Check for updates

SOFTWARE TOOL ARTICLE

Lightweight automatized species and media-invariant colony counting application

[version 1; peer review: awaiting peer review]

Thibaut Soubrié[1], Vadym Zhytniuk[2], Tetiana Holdanova[2], Oleh Mezhenskyi[2], Alexander Cunliffe[3], James Redfern[3]

[1]Preste, Paris, France
[2]Preste LLC, Kyiv, Ukraine
[3]Manchester Metropolitan University Department of Natural Sciences, Manchester, England, UK

v1 **First published:** 23 Sep 2026, **6**:372
https://doi.org/10.12688/openreseurope.23934.1
Latest published: 23 Sep 2026, **6**:372
https://doi.org/10.12688/openreseurope.23934.1

Abstract

Colony counting is a routine procedure in the microbiology field, which is labor-intensive, time-consuming, and prone to human-introduced variations, which can hinder assay accuracy. A simple way to overcome this issue is to use an automated computer system capable of performing fast, accurate, and repeatable colony counting. Historically, for this task, the two most common techniques used were simple edge detection, and generic convolutional neural networks, capable of image processing. In this article, we present a simple desktop application based on a lightweight, accurate deep learning model fine-tuned for colony counting under different conditions.

Keywords

Computer vision, bacterial colony, counting, YOLO, microbiology

HE This article is included in the Horizon Europe gateway.

Open Peer Review

Approval Status *AWAITING PEER REVIEW*

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding authors: Thibaut Soubrié (thibaut.soubrie@preste.ai), Tetiana Holdanova (tetiana.holdanova@preste.ai), Oleh Mezhenskyi (oleh.mezhensklyi@preste.ai)

Author roles: Soubrié T: Validation, Writing – Review & Editing; **Zhytniuk V**: Software; **Holdanova T**: Software, Writing – Original Draft Preparation; **Mezhenskyi O**: Writing – Original Draft Preparation; **Cunliffe A**: Data Curation, Resources, Writing – Review & Editing; **Redfern J**: Data Curation, Resources, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work is funded by the European Union under grant agreement number Horizon Europe Framework Programme No. 101058554 (NOVA). The funder played no role in the study design, data collection, analysis, interpretation of data, or the writing of this manuscript
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Soubrié T *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Soubrié T, Zhytniuk V, Holdanova T *et al.* **Lightweight automatized species and media-invariant colony counting application [version 1; peer review: awaiting peer review]** Open Research Europe 2026, **6**:372 https://doi.org/10.12688/openreseurope.23934.1

First published: 23 Sep 2026, **6**:372 https://doi.org/10.12688/openreseurope.23934.1

Introduction

Classical antimicrobial testing remains the industry standard for assessing the efficacy of materials and coatings intended to inhibit microbial growth or reduce viable microbial presence. Despite its widespread adoption, the method still fundamentally relies on manual colony counting performed by lab personnel. This approach is inherently time-consuming, labour-intensive, and prone to human variability, especially when large sample volumes must be processed under tight timelines. There were several studies that documented substantial interoperator variability associated with manual colony counting, even under similar conditions. As global regulatory frameworks increasingly demand robust, high-throughput testing of novel biomaterials, the hardships associated with antimicrobial assays become more visible.

Artificial intelligence–driven analysis tools offer a compelling solution to these constraints. Deep learning based colony detection and segmentation methods have demonstrated substantial improvements in accuracy, reproducibility, and processing speed compared to manual methods, particularly in scenarios involving dense or heterogeneous microbial growth.[1–5] Automated approaches are also well-suited for high-throughput environments, where a large number of plates must be processed consistently and at scale. Such capabilities are particularly relevant for large-scale research initiatives like the NOVA project, which aims to identify, validate, and compare hundreds of potential antimicrobial materials. In projects of this magnitude, even moderate gains in speed or precision yield significant improvements in throughput, resource allocation, and scientific reliability. Consequently, developing reliable AI modules for antimicrobial screening is not merely convenient but strategically important for modern material science research.

In this article, we present a fully developed, ready-to-use application designed to automate antimicrobial plate analysis end-to-end. We describe the logic of the system, from data ingestion to model output, detailing both the neural architectures used for segmentation and detection, as well as the datasets employed for model training and validation. Special emphasis is placed on practical considerations, dataset diversity, annotation quality, pre- and post-processing strategies, and error-handling mechanisms that influence real-world performance but are often underreported.

Finally, we provide a comprehensive overview of the packaged software solution, including its modular structure, deployment options, and integration with cloud-based computational resources. The application is presented as a ready-to-deploy tool capable of supporting both research groups and industrial laboratories that require scalable, reproducible, and automated antimicrobial testing. By sharing the design principles, implementation details, and performance characteristics of this system, we aim to contribute to the growing ecosystem of AI-enabled tools transforming laboratory workflows across microbiology and biomaterials research.

Dataset

The dataset used in this work was generated by NOVA partners EVONIK and MMU. The agar plates were inoculated with *Staphylococcus aureus* ATCC 6538 and *Escherichia coli* ATCC 8739, incubated at 37 °C for 24 hours, resulting in colony morphologies typical for these organisms. The dataset includes two strategies corresponding to the two-step pipeline: agar-plate detection and extraction from raw photographs, and colony detection on the cropped plate images.

The first dataset comprises photographs of agar plates placed on a table, captured as full-frame images that include background objects and variable lighting. This dataset was used to train a model to localise the agar plate and extract the plate region for downstream analysis. It was split into 12 training, 2 validation, and 3 test images.

After plate extraction, the cropped agar plate images were manually annotated for colony detection. Each colony was labeled with a bounding box, and annotations were stored as its coordinates. The colony dataset comprises of 359 training, 66 validation, and 70 test images of agar plates with colonies.

To improve robustness to color variation and enhance generalisation across acquisition conditions, the colony images were converted to grayscale. In addition, image brightness was increased to strengthen the visual separation of colonies from the background.

NN – architecture

YOLOv8s was selected as a baseline model for colony detection on agar plates because it offers a favorable balance between detection accuracy and inference speed.[6] Importantly for our pipeline, YOLOv8's multi-scale design makes it effective for detecting objects with substantially different sizes, enabling the same model family to be applied to both agar plate localisation and bacterial colony detection tasks.

Architecturally, YOLOv8s is a single-stage detector composed of three main components: a backbone for feature extraction, a neck for multi-scale feature fusion, and a detection head for prediction. The backbone extracts features

hierarchically through a sequence of convolutional layers and cross-stage partial bottleneck blocks with two convolutions (C2f modules). At the final stage, spatial pyramid pooling fusion (SPPF) is applied to aggregate multi-scale contextual information by combining max-pooling operations at multiple window sizes, thereby extending the receptive field without significant computational overhead. The neck employs a PAN-FPN structure to merge high-level semantic information with low-level spatial details through bidirectional pathways and feature concatenation at each scale.

The detection head in YOLOv8 adopts an anchor-free approach, eliminating the need for predefined anchor boxes and instead directly regressing bounding box coordinates and class probabilities at each location. The head uses a decoupled design with separate branches for classification and localisation, allowing each task to be optimised independently.

During training, YOLOv8 employs a composite loss function consisting of three components: focal loss for classification to address class imbalance, intersection over union (IoU) loss for precise bounding box localisation, and objectness loss to focus the model on regions likely to contain objects.

This architecture is particularly well-suited for bacterial colony detection, where colonies exhibit considerable variability in size, shape, and texture depending on strain and growth conditions.

Packaged solution

The software solution is implemented as a GUI application (Figure 1) that captures and analyzes a video stream live from a web camera. After launching the application, the user selects the relevant camera and a CSV file containing the names of the samples to be presented to the application. Once the file is loaded, the application displays the expected sample name in the top left and awaits the introduction of the plate containing the sample into the field of view of the camera. While waiting for the plate, the application runs a neural network to detect an agar plate from the camera stream. When the plate is detected, and its position over the last 5 frames does not change, it is considered static, and the colony-counting process begins.

During the counting process, the colonies are counted by another detection neural network, and their number is displayed in the top middle part of the screen. The colonies are marked with blue bounding boxes to allow visual control by the operator. Due to the stochastic nature of neural networks, the detection results may differ from frame to frame. To account for these variations, a sequence of 10 frames is selected from the camera stream, and one of these 10 images is chosen for the final count based on its proximity to the average number of colonies within the sequence. In this way, the result presented to the user is the one the model is most confident in.

After the number of colonies is calculated and displayed, the video stream from the camera is frozen, and the application waits for the user to either correct the number of colonies or proceed to the next sample. In the first case, the user has the option to indicate, via two check boxes on the top right, if the correction comes from the fact that some colonies have been incorrectly paired and should have been split, or, on the contrary, if some colonies have been separated and should have been considered as a single entity. The results of the counting process are recorded in a separate CSV file, created by the application.

The two detection models used in this work are based on the YOLOv8s architecture. The first one detects a plate, the second one detects CFUs on a cropped image of the plate, built using the bounding box of the detected plate in the original image. This application is written in C/C++ using the Raylib and OpenCV libraries.[7,8] Raylibb is a header-only library written in C (C99 standard). Its purpose is to create computer games and graphical applications. This library was chosen because it is lightweight, has no dependencies, and its functionality is split across multiple header files, which reduces the size of the app overall. This library is used to draw a window and read the user's input (mouse clicks, keyboard actions). The OpenCV library was used for neural network inference and reading video camera streams.

Use case

At Manchester Metropolitan University, the post-training packaged solution has contributed to an increased efficiency of antimicrobial surface testing, particularly when testing high volumes of samples. For example, the commonly used ISO22196, a microbiological test used to assess if a surface can kill microorganisms, requires a recovery of the tested surfaces (each control and active surface is in triplicate) into a separate neutralising solution. For each of these neutralising solutions, a serial dilution and inoculating agar plates in triplicate is performed. For each dilution series, the agar plate containing between 30-300 colonies is counted. When testing a high volume of samples in the same test/multiple tests in one day, upwards of hundreds of agar plates are required to be counted, where any increase in efficiency of this task is of benefit and minimises human error.

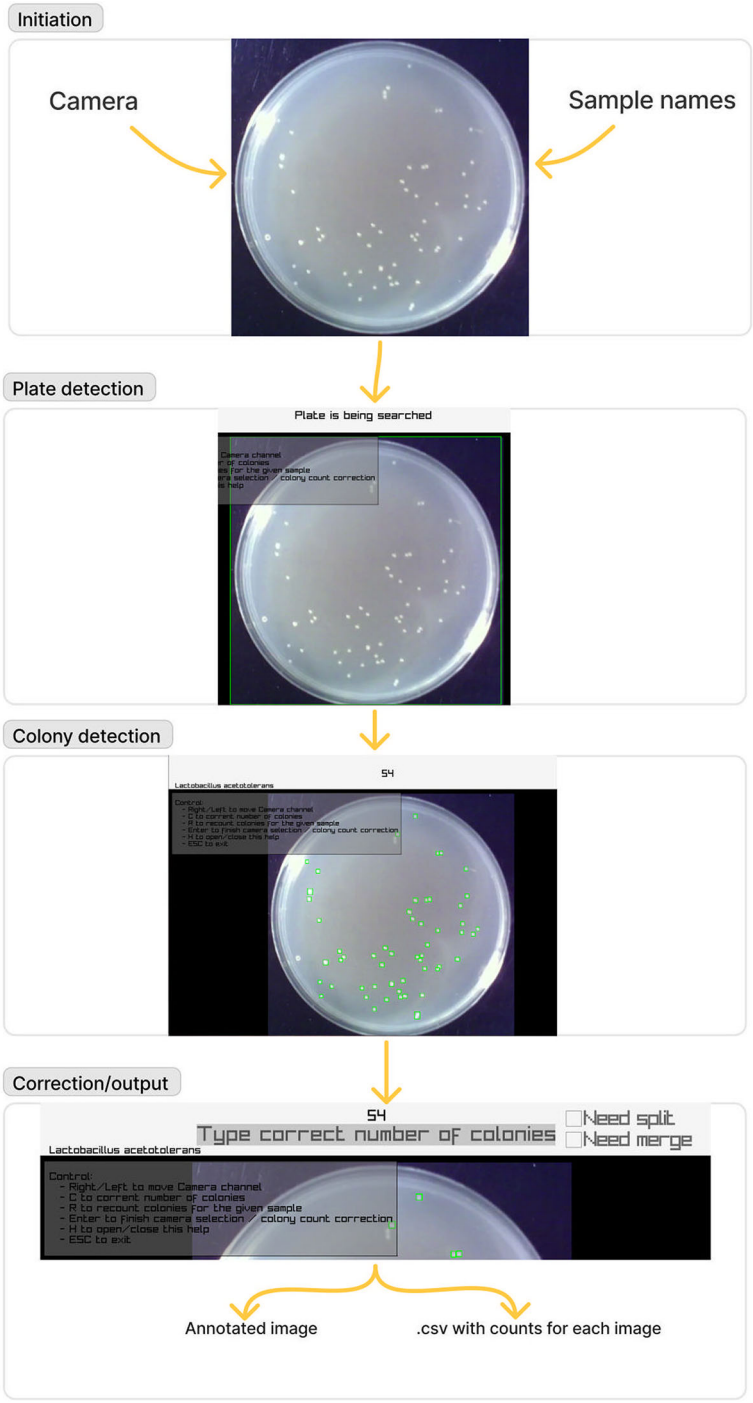

Figure 1. A demonstration of the steps involved in using the packaged solution. 4 steps of app processing. Initiation: user chooses the camera and inputs sample names. Plate detection: The model detects the presence of the agar plate. Colony detection: the model detects and counts colonies. Correction/output: The model provides an interface for count correction, split/merge, and generating a file with output.

When using the counting app, and to ensure accuracy, a dedicated station was created to stabilise the camera, with an opaque cover to minimise environmental light and glare. Within the station, its own dedicated lights were able to adequately visualise the agar plates, and a 3D-printed black-PLA base was used to maximise contrast and visibility of colonies, shown in Figure 2. Once set up, an approximate speed of 10 agar plates per minute could be counted compared to a typical manual counting speed of around 1-2 depending on the number of colonies. The application demonstrated

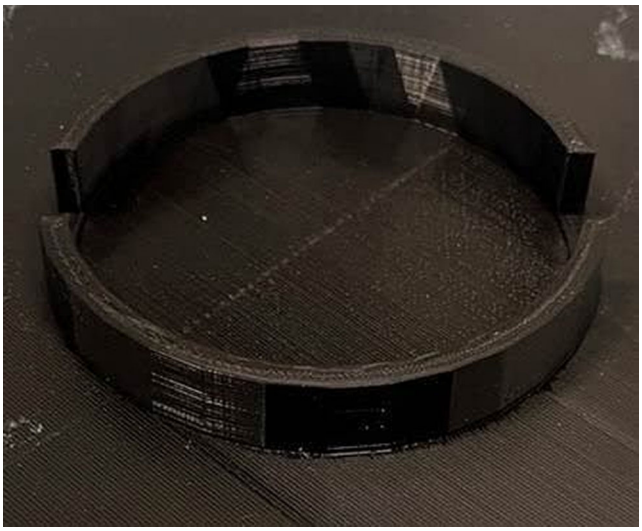

Figure 2. Black 3D-printed base used to maximise contrast with the agar plate. A 3D-printed base produced by IMSL to increase the contrast between the background and the agar plate, improving colony detection accuracy.

several strengths, including increased speed and high accuracy of counted colonies. Furthermore, the app was able to reduce human variation, for example counting merged colonies, which are often counted inconsistently between operators during manual counting.

Whilst the application was sometimes not able to detect all colonies on an agar plate, this was easily corrected manually as undetected colonies were clear to the operator. This was particularly prominent when using selective agars such as mannitol salt that often has a low contrast between colonies and the agar. However, for general use agars such as tryptone soya, nutrient, and malt extract, a sufficient contrast was demonstrated during testing for the application to work efficiently. Difficulties may also present when testing environmental samples with multiple species on the same plate, as colony morphology and colony size could disrupt the application, especially when colonies from different species merge.

Overall, the use of the application can increase efficiency for a testing laboratory that uses a high volume of agar plates, such as antimicrobial surface testing, method development, bioburden and hygiene testing. Also, there is a large benefit to the use of a dedicated station to minimise disturbance to the camera and ensure the highest accuracy possible and reducing human error and variation.

Conclusion

The developed application is a fast and versatile tool suitable for colony counting across multiple species and media combinations. Its performance is primarily dependent on background texture and color, which can be easily controlled under standard laboratory conditions. The publication of this application is expected to contribute to the automation of microbiological testing across both research and industrial settings.

Code availability

Source code available from GitHub: https://github.com/preste-ai/CFUCounter.

Archived software available from: https://doi.org/10.5281/zenodo.22282295 (CFUCounter-main.zip)

License: MIT license.

Ethics and consent

Ethical approval and consent were not required for this study.

Data availability

Repository: Manually annotated plate and colony images available at zenodo under: https://doi.org/10.5281/zenodo.22282295.[9]

Project contains the following underlying data:

- plate_images.zip. (annotated Petri dish plates)
- colony_images.zip (Petri dish plates with annotated colonies)

License: CC BY4 license.

Acknowledgements

The main author of the application, V.Z., is currently participating in the defense against Russian invasion in Ukraine. The authors would like to specifically thank him and all others participating in repelling Russian aggression.

References

1. Fotheringham V: **A Comparison of Manual and Automated Colony Counting.** 2006.
2. Albaradei SA, *et al.*: **Automated Counting of Colony Forming Units Using Deep Transfer Learning From a Model for Congested Scenes Analysis.** *IEEE Access.* 2020; **8**: 164340–164346.
Publisher Full Text
3. Carl SH, *et al.*: **A Fully Automated Deep Learning Pipeline for High-Throughput Colony Segmentation and Classification.** *Biol. Open.* 2020; **9**: article bio052936.
Publisher Full Text
4. Choudhry P: **High-Throughput Method for Automated Colony and Cell Counting by Digital Image Analysis Based on Edge Detection.** *PLoS One.* 2016; **11**: article e0148469.
Publisher Full Text
5. Ferrari A, *et al.*: **Bacterial Colony Counting by Convolutional Neural Networks.** *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* IEEE; 2015; pp. 7458–7461.
Publisher Full Text
6. Yaseen M: **What Is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector.** *arXiv.* 2024; arXiv:2408.15857.
7. Santamaría R: ***raylib.* Version 6.0, GitHub.** 23 Apr. 2026.
Reference Source
8. Bradski G: **The OpenCV Library.** *Dr. Dobb's Journal of Software Tools.* 2000.
9. Soubrié T, *et al.*: **Lightweight Automatized Species- and Media-Invariant Colony Counting Application. Version 1.0.** *Zenodo.* 27 Aug. 2026.
Publisher Full Text