan, 2022). They extended the application of DL in generating synthetic datasets of microbiological images of Petri dishes, which add value for training DL models (Pawłowski, Majchrowska, & Golan, 2022).

An alternative approach by Pritt (2020b) involved utilising CNN in digital microscopy for intestinal parasites. In summary, the researchers developed a model trained to identify the absence of intestinal parasites on slides, concurrently highlighting suspected parasites for subsequent manual verification.

Some of these articles and applications will be reviewed further in the document within the Literature Review section.

1.1.2 Automation in Microbiology Laboratories

Numerous tasks within the microbiology laboratory routine are still performed manually. This is particularly evident when comparing it with the areas of hematology or clinical chemistry. In the latter, automation systems have progressively evolved and gained widespread acceptance, becoming a big part of the daily operational routine (Antonios, Croxatto, & Culbreath, 2021; Bourbeau & Ledeboer, 2013; Novak & Marlowe, 2013; Williams & Trotman, 1969).

In summary, a standard day within a medical microbiology laboratory involves various tasks, from processing samples and the maintenance of cultures to staining procedures, microorganism identification, and antimicrobial tests. While the specific techniques applied may vary depending on the laboratory, the general workflow is similar.

The first step always involves the reception and recording of the sent samples. These samples can be blood, urine, or other bodily fluids specimens obtained from patients. It is important to note that the context being discussed pertains to a medical laboratory specialising in analysing bodily fluids, which could be affiliated with a hospital or may receive samples from diverse centers. It is worth highlighting that microbiological testing extends its scope to numerous other domains, including assessments of water quality, food safety, and industrial applications, which may have different routines and techniques.

Subsequently, samples are prepared for analysis. Different preparation methods are applied based on protocols according to the specific type of sample being examined.

Most of the samples are inoculated onto culture plates containing growth media for analysis. Given that these plates will require an incubation period of at least 12 hours after inoculation, which may extend beyond 24 hours in certain instances, carrying out this step at an early stage is imperative.

Culture media, or growth media, can exist in solid, liquid, or semi-solid forms and are designed to support the growth of microorganism populations. There are different cultural media types, the most common being nutrient broths or agar plates. These culture media all share a common property: providing a nutrient-rich environment that fosters the growth of microorganisms. What makes them different is an array of compositions and properties that confer selective and/or differential attributes. The sample's origin determines the choice of culture media. For instance, respiratory samples are typically subjected to more selective media, as the respiratory tract harbours diverse normal flora present in a healthy individual, becoming pivotal to cultivating solely those microorganisms most likely to be pathogens. Conversely, blood samples, characterised by their physiological sterility, are often cultivated in richer and broader media.

Following these stages, various steps within the daily routine demand significant manual involvement, occurring concurrently or sequentially. These encompass tasks like preparing samples for microscopic assessment and the subsequent microscopic visualisation. Furthermore, the process entails microorganism identification and antimicrobial susceptibility testing, which often involves a range of techniques and may require additional manual inoculation and subsequent rounds of incubation. Ultimately, the conclusive interpretation of all outcomes involves validation, antibiotic selection, and the communication of findings to the medical professional responsible for overseeing the care of the respective patient.

Considering the central theme of this dissertation, it becomes imperative to delve deeper into the steps involving culture media. This elaboration serves a dual purpose: it accentuates the motivation underlying the addressed problem and clarifies the primary objective.

Inoculating and incubating microorganisms on culture plates constitutes a crucial component of microbiology routines. However, it is a process that demands considerable time to be executed meticulously. The primary objective behind incubation is to foster the growth of bacteria into discernible colonies on the plates, which serves as the foundation for subsequent analysis.

To provide an illustrative example, let us consider again a blood sample. A blood sample from a healthy individual is a sterile specimen, meaning that no microorganisms are expected to proliferate on the plate. However, when patients have bloodstream infections, it is expected that colony growth on the plates is manifested. This phenomenon is significant because it confirms the presence of microorganisms in the individual's blood and because these colonies play a crucial role in providing a deeper investigation.

At an initial stage, professionals can obtain valuable and rapid insight from the macroscopic characteristics of the colonies. Observations regarding their shape, size, and texture can often provide information about the group of bacteria present. Moreover, certain tests can be promptly conducted, offering supplementary information in emergencies.

Furthermore, these colonies serve as the foundation for more advanced procedures. Specialised equipment can be utilised to identify the microorganisms at a species level. Additionally, the colonies are instrumental in conducting antibiotic susceptibility testing. This assessment studies the susceptibilities and resistances of the bacteria to a range of antibiotics, providing a comprehensive understanding of the patient's infection and enabling a more suitable therapeutic approach (Savardi et al., 2018).

The daily routine is thus highly manual-intensive to laboratory professionals, further compounded by its time-intensive nature. Completion of medical reports typically takes at least one to two days. This temporal delay can adversely affect the patient's health as initiating effective antimicrobial therapies gets postponed. Consequently, relying on broader-spectrum and empirical therapies is often necessary, resulting in extended treatment duration that can sometimes be inefficient until microbiological results are finalised.

This situation also presents drawbacks in terms of operational efficiency, costs, storage capacity, and processing times within the laboratory setting.

While automated systems for microbiology do exist, they have yet to be widely adopted. Implementing automation in this field can be challenging due to the heterogeneous range of sample types, diversity of specimen processing techniques, and cost-related considerations that laboratories need to factor in, among other factors (Antonios et al., 2021).

Some examples of existing automated systems are inoculation units, robotic incubators, digital photography modules, and post-imaging analysis workstations (Antonios et al., 2021). The full automation of laboratory processes in microbiology has only recently been recognised as a valuable tool, but research is showing its potential benefits. Automation can lead to greater standardisation, improved laboratory efficiency, enhanced workplace safety, and long-term cost savings (Antonios et al., 2021).

In summary, the primary motivation for this dissertation is to address the need for automation and enhanced efficiency in clinical microbiology laboratories. By leveraging the power of AI, this research focuses on applying DL models for classifying microbial colonies grown on agar cultures.

1.2 Problem Description

With the growing exploration and demand for automation in clinical microbiology laboratories, the development of automated microbiological sample analysis based on AI is of great interest.

The evolution of automated image analysis technologies for the detection and/or classification of microbial colonies would, therefore, tackle a time-consuming and error-prone process within the microbiology routine. In great contrast to the manual procedure outlined in the preceding section, one illustrative application of these technologies involves the utilisation of incubators equipped to capture photographs of culture media plates while they reside within the incubator. Subsequently, automated image analysis algorithms can execute computer-assisted culture interpretation, enabling the successful identification and reporting of plates exhibiting none or minimal growth, the recognition of colonies, or even efficiently identifying the microorganisms (Antonios et al., 2021).

This approach would bring many advantages regarding the working routine, considering the high number of plates manually analysed every single day, since many of them sometimes even require a new inoculation and incubation process, thus delaying the report for more hours. Combining an automated incubator that takes photographs of the plates at predefined intervals applying the DL models would bring diverse potential benefits. The models could quickly detect the emergence of colonies in their early stages, identify sterile plates, and categorise plates as either containing colonies from only one species (indicating a high likelihood of it being the infecting pathogen) or mixed species colonies (suggesting contamination and necessitating a fresh sample). Besides, the model could classify the types of colonies, triggering alerts to medical professionals about specific species and delivering many other valuable insights.

With this, the primary objectives of this dissertation are:

- Train and evaluate object detection models for microbial colony detection on a publicly available dataset and on a curated dataset

- Curate a new dataset with annotated images of agar plates with a diverse set of culture media

- Assess the generalization and robustness of deep learning models with transfer learning

- Implement ensemble learning for enhanced microbial colony detection

By conducting this research, a significant contribution is aimed to be made to the progression of DL applications in the field of microbiology, paving the way towards automated analysis and improvement of the efficiency within clinical laboratories.

Ultimately, this work becomes an integral part of the potential transformation in the paradigm concerning the analysis of microbial cultures. By enabling faster and more accurate diagnoses, this project is set to contribute to developing technology that will substantially improve the healthcare system. Importantly, this progression is driven by the overwhelming aim of improving patient care outcomes, underlining the continuing focus of health and technological advancements on benefiting those who matter most – the patients.

Chapter 2

Literature Review

This section will present an overview of the existing literature on DL. Additionally, a theoretical approach to the predominant object detection models will be exposed, including two-stage detectors, like, for example, Region CNN, as well as one-stage models, such as dense (RetinaNet, You Only Look Once (YOLO) and OverFeat), or sparse models (Sparse R-CNN and Deformable End-to-End Detection Transformer (DETR)). This first segment will be followed by an in-depth review of the current applications of DL and object detection models in medical image analysis, specifically, the existing work on their use within the microbiology setting.

2.1 Deep Learning

DL is currently considered a subset within ML, which consists of artificial neural networks composed of three or more layers developed to replicate the behaviour of neurons of the human brain. These multiple hidden layers are made up of interconnected nodes placed between the input and output layers. Each layer represents different levels of abstraction. In summary, the model may start with the raw data, that is, the input, and each layer will apply and learn some non-linear function that will transform this representation to an increasingly abstract level (LeCun, Bengio, & Hinton, 2015). This is the primary differentiating factor between classic ML and DL. While ML requires a manual engineering process and feature extraction to transform the raw data, DL can learn directly from raw input, automating much of the feature extraction process through a general-purpose learning method (LeCun et al., 2015). The neural networks combine data inputs, weight, error, and bias to become accurate. Each layer can be built upon the previous layer so it can learn some aspects and be refined, and it can also be tuned through backpropagation. In this case, errors are calculated, and the weight and biases may be adjusted by returning to previous layers (LeCun et al., 2015).

There are, however, various types of deep neural networks, from CNN, Recurrent Neural Networks, etc. CNN is the most commonly used architecture in computer vision and image classification. When applied to visual recognition and classification, CNN has developed state-of-the-art performance (Krizhevsky, Sutskever, & Hinton, 2017; Rawat & Wang, 2017).

2.2 Convolutional Neural Networks

As already mentioned, CNN have been considered the state-of-the-art for computer vision and image processing problems, such as image classification and segmentation, object detection, and video processing (Khan, Sohail, Zahoora, & Qureshi, 2020). These neural networks have been applied for visual exercises since the late 1980s (Rawat & Wang, 2017). However, it was not until Krizhevsky et al. (2017) won a Visual Recognition Competition by classifying around 1.2 million images into 1000 classes that CNN started dominating the visual classification field (Khan et al., 2020; Rawat & Wang, 2017).

CNN are feedforward networks and, just like the typical artificial neural networks, were biologically inspired by the visual cortex in brains, which are composed of alternating layers of simple and complex layers of cells (Rawat & Wang, 2017). The basic architecture of a CNN, proposed by LeCun et al. (2015), consists of three different types of layers and is structured as a series of stages. These layers are convolutional, pooling or subsampling, and fully-connected layers (Gu et al., 2018). Figure 2.1 shows a representation of a CNN pipeline.

Convolutional Layers

The convolutional layer is the main component of a CNN, and its main objective is to detect local groups of features from previous layers. The input is presented in multiple arrays containing pixel intensities when the input is an image. With this information, a convolutional layer will try to learn feature representations from the input, applying several convolution operations, or kernels, upon the input (LeCun et al., 2015). In a convolutional layer, units are organized into feature maps. Each unit is connected to small regions in the feature maps of the preceding layer through a set of weights (LeCun et al., 2015). The different kernels within each layer will then be used to compute different feature maps, and the result of the locally weighted sum will be passed through non-linear functions (Gu et al., 2018; LeCun et al., 2015). The kernel is applied to various submatrices of the input iteratively. The size of the kernel and the number of positions that should be shifted after each iteration are all parameterized. All this process will promote the detection of interesting features on the input, with local groups of values that are easily detected and distinctive because they are often highly correlated. It will also grant invariance to location and increase efficiency while reducing the number of parameters with a weight-sharing mechanism, granting that a local motive can appear anywhere within the full image (Guo et al., 2023; LeCun et al., 2015).

Pooling Layers

The pooling layers, generally placed between two convolutional layers, operate by grouping similar features into one, reducing the dimensions of feature maps and network parameters. This provides them with the ability to achieve spatial invariance because their computations consider neighbouring pixels, decreasing overfitting, and decreasing further computational requirements (Gu et al., 2018; Khan et al., 2020; LeCun et al., 2015; Rawat & Wang, 2017). There are various ways that the layers can achieve this. The most common methods are average pooling and max pooling aggregation layers. Max pooling is often used to extract the most intense features, like edges, while average pooling works with a smoother approach (Boureau, Ponce, & Lecun, 2010). There are, however, more types of pooling layers, depending on the final task, such as the overlapping, spatial pyramid, and stochastic pooling (Gu et al., 2018; Guo et al., 2023)

Fully-Connected Layers

Several stages of convolution, non-linearity, and pooling layers are stacked to extract more abstract feature representations throughout the network, followed by one or more fully connected layers (LeCun et al., 2015; Rawat & Wang, 2017). These layers connect all the neurons on the previous layer to process global information. They behave like a traditional neural network, containing the majority of the parameters of a CNN. They can feed forward the network into a vector that will then be forwarded for image classification or processing (Guo et al., 2023). For classification problems, the softmax operator is commonly used. However, depending on the final objective, it is possible to use different methods, such as Support Vector Machines (SVM) combined with CNN (Rawat & Wang, 2017). The fully connected layers require big computational power during training, thus being considered a disadvantage of this architecture (Guo et al., 2023).

Activation Function

The activation function can learn multiple patterns. Here, the output of a convolution is multiplied by the activation function, which adds non-linearity and creates a transformed output (Khan et al., 2020). Typical activation functions are the sigmoid, tanh, and rectified linear units (ReLU). The last one is the most common today, having fast convergence and not suffering from the vanishing gradient problem. However, several variations of this function exist that try to achieve faster and better performance. Some examples are Leaky, Parametric, Randomized ReLU, and Exponential Linear Unit (Rawat & Wang, 2017).

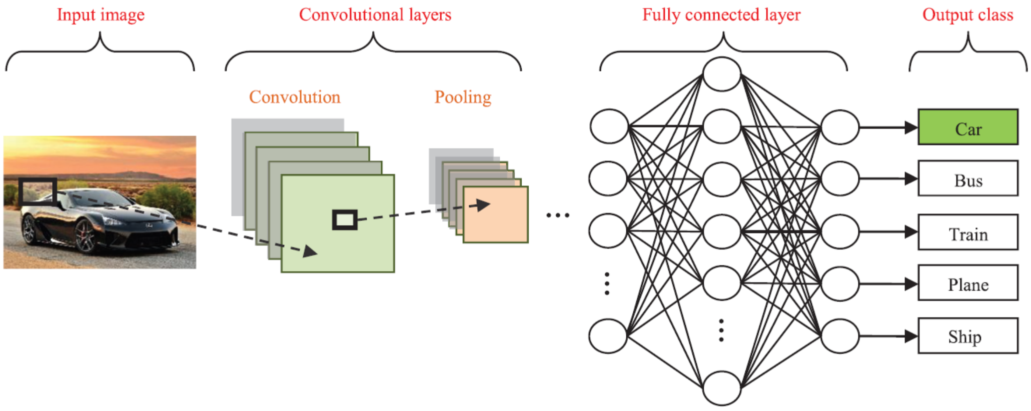

Figure 2.1: CNN pipeline

Training

Training the network consists of a global optimization problem. Similar to a classical neural network, it is performed with a backpropagation algorithm by calculating the gradient vector of a loss function according to weights and biases, with further adjustment of parameters (Rawat & Wang, 2017).

2.3 Object Detection and Recognition

As referred before, the rise in popularity and success of CNN came after the work of Krizhevsky et al. (2017) on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), when millions of labelled images were used to train a large CNN. Similarly to the ILSVRC, the PASCAL Visual Object Classes (VOC) challenge is a benchmark in visual object category recognition and detection (Everingham, Gool, Williams, Winn, & Zisserman, 2010). Object detection requires the solution of two main tasks: recognition, and localization. First, the detector needs to distinguish objects from the background and classify them, and second, it needs to locate and draw bounding boxes to each object (Sun et al., 2020).

While image classification aims to classify full images in categories (for example, correctly classifying pictures of cats and dogs), visual recognition tries to answer slightly different questions, as such:

- Classification - Identify if an object is present in an image;
- Detection - Locate the object in the image;
- Pixel-level Segmentation - Each pixel is assigned a class label;

- Person layout - Identify the location of body parts in an image (as a practical example)

(Everingham et al., 2010)

The former object detection approaches did not rely on CNN (R. B. Girshick, Donahue, Darrell, & Malik, 2013). Some first approaches were based on Histograms of Oriented Gradients (Dalal & Triggs, 2005) and Distinctive Image Features from Scale-Invariant (Lowe, 2004) (X. Wu, Sahoo, & Hoi, 2019). Over the years, several studies applied CNN for these tasks. However, their accuracy on small datasets, while decent, was not record-breaking. Many authors proposed CNN with a sliding window over multiple scales; some researchers have suggested training CNNs to directly predict the specific parameters of objects that need to be located, such as their position relative to the viewing window or their pose; other authors have proposed to use CNN-based segmentation for object localization, with the simplest approach being to train the CNN to classify the central pixel of its viewing window as a boundary or not. However, this approach required dense pixel-level labels for training (Sermanet et al., 2013).

Larger datasets, such as the well-known ImageNet, have enabled CNN to significantly develop the state of the art on object detection and recognition (R. B. Girshick et al., 2013; Sermanet et al., 2013). The work of Sermanet et al. (2013) was then the first to publish a clear explanation of how CNN can be used for localization and detecting ImageNet data. In their work, they provided some new insight into their competition-winning framework:

- Implementing a multi-scale and sliding window approach within a CNN efficiently;
- Presenting a new DL approach for localization by training the model to predict object boundaries;
- Demonstrating that multiple tasks can be learned simultaneously using a single shared network;
- Publishing a feature extractor called OverFeat derived from their best model.

A short time after, R. B. Girshick et al. (2013) published a paper that proposed a new algorithm with higher precision than the previous best result on VOC 2012 and better results than the work of Sermanet et al. (2013). They called their method R-CNN, as in ”Regions with CNN features” (R. B. Girshick et al., 2013). Their approach consisted of a combination of two key elements:

- The use of high-capacity CNNs on bottom-up region proposals for object localization and segmentation;

- Using supervised pre-training for an auxiliary task, followed by fine-tuning for a specific domain, resulted in a significant performance improvement when labelled training data is scarce.

In this section, a summary of the evolution of object detection algorithms over the last few years will be presented. The different approaches to these tasks will be described, from the dense, dense-to-sparse, and sparse methods, as well as some examples of algorithms within each method. The description and focus on the application of CNN for this task and how R-CNN became a solid base with lots of development in the last few years.

2.3.1 Dense Method

As previously mentioned, one approach to object detection is using sliding windows over scales. This method was popular for some years, but its performance reached a limit due to the limitations of traditional feature extraction techniques (Sun et al., 2020). With the advent of CNNs in this field, performance improved. The most common pipelines consisted of one-stage detectors that relied on dense candidates, each directly classified and regressed. These pipelines have some drawbacks, such as the production of redundant results and the requirement for non-maximum suppression post-processing. The final performance was also highly affected by the size, aspect ratio, number of anchor boxes, the density of reference points, and proposal generation algorithms (Sun et al., 2020).

OverFeat, YOLO and RetinaNet are examples of dense methods.

Overfeat

Sermanet et al. (2013) developed a multi-scale, sliding window approach that can be used for classification, localization, and detection. The authors proposed a CNN trained to simultaneously classify, locate, and detect objects in images to improve classification, localization, and detection accuracy. In their work, Sermanet et al. (2013) introduced a new method that accumulates predicted bounding boxes instead of suppressing them, claiming that detection can be performed without training on background samples, avoiding complicated and time-consuming bootstrapping training methods, while increasing detection confidence (Sermanet et al., 2013). A new state of the art for the detection task was defined. Finally, Sermanet et al. (2013) released a feature extractor of the best model called OverFeat. This was one of the first modern one-stage object detectors based on deep networks (Lin, Goyal, Girshick, He, & Dollár, 2017).

RetinaNet

The work of Lin et al. (2017) focused on developing a single-stage model that would be as accurate as the two-stage detectors that defined the state of the art at the time. For this, the authors identified the main obstacle of one-stage detectors as the class imbalance present during training. They proposed a different loss function that would tackle this issue. A new loss function called Focal Loss was introduced, addressing the issue of class imbalance. The loss function dynamically scales the cross-entropy loss, reducing the impact of easy examples and emphasizing hard examples during training. This outperformed previous techniques, such as sampling heuristics and hard example mining for training one-stage detectors. The specific form of the focal loss is not critical, as alternative instantiations yield similar results. To demonstrate its effectiveness, the authors presented RetinaNet, a one-stage object detector that utilizes focal loss along with an efficient pyramid of features in the network and anchor boxes, achieving high accuracy and surpassing the results of both one- and two-stage detectors (Lin et al., 2017). RetinaNet constitutes an integrated neural network architecture comprising a backbone network and two specialized subnetworks. The backbone produces a convolutional feature map encompassing the entire input image based on the Feature Pyramid Network. The first subnetwork focuses on object classification, while the second is responsible for performing bounding box regression for their prediction. These subnetworks are specifically optimized for efficient and comprehensive object detection in a single stage.

YOLO

Redmon, Divvala, Girshick, and Farhadi (2015) approached the object detection task as a single regression problem, directly predicting the image pixels' bounding box coordinates and class probabilities. It uses a single CNN that simultaneously predicts multiple bounding boxes and class probabilities for those boxes, trained on full images and directly optimizing detection performance (Redmon et al., 2015). Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and encodes contextual information about classes and their appearance. Initially, YOLO lacked accuracy but represented a fast object detection algorithm.

This method suffered several improvements over time. The main improvements on YOLO network were the addition of several layers and steps. The original framework added a base grid division; in future models, an anchor with K-means was added, along with a two-stage training and full CNN, as well as the addition of a multi-scale detection (Jiang, Ergu, Liu, Cai, & Ma, 2022).

2.3.2 Dense-to-Sparse Method

Dense-to-sparse methods are two-stage detectors that have dominated object detection tasks for years. The process begins by identifying a small (sparse) number of boxes that likely contain foreground objects from a larger (dense) set of potential regions and then improves the placement of these boxes to locate the objects more accurately. They have a region proposal algorithm (R. B. Girshick et al., 2013; Ren, He, Girshick, & Sun, 2015), and as well as dense methods, also need non-maximum suppression post-processing (Sun et al., 2020).

The main approach of R. B. Girshick et al. (2013) was trying to fill the gap between image classification and object detection. Object detection requires the localization of likely many objects within an image, unlike image classification (R. B. Girshick et al., 2013). For their work, they considered taking localization task as a regression problem, but other authors already stated its inefficiency or adopting a sliding window approach. However, this would be a technical challenge due to having more convolutional layers (R. B. Girshick et al., 2013). They instead approached the task with regional proposals. The algorithm consists of three modules:

1. Generating a set of potential region proposals that will define candidate detections available for the detector;

2. Fixed-length feature extraction from these regions, using a large CNN;

3. A set of class-specific linear SVM to classify and locate the objects within the region.

With this, several studies came after that tried to develop R-CNN for object detection, with better performance and higher speeds. R. Girshick (2015) denotes some drawbacks from the baseline R-CNN, such as being slow due to performing a CNN forward pass for each object proposal without sharing computation, thus training to become computationally expensive and having a greedy proposal search.

R-CNN development

With the publication of the first R-CNN article, several authors followed by developing new methods upon the baseline algorithm to increase performance and speed and to tackle the bigger drawbacks. This topic presented a summary of each advancement and development that was made until the present day.

Fast R-*CNN*

Fast R-CNN brought some advantages compared to the baseline R-CNN. With higher detection quality, less computation is required than in single-stage training. The Fast R-CNN network inputs an entire image and a set of object proposals as input. Firstly, the whole image is processed with several layers, and then, a fixed-length feature vector is extracted from the feature map by a region of interest pooling layer. Each feature vector is fed into a sequence of fully connected layers that will branch into two output layers, one producing softmax probability estimates over determinate object classes and another layer that outputs four real-valued numbers for each of the object classes that will serve to encode refined bounding-box positions for one of the various classes (R. Girshick, 2015). One of the main advantages of Fast R-CNN is that it uses a single-stage training process and can achieve almost real-time rates when not considering the time required for proposal generation (Ren et al., 2015).

Faster R-CNN

The work of Ren et al. (2015) introduced the Region Proposal Network (RPN) as an addition to the Fast R-CNN object detection algorithm. The RPN is a fully convolutional network that predicts object bounds and scores at each position while sharing full-image convolutional features with the detection network. This allows for cost-effective region proposals and improves the accuracy of object detection. The authors trained the RPN on the previous Fast R-CNN. They achieved state-of-the-art object accuracy on several datasets with a limited number of proposals (Ren et al., 2015).

Mask R-CNN

With Mask R-CNN, the authors worked on Faster R-CNN to create a flexible framework for object instance segmentation. In this work, He, Gkioxari, Dollár, and Girshick (2017) developed a method that detected objects while simultaneously generating a high-quality segmentation mask for each instance. They added a new branch to the network responsible for predicting a binary mask for each object instance, in addition to the existing branches for object detection and bounding box regression. This new approach improved the performance of object instance segmentation and outperformed all existing single-model tasks in this area (He et al., 2017).

Cascade R-CNN

Cai and Vasconcelos (2017) developed the Cascade R-CNN framework, an extension of the two-stage R-CNN framework for object detection. They addressed two issues with the traditional R-CNN frameworks: the Intersection over Union (IoU) threshold for defining negatives and positives and overfitting due to a lack of positive samples. In summary, a low threshold will produce noisy detections, and an increasing threshold may degrade detection performance, and overfitting usually happens since there are exponentially vanishing positive samples. The Cascade R-CNN framework consists of a sequence of detectors trained with increasing IoU thresholds, which are more selective against close false positives. Additionally, the framework uses a sequential resampling technique to reduce overfitting by ensuring all detectors have an equivalent size set of positive examples (Cai & Vasconcelos, 2017).

2.3.3 Sparse Method

While both previous methods relied upon dense object candidates, such as having a predetermined number of anchor boxes on every grid of the image feature map as the first stage, sparse methods aim to eliminate those types of dense candidate designs. Sun et al. (2020) and Zhu et al. (2020) have developed full sparse object detection pipelines that fully removed the need for the hundreds of thousands of hand-designed object candidates, achieving exceptional performance and accuracy.

Sparse R-CNN

The authors Sun et al. (2020) proposed a new approach to the object detection task called Sparse R-CNN. They believe that the sparse property should exist in two places: sparse boxes, meaning that a small number of starting boxes is enough to predict every object in an image, and sparse features, meaning that the feature of each box does not need to interact with every other feature over the image.

Their method aimed to eliminate the need for thousands of candidates by choosing object candidates with a fixed small set of learnable bounding boxes instead of predicted ones from the Region Proposal Network from Faster R-CNN. They also introduced two new concepts that followed the sparse ideals: proposal feature, consisting of a high-dimension latent vector that is expected to encode rich instance characteristics and generate customized parameters for recognition, and proposal boxes, which are randomly initialized and optimized along with the other parameters in the network (Sun et al., 2020). Sparse R-CNN is a full sparse method in which the initial input is a sparse set of proposal boxes and features, altogether with the one-to-one dynamic instance interaction, eliminating the need for dense candidates and the global feature interaction in the pipeline (Sun et al., 2020).

Deformable DETR

Carion et al. (2020) proposed DETR, which aims to eliminate the need for hand-designed components in object detection tasks while maintaining high performance. DETR combines

CNN and Transformer encoder-decoders to replace hand-crafted rules. However, DETR had some issues, such as long training times, being slower than Faster R-CNN, and having low performance detecting small objects (Zhu et al., 2020). To address these issues, Zhu et al. (2020) proposed Deformable DETR, which combines the sparse sampling of deformable convolution with the relation modelling capability of Transformers. A deformable attention module attends to a small set of sampling locations as a pre-filter for prominent key elements. It can be extended to multi-scale aggregation without requiring Feature Pyramidal Networks. An iterative bounding box refinement mechanism was used to improve detection performance, as well as a two-stage Deformable DETR, where a variant of Deformable DETR generates the region proposals, and then fed into the decoder for iterative bounding box refinement (Zhu et al., 2020).

2.4 Deep Learning in Medical Microbiology

In this section, the existing literature on this specific topic will be discussed. There are few studies regarding using DL in microbiology, even less for colony detection and classification. Some studies approached the issue by developing algorithms to classify the colonies, while others tried to create methods for colony counting. Other authors applied DL methods for image generation of synthetic colonies and plates upon an existing dataset. Although not upon culture media colonies, some authors researched microscopic images of microorganisms as a means to enhance image recognition in the microbiology area.

Majchrowska, Pawłowski, et al. (2021) developed a whole dataset on images of microbial cultures cultured on an agar plate. Named the Annotated Germs for Automated Recognition (AGAR), this dataset comprises around 18 thousand photos of five different microorganisms. The main objective of the authors was to create and publish a dataset that would serve for the future development of ML models for the microbiology field.

Majchrowska, Pawłowski, et al. (2021) evaluated the performance of the AGAR dataset for building DL models for image-based microorganism recognition by testing two architectures for object detection (Faster R-CNN and Cascade R-CNN) with four different backbones (ResNet-50, ResNet-101, ResNeXt-101, and HRNet). They found that the AGAR dataset can be used to build robust models and is well suited for real data collected in various acquisition setups. The best-performing model was the Cascade R-CNN with the HRNet backbone, with a counting error of 4.92 % on the higher-resolution subset and 3.81 % on the lower-resolution subset. For detection, the mean average precision (mAP) scores ranged from 49.3 % to 59.4 % for different detectors.

On a different paper, Majchrowska, Pawlowski, et al. (2021) evaluated the performance of various object detection methods, that included Faster R-CNN, Cascade R-CNN, Libra R-CNN, CBNetv2, YOLOv4, EfficientDet-D2, and Deformable DETR with different backbones (ResNet-50, CSPDarknet53, EfficientNet-B2, and XCiT-T12) on the higher resolution subset of the AGAR dataset. The authors found that the results did not vary greatly between the different architectures, with mAP values ranging from 0.49 % to 0.53 %. Regarding accuracy and speed, YOLOv4 performed the best, while two-stage architectures performed moderately. Transformer-based architectures achieved the worst results.

Pawłowski et al. (2022) developed a strategy to generate an annotated synthetic dataset of microbiological images of Petri dishes using the AGAR dataset. This dataset can train DL models in a fully supervised fashion, utilizing traditional computer vision methods and neural style transfer techniques.

Graczyk et al. (2022) also used the AGAR dataset to study a density map approach for colony counting. They proposed a self-normalization module in the network called a Self-Normalized Density Map, which improves the model's accuracy by correcting the output density map. However, the efficiency of this approach was similar to that of detector-based models such as Faster R-CNN and Cascade R-CNN.

Ferrari et al. (2017) have also created a dataset of pictures of blood agar plates with hemolytic and non-hemolytic colonies taken under different lighting conditions. The team then used this dataset to test two different ML methods for counting colonies. The first method involved extracting a set of hand-crafted morphometric and radiometric features and using them in a SVM. The second method involved using a CNN. The team also tested different ways of enhancing the dataset to improve performance. They found that the CNN approach performed better than the hand-crafted method.

Savardi et al. (2018) developed a method to detect and classify diagnostically relevant hemolysis effects associated with specific bacteria growing on blood agar plates. The authors used feature evaluation and SVM classification to detect and distinguish between different types of hemolysis on both a single colony and whole plate setups. They reported good results but highlighted that different lighting conditions and plate alignment were major challenges for hemolysis detection.

In the study conducted by Nie et al. (2015), colony classification was approached using unsupervised methods. The goal was to segment and classify bacterial images across different growth phases and environmental contexts. The authors employed Convolutional Deep Belief Networks to provide a deep representation of small image patches. Afterwards, a SVM was trained to classify foreground and background patches accurately. Once the foreground patches were identified, a supervised CNN was trained to predict which bacterial colonies from the pool occurred in a query image. These predictions were then aggregated through a voting scheme to predict the likely species in the image. As a result of this study, the authors concluded that this method outperformed the more commonly used image segmentation and classification methods.

Liu, Huang, Liu, Lin, and Zou (2022) and Whipp and Dong (2022) explored various methods for colony counting to improve the task.

In their study, Liu et al. (2022) aimed to address the problem of limited labelled data and poor performance on new data sources in the colony counting task. They proposed a blending-based augmentation strategy to increase the amount of data and incorporate multiple targets within a single sample to enhance learning complexity. They used a two-stage framework, where data from multiple sources was first processed through a CNN for feature extraction and then trained in a second stage. The authors found that their approach outperformed other methods, such as OpenCFU, TLCC, Mask R-CNN, and CenterNet, due to the novel augmentation technique.

Whipp and Dong (2022) created and evaluated several DL models for automatic microbial colony counting using the YOLO framework, specifically using images of *S. aureus* from the AGAR dataset. They compared different versions of the YOLOV5 model and evaluated the effect of varying image resolutions. They found that more complex models did not improve performance significantly but significantly increased the time required for training.

Andreini, Bonechi, Bianchini, Mecocci, and Scarselli (2020) addressed the challenge of limited data availability by developing image-generation models. They used a CNN to separate colonies from backgrounds and to overcome the lack of annotated images. They designed a generative adversarial network to generate synthetic data that captures the typical distribution of bacterial colonies on agar plates. They then superimposed these generated colony patches on existing background images, considering the background's local appearance and the colonies' opacity. They used a style transfer algorithm to improve visual realism.

Different from the focus on colonies on culture media, the studies by Talo (2019) and Zieliński et al. (2017) used DL techniques on digital microscopy images.

Zieliński et al. (2017) created the Digital Images of Bacteria Species (DIBaS) dataset, a collection of microscopic images that includes 33 different bacteria species, each represented by 20 images. These samples were stained using the Gram method and captured with a 100x objective under oil immersion. They then applied Dense SIFT and CNN techniques to classify bacteria species using this dataset.

Talo (2019) applied a pre-trained ResNet-50 CNN architecture to classify digital bacteria images into 33 categories using the DIBaS dataset. They used a transfer learning technique to speed up the training process of the network and enhance its classification performance. The proposed method achieved an average classification accuracy of 99.2 %.

This dissertation seeks to advance the application of neural networks for the classification of bacterial colonies through various approaches. These approaches include improving the models' performance on the AGAR dataset by employing techniques like transfer learning and ensemble methods. Additionally, the dissertation aims to gain a deeper understanding of how different background types impact model performance. This is accomplished by dividing the dataset into subsets based on background characteristics and training the models accordingly.

Secondly, this project includes the creation of a small dataset consisting of annotated images of agar plates with bacterial colonies, encompassing a diversity of culture media. Existing datasets primarily focus on a single type of culture media. For instance, the AGAR dataset features plates with Trypticase Soy Agar (TSA), while the dataset in Savardi et al. (2018) centers on colonies inoculated on plates with blood Agar. As such, this project aims to curate a dataset that encompasses plates with a diverse array of culture media, including blood agar, chocolate agar, MacConkey agar, and Mannitol agar. Besides its creation, the main objective is to utilize this dataset for training and subsequently enhancing the performance of the models. This enhancement will be achieved through the application of the aforementioned methods, including transfer learning and ensemble techniques.

Chapter 3

Methodology

This chapter delineates the methodologies adopted for this project, providing an exposition on data description, training workflow, and the chosen ensemble method. Additionally, it offers insight into the rationale behind incorporating transfer learning and the metrics employed for evaluation.

The dissertation is divided into two parts.

Part 1: Training on the AGAR Dataset

In this part, the primary aim is to comprehensively assess accuracy across each background type within the dataset and implement transfer learning and ensemble methods to enhance accuracy.

Part 2: Training on the curated dataset

In the second segment, a dataset was curated by capturing photographs of real plates within the laboratory setting. While the bacterial species remained consistent with those utilized in the first phase, there was a variation in the culture media of the plates on which they were inoculated. Initially, a foundational training process establishes a baseline accuracy level. Subsequently, to enhance performance, transfer learning was applied to leverage the insights gained from the models trained on the AGAR dataset, and ensemble methods were implemented.

This project is focused on answering some key questions, such as:

- Which models are better suited for this task?
- How do different background types influence the performance of the models on the AGAR dataset?

- Are there variations in model performance across different bacterial species within both datasets?

- What is the impact of transfer learning on model performance?

- How does the generalization process, where AGAR dataset weights are applied using transfer learning, affect the training of the curated dataset?

- Which combination of models in the ensemble method produces the best results in both datasets?

3.1 Data Description

3.1.1 AGAR dataset

The first part of this project was developed upon the AGAR dataset that was provided by Majchrowska, Pawłowski, et al. (2021). The authors' main objective for creating this dataset was to have a diverse dataset upon which the broader research community could build and advance the field of neural networks in microbiology. The AGAR dataset comprises 18 thousand annotated photographs of Petri dishes, with over 330 thousand labelled microbial colonies. It includes colonies belonging to four different bacterial species (*Staphylococcus aureus*, *Bacillus subtilis*, *Pseudomonas aeruginosa*, *Escherichia coli*), alongside one yeast strain (*Candida albicans*) (Majchrowska, Pawłowski, et al., 2021). These photographs are categorized into two major groups: high-resolution and low-resolution images. The first group is further divided into three subgroups based on lightning conditions: bright, dark, and vague. The distinction between these subgroups is derived from the colour of the plexiglass employed: white for the bright and black for the dark subgroups. The vague subgroup was exposed to ambient lighting, giving away to low-contrast images (Majchrowska, Pawłowski, et al., 2021). Figure 3.1 illustrates the various background settings for this dataset.

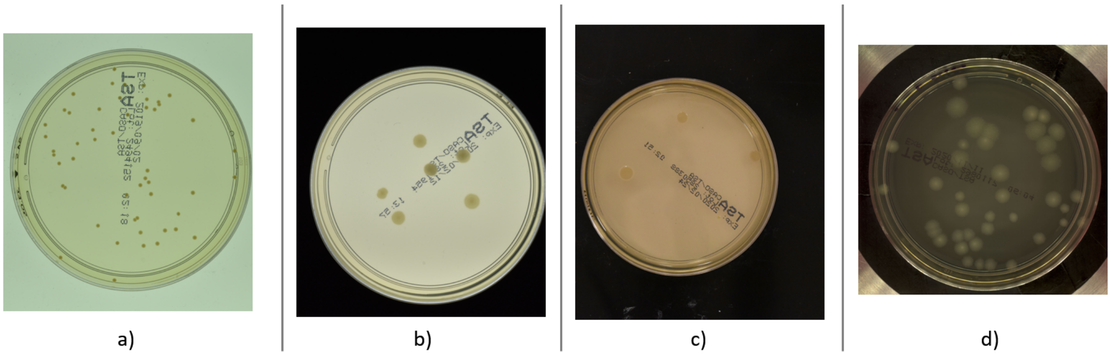

Figure 3.1: Background settings: a) Bright, b) Dark, c) Vague, d) Low Resolution

The dataset comprises images featuring countable colonies, images depicting empty plates, and images showcasing uncountable colonies, where the enumeration of individual colonies is unfeasible.

Figure 3.2 illustrates the distribution of empty, countable, and uncountable plates across various background types and species-specific annotation counts. These graphs are taken from Majchrowska, Pawłowski, et al. (2021).

For this project, only images containing colonies of *S. aureus*, *P. aeruginosa*, and *E. coli* were utilized. This selection was primarily driven by the greater balance observed across the

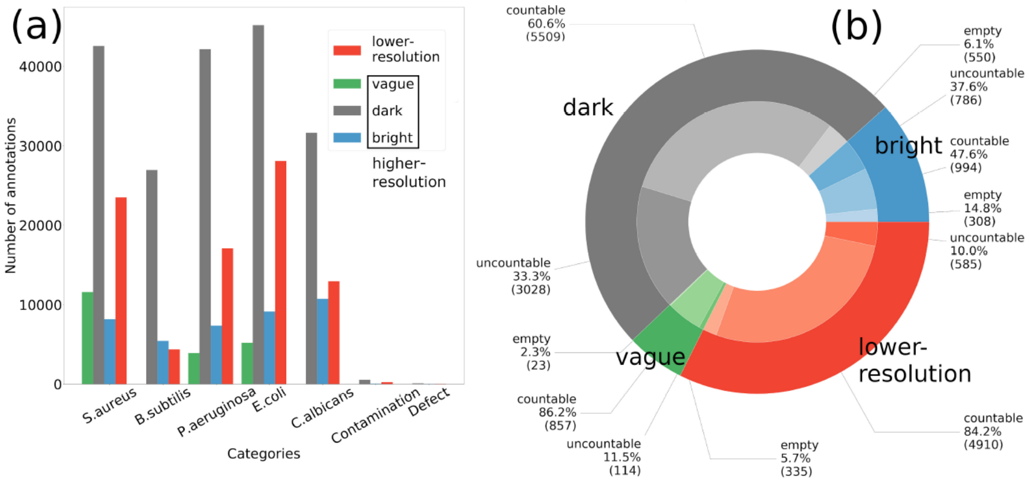

Figure 3.2: Samples distribution in the original dataset: (a) across different microbial species, and (b) among different acquisition setup subgroups.

dataset within these categories but also due to the higher clinical significance those three bacteria represent as prevalent pathogenic agents in human infections. Consequently, from the complete dataset, images that featured annotations of *B. subtilis*, *C. albicans*, ”Contamination,” or ”Defect” were excluded. Furthermore, images containing uncountable colonies were also removed, as well as images that contained more than 100 annotations, for simplification.

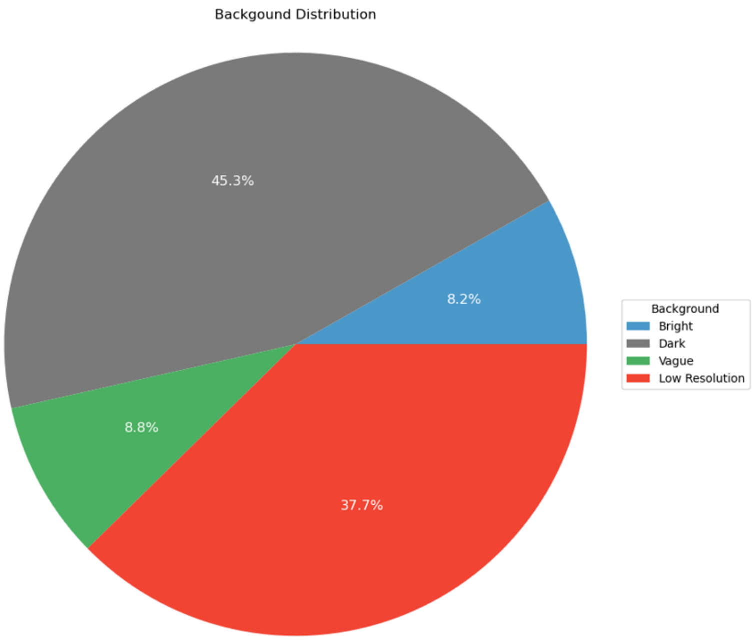

Figure 3.3: distribution of the dataset according to the background type.

With all this, the resultant dataset employed in the project comprises a total of 9851 images, 182864 annotations, and three categories (*S. aureus*, *P. aeruginosa* and *E. coli*).

Out of these, 1217 images featured empty plates, while the remaining images included annotations spanning the three categories. Each image had the potential to encompass one or more of these categories. The distribution of empty and countable plates is illustrated in the Appendix, in Figure A.1.

Approximately 62 % of the dataset comprised high-resolution images, predominantly concentrated within the dark background category. This distribution is visually presented in more detail in Figure 3.3.

The relative frequency of each class across the entire dataset and within each background subgroup was computed, illustrated in Figures 3.4 and 3.5, respectively. The entire dataset is relatively well-balanced. Class *P. aeruginosa* makes up the fewest annotations, accounting for approximately 27 % of the total. Within each subgroup, dark background images display the highest degree of balance. Conversely, the most imbalanced subgroup is the vague background. The distribution of class instances is relatively consistent, except for the vague background subgroup, where pronounced discrepancies are observed.

Finally, concerning the distribution of classes and annotations, Figures A.2 and A.3 graphically depict the number of images based on the number of annotations they contain for the entire dataset and across each subgroup. The majority of images contain twenty annotations or less, while only a few have more than fifty annotations.

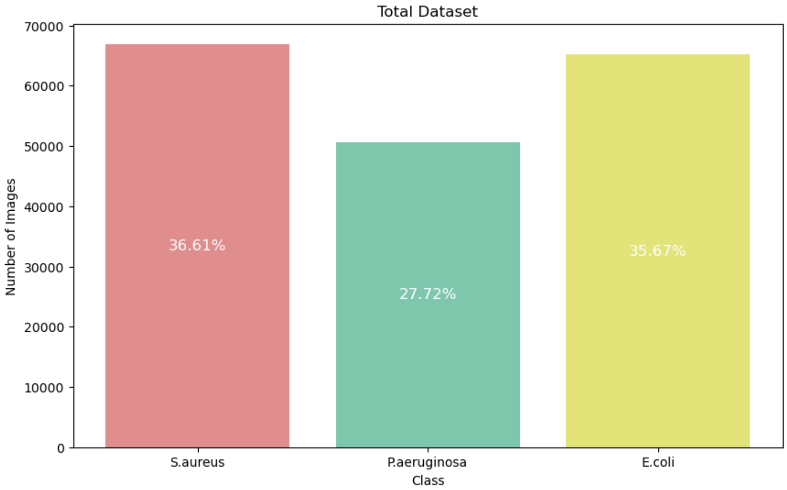

Figure 3.4: Distribution of classes across the entire dataset.

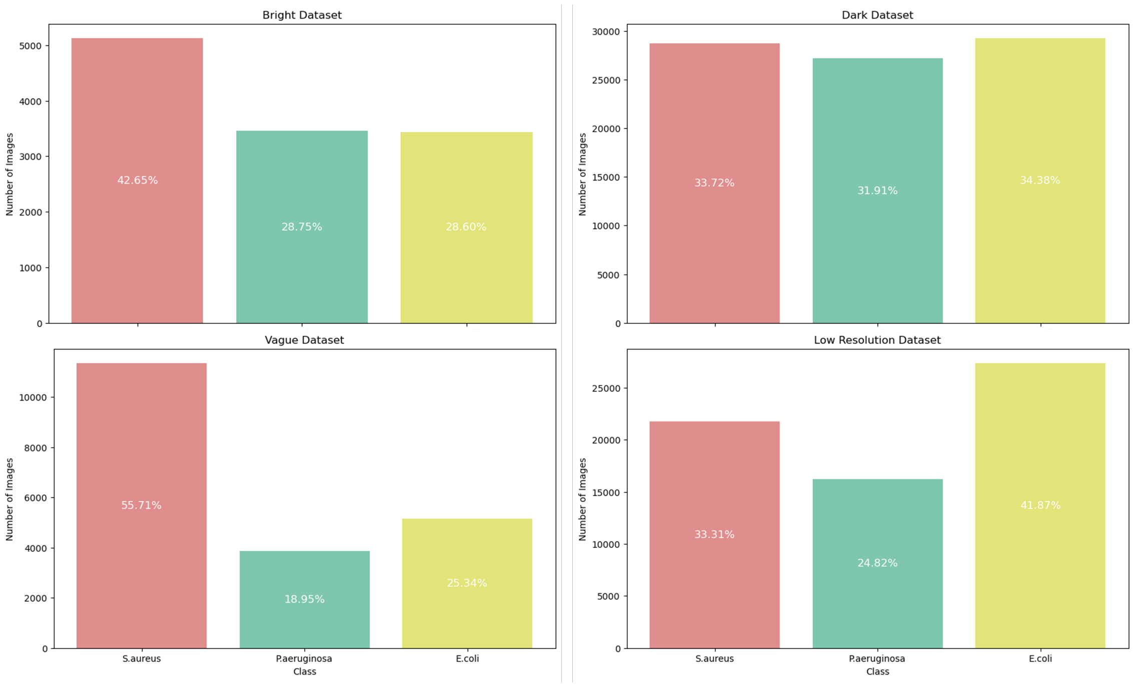

Figure 3.5: Distribution of classes within each background.

3.1.2 New dataset

The second segment of the dissertation started with creating a compact dataset of annotated images by utilizing colonies inoculated on laboratory plates. In order to sustain the continuity of the classification task, the bacterial species inoculated remained consistent with those addressed in the first part, namely *S. aureus*, *P. aeruginosa*, and *E. coli*. Conversely, the plates employed for inoculation diverged from those utilized in the AGAR dataset. The authors opted for plates containing TSA culture media, a nutrient-rich but non-selective and non-differential medium. In this particular dataset, the decision was made to introduce greater diversity in culture media, involving four distinct types of agar: blood agar (Figure 3.6 a), and chocolate agar (Figure 3.6 b) (both enriched and non-selective), MacConkey agar (Figure 3.6 c) (a selective and differentiating medium that exclusively supports the growth of gram-negative bacterial species – such as *E. coli* and *P. aeruginosa*), and Mannitol salt agar (Figure 3.6 d) (a selective and differential medium used to isolate and identify *S. aureus*). The distribution of the type of culture media on the dataset is illustrated on Figure A.4

The various culture media exhibit distinct colours, and the bacteria react differently based on the specific medium in which they were inoculated. Furthermore, even within the same species, there can be slight variations in morphology due to the type of medium utilized. This introduces a layer of diversity and complexity, establishing a dataset to simulate a medical laboratory scenario more faithfully.

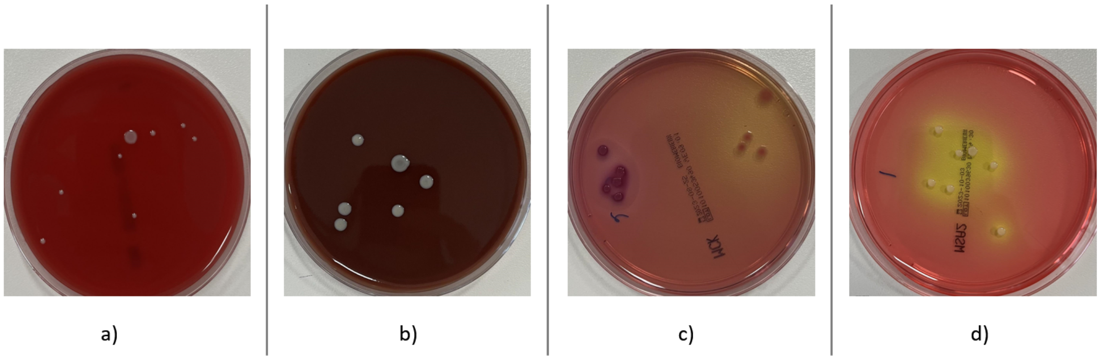

Figure 3.6: Different culture media: a) Blood agar, b) Chocolate agar, c) MacConkey agar, d) Mannitol salt agar.

Figures 3.7 to 3.9 are dedicated to illustrating the different morphologic characteristics exhibited by colonies of different species when inoculated in diverse culture media. Notably, images a) within each figure pertains examples to the AGAR dataset, while the remaining images are from the new dataset.

Colonies of *S. aureus* are typically circular, smooth, raised, and exhibit a glistening appearance. When cultivated on Mannitol salt agar (Figure 3.7 c and d), they manifest an opaque texture and often display a golden-yellow pigmentation. Conversely, TSA (a) and blood agar (b) are characterized by an opaque, greyish-to-yellow hue.

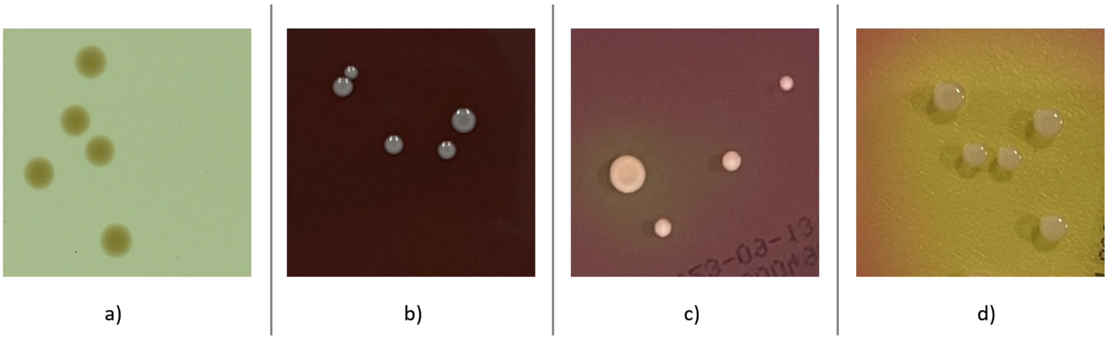

Figure 3.7: *S. aureus* colonies: a) AGAR dataset, b) Blood agar, c) Mannitol salt agar, d) Mannitol salt agar.

E. coli colonies are large, circular, and possess a grey appearance on TSA, blood, and chocolate agar (Figure 3.8 a, c, and d). When cultivated on MacConkey agar (b), they acquire a pink colour due to lactose fermentation.

P. aeruginosa colonies are larger in size, circular, flat, and have irregular margins. On blood and chocolate agar (Figure 3.9 b and d), they showcase a greyish pigmentation, whereas on TSA (a), they appear paler. In the case of MacConkey agar (c), the colonies are colourless, as

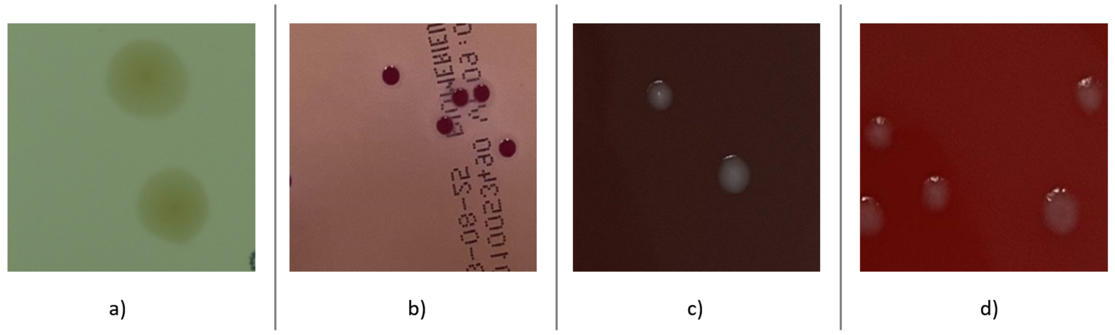

Figure 3.8: *E. coli* colonies: a) AGAR dataset, b) MacConkey agar, c) Chocolate agar, d) Blood agar.

they do not undergo lactose fermentation, unlike *E. coli*.

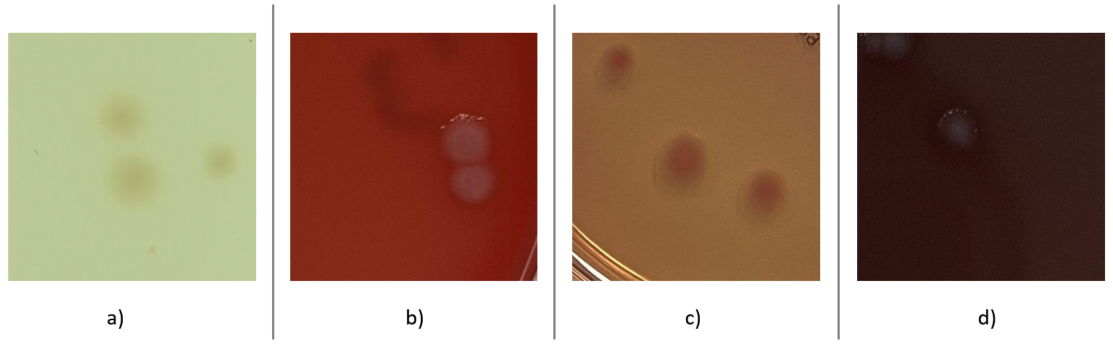

Figure 3.9: *P. aeruginosa* colonies: a) AGAR dataset, b) Blood agar, c) MacConkey agar, d) Chocolate agar.

To the extent of our awareness, this constitutes the first dataset for a deep learning project that incorporates such a broad spectrum of culture media diversity.

The dataset was generated through the capture of photographs using an iPhone 13 mini. These images were obtained under consistent lighting conditions, although without specialized equipment for light control. Given the variation in transparency across different media, photographs of those media were primarily taken against a dark background. The images were cropped so that only the plates were visible, after which they were uploaded and annotated using the Roboflow app.

The dataset is thus composed of 165 images and 1801 annotations. The distribution of the classes is illustrated in Figure 3.10. Among these are four images featuring empty plates, while other images encompass a mixture of multiple species within a single plate. Figure A.4 depicts the distribution of distinct culture media types. Notably, the least represented is the Mannitol media. This occurrence can be attributed to the specificity of this media for a single species, *S. aureus*, while the other media can support the growth of multiple species. In this case, most of the images contain twenty or fewer annotations. The image with the highest number of annotations contains 42 annotations, as depicted in Figures A.5 and A.6.

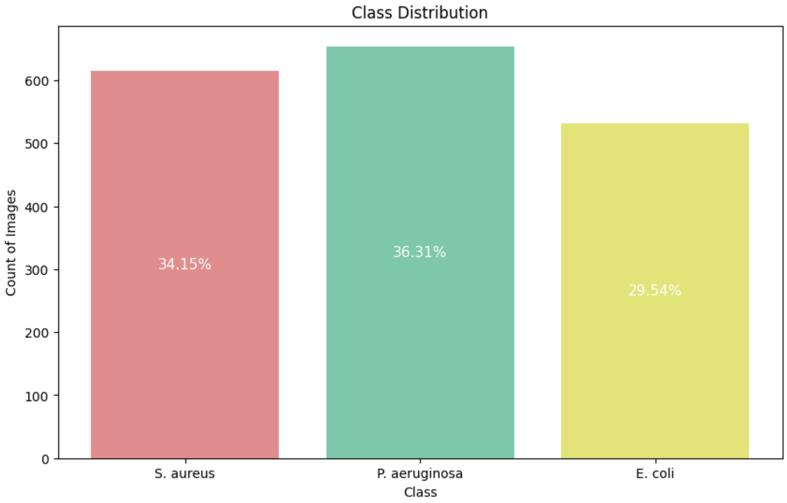

Figure 3.10: Class distribution on the new dataset.

3.2 Training Workflow

The workflow was executed on Google Colab Pro, using a GPU T4 provided by the service. The training of models was accomplished using the Detectron2 framework (Y. Wu, Kirillov, Massa, Lo, & Girshick, 2019), an open-source library developed by Facebook AI Research for tasks related to object detection with bounding boxes, instance segmentation masks, and human pose prediction. This framework is implemented using PyTorch and provides a range of models, including Faster R-CNN, Mask R-CNN, RetinaNet, Cascade R-CNN, Panoptic FPN, and TensorMask.

As previously mentioned, the AGAR dataset was divided based on the background categories, resulting in four distinct subsets that underwent separate training: bright, vague, dark, and low-resolution subsets. Furthermore, the complete dataset was also subjected to comprehensive training.

For the second part, training is conducted solely on the complete dataset rather than being performed for each distinct media type. This decision is primarily driven by the limited quantity of images within the new dataset, as well as the specialized nature of the media. The latter factor would lead to variations in the number of categories for each classification task.

Each dataset was divided into three subsets for training (60 %), validation (20 %), and testing (20 %). The first two sets played a role during the learning phase, serving for training and per-epoch evaluation, including validation loss computation. At the end of each training cycle, the training and validation losses and the Mean Average Precision (mAP) of the validation set were plotted. This aimed to help visualize the training process's progress and identify potential overfitting instances. The test set was left unseen by the model during the learning process, only to be employed for evaluation subsequent to the completion of the training.

For the AGAR dataset, the models underwent training for 10 epochs, in batches of 8 images. The new dataset underwent training cycles of 100 epochs due to the low number of images. Augmentations, such as random rotation, adjustments to brightness and exposure, blurring, and introducing noise, were applied to the images. The training process was also performed on the original dataset, which remained unaltered by any augmentation techniques. Regarding the optimizer, the training process employed the Stochastic Gradient Descent optimizer with a momentum of 0.9. The initial learning rate was set at 0.0005, with the first epoch devoted to a linear warm-up factor. Subsequently, after every three epochs, the learning rate was reduced by 1/10.

Two object detection models were selected for this project: the two-stage Faster R-CNN and the single-stage RetinaNet. Each model was then trained utilizing two underlying architectures: ResNet50 and ResNet101. For the initialization of the models, pre-trained weights from ImageNet were utilized, a common approach within the field.

3.3 Evaluation Metrics

The metrics established for the Common Objects in Context (COCO) dataset in Lin et al. (2015) were adopted for the classification task. Specifically, the mAP was employed, with a focus on evaluating the mAP across various Intersection over Union (IoU) thresholds.

The most common metric used in object detection tasks is the calculation of mAP. To achieve this, defining a true positive for the task is essential. The object detection task differs from the classical image classification problem in that it takes the model to accurately predict the correct class and precisely identify the object's position.

To assess the position and the class of a predicted bounding box, the IoU between the ground truth box and the predicted box is calculated. As the name suggests, the IoU represents the proportion of the area where both bounding boxes intersect relative to the total area of both boxes. This is calculated with the Jaccard similarity coefficient by the following formula:

$$JaccardIndex = \frac{|A \cap B|}{|A \cup B|}$$

A higher the IoU corresponds to a better prediction, meaning that a perfect prediction would result in an IoU of 100%.

Subsequently, the localization problem is approached as a binary classification problem, where the positive class signifies the correct detection of the object. With that, the precision and the recall are computed. The first measures the relevancy or correctness of the predicted items. It calculates the ratio of true positive and false positive predictions to the sum of true positive and false positive predictions. Precision answers the question: ”Out of the items predicted as positive, how many are actually positive?”.

$$Precision = \frac{CorrectPredictions}{TotalPredictions} = \frac{TruePositives}{TruePositives + FalsePositives}$$

Conversely, recall assesses the completeness or coverage of the predicted items. It calculates the ratio of the true positive predictions to the sum of true positive and false negative predictions. Recall answers the question: ”Out of all the positive items, how many were correctly identified?".

$$Recall = \frac{CorrectPredictions}{TotalGroundTruth} = \frac{TruePositives}{TruePositives + FalseNegatives}$$

In a multi-class classification problem, the model assigns conditional probabilities to each class for a given bounding box. These probabilities represent the likelihood of the bounding box belonging to a specific class. The bounding box is classified accordingly by comparing these probabilities against a defined threshold. This implies that higher probabilities indicate a greater chance of the bounding box corresponding to that specific class.

The following step consists of calculating the Average Precision (AP), and its execution has varied among different authors. For this project, the COCO evaluator was adopted. The COCO dataset employs a method to calculate the AP in object detection by generating precision-recall curves for each category. This involves the variation of the confidence threshold of model predictions. It incorporates 101-point interpolation, where precision is calculated across 101 recall thresholds that range from 0 to 1 with increments of 0.01 (Lin et al., 2015). In the equation below, *p* denotes precision, and *r* represents recall.

$$AP = \frac{1}{101} \sum_{\substack{r=0.0 \\ \text{step } 0.01}}^{1.0} p(r)$$

AP is computed individually for each class. Subsequently, mAP is determined by averaging the AP values across all the relevant classes. In the equation below, mAP is calculated as the mean of the AP values across all *k* classes.

$$mAP = \frac{1}{k} \sum_{i}^{1.0} APi$$

The COCO dataset ultimately performs these calculations at different IoU thresholds, typically ranging from 0.5 to 0.95 with increments of 0.05. The final mAP is then calculated as the mean of the mAP values at different thresholds.

The COCO mAP is often used due to its capacity for the detailed assessment of models across different IoU thresholds, thus enabling a more fine-grained evaluation process (Terven & Cordova-Esparza, 2023).

Mean Average Recall (mAR) is also reported alongside mAP, to simultaneously measure both proposal recall and localisation accuracy. mAR reflects the recall of each model for different IoU thresholds (from 0.5 to 1), summarising and rewarding both a high recall and a good localisation of the objects detected by the models (Hosang, Benenson, Dollár, & Schiele, 2015).

3.4 Transfer Learning

The reasoning behind transfer learning is using the overall knowledge acquired by a model trained on a larger dataset to enhance performance on a task with a smaller data fraction.

In practice, the initial training procedure applied to this data and most object detection tasks is founded on the concept of transfer learning. This implies starting the learning process with the weights of pre-trained models, which have been trained on datasets like the well-known ImageNet. Datasets like ImageNet are built upon millions of images. The models trained on these extensive datasets encompass thousands of diverse classes, requiring the models to make predictions across a large number of classes. Thus, the models are well trained to proficiently learn the feature extraction from photographs, allowing them to perform effectively on a given problem.

Working with images of Petri dishes containing bacterial colonies in this project represents a highly specialized task. Consequently, exploring the feasibility of training the most compact and least performing subsets using the weights of a model initially trained on the complete dataset is of great interest. This approach, known as transfer learning, or fine-tuning, aims to optimize the network to improve performance.

The rationale behind utilizing transfer learning as a fine-tuning approach in the first part of the project focused on the AGAR dataset involved a sequence of steps. After the initial training, the models exhibiting the weakest performance or those belonging to subsets with fewer images were selected, along with the model with the highest performance. Subsequently, the weights from the latter model were employed to retrain the models with the poorest performance. Lastly, the retrained models were subjected to a subsequent round of evaluation.

In the second part, the new dataset underwent an initial training phase, which was subsequently succeeded by a second training stage utilizing the weights derived from the top-performing model trained upon the AGAR dataset. As the new dataset does not have subsets, it would not make sense to repeat the training performed on the same entire dataset. Therefore, in this part, the aim was to assess how the knowledge from models trained on a different dataset, specifically a larger scale dataset like AGAR, would affect the results on the new dataset.

3.5 Ensemble of models

Ensemble techniques aim to enhance the performance of a single model by combining multiple algorithms. This combination can mitigate the bias or variance associated with a single model. Although most ensemble methods increase computational time during training or prediction, they provide an effective approach for advancing the state-of-the-art (Vilhelm, Limbert, Audebert, & Ceillier, 2022).

A range of ensemble methods is available, each offering distinct benefits.

- One approach involves data transformations, such as creating sub-samples from the original dataset (bagging or bootstrap aggregating) or applying augmentations to the training data.

- Another method combines models through hyperparameter variations, such as modifying loss functions and optimizers. This approach also contributes to improved outcomes. These two techniques extend the training time while enhancing model performance.

- Additionally, ensemble techniques can involve transforming and combining predictions. This can be achieved by merging predictions or introducing augmentations during testing. This, on the other hand, will consequently increase the prediction time rather than the training time.

Ensemble models, due to their nature, are widely used in applications in scenarios where real-time inference is not a primary concern. As a result, they serve as a favourable option for augmenting the performance of the models in this project.

In this project, having trained multiple models and generated multiple predictions, the chosen approach involved ensemble techniques that focused on combining diverse predictions. Weighted Boxes Fusion (WBF) employs the confidence scores from all proposed bounding boxes to formulate average boxes (Solovyev, Wang, & Gabruseva, 2021). During this stage, a grid search procedure was executed to identify the optimal model combination and determine the most suitable parameters for the WBF ensemble method.

Chapter 4

Results

This chapter outlines the findings derived from evaluating the test sets of both datasets. It commences with assessing the performance of the foundational object detection models across various datasets, including subsets of the AGAR dataset. Subsequently, an analysis of the impact of transfer learning and WBF ensemble method on model performance is conducted.

The results are presented in terms of key metrics, including mAP, mAR, and specific class detection mAP values for *S. aureus*, *P. aeruginosa*, and *E. coli*.

4.1 First part results

4.1.1 Base models evaluation

The comprehensive results for the various subsets of the AGAR dataset, as well as the dataset in its entirety, are presented in Tables 4.1 through 4.5.

When assessed in terms of the mAP metrics, the model that consistently outperforms others across different subsets and the entire dataset is Faster R-CNN with ResNet 101 architecture. Its performance spans from 51.09 % to 62.39 % mAP across subsets, with the bright subset yielding the lowest mAP, and the dark subset achieving the highest mAP.

On the other hand, RetinaNet models exhibit overall weaker performance. The architectural variations within the same model type generally yield minimal differences in performance. For instance, RetinaNet ResNet 50 performs better with the bright subset than RetinaNet ResNet 101, achieving mAP values of 38.23 % and 32.25 %, respectively. In contrast, ResNet 101 performs slightly better within the low-resolution subset than RetinaNet ResNet 50 (55.53 % versus 54.62 %). These differences are, however, marginal.

The mAR results closely mirror the mAP findings, with slightly higher values.

When analyzing the performance for each class, it becomes evident that the class with the most consistent and overall better results is *E. coli.* Its performance ranges from a low mAP value of 54.75 % in the bright subset, using the RetinaNet ResNet 101 model, to a high mAP of 71.18 % in the low-resolution subset, where this subset consistently demonstrates the best performances.

For the classification of *P. aeruginosa* colonies, the models perform relatively poorly in the bright subset, with mAP values ranging from 30.48 % to 43.19 %. Across the remaining subsets, the performance remains consistent within the different models, with the best result of 65.32 % achieved in the low-resolution subset using the Faster R-CNN 101 model.

The classification of *S. aureus* colonies showcase a higher level of inconsistency and greater variations between different model architectures. This class also tends to yield poorer results overall. Notably, the performance of the RetinaNet model for the classification of *S. aureus* is notably poor, particularly evident in the bright subset, where it achieves a mere 11.52 % mAP. Even in its best-performing scenario in the dark subset, the RetinaNet model results reach only 39.15 % mAP. In contrast, Faster R-CNN with ResNet 50 performs well, achieving a mAP of 55.30 % in the dark subset.

In summary, Faster R-CNN models demonstrate superior performance, particularly in subsets with darker backgrounds or images with lower resolutions. Training models using the entire dataset also results in notable achievements.

Model	**Backbone**	**mAP**	**mAR**	S. aureus	P. aeruginosa	E. coli
Faster R-CNN	ResNet 50	47.86	53.80	49.76	38.56	55.27
	ResNet 101	**51.09**	56.80	50.43	43.19	59.66
RetinaNet	ResNet 50	38.23	47.60	17.89	38.46	57.34
	ResNet 101	32.25	39.90	11.52	30.48	54.75

Table 4.1: Bright subset (%)

Model	**Backbone**	**mAP**	**mAR**	S. aureus	P. aeruginosa	E. coli
Faster R-CNN	ResNet 50	51.89	59.60	40.31	50.27	65.09
	ResNet 101	**52.25**	60.00	40.10	51.82	64.82
RetinaNet	ResNet 50	47.04	56.00	23.16	52.10	65.86
	ResNet 101	46.80	55.70	21.82	50.41	68.18

Table 4.2: Vague subset (%)

Model	**Backbone**	**mAP**	**mAR**	S. aureus	P. aeruginosa	E. coli
Faster R-CNN	ResNet 50	62.10	68.30	55.30	63.36	67.65
	ResNet 101	**62.39**	68.70	54.02	64.66	68.51
RetinaNet	ResNet 50	56.26	65.20	39.15	61.64	68.00
	ResNet 101	55.36	64.60	37.38	61.13	67.58

Table 4.3: Dark subset (%)

Model	**Backbone**	**mAP**	**mAR**	S. aureus	P. aeruginosa	E. coli
Faster R-CNN	ResNet 50	62.08	67.90	52.00	64.47	70.07
	ResNet 101	**62.18**	68.00	50.56	65.32	70.36
RetinaNet	ResNet 50	54.62	62.40	30.22	63.47	70.18
	ResNet 101	55.53	63.20	32.26	63.14	71.18

Table 4.4: Low Resolution subset (%)

Model	**Backbone**	**mAP**	**mAR**	S. aureus	P. aeruginosa	E. coli
Faster R-CNN	ResNet 50	61.63	67.70	53.22	63.47	68.21
	ResNet 101	**62.10**	68.00	53.00	63.75	69.56
RetinaNet	ResNet 50	54.49	63.40	34.30	61.21	67.96
	ResNet 101	54.34	62.40	34.30	60.98	67.76

Table 4.5: Total dataset (%)

4.1.2 Transfer Learning Results

The transfer learning process was carried out as previously described in the methods section. This step was conducted based on the outcomes presented in the evaluation process of the test sets for each subset and the entire dataset. After the evaluation process outlined in the earlier subsection, subsets that exhibited poorer results were selected for further training. This additional training phase involved employing the pre-trained weights of the model that demonstrated the best overall performance.

Consequently, the bright, vague, and low-resolution subsets underwent retraining using the Faster R-CNN model with ResNet 101 architecture. However, this time, the retraining was performed using the weights obtained from the initial training process conducted by the same model on the entire dataset.

The outcomes obtained from the evaluation process of the retrained models are presented in Table 4.6. The table includes mAP and mAR values, along with per-class mAP scores. Additionally, the table features the percentage point difference between the results of the retrained models and the original results of the same model as presented in Tables 4.1, 4.2, and 4.4. Although the initial results from the low-resolution subset were quite satisfactory, this retraining process illustrated how transfer learning could benefit both worst-performing and also well-performing models.

The results of the new training process have proven to be highly satisfactory, particularly for the subsets that initially displayed poorer results.

The bright subset had the most significant improvement in its mAP and mAR values, achieving an overall mAP of 58.36 %, which is approximately 7 percentage points higher than the baseline. Moreover, the performance in the classification of *P. aeruginosa* colonies, which originally had the lowest performance within this architecture, had the highest increase of 14.29 percentage points, resulting in a mAP of 57.47 %. Although the results for the other classes also exhibited improvement, the impact was not as pronounced.

Subset	**mAP (Δ)**	**mAR (Δ)**	S. aureus (Δ)	P. aeruginosa (Δ)	E. coli (Δ)
Bright	58.36 **(+7.27)**	63.30 (+6.50)	53.07 (+2.64)	57.47 **(+14.29)**	64.54 (+4.88)
Vague	56.32 (+4.08)	63.20 (+3.20)	41.77 (+1.66)	58.61 (+6.79)	68.59 (+3.77)
Low Res	**63.91 (+1.73)**	69.60 (+1.60)	**53.53 (+2.97)**	**66.32 (+1.01)**	**71.88 (+1.52)**

Table 4.6: Results after applying Transfer Learning for the bright, vague and low-resolution subsets (%) and the difference to the best baseline model results (percentage points)

The vague subset also experienced consistent improvement from the retraining process, although it ultimately achieved an overall mAP lower than the bright subset. In the baseline evaluation, the vague subset had a slightly better result than the bright subset, but the retraining process had a greater impact on the bright subset's performance.

The low-resolution subset, despite starting with a good baseline result, also experienced improvements ranging from 1.01 to 2.97 percentage points. Despite the relatively small differences, the mAP for the low-resolution subset surpassed the overall best result from the baseline testing, which was 62.39 % for the dark subset. Here, the low-resolution subset achieved an overall mAP of 63.91 %.

4.1.3 Ensemble Results

Applying ensemble methods involves various strategies, such as training a combination of different models with different parameters or creating sub-samples within the dataset. Ensemble methods can also involve creating augmentations on the test set during evaluation or combining results obtained from different trained models.

In this project, the WBF ensemble method was applied, where the results obtained from the various evaluation processes were combined to enhance the final performance. A grid search strategy was employed to identify the best combination of models and parameters for WBF, including IoU and skip IoU thresholds. The results presented in Table 4.7 reflect the outcomes of combining all models, including the results from transfer learning models for the respective subsets.

It is important to note that each model was given different weights based on performance. The optimal combination of parameters for WBF was determined to be an IoU threshold of 0.75 and a skip IoU threshold of 0.01.

Upon applying the ensemble method, the performance of every subset experienced an increase. Once again, the bright subset displayed the higthest improvement, as observed in the transfer learning process, achieving a mAP of 60 %. Ultimately, the subset with the best performance was the low-resolution subset, which achieved a mAP of 66,40 %, a 4.22-point increase compared to the baseline model. Conversely, the vague subset was the least improved, achieving a mAP of 57.90 %. Similarly to the transfer learning process, fine-tuning methods appeared to have less impact on the vague subset than the bright subset.

Subset	**mAP (Δ)**	**mAR (Δ)**	S. aureus (Δ)	P. aeruginosa (Δ)	E. coli (Δ)
Bright	60.00 **(+8.91)**	67.20 (+10.40)	56.60 (+6.17)	57.20 **(+14.01)**	66.10 (+6.45)
Vague	57.90 (+5.65)	66.40 (+6.40)	43.90 (+3.80)	59.20 (+7.38)	70.50 (+5.68)
Dark	66.20 (+3.81)	72.80 (+4.10)	**59.00 (+4.98)**	67.60 (+2.94)	72.00 (+3.49)
Low Res	**66.40 (+4.22)**	72.70 (+4.70)	57.30 (+6.74)	**68.40 (+3.09)**	**73.30 (+2.94)**
Total	65.10 (+3.00)	71.70 (+3.70)	56.20 (+3.20)	67.30 (+3.55)	71.60 (+2.04)

Table 4.7: Results after applying WBF for each subset (%) and the difference to the best baseline model results (percentage points)

The ensemble approach also significantly improved the per-class performance. *S. aureus* continued to be the most challenging class to classify, particularly within the vague subset. For *P. aeruginosa*, the bright subset exhibited the lowest performance at 57.20 %, but this subset also experienced the most substantial improvement, with an increase of 14 percentage points. The classification of *E. coli* remained the most consistent across subsets, achieving the best performance among the three classes.

4.2 Second part results

4.2.1 Base models evaluation

The results obtained from evaluating the newly created dataset are presented in Table 4.8. The same models and backbone architectures from the first part of the project were used.

In contrast to the results from the AGAR dataset, the best-performing model in this scenario was RetinaNet with ResNet 101 architecture, achieving a mAP of 52.40 %. On the other hand, the worst performer was Faster R-CNN ResNet 101 model, achieving a mAP of 47.94 %, almost five percentage points lower than the top-performing RetinaNet model.

Similar to the previous part, the classification of *E. coli* colonies demonstrated the best results among the three classes. Conversely, *P. aeruginosa* species proved to be the most challenging to classify, with the worst results across the models. Notably, unlike the AGAR dataset, the RetinaNet models did not struggle in classifying *S. aureus*. In the first part of the project, RetinaNet exhibited the lowest results for classifying this species. However, with the newly created dataset, RetinaNet achieved the best results, attaining a value of 52.20 % mAP.

Model	**Backbone**	**mAP**	**mAR**	S. aureus	P. aeruginosa	E. coli
Faster R-CNN	ResNet 50	48.59	55.80	46.39	43.78	55.61
	ResNet 101	47.94	55.40	45.16	43.88	54.77
RetinaNet	ResNet 50	**52.40**	59.40	52.20	47.77	57.23
	ResNet 101	51.92	58.60	51.39	46.67	57.71

Table 4.8: New dataset (%)

4.2.2 Transfer Learning Results

Continuing with a similar structure as in the first part, the dataset was retrained, utilizing the weights of the previously trained models in the new training process. This step had a slight difference compared to the first part. In this phase, the weights of the pre-trained models were taken from the models trained on the AGAR dataset. Several training cycles were performed for this part, using the weights from the different models trained on the entire AGAR dataset and the weights of the models trained on the different subsets. The objective was to observe how the transfer of knowledge from the models trained on the AGAR dataset would impact the performance of the new dataset.

Having trained the new dataset using the multiple models trained on the AGAR dataset and its subsets, it was concluded that the use of the weights from the model RetinaNet ResNet 50 trained on the entirety of the AGAR dataset resulted in the best performance. These results are illustrated in Table 4.9.

There was a small increase in the overall mAP and mAR metrics, of 0.79 and 0.50 percentage points, respectively. Similarly, the performance for the classification of each class also saw a slight increase, with the classification of *E. coli* and *S. aureus* showing the most improvement, both by around 1 percentage point.

Since the improvements in the results from the other models were lower than the ones mentioned, those results are not illustrated here.

Subset	**mAP (Δ)**	**mAR (Δ)**	S. aureus (Δ)	P. aeruginosa (Δ)	E. coli (Δ)
New Dataset	53.19 (+0.79)	59.90 (+0.50)	53.24 (+1.04)	48.07 (+0.30)	58.28 (+1.05)

Table 4.9: Results after applying Transfer Learning for the dataset (%) and the difference to the the best baseline model results (percentage points)

4.2.3 Ensemble Results

Similar to the first part, various results were assembled using the WBF method. Again, a grid search strategy was adopted to find the best parameter combination, and the models' results were weighted based on their performance. As in the previous part, the optimal threshold values were 0.75 for the IoU threshold and 0.01 for the skip IoU threshold.

In Table 4.10, it is evident that this ensemble method showed a better improvement than transfer learning. However, the use of transfer learning models' results contributed to enhancing the performance. When the ensemble was performed solely with the baseline models, the performance was generally lower by approximately one percentage point compared to the illustrated results.

The WBF method increased almost four percentage points in the overall mAP value, leading to the highest value of 56.30 %. On a per-class basis, the results were increased for every class, with improvements ranging from 3 to 4 percentage points. *P. aeruginosa* remained the class with the lowest classification performance, achieving a mAP of 51.80 %, while *E. coli* had the best classification performance with a value of 60.50 %.

Subset	**mAP (Δ)**	**mAR (Δ)**	S. aureus (Δ)	P. aeruginosa (Δ)	E. coli (Δ)
New Dataset	**56.30 (+3.90)**	62.90 (+3.50)	56.50 (+4.30)	51.80 (+4.03)	60.50 (+3.27)

Table 4.10: Results after applying WBF for the dataset (%) and the difference to the the best baseline model results (percentage points)

Chapter 5

Conclusion

This dissertation employed various deep-learning models designed for object detection to classify bacterial colonies inoculated in culture media using two distinct datasets. The first dataset, made publicly available by Majchrowska, Pawłowski, et al. (2021), consists of thousands of annotated high-resolution images of Petri dishes with TSA culture media with five different species. The second dataset was created during this project, containing 165 annotated images with three species inoculated in various types of culture media.

This study was conducted in two distinct parts. The first part involved utilising the AGAR dataset to classify *E. coli*, *S. aureus*, and *P. aeruginosa* colonies. In this phase, different approaches, such as transfer learning and ensemble methods, were applied to improve the performance reported in the literature.

The second part of the study encompassed the creation and annotation of a smaller dataset containing images to classify the same bacterial species. This dataset included a variety of culture media, such as blood agar and chocolate agar, as well as more specific agars where the bacteria were inoculated. This section aimed to enhance the baseline performance of the models trained in this dataset by employing transfer learning techniques using the weights from models trained on the first dataset. Additionally, the ensemble of results was applied to enhance the classification outcomes further.

As mentioned earlier, the dataset created and presented by Majchrowska, Pawłowski, et al. (2021) served as the foundation for this study. They established the dataset and conducted training using several models for colony classification and counting. Among the models employed were Faster R-CNN with various architectures, including ResNet 50 and 101, also used in this project. Additionally, they explored other models such as Cascade R-CNN, one-stage detectors like YOLOv4, and transformer-based models (Majchrowska, Pawlowski, et al., 2021).

The authors divided their dataset into two subsets for training. One subset consisted of high-resolution images (excluding the vague subset), while the other contained low-resolution images. Their reported results in terms of mAP varied from 49.3 % to 52.3 % for the high-resolution subset and from 56 % to 59.4 % for the low-resolution subset. Notably, the Cascade R-CNN model yielded the best outcomes among their tested models, although it was not applied in this project. Furthermore, when assessing per class mAP values, the class *S. aureus* was the one for which the models achieved the most favourable results.

In this project, the highest performing model on this dataset was Faster R-CNN with ResNet 101 backbone. Notably, there are distinctions between this project and the work conducted by Majchrowska, Pawłowski, et al. (2021). In the present study, an intentional decision was made to focus on only three species, accompanied by the exclusion of images with over 100 annotations. This selection led to a reduction in the dataset size by more than 8000 images. This process was primarily motivated by the clinical relevance of *S. aureus*, *E. coli*, and *P. aeruginosa* as prominent pathogenic agents in human infections. Conversely, *B. subtilis* lacks substantial clinical significance, and *C. albicans*, while prevalent, was excluded due to its classification as a yeast rather than a bacterium.

Furthermore, instead of categorising the dataset into subsets based on high and low resolution, this project divided it according to different background types: bright, dark, vague, and a subset containing low-resolution images, given that images with varying backgrounds were high-resolution. Additionally, the entire dataset was used for training. This was performed to gain insights into the models' performance across different luminosity backgrounds. Notably, the subset that yielded the best results was the dark subset, achieving 62.39 %. Additionally, the low-resolution subset, which can be compared with the one used by the authors, achieved a mAP of 62.18 %, surpassing the authors' result of 57.3 % on the same model. On the other hand, in the high-resolution subset examined by the authors, encompassing both the dark and bright subsets, they managed to achieve approximately 50% mAP using both Faster R-CNN models and reached 52,3% with Cascade R-CNN ResNeXt101. In comparison, in this project, the bright and dark subsets yielded 51.09 % and 62.39 % respectively.

These discrepancies in results can be attributed to the reduction in the number of classes and annotations in this study. Furthermore, limited computational resources prevented the replication of the pre-processing and augmentations employed by the authors during their training process, which contributed to the divergence in outcomes.

In the per-class analysis, the authors exclusively reported results for the model trained on the high-resolution subset, employing Faster R-CNN ResNet50 and Cascade R-CNN HRNet. Notably, their outcomes favoured classifying smaller-sized colonies like *S. aureus*. This contrasts with the findings in this dissertation, where the classification of *S. aureus* colonies proved to be the most challenging across all subsets, except for the bright subset, where Faster R-CNN models exhibited greater difficulty with *P. aeruginosa* colonies.

This divergence in results can be attributed to the pre-processing strategy adopted by the authors, which involved dividing images into multiple lower-resolution patches. In contrast, the images in this study were not segmented in this manner, leading to the resizing of the images by the models. Consequently, smaller colonies became more intricate for the models to detect and classify, particularly noticeable in the case of RetinaNet models. These models exhibited notably lower performance compared to the two-stage models.

Carrying out object detection predictions for small objects proves to be a significant challenge, particularly when employing one-stage models like RetinaNet, in contrast to the performance of two-stage models (Zhou, Li, Peng, Wang, & Du, 2021). Despite often possessing higher inference speeds, single-stage models typically approach the entire network as a regression problem, making simultaneous predictions for object location and category. These models tend to generate many candidate bounding boxes, particularly negative examples corresponding to background or non-object regions. These models need help to effectively handle the challenge posed by the class imbalance between negative and positive regions containing actual objects (Zhou et al., 2021).

In summary, single-stage models such as RetinaNet do not appear to be the most suitable solution for tasks involving the detection of small objects. This holds true unless some pre-processing is applied to the images or adjustments are made to the model architecture. Notably, other one-stage models like the YOLO models mentioned earlier in the dissertation have demonstrated impressive results across a range of object detection tasks. While not explored in this project, these models hold the potential for achieving enhanced performance and fast inference speed.

The lower performance observed in the bright subset can be attributed to reduced contrast and colour similarity between colonies and the background. Within this subset, the classification of *P. aeruginosa* exhibits the weakest performance. This is potentially due to the heterogeneous nature of their colonies, which are often transparent and flat, possess irregular shapes, and tend to aggregate closely with other colonies. These characteristics, visible in Figure 3.9 a), make them challenging for the model to discern accurately. The vague subset similarly exhibited suboptimal performance. The authors characterise this subset as comprising images captured under existing ambient lighting conditions, resulting in low-contrast images that proved challenging to annotate even for a skilled microbiologist. Additionally, the distribution of classes within this subset is imbalanced, as depicted in Figure 3.5, further contributing to less effective model performance.

However, retraining the models using these two underperforming subsets yielded promising results. The transfer of knowledge from the model trained on the complete dataset proved highly beneficial. Fine-tuning models using the weights of that comprehensive model significantly enhanced performance, particularly for the bright subset and the detection of *P. aeruginosa*, leading to a great increase of 14 percentage points.

The ensemble method also significantly enhanced overall performance across all subsets, with notable improvements once again observed in the bright subset, particularly in the detection of *P. aeruginosa*. It is worth emphasising that this ensemble approach is not entirely independent of the transfer learning technique, as it leverages the results from those retrained models. While various combinations were experimented with, the inclusion of the RetinaNet models, despite their lower performances, proved beneficial to the ensemble's final performance. Their inclusion was essential in achieving the elevated results presented.

The dataset for the second part of this project was generated using agar plates from the microbiology laboratory at my institute. Bacterial cultures were inoculated onto agar plates containing various types of culture media. It is crucial to note that the inoculation process for this dataset did not adhere to the standardised protocols typically used in the clinical setting. Instead, it was designed to produce well-separated colonies for this project. In clinical routine, colonies from real patient samples are inoculated using specific techniques that vary depending on the sample type. To maximise the visibility and facilitate the identification of pathogenic microorganisms, colonies in real samples often grow in high numbers and need to be better separated. Following the real clinical protocols for inoculation was not feasible for the annotation process in this project due to the high colony density. Many colonies would not be individually discernible in such cases, making the annotation and model training very challenging.

After the inoculation and incubation of the plates, images were captured. To enhance contrast between colonies and the background, a deliberate effort was made to take photographs with darker background settings. This was particularly important for agar plates containing MacConkey and Mannitol culture media, as they are transparent and allow the background colour to be visible. However, it is essential to mention that lighting conditions could not be standardised, and there was no designated photography area. Despite efforts to minimise them, some images may contain reflections.

With this dataset, RetinaNet with a ResNet 50 backbone emerged as the top-performing model, in contrast to the first part, where Faster R-CNN with ResNet 101 was superior. However, the performance difference between models was relatively small, with RetinaNet outperforming the lowest-performing model by only about 4.5 percentage points.

Similar to the first part, the classification of *P. aeruginosa* proved to be the most challenging for the models. This difficulty can be attributed to the specific morphology of these colonies. Additionally, both *S. aureus* and *E. coli* exhibit distinct characteristics in certain culture media. For instance, *S. aureus* colonies had a rounded shape and a goldish colour, which caused the background around them to turn yellow in Mannitol agar. *E. coli* colonies displayed a distinctive purple colour in MacConkey agar, making them stand out compared to *P. aeruginosa*. In contrast, *P. aeruginosa* colonies typically had a flat, irregular shape, were transparent, and often had a colour similar to the background. They also reflect some light, sometimes having a shiny appearance, which makes them more challenging to detect.

Various training processes were applied, including different pre-processing and augmentation techniques. However, the best results were achieved with a training process that did not involve pre-processing or augmentations of the images. This dataset is relatively small, consisting of only 165 images, and exhibits considerable variability in background colours and colony morphology. Augmentations may have introduced additional variability, potentially obscuring patterns the models needed to learn.

The process of generalisation, tested by fine-tuning a model trained on the second dataset using weights previously trained on the entire AGAR dataset, had a small impact on performance. This can be attributed to the substantial differences between the two datasets. While they share the same bacterial classes, they represent them differently. The images in the AGAR dataset have significantly higher resolutions and exhibit consistent inoculation culture media, resulting in uniform colony morphology. In contrast, the new dataset features a lower resolution and is considerably smaller. Additionally, it encompasses a much wider diversity in terms of background colours and colony characteristics.

To conclude, the WBF method is a crucial asset in improving model performance. It delivers a substantial average increase of 4 percentage points across all metric aspects, signifying a notable enhancement in performance. Similar to the first part, combining results from all trained models, including those from transfer learning, proved to be the most effective approach in achieving the best results.

While not providing a comprehensive explanation, some examples of errors made by the models in the dataset are presented in Figure 5.1. These include, as suspected, false identification of light reflections (Figure 5.1 d), as well as artifacts on the media (Figure 5.1 b) being misclassified as colonies, failures to recognise colonies when they are nearby or form a mesh (Figure 5.1 a and c) (although this can also be challenging for a human annotator at times), as well as some misclassifications of colonies if they have a different morphology than the others. As previously mentioned, better classification tends to occur for colonies with more distinctive characteristics, such as the purple *E. coli* and golden *S. aureus*.

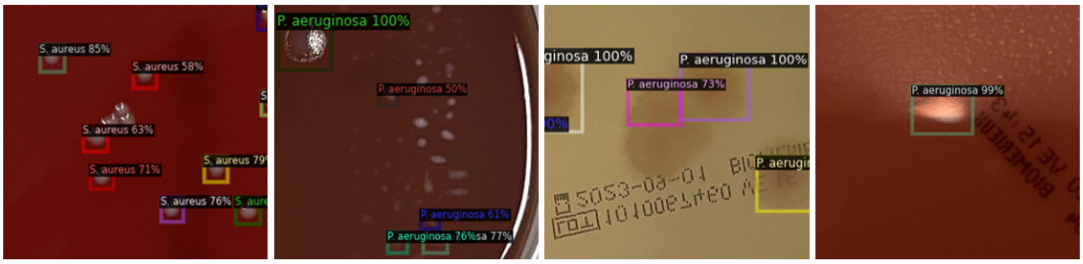

Figure 5.1: Some examples of classification errors.

Interestingly, there are no significant differences in performance when using different object detection models. The models employed in this study vary in depth and the overall architecture of the neural networks, but they all exhibit fairly similar performance on the datasets. However, it is important to note that while the overall performance may be similar, it does not necessarily mean that the models rely on the same image features for classification.

Ensembling proved to be a valuable approach for improving the baseline models. However, achieving these improvements required training multiple models and applying transfer learning and model retraining. These are two techniques for enhancing performance: the training process and the transfer of knowledge from one model to another, and the ensemble method involving the aggregation of knowledge from the results of previously trained models during the inference process. In addition to performance improvements, these techniques also have implications for their task type. In the context of this dissertation, achieving better performance was prioritised over faster inference speed. This is because the incubation periods in a laboratory setting are typically hours long, and real-time identification and classification of colonies are not a strict requirement. While training multiple models can be time-consuming, ensembling the inference results of various models is a feasible and effective way to achieve superior results, even if it comes at the cost of some inference speed.

The ensemble method tested in this project involved combining the various results and fusing them based on probability scores and weights to formulate average boxes. However, other approaches to ensembling could be considered, such as applying ensemble techniques during the training process, where multiple models and parameters are combined. This approach can lead to even better results but may require significantly more computational power.

This study represents the first instance of utilising a dataset encompassing various culture media, consequently incorporating distinct morphological characteristics within the same species. Furthermore, this study is also the first in its approach to encompass the training of two distinct datasets to enhance the performance of the second dataset, as well as in the utilisation of WBF ensemble, specifically within the domain of colony detection and classification.

However, this project has several limitations that should be acknowledged. First, the created dataset needs further improvement regarding the number of images and their quality. Establishing designated areas with better lighting conditions would be beneficial to avoid artifacts in the images. Additionally, using better camera settings, such as standardised vertical distance to the plates, avoiding angles, and capturing higher resolution images, could enhance the dataset's quality. Moreover, exploring more advanced approaches for object detection on high-resolution images can be advantageous.

Another limitation is that this project's inoculation process differs from the clinical setting. Future work should focus on developing models that closely mimic the conditions encountered in clinical microbiology laboratories. However, it is worth noting that the trained models perform reasonably well on real examples of microbiology analysis, effectively detecting and classifying individualised colonies (Figure 5.2). Additionally, since the annotation process for this task is labour-intensive, these models can serve as valuable tools to assist in the annotation process.

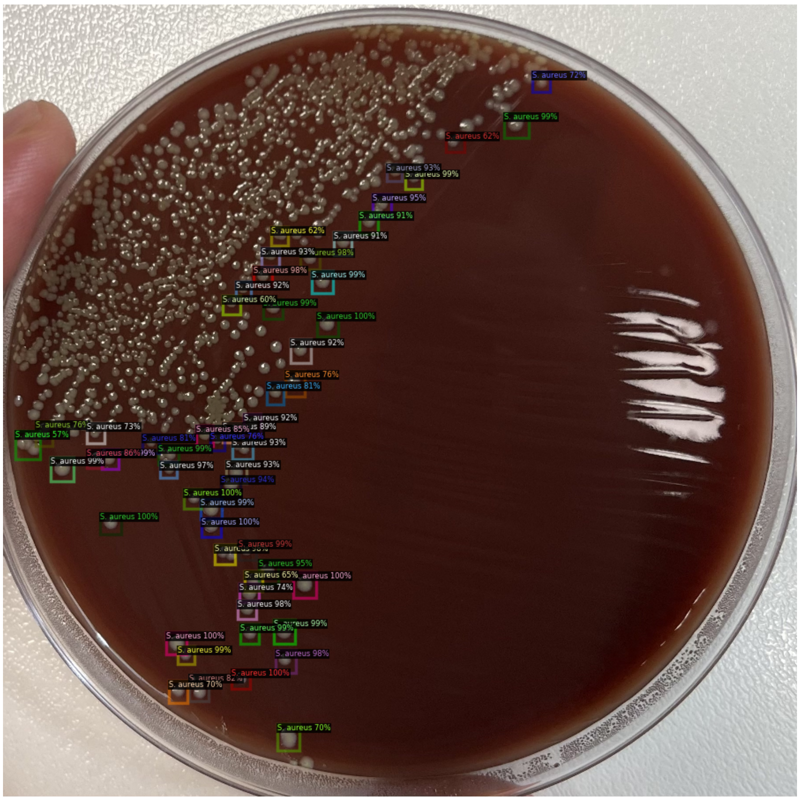

Figure 5.2: Inference on a plate from the work routine.

Furthermore, other models can be trained for this task as part of future work. For instance, Mask R-CNN, commonly used for segmentation tasks instead of bounding boxes, may be beneficial for detecting colonies with irregular shapes, such as *P. aeruginosa*. Other models like the one-stage YOLO family, which are currently achieving impressive performance benchmarks with newer versions like YOLOv8 (Jocher, Chaurasia, & Qiu, 2023), YOLO-NAS (Aharon et al., 2021), and others, could also be explored. Additionally, more complex pre-processing techniques and advanced ensemble models can be tested further to improve the performance of colony detection and classification.

In conclusion, this project successfully achieved all of its proposed objectives. It significantly improved the performance of models on the AGAR dataset through various techniques. Creating a dataset with unique characteristics encompassing diverse culture media, is unprecedented to the best of my knowledge. Moreover, this work explored the generalisation of models and provided valuable insights into the suitability of different models and techniques for various tasks.

Ultimately, this project is now part of a journey towards developing tools that will contribute to advancing the medical and health field.

Bibliography

Aharon, S., Louis-Dupont, Ofri Masad, Yurkova, K., Lotem Fridman, Lkdci, … Eran-Deci (2021). *Super-gradients.* GitHub. Retrieved from https://zenodo.org/record/7789328 doi: 10.5281/ZENODO.7789328

Andreini, P., Bonechi, S., Bianchini, M., Mecocci, A., & Scarselli, F. (2020, February). Image generation by GAN and style transfer for agar plate image segmentation. *Comput. Methods Programs Biomed.*, *184*, 105268. doi: 10.1016/j.cmpb.2019.105268

Antonios, K., Croxatto, A., & Culbreath, K. (2021, December). Current State of Laboratory Automation in Clinical Microbiology Laboratory. *Clin. Chem.*, *68*(1), 99–114. doi: 10.1093/clinchem/hvab242

Bourbeau, P. P., & Ledeboer, N. A. (2013). Automation in clinical microbiology. *Journal of Clinical Microbiology*, *51*(6), 1658-1665. Retrieved from https://journals.asm.org/doi/abs/10.1128/JCM.00301-13 doi: 10.1128/JCM.00301-13

Boureau, Y. L., Ponce, J., & Lecun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning* (pp. 111–118). Retrieved from https://nyuscholars.nyu.edu/en/publications/a-theoretical-analysis-of-feature-pooling-in-visual-recognition

Cabitza, F., & Banfi, G. (2018, April). Machine learning in laboratory medicine: waiting for the flood? *Clinical Chemistry and Laboratory Medicine (CCLM)*, *56*(4), 516–524. doi: 10.1515/cclm-2017-0287

Cai, Z., & Vasconcelos, N. (2017). *Cascade r-cnn: Delving into high quality object detection.* arXiv. Retrieved from https://arxiv.org/abs/1712.00726 doi: 10.48550/ARXIV.1712.00726

Camaggi, C. M., Zavatto, E., Gramantieri, L., Camaggi, V., Strocchi, E., Righini, R., … Bolondi, L. (2010, September). Serum albumin-bound proteomic signature for early detection and staging of hepatocarcinoma: sample variability and data classification. *Clin. Chem. Lab. Med.*, *48*(9), 1319–1326. doi: 10.1515/CCLM.2010.248

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *CoRR*, *abs/2005.12872*. Retrieved from https://arxiv.org/abs/2005.12872

Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* (Vol. 1, pp. 886–893vol.1). IEEE. doi: 10.1109/CVPR.2005.177

Darcy, A. M., Louie, A. K., & Roberts, L. W. (2016, February). Machine Learning and the Profession of Medicine. *JAMA*, *315*(6), 551–552. doi: 10.1001/jama.2015.18421

Deist, T. M., Dankers, F. J. W. M., Valdes, G., Wijsman, R., Hsu, I.-C., Oberije, C., … Lambin, P. (2018, July). Machine learning algorithms for outcome prediction in (chemo)radiotherapy: An empirical comparison of classifiers. *Med. Phys.*, *45*(7), 3449–3459. doi: 10.1002/mp.12967

Dezső, Z., & Ceccarelli, M. (2020, December). Machine learning prediction of oncology drug targets based on protein and network properties. *BMC Bioinf.*, *21*(1), 1–12. doi: 10.1186/s12859-020-3442-9

Everingham, M., Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vision*. Retrieved from https://www.semanticscholar.org/paper/The-Pascal-Visual-Object-Classes-(VOC)-Challenge-Everingham-Gool/82635fb63640ae95f90ee9bdc07832eb461ca881

Ferrari, A., Lombardi, S., & Signoroni, A. (2017). Bacterial colony counting with convolutional neural networks in digital microbiology imaging. *Pattern Recognition*, *61*, 629-640. Retrieved from https://www.sciencedirect.com/science/article/pii/S0031320316301650 doi: https://doi.org/10.1016/j.patcog.2016.07.016

Girshick, R. (2015). *Fast r-cnn.* arXiv. Retrieved from https://arxiv.org/abs/1504.08083 doi: 10.48550/ARXIV.1504.08083

Girshick, R. B., Donahue, J., Darrell, T., & Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, *abs/1311.2524*. Retrieved from http://arxiv.org/abs/1311.2524

Graczyk, K. M., Pawłowski, J., Majchrowska, S., & Golan, T. (2022, June). Self-normalized density map (SNDM) for counting microbiological objects. *Sci. Rep.*, *12*(10583), 1–13. doi: 10.1038/s41598-022-14879-3

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., … Chen, T. (2018, May). Recent advances in convolutional neural networks. *Pattern Recognit.*, *77*, 354–377. doi: 10.1016/j.patcog.2017.10.013

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. (2023, January). *(2016) Deep Learning for Visual Understanding A Review. Neurocomputing, 187, 27-48. - References - Scientific Research Publishing.* Retrieved from https://www.scirp.org/(S(i43dyn45teexjx455qlt3d2q))/reference/referencespapers.aspx?referenceid=2107410 ([Online; accessed 13. Jan. 2023])

He, K., Gkioxari, G., Dollár, P., & Girshick, R. B. (2017). Mask R-CNN. *CoRR*, *abs/1703.06870*. Retrieved from http://arxiv.org/abs/1703.06870

Hosang, J., Benenson, R., Dollár, P., & Schiele, B. (2015, August). What Makes for Effective Detection Proposals? *IEEE Trans. Pattern Anal. Mach. Intell.*, *38*(4), 814–830. doi: 10.1109/TPAMI.2015.2465908

Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022, January). A Review of Yolo Algorithm Developments. *Procedia Comput. Sci.*, *199*, 1066–1073. doi: 10.1016/j.procs.2022.01.135

Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics yolov8*. Retrieved from https://github.com/ultralytics/ultralytics

Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020, December). A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.*, *53*(8), 5455–5516. doi: 10.1007/s10462-020-09825-6

Khouani, A., El Habib Daho, M., Mahmoudi, S. A., Chikh, M. A., & Benzineb, B. (2020, August). Automated recognition of white blood cells using deep learning. *Biomed. Eng. Lett.*, *10*(3), 359–367. doi: 10.1007/s13534-020-00168-3

Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2014, November). Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.*, *13*, 8–17. doi: 10.1016/j.csbj.2014.11.005

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017, May). ImageNet classification with deep convolutional neural networks. *Commun. ACM*, *60*(6), 84–90. doi: 10.1145/3065386

LeCun, Y., Bengio, Y., & Hinton, G. (2015, May). Deep learning. *Nature*, *521*, 436–444. doi: 10.1038/nature14539

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017, August). Focal Loss for Dense Object Detection. *arXiv*. doi: 10.48550/arXiv.1708.02002

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., … Dollár, P. (2015). *Microsoft coco: Common objects in context.*

Liu, S.-J., Huang, P.-C., Liu, X.-S., Lin, J.-J., & Zou, Z. (2022, November). A two-stage deep counting for bacterial colonies from multi-sources. *Appl. Soft Comput.*, *130*, 109706. doi: 10.1016/j.asoc.2022.109706

Lowe, D. G. (2004, November). Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, *60*(2), 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Majchrowska, S., Pawlowski, J., Czerep, N., Górecki, A., Kucinski, J., & Golan, T. (2021). Deep neural networks approach to microbial colony detection - a comparative analysis. *CoRR*, *abs/2108.10103*. Retrieved from https://arxiv.org/abs/2108.10103

Majchrowska, S., Pawłowski, J., Guła, G., Bonus, T., Hanas, A., Loch, A., … Drulis-Kawa,

Z. (2021). *Agar a microbial colony dataset for deep learning detection.*

McBee, M. P., Awan, O. A., Colucci, A. T., Ghobadi, C. W., Kadom, N., Kansagra, A. P., … Auffermann, W. F. (2018, November). Deep Learning in Radiology. *Acad. Radiol.*, *25*(11), 1472–1480. doi: 10.1016/j.acra.2018.02.018

Nie, D., Shank, E. A., & Jojic, V. (2015, September). A deep framework for bacterial image segmentation and classification. In *BCB '15: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics* (pp. 306–314). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/2808719.2808751

Novak, S. M., & Marlowe, E. M. (2013, September). Automation in the clinical microbiology laboratory. *Clin. Lab. Med.*, *33*(3), 567–588. doi: 10.1016/j.cll.2013.03.002

Pawłowski, J., Majchrowska, S., & Golan, T. (2022, March). Generation of microbial colonies dataset with deep learning style transfer. *Sci. Rep.*, *12*(5212), 1–12. doi: 10.1038/s41598-022-09264-z

Pritt, B. S. (2020a, April). Computer Vision and Artificial Intelligence Are Emerging Diagnostic Tools for the Clinical Microbiologist. *J. Clin. Microbiol.*. Retrieved from https://journals.asm.org/doi/10.1128/jcm.00511-20

Pritt, B. S. (2020b, April). Detection of Intestinal Protozoa in Trichrome-Stained Stool Specimens by Use of a Deep Convolutional Neural Network. *J. Clin. Microbiol.*. Retrieved from https://journals.asm.org/doi/10.1128/JCM.02053-19

Rawat, W., & Wang, Z. (2017, September). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Comput.*, *29*(9), 2352–2449. doi: 10.1162/NECO_a_00990

Redmon, J., Divvala, S. K., Girshick, R. B., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, *abs/1506.02640*. Retrieved from http://arxiv.org/abs/1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster r-cnn: Towards real-time object detection with region proposal networks.* arXiv. Retrieved from https://arxiv.org/abs/1506.01497 doi: 10.48550/ARXIV.1506.01497

Ronzio, L., Cabitza, F., Barbaro, A., & Banfi, G. (2021). Has the flood entered the basement? a systematic literature review about machine learning in laboratory medicine. *Diagnostics*, *11*(2). Retrieved from https://www.mdpi.com/2075-4418/11/2/372 doi: 10.3390/diagnostics11020372

Salah, H. T., Muhsen, I. N., Salama, M. E., Owaidah, T., & Hashmi, S. K. (2019, December). Machine learning applications in the diagnosis of leukemia: Current trends and future directions. *Int. J. Labor. Hematol.*, *41*(6), 717–725. doi: 10.1111/ijlh.13089

Savardi, M., Ferrari, A., & Signoroni, A. (2018). Automatic hemolysis identification on aligned dual-lighting images of cultured blood agar plates. *Computer Methods and Programs in Biomedicine*, *156*, 13-24. Retrieved from https://www.sciencedirect.com/science/article/pii/S0169260717307113 doi: https://doi.org/10.1016/j.cmpb.2017.12.017

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013, December). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv*. doi: 10.48550/arXiv.1312.6229

Solovyev, R., Wang, W., & Gabruseva, T. (2021, mar). Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, *107*, 104117. Retrieved from https://doi.org/10.1016%2Fj.imavis.2021.104117 doi: 10.1016/j.imavis.2021.104117

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., … Luo, P. (2020). Sparse R-CNN: end-to-end object detection with learnable proposals. *CoRR*, *abs/2011.12450*. Retrieved from https://arxiv.org/abs/2011.12450

Surinova, S., Choi, M., Tao, S., Schüffler, P. J., Chang, C.-Y., Clough, T., … Aebersold, R. (2015, September). Prediction of colorectal cancer diagnosis based on circulating plasma proteins. *EMBO Mol. Med.*, *7*(9), 1166–1178. doi: 10.15252/emmm.201404873

Talo, M. (2019). *An automated deep learning approach for bacterial image classification.* arXiv. Retrieved from https://arxiv.org/abs/1912.08765 doi: 10.48550/ARXIV.1912.08765

Terven, J., & Cordova-Esparza, D. (2023). *A comprehensive review of yolo: From yolov1 and beyond.*

Vilhelm, A., Limbert, M., Audebert, C., & Ceillier, T. (2022). *Ensemble learning techniques for object detection in high-resolution satellite images.*

Wang, H.-Y., Hsieh, C.-H., Wen, C.-N., Wen, Y.-H., Chen, C.-H., & Lu, J.-J. (2016, June). Cancers Screening in an Asymptomatic Population by Using Multiple Tumour Markers. *PLoS One*, *11*(6), e0158285. doi: 10.1371/journal.pone.0158285

Wang, S., Yang, D. M., Rong, R., Zhan, X., & Xiao, G. (2019, September). Pathology Image Analysis Using Segmentation Deep Learning Algorithms. *Am. J. Pathol.*, *189*(9), 1686. doi: 10.1016/j.ajpath.2019.05.007

Wen, B., Zeng, W.-F., Liao, Y., Shi, Z., Savage, S. R., Jiang, W., & Zhang, B. (2020, November). Deep Learning in Proteomics. *Proteomics*, *20*(21-22), e1900335. doi: 10.1002/pmic.201900335

Whipp, J., & Dong, A. (2022, December). YOLO-based Deep Learning to Automated Bacterial Colony Counting. In *2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM)* (pp. 120–124). IEEE. doi: 10.1109/BigMM55396.2022.00028

Williams, R. E., & Trotman, R. E. (1969). Automation in diagnostic bacteriology. *Journal of Clinical Pathology*, *s2-3*(1), 8–13. Retrieved from https://jcp.bmj.com/content/s2-3/1/

8 doi: 10.1136/jcp.s2-3.1.8

Wu, X., Sahoo, D., & Hoi, S. C. H. (2019). Recent advances in deep learning for object detection. *CoRR*, *abs/1908.03673*. Retrieved from http://arxiv.org/abs/1908.03673

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). *Detectron2*. https://github.com/facebookresearch/detectron2.

Zhou, Z., Li, Y., Peng, C., Wang, H., & Du, S. (2021, feb). Image processing: Facilitating retinanet for detecting small objects. *Journal of Physics: Conference Series*, *1815*(1), 012016. Retrieved from https://dx.doi.org/10.1088/1742-6596/1815/1/012016 doi: 10.1088/1742-6596/1815/1/012016

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2020). *Deformable detr: Deformable transformers for end-to-end object detection.* arXiv. Retrieved from https://arxiv.org/abs/2010.04159 doi: 10.48550/ARXIV.2010.04159

Zieliński, B., Plichta, A., Misztal, K., Spurek, P., Brzychczy-Włoch, M., & Ochońska, D. (2017, September). Deep learning approach to bacterial colony classification. *PLoS One*, *12*(9), e0184554. doi: 10.1371/journal.pone.0184554

Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A., & Telenti, A. (2019, January). A primer on deep learning in genomics. *Nat. Genet.*, *51*(1), 12–18. doi: 10.1038/s41588-018-0295-5

Appendix A

Appendix

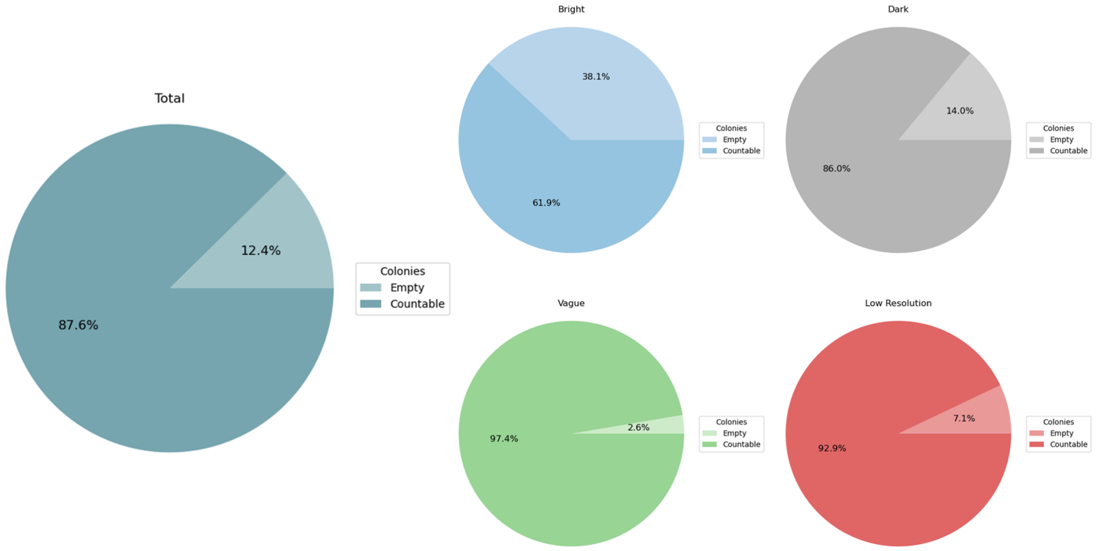

Figure A.1: Distribution of countable and empty images on the whole dataset and within each Background group.

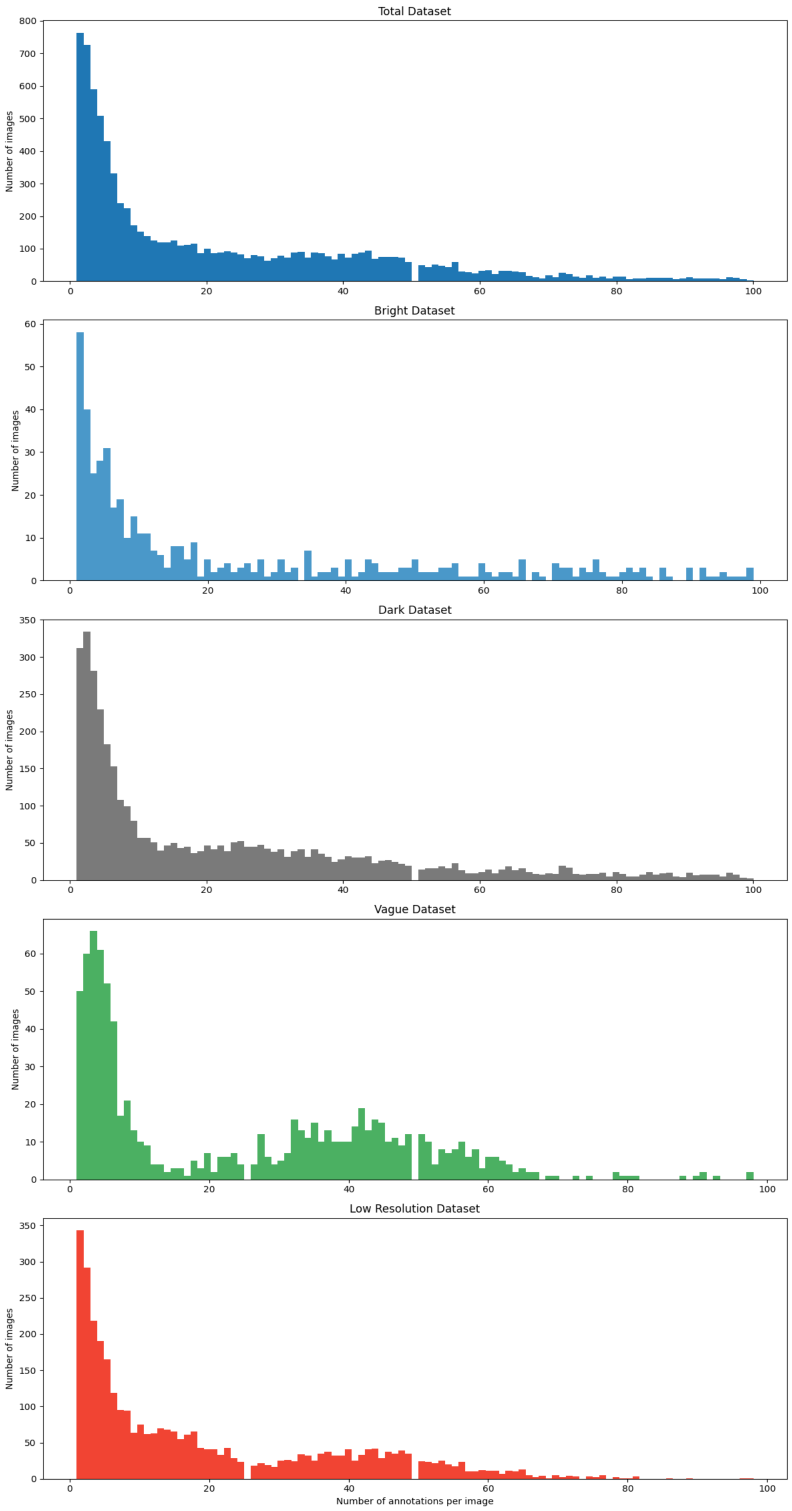

Figure A.2: Distribution of Annotations per Image: number of images based on their annotations for the entire dataset and across each subgroup.

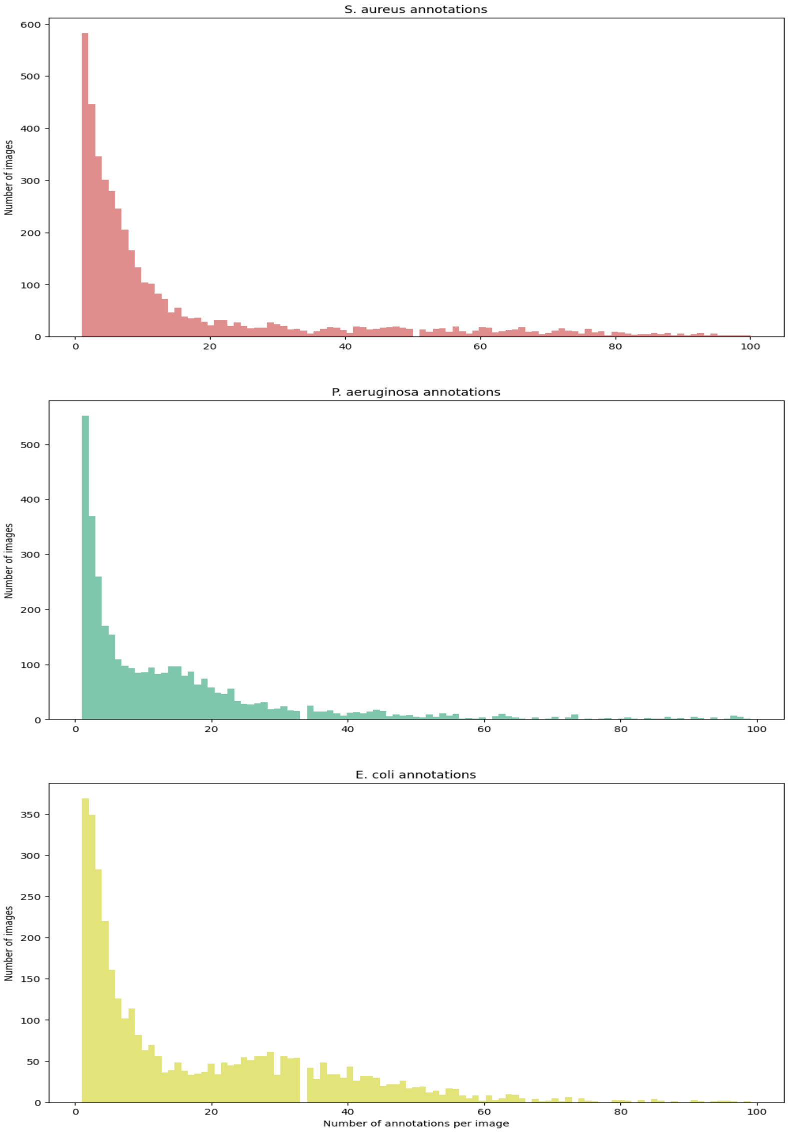

Figure A.3: Distribution of Annotations per Image for Each Annotation Class: number of images based on the number of annotations they contain for each class of annotations.

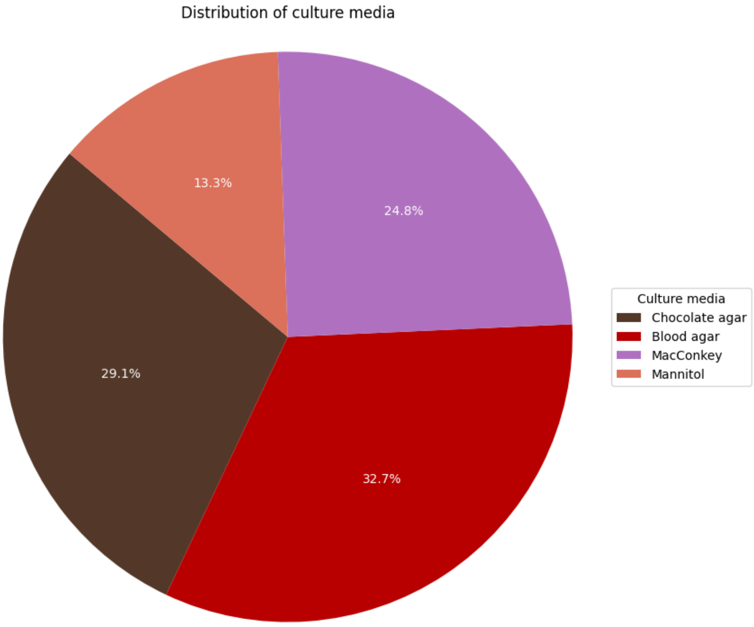

Figure A.4: Distribution of culture media.

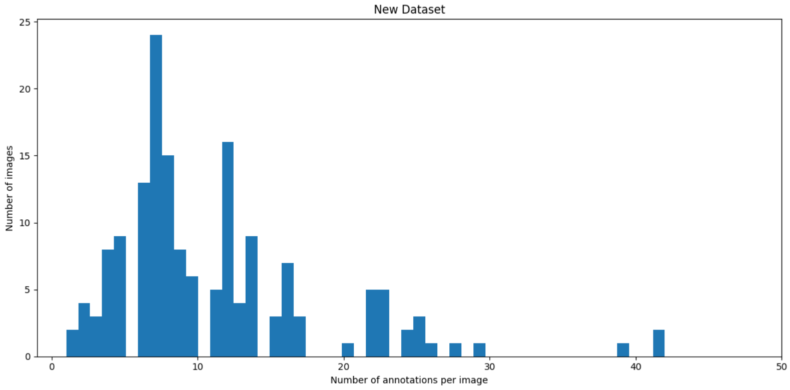

Figure A.5: Distribution of Annotations per Image: number of images based on the number of annotations they contain for the new dataset.

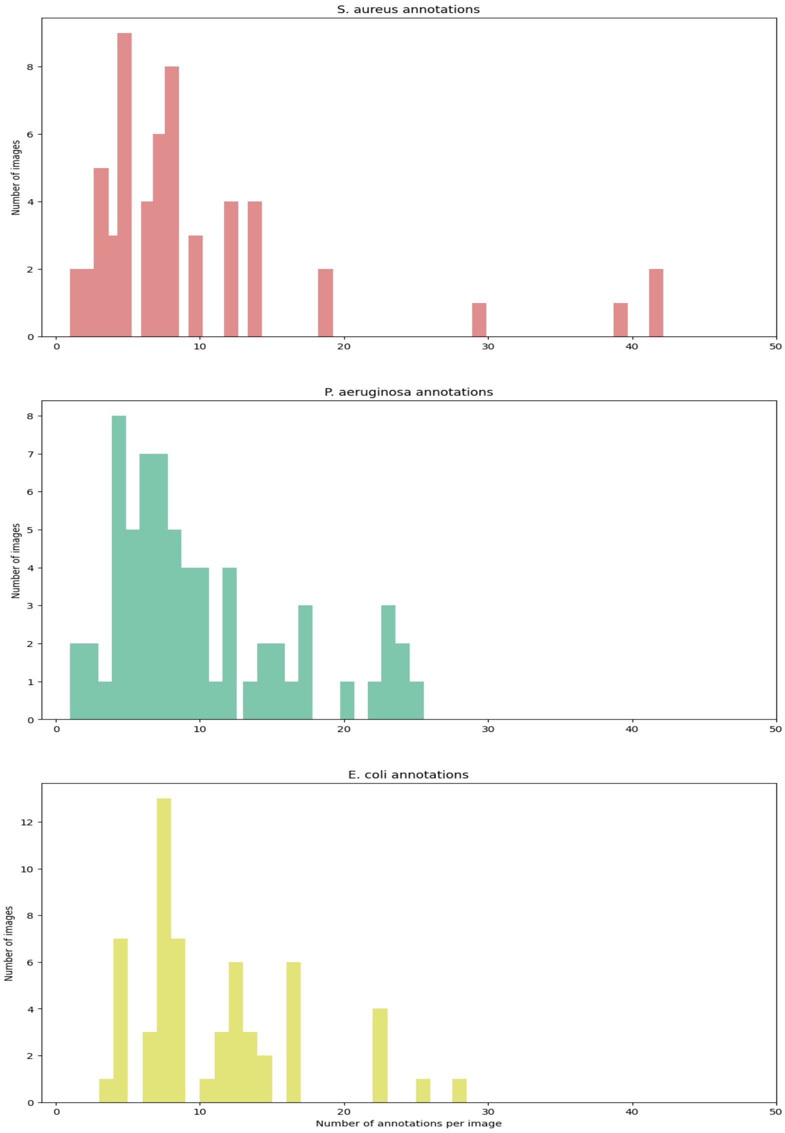

Figure A.6: Distribution of Annotations per Image for Each Annotation Class: number of images based on the number of annotations they contain for each class of annotations.