

Software Engineering Department

Capstone Project Book - Phase 1

23-2-D-20

**AgarVision:** ML-powered app to detect, analyze and count agar plate bacteria colonies.

**Authors:**

Simaan Saada (Soft. Eng. Department)

Ibraheem Ganayim (Soft. Eng. Department)

**Supervisors:**

Reuven Cohen (Soft. Eng. Department)

**Co-Supervisors:**

Dr. Lilach Iasur Kruh (Biotechnol. Eng. Department)

# Table of contents

# 

[**1. Introduction 4**](#_5yfuyocyerg9)

[**2. Background and Related Work 6**](#_9yvwr76jsax)

[2.1. Manually counting 6](#_8a9ifxdpvzqa)

[2.2. Automated Colony Counter Devices 9](#_j6wqdpp41ie1)

[2.3. Software solutions including Image processing, AI & ML 10](#_zf6vu5ph4ryd)

[**3. Expected Achievements 14**](#_h9zlwhvskbs8)

[**4. Research / Engineering Process 17**](#_rnh640d5lqpj)

[4.1. The Process 17](#_64hai8wwquf3)

[4.1.1. Initial research/experiments 18](#_x9tqtfqwfvp7)

[4.2. Research Challenges 19](#_a5cc60ab5ot9)

[4.3. Product 21](#_l02l6kifa8vy)

[4.3.1. Use Case 21](#_u33ojgaxkkm3)

[4.3.2. System Architecture 21](#_1e0pxxctpsf)

[4.3.3. Flow chart 24](#_jhmglr1dl0ip)

[4.3.4. App screens 27](#_6go8dt9kbo2l)

[**5. Verification and Evaluation 30**](#_e5imb776ujvi)

[**6. References 39**](#_oxid5pwrcbrt)

**Abstract**

Accurate and efficient analysis of bacterial colonies on agar plates is a crucial task in biotechnology research. Traditional manual counting methods are time-consuming and prone to errors. To address this challenge, this research book presents an ML-powered mobile application that revolutionizes the colony analysis process. By leveraging advanced convolutional neural networks, specifically the YOLO (You Only Look Once) or SSD (Single-Shot Multibox Detector) algorithms, These algorithms excel in real-time object detection tasks by efficiently analyzing images and identifying objects of interest, such as bacterial colonies, with high precision. The application automates the detection and counting of bacterial colonies in agar plates. Biotechnology researchers can simply capture a photo of an agar plate using the app, enabling the intelligent software to count the number of colonies present accurately.

We have successfully implemented a prototype that detects and counts colonies. Thus, we are confident that our app will meet our demands. This streamlined process not only saves time and improves accuracy but also empowers researchers to focus on their core objectives, ultimately enhancing productivity in the biotechnology field.

**Keywords:** Biotechnology scientists, Agar plates, Colony analysis,ِِ Automation, Machine learning, Mobile application, Image processing, Microbiology, Image analysis, Convolutional Neural Networks (CNN), YOLO, Single-Shot Multibox Detection (SSD), Data analysis, Experimental design, Pattern Recognition, Classification algorithms, Regression analysis, and Feature extraction.

# Introduction

In the dynamic field of biotechnology research and development, accurate and efficient analysis of experimental results is of utmost importance. Microbiology experiments often involve the use of agar plates, which provide a solid surface for the growth of microorganisms. The traditional method for analyzing these experiments involves manually counting the bacteria colonies located on the agar plates. However, this approach is prone to human error and can be time-consuming, impeding research progress in biotechnology laboratories.

|  |
| --- |
| Fig 1-1. Example of an Agar plate. |

Despite recent advancements in artificial intelligence, challenges remain in effectively implementing machine learning algorithms for the detection and analysis of bacterial colonies. Therefore, there is a compelling need to develop a solution that overcomes these limitations and provides researchers with a more accurate and efficient means of analyzing agar plates.

Our solution is a machine learning-powered mobile app that utilizes advanced algorithms to detect and count bacterial colonies. By leveraging the capabilities of machine learning, specifically convolutional neural networks (CNNs), we aim to automate the analysis process of agar plates. The implications of our solution are far-reaching, offering significant benefits to researchers in terms of time savings, cost-effectiveness, and enhanced accuracy.

The app will have a convenient environment for bacterial experiments and will include the capabilities to count bacterial colonies, identify types, save results, preview experiments, create environments, take pictures, and store data. It will help in experiments like counting water bacteria, identifying soil bacteria, testing antibiotics, monitoring growth, and comparing experiment results.

Here are more details about the app's features:

* Capture: The app will use the camera to take pictures of samples.
* Count: The app will count the number of bacteria colonies in a sample.
* Detecting bacteria type: The app will detect the type of bacteria colony.
* Save: The app can save the results of experiments.
* Review experiments: The app will enable reviewing of previous experiments' details.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Capture | Detect & Count | Save | Review |

*Fig 1-2. App Features*

*Images for illustration*

# Background and Related Work

The primary purpose of our customer (Wonder Veggies Ltd. [1]) is to make probiotics plants, especially lettuce, sprouted legumes, and sprouts. In order to evaluate the probiotics in the plants, they need to do experiments and count the number of colonies in each sample.  
  
 Wonder Veggies Ltd. [1] perform agar plate tests on a daily basis. In their experiments, they have predefined colors of each bacteria and always-a-color colonies. The process of counting colonies does not need a microscope or UV-integrated devices due to the fact that they can see the colonies after some days of incubation with the naked eye. Usually, they can count them with the naked eye. If not, they call it a “0 colonies count” experiment. Here is a list of techniques available for counting colonies:

## Manually counting

Agar plate is placed on an illuminated pad and marks each colony with a pen. The mark is registered by a digital application and an audible tone confirms and counts the marks.

|  |
| --- |
|  |
| *Fig 2-1. Counting agar plate manually with a colony counter device.* |

This process requires humans to detect and mark the colonies t using a pen. This process is time-consuming and sometimes takes up to 10 minutes or more. As scientists do have a bunch of hundreds of plates, this process will be prone to human error and consume a lot of time.

* + 1. **The counting process issues:**  
       The colony counting method requires counting each dot colony in the plate. Sometimes the petri dish includes more than one experiment. For example, this image shows 4 different tests located in one agar plate, therefore we need to count each experiment individually.

|  |  |
| --- | --- |
|  | *Fig 2-2. Agar plate. Taken in labs.* |

* + 1. **Sizes of colonies may differ:**

We count the number of colonies and their sizes do not change our count. The main difference between different bacteria types is their shape and features. We can see in *Figure 2-4*, two different bacteria, one with a little pink shadow, and another with no shadow. By this feature, we can know that those are different types of bacteria. While in *Figure 2-3*, the colonies have white color, different from *Figure 2-4*’s colonies; this can tell that these colonies are classified as another type of bacteria.

In other words, The size of the colony does not change anything, only the color and its features (shadow, etc…).

|  |  |
| --- | --- |
|  |  |
| *Fig 2-3. Colonies with the same bacteria.* | *Fig 2-4. Two different bacteria colonies.* |

* + 1. **Some colonies cannot be counted:** sometimes, colonies cannot be counted because they are too close to each other (having too much growth) as shown in *Figure 2-5*:

|  |  |
| --- | --- |
|  | *Fig 2-5. These colonies are counted as one “not countable” and not as regular colonies.* |

## Automated Colony Counter Devices

There are many devices out in the market that can count the colonies with high precision, but they are very expensive and cannot be used by multiple users at once. Here is a list of them:

* + 1. **Synbiosis ProtoCOL 3 [2]:**an instrument for colony counting, zone measurements, (inhibition and AST), membranes, and a range of other applications. Cost: ~14,000$.

|  |  |
| --- | --- |
|  | *Fig 2-6. Synbiosis ProtoCOL 3.* |

* + 1. **GelCount by Oxford Optronix [9]:**A dedicated colony counter. All-in-one solution for imaging, counting and characterizing colonies, spheroids and organoids. Cost: ~8,000$.

|  |  |
| --- | --- |
|  | *Fig 2-7. GelCount by Oxford Optronix.* |

## Software solutions including Image processing, AI & ML

* + 1. **Edge Detection [11]:**

Edge detection is a technique of image processing used to identify points in a digital image with discontinuities or sharp changes in the image brightness. The points where the image brightness changes sharply are called the edges (or boundaries) of the internal regions.  
As shown in *Figure 2-7*, an image of the coin and its edges in *Figure 2-8.*

|  |  |
| --- | --- |
|  |  |
| *Fig 2-7. A coin image.* | *Fig 2-8. The edges of the coin.* |

* + 1. **Circle Hough Transform [6,12]:**

Circle Hough Transform (CHT), is an essential feature extraction technique for detecting circles in imperfect images. He can work in a 2-dimensional space, with a basic equation where the area of the circle is described by:

|  |  |
| --- | --- |
|  | *Fig 2-9. Running the CHT algorithm, we can see that in the middle two bold dots are the circle's centers.* |

This approach is good with its precision. The algorithm is considered to be slow compared with ML and AI algorithms for real-time object detection, like TensorFlow, Yolo(v6 or v7), GoogleML Kit and more.

Many object detection techniques leverage the use of CNN (convolutional neural network), R-CNN (Region-based convolutional neural network) or RNN (Recurrent neural networks) in order to detect objects with high performance. Due to the fact that a convolutional neural network (CNN) or recurrent neural network (RNN) is a type of artificial neural network used primarily for image recognition and processing.

* + 1. **TensorFlow (object detection):**

Tensorflow has been a leading open-source library for developing ML-based solutions like object detection and image classification.

While we are focusing on object detection. Tensorflow can be a good environment to train and identify the data.

* + 1. **YOLOv7 [5]:**

You Only Look Once (YOLO)method proposes using an end-to-end neural network application that enables both the predictions of bounding boxes and classification probabilities all at once. YOLOv7 is considered a state-of-the-art real-time object detection algorithm.

|  |
| --- |
|  |
| *Fig 2-10. YOLOv7 Architecture - Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduces the feature space from preceding layers. We pre-train the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.* |

The strategy followed by YOLO is as follows. First, it divides the given image into an S × S grid. Then, each grid cell is used to analyze whether an object falls into it or not. Hence, each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and how accurate this prediction is. The main disadvantage of YOLO is that it is not sensitive to small objects and easy to miss detection.

|  |  |
| --- | --- |
|  | *Fig 2-11. Comparison between Different YOLO versions. We can see that YOLOv7 is the fastest.* |

* + 1. **Faster R-CNN [14]**:

The algorithm gets the input image and extracts the features of the image using a convolutional neural network (CNN). It uses these features as input to the Region Proposal Network (RPN) to generate potential object-bounding box proposals. Then, scaling the regions into fixed sizes using Region of Interest Pooling (RoI). Then, the fixed-size regions are fed into a fully connected network, which consists of a classifier and a bounding box regressor. The classifier predicts the probability of each proposed region containing an object among a predefined set of classes.

Finally, applying non-maximum suppression to eliminate duplicate detections and refine the final set of objects bounding boxes. This process removes overlapping bounding boxes with lower confidence scores. Keeping the most confident and non-overlapping detections.

* + 1. **Single-Shot Multibox Detection (SSD) [12]:**

The SSD algorithm was introduced by Wei Liu in 2016. The authors presented a method that could identify objects by using a feed-forward convolutional network by using a single forward pass. Single shot means that in a single forward pass of the neural network (single run of the algorithm), the identification and classification of an object are possible.

The base of the model consists of a VGG-16 convolutional neural network followed by some additional convolutional layers, which reduce the dimensions of the input at each layer.

The network constructs a group that includes all the default bounding boxes within an image and produces the possibility of the presence of an object inside this box by applying convolutional filters to the feature maps that are created. The convolutional layers are applied at different scales in order for the network to be capable of detecting objects of different scales and sizes.

The architecture results in Non-Maximal Suppression (NMS), which is a pruning technique that discards the bounding boxes with a confidence level less than a certain threshold. SSD is known for its precision, compared to YOLO but with the slowest speed [12].

|  |
| --- |
|  |
| *Fig 2-12. SSD Architecture.* |

# Expected Achievements

The app will use computer vision and machine learning techniques to identify bacteria in agar plates. The tool will be used to identify bacteria quickly and accurately. The app will be available for any smartphone or tablet.

To use the app, you will have the option to open an account. If there are previous experiments, one can view them using the main menu. To start a new experiment, one can take a photo of an agar plate and select the desired areas of the plate. Then he will select the desired bacteria type to identify and the app will count the colonies and generate the results. The results can be modified manually if necessary, and then the experiment can be saved. One can also see past experiments at any time from the main menu, and export any experiment as an Excel file.

This project aims to achieve the following **outcomes**:

* **User-Friendly Mobile App**

The primary objective is to develop a user-friendly mobile application that provides a seamless and intuitive interface for researchers to analyze bacterial colonies on agar plates efficiently. The app will prioritize ease of use and accessibility, ensuring a positive user experience. Additionally, the app will be available for multiple users at the same time, allowing researchers to collaborate on projects and share data easily, having high availability, low latency and minimal downtime at a large scale.

* **Accurate Colony Counting**

The app will employ advanced object detection algorithms, such as YOLO or SSD, that have been pre-trained on comprehensive bacteria colony datasets. By leveraging these algorithms, the app will accurately count the number of bacteria colonies on an agar plate, providing researchers with reliable quantitative data.

* **Effective Colony Classification**

In addition to colony counting, the app will incorporate advanced techniques for the effective classification of bacteria colonies. The algorithm will consider various factors, including colony sizes, density, and similar colors, to enable accurate identification and differentiation of different colony types.

|  |  |
| --- | --- |
|  | Fig 3-1. We can see in this image that some colonies do have a like-a-pink shadow and some do not. Using this feature, the app can differentiate and detect two types of bacteria. |

* **Experiment Management**

The mobile app will facilitate efficient experiment management for researchers. Users will be able to save experiment details, view previous experiments, and access them at any time for reference and analysis. Furthermore, the app will offer the functionality to export experiment data as an Excel file, allowing for seamless integration with other analytical tools.

* **Ignoring External Noise**

An important feature of the app will be the ability to ignore external noise that may disturb colony recognition on agar plates. By implementing sophisticated image processing techniques, the app will intelligently ignore marks, text and non-related colonies, ensuring that the analysis will focus solely on relevant bacteria colonies, improving the accuracy and reliability of the results.

* **Creating Datasets**

Based on our research, we didn’t find any suitable dataset for our problem.   
Due to that, We will have to collect bacterial agar plates and label them with the appropriate bacteria types. Then build a custom dataset with the given bacteria agar plates and train the YOLO based on it.

# Research / Engineering Process

## The Process

In this initial phase of the project, our team will focus on developing the theoretical framework for our software solution. This will involve conducting a thorough literature review to identify existing methods and tools for bacterial colony detection and analysis.

Based on our research, we will develop a conceptual model for our software solution, outlining the main features, algorithms, and methodologies that will be used to detect and analyze bacterial colonies. We will also develop a detailed project plan outlining the key milestones and deliverables for the development process.

To ensure that our software solution is based on sound scientific principles, we will collaborate closely with domain experts in microbiology and bioinformatics. This will involve working with researchers and professionals to gain a deeper understanding of the biology and behavior of bacteria, as well as the experimental methods and data analysis techniques used in microbiology.

As part of the theoretical development process, we will also identify the technical requirements for our software solution, including hardware and software specifications, data storage and processing requirements, and user interface design. We will use this

information to select the appropriate tools and technologies for the development process, taking into account factors such as compatibility, scalability, and ease of use.

Throughout the theoretical development process, we will conduct regular reviews and evaluations of our progress, using feedback from domain experts and stakeholders to refine our conceptual model and project plan. By the end of this phase, we aim to have a well-defined and comprehensive theoretical framework for our software solution, laying the foundation for the next phase of the project.

## Initial research/experiments

In the initial phase of our project, we conducted extensive research and carried out experiments to develop a Minimum Viable Product (MVP) that addresses the problem of counting bacterial colonies on agar plates. Our MVP focused on counting bacterial colonies in an image, even though it has medium-to-high accuracy.

During the research phase, we explored various image-processing techniques and machine-learning algorithms suitable for object detection and counting. We successfully implemented two different types of approaches for our problem: the first approach is based on a hybrid version of CHT [6]. The second approach was a simplified version of our ML-powered mobile application, which utilized convolutional neural networks for colony detection. The software we developed allows users to upload a photo of an agar plate and provides a count of the bacterial colonies present on the plate.

**MVP demonstration:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input** | **Output Image** | **Output Text** |
| **1** |  |  | **110** total colonies |
| **2** |  |  | **10** total colonies |

While the accuracy of our initial MVP was moderate, it served as a promising starting point for further development. Through our experiments, we identified areas for improvement, such as enhancing the training dataset, refining the object detection algorithm, and incorporating more sophisticated techniques like the YOLO or SSD algorithms.

These initial research and experiments laid the foundation for our project, demonstrating the feasibility of automating colony counting using image analysis and machine learning. The results of our MVP provided valuable insights into the challenges and complexities involved in accurate colony detection. Building upon this initial success, we are confident that further research and development will lead to a more advanced and accurate solution for counting bacterial colonies on agar plates.

## Research Challenges

Several challenges need to be addressed to achieve the expected outcomes:

* **Limited Bacteria Colonies Datasets**

Acquiring comprehensive datasets of bacteria colonies poses a significant challenge. To train the algorithm effectively, it is essential to gather high-quality datasets that encompass a wide range of bacteria types, as well as variations in colony sizes and densities. Overcoming this challenge will require collaboration with relevant stakeholders and meticulous data collection efforts.

* **Selection of Optimal Algorithm**

Identifying the most suitable algorithm for accurate colony detection is crucial. The chosen algorithm must account for various factors, including colony sizes, density, and similar colors while maintaining a high level of accuracy. Thorough evaluation and comparison of algorithms, such as YOLO, SSD, and similar approaches, will be conducted to determine the most effective solution.

* **Complex Pre-processing Procedures.**

The pre-processing of agar plate images involves intricate steps that require careful consideration. Tasks such as accurately detecting the edges of the agar plate, precisely segmenting the plate into desired quarters for analysis, and intelligently disregarding text and non-related colonies are non-trivial challenges. Advanced image processing techniques will be employed to tackle these complexities and ensure robust and reliable results, like edge detection algorithm, Circle Hough Transform and more….

By addressing challenges related to limited datasets, algorithm selection, and complex pre-processing procedures, the proposed solution intends to provide researchers with a powerful tool for efficient colony analysis. The expected achievements encompass not only accurate and reliable analysis results but also enhanced experiment management capabilities, empowering researchers in the field of biotechnology.

## Product

### 

### Use Case

|  |
| --- |
|  |
| *Fig 4-1. Use case diagram of the system.* |

### 

### System Architecture

|  |
| --- |
|  |
| Fig 4-2. System Architecture diagram. |

**The 2 main sides of the app:**

* **Client side (Frontend):**

The client side will be developed in a hybrid cross-platform development framework developed by Google called **Flutter**. The main reason for selecting this framework is its ability to be natively compiled and run on both iOS and Android. We can develop one codebase and run it on iOS and Android, with no additional breaking changes in our code.

The second reason for choosing this framework is that it has a very big community to support and fix bugs, (a.k.a published LTS and stable version). So we can develop and have it as our main client-side development framework for a long time (about 4-5 years with no breaking changes).

* **Server side (Backend):**

The server side will be developed with a very popular architecture called “Modular system” which means, it will be divided into 2 main modules, the main reason for dividing the server-side into 2 main module, due to the fact that every module will have access to different data (like microkernels and microservices) so each module will manage the access to the data depends on inner logic:

**RESTful Web API module:**

This module will be responsible for handling all requests from different clients' applications across the internet. It will also handle all access to the database, both from and to the application. This module will be responsible for ensuring that all requests are handled correctly and that all data is properly secured.

Here are some of the specific tasks that this module will be responsible for:

* Parsing and processing incoming requests from clients.
* Authenticating users and granting them access to the appropriate data.
* Updating and retrieving data from the database.
* Generating responses to client requests.

**ML module:**

|  |  |
| --- | --- |
|  |  |
|  |  |
| *Fig 4-3. We can see 3 objects detected using the ML algorithm.* | *Fig 4-4. Example of object detection result.* |

This module will handle all the machine learning logic (do not have access from outside the “Server side” so we can manage the security in the RESTful API module). It will be connected to persistent storage in the cloud like Amazon S3 or Azure Blob Storage to save the ML model (weights). The module will be responsible for loading the model, making predictions, and saving the model's weights.

The module will also be responsible for monitoring the model's performance and updating the model as needed. The module will be designed to be scalable and efficient, and it will be able to handle a large volume of data.

The ML module will be a critical component of the overall system. It will be responsible for providing accurate and reliable predictions, which will be essential for the system's success.

### Flow chart

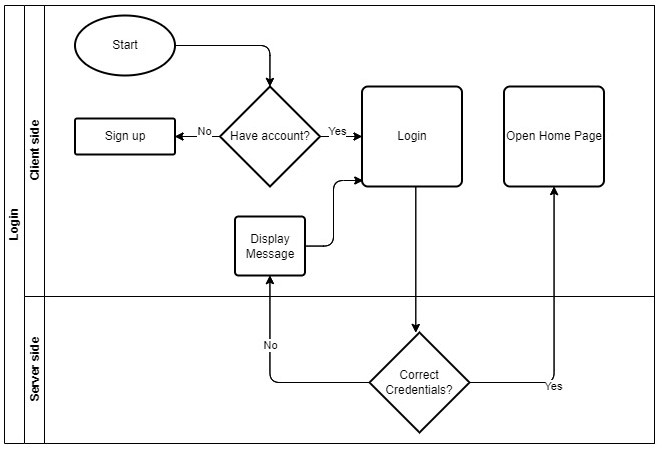
The following Flow Chart provides a step-by-step guide to interacting with the system. The chart shows the two main components of the system: the client side and the server side. The process begins with the user logging in to the system on the client side and ends with the user adding a new experiment and running the prediction algorithm.

Furthermore, you can see the interaction between the client and server side

(from a high-level perspective).

* **Login Flow**

When the app starts, if the user has an account, they log in. If they don’t, they sign up. After entering their login credentials, the app validates them with the server and redirects the user to the home page.

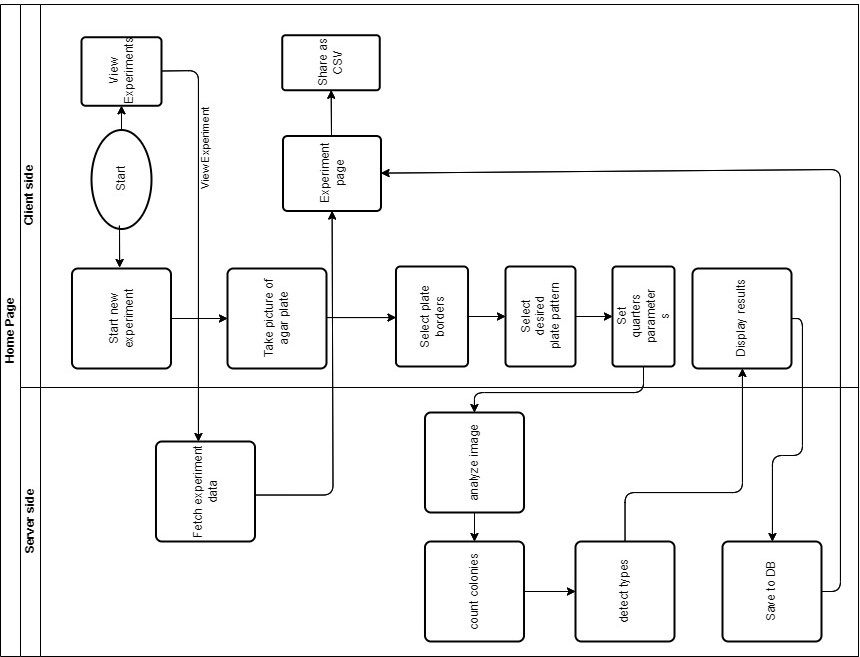
****

* **New Experiment Flow**

To start a new experiment, the user clicks the capture button and is redirected to the capture page. They then take a photo of the agar plate, set the borders of the plate, select the desired quarters, and set the sample parameters. The image is then sent to the server side, where it is processed to determine the number of colonies in the photo and the types of colonies. The results are then sent back to the client side to display the experiment details, and the data is saved to the database by the server.

* **View Experiment Flow**

When a user clicks on an experiment to view its details, a fetch request is sent to the server to retrieve all the experiment data. The data is then displayed to the user as a list on the client side, where the user can choose to share the experiment as a CSV file.

****

### App screens

|  |  |
| --- | --- |
| 1. **Welcome screen (Sign up, sign in or take a quick scan as a guest).** | 1. **View all previous experiments with a brief summary, click one to show more details.** |
|  |  |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

|  |  |
| --- | --- |
| 1. **Take a photo of an agar plate.** | 1. **View details of specific experiments which include (No. of colonies, Types..).** |
|  |  |

|  |
| --- |
| 1. **Set desired parameters for each new experiment such as (Bacteria type, sizes and colors..).** |
|  |

# 5. Verification and Evaluation

To verify the effectiveness of our proposed software solution, we will conduct a series of experiments using both synthetic and real-world data. Our evaluation process will utilize previously conducted experiments as a benchmark for comparing the accuracy and efficiency of our software with traditional colony detection methods. Our goal is to achieve a high level of accuracy in colony detection and analysis, along with a significant reduction in time and human error.

We will use a combination of metrics, including precision, recall, and F1-score, along with a visual inspection of the results, to test our software. We will also conduct experiments to evaluate the performance of our software on a range of agar plates and under various experimental conditions. By doing so, we hope to gain a comprehensive understanding of the capabilities and limitations of our software solution, which will inform future development and improvements.

In addition to quantitative analysis, we will conduct user testing to evaluate the usability and user-friendliness of our software. This will involve collecting feedback from users who perform tasks using the software. Our verification and evaluation process, which includes both quantitative and qualitative analysis, will provide valuable insights into the performance and usability of our software solution, identifying areas for further improvement and development.

To ensure the accuracy and reliability of our software, we will employ specific verification methods, including

* **Synthetic and Real-World Data Experiments**

We will use synthetic data to assess the accuracy of colony detection and real-world data to evaluate the software's performance under diverse conditions.

* **Benchmarking against Manual Counting**

We will compare the results obtained from our software with manually counted colonies on agar plates to validate the accuracy and efficiency of our solution.

* **Metrics Evaluation**

Precision, recall, and F1-score will be calculated to quantify the performance of our software in terms of colony detection accuracy and completeness.

* **Visual Inspection and Validation**

The results obtained from the software will be visually inspected and validated by experts to ensure the reliability and accuracy of the detected colonies.

* **Performance Testing**

Experiments using various bacterial strains and agar plates will be conducted to evaluate the software's performance in detecting colonies of different sizes, densities, and colors.

* **Usability Testing**

User testing sessions will be carried out to assess the usability and user-friendliness of our software. Feedback from participants will help identify areas for improvement and optimize the user experience.

* **Comparative Analysis**

We will compare the performance of our software with existing colony detection algorithms, such as YOLO or SSD, to determine the most effective approach for our solution.

By employing these rigorous verification and evaluation methods, we aim to ensure the accuracy, efficiency, and usability of our software solution. The insights gained from these tests will guide us in refining and enhancing our software, ultimately providing biotechnology researchers with a reliable, user-friendly tool for efficient colony analysis.

**Login Page:**

| **#** | **Test ID** | **Description** | **Excepted Result** | **Comments** |
| --- | --- | --- | --- | --- |
| **1** | successfullLogin | Press on “login”. | Redirect to HomePage “View All Experiments” and save credentials to devices persistent storage. | After inputting username and password. |
| **2** | failedLogin | Press “login” with wrong or non-existing credentials. | Display error message and stay at the login page. | After inputting a wrong or non-existing username or password. |
| **3** | emptyUserName | Press the “Login” button. | Display appropriate messages. | After clicking “Login” when the username field is empty. |
| **4** | emptyPassword | Press the “Login” button. | Display appropriate messages. | After clicking “Login” when the password field is empty. |

**View All Experiments Page:**

| **#** | **Test ID** | **Description** | **Excepted Result** | **Comments** |
| --- | --- | --- | --- | --- |
| **1** | viewExpirements | First page after init the app. | Display all experiments conducted by the logged in user. | First page after successful login too. |
| **2** | newExpirementSuccess | Click on the “New experiment” button. | Display a new experiment page. |  |
| **3** | noCameraPermission | Checked when the “new experiment” page initiated | Display “need permission” prompt. | With two buttons, accept and deny. |
| **4** | grantedCameraPermission | When “new experiment” page initiated | Display the real time camera to the user. | take the agar plate picture in this process |
| **5** | takenPictureSuccess | When the user takes picture and press “Done” | Display “Select Boundaries page” | The user can select the boundaries of the plate. |
| **6** | takenPictureFailed | When user press “Cancel” in camera | Return to page “All Experiments” |  |

**Select Boundaries Page:**

| **#** | **Test ID** | **Description** | **Excepted Result** | **Comments** |
| --- | --- | --- | --- | --- |
| **1** | successDisplayImageInViewer | Checked when page initiated | Display the taken image in the viewer. | Default behavior. |
| **2** | failedDisplayImageInViewer | Checked when page initiated | Display appropriate message. And go back to take new picture page. |  |
| **3** | boundariesNotSelected | Checked when click “finish” button | Display appropriate message. | Stay in same page. |
| **4** | boundariesSelectedSuccess | Checked when click “finish” button | Close this page and continue to “Set Experiment Params Page” |  |

**Set Experiment Parameters Page:**

| **#** | **Test ID** | **Description** | **Excepted Result** | **Comments** |
| --- | --- | --- | --- | --- |
| **1** | emptyBacteriaTypeField | Checked when click “finish” button | Display appropriate message. | Stay in same page. |
| **2** | emptyDilutionField | Checked when click “finish” button | Display appropriate message. | Stay in same page. |
| **3** | emptyColorField | Checked when click “finish” button | Display appropriate message. | Stay in same page. |
| **4** | successSetParams | Checked when click “finish” button | Close this page and go back to the last page. |  |

**Results Experiment Page:**

| **#** | **Test ID** | **Description** | **Excepted Result** | **Comments** |
| --- | --- | --- | --- | --- |
| **1** | Input: | Display the plate image in the top viewer | Display the plate image in the top viewer. | Stay in same page. |
| **2** | countingProccessSuccess | Result from Server-side | Display the quarters list in the bottom panel of the page. | Stay in same page. |
| **3** | waitingForCoutingProccessToFinish | Result from Server-side | Display loading message. | Stay in same page. |
| **4** | countingProccessFailed | Result from Server-side | Display error message, save the params and the image to db. With no results. | The user can re-count when entering the “Experiment Page” |

**Counting Process (Server Side):**

| **#** | **Test ID** | **Description** | **Excepted Result** |
| --- | --- | --- | --- |
| **1** | Input: | Divide the agar plate to 4 quarters | 4 quarters. |
| **2** | Input: | Zero probiotics bacteria. and all others are not recognized. | ~110 colonies. Not recognized bacteria. |
| **3** | Input: | Count the quarter. | 0 colonies. |
| **4** | Input: | Divide the agar plate to 4 quarters. | 4 quarters. |
| **5** | Input: | Count the quarter | 10 colonies. Bacteria B. 1 colony.Bacteria: Not recognized. |

# References

1. Wonder Veggies Ltd. the company we are building the tool to. <http://wonderveggies.co/>
2. [ProtoCOL 3 - Automatic colony counting and zone measuring](https://www.synbiosis.com/product/automated-colony-counting-zone-measurement-protocol-3/)
3. The Importance of Colony Counting. [Colony Counters - Bacteriological Tools | Weber Scientific](https://www.weberscientific.com/colony-counters#:~:text=The%20purpose%20of%20colony%20counting,state%20of%20the%20nutrient%20medium).
4. Colony Forming Units — What Are They and How Many Do You Need?. [Colony Forming Units: What Are CFUs and How Many Do You Need?](https://www.omnibioticlife.com/what-are-colony-forming-units/#:~:text=CFU%20stands%20for%20Colony%20Forming,yeast%20or%20mold%20growing%20together.)
5. YOLOv7 object detection. [GitHub - WongKinYiu/yolov7: Implementation of paper - YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors](https://github.com/WongKinYiu/yolov7)
6. Circle Hough Transform. [Circle Hough Transform - Wikipedia](https://en.wikipedia.org/wiki/Circle_Hough_Transform)
7. Google ML kit. [ML Kit | Google for Developers](https://developers.google.com/ml-kit)
8. Han, S., Lee, S., Lee, J., Lee, S., Lee, H., Lee, H., & Park, S. (2021). Automatic colony detection using machine learning and image processing for high-throughput screening. Scientific Reports, 11(1), 1-10. <https://doi.org/10.1038/s41598-021-83591-1>
9. [GelCount™ Mammalian-cell colony, spheroid and organoid counter](https://www.oxford-optronix.com/gelcount-cell-colony-counter)
10. Yuen, H. K., et al. "Comparative study of Hough transform methods for circle finding." Image and vision computing 8.1 (1990): 71-77.  
     [Comparative study of Hough Transform methods for circle finding - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/026288569090059E)
11. Shrivakshan, G. T., and Chandramouli Chandrasekar. "A comparison of various edge detection techniques used in image processing." International Journal of Computer Science Issues (IJCSI) 9.5 (2012): 269.  
    <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=685e8ecc440343d59f157ca1add377b956b2199d>
12. Tan, L., Huangfu, T., Wu, L. et al. Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. BMC Med Inform Decis Mak 21, 324 (2021).  
    <https://doi.org/10.1186/s12911-021-01691-8>
13. D. Kreuzberger, N. Kühl and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," in IEEE Access, vol. 11, pp. 31866-31879, 2023, doi: 10.1109/ACCESS.2023.3262138.  
    [Machine Learning Operations (MLOps): Overview, Definition, and Architecture | IEEE Journals & Magazine](https://ieeexplore.ieee.org/abstract/document/10081336)
14. Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).