NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

ANDRÉ FRANCISCO ROSA MATOS

Degree in Computer Science

DEVELOPMENT OF A DATABASE OF SPATIAL STRUCTURE IN BIOFILMS

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon

September, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF
COMPUTER SCIENCE

DEVELOPMENT OF A DATABASE OF SPATIAL STRUCTURE IN BIOFILMS

ANDRÉ FRANCISCO ROSA MATOS

Degree in Computer Science

Adviser: Rafael Costa
Assistant Professor, NOVA School of Science and Technology, NOVA University Lisbon

Co-advisers: Pedro Barahona
Full Professor, NOVA School of Science and Technology, NOVA University Lisbon

Nuno Azevedo
Associate Professor, LEPABE-Faculty of Engineering of Porto, University of Porto

MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
September, 2024

Development of a database of spatial structure in Biofilms

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.0.2) [1].

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Rafael Costa, and co-advisers, Pedro Barahona and Nuno Azevedo, for their invaluable guidance, support, and encouragement throughout this research. Their expertise and insights have been instrumental in shaping the direction of this project, and I am profoundly grateful for the opportunity to learn from them.

I would also like to extend my thanks to the faculty and staff of the Computer Science Department at Nova School Of Science and Technology, who provided the resources and environment necessary for this work. Additionally, I acknowledge the support of the Department of Chemical Engineering (@LEPABE) at University of Porto (FEUP) for their assistance, which made this research possible.

On a personal note, I would like to thank my friends and colleagues for their companionship and support during this journey. A special thank you goes to my family, whose love, patience, and unwavering belief in me provided the foundation I needed to persevere. To my parents, grandparents, and sister, your encouragement has meant the world to me, and this achievement would not have been possible without you. I would also like to extend my gratitude to my coworkers at wTVision for their continuous support and motivation, which helped me stay focused and driven throughout this process.

Abstract

Biofilms, complex microbial communities adhering to surfaces, play a crucial role in numerous natural and industrial processes. Understanding the intricate architecture and dynamics of biofilms is essential for tackling challenges ranging from medical infections to environmental remediation. Since biofilms are responsible for more than 80 percent of all microbial infections in the body and play a role in the development of antibiotic resistance, their existence has significant implications for human health. There is a wealth of data in this sector thanks to extensive research. For the field to advance, this knowledge must be effectively managed and used.

This work presents the development of a comprehensive database for biofilm structures, aiming to facilitate data organization, integration, and analysis in the field of biofilm research. The objective is to represent the biofilm architecture, composition and associated metadata, so there is a framework to store multiple biofilm datasets. Using modern database technologies, efficient data retrieval is enabled and also the support of advanced querying capabilities, allowing researchers to explore and study the accumulated biofilm data. Furthermore, the web-based database incorporates mechanisms for data sharing, collaboration, and growth, making it a community-driven approach to biofilm research.

The database also offers visualization tools to better understand the data and also supports the integration of computational algorithms for biofilm analysis and recognition. We expect that this database will facilitate the widespread access of biofilms data do understand the biofilm formation and identify numerically the different components of the biofilm in three dimensions.

Keywords: biofilms, database, machine learning, data management, FAIR data, data visualization

Resumo

Biofilmes, comunidades microbianas complexas que aderem a superfícies, desempenham um papel crucial em inúmeros processos naturais e industriais. Compreender a arquitetura intríseca e as dinâmicas dos biofilmes é essencial para enfrentar desafios que vão desde infeções médicas até à remediação ambiental. Dado que os biofilmes são responsáveis por mais de 80 porcento de todas as infeções microbianas no corpo e desempenham um papel no desenvolvimento da resistência aos antibióticos, a sua existência tem implicações significativas para a saúde humana. Há uma vasta quantidade de dados neste setor graças à extensa pesquisa realizada. Para que o campo avance, este conhecimento deve ser gerido e utilizado de forma eficaz.

Este trabalho apresenta o desenvolvimento de uma base de dados abrangente sobre estruturas de biofilmes, com o objetivo de facilitar a organização, integração e análise de dados na área de pesquisa sobre biofilmes. O objetivo é representar a arquitetura do biofilme, a sua composição e os metadados associados, criando uma estrutura para armazenar múltiplos conjuntos de dados de biofilmes. Utilizando tecnologias modernas de bases de dados, é possível uma recuperação eficiente de dados e também o suporte a capacidades avançadas de consulta, permitindo que os investigadores explorem e estudem os dados acumulados sobre biofilmes. Além disso, a base de dados baseada na web incorpora mecanismos para partilha de dados, colaboração e crescimento, tornando-se numa abordagem orientada pela comunidade para a pesquisa sobre biofilmes.

A base de dados também oferece ferramentas de visualização para melhor compreender os dados e suporta a integração de algoritmos computacionais para a análise e reconhecimento de biofilmes. Esperamos que esta base de dados facilite o amplo acesso aos dados de biofilmes para compreender a formação de biofilmes e identificar numericamente os diferentes componentes do biofilme em três dimensões.

Palavras-chave: Biofilmes, Base de Dados, Gestão de dados, Dados FAIR, Visualização de dados

Contents

List of Figures

List of Tables

1

Introduction

1.1 Context

Biofilms, which are microbial communities that stick to surfaces and are immersed in a self-made matrix, are becoming more widely recognised as important participants in a variety of natural and artificial ecosystems [2]. Biofilms have proven their astonishing capacity to endure and flourish in a variety of settings, from medical settings to industrial processes [3]. This presents special problems for research, management, and control. It is essential to comprehend the intricate structure and dynamic behaviour of biofilms in order to understand their mechanisms of development, discover the variety of ecological functions they play, and create efficient mitigation measures for their negative consequences [4].

Historically, experiments, microscope imaging, and microbiological culturing methods have been used in biofilm studies [5]. The sheer amount and complexity of biofilm-related data gathered over the years have presented obstacles in data management, integration, and analysis, even though these approaches have given vital insights into biofilm production and composition [6]. There is an urgent need for a centralised and comprehensive database that can operate as an invaluable resource for researchers and practitioners in the field as biofilm research increasingly embraces interdisciplinary techniques and larger-scale studies.

Databases play an essential position in organizing, storing, and retrieving widespread quantities of statistics in an established and efficient manner [7]. They are extensively utilized in numerous fields, which includes clinical research, healthcare, to name a few [8]. In the context of biofilm studies, the improvement of a committed database can substantially enhance our know-how of biofilm systems and their implications.

1.2 Motivation

Advancements in biofilm studies have established the great impact of biofilms on various ecosystems, human health, and and industrial processes. Understanding the tricky

structure and dynamic behavior of biofilms is vital for comprehending their mechanisms of improvement, exploring their ecological capabilities, and growing powerful mitigation techniques. However, the sheer extent and complexity of biofilm-associated data gift challenges in data management, integration, and analysis. As biofilm studies increasingly embraces interdisciplinary techniques and larger-scale studies, there's a pressing need for a centralized and comprehensive repository dedicated to biofilm systems. Furthermore, this project was accomplished in collaboration with the Department of Chemical Engineering (@LEPABE), University of Porto (FEUP) in the context of the e.Biofilm project (https://ebiofilm.fe.up.pt/).

1.3 Objectives

The aim of this thesis project is to address this need by proposing the development of a dedicated web-based database for biofilm structures. Such a database would not only provide a platform for efficient storage and retrieval of biofilm-related data but also facilitate the integration of diverse experimental datasets and foster collaboration among researchers in this field. By capturing essential information on biofilm architecture, composition, and associated metadata, the database will enable researchers to explore and analyse data in a standardized manner, accelerating the pace of discoveries and advancements in biofilm research.

The database design incorporates modern database technologies, allowing for efficient data organization, retrieval, and querying. Furthermore, it provides visualization tools to facilitate data exploration and analysis, empowering researchers to uncover patterns, correlations, and trends within biofilm structures. To ensure the database's usefulness and relevance, it will be designed with flexibility and extensibility in mind, allowing for the incorporation of emerging data types and accommodating advancements in biofilm research methodologies.

By establishing a centralized resource for biofilm structures, this thesis aims to foster collaboration, knowledge sharing, and data-driven approaches in biofilm research. The availability of a comprehensive database will enable researchers to test hypotheses, validate models, and explore new avenues for biofilm control and management. Additionally, the database will provide a valuable resource for educational purposes, enabling students and researchers to access curated datasets and learn from established biofilm research findings.

As such, the aims of this thesis are the following:

- Build a free-access database for biofilms research;
- Propose a Web-based component to visualize, analyze and add data do this database;
- Development and application of predictive models to automatically detect which bacteria are in the biofilm.

1.4 Document Structure

The current document is organized into seven chapters:

- **Chapter 1: Introduction** introduces the dissertation's context and outlines the motivation, objectives, and overall scope of the research project.
- **Chapter 2: Background** provides an overview of key concepts necessary for understanding the thesis. It covers biofilms, databases, and machine learning principles, laying the foundation for the research conducted.
- **Chapter 3: Related Work** summarizes relevant literature and existing research in biofilm studies, databases, and machine learning applications in the context of biofilm analysis. This chapter provides insights into current gaps and how this thesis aims to address them.
- **Chapter 4: Development of the Biofilm Database and Website** details the methodology used in developing the biofilm database and the website. It includes the design, structure, and technology stack used for building the database, as well as the approach to ensure data management, integration, and accessibility.
- **Chapter 5: Development of the Front-End Interface** describes the creation of the web-based interface that allows users to interact with the biofilm database. It includes details on the user experience design, technologies used for the front-end, and how the interface supports collaboration and data analysis.
- **Chapter 6: Development and Optimization of Convolutional Neural Networks for Bacterial Image Classification** discusses the machine learning models developed as part of the project. This chapter explains how convolutional neural networks (CNNs) are used to classify microbial images within the biofilm dataset, and how the models were trained and optimized.
- **Chapter 7: Conclusion and Future Work** summarizes the outcomes of the thesis, highlighting the development of the biofilm database and its contributions to the field. It discusses the platform's impact on biofilm research through enhanced data management, image processing, and machine learning integration. Additionally, this chapter outlines several future directions to expand the platform's capabilities, including improvements to predictive models, support for additional file formats, and better user interfaces and visualization tools. The potential for advancing biofilm research by incorporating new techniques and adhering to FAIR data principles is also explored.

2

Background

This chapter introduces key concepts, algorithms, and techniques that are fundamental to understanding both the subject matter and the research carried out in this thesis. The background information provided here is meant to give the reader a thorough understanding of the essential principles required to appreciate the subsequent chapters.

This chapter is divided into three sections: Section 2.1 provides an extensive overview of biofilms, Section 2.2 introduces key concepts regarding databases and data management, and Section 2.3 discusses machine learning principles, with a focus on how these concepts apply to biofilm research.

2.1 Biofilms

Biofilms are structured communities of microbial cells that are attached to a surface or interface and are surrounded by a self-produced extracellular polymeric matrix [9]. Biofilm formation is an ubiquitous phenomenon in nature, where microbial cells come together to form a community, displaying collective behavior and enhanced survival strategies compared to their planktonic (free-living) counterparts. These multicellular structures protect resident bacteria from environmental stresses, such as antibiotics, desiccation, and immune responses, and are involved in a wide variety of industrial, environmental, and medical settings [10].

2.1.1 Biofilm Structure and Composition

The biofilm matrix, often called the extracellular polymeric substance (EPS), is primarily composed of polysaccharides, proteins, lipids, and extracellular DNA (eDNA). This matrix provides both structural integrity and biochemical protection for the biofilm community. EPS acts as a scaffold that holds the biofilm together and facilitates the formation of water channels, allowing nutrients and waste products to be transported within the biofilm [10]. The matrix also acts as a barrier to antimicrobial agents, making biofilms notoriously difficult to eradicate.

Interestingly, the composition of the EPS varies significantly depending on the microbial species involved, the surface to which the biofilm is attached, and the environmental conditions under which the biofilm forms. For example, biofilms formed on medical implants by *Staphylococcus aureus* or *Pseudomonas aeruginosa* have a different composition compared to those formed in natural environments, such as marine ecosystems [11]. Understanding the structural diversity of biofilms is crucial for developing targeted strategies to control or manipulate biofilms in various contexts.

2.1.2 Biofilm Development Stages

Biofilm formation is a dynamic process that occurs in multiple stages. Although the specific mechanisms of biofilm development may vary depending on the microorganism and environmental conditions, the general process can be divided into the following stages [12]:

- **Attachment:** The initial step involves the adhesion of planktonic bacteria to a surface. Bacterial appendages, such as *flagella*, *pili*, or *fimbriae*, play a crucial role in mediating this attachment. The surface characteristics, including roughness, hydrophobicity, and charge, also influence the adhesion process.

- **Microcolony Formation:** After attachment, bacterial cells begin to multiply and form microcolonies. At this stage, bacteria start producing the extracellular matrix (EPS), which stabilizes the growing microcolonies.

- **Maturation:** As the biofilm grows, it undergoes structural maturation, forming complex three-dimensional architectures with channels that allow nutrient flow and waste removal. During this phase, bacteria can communicate via quorum sensing, a cell-to-cell signaling mechanism that coordinates biofilm development.

- **Dispersion:** The final stage of biofilm development is dispersion, where some cells break away from the biofilm to revert to the planktonic state and colonize new surfaces. This dispersal stage is essential for the spread of biofilms and their ability to infect new environments [11].

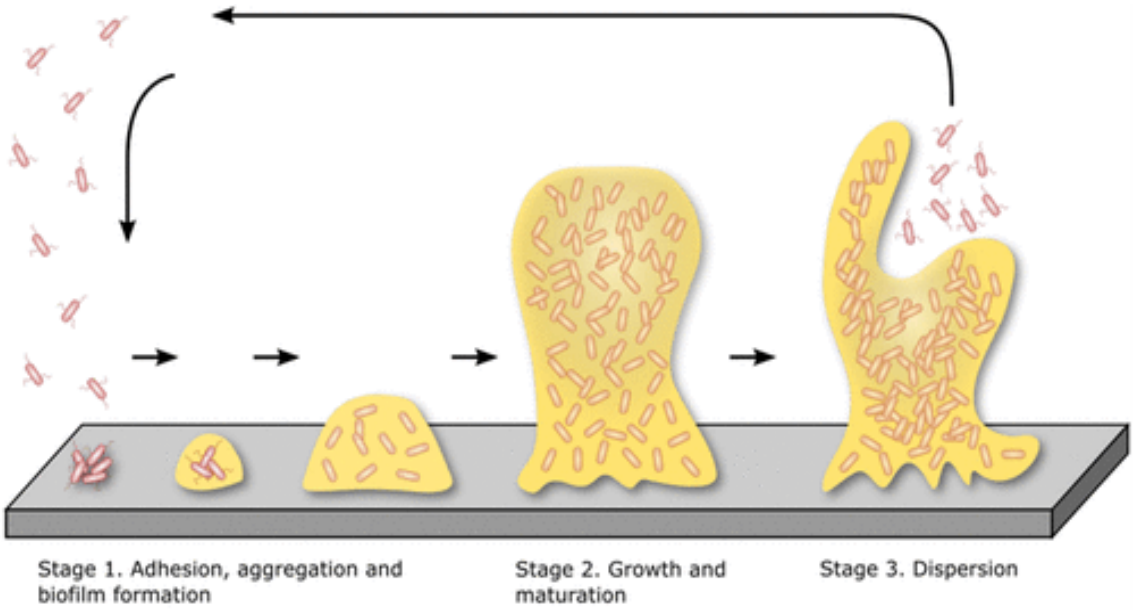

Figure 2.1: Key stages of biofilm development and life cycle. (1) Adhesion, aggregation, and biofilm formation. Bacterial appendages promote cell adhesion. (2) Biofilm growth and maturation. The extracellular polymeric substances (EPSs) stabilize the biofilm. (3) Dispersion. Planktonic bacteria are released for further colonization(adapted from [12]).

2.1.3 Biofilms in Medicine and Industry

Biofilms play an important role in both medicine and industry. In clinical settings, biofilms are often associated with persistent infections, particularly those involving medical devices in the home such as catheters, pacemakers, and prosthetic joints [11]. The EPS matrix shields bacteria from antibiotic treatments and the host's immune system, making infections difficult to treat. For example, catheter-associated urinary tract infections (CAUTIs) are often caused by multispecies biofilms containing *Escherichia coli* and *Enterobacter sp.*, which adhere to catheter surfaces and resist antibiotic penetration [13].

In industrial settings, biofilms are responsible for biofouling, the undesirable accumulation of microorganisms on surfaces, such as ship hulls, pipelines, and water treatment systems. This can lead to corrosion, reduced efficiency, and increased operational costs. However, biofilms can also be harnessed for beneficial applications, such as in bioremediation, wastewater treatment, and bioenergy production, where microbial communities in biofilms are used to degrade pollutants or generate energy [10].

Combination ID	Color ID	Species
C1		*Escherichia coli*
C2		*Proteus mirabilis*
C3		*Enterococcus faecalis*
C4		*Candida albicans*
C5		*E. coli and P. mirabilis*
C6		*E. coli and E. faecalis*
C7		*E. coli and C. albicans*
C8		*E. coli and P. mirabilis and E. faecalis*
C9		*E. coli and P. mirabilis and C. albicans*
C10		*E. coli and E. faecalis and C. albicans*
C11		All 4

Figure 2.2: ID designations for different biofilm species combinations, as seen in Figure 2.3.

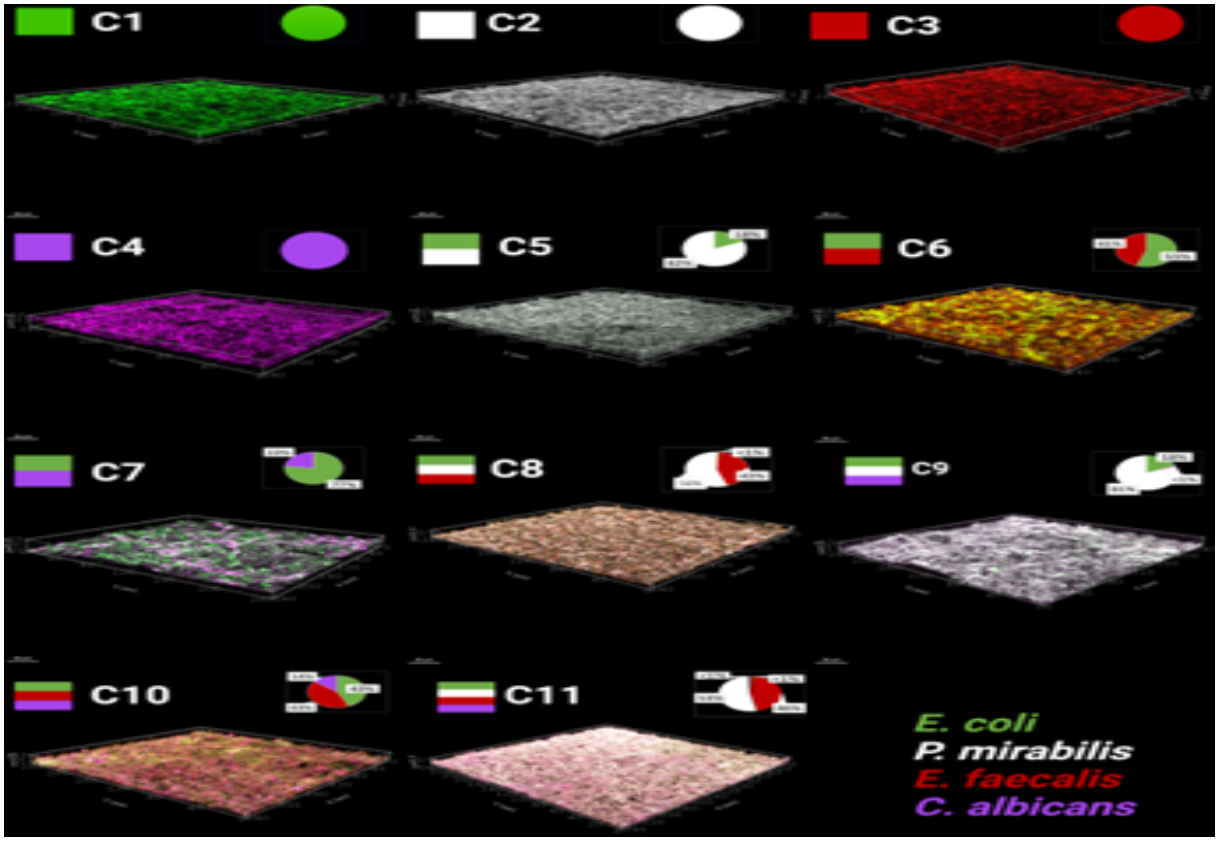

Figure 2.3: Spatial organization of biofilms per combination. Each species is color-tagged: *Escherichia coli* (green), *Proteus mirabilis* (white), *Enterococcus faecalis* (red), and *Candida albicans* (purple). Species abundance is based on plate count data [13].

2.1.4 Multispecies Biofilms

Many biofilms are polymicrobial in nature, which means that more than one species of microorganism is present. These multispecies biofilms show complex interactions between species, including competition for resources, synergistic relationships, and horizontal gene transfer. In some cases, the presence of multiple species can enhance the resistance of the biofilm to environmental stresses and antimicrobials [13].

One prominent example of multi-species biofilms is in the human oral cavity, where microbial communities contribute to the formation of dental plaque and tooth decay. Another example is biofilms found in chronic wounds, which are composed of diverse microbial communities that hinder healing and are difficult to treat.

2.2 Databases

In the modern era of research, databases have become indispensable tools for organizing, storing, and retrieving large amounts of data. A database is essentially an organized collection of data that can be accessed, managed, and updated efficiently. In scientific research, databases serve as centralized repositories for data generated from experiments, simulations, and observations. These repositories allow researchers to share data, perform complex queries, and perform large-scale analyses across multiple datasets.

2.2.1 Types of Databases

Over time, various database models have been developed to handle specific types of data and use cases. The following are some of the most commonly used database models:

2.2.1.1 Relational Databases

Relational databases store data in a structured format using tables with predefined columns and data types. Each table represents an entity (e.g., customer, product, experiment), and relationships between tables are defined through primary and foreign keys. Relational databases are particularly well suited for applications where data integrity and consistency are critical, such as in banking, e-commerce, and scientific research [14].

2.2.1.2 NoSQL Databases

NoSQL databases were developed to address the limitations of relational databases in handling large volumes of unstructured or semi-structured data. These databases are designed for horizontal scalability, high performance, and flexibility. NoSQL databases are commonly used in big data applications, such as social media, IoT, and genomics [15]. Unlike relational databases, NoSQL databases do not require a fixed schema, allowing them to store various types of data, including key-value pairs, documents, graphs, and column-family data.

2.2.1.3 Object-Oriented Databases

Object-oriented databases are designed to store data as objects, which are instances of classes defined by object-oriented programming languages. This approach allows for the representation of complex data types, such as images, multimedia, and custom data structures. Object-oriented databases are commonly used in applications that require complex data modeling, such as computer-aided design (CAD) and multimedia systems.

2.2.2 Key Features of Modern Databases

Modern databases provide several features that make them indispensable tools for research and industry:

- **Data Integrity:** Databases enforce rules and constraints to ensure the accuracy and consistency of stored data. This is particularly important in scientific research, where data integrity is crucial for reproducibility and validation of results.
- **Scalability:** With the rise of big data, databases need to handle large volumes of data efficiently. Distributed databases, which store data across multiple nodes, allow for horizontal scaling, enabling databases to handle increasing amounts of data without compromising performance.
- **Querying:** Modern databases support powerful query languages, such as SQL for relational databases and specialized query languages for NoSQL databases. These querying capabilities allow researchers to retrieve specific subsets of data, perform aggregations, and execute complex analytical queries.
- **Visualization:** Some databases integrate visualization tools that allow users to explore and analyze data visually. For example, biofilm databases may include tools for visualizing microbial communities, biofilm structure, or temporal changes in biofilm growth.

2.2.3 The Role of Databases in Biofilm Research

In biofilm research, databases play a critical role in organizing, sharing, and analyzing experimental data. A well-designed biofilm database serves as a centralized repository for storing various types of biofilm-related information, including experimental results, genomic data, and environmental conditions [16]. By capturing key metadata associated with biofilm formation (e.g., species involved, growth conditions, imaging data), databases enable researchers to compare biofilm characteristics across different studies and experimental setups.

Biofilm databases can also incorporate machine learning algorithms for predictive modeling, allowing researchers to make informed predictions about biofilm formation, antimicrobial resistance, or biofilm-related pathogenicity.

2.3 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence (AI) that focuses on the development of algorithms capable of learning from and making predictions based on data. Machine learning is becoming increasingly important in biofilm research, where it is used for tasks such as predicting biofilm growth, classifying biofilm phenotypes, and identifying biofilm-related genes. The following sections introduce the key concepts in machine learning and explain how these concepts are applied to biofilm research.

2.3.1 Supervised Learning

Supervised learning is a type of machine learning in which the algorithm is trained on labeled data, where the input-output pairs are known. The goal of supervised learning is to map input data to the correct output, based on the patterns learned from the training data. In biofilm research, supervised learning can be used to predict biofilm formation based on input variables such as nutrient concentration, microbial species, and environmental conditions [17].

Supervised learning models can be divided into two main categories:

- **Classification:** Classification models assign input data to predefined categories. In biofilm research, classification algorithms can be used to classify microbial strains based on their biofilm-forming potential.
- **Regression:** Regression models predict continuous values. In biofilm research, regression algorithms can be used to predict the rate of biofilm growth or the concentration of extracellular polymeric substances (EPS) in biofilms [18].

2.3.2 Unsupervised Learning

Unsupervised learning involves training machine learning models on unlabeled data, where the goal is to discover hidden patterns or relationships in the data. In biofilm research, unsupervised learning can be used to group microbial species according to their biofilm formation behavior or to identify patterns in biofilm phenotypes across different environmental conditions.

Two common types of unsupervised learning are:

- **Clustering:** Clustering algorithms group data points into clusters based on similarities. In biofilm research, clustering can be used to identify microbial communities that share similar biofilm-forming characteristics.
- **Dimensionality Reduction:** Dimensionality reduction techniques, such as principal component analysis (PCA), reduce the number of variables in a dataset while preserving important information. This is useful for visualizing high-dimensional biofilm data or for simplifying complex datasets.

2.3.3 Convolutional Neural Networks (CNNs)

2.3.3.1 Overview of CNNs

Convolutional Neural Networks (CNNs) are a class of deep learning models that have been particularly effective for image-related tasks. Inspired by the structure and functionality of the human visual cortex, CNNs were initially developed to mimic the way biological systems process visual information [19]. Neurons in the visual cortex are activated by specific features of an image, such as edges, textures, and colors. This biological inspiration led to the development of the first CNNs in the 1980s, notably by Yann LeCun, who introduced the LeNet architecture for digit recognition tasks such as the MNIST handwritten digit dataset [20]. Since then, CNNs have evolved significantly and have become one of the most powerful tools for a wide variety of tasks, including image classification, object detection, image segmentation, and more [19, 21].

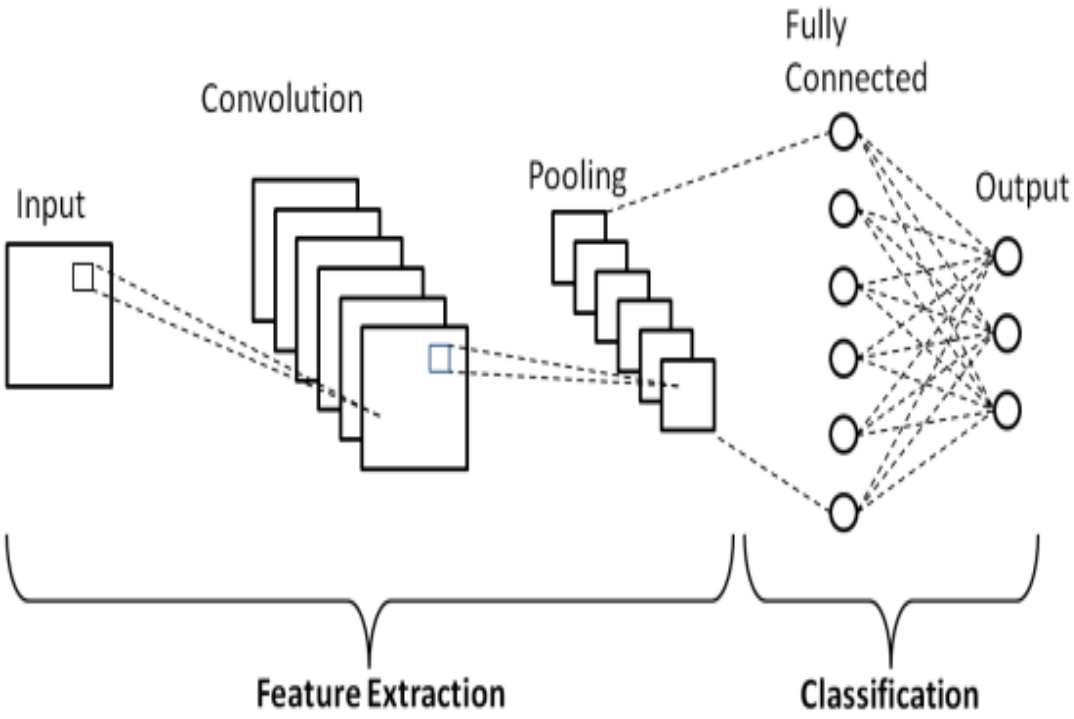

Figure 2.4: Schematic diagram of a basic convolutional neural network (CNN) architecture

CNNs are structured as multi-layered networks that progressively transform raw image data into high-level features that can be used for classification. The ability of CNNs to capture hierarchical feature representations—ranging from low-level edges to high-level object parts—has made them highly effective for complex image classification tasks [21, 22]. The primary components of a CNN include:

- **Convolutional Layers**: These layers form the backbone of a CNN and are responsible for feature extraction. Filters, also known as kernels, slide over the input image to

compute feature maps by applying a convolution operation. Each filter is designed to detect specific features within the image, such as edges or textures [22]. As the network deepens, the convolutional layers are able to detect increasingly complex patterns. In the initial layers, filters capture simple features like edges or corners, while in deeper layers, they detect more abstract features such as textures, shapes, and even parts of objects [20].

- **Pooling Layers**: Pooling layers are used to reduce the spatial dimensions of the feature maps produced by the convolutional layers. This dimensionality reduction helps to lower the computational load, making the network more efficient while preserving essential spatial information [22]. Max pooling is the most commonly used pooling method, where the maximum value from each region of the feature map is selected, ensuring that the most prominent features are retained [20]. Pooling also makes the model more robust to minor translations and distortions in the input image, which is particularly important for bacterial images, where variations in orientation and positioning can occur.

- **Fully Connected Layers**: After several convolution and pooling operations, the resulting high-dimensional feature maps are flattened and passed into fully connected layers. These layers function like a traditional neural network, where each neuron is connected to every other neuron in the next layer [20]. The fully connected layers interpret the features extracted by the convolutional layers and make predictions based on them. The final fully connected layer is typically followed by a softmax function to produce class probabilities for multi-class classification tasks [22].

- **Activation Functions**: Activation functions introduce non-linearity into the network, allowing it to model complex, non-linear relationships in the data. The Rectified Linear Unit (ReLU) activation function is the most widely used in CNNs due to its simplicity and effectiveness [20, 23]. ReLU effectively removes negative values from the feature maps, retaining only positive activations. Other activation functions, such as the sigmoid or tanh functions, are used less frequently due to their tendency to suffer from the vanishing gradient problem, which can hinder learning in deep networks [20].

The hierarchical structure of CNNs, with their ability to learn increasingly complex features as depth increases, makes them particularly well-suited for image classification tasks. In the case of bacterial image classification, bacteria often exhibit subtle differences in shape, size, and structure that are difficult to distinguish using traditional machine learning techniques, which rely on hand-crafted features. CNNs, by contrast, can learn these subtle differences directly from the raw image data, making them an ideal choice for this task.

2.3.3.2 Why Use CNNs in Biofilm Analysis?

CNNs offer several key advantages over traditional machine learning models for image classification tasks, particularly in the context of bacterial image classification:

- **Automatic Feature Extraction**: One of the most significant benefits of CNNs is their ability to automatically learn and extract relevant features directly from raw image data. Traditional machine learning models often rely on manual feature engineering, which can be time-consuming and may not capture all relevant aspects of the data. CNNs, on the other hand, are capable of learning spatial features and hierarchies without requiring human intervention, which is particularly useful for complex tasks like bacterial image classification.

- **Capturing Spatial Hierarchies**: CNNs excel at capturing spatial relationships within images, learning features at multiple levels of abstraction. In the context of bacterial classification, where differences between bacterial species can be subtle, CNNs are particularly effective at distinguishing between these classes by learning to detect low-level features such as edges, as well as higher-level patterns like textures and shapes. This ability to capture spatial hierarchies allows CNNs to outperform traditional machine learning models that rely on pre-defined, hand-crafted features.

- **State-of-the-Art Performance**: CNNs have consistently demonstrated state-of-the-art performance in various image classification tasks, outperforming traditional machine learning models in terms of accuracy, precision, and generalization. For tasks involving bacterial image classification, CNNs are capable of generalizing well to new, unseen data due to their ability to learn robust and representative features from the training data.

- **Scalability and Adaptability**: CNNs are highly scalable and can be adapted to a wide range of applications by adjusting the depth, width, and number of layers in the network. Pre-trained models, such as VGG16, ResNet, and Inception, can be fine-tuned for specific tasks using transfer learning, which reduces the need for large amounts of training data. This adaptability makes CNNs applicable not only to bacterial image classification but also to a variety of other tasks, such as medical image analysis, facial recognition, and even natural language processing when applied to tasks like handwriting recognition.

2.3.3.3 Transfer Learning in CNNs

One powerful technique that has greatly enhanced the effectiveness of CNNs, particularly in specialized domains like bacterial image classification, is **transfer learning**. Transfer learning allows a model pre-trained on a large dataset (often general-purpose datasets such as ImageNet) to be fine-tuned for a specific task with relatively small amounts of

task-specific data. This is especially useful in biofilm or bacterial image classification, where obtaining large labeled datasets can be challenging.

How Transfer Learning Works In a typical transfer learning setup, the early layers of a pre-trained CNN (e.g., VGG16, ResNet, or Inception) are retained, as they tend to capture generic low-level features such as edges, textures, and shapes that are useful across various image classification tasks. The later layers, which are more specific to the original task the model was trained on, are often fine-tuned or replaced with new layers tailored to the target task [22].

For bacterial classification, we employ transfer learning by leveraging pre-trained models on large datasets, then adjusting the fully connected layers and output layers to reflect the specific categories of bacteria we aim to classify. This process not only saves time and computational resources but also boosts model performance, as the pre-trained models already contain valuable feature representations that can be adapted to the bacterial domain.

Benefits of Transfer Learning

- **Reduced Training Time**: Since the base model has already learned generic features from a large dataset, transfer learning significantly reduces the time and computational resources required to train the model on the target task.
- **Improved Accuracy with Limited Data**: Transfer learning is particularly useful when the amount of labeled data for the specific task is limited. In bacterial image classification, acquiring large, labeled datasets can be difficult, but transfer learning enables high accuracy even with smaller datasets by leveraging knowledge from the pre-trained model.
- **Better Generalization**: Since the base model has been trained on diverse datasets, it often generalizes better to new, unseen data in the target domain. For example, models pre-trained on large-scale datasets like ImageNet are well-equipped to recognize general image features, which are transferable to bacterial image classification.

Incorporating transfer learning into our bacterial classification model not only improves performance but also makes the system more efficient and adaptable to new classification tasks. This approach ensures that our model leverages state-of-the-art techniques for high accuracy and robustness in identifying bacteria in biofilms.

In summary, this chapter has provided a detailed overview of the essential background concepts relevant to this thesis. Biofilms are complex microbial communities with significant implications for both medicine and industry, and their study requires a multidisciplinary approach. Databases play a critical role in organizing and analyzing biofilm data, while machine learning offers powerful tools for predicting biofilm behavior and identifying patterns in biofilm-related data. These foundational concepts will be built

upon in the following chapters, where we will explore specific applications of machine learning and database design in the study of biofilms.

3

Related Work

The related work chapter aims to provide a comprehensive overview of existing research, studies, and literature related to the topic of this thesis. The goal is to place the current work in the context of broader scientific dialogue by discussing relevant databases, computational methods, and the progress made in biofilm research. The scope of biofilm research has been rapidly expanding, with a particular emphasis on machine learning techniques, and various bioinformatics databases have been instrumental in supporting these advancements.

The discussion begins with the introduction of key biofilm databases that have supported this research and continues with a critical analysis of their roles in advancing the understanding of biofilms, with particular emphasis on their contributions to biofilm formation, prediction, and control. A summary of the notable biofilm repositories is presented in Table 3.1, followed by an in-depth review of each.

Table 3.1: Overview of all currently available biofilm databases.

Name	Year	Website	Reference
Quorumpeps	2012	http://quorumpeps.ugent.be/	Wynendaele et al. [24]
BiofOmics	2012	http://www.biofomics.org/	Lourenço et al. [25]
BaAMPs	2015	http://www.baamps.it/	Di Luca et al. [26]
BSD	2020	https://biosim.pt/biofilms/	Magalhães et al. [27]
PATRIC	2015	https://www.patricbrc.org/	Wattam et al. [28]
VFDB	2004	http://www.mgc.ac.cn/VFs/	Chen et al. [29]

3.1 Quorumpeps

Quorumpeps, developed by Ghent University, is one of the pioneering databases specifically focused on quorum sensing (QS) peptides, a crucial aspect of microbial communication that regulates biofilm formation and virulence. Introduced in 2012, Quorumpeps offers a systematic repository of quorum sensing peptides, which are small signalling

molecules secreted by bacteria to coordinate collective behavior, including biofilm formation, virulence factor expression, and antibiotic resistance.

The database has been indispensable in the study of QS pathways, as it provides a comprehensive collection of peptide structures, along with their activity and physicochemical properties. The ability to search for peptides based on various criteria such as chemical structure and biological activity has made Quorumpeps a valuable resource for researchers attempting to disrupt quorum sensing as a potential strategy for biofilm control [24].

Quorumpeps is also highly cited in studies that explore the chemical space of quorum sensing peptides, providing critical insights into the diversity of QS peptides across different microbial species. Additionally, the integration of literature references with each peptide entry allows users to quickly access the associated research, making it a convenient platform for cross-referencing experimental findings.

Despite its utility, Quorumpeps has its limitations. The primary drawback is that it focuses solely on peptides and does not provide information on other quorum sensing molecules like autoinducers, which play significant roles in QS systems such as *Pseudomonas aeruginosa*'s las and rhl systems. Expanding the database to include these molecules could greatly enhance its relevance to a wider range of biofilm research.

3.2 BiofOmics

Launched in 2012, BiofOmics was designed as a collaborative platform to promote data exchange across biofilm research laboratories globally. The database was developed by the Institute for Biotechnology and Bioengineering at the University of Minho, in conjunction with other Portuguese institutions. It primarily aims to serve as a hub for standardizing and sharing biofilm experimental data.

BiofOmics has a distinct focus on standardizing the collection and reporting of biofilm-related data, which is critical for ensuring that biofilm experiments can be replicated and compared across different research groups. This is particularly important in biofilm research, where variations in experimental conditions—such as microbial strains, growth media, and environmental factors—can lead to significantly different outcomes. BiofOmics attempts to address this by offering a three-step protocol that allows researchers to describe, standardize, and upload their experimental data [25].

One of the platform's strengths is its capability for data interoperability. Researchers can submit their datasets in a standardized format, ensuring that the information is both accessible and comparable. This not only facilitates better collaboration but also accelerates research progress by enabling meta-analyses and large-scale studies that require aggregated data.

However, despite its ambitions, BiofOmics has struggled with technical issues, particularly with website functionality. At the time of writing, the website has experienced

frequent downtimes, limiting access to its rich dataset. If the platform's technical stability can be improved, BiofOmics could play a more central role in biofilm research.

The data submission process in BiofOmics is one of its most innovative features, allowing for detailed characterization of the experimental conditions, including the microbial species, biofilm-forming devices, and growth conditions. This level of detail ensures that the experimental context is thoroughly documented, which is vital for reproducibility. The platform also promotes open access, encouraging researchers to freely share their findings with the global community, thus fostering greater collaboration.

3.3 BaAMPs

BaAMPs, which stands for Biofilm Active Antimicrobial Peptides, was established in 2015 with a focus on antimicrobial peptides (AMPs) that target biofilms. Antimicrobial peptides are short proteins that exhibit antibacterial properties, and they have emerged as one of the most promising strategies for controlling biofilm-related infections due to their ability to penetrate biofilms and disrupt microbial cells [26].

BaAMPs was developed in response to the growing need for a centralized repository that consolidates experimental data on AMPs with specific biofilm inhibition or eradication properties. This data includes peptide sequences, chemical modifications, and experimental details such as concentration, biofilm inhibition percentage, and the biofilm-forming organisms tested. BaAMPs also links each entry to its corresponding scientific publication, providing an easily navigable interface for users to access relevant research.

One of the standout features of BaAMPs is its user submission mechanism. Researchers can contribute new peptide sequences and experimental data through a straightforward upload system. Submitted data undergoes a validation process by the database administrators, ensuring that only high-quality, peer-reviewed information is made available to the public. This contributes to the reliability and credibility of the data hosted on BaAMPs, making it a trusted resource for researchers exploring biofilm inhibition strategies.

From a technical perspective, BaAMPs incorporates several useful search functions that allow users to filter peptides based on criteria such as peptide sequence, biofilm inhibition percentage, and target organism. This makes it easier to identify peptides that might be effective against specific biofilm-forming bacteria, which is particularly useful for researchers developing targeted therapies for biofilm-associated infections.

Despite its strengths, BaAMPs does have some limitations. As the database is relatively new, its coverage of AMPs is still limited, and it lacks extensive data on synthetic or modified AMPs. Additionally, while BaAMPs is useful for identifying peptides that have been tested in vitro, it does not provide information on the efficacy of these peptides in vivo, which is crucial for the development of clinical treatments.

3.4 BSD

The Biofilms Structural Database (BSD) is a relatively new addition to the field, launched in 2020. BSD distinguishes itself from other biofilm-related databases by focusing on the atomic structures of biofilm-associated proteins. This database compiles structural information from the Protein Data Bank (PDB) and scientific literature, providing insights into the mechanisms of biofilm formation, resistance, and dispersal at the molecular level [27].

BSD's primary aim is to catalog proteins involved in biofilm processes, including those that participate in quorum sensing, motility, and extracellular polymeric matrix (EPS) formation. The database also includes proteins linked to biofilm dispersion, a critical stage in the biofilm lifecycle that is often targeted for therapeutic intervention. By understanding the structure of these proteins, researchers can develop small molecules or peptides that inhibit biofilm formation or promote biofilm disassembly.

A key strength of BSD is its integration of advanced visualization tools. The database provides interactive models of protein-ligand interactions, allowing users to explore the binding mechanisms that underlie biofilm-related processes. These visualization tools are particularly useful for researchers involved in drug discovery, as they enable the identification of potential binding sites for novel biofilm inhibitors.

BSD also links to several external databases, such as ChEMBL, BindingDB, and UniProt, enhancing its utility for researchers who require access to additional bioinformatics resources. By combining structural data with functional annotations, BSD offers a comprehensive platform for the study of biofilm-associated proteins.

One of the challenges faced by BSD is the manual curation of data, which can limit the speed at which new entries are added. However, the rigorous validation process ensures that the data is highly reliable. In the future, BSD may benefit from automated data-mining techniques that could expedite the inclusion of new biofilm-related structures.

3.5 PATRIC

The Pathosystems Resource Integration Center (PATRIC) was developed as a comprehensive bioinformatics resource for bacterial pathogens, and while it is not exclusively focused on biofilms, it has become a valuable tool for biofilm research. PATRIC integrates genomic, transcriptomic, and proteomic data for a wide range of bacterial species, many of which are known biofilm formers [28].

PATRIC provides access to a vast array of bacterial genomes, enabling researchers to study the genetic factors that contribute to biofilm formation. By leveraging PATRIC's extensive collection of genomic data, researchers can identify biofilm-associated genes and investigate how these genes are regulated under different environmental conditions. This is particularly important for understanding the role of biofilms in infectious diseases, where biofilm formation can contribute to increased virulence and antibiotic resistance.

In addition to genomic data, PATRIC offers bioinformatics tools that allow researchers to conduct comparative genomics, functional annotation, and metabolic pathway analysis. These tools are instrumental in dissecting the molecular mechanisms that underlie biofilm formation and maintenance. PATRIC also integrates data from other bioinformatics resources, such as the National Center for Biotechnology Information (NCBI), providing users with a comprehensive platform for microbial research.

While PATRIC's primary focus is on pathogens, its datasets and tools are widely applicable to non-pathogenic biofilm-forming bacteria as well. This makes PATRIC a versatile resource for biofilm research across a range of microbial systems.

3.6 VFDB

The Virulence Factor Database (VFDB), developed in 2004, focuses on virulence factors, which are molecules produced by bacteria to enhance their ability to cause disease. Many virulence factors are directly linked to biofilm formation, as biofilms can protect bacterial communities from the host immune system and increase their resistance to antibiotics [29].

VFDB provides detailed information on virulence factors from a wide range of bacterial pathogens, including *Pseudomonas aeruginosa*, *Staphylococcus aureus*, and *Escherichia coli*, all of which are known for their ability to form biofilms. The database includes genetic and functional annotations for each virulence factor, as well as information on its role in the infection process.

One of the strengths of VFDB is its integration of genomic and proteomic data, allowing researchers to link virulence factors to specific genes and proteins. This is particularly useful for studying biofilm formation, as many of the genes involved in biofilm production are also key virulence factors. VFDB also provides comparative genomics tools that enable researchers to identify conserved virulence factors across different bacterial species.

VFDB is widely used in studies that aim to understand the role of biofilms in chronic infections, such as cystic fibrosis and wound infections. By providing a detailed catalog of biofilm-associated virulence factors, VFDB has become an essential resource for researchers investigating the link between biofilms and bacterial pathogenesis.

3.7 Machine Learning Applications in Biofilm Research

Over the past decade, the application of deep learning models, particularly CNNs, to image classification tasks has gained significant attention across various domains, including medical imaging, environmental monitoring, and microbial analysis. Traditional methods of bacterial classification typically involved manual feature extraction from images, such as texture analysis, shape descriptors, and histogram-based techniques. While effective to some extent, these methods often fall short in capturing the full complexity and variability present in bacterial species.

In contrast, CNNs have proven to be highly effective in learning complex patterns and relationships in image data. Several studies have demonstrated the applicability of CNNs in medical and microbial imaging, showing significant improvements over traditional methods in terms of accuracy and generalization.

L. Huang and T. Wu [30] applied a CNN-based approach to classify bacterial images into different species and demonstrated that CNNs achieved significantly higher accuracy compared to traditional machine learning techniques, such as Support Vector Machines (SVM) and Random Forests. Their study also highlighted the ability of CNNs to generalize well to new data, making them suitable for tasks where unseen bacterial species may need to be classified.

Another study by Ö. F. Nasip and K. Zengin [31] explored the use of transfer learning in bacterial classification. By fine-tuning pre-trained CNN models such as AlexNet and VGGNet, they achieved high classification accuracy with relatively small amounts of training data. This study emphasized the importance of transfer learning for bacterial image classification, especially when the dataset is limited in size.

Recent advances in the field have also focused on improving model generalization and reducing overfitting. Regularization techniques such as dropout, batch normalization, and data augmentation have been applied to improve CNN performance on small datasets. Additionally, methods such as synthetic data generation and the use of generative models, such as GANs (Generative Adversarial Networks), have been explored to augment datasets and improve model generalization.

3.8 Contributions of This Work

While the databases discussed above provide significant contributions to biofilm research, none of them fully address the integration of microscopy data with biofilm metadata in an accessible and standardized way. The work presented in this thesis aims to fill this gap by combining image processing, data standardization, and biofilm metadata into a single cohesive platform. This approach provides several unique advantages:

- **Integration of Microscopy Images with Metadata:** Unlike existing platforms, this system allows the submission and storage of biofilm images (such as Z-stacks and multi-channel microscopy data) along with detailed metadata, including the microorganism species, imaging parameters, and experimental conditions. This tight integration allows researchers to not only access raw biofilm data but also visually examine the biofilm structures.
- **Real-time Image Processing and Visualization:** The platform developed in this thesis incorporates real-time image processing that transforms complex microscopy data into usable formats (e.g., PNG) for immediate visualization. This feature is particularly valuable for researchers who require quick feedback on biofilm morphology and structure without needing external image processing tools.

- **Enhanced Data Submission Workflow:** The developed submission pipeline supports standardized submission of biofilm data, combining metadata, imaging data, and bibliographic references in a single step. This streamlined workflow minimizes the chance of errors and ensures that all relevant data is captured for each biofilm sample.
- **Focused on Data Integrity and Accessibility:** The focus of this platform is not just on data collection but also on ensuring high data integrity through validation mechanisms, error reporting, and real-time feedback to users. The platform is designed to maintain accessibility and usability, addressing the technical limitations seen in other biofilm databases.
- **Integrated Bacterial Species Prediction and Concentration Estimation:** This platform incorporates a machine learning-based model that can predict the bacterial species present in a biofilm based on its morphology and related metadata. The prediction model uses convolutional neural networks (CNNs) trained on microscopy images, offering immediate insights into biofilm composition without requiring additional testing. Additionally, the platform predicts the concentration of bacteria within the biofilm, providing an estimate of bacterial density alongside species identification, further enhancing the biological insight researchers can gain from their data.

The combination of these features provides a comprehensive platform that not only enhances the workflow for biofilm data submission and analysis but also integrates machine learning to predict bacterial species. This positions the platform as a valuable tool for both experimental analysis and predictive modeling of biofilms.

3.9 Key Takeaways

In short, the biofilm databases and computational tools discussed in this chapter are critical resources for advancing our understanding of biofilm formation, prediction, and control. The rapid development of machine learning methods, combined with the wealth of data available in these databases, has opened up new avenues for biofilm research. However, challenges such as data standardization and platform stability need to be addressed to fully realize the potential of these resources. Future research will likely focus on integrating these databases with more advanced computational models to better understand the complexities of biofilm biology.

4

Development of the Biofilm Database and Website

This chapter presents the design and development of the biofilm database and the associated web-based platform for interacting with biofilm data. The database serves as a structured repository for biofilm metadata and microscopy images, while the web application and API enable users to submit, retrieve, and analyze biofilm samples efficiently. In this chapter, we explore the architecture of the database, the image processing techniques for handling microscopy data, and the overall workflow of interaction between the API and the front-end.

The following sections provide a detailed overview of the core components: Section 4.1 focuses on the database design and relational model used for organizing biofilm data. Section 4.2 describes the development of the RESTful API that facilitates interactions with the database. Section 4.3 explains how user control was implemented in this application. Section 4.4 discusses the image processing pipeline, specifically focusing on the LIF (Leica Image File) [32] format. Finally, Section 4.5 details the submission process for biofilm samples and the various validation checks implemented to ensure data integrity.

4.1 Database Design

The foundation of the biofilm research platform is a MySQL [33] relational database, which is used to store structured data on biofilms and associated microscopy images. Relational databases provide the necessary consistency, flexibility, and query performance for managing complex, interconnected data, making MySQL a fitting choice for this application.

4.1.1 Relational Schema

The database is structured using a relational schema where biofilm-related data is organized across multiple tables, with relationships defined between them. The main entities in this schema are as follows:

- `LifFile`: This table stores metadata for each LIF file related to biofilm samples. It includes fields such as the filename, number of images, reference to the related scientific article, culture media used, staining method, biofilm-forming device, and a foreign key linking it to the `User` who uploaded the file.

- `Biofilm`: This table records details about each biofilm sample, including the number of channels, spatial dimensions (X, Y, Z), and a foreign key reference to the corresponding `LifFile` entry.

- `Microorganisms`: This table lists the microorganisms identified in the biofilm samples, with each entry corresponding to a unique microorganism name.

- `Images`: This table holds information about each image associated with the biofilm sample, including the file path, image filename, and a foreign key reference to the corresponding `Biofilm` entry.

- `LifFile_Microorganisms`: This is a join table that represents the many-to-many relationship between `LifFile` and `Microorganisms`, linking biofilm samples to the microorganisms present.

- `User`: This table stores information about users, such as username, email, and researcher status, and each `LifFile` is linked to a user via a foreign key.

The relationships between these tables form the backbone of the database. Each `LifFile` can be linked to multiple biofilm entries in the `Biofilm` table. Furthermore, each biofilm sample can have multiple corresponding images, stored in the `Images` table. The many-to-many relationship between `LifFile` and `Microorganisms` is managed by the `LifFile_Microorganisms` join table.

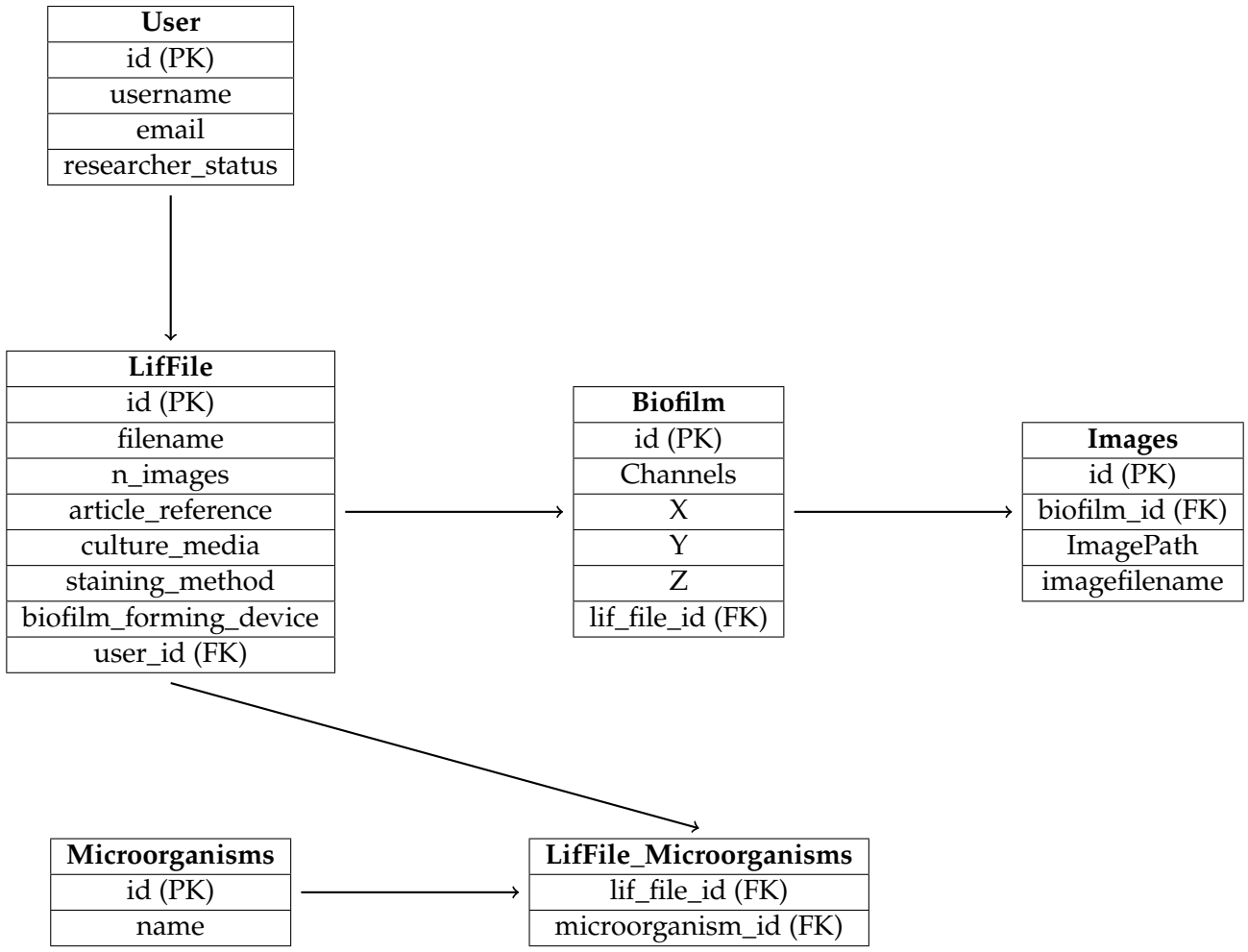

Figure 4.1: Schema representation of the database.

4.1.2 Normalization and Optimization

The database schema is normalized to the third normal form (3NF) [34] to ensure minimal redundancy and maintain data consistency. In doing so, each piece of information is stored only once, and relationships are handled through foreign keys. This structure allows efficient updates, deletions, and insertions without the risk of introducing anomalies. For example, changes to biofilm metadata are reflected without affecting the related image data, which remains linked via the foreign key.

MySQL indexing is employed on key columns like `biofilm_id` and `microorganism_id`, which improves query performance, particularly when dealing with large datasets. The primary keys on the `LifFile` and `Biofilm` tables ensure that each record is uniquely identifiable, while the foreign key constraints in the `Images` and `LifFile_Microorganisms` tables maintain referential integrity.

4.1.3 Data Integrity and Constraints

Maintaining data integrity is essential for ensuring the accuracy of biofilm records. The following constraints are implemented within the database:

- **Primary Keys**: Each entry in the `LifFile` and `Biofilm` tables is uniquely identified by its primary key (`id`).
- **Foreign Keys**: The `Images` table contains a foreign key that references the `id` field in the `Biofilm` table, ensuring the association of each image with a specific biofilm sample. Similarly, the `LifFile_Microorganisms` table includes foreign keys that link `LifFile` entries to corresponding microorganisms.
- **Field Constraints**: To maintain data validity, specific constraints are applied to fields. For example, spatial dimensions (X, Y, Z) in the `Biofilm` table are constrained to positive integer values to prevent invalid data entries.

These constraints, combined with logical data checks during submission, help ensure that the biofilm database maintains high data quality and integrity.

4.2 API Design and Development

To allow external applications, including the front-end and analytical tools, to interact with the biofilm database, a RESTful API was developed using Flask [35], a lightweight and modular web framework written in Python. The API exposes various endpoints that enable the submission, retrieval, and querying of biofilm data and images, making the platform accessible to researchers in a user-friendly and programmatic way.

4.2.1 RESTful Architecture

The API follows the REST (Representational State Transfer) architectural style, which is widely adopted for web services due to its simplicity and scalability. RESTful APIs interact with resources through standardized HTTP methods:

- **GET**: Retrieves information from the server. For instance, a `GET` request to the `/database` endpoint retrieves all biofilm records, while a request to `/biofilm/{id}` retrieves detailed information about a specific biofilm sample, including links to associated images.
- **POST**: Submits new data to the server. The `/submit` endpoint allows users to upload new biofilm data and images, which are then validated and stored in the database.

This RESTful design ensures that the API is stateless, meaning each request is processed independently without relying on any session information stored on the server. This improves scalability and allows for easier load balancing across multiple servers.

4.2.2 Handling Cross-Origin Requests

Given that the front-end application is hosted on a different domain than the API, Cross-Origin Resource Sharing (CORS) was enabled to allow the web client to communicate with the API securely. CORS is a security feature implemented by web browsers to prevent unauthorized domains from making requests to the server. By configuring CORS to permit requests from trusted origins, such as the front-end domain, the API allows legitimate interactions while blocking potential cross-site attacks.

4.2.3 Security Considerations

Several security measures were implemented to protect the integrity of the biofilm data and prevent unauthorized access:

- **Input Validation**: All user inputs, particularly file uploads and metadata fields, are validated at the API level. For instance, the file type is checked to ensure that only LIF files are accepted, and metadata fields are verified to contain valid, expected values.
- **Parameterized Queries**: To protect against SQL injection attacks, all database interactions are executed using parameterized queries. This ensures that user inputs are safely escaped before being passed into SQL commands, preventing malicious code from being executed.
- **HTTPS**: Although not detailed in the API implementation, securing communication via HTTPS is a best practice for encrypting data transmitted between the client and the server, ensuring that sensitive information such as login credentials and biofilm data are not exposed to interception.

By implementing these security measures, the API remains robust and secure, allowing only authorized and valid data submissions and queries.

4.3 User Control and Role Management

A critical aspect of the biofilm database and website is user control, ensuring that only authorized personnel have access to sensitive data and administrative actions. The platform incorporates user authentication and role-based access control (RBAC), managed through a combination of Flask-Login for user authentication and session management, and Flask-JWT-Extended for secure token-based authorization. The user control system is designed to provide a seamless and secure experience, allowing users to register, log in, and access the platform based on their assigned roles, such as regular users or administrators.

This section details the user authentication process, role-based access management, and the different levels of permissions implemented to protect and manage the data.

4.3.1 User Authentication

User authentication is handled using `Flask-Login`, which provides session management and user state tracking. Upon successful login, users are authenticated and a session is created. The session allows users to maintain their authentication status while interacting with various features of the platform without needing to re-enter credentials.

In parallel, `Flask-JWT-Extended` is employed to issue JSON Web Tokens (JWT) to the users. JWT tokens are created during login and used in API requests to authenticate and authorize users without requiring session persistence on the server. The authentication workflow is as follows:

1. A user submits their login credentials (username and password).

2. The system verifies the credentials against the stored password hash in the `User` database.

3. Upon successful verification, an access token (JWT) is generated, embedding the user's unique identity.

4. This JWT is returned to the client and must be included in subsequent requests to protected API endpoints.

The JWT tokens have a set expiration time, and when they expire, users must log in again to obtain a new token. This process ensures secure and stateless authentication across multiple devices or front-end clients.

4.3.2 Role-Based Access Control (RBAC)

The system implements Role-Based Access Control (RBAC) to assign user permissions based on their roles. There are now three primary roles:

- **User**: Regular users can register, view, and predict biofilm data. Their access is limited to publicly available datasets, and they are restricted from administrative actions.

- **Researcher**: Once a user applies for and is granted researcher status, they gain additional privileges, such as access to private or restricted datasets and tools and to submit data. This status must be requested and approved by an administrator.

- **Administrator**: Administrators have complete control over the system, including managing biofilm data, editing or deleting entries, managing users, approving researcher requests, and performing administrative tasks like database maintenance.

4.3.2.1 User Role Management in the Database

User roles are stored in the `User` table, with each user having a role field indicating their access level. For example:

User
id (PK)
username
email
password
role
researcher_status

Figure 4.2: Schema representation of the `User` table in the database.

The `role` field can hold the values `'user'`, `'researcher'`, or `'admin'`. Newly registered users are automatically assigned the `'user'` role by default. Users can apply for researcher status, which will set their `researcher_status` to `'pending'` until approved by an administrator. Once approved, their status will be updated to `'approved'`.

4.3.3 Access Control with Flask-Login and JWT

`Flask-Login` provides decorators such as `@login_required` to ensure that certain views are protected from unauthenticated users. Additionally, the platform uses the `@jwt_required` decorator from `Flask-JWT-Extended` to protect API routes that require a valid JWT token.

For role-based restrictions, the system checks the user's role during API requests. For example, when an API endpoint requires administrative privileges, the request is validated to ensure that the user's JWT contains the role of `admin`. If the user's role does not match the required access level, an error response is returned.

4.3.4 User Registration and Login Workflow

Registration: Users can register by providing a unique username, email, and password. The password is securely hashed using `werkzeug`'s `generate_password_hash` function before being stored in the database.

Once the user's data is validated and no conflicts (e.g., duplicate usernames or emails) are found, the user is added to the database with the default `'user'` role.

Login: Users log in by submitting their credentials. The system verifies the credentials, and if correct, generates a JWT token for further interactions with the platform.

Logout: Upon logout, the session is cleared using `Flask-Login`'s `logout_user` function, and the user is required to log in again for further access.

4.3.5 Security Considerations

To ensure the integrity and security of the user management system, several precautions are in place:

- **Password Hashing:** User passwords are never stored in plaintext. `Flask-Login` and `Werkzeug` provide tools to securely hash passwords.
- **Token Expiration:** JWT tokens have an expiration time to limit the exposure of compromised tokens. After expiration, users must re-authenticate to obtain a new token.
- **Input Validation:** User inputs, such as usernames, passwords, and email addresses, are validated to prevent SQL injection, XSS (Cross-Site Scripting), and other common attacks.

By combining authentication, role-based access control, and robust security practices, the platform ensures that only authorized users can interact with sensitive biofilm data and administrative features.

4.4 Image Processing Pipeline: LIF Image Extraction

The extraction and processing of microscopy images from the LIF (Leica Image File) format is a crucial step in handling the multi-dimensional data generated by confocal microscopy. LIF files store extensive image data, including 3D Z-stacks and multiple channels, and must be processed systematically to capture the biofilm's structural and biological details. This section details the steps involved in extracting, processing, and storing this complex data.

4.4.1 LIF File Structure and Metadata

LIF files, proprietary to Leica confocal microscopy systems, serve as a container for large microscopy datasets. Each file typically contains:

- **Z-stacks**: A series of 2D images captured at different focal depths, allowing the reconstruction of a 3D representation of the biofilm. Each Z-stack represents an image taken at a specific depth in the sample.
- **Channels**: Each Z-stack often includes multiple channels, where each channel corresponds to a specific fluorescence wavelength. These channels are used to visualize different components or species within the biofilm, such as different microorganisms or fluorescent markers.
- **Metadata**: LIF files also store essential metadata about the images, such as spatial dimensions (X, Y, Z), the number of channels, the pixel resolution, and the acquisition settings used during imaging (e.g., magnification and exposure).

This structure enables high-dimensional imaging but requires a specialized process for extracting and handling the data in a usable format.

4.4.2 Image Extraction Process

The process of extracting images from a LIF [32] file begins as soon as the user uploads the file via the web interface. The system processes the file using a specialized Python library designed to handle LIF files. Below is a step-by-step explanation of the image extraction workflow.

4.4.2.1 Step 1: File Initialization

Once the file is uploaded, it is loaded into memory, allowing the system to efficiently read and process its contents. At this stage, the file's structure, including its internal datasets, is analyzed, preparing the system for further operations. This initialization is crucial for handling potentially large LIF files without the need for intermediate file storage on the server.

4.4.2.2 Step 2: Iterating Over the Images

Each LIF file contains one or more images, organized into sets of Z-stacks and channels. The system iterates through the images, treating each Z-stack and channel combination as an individual dataset. For each image, key metadata is extracted, including the number of Z-planes and channels, as well as the dimensions of each plane. This metadata provides the necessary context for how the images are captured and ensures that the system processes them correctly.

4.4.2.3 Step 3: Extracting Biofilm Metadata

For each image extracted from the LIF file, the corresponding biofilm metadata is also retrieved. This includes information such as the spatial resolution (X, Y, Z dimensions), the number of channels, and the total number of images in the Z-stack. This metadata is then stored in the database, where it is linked to the specific biofilm entry. The extracted metadata is crucial for downstream analysis, as it defines the biofilm sample's physical characteristics.

4.4.2.4 Step 4: Processing Z-stacks and Channels

Within each image, multiple Z-stacks represent the biofilm's structure at different focal depths. The system processes each Z-stack individually, extracting the 2D image corresponding to that particular depth. Similarly, each channel is processed to extract the fluorescence signal for a specific wavelength. This step ensures that each combination of Z-stack and channel is captured, preserving the full depth and spectral information of the biofilm.

4.4.2.5 Step 5: Image Transformation and RGB Conversion

Once the individual Z-stacks and channels are extracted, they are processed to create a composite RGB image. Each channel is associated with a specific color (e.g., red, green, or blue), and the intensity values from the microscopy images are mapped to these color channels. The extracted images are normalized and thresholded to enhance contrast, and then combined into a single RGB image. This process allows multiple fluorescence signals to be visualized simultaneously in a single image.

4.4.2.6 Step 6: Image File Naming and Storage

To ensure that each extracted image is stored correctly, the system generates descriptive filenames based on the bacteria or structures present in the biofilm, the Z-plane, and the channel. These filenames are constructed by concatenating the names of the bacteria detected in the image, along with the image and Z-stack numbers. The images are then saved to the server's filesystem in a standard format such as PNG, which allows for easy access and display on the web interface.

4.4.2.7 Step 7: Insertion of Image Metadata into the Database

For each processed image, metadata such as the file path, image dimensions, and the bacteria present in the image is inserted into the database. This metadata is linked to the biofilm entry, ensuring that researchers can query and retrieve both the images and the associated biofilm information. The relational structure of the database facilitates efficient storage and retrieval, even for large datasets with multiple images per biofilm.

4.4.3 Data Integrity and Error Handling

Throughout the image extraction process, several validation checks are performed to ensure that the data is correctly extracted and stored. If any part of the file is malformed or if the images cannot be processed, the system generates an error message and halts further processing. This prevents incomplete or incorrect data from being stored in the database. Additionally, constraints on the spatial dimensions and channel numbers ensure that the images conform to expected formats.

By implementing these steps, the system efficiently handles complex LIF datasets, extracting and processing multi-dimensional microscopy images while preserving their integrity for future analysis and visualization.

4.4.4 Image Storage and Access

Rather than storing images directly in the database, which would increase its size and complexity, the system stores images as files on the server's filesystem. The file paths are stored in the database, enabling efficient retrieval when necessary. This approach reduces the load on the database and simplifies image management.

Whenever an image is requested via the API (for example, when a researcher queries a specific biofilm sample), the server dynamically generates a URL pointing to the image location. The front-end can then access and display the images using these URLs, ensuring fast and efficient image delivery without burdening the database with large binary objects.

4.5 Data Submission Process

The system allows researchers to submit new biofilm data through a web interface, which is linked to the API's `/submit` endpoint. The submission process involves uploading both the biofilm metadata and the microscopy images (in LIF format), which are then validated, processed, and stored in the database and filesystem.

4.5.1 Submission Workflow

The submission process involves the following steps:

1. The researcher uploads the biofilm metadata, including fields such as the microorganisms involved, the imaging dimensions (X, Y, Z), and the number of channels used.
2. The LIF file containing the microscopy images is uploaded alongside the metadata. The system checks the file extension to ensure that only LIF files are accepted.
3. Upon receiving the data, the API performs a series of validation checks. These include verifying that the metadata fields are complete and that the LIF file can be processed correctly.
4. If the data passes validation, the biofilm metadata is inserted into the `Biofilm` table, and the individual images extracted from the LIF file are stored in the filesystem. The file paths are linked to the biofilm record via the `Images` table.

4.5.2 Validation and Error Handling

Ensuring that only valid and well-formed data is stored in the system is critical for maintaining the integrity of the biofilm dataset. Several validation mechanisms are in place:

- **File Type Validation**: Only LIF files are accepted. Any attempt to upload a file in an unsupported format is rejected, and the user is informed of the issue.
- **Metadata Validation**: Each metadata field is checked to ensure that it contains valid values. For instance, the spatial dimensions must be positive integers, and the number of channels must correspond to the data in the LIF file.
- **Error Reporting**: If any validation check fails, the API returns an error message detailing the specific issue, allowing the user to correct the submission.

By implementing these validation checks, the system ensures that the submitted data is both accurate and complete, reducing the likelihood of errors or inconsistencies in the dataset.

5

Development of the Front-End Interface

The front-end interface of the biofilm platform is a web-based application designed to provide researchers with intuitive access to the biofilm database. The front-end enables users to interact with the system, including viewing biofilm data, submitting new biofilm entries, and navigating through images of biofilm samples. This chapter details the technologies used in the front-end development, the architecture of the application, and the user interface features, with considerations for screenshots that will be incorporated to demonstrate the user experience.

The front-end was developed using React, a popular JavaScript library for building user interfaces [36], and Material-UI (MUI) for responsive and accessible components [37]. Additionally, Axios was used to handle HTTP requests, facilitating seamless communication between the front-end and the back-end API described in Chapter 4 [38].

5.1 Technologies and Frameworks

The front-end of the biofilm platform is built with a combination of modern web technologies that ensure efficiency, responsiveness, and ease of use. The following key technologies were used:

5.1.1 React

React is a JavaScript library developed by Facebook, widely adopted for building user interfaces due to its component-based architecture [36]. Each part of the biofilm application is built as a React component, which allows for reusable, independent sections of code that handle their own state and logic. This modular approach makes it easy to maintain and scale the application as new features are added. React also provides an efficient way to manage the user interface through its virtual DOM (Document Object Model), ensuring that only the components that need updating are re-rendered, improving performance.

5.1.2 Material-UI (MUI)

Material-UI is a library of pre-built, customizable React components based on Google's Material Design guidelines [39]. MUI was chosen for the biofilm platform to create a consistent and visually appealing user interface with minimal custom styling. The platform's components, such as tables, buttons, and input fields, are all based on Material-UI elements, ensuring that the interface is both functional and aesthetically pleasing.

5.1.3 Axios

Axios is a promise-based HTTP client used to manage communication between the front-end and the API [38]. It handles all requests for data retrieval and submission, such as fetching the list of biofilm entries from the database or submitting a new biofilm entry. Axios was selected due to its simplicity in handling asynchronous operations and its ability to intercept and manage HTTP responses.

5.2 Application Structure and Component Architecture

The biofilm platform is structured as a single-page application (SPA), where navigation between pages occurs dynamically within the browser, without requiring full-page reloads. This structure enhances the user experience by providing faster interactions and a more fluid interface [40]. The database is available at www.dbiofilms.com.

5.2.1 Home Component

The `Home` component serves as the landing page for users accessing the biofilm platform. It introduces the platform, briefly explains its purpose, and dynamically displays the number of biofilm entries currently stored in the database. This component interacts with the API by sending a request to retrieve the total count of biofilm entries upon loading the page.

The user interface for the `Home` component is simple yet informative, consisting of welcome text and a few brief instructions on how to navigate to the database of biofilms. The number of entries is dynamically updated by React's `useState` and `useEffect` hooks, which allow the component to fetch and display the current count from the API when the page is first rendered [41].

A screenshot of the `Home` component (Figure 5.1) demonstrates the welcoming interface, with the dynamically updated count of biofilm entries prominently displayed in the center.

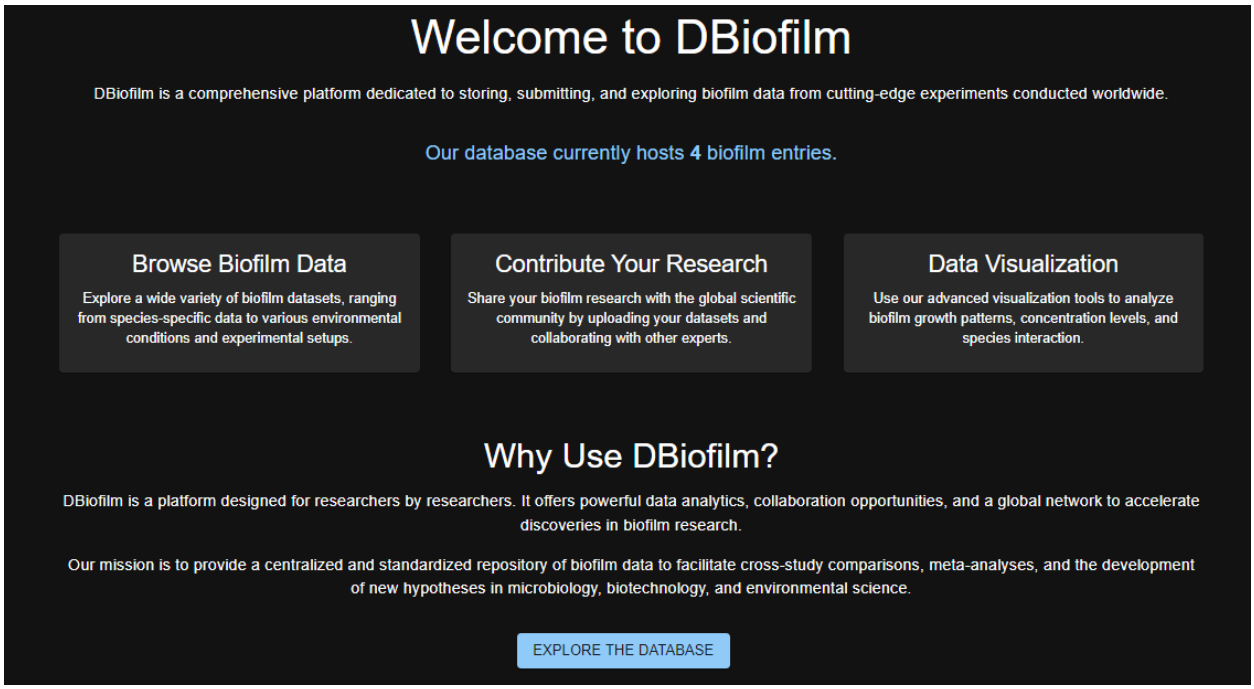

Figure 5.1: Screenshot of the `Home` Component displaying the biofilm entry count.

5.2.2 Database Component

The `Database` component serves as the central hub of the platform, allowing users to view a table displaying biofilm entries from the database. Each entry includes essential details such as the filename, the number of 3D images associated with the sample, the microorganisms involved, and any references to related publications.

Material-UI's `Table` component is utilized to create a responsive and structured table, organizing biofilm data efficiently for users [37]. The data is fetched from the API using the `getData` function when the component is first mounted, as implemented in React's `useEffect` hook [41]. This ensures that the data is dynamically loaded and presented to the user as soon as the page is accessed.

In addition to displaying biofilm data, the `Database` component includes several interactive features:

- A search bar powered by a `TextField` component allows users to filter entries by file name, providing quick access to relevant biofilms.
- Clicking on a biofilm entry redirects the user to a detailed view of the biofilm via a unique ID, linking to the `Biofilm` component.
- If a user has a `researcher` or `admin` role, they can submit new biofilms through a "Submit Biofilm" button. For logged-in users who are not researchers, there is an option to request `researcher` privileges using the "Request Researcher Role" button.

- Users with `admin` privileges are presented with an additional "Admin Requests" button, providing access to administrative tools.

If the data is still being loaded from the API, a `CircularProgress` spinner is displayed to indicate the loading state. Once the data is loaded, it is filtered based on the user's search input and displayed in the table.

Figures 5.2, 5.3 and 5.4 provides screenshots of the `Database` component, illustrating the layout of the table, search functionality, and the action buttons for users with different roles.

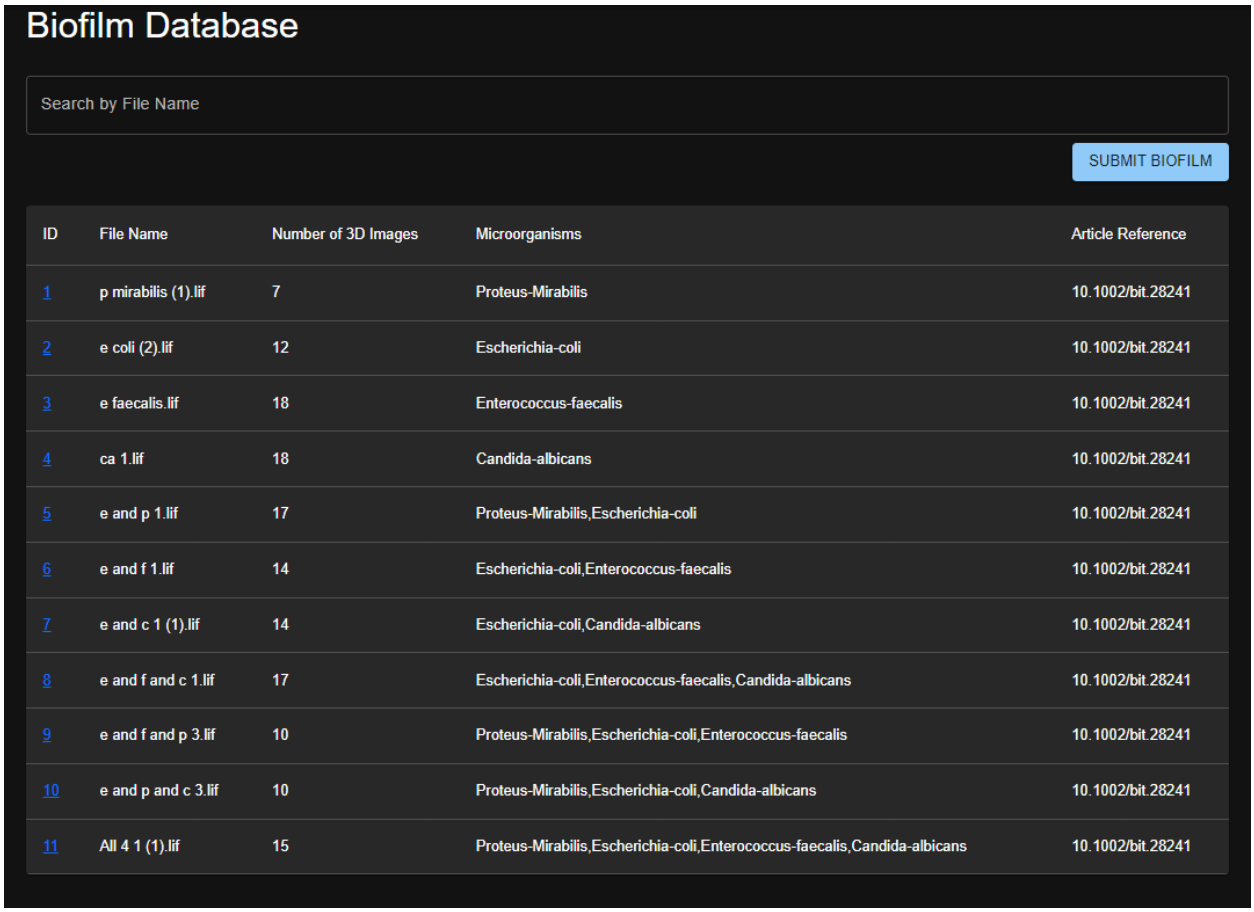

Figure 5.2: Screenshot of the `Database` Component with biofilm entries displayed in a structured table, as a researcher role.

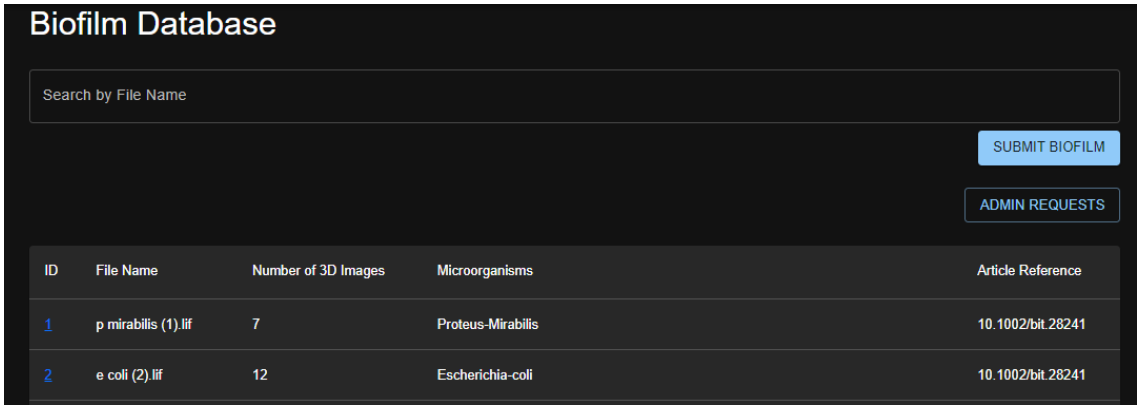

Figure 5.3: Screenshot of the `Database` Component for the administrator.

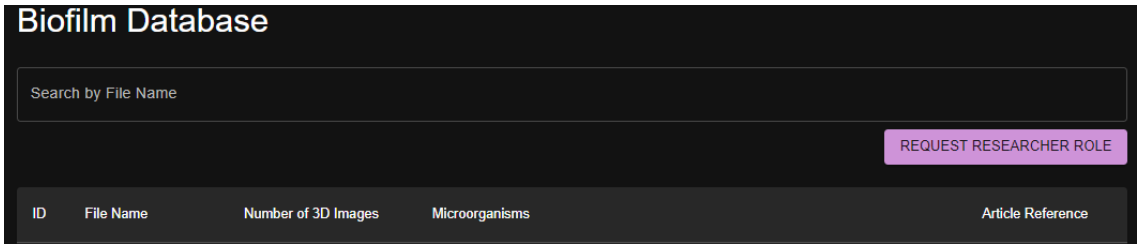

Figure 5.4: Screenshot of the `Database` Component for the user, where he can request to be a researcher.

5.2.3 Biofilm Detail Component

The `Biofilm` component provides a detailed view of individual biofilm samples. When a user selects a biofilm entry from the `Database` component, they are redirected to this page, which displays comprehensive metadata about the chosen biofilm. This metadata includes the biofilm file name, article reference, imaging dimensions (X, Y, Z), the staining method, the biofilm-forming device (if available), and any microorganisms present in the sample.

The component also features an interactive image viewer, where users can navigate through the Z-stack images of the biofilm. These images are retrieved from an API, and users can cycle through them using navigation arrows. The currently displayed Z-slice is managed by React's state, with the `currentZIndex` determining which Z-stack image is visible. Users can change the biofilm or Z-index using dropdown menus, which update the displayed images dynamically.

If a biofilm has multiple 3D images, the user can switch between them using a dropdown list that shows available images with their respective dimensions. When a different biofilm is selected, the Z-stack is reset to display the first Z-slice of the newly selected biofilm. If no images are available, an informative message is displayed to the user.

The component also handles data fetching, displaying a loading spinner while the biofilm data is being retrieved. In the case of an error during the data request, the user is notified with an appropriate error message. A screenshot of the `Biofilm` component (Figure 5.5) illustrates the layout, which includes the metadata section and the interactive image viewer.

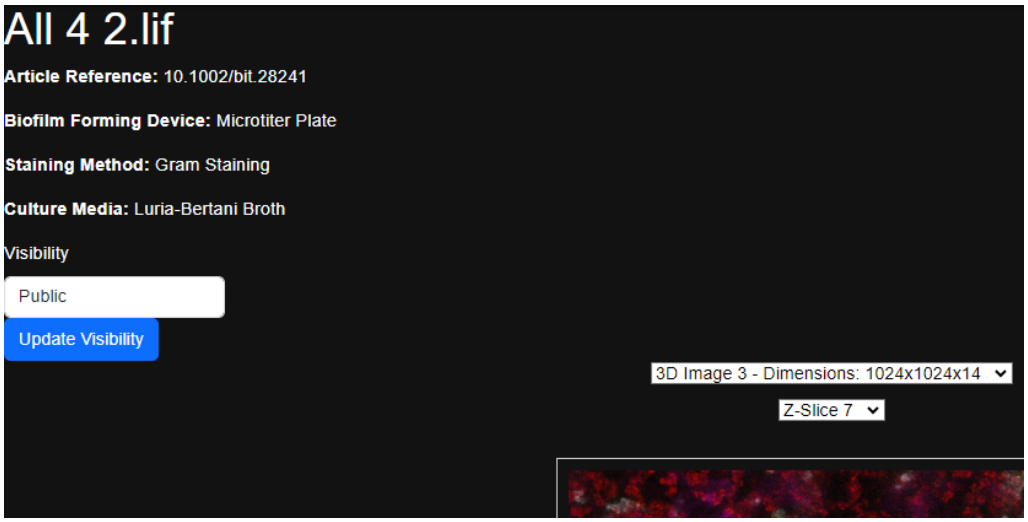

Figure 5.5: Screenshot of the `Biofilm` Component displaying detailed biofilm information and an interactive image viewer.

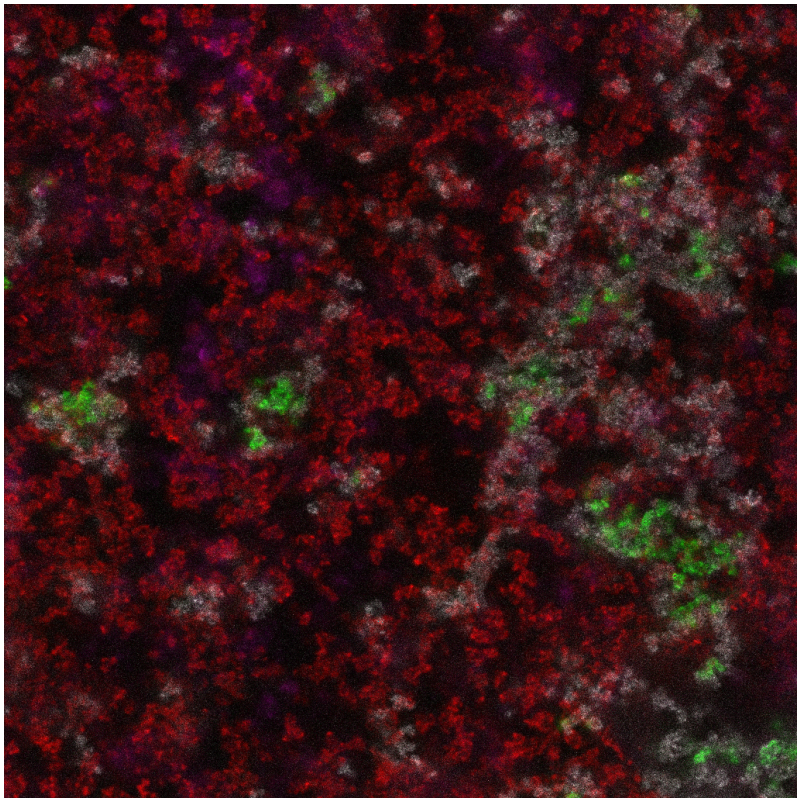

Figure 5.6: Example of a `Biofilm` image being displayed.

5.2.4 Admin Researcher Requests Component

The `AdminResearcherRequests` component provides administrative functionality to manage pending requests from users seeking researcher status. Admins are able to approve or decline each request directly from this interface.

Upon mounting, the component fetches the list of pending researcher requests from the API using the `getPendingResearchers` function. This API call is initiated in the `useEffect` hook and populates the list of pending requests, which is then displayed to the admin.

Key features of the `AdminResearcherRequests` component include:

- **Pending Request List**: If there are pending researcher requests, the usernames and email addresses of the users are displayed in a structured list. Each request is contained within a box, and the admin can approve or decline the request using buttons.

- **Approve and Decline Actions**: Admins are provided with "Approve" and "Decline" buttons for each pending user. Upon approving or declining a user, the `approveResearcher` and `declineResearcher` functions are called respectively, sending the user ID to the API. A success or error message is displayed based on the outcome, and the user's entry is removed from the list once processed.

- **Loading State**: While the list of pending requests is being fetched from the API, a loading spinner (`CircularProgress`) is displayed to inform the admin that data retrieval is in progress.

- **Error Handling**: If the API call fails, an error message (`Alert` component) is displayed, informing the admin that the pending researcher requests could not be fetched.

If there are no pending requests, a message stating "No pending researcher requests at this time" is shown, indicating that all current requests have been processed. Figure 5.7 illustrates the layout of the component, including the display of a pending request with "Approve" and "Decline" options.

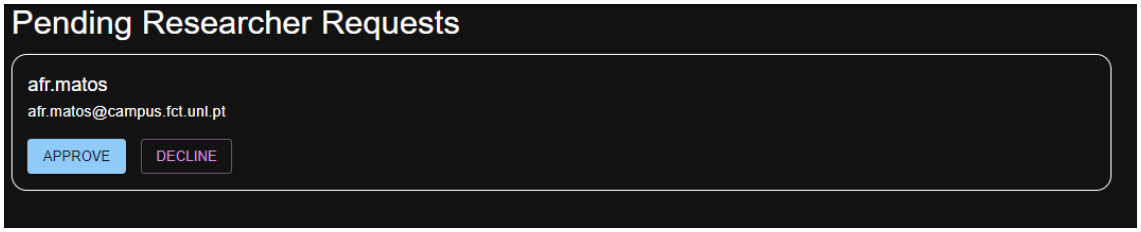

Figure 5.7: Screenshot of the `AdminResearcherRequests` Component showing pending researcher requests with action buttons for approval or decline.

5.2.5 Submit New Biofilm Data Component

The `Submit` component allows researchers to upload new biofilm data to the platform. This component includes a file upload input, where users can select their LIF microscopy files, and text fields for entering metadata such as the microorganisms involved and the staining method used.

Once the user fills out the form and uploads their LIF file, the `Submit` component sends the data to the API via an HTTP POST request using Axios [38]. The form submission is handled by packaging the data into a `FormData` object, which is then transmitted to the server for processing.

The interface provides feedback to the user by displaying alerts when the submission is successful or if any errors occur during the process. This component ensures that the submission of biofilm data is straightforward and user-friendly.

A screenshot of the `Submit` component (Figure 5.8) illustrates the layout of the form and the file upload functionality.

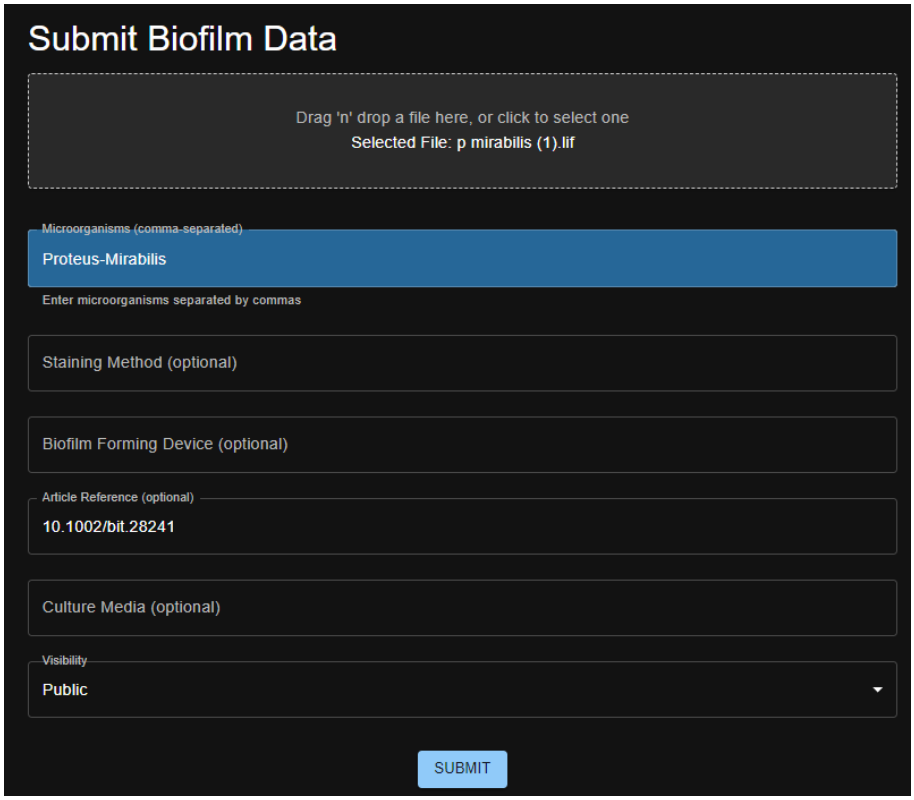

Figure 5.8: Screenshot of the `Submit` Component, showing the form for uploading biofilm data and LIF files.

5.2.6 Navigation and Layout

To provide users with an intuitive and consistent navigation experience, the front-end application includes a navigation bar built using Bootstrap. The `NavbarBootstrap` component creates a responsive and accessible navigation bar at the top of the page, allowing

users to easily switch between the `Database`, `Predict` and `Login/Logout/Register`.

Bootstrap was chosen for its lightweight yet powerful grid system and pre-built components, enabling a clean and functional layout without the need for extensive custom CSS [42]. The navigation bar is designed to work seamlessly across different screen sizes, ensuring that the interface remains usable on both desktop and mobile devices.

A screenshot of the navigation bar (Figure 5.9) will shows it integrates with the rest of the front-end application, providing easy access to the core features.

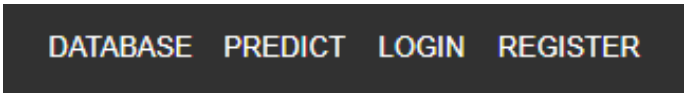

Figure 5.9: Screenshot of the navigation bar, showing links to the `Database`, `Predict`, and `Login/Register` pages.

5.3 State Management and Interactivity

One of the critical aspects of the front-end is how it manages and updates state dynamically to ensure interactivity and responsiveness. React's `useState` and `useEffect` hooks are extensively used to manage component state and handle side effects such as data fetching and user interactions [41].

5.3.1 useState and useEffect Hooks

The `useState` hook is used to manage the local state within each component. For instance, in the `Biofilm` component, the currently displayed image is tracked using the `currentIndex` state variable. When the user interacts with the image viewer by clicking the navigation arrows, this state is updated, triggering a re-render to display the next or previous image.

The `useEffect` hook is employed to handle side effects such as data fetching from the API. For example, in the `Database` component, `useEffect` is used to fetch biofilm entries from the API when the component is first loaded. The empty dependency array ensures that the API request is only made once, avoiding unnecessary re-fetching [41].

5.3.2 Form Handling and Submission

Form handling is an essential feature of the `Submit` component, where user inputs such as file uploads and text fields are managed via controlled components. When the form is submitted, the data is sent to the API using Axios, and appropriate feedback is displayed to the user depending on the response from the server [38].

React's controlled components and state management make the form submission process intuitive and responsive, ensuring that users can submit their biofilm data with ease.

5.3.3 Number of Microbial Cells Prediction Page

An important feature of the biofilm platform is the "`Predict`" page, which allows users to upload LIF microscopy files and predict the bacteria species present within biofilm samples. This page integrates the functionality provided by the back-end API, described in Chapter 4, and offers users a straightforward interface for interacting with the prediction model.

The `Predict` page includes a file upload input where users can select their '.lif' file. Once the file is uploaded, the system processes the image data to predict the number of microbial cells species and calculate their concentration based on the volume of the pictures, as describes by the XyZ coordinates. This procedure is applicable to all wavelengths(channels) tested. The platform supports both single-channel and multi-channel LIF files, adjusting the prediction and concentration analysis accordingly.

5.3.3.1 Handling Multi-Channel and Single-Channel Biofilms for Concentration Prediction

Biofilm imaging can capture varying levels of complexity depending on the number of fluorescence channels used during microscopy. These channels are often indicative of different bacterial species, fluorescent markers, or specific biological processes. Therefore, handling single-channel and multi-channel biofilm images for bacterial concentration prediction requires different approaches to account for the amount of biological information each channel carries. This section discusses the logic and methodology used in the system to process and predict bacterial concentrations for both single-channel and multi-channel biofilm samples, emphasizing the underlying reasons for using distinct methods.

5.3.3.2 Single-Channel Biofilms

In single-channel biofilms, all the biological information is captured through a single fluorescence channel, typically representing one bacterial species or a specific stain applied uniformly across the sample. Given the simplicity of this data, the primary task of the prediction pipeline is bacterial classification rather than estimating concentrations.

The system processes single-channel biofilms using a pre-trained convolutional neural network (CNN) model, optimized for bacterial classification tasks. Each slice of the Z-stack, representing a two-dimensional cross-section of the biofilm, is resized and normalized to match the input size and format expected by the model (e.g., 224x224 pixels with pixel values normalized between 0 and 1). This preprocessing ensures that the model can effectively interpret the biological data regardless of the original image dimensions or fluorescence intensities.

After preprocessing, each image slice is passed through the CNN model, which outputs a prediction of the bacterial species for that slice. These predictions are made independently for each Z-stack slice, which can contain slight variations due to noise or subtle structural

differences within the biofilm. Therefore, to ensure a robust classification across the entire biofilm sample, the system applies a majority voting mechanism. The majority voting technique works by aggregating the predictions from all slices and selecting the species that appears most frequently as the final classification.

Why Use Majority Voting in Single-Channel Biofilms? The choice to use majority voting for single-channel biofilms stems from the need to account for potential slice-by-slice variability. While individual Z-slices may contain noise or imperfect fluorescence signals, the overall structure of the biofilm is often consistent, meaning that the dominant bacterial species will appear repeatedly across the Z-stack. Majority voting helps mitigate the impact of outliers and inconsistencies that might occur in individual slices, leading to a more reliable overall prediction.

Additionally, single-channel biofilms usually represent a homogeneous biological environment, with one dominant bacterial species. This makes classification based on the most frequent prediction a logical and effective approach, as it reflects the assumption that the bacterial species remains consistent throughout the biofilm.

5.3.3.3 Multi-Channel Biofilms

In contrast, multi-channel biofilms capture a more complex biological environment, where each fluorescence channel corresponds to a different bacterial species or biological marker. In these cases, the system must not only identify the species present but also estimate their relative concentrations. Each channel provides distinct information, and the system processes them independently to ensure that the contribution of each bacterial species is accurately represented.

For multi-channel biofilms, the concentration prediction pipeline processes each channel separately. Each slice of the Z-stack for each channel is analyzed by applying a combination of normalization and thresholding techniques. The purpose of this step is to standardize the pixel intensities, which correspond to the fluorescence signal emitted by the bacteria or marker in question, and filter out background noise.

After normalization, the fluorescence signal in each slice is summed to estimate the bacterial concentration within that channel. This sum is accumulated across all slices of the Z-stack to provide a total fluorescence signal for each bacterial species. The system maps each channel to a specific bacterium based on predefined fluorescence channel mappings, ensuring that the bacterial species and fluorescence signals are correctly paired.

Why Use Concentration Calculations in Multi-Channel Biofilms? The use of concentration calculations in multi-channel biofilms is necessary due to the presence of multiple bacterial species within the same sample. Unlike single-channel biofilms, where the bacterial composition is uniform, multi-channel biofilms represent a heterogeneous environment where different species might coexist in varying proportions. Therefore, it is

essential not only to identify which species are present but also to determine their relative abundances.

By summing the fluorescence intensity for each channel, the system calculates the total bacterial concentration. These values are then normalized across all channels to provide a proportional representation of the bacteria in the biofilm. This approach ensures that the bacterial concentrations reflect the actual biological complexity of the sample, offering researchers more detailed insights into the biofilm's composition.

5.3.3.4 Error Handling and Validation

To ensure that predictions are accurate and meaningful, the system includes rigorous error handling and validation steps. Upon receiving an uploaded LIF file, the system first validates the file format and ensures that the fluorescence channels are properly mapped to the expected bacterial species. If the file does not meet the required format, or if the metadata is missing or incomplete, the system rejects the file and returns an error message to the user. This ensures that the prediction and concentration calculations are based on valid, reliable data.

Furthermore, the system checks for the presence of fluorescence signals in each channel, ensuring that only channels with detectable fluorescence are used in the concentration calculations. This prevents the system from generating misleading results in cases where a channel may not contain any signal.

Robustness and Scalability The pipeline is designed to handle a wide variety of biofilm samples, from simple single-channel data to complex multi-channel environments, providing scalable and accurate bacterial classification and concentration predictions. By validating input data and employing robust prediction methods, the system ensures that researchers receive reliable and actionable insights into the bacterial composition of their biofilm samples.

5.3.4 User Interface and Interaction

The user interface of the `Predict` page is designed for ease of use. The user is prompted to upload a '.lif' file, after which the prediction is initiated by pressing the `Submit` button. Once the prediction is complete, the results, including the identified bacteria species and their respective concentrations, are displayed in a well-structured format.

A screenshot of the `Predict` page (Figure 5.11) will illustrate the interface, showing the file upload field, the submission button, and the predicted results.

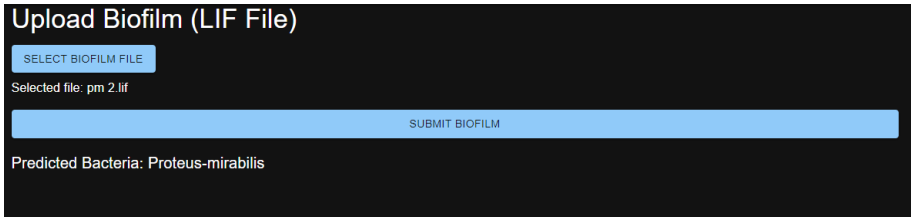

Figure 5.10: Screenshot of the `Predict` Page, displaying the file upload input and predicted bacteria result for single bacteria.

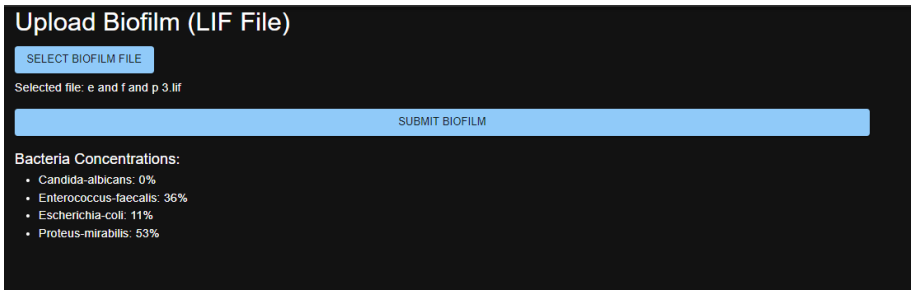

Figure 5.11: Screenshot of the `Predict` Page, displaying the file upload input and predicted bacteria results for multiple bacteria.

The page also handles errors gracefully, providing feedback if an invalid file format is uploaded or if the prediction process encounters issues. The platform ensures that users are always informed about the status of their prediction requests, either through success messages or error alerts.

By integrating the back-end's bacteria prediction capabilities into the front-end, the platform offers researchers an efficient way to analyze their biofilm samples, providing valuable insights into the bacterial composition of the biofilms under investigation.

6

Development and Optimization of Convolutional Neural Networks for Bacterial Image Classification

6.1 Overview

The classification of microbial species is a critical task across various disciplines such as healthcare, pharmaceuticals, environmental sciences, and biosecurity. Precise microbial identification aids in diagnosing infections, determining appropriate antimicrobial treatments, and performing environmental monitoring. Traditional microbial classification techniques often rely on methods like biochemical testing, microscopy, and culture-based approaches. These approaches, although widely used, tend to be time-consuming, labor-intensive, and prone to errors due to the complexity of microbial morphology and human interpretation of results. Moreover, traditional methods can sometimes lack the precision required for distinguishing between closely related microbial species.

As advancements in machine learning have evolved, image-based classification, particularly with the application of deep learning models, has shown great potential in improving microbial classification tasks. The development of computational methods capable of automatically analyzing microbial images has introduced the possibility of more rapid, accurate, and scalable microbial identification techniques, which are crucial in fields such as infectious disease diagnosis and antibiotic resistance monitoring.

In this chapter, we explore the development and optimization of Convolutional Neural Networks (CNNs) for classifying microbial images based on their morphological characteristics. CNNs have gained prominence in the field of computer vision due to their capacity to learn spatial hierarchies from image data, making them highly suitable for tasks that require the identification of fine-grained differences between classes. Unlike traditional machine learning algorithms that rely on manual feature extraction, CNNs automatically extract relevant features from raw images, enhancing classification accuracy and efficiency. This characteristic is particularly advantageous for microbial classification, where the subtle morphological differences between species may be challenging to capture

using handcrafted features.

The chapter presents a comparative analysis of several CNN architectures to determine the most effective approach for microbial image classification. The architectures examined include a simple CNN, a ResNet50-based model, and a custom hybrid model. In addition to architecture design, we also discuss the preprocessing techniques and data augmentation strategies employed to enhance model performance and reduce overfitting.

The remainder of this chapter is structured as follows: Section 6.2 provides a detailed overview of the CNN architectures employed in this study, including their components and relevance to microbial image classification. Section 6.3 describes the dataset, along with the preprocessing steps and data augmentation techniques used to enhance model performance. Section 6.4 presents the design and architecture of the three CNN models—a simple CNN, a ResNet50-based model, and a custom hybrid model—highlighting their structure and key characteristics. Section 6.5 focuses on the training methodologies, optimization strategies, and regularization techniques used to fine-tune the models and improve their performance. In Section 6.6, we evaluate and compare the models based on their accuracy and loss, presenting the results alongside the confusion matrices. Section 6.7 provides a comprehensive discussion of the model performances, identifying strengths, limitations, and areas for improvement. Finally, Section 6.8 summarizes the key findings and outlines potential directions for future research, including advanced architectures, data augmentation, and synthetic data generation techniques.

6.2 Dataset Description

The dataset used in this study consists of 4,936 images of organisms, classified into four distinct categories. These images were captured via microscopy, showcasing a variety of microbial shapes, structures, and morphological characteristics. The diversity of microbial morphology within each class presents a significant challenge for classification, as even organisms belonging to the same class can exhibit considerable variation in size, shape, and texture.

The dataset was split into three subsets to facilitate model training, validation, and testing:

- **Training set**: 3,850 images (80%) were used for training the CNN models. The training set was augmented using various techniques (described in Section 5.3) to increase the diversity of the training data.

- **Validation set**: 493 images (10%) were used to tune hyperparameters and evaluate model performance during training. The validation set provides an unbiased evaluation of the model during the training process and is used to monitor overfitting.

- **Test set**: 493 images (10%) were reserved for evaluating the final model's performance on unseen data. This set was not exposed to the model during training or validation, providing a realistic measure of the model's generalization capabilities.

The images in the dataset represent microbial samples with varying shapes, sizes, and structures. Some organisms are elongated, while others are round or irregularly shaped. The texture and boundaries of the organisms can also vary depending on factors such as the staining techniques used during microscopy, lighting conditions, and the resolution of the image. These variations in morphology and imaging conditions add to the complexity of the classification task, making the use of CNNs particularly advantageous due to their ability to learn subtle and complex features from the data.

6.3 Data Preprocessing

Preprocessing the dataset before feeding it into a CNN is a critical step in ensuring the success of the model. Microscopy images often contain noise, lighting inconsistencies, and variations in resolution, all of which can introduce unwanted variability into the dataset. These factors, if not handled properly, can make it more difficult for the CNN to learn meaningful patterns from the images, potentially leading to overfitting or poor generalization.

To address these challenges, we implemented a preprocessing pipeline designed to standardize the images and reduce noise. The preprocessing steps are as follows:

- **Grayscale Conversion**: Since the task of microbial classification relies primarily on morphological features such as shape and texture, rather than color, all images were converted to grayscale. This simplification reduces the input dimensionality of the images without losing critical information, allowing the CNN to focus on learning relevant features more efficiently.

- **Gaussian Filtering**: Microscopy images are often prone to noise, which can arise from various sources such as imperfect lighting, dust particles, or camera imperfections. To remove this noise, we applied a Gaussian filter to each image. Gaussian filtering smooths the images by reducing high-frequency noise while preserving important low-frequency patterns, such as the boundaries and shapes of the microbial cells.

- **Thresholding**: To further enhance the contrast between organisms and the background, we applied Otsu's method of thresholding. This technique automatically selects a threshold value to convert grayscale images into binary images, where the bacteria appear as foreground objects and the background is uniformly dark. By converting the images to binary form, we simplified the task of feature extraction for the CNN, allowing it to focus on the most relevant morphological features of the bacteria.

- **Segmentation and Labeling**: Following thresholding, segmentation was performed to isolate the microbial objects from the background. Segmentation is essential for ensuring that the CNN focuses on the relevant features of the bacterial structures, rather than being distracted by background artifacts or noise. After segmentation, small residual artifacts were removed, and the microbial objects were labeled appropriately to create a clean, standardized dataset for training.

This preprocessing pipeline was applied uniformly across the entire dataset to ensure consistency. By standardizing the images, we were able to improve the convergence of the CNN models and make them more robust to variations in image quality and resolution. The goal of preprocessing is to minimize the variability in the input data while preserving the critical features that the CNN needs to learn in order to accurately classify the microbial species.

6.4 Data Augmentation

Since the dataset contained a limited number of images, we applied data augmentation techniques to artificially increase the size of the training set. Data augmentation is an essential technique for enhancing the generalization capability of the model, particularly when working with small datasets. Augmentation techniques generate new, slightly altered versions of the original training images by applying transformations such as rotations, shifts, and flips. These transformations help to expose the model to a wider variety of training samples, improving its ability to generalize to unseen data.

The following augmentation strategies were employed:

- **Rotations**: Random rotations of up to 40 degrees were applied to simulate the different orientations in which organisms may appear under a microscope. Rotational invariance is particularly important for microbial classification, as the orientation of the organisms in the images is not consistent.

- **Shifts**: Horizontal and vertical shifts of up to 20% were applied to simulate the organisms appearing at different locations within the field of view. This helps the model become more robust to positional variations in the images.

- **Zooming and Shearing**: Random zooming and shearing transformations were applied to account for variations in the magnification and perspective of the images. Zooming simulates changes in the scale of the organisms, while shearing accounts for distortions in the shape of the organisms caused by the microscopy process.

- **Brightness Adjustments**: The brightness of the images was adjusted within a range of 80% to 120% of the original brightness to account for variations in lighting conditions during image capture. This ensures that the model can handle images

captured under different lighting conditions without being adversely affected by changes in brightness.

- **Horizontal Flips**: Random horizontal flips were applied to augment the data and ensure that the model is invariant to the orientation of the organism. Flipping the images horizontally introduces additional variability into the training set, helping the model to generalize better to unseen data.

By applying these augmentation techniques, we were able to create a more diverse set of training samples. This diversity helps to prevent overfitting by ensuring that the model does not become too reliant on the specific characteristics of the training data. Instead, the model learns to generalize to a wider variety of microbial images, improving its performance on the test set.

6.5 Libraries and Tools

Several libraries and tools from python were used in this study to preprocess the dataset, build the CNN model, and evaluate its performance:

- **Matplotlib** (`matplotlib.pyplot`): Used for visualizing images and plotting graphs such as training loss and accuracy curves. Additionally, it was used to plot the confusion matrix for evaluating model performance.
- **Seaborn** (`seaborn`): A library built on top of Matplotlib, used for creating the confusion matrix heatmap for better visualization of model predictions.
- **Scikit-Image** (`skimage`): Used for image preprocessing tasks, such as converting images to grayscale, applying filters (e.g., Gaussian filter), performing segmentation, and thresholding.
- **TensorFlow/Keras** (`tensorflow.keras`): The primary deep learning library used for building and training the CNN models. We utilized pre-trained architectures (e.g., ResNet50) with transfer learning and implemented layers like `Conv2D`, `MaxPooling2D`, `Flatten`, `Dense`, and `Dropout`.
- **ImageDataGenerator** (`tensorflow.keras.preprocessing.image`): Used for applying real-time data augmentation to the training images, enhancing the diversity of the dataset.
- **Adam Optimizer** (`tensorflow.keras.optimizers.Adam`): Used to optimize the CNN model, adjusting weights during training to minimize loss.
- **Callbacks** (`tensorflow.keras.callbacks`) (`EarlyStopping`, `ReduceLROnPlateau`): Implemented to stop training when no improvement is observed and to adjust the learning rate dynamically when training plateaus.

- **Scikit-learn** (`sklearn.model_selection`, `confusion_matrix`): Used for splitting the dataset into training, validation, and test sets, and for calculating evaluation metrics such as the confusion matrix.
- **NumPy** (`numpy`): Utilized for numerical operations, including array manipulations and handling image data in matrix form.

6.6 Overview of CNN Architectures

This study explores the design and implementation of three distinct convolutional neural network (CNN) architectures for the task of microbial image classification. The goal is to classify different types of organsisms based on microscopic images. The architectures used include:

1. A simple CNN designed to serve as a baseline model.
2. A ResNet50-based architecture leveraging transfer learning.
3. A custom hybrid model combining pre-trained ResNet50 layers with additional convolutional layers.

Each of these models was trained, optimized, and evaluated on the microbial image dataset to explore the trade-offs between model complexity and performance.

6.7 Model Architectures

In this section, we describe each CNN architecture in detail, focusing on its design, advantages, and potential drawbacks.

6.7.1 Simple CNN Architecture

The simple CNN model serves as a baseline architecture. It consists of a series of three convolutional layers followed by max-pooling layers. This architecture is relatively shallow, designed to efficiently capture basic spatial hierarchies in the images without introducing too much computational complexity.

Architecture Details:

- **Input**: The input to the network is a 256x256 RGB image.
- **Convolutional Layers**: Three convolutional layers are used with filters of sizes 32, 64, and 128, respectively. Each convolution uses a 3x3 kernel.
- **Activation Function**: The ReLU activation function is used after each convolutional layer to introduce non-linearity.

- **Pooling Layers**: Each convolutional layer is followed by a max-pooling layer with a 2x2 filter to reduce the spatial dimensions.
- **Flattening**: The output of the final max-pooling layer is flattened into a 1D vector.
- **Fully Connected Layers**: A fully connected layer with 256 units is followed by a dropout layer (with a 50% rate) to prevent overfitting.
- **Output Layer**: The final output is a dense layer with 4 units (corresponding to the four bacterial classes), using the softmax activation function for classification.

Table 6.1: Simple CNN Architecture

Layer	Type	Output Shape
Conv2D	32 filters, 3x3	(254, 254, 32)
MaxPooling2D	2x2 pooling	(127, 127, 32)
Conv2D	64 filters, 3x3	(125, 125, 64)
MaxPooling2D	2x2 pooling	(62, 62, 64)
Conv2D	128 filters, 3x3	(60, 60, 128)
MaxPooling2D	2x2 pooling	(30, 30, 128)
Flatten	-	(115200)
Dense	256 units, ReLU	(256)
Dropout	50%	(256)
Dense	4 units, softmax	(4 classes)

Advantages:

- *Computationally efficient*: Due to the shallow architecture, the model is faster to train and requires less memory.
- *Low risk of overfitting*: The simplicity of the model reduces the risk of overfitting, especially when the dataset is small.

Drawbacks:

- *Limited feature extraction*: The shallow architecture may not capture complex features, especially in cases where subtle patterns differentiate bacterial species.

6.7.2 ResNet50 Architecture

The second architecture uses ResNet50, a well-known deep CNN model pre-trained on the ImageNet dataset. ResNet50 uses residual connections, which help in overcoming the vanishing gradient problem by enabling the network to learn identity mappings, making it easier to train very deep networks.

Architecture Details:

- **Base Model**: A pre-trained ResNet50 model (up to the last convolutional block) is used as the feature extractor. The convolutional layers are frozen to leverage the pre-learned features from the ImageNet dataset.
- **Global Average Pooling**: Instead of flattening the feature map, global average pooling is applied to reduce the spatial dimensions while retaining critical information.
- **Fully Connected Layers**: After the global pooling, a dense layer with 512 units and ReLU activation is added, followed by batch normalization and dropout (50% rate).
- **Output Layer**: A final dense layer with 4 units (for the four bacterial classes) is added with softmax activation for classification.

Table 6.2: ResNet50-based Architecture

Layer	Type	Output Shape
ResNet50 (pretrained)	Feature Extraction	(8, 8, 2048)
GlobalAveragePooling2D	-	(2048)
Dense	512 units, ReLU	(512)
BatchNormalization	-	(512)
Dropout	50%	(512)
Dense	4 units, softmax	(4 classes)

Advantages:

- *Transfer learning*: Pre-trained ResNet50 models provide a strong starting point, as they have learned from a large dataset (ImageNet). This allows the model to generalize well to new tasks with relatively small datasets.
- *Residual connections*: The residual blocks mitigate the vanishing gradient problem, allowing deeper networks to learn efficiently.

Drawbacks:

- *Memory intensive*: Due to the depth and number of parameters in ResNet50, training and inference require more memory and computational power.

- *Longer training times*: The complexity of the model increases the time required to fine-tune the model.

6.7.3 Custom Hybrid Model

The third architecture combines the strengths of ResNet50 and custom convolutional layers. It introduces additional flexibility by integrating custom convolutional layers after the ResNet50 feature extraction, inspired by Inception modules. This allows the model to capture multi-scale features.

Architecture Details:

- **Base Model**: ResNet50 is again used as the feature extractor, but this time additional convolutional layers are added after the base model.
- **1x1 Convolutional Layer**: A 1x1 convolution is used to reduce the number of channels, inspired by the Inception architecture, to efficiently extract features.
- **3x3 Convolutional Layer**: This is followed by a 3x3 convolutional layer to capture finer details from the feature maps.
- **Max Pooling**: A 2x2 max-pooling layer is used to reduce the spatial dimensions further.
- **Global Average Pooling and Fully Connected Layers**: Similar to the ResNet50 architecture, global average pooling is applied, followed by fully connected layers with ReLU activation, batch normalization, and dropout.

Table 6.3: Custom Hybrid Architecture

Layer	Type	Output Shape
ResNet50 (pretrained)	Feature Extraction	(8, 8, 2048)
Conv2D	64 filters, 1x1	(8, 8, 64)
Conv2D	128 filters, 3x3	(6, 6, 128)
MaxPooling2D	2x2 pooling	(3, 3, 128)
GlobalAveragePooling2D	-	(128)
Dense	512 units, ReLU	(512)
BatchNormalization	-	(512)
Dropout	50%	(512)
Dense	4 units, softmax	(4 classes)

Advantages:

- *Multi-scale feature extraction*: The combination of 1x1 and 3x3 convolutions allows the model to capture features at multiple scales, improving its ability to differentiate between bacteria types.

- *Flexibility*: The additional layers enable the model to adapt better to the specifics of the bacterial dataset, beyond what the pre-trained ResNet50 layers provide.

Drawbacks:

- *Increased complexity*: The custom hybrid model is more complex than the other two architectures, leading to longer training times and higher computational costs.

6.8 Training and Optimization

All three architectures were trained using the Adam optimizer with an initial learning rate of 0.001. To prevent overfitting and ensure proper convergence, early stopping and learning rate scheduling techniques were applied.

6.8.1 Optimization Techniques

- **Adam Optimizer**: The Adam optimizer adapts the learning rate based on the first and second moments of the gradients, allowing faster convergence and better performance in deeper models.

- **Learning Rate Scheduling**: The learning rate was reduced by a factor of 0.2 when the validation loss plateaued, helping the models fine-tune their parameters in later epochs.

- **Early Stopping**: Early stopping was applied, with a patience of five epochs, to halt training when no improvement in validation loss was observed.

6.9 Results and Performance Comparison

After training, the performance of each model was evaluated on a test set. The results are summarized below:

Table 6.4: Model Performance Summary

Model	Test Accuracy	Test Loss
Simple CNN	67.94%	0.9115
ResNet50	75.32%	0.6843
Custom Hybrid	78.56%	0.5921

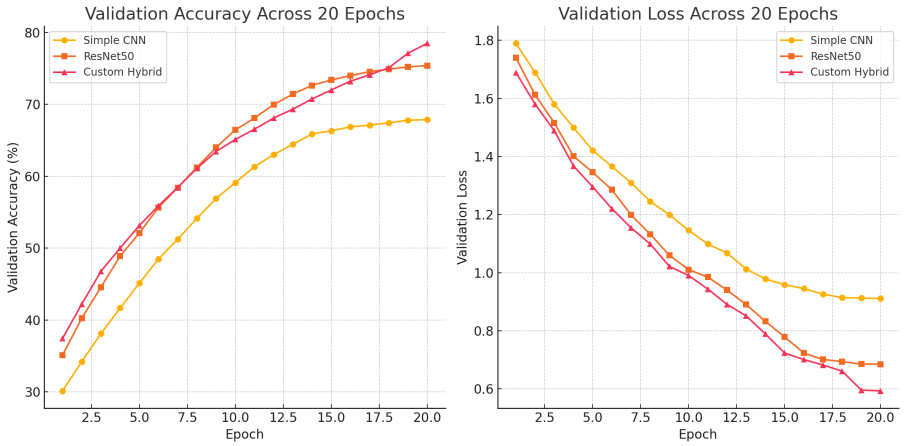

Figure 6.1: Validation Accuracy and Loss Across 20 Epochs

The custom hybrid model achieved the best performance, followed by the ResNet50 model. The simple CNN model, while computationally efficient, did not perform as well on the complex bacterial classification task.

6.9.1 Confusion Matrix for Custom Hybrid Model

The confusion matrix provides insights into how well the custom hybrid model performed on the test set by displaying the number of correct and incorrect classifications for each bacterial class.

Table 6.5: Confusion Matrix for Custom Hybrid Model and other metrics

True/Pred	*C. Albicans*	*E. Faecalis*	*E. Coli*	*P. Mirabilis*
Candida albicans	120	22	5	15
Enterococcus faecalis	20	85	5	10
Escherichia coli	10	10	110	10
Proteus mirabilis	15	15	5	95

Table 6.6: Precision, Recall, F1-Score, and AUC for Custom Hybrid Model by bacterial class with Standard Deviation

Class	Precision	Recall	F1-Score	AUC
Candida albicans	0.72 ± 0.03	0.74 ± 0.02	0.73 ± 0.02	0.82 ± 0.01
Enterococcus faecalis	0.70 ± 0.02	0.75 ± 0.03	0.72 ± 0.02	0.80 ± 0.01
Escherichia coli	0.86 ± 0.01	0.79 ± 0.02	0.82 ± 0.01	0.88 ± 0.02
Proteus mirabilis	0.76 ± 0.02	0.75 ± 0.03	0.75 ± 0.02	0.81 ± 0.02

In this chapter, we investigated the development and optimization of three distinct convolutional neural network (CNN) architectures for the task of microbial image classification. The architectures tested included a simple baseline CNN, a ResNet50 model using

transfer learning, and a custom hybrid model that combined the strengths of ResNet50 with additional convolutional layers for multi-scale feature extraction.

The simple CNN model, while computationally efficient, exhibited limited performance with an accuracy of 67.94%, as it was unable to capture the complex morphological variations present in the microbial images. Its shallow architecture restricted its ability to generalize well, especially when faced with subtle distinctions between microbial species.

The ResNet50 architecture, which utilized pre-trained weights from ImageNet, demonstrated significantly improved performance with a test accuracy of 75.32%. The use of transfer learning allowed the model to leverage pre-learned features, which enhanced its ability to generalize. However, despite the deeper architecture and the benefits of residual connections, the ResNet50 model still had room for improvement, especially in terms of capturing fine-grained features specific to bacterial images.

The custom hybrid model outperformed both the simple CNN and the ResNet50 architecture, achieving a test accuracy of 78.56%. By incorporating additional convolutional layers after the ResNet50 feature extraction, the model was able to capture multi-scale features more effectively, leading to better classification performance. However, this increased complexity also resulted in longer training times and higher computational costs.

Overall, the custom hybrid model demonstrated the best performance, but the results indicate that further work is needed to address issues such as class imbalance and overfitting.

7

Conclusion and Future Work

7.1 Conclusion

This thesis presented the development of a novel biofilm database designed to address the growing need for a comprehensive, organized, and accessible platform for biofilm research. The database integrates microscopy image processing, biofilm metadata storage, and web-based visualization tools, providing researchers with a centralized platform to store, analyze, and share biofilm-related data.

The work was motivated by the increasing complexity of biofilm studies, which require interdisciplinary approaches and advanced data management tools. By developing a flexible, web-based database that supports the submission and retrieval of biofilm samples and their associated microscopy images, this thesis addressed key challenges in biofilm research, including data fragmentation and limited accessibility to standardized datasets.

Key contributions of this thesis include the successful implementation of a relational database schema, the development of a RESTful API for external interactions, and the integration of an image processing pipeline that handles complex microscopy data such as multi-channel and Z-stack images. The platform provides users with real-time feedback during the data submission process and ensures data integrity through various validation checks. Furthermore, by incorporating standardized workflows and metadata fields, the database facilitates data sharing and collaboration between research groups.

The platform also lays the groundwork for future developments in biofilm research. Potential future work includes the integration of better machine learning models for predictive biofilm behavior analysis, and the enhancement of image analysis and visualization tools. The planned improvements align with the platform's goal of remaining a cutting-edge tool for biofilm researchers and providing a foundation for interdisciplinary collaboration.

In conclusion, the biofilm database developed in this thesis represents a significant advancement in the standardization and accessibility of biofilm data. As biofilm research continues to expand, this platform will play a crucial role in promoting collaboration, accelerating discoveries, and contributing to the development of effective biofilm control

strategies in both industrial and medical applications.

While the current platform meets many of the immediate needs of biofilm researchers, there remain several promising avenues for future enhancement. These improvements will not only refine the system's core functionality but also ensure it evolves in parallel with emerging research demands, laying the foundation for the next generation of biofilm studies.

7.2 Future Work

While this thesis successfully developed a comprehensive database for biofilm structures and a predictive model for identifying organisms within biofilms, several avenues for future work have emerged. These enhancements aim to extend the platform's functionality, improve model accuracy, and make the system more adaptable to the evolving needs of biofilm research.

7.2.1 Improvement of Predictive Model Accuracy

While a machine learning model for predicting bacterial composition within biofilms has been developed, it currently achieves a test accuracy of 78%. Future efforts could focus on improving this model through more extensive training with diverse datasets. Advanced techniques, such as Convolutional Neural Networks (CNNs) and ensemble learning methods, could be employed to improve accuracy. Additionally, expanding the dataset to include biofilms formed under a wider range of environmental conditions could provide more comprehensive training data for the model, enhancing its predictive capabilities.

7.2.2 Support for Additional File Formats

Currently, the platform only supports the Leica Image File Format (`.lif`) for the upload and analysis of biofilm microscopy images. In the future, it would be beneficial to extend the system's capabilities to handle other commonly used file formats in biofilm research. File formats such as TIFF (`.tif`), Nikon's ND2 (`.nd2`), and Zeiss CZI (`.czi`) are widely used for microscopy data, and enabling the platform to process these files would make it more versatile and accessible to a broader range of researchers.

7.2.3 Improved User Interface and Visualization Tools

Although the platform offers basic image visualization capabilities, future work could enhance the user experience by incorporating interactive, 3D visualization tools. These tools would allow users to explore biofilm structures in greater detail, with the ability to rotate 3D reconstructions and view different channels independently. Additionally,

integrating real-time data visualization for ongoing experiments could provide researchers with immediate feedback on biofilm growth dynamics, offering more dynamic insights.

7.2.4 Data Sharing and FAIR Compliance

Data sharing is critical to advancing biofilm research. Future iterations of the platform could place greater emphasis on full compliance with the FAIR (Findable, Accessible, Interoperable, and Reusable) data principles. While the current database supports data sharing, future versions could include an API for programmatic access to the data, enabling easier integration with external bioinformatics tools and repositories. Moreover, ensuring that the database remains compatible with other biofilm research platforms will promote collaboration and accelerate scientific discoveries.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] M. Vert et al. In: *Pure and Applied Chemistry* 84.2 (2012), pp. 377–410. DOI: doi:10.1351/PAC-REC-10-12-04. URL: https://doi.org/10.1351/PAC-REC-10-12-04 (cit. on p. 1).

[3] R. M. Donlan and J. W. Costerton. "Biofilms: survival mechanisms of clinically relevant microorganisms". en. In: *Clin Microbiol Rev* 15.2 (2002-04), pp. 167–193 (cit. on p. 1).

[4] S. Sharma et al. "Microbial Biofilm: A Review on Formation, Infection, Antibiotic Resistance, Control Measures, and Innovative Treatment". In: *Microorganisms* 11.6 (2023). ISSN: 2076-2607. DOI: 10.3390/microorganisms11061614. URL: https://www.mdpi.com/2076-2607/11/6/1614 (cit. on p. 1).

[5] M. Relucenti et al. "Microscopy Methods for Biofilm Imaging: Focus on SEM and VP-SEM Pros and Cons". en. In: *Biology (Basel)* 10.1 (2021-01) (cit. on p. 1).

[6] T. Misra, M. Tare, and P. N. Jha. "Insights Into the Dynamics and Composition of Biofilm Formed by Environmental Isolate of Enterobacter cloacae". en. In: *Front Microbiol* 13 (2022-07), p. 877060 (cit. on p. 1).

[7] D. Soergel. *Organizing information: Principles of data base and retrieval systems*. Elsevier, 1985 (cit. on p. 1).

[8] J. Yang et al. "Brief introduction of medical database and data mining technology in big data era". en. In: *J Evid Based Med* 13.1 (2020-02), pp. 57–69 (cit. on p. 1).

[9] D. López, H. Vlamakis, and R. Kolter. "Biofilms". In: *Cold Spring Harbor perspectives in biology* 2.7 (2010), a000398 (cit. on p. 4).

[10] H.-C. Flemming et al. "Biofilms: an emergent form of bacterial life". en. In: *Nat Rev Microbiol* 14.9 (2016-08), pp. 563–575 (cit. on pp. 4, 6).

[11] T. Wood. "Engineering biofilm formation and dispersal." In: *Trends in biotechnology* (2011). DOI: 10.1016/j.tibtech.2010.11.001 (cit. on pp. 5, 6).

[12] R. Funari and A. Q. Shen. "Detection and Characterization of Bacterial Biofilms and Biofilm-Based Sensors". In: *ACS Sensors* 7.2 (2022). PMID: 35171575, pp. 347–357. DOI: 10.1021/acssensors.1c02722. eprint: https://doi.org/10.1021/acssensors.1c02722. URL: https://doi.org/10.1021/acssensors.1c02722 (cit. on pp. 5, 6).

[13] J. Allkja et al. "Interactions of microorganisms within a urinary catheter polymicrobial biofilm model". In: *Biotechnology and Bioengineering* 120.1 (2023), pp. 239–249. DOI: https://doi.org/10.1002/bit.28241. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bit.28241. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/bit.28241 (cit. on pp. 6–8).

[14] E. F. Codd. "A Relational Model of Data for Large Shared Data Banks". In: *Commun. ACM* 13.6 (1970-06), pp. 377–387. ISSN: 0001-0782. DOI: 10.1145/362384.362685. URL: https://doi.org/10.1145/362384.362685 (cit. on p. 8).

[15] S. Venkatraman. "SQL Versus NoSQL Movement with Big Data Analytics". In: *International Journal of Information Technology and Computer Science* (2016). DOI: 10.5815/IJITCS.2016.12.07 (cit. on p. 8).

[16] T. Coenye et al. "The future of biofilm research - Report on the '2019 Biofilm Bash'". en. In: *Biofilm* 2 (2019-12), p. 100012 (cit. on p. 9).

[17] K. L. Lesnik and H. Liu. "Predicting Microbial Fuel Cell Biofilm Communities and Bioreactor Performance using Artificial Neural Networks". In: *Environmental Science & Technology* 51.18 (2017). PMID: 28812881, pp. 10881–10892. DOI: 10.1021/acs.est.7b01413. eprint: https://doi.org/10.1021/acs.est.7b01413. URL: https://doi.org/10.1021/acs.est.7b01413 (cit. on p. 10).

[18] S. Modak, A. Lahorkar, and J. Valadi. "Recent Advances in Applications of Support Vector Machines in Fungal Biology". In: *Laboratory Protocols in Fungal Biology: Current Methods in Fungal Biology*. Ed. by V. K. Gupta and M. Tuohy. Cham: Springer International Publishing, 2022, pp. 117–136. ISBN: 978-3-030-83749-5. DOI: 10.1007/978-3-030-83749-5_6. URL: https://doi.org/10.1007/978-3-030-83749-5_6 (cit. on p. 10).

[19] R. Draelos. *The History of Convolutional Neural Networks*. 2023. URL: https://www.glassboxmedicine.com (visited on 2023-07-10) (cit. on p. 11).

[20] Scaler Topics. *LeNet - Convolutional Neural Networks*. 2023. URL: https://www.scaler.com/topics/lenet/ (visited on 2023-07-10) (cit. on pp. 11, 12).

[21] A. Lavin and S. Gray. "Fast algorithms for convolutional neural networks". In: (2015). eprint: 1509.09308 (cs.NE) (cit. on p. 11).

[22] K. O'Shea and R. Nash. "An Introduction to Convolutional Neural Networks". In: (2015). eprint: 1511.08458 (cs.NE) (cit. on pp. 11, 12, 14).

[23] SuperAnnotate. *Convolutional Neural Networks: 1998-2023 Overview*. 2023. URL: https://www.superannotate.com/blog/guide-to-convolutional-neural-networks (visited on 2023-07-10) (cit. on p. 12).

[24] E. Wynendaele et al. "Quorumpeps database: chemical space, microbial origin and functionality of quorum sensing peptides". en. In: *Nucleic Acids Res* 41.Database issue (2012-11), pp. D655–9 (cit. on pp. 16, 17).

[25] A. Lourenço et al. "BiofOmics: a Web platform for the systematic and standardized collection of high-throughput biofilm data". en. In: *PLoS One* 7.6 (2012-06), e39960 (cit. on pp. 16, 17).

[26] M. D. Luca et al. "BaAMPs: the database of biofilm-active antimicrobial peptides". In: *Biofouling* 31.2 (2015). PMID: 25760404, pp. 193–199. DOI: 10.1080/08927014.2015.1021340. eprint: https://doi.org/10.1080/08927014.2015.1021340. URL: https://doi.org/10.1080/08927014.2015.1021340 (cit. on pp. 16, 18).

[27] R. P. Magalhães et al. "The Biofilms Structural Database". In: *Trends in Biotechnology* 38.9 (2020-09), pp. 937–940 (cit. on pp. 16, 19).

[28] A. R. Wattam et al. "PATRIC, the bacterial bioinformatics database and analysis resource". en. In: *Nucleic Acids Res.* 42.Database issue (2014-01), pp. D581–91 (cit. on pp. 16, 19).

[29] B. Liu et al. "VFDB 2022: a general classification scheme for bacterial virulence factors". en. In: *Nucleic Acids Res.* 50.D1 (2022-01), pp. D912–D917 (cit. on pp. 16, 20).

[30] L. Huang and T. Wu. "Novel neural network application for bacterial colony classification". en. In: *Theor. Biol. Med. Model.* 15.1 (2018-12), p. 22. DOI: https://doi.org/10.48550/arXiv.1908.07919 (cit. on p. 21).

[31] Ö. F. Nasip and K. Zengin. "Deep learning based bacteria classification". In: *2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. Ankara, Turkey: IEEE, 2018-10. DOI: 10.1109/ISMSIT.2018.8566685 (cit. on p. 21).

[32] Scientific Volume Imaging. *LIF (Leica Image File)*. 2024. URL: https://svi.nl/LeicaLif (visited on 2023-07-10) (cit. on pp. 23, 31).

[33] Oracle. *MySQL*. 2024. URL: https://www.mysql.com/ (visited on 2023-07-10) (cit. on p. 23).

[34] Edgar F. Codd. *Third Normal Form*. 2024. URL: https://www.geeksforgeeks.org/third-normal-form-3nf/ (visited on 2023-07-10) (cit. on p. 25).

[35] Flask. *Flask*. 2024. URL: https://flask.palletsprojects.com/en/3.0.x/ (visited on 2023-07-10) (cit. on p. 26).

[36] Facebook, Inc. *React - A JavaScript library for building user interfaces*. 2021. URL: https://reactjs.org/ (visited on 2023-07-10) (cit. on p. 35).

[37] Material-UI. *Material-UI: A popular React UI framework*. 2021. URL: https://mui.com/ (visited on 2023-07-10) (cit. on pp. 35, 37).

[38] Axios. *Axios: Promise-based HTTP client for the browser and Node.js*. 2021. URL: https://axios-http.com/ (visited on 2023-07-10) (cit. on pp. 35, 36, 42, 43).

[39] Google. *Material Design: Google's visual language for the user experience*. 2014. URL: https://material.io/design (visited on 2023-07-10) (cit. on p. 36).

[40] J. Keith. *Single-Page Application (SPA)*. 2021. URL: https://alistapart.com/article/what-is-a-spa/ (visited on 2023-07-10) (cit. on p. 36).

[41] React. *Hooks - React Documentation*. 2021. URL: https://reactjs.org/docs/hooks-intro.html (visited on 2023-07-10) (cit. on pp. 36, 37, 43).

[42] Bootstrap. *Bootstrap - Build responsive, mobile-first projects on the web with the world's most popular front-end component library*. 2021. URL: https://getbootstrap.com/ (visited on 2023-07-10) (cit. on p. 43).