Automated Bacteria Colony Counting using Hybrid Image Segmentation Algorithm and YOLOv5 Transfer Learning Model

Reznee Mariel R. Galope [1], Cherry B. Lisondra [2], Audrey Carmel Jay J. Nanual [3]

[1, 3] Student, [2] Faculty,
College of Computer and Information Science, Mapua Malayan Colleges Mindanao, Philippines.

Presented at International Conference on Innovative Practices in Management, Engineering & Social Sciences (IPMESS-24), Singapore, 3-6 January 2024.

https://doi.org/10.37082/IJIRMPS.IPMESS-24.6

Published in IJIRMPS (E-ISSN: 2349-7300), IPMESS-24

Abstract

Bacteria colony counting is a critical process in microbiology research, but manual colony counting remains tedious and error-prone, motivating the need for automation. This study aimed to develop an automated system for accurately detecting and counting *E. coli* colonies on agar plates. The research objectives were achieved by generating a dataset of *E. coli* colony images, developing a hybrid image segmentation algorithm, and training a YOLOv5 transfer learning model. The dataset was created by capturing images of *E. coli* colonies on agar plates under controlled conditions. The cultured agar plates were generated by Davao Oriental State University (DOrSU). A novel framework for a hybrid image segmentation algorithm combining Watershed and Falling-Ball was developed to address the challenge of accurately segmenting colonies from complex backgrounds. The algorithm utilized the output of the Watershed Algorithm to create a binary mask, which the Falling-Ball Algorithm further refined to improve edge detection and fill gaps. The YOLOv5 transfer learning model was trained using the generated dataset to detect and count *E. coli* colonies. The model achieved a detection accuracy of up to 75%, providing a reliable automated solution for colony counting. Performance evaluation metrics such as precision, recall, and mAP_0.5 were utilized to assess the model's performance. However, training the model using the dataset that underwent the framework could not proceed due to its resource-intensive requirements.

Keywords: Automated Bacteria Colony Counting, Escherichia coli, E. coli, Image Segmentation, Novel Framework, hybrid algorithm, Watershed, Falling-Ball, YOLOv5, Transfer Learning

1. Introduction

It is routine in clinical, culinary, dairy, and pharmaceutical microbiology to get a total microbial count of certain substances. Typical applications of bacteria colony counting include Ames testing, bacterial mutation assays, and Escherichia coli bacterial colonies. Bacteria colonies are traditionally counted with manual methods, which are arduous, time-consuming, and require intense concentration. Colony counters provide an alternative, long-term answer to this problem by permitting colonies' rapid and accurate counting.

A colony counter is a tool to count microorganism colonies developing on agar plates. In the microbiology laboratory, an agar plate is a thin layer of the nutritional gel used to cultivate bacteria and fungus in a Petri dish. Various colony counters allow for the exact and rapid counting of bacteria and yeast colonies. Some colony counters require manual operation, while others are automated (Shrestha, 2022).

The principle of manual counting is straightforward, though tedious. Sometimes, colonies are tiny and crowded, making them difficult to count, especially when the technician only counts the colonies with the naked eye. Counting can be convenient by dividing the plate into several square divisions and magnifying the colonies with a magnifying glass, which makes counting easy. The manual counting process allows a very low throughput and is time-consuming, tedious, and labor-intensive. Also, wide variations are often observed when more than one technician makes counts.

Alternatively, semi-automatic or fully automatic systems have been developed to ensure reliability and minimize operational problems. The semi-automatic methods are based on software alone, whereas automatic methods use hardware and software solutions. An increased area of focus in microbiology is the automation of counting methods. The challenge in automated counting systems includes handling colonies that touch or overlap other colonies and identifying each colony as a unit despite differing shapes, sizes, textures, colors, light intensities, and others (Compendium of Biomedical Instrumentation, n.d.).

Escherichia coli (E. coli) is a Gram-negative bacterium commonly found in diverse environments, including soil, water, and animal intestines. In microbiology research and clinical laboratories, E. coli colonies on agar plates are frequently analyzed to study bacterial growth, antibiotic resistance, and pathogenicity (WebMD, n.d.).

Image segmentation is crucial in automated counting by separating colonies from the background and enhancing their visibility. While various image segmentation techniques exist, the watershed algorithm is commonly used in bacterial colony counting due to its simplicity and effectiveness in touching colonies (Datagen, n.d.).

However, the watershed algorithm encounters challenges of over-segmentation and under-segmentation, resulting in counting inaccuracies. To address these issues, researchers have proposed modifications, such as marker-controlled, seeded, and hierarchical watersheds.

This study focuses on developing a novel framework for a hybrid image segmentation algorithm that combines the Watershed and Falling-Ball algorithms. The hybrid algorithm aims to improve the

accuracy of colony separation and enhance visibility in complex images with overlapping colonies or uneven background illumination. By combining these algorithms, the proposed framework offers a more sophisticated approach to image segmentation for automated colony counting. Moreover, the study utilizes YOLOv5, a convolutional neural network (CNN) model specifically designed for object detection and image localization. By training the YOLOv5 model using a generated dataset, the study aims to accurately detect and count E. coli colonies on agar plates through transfer learning.

The proposed system offers an automated and accurate solution for counting E. coli colonies, alleviating the workload for laboratory technicians and providing reliable and reproducible results. The outcomes of this study contribute to the advancement of automated systems for bacterial colony counting in research laboratories and clinical settings, improving efficiency and facilitating further scientific investigations.

The study aimed to address the following problems in automated bacteria colony counting:

(1.1) Limited availability of diverse and localized datasets for Escherichia coli (E. coli) colony counting

The current availability of E. coli colony counting datasets is limited to established datasets like the AGAR (accessible at https://agar.neurosys.com/). While these existing datasets provide a valuable resource for research, they may not fully capture the diversity and characteristics of E. coli colonies found in specific local environments. This scarcity of localized datasets hamper may hamper the ability to explore and address the challenges specific to local E. coli colonies. Thus, the researchers found it necessary to create a new dataset encompassing samples collected from local environments, enabling them to develop more context-specific and accurate automated counting systems.

(1.2) Limitations of existing image segmentation techniques in accurately identifying E. coli colonies

Current image segmentation techniques struggle to accurately identify E. coli colonies on agar plates due to irregular colony shapes, overlapping colonies, and variations in colony sizes and colors. Traditional segmentation algorithms, such as thresholding, may fail to accurately separate colonies in these complex scenarios, leading to inaccurate counting results. The researchers found it crucial to develop an improved segmentation algorithm that can robustly handle these challenges and accurately identify individual E. coli colonies, enabling precise and reliable counting.

(1.3) Underutilization and limited exploration of YOLOv5 in automated bacteria colony counting

Despite the success of YOLOv5 in various object detection tasks, its application and exploration in the specific domain of automated bacteria colony counting have been limited. The potential benefits and performance of YOLOv5 for accurately detecting and localizing bacteria colonies on agar plates remain largely unexplored. This lack of utilization and exploration hinders the advancement of automated counting systems, limiting the adoption of YOLOv5's capabilities and its potential to improve the accuracy and efficiency of bacteria colony counting processes. The researchers found it crucial to investigate and evaluate the effectiveness of YOLOv5, specifically in the context of automated bacteria colony counting, to unlock its full potential and advance the field. Additionally, YOLOv5 offers efficiency in the study by reducing hardware requirements, such as RAM and GPU usage. Its streamlined architecture minimizes the memory footprint, allowing for more efficient training.

The general objective of this study was to develop an automated and accurate system for counting Escherichia coli (E. coli) colonies on agar plates. To achieve this, three specific research objectives were pursued:

(1.A) Generate images of E. coli colonies on agar plates

In collaboration with Davao Oriental State University (DOrSU), the creation of agar plates with cultured E. coli was undertaken. The university prepared the agar plates, ensuring the presence of viable E. coli colonies under controlled conditions. High-quality photographs of the prepared plates were taken, capturing a diverse range of colonies with variations in size, shape, color, and spatial distribution. Data cleaning techniques were applied to remove anomalies or artifacts from the acquired images to ensure the dataset's accuracy and reliability. Subsequently, the images were manually annotated using Roboflow, a robust annotation tool, to provide precise and detailed labels for the colonies. This comprehensive dataset of E. coli colonies on agar plates formed the foundation for subsequent training and evaluation of the automated counting system.

(1.B) Develop a novel framework for a hybrid image segmentation algorithm combining Watershed and Falling-Ball methods

Addressing the limitations of existing image segmentation techniques, a novel hybrid framework was developed by integrating the Watershed and Falling-Ball algorithms. The Watershed Algorithm, known for its ability to separate touching objects, was employed to produce a visual mask representing the colonies on the agar plates. To enhance the mask's accuracy and clarity, the Watershed Algorithm output was further processed using the Falling-Ball Algorithm, which utilized morphological sculpting tools to smooth out edges and fill gaps. This hybrid approach aimed to improve the reliability and precision of the object segmentation process, particularly in cases where the original visual mask exhibited high levels of noise or artifacts.

(1.C) Train the YOLOv5 transfer learning model using the generated dataset and evaluate its performance

To enable automated counting and localization of E. coli colonies, the YOLOv5 transfer learning model was employed. The YOLOv5 model was trained to recognize and classify E. coli colonies in the images using the generated dataset. Transfer learning techniques were utilized to leverage the pre-trained features of the YOLOv5 model, facilitating efficient training with a limited amount of labeled data. The performance of the trained model was evaluated using established metrics such as precision, recall, and mean Average Precision (mAP_0.5). These metrics provided insights into the model's accuracy, detection capability, and localization precision.

The findings of this study may benefit the following:

Microbiologists and researchers in related fields rely on accurate and efficient counting of bacterial colonies. The proposed system can help reduce the time and effort needed for colony counting, allowing researchers to focus on analyzing the results of their experiments.

Laboratories and clinics that perform microbiological analyses for diagnostic or monitoring purposes. The proposed system can improve the accuracy and speed of colony counting, leading to more reliable and timely results.

Public health agencies and environmental monitoring organizations use bacterial colony counting to detect and monitor the spread of infectious diseases or environmental contaminants. The proposed system can provide a more efficient and reliable way to count colonies, potentially leading to better disease control and environmental management.

Educational institutions teaching microbiology and related fields. The proposed system can provide a practical and hands-on example of the application of image processing and machine learning techniques in microbiology, helping to inspire and train the next generation of researchers and practitioners in the field.

The scope of this study encompasses the development and evaluation of an automated system for counting Escherichia coli (E. coli) colonies on agar plates. The study focused on generating a dataset comprising images of E. coli colonies, developing a hybrid image segmentation algorithm combining Watershed and Falling-Ball methods, and training and evaluating the YOLOv5 transfer learning model for accurate colony detection and counting.

The research revolved around E. coli colonies on agar plates, commonly used in clinical, culinary, dairy, and pharmaceutical microbiology for microbial count determination. The study aims to provide an automated solution that significantly reducing the manual effort and time required for accurate colony counting.

Several delimitations guide the boundaries of this study.

(1.a) The target organism and growth medium are restricted to E. coli colonies on agar plates.
This research does not extend to counting colonies from other bacterial species or different growth media. As a result, the findings and methodologies may not readily apply to different microorganisms or diverse growth conditions.

(1.b) Second, the acquisition of images and the generation of the dataset are integral to this study.
The dataset comprises images from agar plates collaboratively prepared with Davao Oriental State University. Although efforts were made to ensure diversity within the dataset, its representativeness may be subject to limitations, including variations in colony morphology and environmental factors.

(1.c) Developing a hybrid image segmentation algorithm, which combines the Watershed and Falling-Ball methods, constitutes a key aspect of this research.
While the proposed algorithm demonstrates effectiveness, other image segmentation approaches beyond the scope of this study are not extensively explored or compared.

(1.d) The study focuses on training and evaluating the YOLOv5 transfer learning model for colony detection and counting.
While this model is chosen for its efficacy in object detection tasks, alternative transfer learning models or architectures are not thoroughly investigated within this research.

(1.e) The performance evaluation of the developed system primarily employs precision, recall, and mean Average Precision (mAP) metrics.

Although these metrics provide valuable insights, this study does not extensively consider other evaluation measures, such as F1-score, intersection over union (IoU), or accuracy at various IoU thresholds.

(1.f) This research predominantly centers on the technical aspects of automated colony counting.

Application-specific analyses and broader implications of the counted colonies, such as bacterial identification, antibiotic resistance analysis, or specific research contexts, lie beyond the immediate scope of this study.

2. Related Works

This section provides an overview of previous research on the topic at hand. Studies from various disciplines, including image processing, machine learning, and laboratory technology, have investigated how to automate the bacteria colony counting process. The literature review begins by examining why E. coli was chosen as the study's model organism before discussing the various methods proposed by other studies to automate the counting of bacteria colonies.

2.1. E. coli as a model organism

Escherichia coli (E. coli) is a bacterium commonly found in the gut of humans and warm-blooded animals. Most strains of E. coli are harmless. Some strains, however, such as Shiga toxin-producing E. coli (STEC), can cause severe foodborne disease. It is transmitted to humans primarily through consuming contaminated foods, such as raw or undercooked ground meat products, raw milk, and contaminated raw vegetables and sprouts (World Health Organization, 2018).

E. coli has been a key model organism from the very earliest work on molecular genetics and continues to play an essential role today. Much of our understanding of the fundamental concepts of molecular biology, such as replication, gene expression, and protein synthesis, have all been achieved through studies of E. coli (Mullan & Marsh, 2019).

Two reasons for choosing E. coli as an experimental organism were that the bacterium grows rapidly on chemically defined growth media, and the cells do not clump. The genetic analysis depends on populations derived (cloned) from an individual cell, and cloning is simplified when the cells do not clump. Another reason for the choice of E. coli is that it was the host for many widely studied viruses that provided a foundation of molecular biology (Cairns et al., 1966) and provided many tools for genetic and biotechnological manipulations of this bacterium. The ease of biochemical experiments with E. coli also contributed to the popularity of the organism. The investigator could readily grow (or purchase) vast quantities of cells, and the proteins are readily extracted from the cells (Cronan, 2014).

2.2. Automated bacteria colony counting

A fully automated system for counting bacteria colonies would collect images using a digital image-capturing device, such as a document scanner, Charge-Coupled Device (CCD), digital camera/webcam, or video equipment. Colonies can be counted from pictures of plates using software tools. A picture of each plate is taken, and all the pictures are subsequently analyzed. It takes less than 10 seconds to take a single picture, as opposed to several minutes to count colonies manually, thus saving considerable time.

The captured images are then digitized on a computer utilizing an image-processing software package with programming capabilities. The digitized picture is processed using single/multi-threshold segmentation procedures to separate and detect the colonies present (Compendium of Biomedical Instrumentation, n.d.).

2.3. Leveraging image segmentation algorithms

Image segmentation is a method of dividing a digital image into subgroups called image segments, reducing the complexity of the image and enabling further processing or analysis of each image segment. Technically, segmentation is the assignment of labels to pixels to identify objects, people, or other vital elements in the image.

A common use of image segmentation is in object detection. Instead of processing the entire image, a common practice is using an image segmentation algorithm to find objects of interest in the image. Then, the object detector can operate on a bounding box already defined by the segmentation algorithm. This prevents the detector from processing the entire image, improving accuracy and reducing inference time (Image segmentation: The basics and 5 key techniques, 2022).

2.3.1. Watershed and related techniques

One common segmentation technique is watershed segmentation. Watershed segmentation algorithms treat images like topographic maps, with pixel brightness determining elevation (height). This technique detects lines forming ridges and basins, marking the areas between the watershed lines. It divides images into multiple regions based on pixel height, grouping pixels with the same grey value. The watershed technique has several important use cases, including medical image processing. For example, it can help identify differences between lighter and darker regions in an MRI scan, potentially assisting with diagnosis.

Other segmentation techniques include edge-based, threshold-based, region-based, and cluster-based segmentation. Edge-based helps locate features of associated objects in the image using the information from the edges. Threshold-based divides pixels based on their intensity relative to a given value or threshold, making it suitable for segmenting objects with higher intensity than other objects or backgrounds. Region-based involves dividing an image into regions with similar characteristics. Lastly, cluster-based divides images into clusters of pixels with similar characteristics, separating data elements and grouping similar elements into clusters (Image segmentation: The basics and 5 key techniques, 2022).

2.3.2. Falling-ball segmentation technique

Aslam et al. (2021) presented a novel Falling-Ball algorithm, a region-based segmentation algorithm, and an alternative to watershed transform.

The proposed algorithm detects the catchment basins by assuming that a ball falling from hilly terrains will stop in a catchment basin. Once catchment basins are identified, the association of each pixel with one of the catchment basins is obtained using multi-criterion fuzzy logic. Edges are constructed by dividing an image into different catchment basins with the help of a membership function. Finally, a

closed contour algorithm is applied to find closed regions, where objects within these closed regions are segmented using intensity information.

2.3.3. Other image-processing techniques

Automating the identification, segmentation, and counting of stained vitro cell colonies involve challenges with background noise and contaminations. Arous et al. (2022) presented a machine learning procedure to amend these issues by characterizing, extracting, and segmenting inquired cell colonies using principal component analysis, k-means clustering, and a modified watershed segmentation algorithm to identify visible colonies automatically. The proposed segmentation algorithm was tested on two data sets: a T-47D (proprietary) cell colony and a bacteria (open source) data set. High scores and low absolute percentage errors (for T-47D and bacterial images) underlined good agreement with ground truth data.

Jagga & Singh (2018) proposed combining image processing techniques to automate counting bacteria colonies. The proposed method takes an image of bacterial colonies on an agar plate and converts it into a grayscale image. Otsu thresholding is first applied to segment the image and further its conversion into a binary image. Morphological operations are then applied to clean up the image by removing noise and unnecessary pixels. Lastly, distance and watershed transformations are applied to the binary image to create partitions among overlapped and joint bacteria. The segmented image's region properties and labeling information are used to count bacterial colonies.

2.4. YOLOv5 for object detection

The advent of deep learning models has revolutionized the field of object detection, enabling remarkable advancements in automated tasks such as bacteria colony counting. The YOLOv5 (You Only Look Once) architecture is a prominent model that has gained substantial attention. YOLOv5 has demonstrated exceptional performance in real-time object detection, with applications ranging from general object recognition to specific tasks like bacteria colony counting. This section explores the utilization of YOLOv5 models in automated bacteria colony counting and related object detection tasks, examining the strengths, limitations, and significant contributions of this approach in microbiology and beyond.

2.4.1. YOLOv5 for automated bacteria colony counting

In vaccine development, manual colony counting is labor-intensive and error-prone. To address this challenge, Whipp & Dong (2022) focused on developing and evaluating various deep learning models within the YOLO (You Only Look Once) framework for automating microbial colony counting. The evaluation was conducted using S. aureus images obtained from the AGAR dataset, and the developed models demonstrated impressive performance, achieving mAP@0.5 scores ranging from 96% to 99%. The study found that increasing model complexity did not significantly enhance performance. Leveraging the availability of GPUs through Google Colab Pro, the small YOLOv5 model showcased an impressive inference time of approximately 9 milliseconds per image. These findings underscore the potential of YOLO-based deep learning models in facilitating automated, real-time microbial colony counting, offering promising prospects for improving efficiency and accuracy in this crucial aspect of vaccine development.

2.4.2. YOLOv5 in other biological applications

Sun et al. (2022) aimed to develop a high-speed and nondestructive method for detecting mildewed rice grains, using microscopic images and YOLO-v5 models to identify regions contaminated by Aspergillus niger, Penicillium citrinum, and Aspergillus cinerea. The models achieved accuracies of 89.26%, 91.15%, and 90.19% for detecting mildewed regions, and the study established a logarithmic correlation between the proportion of mildewed area and the total number of colonies. This research provides valuable insights for future research on high-speed detection methods for mildewed rice grains based on MCV technology.

Muhammad et al. (2022) aimed to develop an efficient model for detecting rice leaf disease using the YOLOv5 deep learning model. By leveraging the upgraded YOLOv5 version, their model outperformed YOLOv3 and YOLOv4 in terms of performance and accuracy in object detection. The researchers utilized a dataset of 400 rice leaf images obtained from Kaggle containing various diseases. Training, validation, and testing of the model were conducted using the Google Colab platform. The model achieved excellent results with precision, recall, and mAP values of 1.00, 0.94, and 0.62 after training for 100 epochs (Muhammad et al., 2022).

The enumeration of biological entities is time-consuming and prone to inaccuracies when performed manually or with OpenCV-based software. Mehdi et al. (2022) proposed an online platform utilizing multiple trained machine-learning weights to detect yeast colonies, bacterial colonies, and melanoma clusters. Their Pytri model achieved median relative error rates of 7.56% for bacterial and yeast colonies on Petri dishes, 6.58% for colonies on 96-well plates, and 10.28% for melanoma cluster microscopy images. This study demonstrates the application of advanced deep learning tools in bacterial entity detection, offering superior accuracy compared to traditional counting methods.

2.4.3. YOLOv5 in other computer vision tasks

In their study, Mantau et al. (2022) enhanced the YOLOv5 framework for human object detection in UAV perspective images by incorporating a Genetic Algorithm (GA) to optimize the Hyperparameters. Using a dataset combining RGB and Thermal Infrared (TIR) images, their YOLOv5-based transfer learning method achieved higher accuracy than the original YOLOv5 approach. This research aimed to develop a surveillance system using autonomous UAVs to monitor wide areas and address the challenges of limited human resources in tackling illegal activities.

The COVID-19 pandemic that heightened in 2020 has highlighted the importance of wearing face masks in public settings to reduce the transmission of the virus. In their study, Ieamsaard et al. (2021) explored an effective approach for face mask detection utilizing the YOLOv5 deep learning model. To compare their performance, multiple models were developed with varying numbers of epochs, namely 20, 50, 100, 300, and 500. The experimental findings revealed that the deep learning model trained for 300 epochs exhibited the highest accuracy, reaching 96.5%. This research provides valuable insights into developing a robust face mask detection system, contributing to mitigating the spread of COVID-19 through effective monitoring of face mask usage.

In viticulture, accurately estimating grape yields is essential for effective crop management. Sozzi et al. (2022) conducted a study to evaluate different versions of the YOLO (You Only Look Once) object detection algorithm for real-time detection and counting of grape bunches. They tested various YOLO models, including YOLOv3, YOLOv4, and YOLOv5, using a diverse dataset of images captured under

different conditions. The results showed that YOLOv5x and YOLOv4 performed well, achieving F1-scores of 0.76 and 0.77, respectively, with detection speeds of 31 and 32 frames per second. YOLOv4-tiny showed the best balance between accuracy and speed, making it a suitable choice for real-time grape yield estimation. On the other hand, YOLOv3 had some limitations due to compensation for false positives and false negatives, resulting in decreased accuracy.

With the growing use of drones or unmanned aerial vehicles (UAVs), there is an increased need for monitoring and detecting illicit drone activities in restricted areas. Al-Qubaydhi et al. (2022) proposed an automated image-based drone-detection system that utilizes the YOLOv5 deep-learning algorithm. The system achieved excellent precision, recall, and accuracy in detecting drones in surveillance videos. The loss value, indicating the model's effectiveness, was consistently low, indicating accurate predictions. The system successfully identified the location of drones and marked them with bounding boxes, enabling effective monitoring and defense against unauthorized drone incursions.

2.5. Common tools in computer vision tasks

The study's proposed system utilized common tools for computer vision projects, such as Roboflow and PyTorch. Many studies in various applications have used these tools.

2.5.1. Roboflow

Roboflow is an online platform and set of tools designed to simplify and streamline the process of training computer vision models. It provides a comprehensive suite of features and workflows that assist users in annotating, preprocessing, augmenting, and managing image datasets for machine learning tasks. Roboflow offers an intuitive user interface, annotation tools, and automated data preprocessing capabilities, allowing users to prepare their image data efficiently and effectively. Additionally, Roboflow supports popular deep-learning frameworks and provides APIs and export options for seamless integration with custom models and applications. Overall, it aims to facilitate the development and deployment of computer vision models by simplifying and automating key aspects of the data preparation pipeline (Roboflow, n.d.).

Furthermore, RoboFlow is a data-centric cloud-based workflow management system designed for developing AI-enhanced robots. It offers a streamlined approach to robotic development by organizing the process into four building modules: data processing, algorithmic development, backtesting, and application adaptation. The system's containerized and orchestrated design enhances maintainability and enables parallel development. In a study by Lin et al. (2022), RoboFlow was utilized to develop two prototype systems, "Egomobility" and "Egoplan," showcasing its effectiveness in providing navigation functionalities and solving path-planning problems for robotic applications. The results demonstrate the system's ability to streamline the development lifecycle and its potential for various intelligent robotic applications.

Sharma et al. (2022) developed an intelligent vehicle detection system using the You Only Look Once (YOLO) v5 model to identify cars, traffic lights, and pedestrians in various weather conditions for real-time identification in a typical vehicular environment. Object detection in driving can be affected by bad weather conditions, which makes driving dangerous. The proposed system was trained using Roboflow datasets to recognize 11 distinct classes of vehicles, pedestrians, and traffic signals for rainy and regular weather scenarios. Real video sequences of road traffic were also used to evaluate the

system's performance, which showed satisfactory results. This study highlights the need for intelligent traffic monitoring while driving and provides a potential solution to address traffic congestion concerns in large cities with expanding populations.

2.5.2. PyTorch

PyTorch is an open-source machine learning framework that provides a flexible and efficient platform for building and training deep learning models. It is widely used in research and industry for various tasks such as computer vision, natural language processing, and reinforcement learning. It offers dynamic computational graphs, allowing users to define and modify their models on the fly, which makes it particularly suitable for tasks that involve complex or changing architectures. It also provides a rich set of tools and libraries that facilitate data loading, model optimization, and deployment. With its Pythonic programming interface and strong community support, PyTorch has gained popularity as a robust framework for deep learning development (PyTorch, n.d.).

In their paper, Paszke et al. (2019) detailed PyTorch's implementation principles and architecture, emphasizing its compatibility with an imperative and Pythonic programming style. PyTorch allows for code as a model, making debugging easy and ensuring consistency with popular scientific computing libraries. It efficiently utilizes hardware accelerators like GPUs while remaining user-controlled. The authors demonstrate the performance of PyTorch through subsystem analyses and benchmarks on commonly used datasets. This research showcases PyTorch as a powerful tool for deep learning, enabling efficient and user-friendly development.

2.6. CNNs for automated bacteria colony counting

For texture analysis to classify genera and species of bacteria, Zieliński et al. (2017) used deep Convolutional Neural Networks to obtain image descriptors, which are then encoded and classified with Support Vector Machine or Random Forest. To evaluate this approach and to make it comparable with other systems, they provide a new dataset of images called the DIBaS dataset (Digital Image of Bacterial Species), which contains 660 images with 33 different genera and species of bacteria.

Most fully automated techniques were developed using deep learning (DL), which often encountered problems with the need for sizeable collections of annotated plate images. For this reason, Albaradei et al. (2020) proposed an application of transfer learning to cell colony counting by exploiting existing models trained for other tasks. The proponents presented how a small dataset could transform a deep learning model designed for counting objects in congested scenes into a specialized cell colony counting model and achieve better performance than existing, more widely used models.

2.7. Other approaches to automated bacteria colony counting

Different approaches to automating the bacteria colony-counting process include the following:

2.7.1. Near-infrared light

The results of some developed automatic counting methods could save labor and time but are easily affected by uneven illumination and reflection of visible light. Zhu et al. (2018) constructed a convenient and cost-effective system to obtain images of colonies at near-infrared light to offer a method that counts colonies automatically and is robust to light. They then proposed an automated method to detect and measure colonies by processing images. The colonies cultured using raw cows'

milk were used as identification objects. The developed system mainly consisted of a visible/near-infrared camera and a circular near-infrared illuminator.

The proposed method included four steps, i.e., eliminating noises outside the agar plate, removing plate rim and wall, identifying and separating clustered or overlapped colonies, and counting colonies using connected region labeling, distance transform, and watershed algorithms. A graphic user interface was also developed for the proposed method.

The automatic counting method's relative error and counting time were compared with manual counting. The results showed that the relative error of the automatic counting method was −7.4%~ + 8.3%, with an average relative error of 0.2%, and the time used for counting colonies on each agar plate was 11–21 s, which was 15–75% of the time used in manual counting, depending on the numbers of colonies on agar plates. The proposed system and automatic counting method demonstrated promising performance in terms of precision, and they are robust and efficient in terms of labor- and time-savings.

2.7.2. Focusing on hyperspectral features of agar plates

For food quality assessment purposes, Shi et al. (2019) developed a noise-free bacterial colony counting method that identified noise (i.e., sausage, bacon, and millet fragments) that have similar colors or shapes to those of bacteria colonies.

First, spectral features corresponding to colony cluster regions and background regions (agar medium and food fragments) were extracted after collecting hyperspectral images. A cluster-segmenting calibration model was developed to identify colony clusters and background regions. Second, spectral features of colony centers and borders were extracted, and a colony-separating calibration model that could separate single colonies from clusters (multiple colonies contacting each other) was developed. Third, each pixel of an agar plate hyperspectral image was identified using established calibration models, enabling the colonies on the agar plate to be counted successfully ($R^2 = 0.9998$). The results demonstrated that the proposed method could identify the noises caused by food fragments with similar colors or shapes to those of colonies.

2.8. Mobile app-based colony counters

Many colony counter solutions are benchtop-based, which are often bulky and expensive. Austerjost et al. (2017) investigated a cost-effective way to automate the colony counting process with smart devices, using their built-in camera features and a server-based image processing algorithm. The performance of the developed solution is compared to a commercially available smartphone colony counter app and the manual counts of two scientists trained in biological experiments. The comparisons show the high accuracy of the presented system and demonstrate the potential of smart devices to displace well-established laboratory equipment.

Kumar et al. (2017) developed and demonstrated a simple Android-based automated colony counter called “Colonizer” that has been freely available on Google Playstore. Using any Android-based device, the algorithm can threshold and segment overlapping colonies, providing rapid and accurate counts across various colony densities (from 50-500 colonies/plate) for different microbes.

Taithong et al. (2022) presented a new smartphone-based method called Bacillus Cereus Image Counting System (BCICS, pronounced as "B. kiks") for the automatic counting of B. cereus bacteria colonies. BCICS uses Projection Profiling, Circle Hough Transformation, Adaptive Thresholding, and Power-Law Transformation image processing techniques to achieve high image clarity. It then uses the Connected-Component Labeling (CCL) technique to correctly count the colonies, including overlapping ones. These techniques were built into an Android smartphone application.

The results of counting the colonies with BCICS were compared with the results of hand-counting the same dishes. The accuracy rate of each dish count and the average dish accuracy across all dishes were calculated. BCICS counted total colonies with an accuracy of 90.14%, close to hand counting since hand counting commonly involves an error rate of 5 to 10%. The application reportedly took only 3-5 seconds to count one Petri dish, more than 74 times faster than the time required for manual counting.

2.9. IoT-based colony counter

Vongmanee et al. (2018) proposed an automated bacteria colony counter that captures images of cultured bacteria on agar plates using a webcam camera. It uses Raspberry Pi for processing with capturing, thresholding, circular Hough transform, watershed segmentation, and displaying results of the number of colonies on the culture media.

2.10. Synthesis

Pursuing an optimal technique for automating bacteria colony counting is a subject of active research. Existing literature encompasses a range of approaches to developing effective solutions, including exploring modified image segmentation algorithms, integrating multiple layers of image processing techniques, and utilizing deep learning methodologies. In addition, studies have investigated methods for enhancing image quality and resolution in the dataset to improve system performance. Furthermore, research efforts have extended to developing user-friendly interfaces, such as smartphone apps, desktop applications, and IoT devices, to facilitate user interaction with automated colony counting systems.

In line with this ongoing research, the present study aimed to develop an automated bacteria colony counting system by developing a novel framework for a hybrid image segmentation algorithm and harnessing the power of the YOLOv5 transfer learning model. By leveraging the strengths of these techniques and tools, the system can be trained to identify and count E. coli colonies on agar plates accurately. This hybrid approach is expected to enhance the accuracy and efficiency of the colony counting process, offering a more robust and reliable solution than traditional methods. Moreover, to ensure usability and accessibility, the proposed system is seamlessly integrated with a smartphone app-based graphical user interface. This integration enables users to visualize and analyze the total colony-forming unit count conveniently, empowering researchers and practitioners with efficient and intuitive tools for their work. By implementing this advanced methodology, the study aims to significantly contribute to automated bacteria colony counting, paving the way for improved scientific research and practical applications.

3. Materials and Methods

3.1. Research Design

This study employed an *applied research* design based on the intended application of its findings, which is to contribute to the ongoing pursuit of an accurate and reliable way to automate the bacteria

colony counting process. The term "applied research" describes scientific inquiry and investigation that addresses real-world issues, making it crucial for addressing problems that often affect people's lives, livelihoods, health, and general well-being (Formplus Blog, 2020).

This study employed an experimental *research* design based on its objectives. "Exploratory research" is a methodological approach that looks at research issues that have not been thoroughly investigated before (George, 2022). The proponents aimed to explore the possibility of improved performance within the proposed automated bacteria colony counter by incorporating a hybrid image segmentation algorithm (using Watershed and Falling-Ball) into it.

3.2. Research Local

The study's data collection process was conducted at Davao Oriental State University (DOrSU) of Mati City, Davao Oriental, Philippines. The proponents captured images of cultured *E.* coli bacteria colonies on agar plates prepared by DOrSU's Institute of Agriculture and Life Sciences (IALS). The bacteria culturing process was conducted in the microbiology laboratory of the university's Science Building by select faculty members and students at the said institute.

The software development aspect was conducted in Davao City, Davao del Sur. Advising sessions with the proponents' thesis adviser and oral defenses were conducted in Mapúa Malayan Colleges Mindanao.

3.3. Conceptual Framework

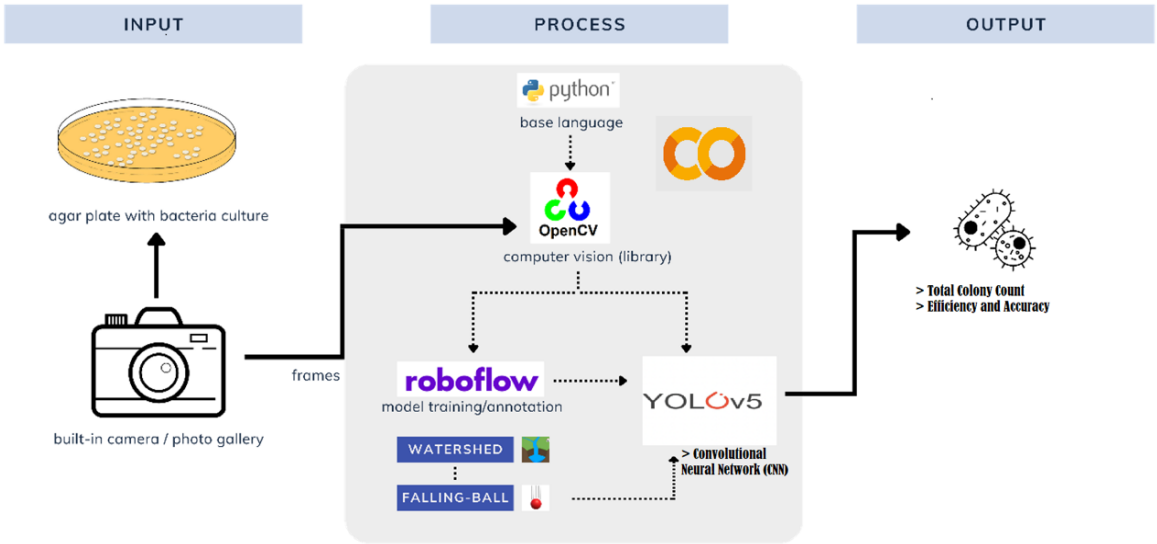

Figure 1: Conceptual Framework

Figure 1 presents an Input-Process-Output (IPO) model, illustrating the inputs and processing tasks necessary for obtaining the study's desired output. The study encompassed two primary inputs. The first was image input, which involved uploading images of cultured *E. coli* on agar plates to the proposed system. This resulted in the system displaying visualized data indicating the total count of *E. coli* colonies on the captured plate. The second input comprised training data, representing an image dataset specifically collected for training the image recognition capabilities of the YOLOv5 model.

The study utilized Roboflow, a comprehensive platform designed for constructing and deploying computer vision applications to identify significant shapes and objects within the images. It provides tools for data preparation, model training, deployment, and integrations with popular deep-learning frameworks and cloud services. With Roboflow, users can annotate images and videos, create custom datasets, train computer vision models, and deploy those models to various platforms. Roboflow supports various computer vision tasks and can be used in multiple industries.

The development of the hybrid image segmentation algorithm involved employing OpenCV (Open Source Computer Vision), a widely adopted open-source library offering developers a range of tools and algorithms for computer vision, image and video processing, and machine learning. OpenCV has applications in numerous fields, such as robotics, surveillance, autonomous vehicles, and augmented reality.

For detecting and counting *E. coli* colonies through transfer learning, the study utilized YOLOv5, a neural network model recognized for object detection and classification capability. YOLOv5 is a continuation of the YOLO (You Only Look Once) series of models, featuring enhanced accuracy and speed compared to its predecessors. It has widespread usage across computer vision applications, including autonomous driving, surveillance, and robotics. Transfer learning, employed in this study, refers to reusing a pre-trained model developed for one task as a starting point for a different but related task. Leveraging the knowledge acquired by the pre-trained model significantly reduced the data and time requirements for training a new model while improving performance.

Lastly, the model's performance in *E. coli* colony detection and counting on agar plates was evaluated using PyTorch, a prominent open-source machine-learning library renowned for constructing and training neural networks. Developed by Facebook's AI research team and based on the Torch library, PyTorch provides a flexible and intuitive platform for implementing deep learning models, supporting dynamic computation graphs, automatic differentiation, and seamless deployment across CPUs, GPUs, and other devices.

3.4. Data Collection

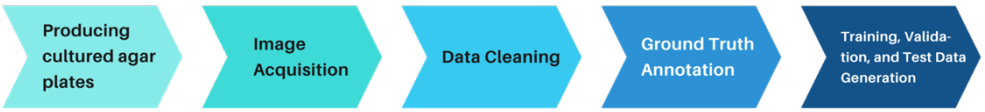

Figure 2. Framework for dataset generation

This section presents the steps involved in acquiring and preparing the dataset for the study. It encompasses sample preparation, where *E. coli* colonies were cultivated on agar plates and the subsequent acquisition of images capturing these plates. The images underwent cleaning, annotation, and splitting into training, validation, and testing datasets.

3.4.1. Producing Agar Plates with Cultured *E. Coli*

Production of agar plates with cultured *E. coli* was headed by the representatives of DOrSU-IALS. They used the CFU (Colony Forming Unit) method, a widely recognized and established technique for quantifying the number of viable microorganisms in a given sample. It is commonly used in microbiology to estimate the concentration or count of bacteria or other microorganisms in a liquid or

solid culture medium. The method involves diluting and plating the sample on an appropriate agar medium, allowing the viable organisms to grow into visible colonies. These colonies are usually counted, and the results are expressed as CFUs per unit volume or weight of the original sample.

It is important to note that the CFU process only counts viable bacterial cells, meaning cells capable of growing and dividing on the agar medium. It does not count non-viable cells, such as those killed by heat, chemicals, or other factors. Additionally, the CFU process assumes that each colony represents a single bacterial cell, which may not always be accurate, particularly for bacteria that grow in clusters or chains. Despite these limitations, the CFU process is widely used and reliable for estimating the number of viable bacteria in a sample.

Preparation of Materials

The sample preparation process required the materials and equipment enumerated in Table 1, which were acquired from various vendors and on-hand stocks in the DOrSU microbiology department.

Table 1: Lab equipment and materials for sample preparation

Material / Equipment	Description
Petri dishes	Petri dishes are shallow, cylindrical, transparent plastic or glass dishes with lids. Also known as *Petri plates* or *culture* dishes, they are used as containers for solid agar medium. They provide a flat surface on which the diluted sample is plated and allow the growth of bacterial colonies to be observed and counted.
EBM agar	EBM agar refers to a specific type of agar medium called *Eosin Methylene Blue agar*. Agar is a gelatinous substance from seaweed as a solidifying agent in microbiological culture media. EBM agar contains specific indicators and nutrients that help distinguish between different types of bacteria, such as *E. coli*, based on their growth and colony characteristics.
Nutrient broth	The nutrient broth is a liquid medium containing essential nutrients for the growth of microorganisms. The CFU method is used as a diluent to disperse and dilute the water sample, making colony counting easier. Additionally, the nutrient broth provides a supportive environment for bacterial growth during incubation, enabling the formation of visible colonies on agar plates for accurate quantification.
Dilution tubes	Dilution tubes are sterile containers, typically glass or plastic, with calibrated volume markings. They are used to perform serial dilutions of the original sample. Each dilution tube in the series contains a progressively lower concentration of bacteria, allowing for the estimation of viable bacterial counts within a countable range.
Pipettes	Pipettes are slender, calibrated tools with tapered tips that transfer precise volumes of the original sample or dilutions from one container to another. They ensure accurate and controlled dispensing of liquids during the dilution process, plating, and other steps in the CFU plating process.
Incubator	An incubator is a controlled environment chamber used to maintain optimal temperature and humidity conditions for the growth of microorganisms. Incubators are essential for promoting the growth of bacteria on the agar plates.
Autoclave	An autoclave is a device used for sterilizing equipment and materials through high-pressure saturated steam. It sterilizes the agar medium, dilution tubes, and other items to eliminate potential contaminants and ensure aseptic conditions during the procedure, preventing unwanted microbial growth that could interfere with accurately determining bacterial colony-forming units (CFUs).

Sample Collection

Sample collection involved obtaining representative water samples from local faucets located within the DOrSU campus that had confirmed *E. coli* presence. Glass sterile bottles that were pre-rinsed with sterile water were used to collect the samples to prevent any potential contaminants. The samples were

immediately transported to the laboratory promptly to prevent any potential changes or alterations in the microbial population due to environmental factors or natural fluctuations.

Figure 3: Water samples from local faucets within DOrSU

Serial Dilution

Serial dilution involved diluting the collected water samples to obtain a range of dilutions suitable for CFU plating.

Labeling the Dilution Tubes

A series of dilution tubes was first prepared, with each tube labeled with a corresponding dilution factor, such as 10^1, 10^2, 10^3, and so on. A dilution factor refers to the extent to which a sample is diluted during serial dilution. It represents the ratio of the volume of the original sample to the volume of the diluent (usually a sterile liquid, such as saline or broth) added to achieve the desired dilution. For example, a dilution factor of 10^2 indicates that the original sample was diluted by a factor of 100 (1 part sample to 99 parts diluent). Dilution factors control the concentration of bacteria in subsequent dilutions, allowing for a range of dilutions that can yield countable colonies on agar plates for accurate quantification.

Transferring the Sample

A sterile pipette transferred 1 mL of the water sample from the collection container to the first dilution tube. The dilution tube contained a diluent, a sterile liquid used to dilute the sample. The DOrSU-IALS representatives used sterile broth as the diluent.

Mixing the Contents

The contents of the dilution tube were then mixed thoroughly to ensure homogeneity. This was done by swirling the tube in a vortex motion or gently inverting it several times. The diluent served as a medium for the bacteria in the sample to disperse, ensuring a more accurate representation of the bacterial concentration. It also provided an environment that supports bacterial growth during the upcoming incubation step.

Serial Dilution Process

To perform the subsequent dilutions, 1 mL was taken from the first dilution tube and transferred to the second dilution tube. The contents were mixed as before. This process was then repeated for each subsequent dilution tube. This created a logarithmic dilution series, with each tube having a higher dilution factor than the previous one. The serial dilution process continued until the desired dilutions

were achieved. The number of dilutions in a series usually depends on the anticipated bacterial concentration and the expected range of colony counts on the agar plates.

Figure 4: Preparing and labeling the dilution tubes

Plating and Incubation

Plating and incubation involved spreading the diluted sample onto solid agar plates and allowing the bacteria to grow into visible colonies.

Agar Preparation

The EBM agar was prepared by dissolving the powder in distilled water, heating the resulting mixture to melt the agar, and then sterilizing it using an autoclave to ensure it was free from contaminants.

Figure 5: Weighing agar powder to be dissolved with distilled water

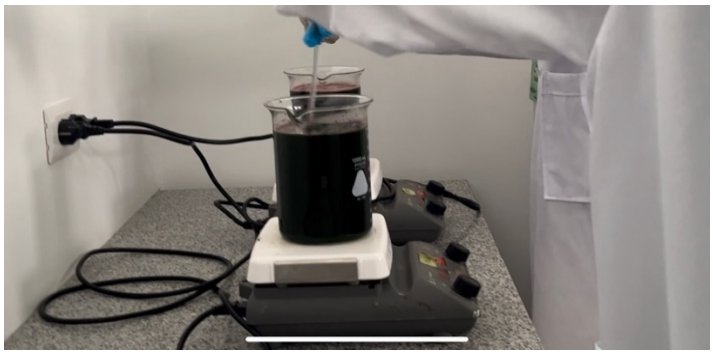

Figure 6: Stirring and heating agar mixture

Figure 7: Sterilizing agar mixture in the autoclave

Pouring Agar Plates

Once the EBM agar was prepared and sterilized, a measured amount of the molten agar was poured into each sterile Petri dish. The plates were left to cool and solidify completely, creating a flat surface for bacterial growth.

Figure 8: Pouring sterilized agar onto Petri dishes

Spreading the Sample

Using a sterile pipette, about 0.1 mL was taken from the desired dilution tube containing the diluted sample and transferred onto the surface of the agar plate. The diluted sample was spread evenly by swirling the plate to allow the bacteria to form isolated colonies.

Figure 9: Transferring 0.1 mL of diluted sample onto an agar plate

Incubation

The agar plates were allowed to sit at room temperature for a few minutes to absorb the liquid into the agar. Once the agar plates had solidified, they were inverted and placed in an incubator set at the

appropriate temperature for E. coli growth, typically around 37°C (98.6°F). The plates were incubated for 24 to 48 hours. E. coli bacteria will grow during incubation and form visible colonies on the agar surface. These colonies arise from a single viable bacterium or a group of closely spaced viable bacteria. Each colony represents the progeny of a single cell or a small group of identical cells.

Figure 10: Incubator housing the agar plates with E. coli bacteria

3.4.2. Image Acquisition

Eight (8) batches of agar plates with cultured *E. coli* were produced and digitally captured between December 2022 and January 2023. The agar plates were photographed using an Olympus Tough TG-6 Digital Camera. The resulting images had a resolution of 2048 × 1536 pixels and were saved in JPEG format, the default format of the used camera.

The plates were captured inside the microbiology laboratory, where natural light from the only glass window and artificial light from the ceiling lights caused glares and shadows to appear in the images. To lessen their presence in the photos to some extent, the plates had to be held up by hand and titled at a certain angle.

The camera was positioned perpendicular to the surface of the agar, which was directly above or in front of the plate, depending on whether it was laid on a flat surface or held up at a certain angle. However, some images from the first batches of samples captured tilted plates as the prospects were still experimenting with the angle and lighting.

3.4.3. Data Cleaning

The proponents and DOrSU-IALS representatives collected over 1,900 photos of cultured agar plates. However, most of the photos had to be removed from the dataset due to blurriness, duplication, or their plates having colonies beyond the countable range. The resulting dataset contained a total of 364 photos.

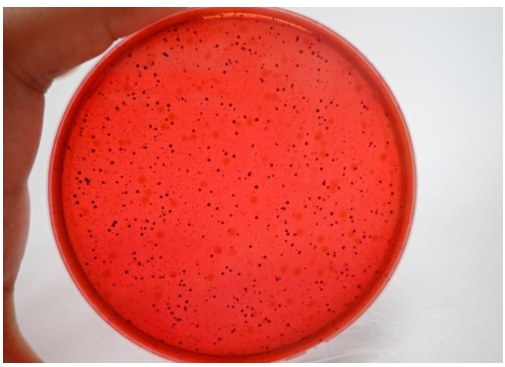

Figure 11: E. coli colonies on an agar plate

3.4.4. Ground Truth Annotation

Ground truth annotation refers to manually labeling or annotating data with accurate or reliable information, serving as the reference or "truth" for subsequent analysis or model training. Computer vision projects involve marking or tagging specific objects, regions, or attributes of interest within images or videos. Crucial for developing and validating computer vision systems, as it provides a basis for comparison and enables the system to learn and make accurate predictions or classifications based on the annotated data.

While the resulting dataset had 364 photos, the proponents only annotated 130 due to the time constraints when conducting the study. The 130 photos of *E. coli* colonies on agar plates were manually annotated using Roboflow, an advanced computer vision platform.

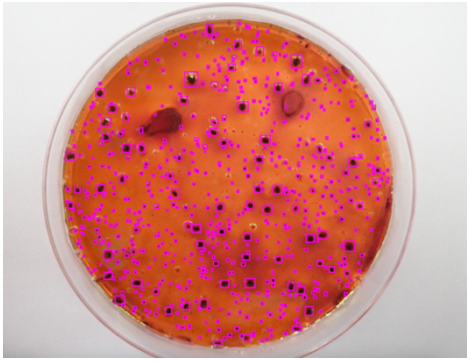

Figure 12: E. coli colonies on agar plate marked with bounding boxes

Each *E. coli* colony was carefully marked with a bounding box for each photo. Each bounding box was labeled with a unique identifier to track each colony across different images. The final dataset consisted of 21,056 annotated *E. coli* colonies.

3.4.5. Training, Validation, and Testing Data Generation

The annotated dataset was then split into three subsets: a training set of 89 images (16,244 annotated colonies), a validation set of 25 images (3,059 annotated colonies), and a testing set of 16 images (1,753 annotated colonies).

The original training dataset contained 89 images with 16,858 annotations of *E. coli* colonies. After adding two augmentations, namely *flipping* and *noise control increase*, the number of images in the training data folder increased to 266 with 49,272 annotations of *E. coli* colonies. *Flipping* refers to transforming an image horizontally or vertically, effectively creating a mirrored version of the original image. *Noise control increase* refers to adding simulated noise or perturbations to the images. This can involve introducing random variations, such as altering the brightness, contrast, or color levels, or adding synthetic noise patterns to mimic real-world variations or imperfections in the data. These augmentation techniques introduce variations in the dataset, enhancing the model's ability to generalize and improve its robustness in handling different orientations and real-world variations in the images.

3.5. Novel Framework for Hybrid Image Segmentation Algorithm

Image segmentation is dividing an image into meaningful and visually coherent regions or segments to facilitate analysis, recognition, and manipulation of specific objects or areas within the image. In the context of this study, it refers to the process of partitioning an image containing bacterial colonies into distinct and individual segments or regions corresponding to each colony. It involves identifying and separating the colonies from the background and neighboring colonies, allowing for precise localization and counting.

The study developed a novel framework for a hybrid image segmentation algorithm that combines the Watershed and Falling-Ball image segmentation algorithms. Leveraging their complementary features, the proposed framework aims to enhance the accuracy and efficiency of image segmentation by effectively delineating objects and boundaries.

The *Watershed* algorithm is a region-based segmentation method that simulates the behavior of water flowing into different catchment basins. It treats the grayscale image as a topographic relief map, where regions correspond to catchment basins. By identifying regional minima and flooding them with water, the algorithm separates different objects or regions based on intensity discontinuities. On the other hand, the *Falling-Ball* algorithm is a boundary-based segmentation method that operates by rolling a virtual ball over the image. The ball starts at different positions and gradually grows, adhering to the local intensity gradients. As the ball grows, it detects boundaries or edges between objects based on the change in pixel intensities, thus delineating distinct regions.

The Watershed algorithm provides excellent region-based segmentation capabilities, while the Falling-Ball algorithm excels in boundary detection. Combining the Watershed and Falling-Ball algorithms in a hybrid framework can harness the strengths of both approaches. The proposed framework combines different image processing techniques to identify and count bacterial colonies in an image accurately.

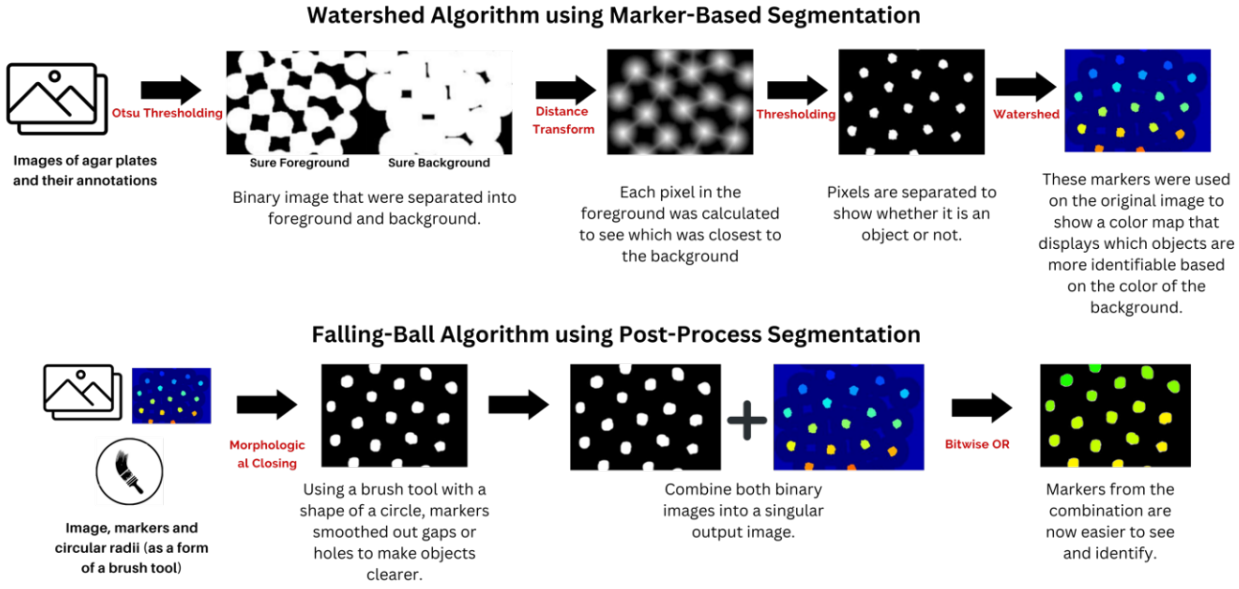

Figure 13: A proposed novel framework for the hybrid image segmentation algorithm

Figure 13 shows the proposed novel framework for the hybrid image segmentation algorithm that combines Watershed and Falling-Ball. The following sections explain the illustrated steps in detail.

3.5.1. Watershed Image Segmentation Algorithm

First, an input image from the dataset and its corresponding set of annotations are processed. This first step involves converting the input image into a simplified binary image. This process uses Otsu's thresholding method, which separates the image into the foreground (bacterial colonies) and background regions.

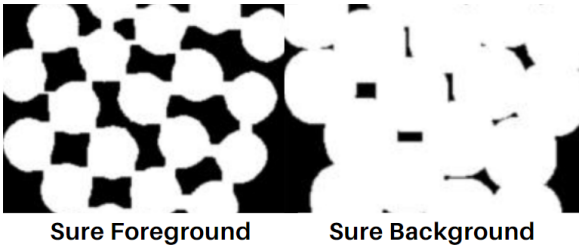

Figure 14: A binary image separated into foreground and background

A mask is created to isolate the bacterial colonies from the background by drawing rectangles around the labeled annotations in the image. This mask acts as a guide to separate the colonies from the background. Then, a technique called morphological opening is applied further to enhance the quality of the colonies' representation. This technique helps remove any unwanted noise present in the colonies' boundaries.

Furthermore, a technique called Distance Transform calculates the distance of each foreground pixel (colony) to the nearest background pixel. This information is then used to create a map that represents the distances. By applying a threshold to this map, the pixels are categorized as belonging to a colony or part of the background.

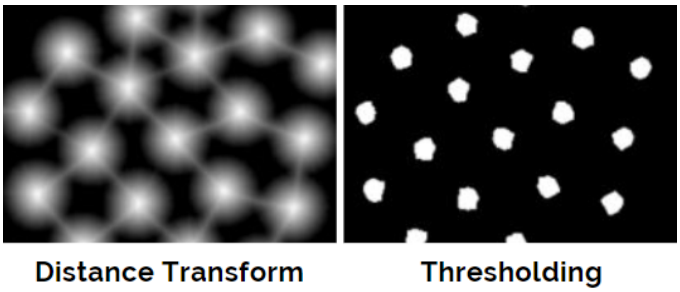

Figure 15: Applying distance transform and thresholding to the image

The Watershed algorithm is then employed on the original image using the markers obtained from the previous step. This algorithm helps segment the image, separating the individual colonies from one another.

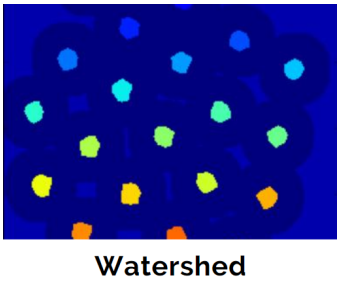

Figure 16: Applying the watershed algorithm to the image

3.5.2. Falling-Ball Image Segmentation Algorithm

With the Falling-Ball component, the input image is converted to grayscale, a simplified image form where shades of gray represent colors. The markers obtained from the Watershed Algorithm are transformed into a binary mask, meaning each pixel is assigned a value of either 1 or 0. Pixels with marker values greater than 1 are set to 1, while the rest are set to 0.

Figure 17: Applying morphological closing to the image

The framework applies a process called morphological closing using different circular structures of various sizes. This process helps fill gaps or holes within the markers using shapes resembling circles. This step ensures that the colonies are well-defined and connected.

The resulting mask obtained from the morphological closing is then used to generate a segmented image. This image isolates areas within the markers not filled during the closing operation, providing a more accurate representation of the colonies.

Finally, a thresholding technique is used to create a binary image where the colonies are represented as distinct objects. A bitwise OR operation combines the information from multiple binary images into a single output image. This operation merges the binary images, resulting in a comprehensive representation of the bacterial colonies in the original image.

3.6. Training YOLOv5 Weights Using Annotated Dataset

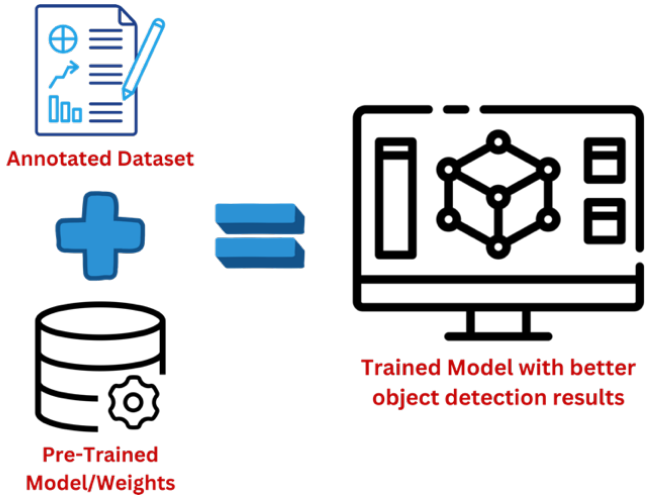

Figure 18: Using the annotated dataset to train pre-trained YOLOv5 weights

In order to train a model specifically for the recognition and counting of E. coli colonies, the proposed framework employed an annotated dataset along with pre-trained weights obtained from YOLOv5. This training process aimed to enhance the model's capabilities and produce improved weights.

It is important to note that the pre-trained weights utilized in this framework are not from an outdated model. Instead, they are derived from a "pre-trained" backbone network based on the architecture of Resnet50 networks. This backbone network has been trained on a large dataset to learn and extract general features from various types of images. By leveraging these pre-trained weights, the model benefits from the prior knowledge and feature extraction capabilities of the Resnet50 architecture, which aids in accurately identifying and classifying *E. coli* colonies.

3.7. Performance Evaluation

The system was evaluated by comparing the results obtained from two different sets of weights. The first set of weights was trained using the original annotated dataset and pre-trained model provided by YOLOv5. The second set of weights was trained using the annotated dataset that underwent the hybrid framework.

The evaluation of the models' effectiveness relied on well-established metrics: precision, recall, and mAP_0.5. Precision measures the proportion of correctly identified E. coli colonies out of all the colonies predicted by the system. Recall quantifies the ratio of correctly detected E. coli colonies against the total number of actual colonies in the dataset. Lastly, mAP_0.5 provides an aggregate evaluation by computing the mean average precision at an intersection over a union (IoU) threshold 0.5.

By comparing the performance and results obtained from the two sets of weights, insights can be gained into the effectiveness of the hybrid framework in improving the accuracy of E. coli colony recognition and counting. This evaluation process allows for determining how much the hybrid framework enhances the system's performance compared to the baseline provided by the YOLOv5 pre-trained weights.

3.8. Limitations of the Methodology

The following section discusses the limitations inherent in the methodology employed for the automated bacteria colony counting system using the hybrid image segmentation algorithm framework and YOLOv5 transfer learning model.

Dependence on Image Quality

The proposed system's accuracy may heavily depend on the input image's quality. If the image is of low resolution, has uneven lighting, or has an inconsistent background, the algorithm's performance may be adversely affected, resulting in inaccurate colony detection and counting.

Limited Flexibility

The proposed system was specifically designed for *E. coli* colonies on EBM agar plates and may not be suitable for other bacteria species, agar variants, or counting methods. Adapting the system to other use cases may require significant modifications to the underlying algorithms and training datasets.

Overlapping Colonies

The system may have difficulty accurately detecting and counting overlapping colonies on the agar plate. This is particularly true when the colonies are too close to each other, as it may be challenging to separate them accurately using the proposed hybrid segmentation algorithm.

Computational Resources

The development of the system, especially in the model training steps, requires significant computational resources, including high-performance computing (HPC) clusters and graphics processing units (GPUs). Limited computational resources may affect the performance and accuracy of the system.

Processing Time. The proposed methodology involves several computationally intensive steps, such as image segmentation and neural network processing, which can take significant time to complete. This can be a limitation in scenarios requiring quick and real-time analysis.

3.9. Ethical Considerations

The following ethical considerations were taken into account during the development of the system:

Protection of Participants

The use of bacterial cultures in this research raises concerns regarding the safety and welfare of laboratory personnel. Safety protocols have been established and followed throughout the project to minimize potential hazards associated with bacterial cultures. These protocols include using personal protective equipment (PPE), properly disposing of materials, and using designated laboratory spaces. In addition, all laboratory personnel have undergone training in properly handling and disposing of bacterial cultures.

Environmental Impact

The development of the system may have environmental impacts, particularly in biowaste generation. The proponents and lab personnel ensured the proper disposal of materials and the implementation of sustainable practices in the laboratory.

4. Results and Analysis

4.1. Generate images of *E. coli* colonies on agar plates

The study's first objective aimed to generate images of E. coli colonies on agar plates. To achieve this, laboratory steps were performed to culture E. coli on agar plates as described in the methodology. A total of approximately 1,900 images were captured from these plates. However, several images were excluded from the dataset during the data curation due to factors such as blurriness, duplication, or colonies beyond the countable range. As a result, the final dataset for analysis comprised a total of 364 images.

Due to time constraints, manual annotation of the images was conducted on a subset of the dataset. Specifically, 130 images were manually annotated to provide ground truth data for subsequent algorithm development and evaluation. Roboflow, an annotation tool, was utilized to ensure consistent and accurate annotations for the annotated images.

The resulting dataset of 130 annotated images was a valuable resource for training and validating algorithms designed for automated *E. coli* colony recognition and counting. These annotated images

encompass diverse colony sizes, shapes, and spatial distributions, capturing the inherent variability encountered in real-world scenarios. The availability of such a dataset will contribute to developing and refining algorithms aiming to automate the colony counting process, facilitating rapid and accurate analysis of bacterial colonies on agar plates.

4.2. Develop a hybrid image segmentation algorithm framework using Watershed and Falling-Ball

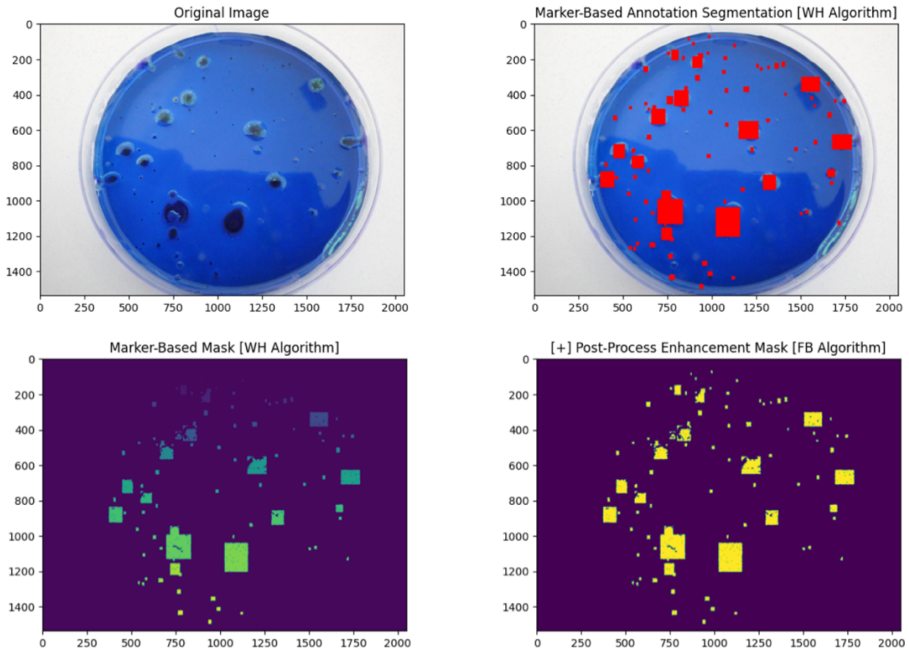

Figure 23: Mask output results

This study's hybrid image segmentation algorithm combined the Watershed Algorithm and the Falling-Ball Algorithm. The Watershed Algorithm generates a visual mask, presented in a gradient format indicating difficulty identifying bounding boxes within the mask. Bounding boxes closer to the background color are more challenging to identify than those farther away.

In the Falling-Ball Algorithm, the output from the Watershed Algorithm is utilized to create a binary mask. This is achieved by applying a morphological sculpting tool, which focuses on the region of gradient bounding boxes. This step aims to smooth out the edges and fill in gaps, thereby enhancing the clarity and ease of identification.

The integration of these two algorithms has the potential to improve the accuracy and reliability of object detection outcomes, particularly when dealing with visual masks that exhibit high levels of noise or artifacts. By refining the initial mask through the Falling-Ball Algorithm, the resulting binary mask facilitates more precise identification of objects of interest.

4.3. Train the YOLOv5 transfer learning model using the generated dataset and evaluate its performance

Table 2 presents the results of training the YOLOv5 transfer learning model using the base annotated dataset, which refers to the dataset that has not undergone the developed hybrid framework. The evaluation metrics utilized in this analysis include precision, recall, and mAP_0.5.

Table 2: Metric results of the model trained using the original annotated dataset

No. of Epoch	Precision	Recall	mAP_0.5
0	0	0	0
9	0.011691	0.0058843	0.0088766
18	0.080299	0.061131	0.027402
27	0.12961	0.18797	0.06654
36	0.14678	0.19451	0.082075
45	0.22962	0.25564	0.15364
54	0.30963	0.31579	0.1945
63	0.32975	0.35796	0.22789
72	0.4021	0.41092	0.30671
81	0.51963	0.43707	0.38139
90	0. 5890	0.495	0.40437
99	0.52803	0.45178	0.38868

The table displays numerical results organized chronologically based on the number of training epochs, with an interval of 10 epochs spanning from 0 to 99.

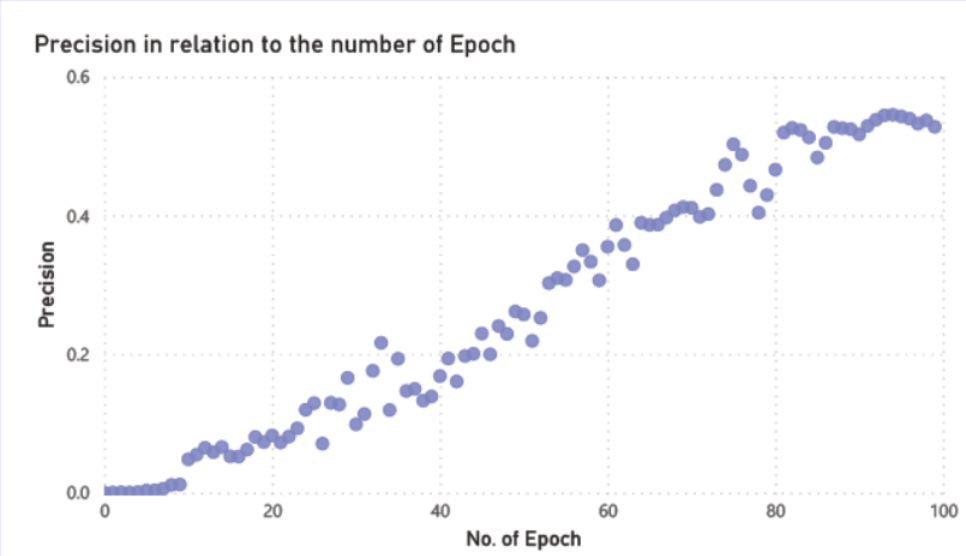

Figure 24: Precision metrics of the model trained using the base annotated dataset

Precision, a metric assessing the accuracy of positive predictions made by the model, yielded a relatively high score of 0.5890, as seen in Figure 23 and enumerated in Table 2. This indicates that the trained model is proficient in making correct predictions based on the provided dataset.

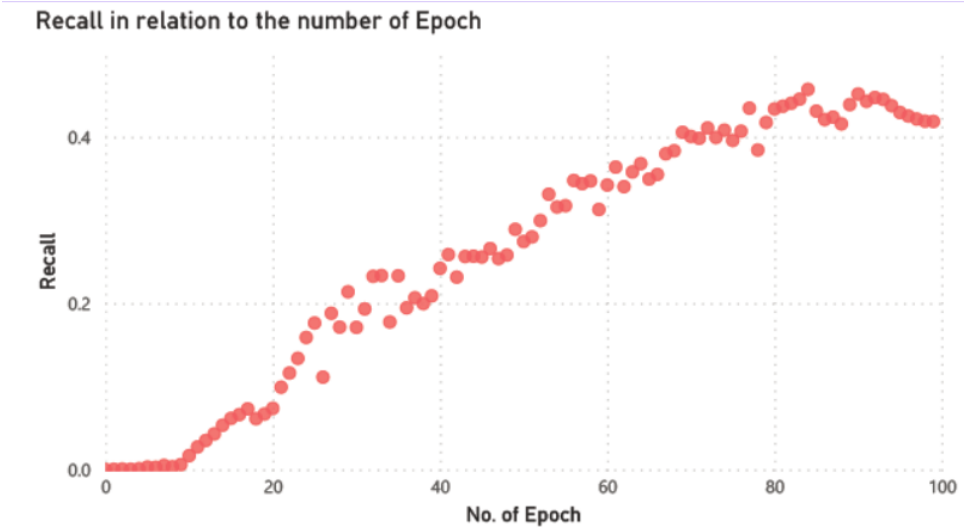

Figure 25: Recall metrics of the model trained using the base annotated dataset

Regarding the recall values depicted in Figure 24 and enumerated in Table 2, the highest value recorded is 0.495. This implies that the model can accurately identify a significant proportion of positive instances present within the dataset.

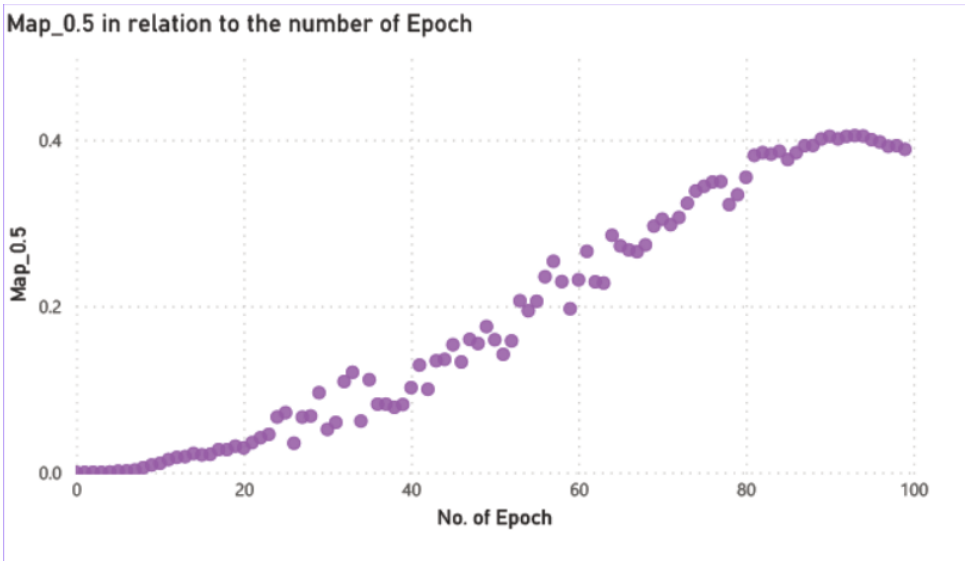

Figure 26: mAP_0.5 metrics of the model trained using the base annotated dataset

Analyzing mAP_0.5, the highest value observed is 0.40437, as depicted in Figure 25 and enumerated in Table 2, signifying that the model successfully detected and localized approximately 40% of objects in the images with a high degree of confidence, as determined by the intersection over union (IoU) threshold of 0.5.

Figure 27: Object detection accuracy results of the trained model used on test images

As depicted in Figure 26, the overall performance of the trained model demonstrated success, achieving the ability to detect multiple objects in an image with a detection accuracy as high as 75% for *E. coli* colonies in the image.

```
0 0.399625 0.443 0.00825 0.0273333 0.100066
0 0.309125 0.223 0.00925 0.0126667 0.100208
0 0.436375 0.2515 0.01375 0.0183333 0.100615
0 0.46225 0.612167 0.014 0.0216667 0.100997
0 0.397875 0.1545 0.00875 0.013 0.10122
0 0.216875 0.286833 0.00675 0.0103333 0.10197
0 0.480375 0.184667 0.01675 0.0266667 0.102015
0 0.571375 0.7535 0.00975 0.0156667 0.102152
0 0.215875 0.289833 0.00625 0.0103333 0.102385
0 0.152125 0.5925 0.00775 0.015 0.102412
0 0.387 0.3305 0.036 0.051 0.10343
0 0.472125 0.184333 0.01325 0.018 0.103433
0 0.358 0.819 0.0225 0.028 0.106881
0 0.6375 0.394333 0.0115 0.0246667 0.107913
0 0.482875 0.186667 0.01025 0.0126667 0.1081
0 0.5685 0.517667 0.021 0.0266667 0.108597
0 0.500875 0.474667 0.00575 0.0273333 0.108774
0 0.554375 0.780167 0.01275 0.0216667 0.112261
0 0.6735 0.7915 0.0135 0.0156667 0.112401
0 0.221625 0.398 0.01825 0.0293333 0.114263
0 0.678 0.428333 0.01 0.012 0.11433
0 0.728625 0.302 0.00825 0.012 0.114597
0 0.481125 0.696833 0.02425 0.0336667 0.115059
0 0.2595 0.428167 0.008 0.0163333 0.11545
0 0.786125 0.6215 0.00975 0.0156667 0.11605
0 0.794125 0.553833 0.01475 0.021 0.117591
0 0.786 0.643833 0.01 0.013 0.118438
0 0.19 0.542667 0.014 0.0206667 0.118727
0 0.500375 0.474333 0.01175 0.016 0.120658
0 0.157375 0.482167 0.00825 0.0136667 0.120933
0 0.377 0.799167 0.0125 0.017 0.121326
0 0.36575 0.6435 0.008 0.017 0.122816
```

Figure 28: Counting results of the trained model used on test images

Additionally, the model not only detects *E.coli* bacteria but also counts them, with each line in the text file shown in Figure 27 representing an *E. coli* colony.

However, when attempting to evaluate the model trained using the annotated dataset that underwent the developed framework, the researchers encountered several challenges that prevented obtaining conclusive results:

- The training requirements were excessively demanding, as training the model with the base annotated dataset alone necessitated a minimum of 19 to 24 GB of training resources. Incorporating the masks generated from Watershed or Falling-Ball would further significantly increase the system's RAM and GPU requirements that the researchers could not access.
- Modifying multiple YOLOv5 Model files to enable acceptance of binary images for training posed considerable challenges. Unfortunately, the lack of documentation or guidance pertaining to such modifications hindered the resolution of this issue.

Given these limitations and technical complexities, the desired evaluation of the model trained using the dataset after applying the hybrid framework could not be successfully conducted.

5. Conclusions and Recommendations

5.1. Conclusions

The present study aimed to achieve three primary research objectives. Firstly, it generated a dataset of images depicting *E. coli* colonies on agar plates. Secondly, a novel framework was developed, integrating the Watershed and Falling-Ball algorithms to create a hybrid image segmentation approach. Finally, the YOLOv5 transfer learning model was trained using the generated dataset, and its performance was evaluated.

The results of the second research objective revealed that the hybrid image segmentation algorithm effectively enhanced the accuracy and reliability of object detection. By utilizing the Watershed Algorithm to produce a visual mask and refining it through the Falling-Ball Algorithm, the framework successfully addressed issues such as noise and artifacts in the original masks. This process led to improved identification and delineation of E. coli colonies, enhancing the overall performance of the detection system.

Moving on to the third research objective, training the YOLOv5 transfer learning model using the generated dataset showcased promising outcomes. The model exhibited a precision score of 0.5890, indicating high accuracy in positive predictions. Additionally, a recall value of 0.495 demonstrated the model's ability to identify many positive instances within the dataset correctly. Furthermore, the model achieved a mAP_0.5 score of 0.405, suggesting successful detection and localization of approximately 40% of objects with high confidence. Consequently, the model proved proficient in detecting and counting E. coli colonies on agar plates.

In conclusion, this study made significant strides in automating bacteria colony counting by developing a comprehensive framework. The integration of the hybrid image segmentation algorithm and the YOLOv5 transfer learning model demonstrated promising results, enhancing the accuracy and reliability of the detection system. By leveraging the power of image analysis techniques and deep learning models, this research advances automated bacteria colony counting methodologies, offering potential

applications in various fields such as medical research, microbiology, and environmental monitoring. However, it is important to acknowledge the limitations encountered in training the model with the dataset that underwent the hybrid framework due to resource-intensive requirements and technical challenges. Future research efforts should focus on addressing these limitations to evaluate the developed framework's potential fully.

5.2. Recommendations for Future Work

Based on the findings and limitations encountered in this study, several recommendations for future work can be proposed to enhance further the automated bacteria colony counting system and improve its applicability. These recommendations aim to address the identified limitations and explore potential avenues for advancements in methodology and technology.

Dataset Expansion and Diversity

The current study generated a dataset of *E. coli* colonies on agar plates; however, future work could benefit from expanding the dataset by including images of colonies from various bacterial species and different growth conditions. Incorporating diverse images would enhance the model's robustness and generalization capabilities, allowing it to accurately identify and count colonies in a wider range of scenarios.

Hybrid Algorithm Refinement

Although the hybrid image segmentation algorithm showed promising results, further refinement, and optimization can be explored. Future studies could investigate alternative algorithms or variations of the Watershed and Falling-Ball methods to improve the accuracy and efficiency of the segmentation process. Additionally, exploring other noise reduction and artifact removal techniques could further enhance the quality of the visual masks generated by the algorithm.

Model Training with the Developed Framework

The resource-intensive requirements and technical challenges encountered during training using the annotated dataset that underwent the hybrid framework highlight the need for further investigation. Future work should optimize the training process by considering methods to mitigate the high RAM and GPU requirements. Additionally, thorough documentation and guidelines on modifying the YOLOv5 model files to accept binary images for training would facilitate the integration of the hybrid framework into the training pipeline.

More Performance Evaluation Metrics

While the current study assessed the performance of the YOLOv5 model using precision, recall, and mAP_0.5 metrics, future research could consider incorporating additional evaluation metrics. Metrics such as F1-score, intersection over union (IoU), and accuracy at various IoU thresholds would provide a more comprehensive evaluation of the model's performance and enable a more nuanced analysis of its strengths and weaknesses.

Integration of Additional Features

Expanding the capabilities of the automated system by integrating additional features could be an interesting avenue for future work. For instance, incorporating the ability to differentiate between

different bacterial species or detecting and analyzing antibiotic resistance patterns within the colonies would provide valuable insights for microbiological research and clinical applications.

Glossary

Petri dish	A shallow glass or plastic cylindrical dish is used to cultivate microorganisms.
Agar	Agar is a gelatinous substance derived from seaweed commonly used as a solidifying agent in microbiology laboratories to provide a solid surface for microorganisms to grow on and can be sterilized by autoclaving without being melted.
Agar plate	An agar plate is a petri dish containing a layer of agar used to culture microorganisms, providing a solid surface on which bacteria, fungi, or other microorganisms can grow and form visible colonies.
E. coli	A type of bacteria that is commonly found in the intestines of humans and animals. Some strains can cause illness.
Image segmentation	The process of partitioning a digital image into multiple segments (sets of pixels), each corresponding to a region of the image with similar properties.
Watershed algorithm	A type of image segmentation algorithm that is based on the idea of flooding a grayscale image from its regional minima.
Falling-ball algorithm	An image segmentation algorithm is based on rolling a ball of a given radius over an image and identifying the regions where the ball fits.
YOLOv5	A family of compound-scaled object detection models trained on the COCO dataset.
YOLOv5 transfer learning model	Refers to the utilization of a pre-trained YOLOv5 model, developed for a different but related task, as a starting point to train a new model specifically for detecting and counting E. coli colonies, leveraging the knowledge learned from the pre-trained model to improve performance and reduce training time and data requirements.
Hybrid algorithm	A combination of two or more algorithms to achieve a better result than using a single algorithm.
Binary mask	A tool used in machine learning to remove some parts of the input or output data during training can help prevent the model from overfitting or memorizing the training data too well. It works by specifying which parts of the data should be removed or "masked out" by setting them to zero and which parts should be kept.

Acknowledgement

First, we would like to express our gratitude to our thesis advisor, Prof. Cherry B. Lisondra, for her guidance and support throughout the research and writing process. Her expertise and dedication to our success were vital to completing our study.

We are also grateful to the College of Computer and Information Science faculty and staff at Mapúa Malayan Colleges Mindanao for providing us with the resources, guidance, and opportunities to conduct our research. Furthermore, we would like to express our gratitude to the administrators, faculty, and staff of Davao Oriental State University for allowing us to document the lab procedures during their microbiology classes and the resulting cultured E. coli bacteria on agar plates for our dataset. We thank Dr. Bernadette Nanual for sponsoring the lacking materials needed for the lab exercises and Ms. Glory Miranda and Ms. Chella Sison for facilitating the lab procedures.

Finally, we thank our classmates and friends who supported and encouraged us throughout our thesis. The countless hours we spent together in coffee shops during all-nighters were some of the most memorable moments of our academic journey. The friendship, humor, and companionship motivated us when the workload seemed insurmountable.

We extend our deepest gratitude and appreciation to all who have participated in our academic journey. Thank you for being there for us and helping us accomplish this feat.

Authors' Biography

Reznee Mariel R. galope and Audrey Carmel Jay J. Nanual are 4th year BS in Computer Science Students.

Cherry B. Lisondra is a faculty member and a Program Coordinator of BSIS Program in the College of Computer and Information Science at the Mapua Malayan Colleges Mindanao. A degree holder of BS in Information Technology, with a Masters in Information Technology, and completed my academic requirements in Doctor in Information Technology.

References

[1] Al-Qubaydhi, N., Alenezi, A., Alanazi, T., Senyor, A., Alanezi, N., Alotaibi, B., Alotaibi, M., Razaque, A., Abdelhamid, A. A., & Alotaibi, A. (2022). Detection of unauthorized unmanned aerial vehicles using Yolov5 and transfer learning. Electronics, 11(17), 2669. https://doi.org/10.3390/electronics11172669

[2] Albaradei, S. A., Napolitano, F., Uludag, M., Thafar, M., Napolitano, S., Essack, M., Bajic, V. B., & Gao, X. (2020). Automated counting of colony forming units using deep transfer learning from a model for congested scenes analysis. IEEE Access, 8, 164340–164346. https://doi.org/10.1109/ACCESS.2020.3021656

[3] Arous, D., Schrunner, S., Hanson, I., Frederike Jeppesen Edin, N., & Malinen, E. (2022). Principal component-based image segmentation: a new approach to outline in vitro cell colonies. Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization. https://doi.org/10.1080/21681163.2022.2035822

[4] Aslam, A., Khan, E., Ansari, M. S., & Beg, M. M. S. (2021). A Novel Falling-Ball Algorithm for Image Segmentation. http://arxiv.org/abs/2105.02615

[5] Austerjost, J., Marquard, D., Raddatz, L., Geier, D., Becker, T., Scheper, T., Lindner, P., & Beutel, S. (2017). A smart device application for the automated determination of E. coli colonies on agar plates. Engineering in Life Sciences, 17(8), 959–966. https://doi.org/10.1002/elsc.201700056

[6] Brush, K., & Silverthorne, V. (2022, November 15). What is Agile Software Development (agile methodologies)? TechTarget. (Retrieved on March 29, 2023) https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

[7] Cairns, J., Stent, G. S., & Watson, J. D. (1966). Phage and the Origins of Molecular Biology. Cold Spring Harbor, NY: Cold Spring Harbor Laboratory of Quantitative Biology.

[8] Cronan, J. E. (2014). Escherichia coli as an Experimental Organism. eLS Wiley. https://doi.org/10.1002/9780470015902.a0002026.pub2
[9] Dey, V. (2021, October 7). Understanding the importance of data cleaning and normalization. Analytics India Magazine. (Retrieved on January 8, 2023) https://analyticsindiamag.com/understanding-the-importance-of-data-cleaning-and-normalization/
[10] Ferrari, A., Lombardi, S., & Signoroni, A. (2017). Bacterial colony counting with Convolutional Neural Networks in Digital Microbiology Imaging. Pattern Recognition, 61, 629–640. https://doi.org/10.1016/j.patcog.2016.07.016
[11] Fleetwood, D. (2022, January 10). Quantitative research: What it is, Tips & Examples. QuestionPro. (Retrieved on January 15, 2023) https://www.questionpro.com/blog/quantitative-research
[12] Formplus Blog. (2020, May 7). What is applied research? + [types, examples & method]. Formplus. (Retrieved on January 21, 2023) https://www.formpl.us/blog/applied-research
[13] Goyal, K. (2021, July 15). Data preprocessing in Machine Learning: 7 easy steps to follow. upGrad blog. (Retrieved on January 16, 2023) https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/
[14] Ieamsaard, J., Charoensook, S. N., & Yammen, S. (2021). Deep learning-based face mask detection using yolov5. 2021 9th International Electrical Engineering Congress (iEECON). https://doi.org/10.1109/ieecon51072.2021.9440346
[15] Image annotation for computer vision: A practical guide. Datagen. (2023, January 4). (Retrieved on January 12, 2023) https://datagen.tech/guides/image-annotation/image-annotation/
[16] Image segmentation: The basics and 5 key techniques. Datagen. (2022, October 25). (Retrieved on January 12, 2023) https://datagen.tech/guides/image-annotation/image-segmentation/
[17] Jagga, B., & Singh, D. (2018). CSEIT183116 | Image Processing Based Bacterial Colony Counter. International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), 1(3).
[18] Kumar, S., Vaswata, R., Choudhury, R., & Laxman, S. (2017). Colonizer: An Android OS based automated microbial colony counter. https://doi.org/10.7287/peerj.preprints.2792v1
[19] Lin, Q., Ye, G., Wang, J. & Liu, H.. (2022). RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots. Proceedings of the 5th Conference on Robot Learning, in Proceedings of Machine Learning Research, 164, 1789-1794. https://proceedings.mlr.press/v164/lin22c.html
[20] Mantau, A. J., Widayat, I. W., Adhitya, Y., Prakosa, S. W., Leu, J.-S., & Koppen, M. (2022). A Ga-based learning strategy applied to Yolov5 for human object detection in UAV surveillance system. 2022 IEEE 17th International Conference on Control & Automation (ICCA). https://doi.org/10.1109/icca54724.2022.9831954
[21] Mehdi, A., Vu, K.-N., & Lambert, F. (2022). Pytri: A Multi-Weight Detection System for Biological Entities. https://doi.org/10.1101/2022.05.11.491558
[22] Muhammad Juman Jhatial, Shaikh, D. R., Noor Ahmed Shaikh, Samina Rajper, Rafaqat Hussain Arain, Ghulam Hussain Chandio, Abdul Qadir Bhangwar, Hidayatullah Shaikh, & Kashif Hussain Shaikh. (2022). Deep learning-based rice leaf diseases detection using yolov5. Sukkur IBA Journal of Computing and Mathematical Sciences, 6(1), 49–61. https://doi.org/10.30537/sjcms.v6i1.1009
[23] Mullan, A., & Marsh, A. (2019, February). Advantages of using escherichia coli as a model organism. Oxford Instruments. (Retrieved on January 13, 2023) https://andor.oxinst.com/learning/view/article/advantages-of-using-escherichia-coli-as-a-model-organism
[24] Overview. Roboflow. (n.d.). https://docs.roboflow.com/
[25] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019, January 1). Pytorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing

Systems. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
[26] Percentage frequency distribution. (Retrieved on January 7, 2023) https://www.jove.com/science-education/12584/percentage-frequency-distribution
[27] Pytorch. https://pytorch.org/
[28] Shah, D. (2023, January 3). The Essential Guide to data augmentation in Deep Learning. V7. (Retrieved on January 11, 2023) https://www.v7labs.com/blog/data-augmentation-guide
[29] Sharma, T., Debaque, B., Duclos, N., Chehri, A., Kinder, B., & Fortier, P. (2022). Deep learning-based object detection and scene perception under bad weather conditions. Electronics, 11(4), 563. https://doi.org/10.3390/electronics11040563
[30] Shi, J., Zhang, F., Wu, S., Guo, Z., Huang, X., Hu, X., Holmes, M., & Zou, X. (2019). Noise-free microbial colony counting method based on hyperspectral features of agar plates. Food Chemistry, 274, 925–932. https://doi.org/10.1016/j.foodchem.2018.09.058
[31] Shrestha, A. (2022, November 4). Colony counters: Types, principles and uses. Microbe Online. (Retrieved on January 14, 2023) https://microbeonline.com/colony-counter/
[32] Sun, K., Zhang, Y.-J., Tong, S.-Y., Tang, M.-D., & Wang, C.-B. (2022). Study on rice grain mildewed region recognition based on microscopic computer vision and Yolo-V5 model. Foods, 11(24), 4031. https://doi.org/10.3390/foods11244031
[33] Sozzi, M., Cantalamessa, S., Cogato, A., Kayad, A., & Marinello, F. (2022). Automatic bunch detection in white grape varieties using yolov3, Yolov4, and Yolov5 deep learning algorithms. Agronomy, 12(2), 319. https://doi.org/10.3390/agronomy12020319
[34] Taithong, P., Wichai, S., Waranusast, R., & Riyamongkol, P. Image-based Automatic Counting of Bacillus cereus Colonies using Smartphone. In International Journal of Advanced Computer Science and Applications (IJACSA), 13(2).
[35] Vongmanee, N., Bunmak, A., & Visitsattapongse, S. (2018). An Automated Colony Counter for Serial-Dilution Culture Method. ACM International Conference Proceeding Series, 42–45. https://doi.org/10.1145/3232059.3232074
[36] WebMD. E. coli bacteria infection: Symptoms, treatment, causes & prevention. WebMD. (Retrieved on March 30, 2023) https://www.webmd.com/food-recipes/food-poisoning/what-is-e-coli
[37] Whipp, J., & Dong, A. (2022). Yolo-based deep learning to automated bacterial colony counting. 2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM). https://doi.org/10.1109/bigmm55396.2022.00028
[38] World Health Organization. (2018, February 7). E. Coli. World Health Organization. (Retrieved on January 12, 2023) https://www.who.int/news-room/fact-sheets/detail/e-coli
[39] Zhu, G., Yan, B., Xing, M., & Tian, C. (2018). Automated counting of bacterial colonies on agar plates based on images captured at near-infrared light. Journal of Microbiological Methods, 153, 66–73. https://doi.org/10.1016/j.mimet.2018.09.004
[40] Zieliński, B., Plichta, A., Misztal, K., Spurek, P., Brzychczy-Włoch, M., & Ochońska, D. (2017). Deep learning approach to bacterial colony classification. PLoS ONE, 12(9). https://doi.org/10.1371/journal.pone.0184554