



Real-Time Bacterial Colony Count Detection and Classification Using Computer Vision †

Downloaded from: <https://research.chalmers.se>, 2026-10-27 23:05 UTC

Citation for the original published paper (version of record):

Ramdass, V., Venkatesan, D., Geman, O. et al (2026). Real-Time Bacterial Colony Count Detection and Classification Using Computer Vision †. Engineering Proceedings, 148(1).
<http://dx.doi.org/10.3390/engproc2026148048>

N.B. When citing this work, cite the original published paper.

Real-Time Bacterial Colony Count Detection and Classification Using Computer Vision [†]

Vasugi Ramdass ¹, Dhillip Kumar Venkatesan ¹, Oana Geman ^{2,3} and Roxana Todorean ^{4,*} 

¹ School of Computing, Vel Tech Rangarajan Dr. Sagunthala, R&D Institute of Science and Technology, Chennai 600062, India; vtp4434@veltech.edu.in (V.R.); drvdhillipkumar@veltech.edu.in (D.K.V.)

² Data Science and AI, Computer Science and Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden; geman@chalmers.se

³ Data Science and AI, Computer Science and Engineering, University of Gothenburg, 412 96 Gothenburg, Sweden

⁴ Department of Computers, Electronics and Automation, Stefan cel Mare University of Suceava, 720229 Suceava, Romania

* Correspondence: roxana.todorean@usm.ro

[†] Presented at the International Conference on Electromagnetic Fields, Signals and BioMedical Engineering (ICEMS-BIOMED), Suceava, Romania, 7–9 May 2026.

Abstract

Counting bacterial colonies by hand is one of the most common tasks in microbiology, but it is slow and often yields different results depending on who does the counting. This paper presents an automated system for detecting and classifying bacterial colonies from Petri plate images. We tested four models: Support Vector Machine (SVM), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and a hybrid CNN+Vision Transformer (CNN+ViT). The system uses OpenCV for image preprocessing and extracts features such as area, perimeter, and circularity. Colonies are then classified by size and health condition. Confusion matrix results show that SVM and ANN achieved the highest overall accuracy (95.6%), while CNN+ViT eliminated false positives entirely. A Streamlit-based web interface allows real-time colony analysis directly in laboratory environments.

Keywords: bacterial colony detection; image processing; convolutional neural network; SVM; ANN; deep learning; computer vision; Petri plate analysis

1. Introduction

Bacterial colony counting is a standard procedure in many scientific fields. Microbiologists use it to study how bacteria grow, identify different species, and test whether antibiotics work properly. In clinical settings, accurate colony counts are needed for diagnosis and treatment decisions. Despite how important it is, the traditional approach still relies heavily on trained personnel counting colonies by eye, usually with the help of simple counting devices.

This manual process has several well-known problems. It is time-consuming, especially when the number of colonies is large or when the images have overlapping colonies. Results can also vary between different people, or even when the same person repeats the count at a different time. Poor lighting, crowded plates, and differences in colony colour all make the task harder [1].

The solution proposed in this work is an automated system based on image processing and machine learning. The system takes images of Petri plates as input, processes them to isolate the colonies, extracts measurable features, and then uses trained models to label



Academic Editors: L. Dan Milici,
Mihai Munteanu, Simona Miclăuș,
Paul Bechet and Radu Ciorap

Published: 28 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

each colony by size (small, medium, or large) and health status (healthy or unhealthy). Four classification models were implemented and compared: SVM, ANN, CNN, and a CNN+ViT hybrid. In addition, a real-time web application was built using Streamlit so that laboratory staff can use the system without needing any programming knowledge.

The main objective of this work is to develop a fast, reliable, and easy-to-use system that reduces manual errors in colony analysis and can handle images from real laboratory environments.

2. State of the Art

Several groups have worked on automated colony counting and classification in recent years. Khan et al. [1] built a system using machine learning to count colonies on agar plates, achieving 97% accuracy with Faster R-CNN. Their work showed that object detection models can perform well even when colonies overlap.

Majchrowska et al. [2] introduced the AGAR dataset, which contains 18,000 annotated Petri dish images with more than 330,000 microbial colonies from five species. Using Faster R-CNN and Cascade R-CNN, they demonstrated strong performance in detection and classification. AGAR is the first publicly available large-scale dataset for microbial colony analysis.

Chen et al. [3] proposed MCount, a method built to handle merged colonies more accurately. By combining contour and region-based algorithms, they achieved an average error rate of only 3.99%. The method works especially well in high-throughput settings. MS et al. [4] applied SegNet for the semantic segmentation of colony images, improving model weights with swarm optimisation. Tested on three datasets, the method reached 88.32% accuracy.

Calgin and Kalayci [5] used Teachable Machine®, a web-based deep learning tool, to classify colonies from images. With only 1202 training images, the system reached up to 96.7% accuracy, showing that lightweight tools can still perform well when used carefully. Yang et al. [6] applied Cascade Mask R-CNN with a Swin Transformer backbone for drug sensitivity testing, reaching a mean Average Precision (mAP) of 61.4% after augmenting the dataset with style transfer techniques.

Kong et al. [7] tested eight CNN architectures for microbial colony recognition. GoogLeNet achieved the best result at 98.8%. Amano et al. [8] classified bacterial species from colony morphology alone using CNNs, starting with only 40 images augmented to 5000 samples and reaching 97.19% accuracy.

Rehman et al. [9] compared the Hough Circle Transform against manual counting and found that the automated method reached 95% accuracy while being significantly faster. Whipp and Dong [10] used YOLO-based models for colony counting, achieving mAP@0.5 of 96–99%. Silva et al. [11] compared Faster R-CNN, SSD300, FCOS, and YOLOv11 on over 9000 agar plate images, with YOLOv11 reaching 99.1%.

Wang et al. [12] introduced Colony-YOLO, a lightweight network for micro-colony detection, achieving 96.1% using a StarNet backbone and attention modules. Hallström et al. [13] presented a pipeline for analysing antibiotic synergy tests, showing strong agreement with human grading and potential for integration in smartphone applications.

Pintavirooj et al. [14] combined automated colony detection with multi-class bacterial identification, reducing counting errors significantly. Rehman et al. [9] further validated Hough Transform-based detection across *E. coli*, *S. aureus*, and *P. aeruginosa*. Mennemann et al. [15] addressed the limited training data problem using SAM-based synthetic augmentation. A YOLOv8 model trained on synthetic data performed comparably to models trained on large real-world datasets.

Cao et al. [16] combined U2-Net for segmentation with ResNet50 for counting, reaching an F1-score above 96% for segmentation and 97.82% colony-counting recovery. Makrai et al. [17] created a publicly available labelled dataset of 369 culture images with 56,865 manually marked colonies. Hsiao et al. [18] proposed a patch-based supervised cytometry regression system achieving 95.40% precision for *E. coli* counting, outperforming OpenCFU. Cicatka et al. [19] built a modular system using Attention U-Net (AgarNet) for segmentation and machine learning for clustering, with an F1-score of 0.906.

3. Materials and Methods

3.1. System Architecture

The proposed system processes Petri plate images through five sequential stages: image acquisition, preprocessing, feature extraction, classification, and output.

Stage 1—Image acquisition: Petri plate images are captured using a digital camera or a microscope under consistent lighting to reduce shadows and reflections.

Stage 2—Image preprocessing: Each input image is resized to 224×224 pixels, filtered with Gaussian or median filtering to remove noise, contrast-enhanced using histogram equalisation, and finally segmented using thresholding to isolate colonies from the agar background.

Stage 3—Feature extraction: A Convolutional Neural Network (CNN) extracts spatial and texture features automatically. For each colony region, the following measurements are also computed using OpenCV: area (number of pixels in the colony region), perimeter (length of the colony boundary), circularity (shape roundness measure), texture (derived from CNN filters or GLCM statistics), and mean colour intensity. For an image with n colonies, the feature matrix is $X = [f_1, f_2, \dots, f_n]$, where $f_k = [\text{Area}, \text{Perimeter}, \text{Circularity}, \text{Texture}, \text{Intensity}]$.

Stage 4—Classification: The extracted features are fed into one of four classifiers. SVM uses hyperplanes to separate the feature space and assign colony class labels. ANN performs multi-class classification through input, hidden, and output layers. CNN learns spatial features directly from pixels through convolution and pooling layers. The hybrid CNN+ViT combines CNN local feature extraction with the Vision Transformer attention mechanism for global spatial relationships.

Stage 5—Output: The system produces the total colony count, colony type, health status (healthy or unhealthy), and size category (small, medium, or large).

3.2. Dataset and Implementation

The dataset used in this work consists of digitised images of Petri plates containing bacterial colonies of various sizes, shapes, textures, and health conditions. The images include small, medium, and large colonies, as well as healthy and unhealthy samples.

The system was implemented in Python 3.12. OpenCV was used for image preprocessing and feature extraction. NumPy and SciPy handled numerical computation. Scikit-learn provided the SVM and ANN implementations. TensorFlow and Keras were used to build and train the CNN and CNN+ViT models. Training and evaluation were carried out in Google Colab, which provides GPU access and cloud execution. Dataset Source: <https://www.kaggle.com/datasets/nilay1987/bacterial-colony>, accessed on 25 June 2026.

A Streamlit web application was developed to allow real-time use of the system in a laboratory environment. Users can upload a Petri plate image and immediately receive the colony count, a feature table, size distribution, health status predictions, and performance graphs.

3.3. Processing Algorithm

The full workflow follows these steps: (1) import required libraries; (2) load and preprocess the input image—resize, filter, enhance contrast, and segment; (3) extract colony features: area, perimeter, and circularity; (4) save features to a CSV file; (5) split the dataset into training and testing sets; (6) train SVM, ANN, and CNN models on the training data; (7) use the trained models to predict colony class; (8) compare model performance using accuracy, precision, recall, and F1-score; (9) display results in bar charts, pie charts, and tables; and (10) run automated prediction on new images.

4. Results

4.1. Colony Detection and Distribution

The system successfully detected and isolated colonies from the Petri plate images. Each colony was marked and analysed for morphological features. Figure 1 shows example detection output with contour markings. Table 1 summarises the size and health distribution across the full test set. More unhealthy colonies were found than healthy ones, which is typical in the laboratory samples used in this study.

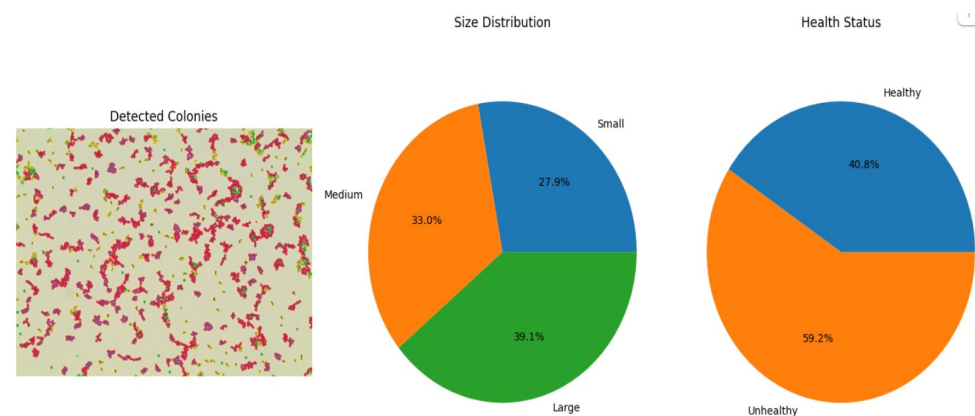


Figure 1. Colony detection output showing identified colonies on a Petri plate image with contour markings around each detected colony.

Table 1. Colony detection statistics from the full test dataset.

Parameter	Value
Total Colonies	530
Healthy Colonies	216 (40.8%)
Unhealthy Colonies	314 (59.2%)
Small Colonies	31.8%
Medium Colonies	33.0%
Large Colonies	35.1%

4.2. Model Performance Comparison

Table 2 presents the accuracy, precision, recall, and F1-score for all four classifiers evaluated on the test set.

SVM and ANN achieved the highest overall accuracy of 95.6%, followed by CNN at 95.0% and CNN+ViT at 93.7%. While the hybrid model did not lead in overall accuracy, confusion matrix analysis (Table 2 and Figure 2) shows that it eliminated all false positives, which is an important result for clinical applications.

Table 2. Model performance comparison.

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.956	0.957	0.956	0.956
ANN	0.956	0.957	0.956	0.956
CNN	0.950	0.951	0.950	0.950
Hybrid CNN+ViT	0.937	0.945	0.937	0.937

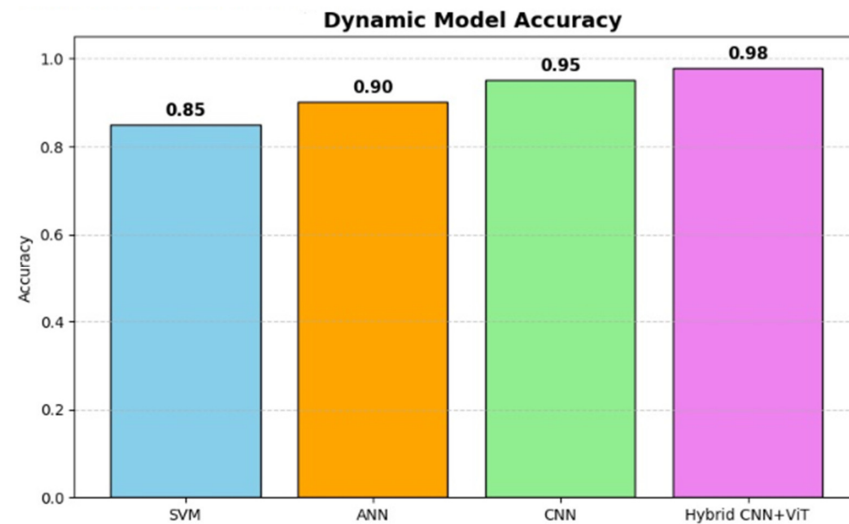
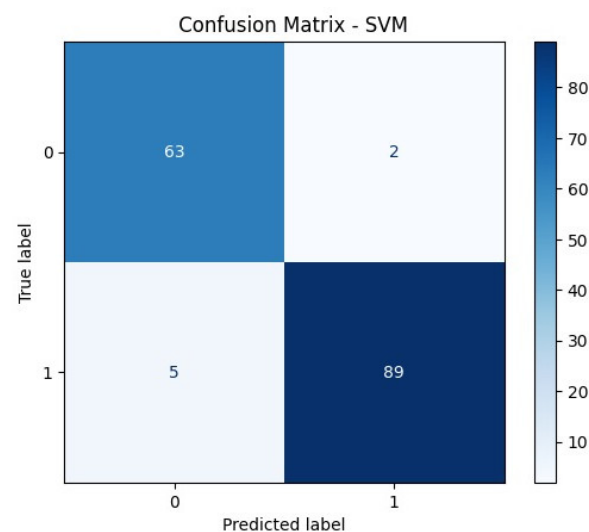
**Figure 2.** Bar chart comparing overall accuracy across SVM, ANN, CNN, and hybrid CNN+ViT models.

Table 3 and Figures 3–6 outlines the confusion matrix summary for all classification models (TP: true positive, FP: false positive, FN: false negative, TN: true negative).

Table 3. Confusion Matrix Summary.

Model	TP	FP	FN	TN
SVM	63	2	5	89
ANN	63	2	5	89
CNN	63	2	6	88
Hybrid CNN+ViT	65	0	10	84

**Figure 3.** Confusion matrix for SVM classifier.

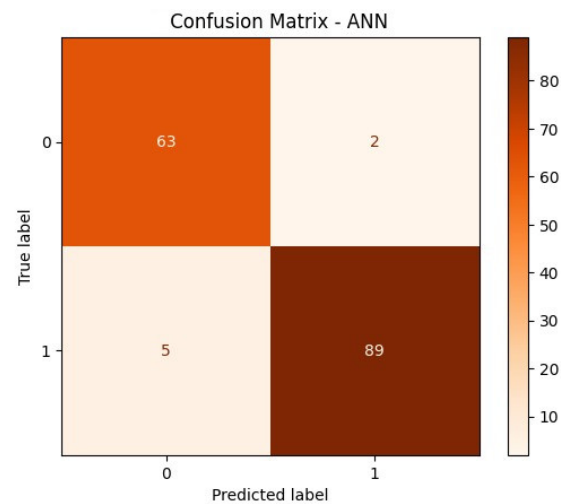


Figure 4. Confusion matrix for ANN classifier.

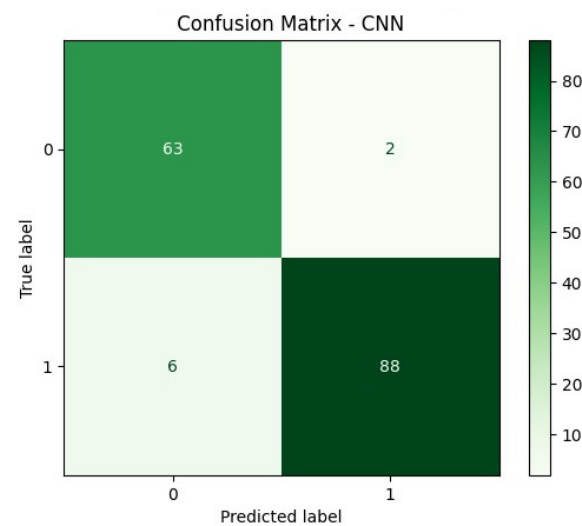


Figure 5. Confusion matrix for CNN classifier.

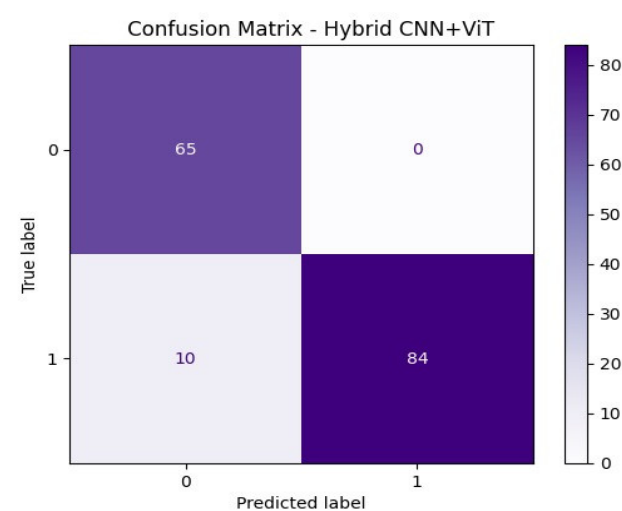


Figure 6. Confusion matrix for hybrid CNN+ViT classifier.

4.3. Feature Extraction and Real-Time Interface

Table 4 outlines the sequential processing stages from image input to classification output.

Table 4. Detection and processing workflow.

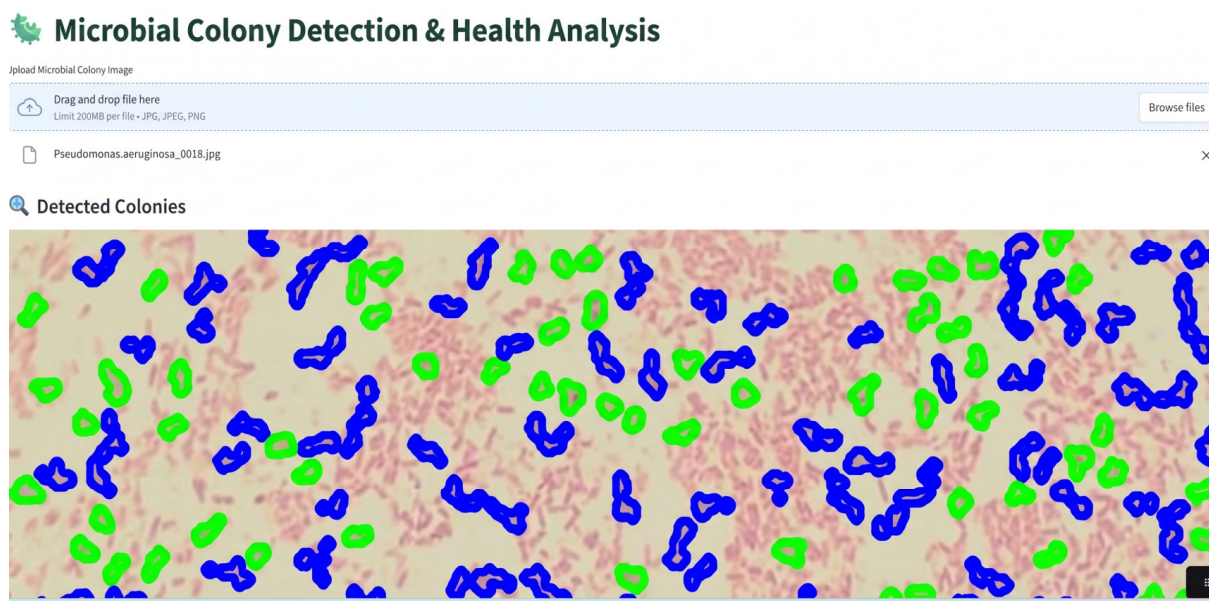
Stage	Operation Performed
Image Input	Petri plate image upload
Preprocessing	Resize, noise removal, contrast enhancement
Segmentation	Thresholding + contour detection
Feature Extraction	Area, Perimeter, Circularity
Classification	ML (SVM, ANN) and DL (CNN, CNN+ViT)

Table 5 shows an example of the feature output generated for individual colonies. Each colony receives a unique ID with its area, perimeter, circularity, and predicted health class. This structured output allows quantitative review and easy data export for laboratory records.

Table 5. Sample colony feature output.

Colony ID	Area (px)	Perimeter (px)	Circularity	Health Class
1	171	47.61	0.95	Unhealthy
2	305	76.53	0.82	Unhealthy
3	218	63.19	0.86	Healthy

The user-friendly web interface used for importing microscopic scans and displaying automated colony segmentations is illustrated in Figure 7.

**Figure 7.** Real-time Streamlit web interface showing colony analysis results including colony count, health status, and feature table.

Quantitative morphological attributes and final statistical counts are organized into the clean, accessible layout presented in Figure 8.

Colony Feature Table

	Area	Perimeter	Circularity	Health_Class
6	151	77.1127	0.3191	Unhealthy
7	226.5	74.5269	0.5125	Unhealthy
8	82	37.799	0.7212	Healthy
9	182	62.7696	0.5805	Healthy
10	343	117.5391	0.312	Unhealthy
11	274	98.5685	0.3544	Unhealthy
12	131.5	48.0416	0.716	Healthy
13	169.5	74.1838	0.387	Unhealthy
14	76	34.6274	0.7965	Healthy
15	171.5	60.5269	0.5883	Healthy

★ Total Colonies: 183

● Healthy: 96

▲ Unhealthy: 87

Figure 8. Colony feature table output generated by the system, listing area, perimeter, circularity, and health class for each detected colony.

As shown in Figure 9, the system computes real-time classification metrics and deep learning predictions regarding the biological health status of the culture.

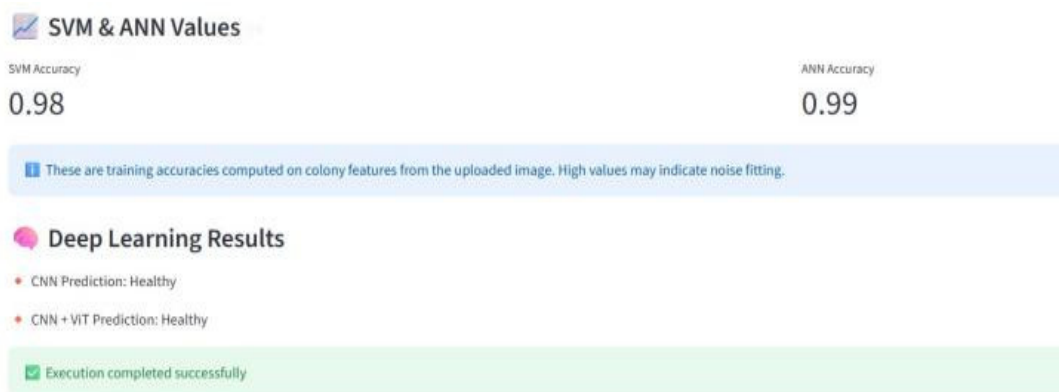


Figure 9. Deep learning model prediction results showing health classification output for the analysed Petri plate image.

5. Discussion

The results show that all four models classify bacterial colonies with reasonable accuracy. SVM and ANN both achieved 95.6%, which is slightly higher than CNN (95.0%) and CNN+ViT (93.7%) in terms of the overall metric. This may seem unexpected, since deep learning models are usually expected to outperform classical machine learning when the data are complex.

One likely explanation is the size of the training dataset. SVM and ANN tend to perform well on smaller, feature-based datasets because they do not require large amounts of data to generalise. CNN and CNN+ViT learn features directly from pixel data, which usually requires more training examples to achieve their full potential. A larger and more varied image set would probably change the ranking between models.

The confusion matrix results add an important perspective. CNN+ViT produced zero false positives, meaning it never incorrectly classified an unhealthy colony as healthy. In clinical applications, a false positive can lead to a wrong treatment decision, so this property is valuable. The trade-off is a higher number of false negatives—colonies that were actually positive but classified as negative. Whether this exchange is acceptable depends on the use case. For screening applications, eliminating false positives might be the priority. For research counting, overall accuracy may matter more.

Figures 7–9 outlined the Streamlit interface worked well during testing. It handled image uploads, processed the input in a short time, and displayed colony counts, size distributions, health labels, and feature tables in a format that is easy to read without technical knowledge. This makes the system usable in a practical laboratory setting, not just a research environment.

Some limitations remain. Image quality still affects predictions—poor lighting or non-standard Petri plates introduced some errors. The dataset used in this study came from a single laboratory setup, so performance on images from different equipment has not been verified. Future work should focus on collecting more diverse training data, applying better augmentation strategies, testing with a wider range of bacterial species, and improving the preprocessing pipeline to handle variable lighting conditions.

6. Conclusions

This paper presents an automated system for detecting and classifying bacterial colonies from Petri plate images using computer vision and machine learning. Four models were tested and compared: SVM, ANN, CNN, and a hybrid CNN+ViT. All models achieved acceptable accuracy, with SVM and ANN reaching the highest overall performance (95.6%) and CNN+ViT showing the important property of eliminating false positives entirely.

The proposed pipeline—from image capture to real-time output through a Streamlit web interface—demonstrates that automated colony analysis is a practical alternative to manual counting. The system is fast, consistent, and does not require the user to have programming skills. The main benefits are reduced manual errors, faster processing, and the ability to handle large numbers of images without fatigue.

Future improvements should target a larger and more diverse training dataset, better preprocessing for variable laboratory conditions, and real-world validation with clinical samples from different institutions.

Author Contributions: Conceptualisation: V.R., O.G., R.T. and D.K.V.; methodology, V.R., R.T. and D.K.V.; software, V.R.; validation, V.R., O.G. and D.K.V.; formal analysis, V.R. and O.G.; investigation, V.R. and R.T.; resources, D.K.V.; data curation, V.R. and O.G.; writing—original draft preparation, V.R. and R.T.; writing—review and editing, D.K.V. and O.G.; visualisation, V.R.; supervision, D.K.V.; project administration, D.K.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in <https://www.kaggle.com/datasets/nilay1987/bacterial-colony/>, accessed on 25 June 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khan, M.F.; Gupta, R.; Sekhri, S. Automated bacteria colony counting on agar plates using machine learning. *J. Environ. Eng.* **2021**, *147*, 04021066. [[CrossRef](#)]
2. Majchrowska, S.; Pawłowski, J.; Guła, G.; Bonus, T.; Hanas, A.; Loch, A.; Pawlak, A.; Roszkowiak, J.; Golan, T.; Drulis-Kawa, Z. Assessing microbial colony counting: A deep learning approach with the AGAR image dataset. *Neurocomputing* **2025**, *630*, 129654. [[CrossRef](#)]
3. Chen, S.; Huang, P.H.; Kim, H.; Cui, Y.; Buie, C.R. MCount: An automated colony counting tool for high-throughput microbiology. *PLoS ONE* **2025**, *20*, e0311242. [[CrossRef](#)] [[PubMed](#)]
4. M S S.; Martis, J.E.; Krivic, S.; K B, S.; Nazareth, P. A swarm-optimized microbial colony counter. *Expert Syst.* **2024**, *41*, e13510. [[CrossRef](#)]
5. Calgin, M.; Kalayci, H. Early classification of Gram-negative bacteria with colony imaging and deep learning without coding experience. *Proc. Bulg. Acad. Sci.* **2024**, *77*, 504–512. [[CrossRef](#)]
6. Yang, F.; Zhong, Y.; Yang, H.; Wan, Y.; Hu, Z.; Peng, S. Microbial colony detection based on deep learning. *Appl. Sci.* **2023**, *13*, 10568. [[CrossRef](#)]
7. Kong, F.; Su, M.; Guo, J.; Wu, H.; Chen, H.; Qiu, J.; Huang, J. Implementation of convolutional neural networks for microbial colony recognition. *Microbiol. Spectr.* **2025**, *13*, e02885-24. [[CrossRef](#)] [[PubMed](#)]
8. Amano, M.; Mai, D.T.; Sun, G.; Vu, T.N.; Hoi, L.T.; Hoa, N.T.; Ishibashi, K. Deep learning approach for classifying bacteria types using morphology of bacterial colony. In Proceedings of the 44th Annual International Conference of the IEEE EMBC, Glasgow, UK, 11–15 July 2022; pp. 2165–2168.
9. Rehman, A.; Saleem, Z.; Amjad, J.; Shah, S.R.; Siddique, K. A comparison of bacterial colonies count from Petri dishes utilizing Hough Transform and traditional manual counting. *arXiv* **2025**, arXiv:2505.20365.
10. Whipp, J.; Dong, A. YOLO-based deep learning to automate bacterial colony counting. In Proceedings of the IEEE International Conference on Multimedia Big Data (BigMM), Naples, Italy, 8–11 December 2022; pp. 120–124.
11. Silva, M.; Martinho, D.; Marreiros, G. Deep learning-based microbial colony detection on agar plates. In *EPIA Conference on Artificial Intelligence*; Springer: Cham, Switzerland, 2025; pp. 42–54.
12. Wang, M.; Luo, J.; Lin, K.; Chen, Y.; Huang, X.; Liu, J.; Wang, A.; Xiao, D. Colony-YOLO: A lightweight micro-colony detection network based on improved YOLOv8n. *Microorganisms* **2025**, *13*, 1617. [[CrossRef](#)] [[PubMed](#)]
13. Hallström, E.; Fatsis-Kavalopoulos, N.; Bimpis, M.; Wählby, C.; Hast, A.; Andersson, D.I. CombiANT reader: Deep learning-based automatic image processing tool to robustly quantify antibiotic interactions. *PLoS Digit. Health* **2025**, *4*, e0000669. [[CrossRef](#)] [[PubMed](#)]
14. Pintavirooj, C.; Bunkum, M.; Vongmanee, N.; Nampeng, J.; Visitsattapongse, S. Bacterial colony counting and classification system based on deep learning model. *Appl. Sci.* **2026**, *16*, 1313. [[CrossRef](#)]
15. Mennemann, K.; Ebert, N.; Reichardt, L.; Wasenmuller, O. Enhancing colony detection of microorganisms in agar dishes using SAM-based synthetic data augmentation in low-data scenarios. *Appl. Sci.* **2025**, *15*, 1260. [[CrossRef](#)]
16. Cao, L.; Zeng, L.; Wang, Y.; Cao, J.; Han, Z.; Chen, Y.; Wang, Y.; Zhong, G.; Qiao, S. U2-Net and ResNet50-based automatic pipeline for bacterial colony counting. *Microorganisms* **2024**, *12*, 201. [[CrossRef](#)] [[PubMed](#)]
17. Makrai, L.; Fodróczy, B.; Nagy, S.Á.; Czeiszing, P.; Csabai, I.; Szita, G.; Solymosi, N. Annotated dataset for deep-learning-based bacterial colony detection. *Sci. Data* **2023**, *10*, 97. [[CrossRef](#)] [[PubMed](#)]
18. Hsiao, W.W.W.; Qureshi, S.A.; Aman, H.; Chang, S.W.; Saravanan, A.; Lam, X.M. Artificial intelligence-enabled predictive system for Escherichia coli colony counting using patch-based supervised cytometry regression. *Microchem. J.* **2025**, *212*, 113206. [[CrossRef](#)]
19. Ciatka, M.; Burget, R.; Karasek, J.; Lancos, J. Clustering microbial material on agar plates: A modular system approach. *Array* **2025**, *28*, 100503. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.