DeepGEEP: Data-Driven Prediction of Bacterial Biofilm Gene Expression Profiles

Hamidreza Arjmandi*, Christophe Corre†, Hamidreza Jahangir, Adam Noel*
*School of Engineering, University of Warwick, Coventry, UK,
†School of Life Sciences and Department of Chemistry, University of Warwick, Coventry, UK.

***Abstract*—Predicting the gene expression profile (GEEP) of bacterial biofilms in response to spatial, temporal, and concentration profiles of stimulus molecules holds significant potential across microbiology, biotechnology, and synthetic biology domains. However, the resource and time-intensive nature of experiments within Petri dishes presents significant challenges. Data-driven methods offer a promising avenue to replace or reduce such experiments given sufficient data. Through well-crafted data generation techniques, the data scarcity issue can be effectively addressed. In this paper, an innovative methodology is presented for generating GEEP data over Petri dish resulting from a specific chemical stimulus release profile. We subsequently introduce a two-dimensional convolutional neural network (2D-CNN) architecture to leverage the synthesized dataset to predict GEEP variations across bacterial biofilms within the Petri dish. Our approach, known as DeepGEEP, is applied to data generated by a particle-based simulator (PBS), affording us the flexibility to evaluate its efficacy. Our proposed method attains a significant level of accuracy in comparison to established benchmark models such as Linear SVM, Radial Basis Function SVM, Decision Tree, and Random Forest.**

***Index Terms*—Gene expression profile, Deep learning, Convolutional neural networks, Bacterial biofilm**

I. Introduction

The responses of cells within living organisms to diverse stimuli underscore the remarkable complexity of biological systems. These responses invariably entail the modulation of gene expression triggered by the given stimulus. Gene expression refers to the process by which information encoded in a gene is used to synthesize a functional gene product, such as a protein or RNA molecule. It is the fundamental mechanism by which cells respond to external stimuli and adapt to their environment [1].

Gene expression can be detected and visualized in a Petri dish experiment typically using various molecular biology techniques or reporter systems. For example, chemically inducible promoters can be engineered to drive the expression and production of fluorescent proteins, such as Green Fluorescent Protein (GFP), that emit light when exposed to specific wavelengths. By fusing a promoter upstream of the GFP gene, we can track and visualize gene expression under a fluorescence microscope or even under the naked eye in specific conditions.

A. Motivation

Bacteria are relatively simple living organisms but understanding their various gene functionalities could have a huge impact on our health and the environment. Bacteria grow in biofilms as communities that adhere to surfaces. Therefore, bacteria culture growing in a Petri dish is a very common environment for *in vivo* experiments.

Predicting the GFP expression profile (GEEP) of a bacteria biofilm over a Petri dish based on the chemical stimulus release profile (i.e., spatial, time, and concentration) has several potential applications. GEEP prediction can help researchers understand how biofilms respond to different chemical stimulus release profiles, aiding in the development of strategies to control or enhance biofilm formation in various natural and engineered environments [2]. Also, profile prediction can help to identify novel targets for antimicrobial therapies and develop strategies to disrupt or prevent biofilm formation [3], [4]. Understanding how biofilm behavior changes in response to varying stimulus release profiles helps to optimize bioremediation processes and improve pollutant removal efficiency [5].

In biotechnology and industrial applications, biofilms are used for various purposes, such as biofuel production [6]. Predicting GEEP can guide the design and optimization of bioprocesses involving biofilm formation [7]. Also, engineers and researchers working in synthetic biology can use the predicted GEEPs to design and engineer biofilms with specific behaviors or responses to different environmental stimuli. This could lead to the development of biosensors or biodevices that respond to changing conditions [8].

B. Different approaches for GEEP prediction

Conducting GEEP experiments within Petri dishes can be both expensive and time-consuming [9]. Attempting to replicate these experiments across a broad spectrum of situations to encompass all potential scenarios often becomes infeasible and hinders the exploration of many biophysical hypothesises. Furthermore, the outcomes of these experiments are susceptible to human errors, which can significantly impact the accuracy of results. Additionally, the extensive execution of such experiments contributes to environmental pollution, raising significant ecological concerns. Furthermore, personnel safety when conducting numerous experiments is another concern.

These drawbacks of experimental methodologies have paved the way for the adoption of computational modeling approaches, as highlighted in [10]. These computational strategies are broadly categorized into two distinct paradigms:

traditional computational parametric models and data-driven models.

Researchers have increasingly used computational parametric models employing partial differential equations (PDEs) to unravel the intricate processes governing signaling within bacterial cultures, e.g., quorum sensing. This line of inquiry aims to elucidate the spatio-temporal distribution of signaling molecules and its consequential impact on gene expression [11]–[14]. In [18], a biophysical model is introduced to predict the time-dependent concentration distribution of N-acyl homoserine lactone (AHL) within an agarose matrix. Authors in [19] conducted an experiment where a bacterial strain was embedded in a short agar lane. An exogenous autoinducer was introduced at one end of the lane to track the expression of a quorum sensing (QS) reporter over time and space, allowing the measurement of autoinducer diffusion effects along the lane. Furthermore, a droplet-based bacterial communication system was devised in [20]. This system employed *Escherichia coli* (*E. coli*) bacteria enclosed within water-in-oil emulsion droplets of significant populations. This work demonstrated the potential of this system for genetically-programmed pattern formation and distributed computing. Characterization of diffusion sensing by individual cells [15], the clustering patterns of bacterial aggregates within a biofilm matrix [16], the phenomena of collective sensing, and the adaptability of bacterial cultures to heterogeneous environments [17] are also facets of bacterial culture behavior that have received attention from a computational perspective.

However, conventional parametric methods suffer from significant limitations. The accurate estimation of parameters within these experimental setups is not readily attainable through a limited number of trials. Moreover, their adaptability is constrained, and their ability to obtain insights from experience is limited.

Data-driven approaches might initially appear to contradict the very essence of our objectives within this paper. One might assume that such methodologies necessitate an extensive array of experiments to generate the requisite data. Nonetheless, through a thoughtfully designed data-driven approach, we can effectively tackle this challenge by leveraging our existing comprehension of the system. By thoughtfully defining features and data samples based on our knowledge of the underlying processes, we can generate an extensive dataset from a single experiment, mitigating the aforementioned concern. The utilization of data-driven methodologies is increasing to extract features from the biofim images, such as individual cell geometry [22], and the classification of cells in comparison to other substances [23].

In this paper, we introduce a novel methodology for generating data, tailored to the data-driven prediction of gene expression profiles (DeepGEEP) within bacterial biofilms on a Petri dish in response to a specific chemical stimulus release profile. This approach is based on the notion that cell colonies on identical plates have different distances from the release area points (sources) and boundaries. We harness these attributes within the data generation technique, avoiding the need to teach the machine to extract these features from a multitude of images corresponding to numerous experiments. Subsequently, we propose the utilization of a two-dimensional convolutional neural network (2D-CNN) architecture, which harnesses the generated dataset to predict the GEEP within the bacterial biofilm across the Petri dish.

C. Contributions

In this study, we produce our data using a particle-based simulator (PBS). This simulator traces the movement of diffusing molecules within the Petri dish environment, granting us the flexibility to assess the viability of our approach. The PBS takes into account diffusion within the biofilm and agar, boundary conditions, and simplified gene expression process. Our research introduces several notable contributions:

- We present DeepGEEP as a deep learning strategy aimed at predicting Green Fluorescent Protein (GFP) expression profiles across the Petri dish, offering a potential reduction in the need for numerous costly and time-consuming experiments. This approach serves to address the limitations associated with computational parametric methods.
- We propose a novel approach to generate data from a limited number of simulations (experiments) to train the deep learning model. This method involves sampling and segmenting the experimental (simulation) environment, enabling us to produce a diverse array of data samples from each experiment (simulation).
- We introduce a specific 2D-CNN architecture that achieves a notably high level of accuracy when compared to benchmark models such as Linear support vector machine (SVM), Radial Basis Function SVM, Decision Tree, and Random Forest.

II. System Model and Problem Statement

A. System Model

In this section, we provide an overview of the fundamental model involving an agar Petri dish, which serves as the environment for bacterial cultivation, and the controlled release of chemical stimulus.

We assume a cylindrical Petri dish with a radius of r_p m filled with agar to a height of h_a m. We employ cylindrical coordinates with the origin situated at the center of the top of the agar. The coordinates (r, θ, z) correspond to the radial, azimuthal, and axial dimensions, respectively.

The starting bacterial culture, diluted with water, has a concentration denoted as C_b $\mathrm{mol} \cdot \mathrm{m}^{-3}$[1]. This concentration can be practically quantifiable through optical spectroscopy. A specified volume of V_b m^3 from this bacterial source is uniformly distributed over the entirety of the top agar surface.

We consider the dilution of signaling molecules by mixing them with water, resulting in a concentration denoted as C_m $\mathrm{mol} \cdot \mathrm{m}^{-3}$. A droplet of volume V_d m^{-3} of this solution is deposited at a specified point (r_0, θ_0, z_0) on the agar

[1] "mol" refers to the number of molecules

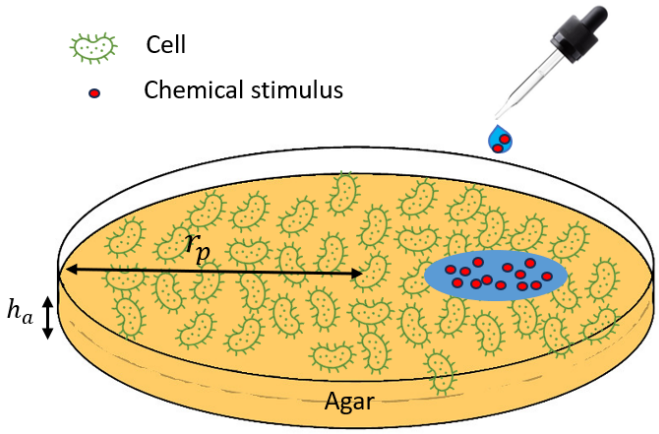


Fig. 1: Schematic of the system model. A droplet carrying the chemical stimulus is deposited at a specific point on the agar surface that is hosting the cells.

surface, forming an initial area of A_d m^2. As time progresses, the droplet, carrying the signaling molecules, infiltrates the diffusible agar medium and undergoes a reduction in size, resulting in a time-varying area denoted as $A_d(t)$. It is important to note that the initial height of the droplet, denoted as h_d, is determined by the ratio of droplet volume to the initial area, calculated as $V_d/A_d(0)$. The signaling molecules diffuse through the agar matrix and may interact with cells present on the surface through their diffusive movement.

After dropping the droplet onto the plate, the plate may be placed in an incubator (37°C) to enhance the growth of the bacteria and to form a biofilm. The cells' response to a stimulus may vary over the plate depending on several factors including the concentration of received molecules, cell-to-cell variation, and cell-to-cell communication. Given that the bacteria have been engineered to produce GFP, then we can observe a green pattern over the Petri dish when exposed to blue light. The goal is to predict the cell's response pattern (i.e., gene expression pattern) over the Petri dish given the release area of the droplet. Fig. 1 depicts the schematic of the system model.

Example: An experiment conducted in molecular microbiology laboratory studies focuses on the *E. coli* bacteria and Isopropyl β-D-1-thiogalactopyranoside (IPTG) as the signaling molecule.

In *E. coli* research, the gene responsible for encoding GFP is frequently controlled by an inducible promoter (e.g., the lac promoter) that responds to IPTG via the interaction of IPTG with the trancripsional repressor protein LacI. This strategic fusion enables researchers to regulate GFP expression by introducing IPTG. The pairing of an IPTG-responsive promoter with GFP forms a reporter gene construct used for visualizing gene expression within *E. coli*.

III. Data Generation

To generate the data, we employ a particle-based simulator. Leveraging the PBS enables us to modify parameters and systematically introduce, adjust, or modify the relevant processes as needed to investigate the feasibility of our approach for future experimental data. This strategy is notably efficient in terms of both time and cost.

A. Particle-based Simulator

To simulate the process, we begin by elaborating on the processes involved in *E. coli* GFP expression in response to releasing IPTG over the agar surface. Subsequently, we outline how the PBS effectively replicates each of these processes. The experiment comprises the following key processes:

Bacteria Spreading: As previously indicated, a solution containing bacteria (bacteria diluted in sterile water) with a concentration denoted as C_b is uniformly spread across the entire surface.

Droplet soaking:

Following the application of the bacteria solution, a droplet of IPTG solution is introduced. This droplet with a volume of V_d m^{-3} and the concentration C_m, is dropped at coordinates (r_0, θ_0, z_0) on the agar surface. This leads to the formation of a circular region with an area of A_d m^2 and an initial height of h_d.

Given the soaking rate of k_s $\mathrm{m\,s}^{-1}$ for the droplet into agar, the temporal evolution of the droplet's area is calculated as [21]

$$A_d(t) = A_d(0)e^{\frac{-k_s}{h_d}t}. \tag{1}$$

This soaking process serves as the molecule source in the PBS. To simulate this process within a given interval $[t, \Delta t]$, we account for the quantity of molecules that permeate the agar due to the soaking rate during this timeframe. Over this period, the droplet's area transforms from $A_d(t)$ to $A_d(t{+}\Delta t)$. Consequently, the corresponding change in volume is computed as $h_d(A_d(t) - A_d(t + \Delta t))$, which equates to $C_d h_0(A_d(t) - A_d(t + \Delta t))$ molecules. It is assumed that this quantity of molecules infiltrates uniformly across the area $A_d(t)$ of the droplet into the agar.

Diffusion in agar: Molecules residing within the agar undergo free diffusion. At each time increment, the positions of information molecules are adjusted in accordance with stochastic Brownian motion, occurring independently for each molecule.

The transition of a molecule in each spatial dimension across Δt seconds adheres to a Gaussian distribution with a mean of zero and a variance of $2D_a\Delta t$ in which D_a is the diffusion coefficient of IPTG in the agar. If a molecule's trajectory during this temporal span results in it colliding with the plate's boundaries, it is redirected back within the agar. Similarly, should a molecule intersect the surface, it may adhere to bacteria with some probability, or it can be reflected back.

IPTG sensing by bacteria: Once spread across the surface, bacteria grow at a rate that is accelerated within an incubator with elevated temperatures. The interaction between IPTG and bacteria on the surface depends on the population of bacteria present. For the sake of simplicity, we consider an irreversible

reaction between molecules (M) and cells (C) that sequesters the molecule from the environment:

$$\mathrm{M} + \mathrm{C} \xrightarrow{k_b(t)} \mathrm{MC}, \tag{2}$$

where $k_b(t)$ is the consumption rate of cell population area over the surface at time t. As cell growth progresses across the surface, the rate of consumption amplifies with time. To this end, we consider that the consumption rate increases linearly with the cell population. This assumption is described by the following equation:

$$k_b(t) = \frac{k_b(0)P_b(t)}{P_b(0)}, \tag{3}$$

where $P_b(t)$ represents the cell population at time t.

Bacteria growth over the plate: Bacteria undergo a proliferation process and develop a biofilm structure. During the growth phase, the *E. coli* cell population doubles very 20 minutes. However, this exponential proliferation eventually halts due to spatial and nutritional constraints. Based on experimental observations [21], we assume that bacterial growth ceases after H_b hours which is marked by an absence of discernible bacterial boundary expansion. Consequently, by considering the initial consumption rate $k_b(0)$, we can infer the time-dependent consumption rate during the growth stage as follows:

$$k_b(t) = k_b(0)2^{\min(t, H_b \times 3600)/(1200)}. \tag{4}$$

In cases where a molecule hits the bacterial surface, it stands the chance of being consumed, with a probability of $k_b(t)\Delta t$ during a time interval Δt seconds.

Biofilm formation and diffusion in biofilm: Another critical aspect of this experiment pertains to the moment of biofilm formation. At this time, molecules that hit the biofilm surface have the opportunity to infiltrate it, leveraging the water channels within to facilitate their diffusion [24]. We postulate that the biofilm forms after a duration of T_{bf} hours. Following this threshold time, molecules colliding with the surface can perfectly access the biofilm's water channels, resulting in significantly accelerated movement compared to diffusion within the agar medium. While the actual structure of these water channels is intricate [24], we simplify its representation as a form of free-surface diffusion, characterized by an effective diffusion coefficient D_b greater than that of the agar diffusion coefficient D_a.

Assuming that molecules hitting the surface after T_{bf} enter the biofilm, we can describe their behavior during a time increment Δt. During this interval, a molecule may either be absorbed by bacteria with a probability of $1 - \exp(-k_b\Delta t)$, or it can engage in random movement along the x and y directions, characterized by Gaussian-distributed random variables with zero mean and a variance of $2D_b\Delta t$.

In the event that a molecule is consumed/sequestered by a bacterium, it becomes permanently fixed at that location. This process results in the gradual consumption of nearly all molecules across the plate over time.

Gene Expression: Our experimental results [21] exhibit that bacteria respond to a specific range of concentration of IPTG. When the IPTG concentration surpasses a certain threshold, the bacteria might experience lethality or fail to exhibit a response. Conversely, concentrations below this range do not elicit gene expression. Building upon this observation, we establish a threshold range $[C_L, C_U]$, signifying the concentration levels within which the bacteria exhibit a response to IPTG as GFP expression.

IV. DeepGEEP Design

A. Training Data Generation

We can consider several approaches to generate training data. A seemingly straightforward method involves creating many distinct droplet areas and concentrations as training input data. These data take the form of 2D images, encapsulating the relevant information. In parallel, the resultant GFP patterns corresponding to each input image serve as the training output data. However, the execution of numerous time-consuming and expensive lab experiments poses a significant challenge and requires a resolution.

We introduce a novel approach that enables the resourceful generation of data using a limited number of experiments. The premise of this approach rests upon the notion that cell colonies on identical plates have different distance patterns from the release area points and boundaries. These attributes can be preprocessed and harnessed within the data generation for the machine learning framework, avoiding the need to teach the machine to extract these features from a multitude of images corresponding to numerous experiments.

Imagine the smallest square that completely encloses the circular plate. Now, picture this square being divided both vertically and horizontally into N_x equal parts. This division results in a matrix with dimensions $N_x \times N_x$. In this matrix, each element, represented by (j, l), actually stands for a small square piece within the surrounding square centered at a specific point (X_j, Y_l).

Let's approximate the total number of square pieces (i.e., samples) inside the circular plate as N_s and that each one will represent a group of bacteria. Each sample $i \in 1, \ldots, N_s$ is associated with a specific element in our matrix, let's call it $(j(i), l(i))$. The center of matrix element $(j(i), l(i))$ is located at $(X_{j(i)}, Y_{l(i)}, 0)$ on the plate.

To make things simple, we label each sample with a value for GFP expression, denoted as $GFP(i)$. This value can either be 0 or 1. When $GFP(i) = 1$, it means that the group of bacteria at $(X_{j(i)}, Y_{l(i)}, 0)$ has triggered the production of GFP. In the PBS, this indicates that the amount of IPTG molecules absorbed within this square piece is within the interval $[C_L, C_U]$. From the perspective of the experiments, it is like saying the fluorescent light is observed brighter than some threshold level.

We note that the location of each sample $i \in 1, \ldots N_s$, i.e., $(X_{j(i)}, Y_{l(i)})$ should be considered as part of the training input data. Then, for sample i we may consider a grid matrix $\mathcal{M}_1$

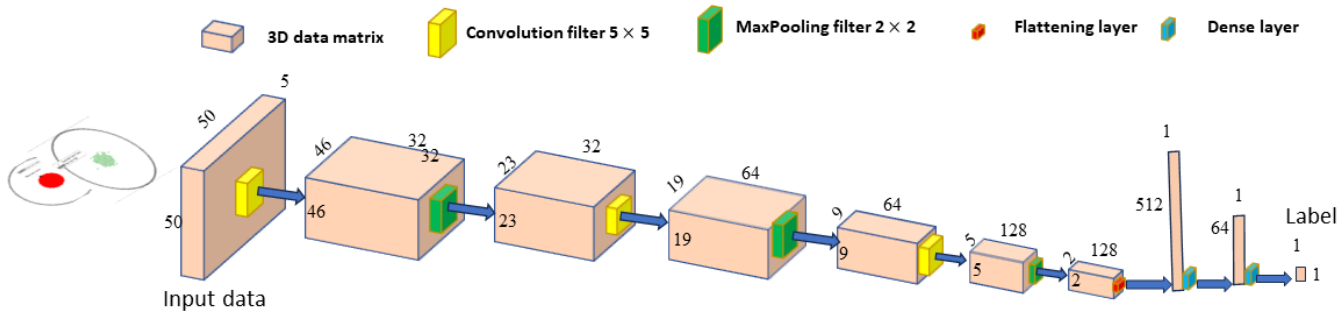


Fig. 2: Schematic structure of 2D-CNN

that is zero everywhere except the element $(j(i), l(i))$ corresponding with the location of the sample, i.e., $(X_{j(i)}, Y_{l(i)})$.

The position of each sample $i \in 1, \cdots N_s$, is incorporated as a component of our training input data. For each individual sample i, we define a grid matrix, $\mathcal{M}_1$. In this matrix, every element is initially set to zero except for the element $(j(i), l(i))$, which exactly corresponds to the location of that particular sample, namely $(X_{j(i)}, Y_{l(i)})$, i.e.,

$$\mathcal{M}_1(j,l) = \begin{cases} 1, & (j,l) = (j(i), l(i)) \\ 0, & \text{otherwise} \end{cases}. \tag{5}$$

The distance between each point within the system and the sample point holds significant informative value for the learning process due to the diffusion process. Then, we establish another grid matrix named $\mathcal{M}_2$. Within this matrix, each element (j,l) is calculated as the Euclidean distance between (X_j, Y_l) and the location of the sample at $(X_{j(i)}, Y_{l(i)})$, then normalized by a factor of $\frac{1}{2r_p}$, i.e., we have

$$\mathcal{M}_2(j,l) = \frac{(X_j - X_{j(i)})^2 + (Y_l - Y_{l(i)})^2}{2r_p}. \tag{6}$$

Another factor influencing the GFP expression (concentration) for a given sample is its positioning relative to the boundary points of the plate. To account for this aspect, we introduce the grid matrix $\mathcal{M}_3$ for each sample i. In this matrix, every element (j,l) located along the plate's border is determined by the square of the Euclidean distance between (X_j, Y_l) and the position of the sample at $(Xj(i), Y_{l(i)})$. This value is then normalized by a factor of $\frac{1}{2r_p}$. In other words, we calculate this distance-related effect for every element at the plate's border, considering its relationship to the sample's location, i.e.,

$$\mathcal{M}_3(j,l) = \frac{(X_j - X_{j(i)})^2 + (Y_l - Y_{l(i)})^2}{2r_p}. \tag{7}$$

Obviously, for higher distances from the sample, the boundary point has less impact.

Another aspect of the input data should account for the region where the stimulus molecules are released, as well as the quantity of molecules released. One challenge that might arise is the time aspect of molecule release. However, we can simplify the situation by ignoring the release time of molecules during the soaking. This assumption stems from the fact that molecules move independently, and we are observing the system at a steady state. Consequently, the timing of molecule release becomes irrelevant; what matters is where molecules are released.

The total number of molecules released within each square piece of the designated release area is determined using the PBS capturing the droplet soaking mechanism outlined in subsection III.A. This allows the PBS to effectively estimate the total molecules released within each square area. As a result, a portion of the input data is represented by a grid matrix, $\mathcal{M}_4$, where each element (j,l) denotes the accumulated number of molecules released within its corresponding region denoted by $N_{Rel}(j,l)$. It is evident that the maximum molecule count would be at the droplet's center, while the count decreases as we move towards the initial border of the droplet. In all other areas, the molecule count is zero. Therefore, we have

$$\mathcal{M}_4(j,l) = N_{Rel}(j,l). \tag{8}$$

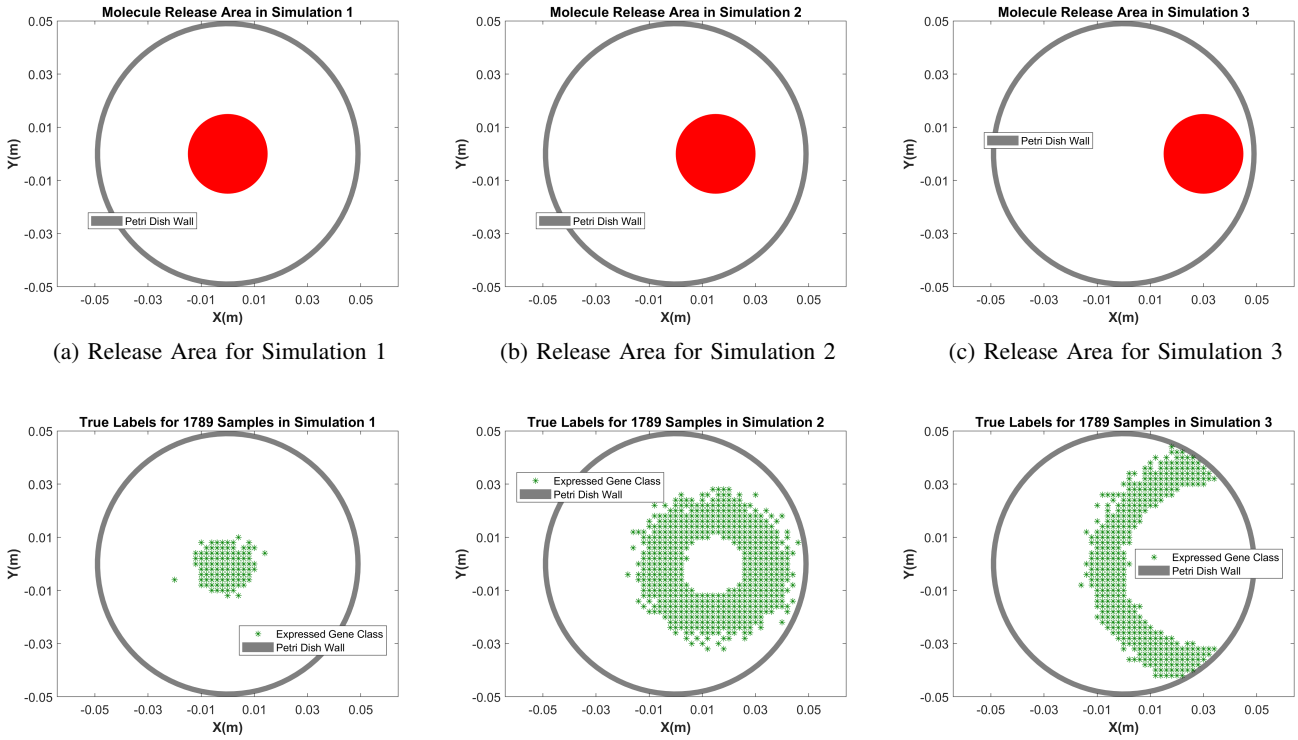
It is important to recognize that $\mathcal{M}_4(j,l)$ for a sample does not provide sufficient information to differentiate between the samples. This is because it remains the same for all samples within a single experiment or simulation. To enhance the efficiency of the training process, we introduce the grid matrix $\mathcal{M}_5$ whose element (j,l) within the release area is determined by the square of the Euclidean distance between (X_j, Y_l) and the location of the sample at $(X_{j(i)}, Y_{l(i)})$, i.e.,

$$\mathcal{M}_5(j,l) = (X_j - X_{j(i)})^2 + (Y_l - Y_{l(i)})^2 \times \mathbf{1}_{Rel}(j,l), \tag{9}$$

in which $\mathbf{1}_{Rel}(j,l)$ is a function that indicates whether the (j,l) element corresponds to a release point. We chose the square of the Euclidean distance because it is known that the peak concentration at a point in an unbounded environment is inversely related to the square of the distance between the release and observation points.

B. 2D-CNN structure

In our interpretation, the input data resembles an image of dimensions $N_x \times N_y$ with five channels. This suggests using the 2D-CNN structure for this type of data. Fig. 2 presents an overview of the architecture of the 2D-CNN devised for the



(a) Release Area for Simulation 1 (b) Release Area for Simulation 2 (c) Release Area for Simulation 3

(d) True Labels for Simulation 1 (e) True Labels for Simulation 2 (f) True Labels for Simulation 3

Fig. 3: Released area and true labels for Simulations 1, 2, and 3 for generating 3×1789 data samples.

proposed data structure aiming to identify the gene expression pattern. Given N_{sim} simulation senarios, the input dataset takes the form of $N_{sim} \times N_s$ samples, each with dimensions $N_x \times N_y \times 5$, while the output consists of binary classifications denoted as $0, 1$.

The architecture of the 2D-CNN is as follows: The first layer is a 2D convolutional layer with 32 filters using the ReLU activation function. The input shape for this layer is (N_x, N_y, 5), which corresponds to the size of each input sample. After the first convolutional layer, a 2D MaxPooling layer with a pool size of (2, 2) is applied. MaxPooling reduces the spatial dimensions of the feature maps, making the network more efficient and capable of capturing important information. The process is repeated with another 2D convolutional layer with 64 filters and a ReLU activation function, followed by another MaxPooling layer.

A third 2D convolutional layer with 128 filters and ReLU activation is added, followed by another MaxPooling layer. The output of the third MaxPooling layer is then flattened into a one-dimensional vector. A fully connected (Dense) layer with 64 units and ReLU activation is added to further process the flattened features. Finally, there is an output Dense layer with a single unit and a sigmoid activation function ensures that the model's prediction falls within the range [0, 1], allowing it to produce a probability indicating the likelihood of each input sample belonging to the positive class. This last layer is responsible for binary classification, as it outputs a probability value between 0 and 1.

The model is trained using 15 epochs and 32 batches, and it aims to learn patterns and features from the input data to perform the binary classification task.

V. Results and Discussion

A. Simulation Setup for Data Generation

We simulated three experiments ($N_{sim} = 3$) to generate training and test data. In each simulation, we considered a droplet of volume 10 μL containing IPTG placed on a surface. It sat like a circle with a radius of 0.015 meters on the surface, and it had a height of $h_0 = 1.4 \times 10^{-5}$ m.

For the first experiment, we imagined that the center of the droplet was right in the middle of the circle, $r = 0$, for the second experiment it was at the edge of the circle, $r = 0.015$, and for the third experiment, it was further away from the center $r = 0.03$ m. The shape of the setup was always like a circle.

Obviously, the system geometry is circular symmetric. Then, without loss of generality, we have assumed $\theta = 0$ for all three release points. The number of molecules inside the tiny droplet was set at 10^4, 10^5, and 10^6 for the first, second, and third simulations respectively. In simpler terms, we are carrying out these simulations with different numbers of molecules in the droplet for the three experiments. This covers a wide range of stimulus concentrations.

We have used a grid matrix with dimensions of 50×50 (with $N_x = 50$ and $N_y = 50$). Out of the total 2500 squares in this matrix, $N_s = 1789$ squares fall within the circular Petri

dish. This means that we can gather 1789 data samples from each simulation, resulting in a total of 3×1789 samples. Table II provides the detailed structure of the proposed 2D-CNN.

In each simulation, when the system reaches a stable state, the bacteria colonies within each small section of the Petri dish may or may not produce GFP (the gene may be expressed or not). This depends on various factors, including the distance from release points and the boundary points. Fig. 3 displays both the area where droplets are released and the labels for the square pieces within the Petri dish for all three simulations.

In the diagrams on the bottom of the figure, the green star symbols represent square areas the genes were expressed and produced GFP. The plain white represents the pieces where the gene was not expressed (i.e., GFP was not expressed). The gene expression corresponds to received IPTG concentrations between $C_L = 10/(0.1/50)^2$ and $C_U = 100/(0.1/50)^2$. The concentration of IPTG soaking into the release area is not uniform as mentioned before and is calculated using the model that takes into account the shrinking of the droplet, as described by equation (1). In this equation, the values are $k_s = 1.3 \times 10^{-6}$ ms^{-1} and $h_d = 1.4 \times 10^{-5}$ m.

Using this simulation configuration, we produced a total of 5367 data samples. Each input data has a structure resembling an image, with dimensions of 50×50 and 5 different channels. This setup is designed to work well with the proposed 2D-CNN model.

Additionally, we generated a separate set of data for evaluation purposes, achieved through Simulation 4. In this simulation, we released a total of 500,000 molecules, and the droplet's center was positioned at a distance of 0.02 m from the center ($r = 0.02$ m).

In Fig. 5, the top part of the figure presents both the release area and the actual labels used for this particular simulation.

B. Results

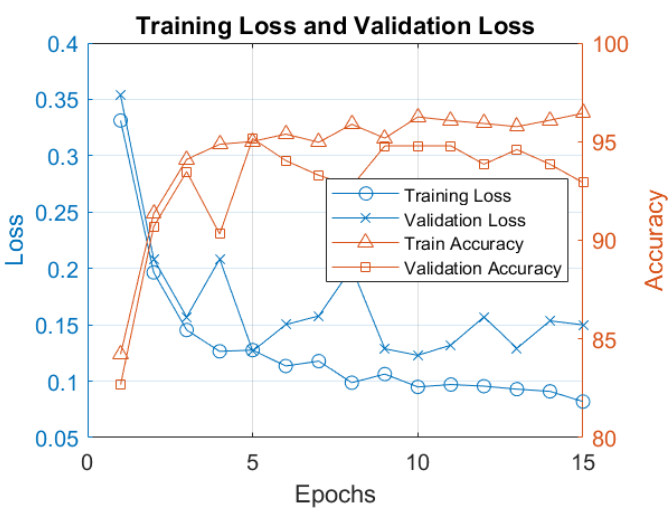


Fig. 4: Training loss and accuracy of proposed 2D CNN model.

We used the data we created through our simulation setup to train the proposed 2D-CNN model. This method was implemented using Python 3.10 along with the TensorFlow framework. For training, we used a loss function called categorical binary cross-entropy, and the training process was carried out using the Adam optimizer. You observe the general progress of the training in Fig. 4, which displays the training loss and accuracy for both the validation and test datasets.

TABLE I: Confusion Matrix for 2D CNN

		Predicted	
		Positive	Negative
Actual	Positive	472	103
	Negative	150	1064

TABLE II: Detailed Structure of 2D-CNN

Layer	Filter Size	Output Data Shape
Input	-	(50, 50, 5)
Conv2D MaxPooling2D	5×5 (2, 2)	(46, 46, 32) (23, 23, 32)
Conv2D MaxPooling2D	5×5 (2, 2)	(19, 19, 64) (9, 9, 64)
Conv2D MaxPooling2D	5×5 (2, 2)	(5, 5, 128) (2, 2, 128)
Flatten	-	512
Dense	-	64
Dense	-	1

The proposed 2D-CNN models achieves an accuracy of 0.85 when evaluated on the test data. In Fig. 5 (bottom), the left side shows the labels predicted by the 2D-CNN, and on the right, a visual comparison is made between the actual labels and those predicted. It's clear from the comparison that the 2D-CNN, utilizing the proposed data structure, has learned the gene expression system on the plate with high performance.

The confusion matrix is available in Table I. The corresponding false positive and false negative rates are 0.13 and 0.17, respectively, which is very low for this particular application. Notably, when examining the combined scatter plot depicted in Figure 5, we can observe that the errors in predictions occur mostly at the border of the gene-expressed region. This phenomenon could be attributed to the inherent noise in diffusion across the plate, causing some uncertainty in delineating the precise borders of the expressed region. This is evident in the scatter plot of the true class in Fig. 5 upper right.

We conducted a comparison of the performance metrics of the 2D-CNN with several benchmark models including Linear SVM, Radial Basis Function SVM, Decision Tree, and Random Forest. In Figure 6, we see the gene expression patterns predicted by all these benchmarks for the test data. These plots clearly illustrate that the 2D-CNN outperforms the benchmarks. For a detailed comparison of performance, Table III provides metrics including precision, recall, F1-score, and accuracy for both the CNN and the benchmark models.

While the accuracy of Linear SVM is 0.66, the Linear SVM

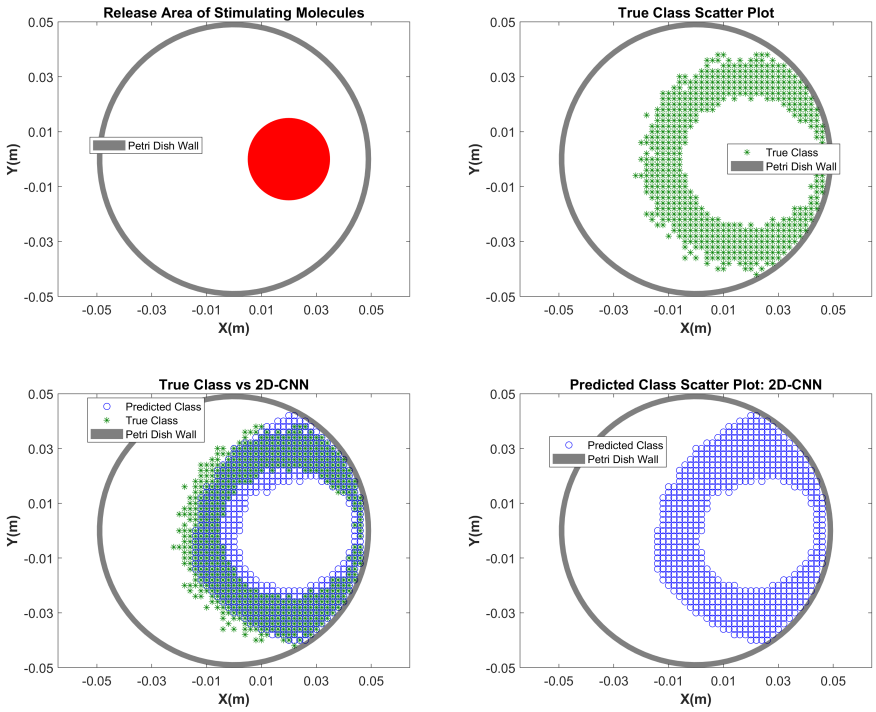


Fig. 5: Visual evaluation of 2D CNN prediction of gene expression for test data (Simulation 4) that includes 1789 samples.

TABLE III: Comparison of Model Performance

Model	Precision	Recall	F1-Score	Accuracy
2D CNN	0.76	0.82	0.79	0.85
Linear SVM	0.48	0.91	0.63	0.66
RBF SVM	0.33	0.30	0.31	0.58
Decision Tree	0.38	0.38	0.38	0.60
Random Forest	0.36	0.34	0.35	0.60

model exhibits a high recall value of 0.91, indicating a low rate of type II error (False Negative). The profile prediction of the Linear SVM, shown in Figure 6 (Upper left), demonstrates that the hyperplane provided by this approach divides the plate into two continuous regions: the positive label area that corresponds to both gene expressed area producing GFP and also the area without gene expression due to an excessive concentration of IPTG which is labelled, while the other encompasses the remaining plate area labeled as negative. This strategy allows Linear SVM to correctly predict almost all positive labels. However, it struggles with predicting negative labels (non-expressed regions), leading to a high rate of type I error (False Positive) (1-0.48) and an overall accuracy of only 0.66. This discrepancy arises because the provided data structure for training doesn't yield a hyperplane capable of accurately classifying elements in the non-expressed region due to the high concentration of molecules (surrounded by the expressed region).

Similarly, the Radial SVM also faces difficulty distinguishing within the expressed region, where the IPTG concentration is high and genes are not expressed.

Decision Tree and Random Forest models (with 100 estimators) yield similar outcomes. In contrast to the Linear SVM, they exhibit improved performance in identifying the inner non-expressed region, as anticipated. However, they struggle to detect positive labels compared to the Linear SVM. As a result, Decision Tree and Random Forest achieve an overall accuracy of 0.6.

VI. Conclusion

In this paper, we introduced DeepGEEP, a deep learning approach aimed at predicting gene expression profiles based on the observed stimulus release pattern. DeepGEEP is able to reduce the resource and time-intensive experiments within Petri dishes which is very common across microbiology, biotechnology, and synthetic biology domains. Our proposed method incorporates a novel data generation approach that attains high accuracy with just a few experiments. Leveraging this training data structure, we designed a 2D-CNN model to predict gene expression patterns that attains 0.85% accuracy. We compared the proposed model against benchmark methods, including Linear SVM, Radial SVM, Decision Tree, and Random Forest. The simulation results offer an adaptable, cost-effective means of gaining insights to refine our proposed deep learning structure and explore potential enhancements for deepGEEP performance. Looking ahead, our focus will shift to the application of this structure on real experimental data. We intend to delve into the creation of new features that could effectively instruct the 2D-CNN to make more accurate

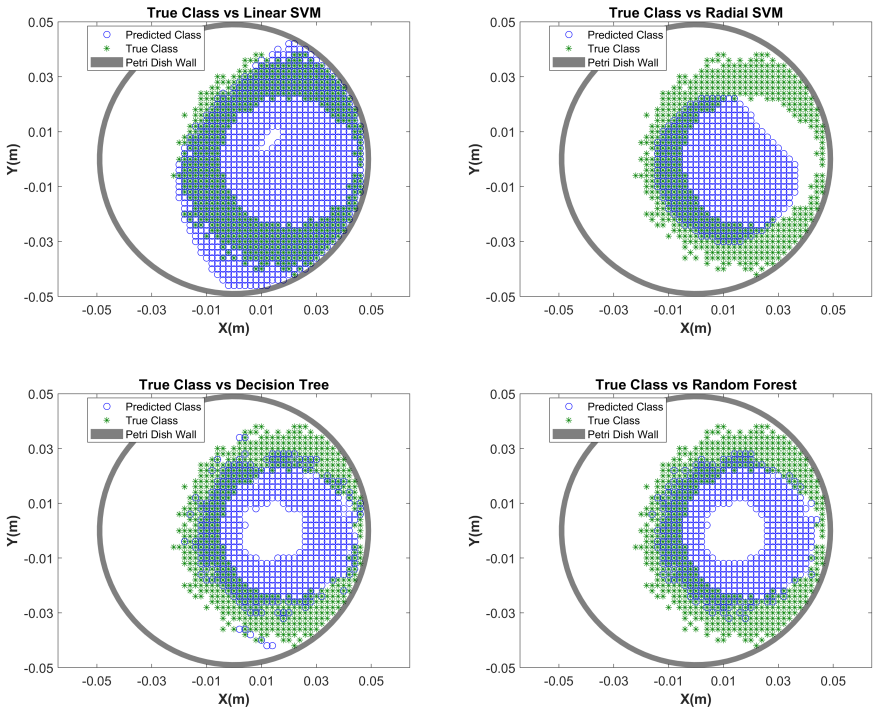


Fig. 6: Gene expression prediction by benchmarks of Linear SVM, RBF SVM, Decision Tree, and Random Forest.

predictions. Specifically, we aim to define composite features capable of encapsulating both diffusion and reaction properties inherent in the input data. Our current framework assumes a uniform spread of bacteria across the plate. Future efforts will involve introducing specified regions for bacterial activity, potentially leading to structural asymmetry. Furthermore, we aim to expand our approach to encompass variable release regions, moving beyond the current assumption of a constant release area. To accommodate these generalizations, we may need to embrace more intricate deep learning architectures, such as graph CNNs.

Acknowledgment

References

[1] Alberts, B. (2017). Molecular biology of the cell. Garland science.

[2] Pombo, J. P., Ebenberger, S. P., Müller, A. M., Wolinski, H., & Schild, S. (2022). Impact of gene repression on biofilm formation of Vibrio cholerae. Frontiers in Microbiology, 13, 912297.

[3] Ghuneim, L. A. J., Raghuvanshi, R., Neugebauer, K. A., Guzior, D. V., Christian, M. H., Schena, B., ... & Quinn, R. A. (2022). Complex and unexpected outcomes of antibiotic therapy against a polymicrobial infection. The ISME Journal, 16(9), 2065-2075.

[4] Aljeldah, M. M. (2022). Antimicrobial resistance and its spread is a global threat. Antibiotics, 11(8), 1082.

[5] Jhariya, U., Srivastava, S., Das, S., Bombaywala, S., Mahallea, S., & Dafale, N. A. (2022). Understanding the role of genetic and protein networking involved in microbial bioremediation. Bioremediation of Environmental Pollutants: Emerging Trends and Strategies, 187-219.

[6] Dutta, B., Nag, M., Lahiri, D., Ghosh, S., Dey, A., Joshi, S., & Ray, R. R. (2022). Biofilms for Biofuel Production. In Bio-Clean Energy Technologies Volume 2 (pp. 219-237). Singapore: Springer Nature Singapore.

[7] Contreas, L., Hook, A. L., Winkler, D. A., Figueredo, G., Williams, P., Laughton, C. A., ... & Williams, P. M. (2023). Linear Binary Classifier to Predict Bacterial Biofilm Formation on Polyacrylates. ACS Applied Materials & Interfaces, 15(11), 14155-14163.

[8] An, B., Wang, Y., Huang, Y., Wang, X., Liu, Y., Xun, D., ... & Zhong, C. (2022). Engineered living materials for sustainability. Chemical Reviews, 123(5), 2349-2419.

[9] Stepanov, D., Buchmann, D., Schultze, N., Wolber, G., Schaufler, K., Guenther, S., & Belik, V. (2022). A Combined Bayesian and Similarity-Based Approach for Predicting E. coli Biofilm Inhibition by Phenolic Natural Compounds. Journal of Natural Products, 85(10), 2255-2265.

[10] Delavar, M. A., & Wang, J. (2022). Advanced Methods and Mathematical Modeling of Biofilms: Applications in health care, medicine, food, aquaculture, environment, and industry. Academic Press.

[11] J. P. Ward, J. R. King, A. Koerber, P. Williams, J. Croft, and R. Sockett, "Mathematical modelling of quorum sensing in bacteria," Mathematical Medicine and Biology, vol. 18, no. 3, pp. 263–292, 2001.

[12] J. Muller and H. Uecker, "Approximating the dynamics of communicating cells in a diffusive medium by odes—homogenization with localization," Journal of mathematical biology, vol. 67, no. 5, pp. 1023–1065, 2013.

[13] M. Marenda, M. Zanardo, A. Trovato, F. Seno, and A. Squartini, "Modeling quorum sensing trade-offs between bacterial cell density and system extension from open boundaries," Scientific Reports, vol. 6, no. 1, pp. 1–10, 2016.

[14] B. A. Hense and M. Schuster, "Core principles of bacterial autoinducer systems," Microbiology and Molecular Biology Reviews, vol. 79, no. 1, pp. 153–169, 2015.

[15] R. J. Redfield, "Is quorum sensing a side effect of diffusion sensing?" Trends in microbiology, vol. 10, no. 8, pp. 365–370, 2002.

[16] L. T. Lui, X. Xue, C. Sui, A. Brown, D. I. Pritchard, N. Halliday, K. Winzer, S. M. Howdle, F. Fernandez-Trillo, N. Krasnogor et al., "Bacteria clustering by polymers induces the expression of quorumsensing-controlled phenotypes," Nature chemistry, vol. 5, no. 12, pp. 1058–1065, 2013.

[17] R. Popat, D. Cornforth, L. McNally, and S. P. Brown, "Collective sensing and collective responses in quorum-sensing bacteria," Journal of the Royal Society Interface, vol. 12, no. 103, p. 20140882, 2015.

[18] A. Trovato, F. Seno, M. Zanardo, S. Alberghini, A. Tondello, and A. Squartini, "Quorum vs. diffusion sensing: a quantitative analysis of the relevance of absorbing or reflecting boundaries," FEMS microbiology letters, vol. 352, no. 2, pp. 198–203, 2014.

[19] G. E. Dilanji, J. B. Langebrake, P. De Leenheer, and S. J. Hagen, "Quorum activation at a distance: spatiotemporal patterns of gene regulation from diffusion of an autoinducer signal," Journal of the American Chemical Society, vol. 134, no. 12, pp. 5618–5626, 2012.

[20] M. Weitz, A. Muckl, K. Kapsner, R. Berg, A. Meyer, and F. C. Simmel, "Communication and computation by bacteria compartmentalized within microemulsion droplets," Journal of the American Chemical Society, vol. 136, no. 1, pp. 72–75, 2014.

[21] Isik, I., Arjmandi, H., Corre, C., & Noel, A. Biophysical Model for Signal-Embedded Droplet Soaking into 2D Cell Culture. Accepted in *10th ACM International Conference on Nanoscale Computing and Communication*, Coventry, UK, 2023

[22] Abeyrathna, D., Ashaduzzaman, M., Malshe, M., Kalimuthu, J., Gadhamshetty, V., Chundi, P., & Subramaniam, M. (2022). An AI-based approach for detecting cells and microbial byproducts in low volume scanning electron microscope images of biofilms. Frontiers in Microbiology, 13, 996400.

[23] Hafizur Rahman, M., Azam, M. A., Abir Hossen, M., Ragi, S., & Gadhamshetty, V. (2023). BiofilmScanner: A Computational Intelligence Approach to Obtain Bacterial Cell Morphological Attributes from Biofilm Image. arXiv e-prints, arXiv-2302.

[24] Quan, K., Hou, J., Zhang, Z., Ren, Y., Peterson, B. W., Flemming, H. C., ... & van der Mei, H. C. (2022). Water in bacterial biofilms: pores and channels, storage and transport functions. Critical reviews in microbiology, 48(3), 283-302.