

Bacterial Species Classification from Full-Plate Culture Images Using Deep Learning: A Comparative Study of CNNs and Vision Transformers

Abdulsalam Abdulsattar Abdulazez

General Directorate of Education in Anbar

Hussein Salim Abed

University of Al Maarif

Aythem Khairi Kareem

ayt19c1004@uoanbar.edu.iq

General Directorate of Education in Anbar

Mustafa Nadhim OWAID

General Directorate of Education in Anbar

Ahmed Adil Nafea

University of Anbar

Research Article

Keywords:

Posted Date: October 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11151354/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Bacterial Species Classification from Full-Plate Culture Images Using Deep Learning: A Comparative Study of CNNs and Vision Transformers

Abdulsalam Abdulsattar Abdulazez¹, Hussein Salim Abed²,
Aythem Khairi Kareem^{1*}, Mustafa Nadhim OWAID^{1,3},
Ahmed Adil Nafea⁴

¹Department of Heet Education, General Directorate of Education in Anbar, Ministry of Education, Heet, 31007, Anbar, Iraq.

²Department of Medical Laboratory Techniques, College of Health and Medical Technologies, Al-Maarif University College, Ramadi, Al-Anbar, Iraq.

³Department of Environment, College of Applied Science-Heet, University of Anbar, Heet, 31007, Anbar, Iraq.

⁴Department of Artificial Intelligence, College of Computer Science and IT, University of Anbar, Ramadi, Anbar, Iraq.

*Corresponding author(s). E-mail(s): ayt19c1004@uoanbar.edu.iq;
Contributing authors: alhetysalam@gmail.com;
Hussein.salem@uoanbar.edu.iq; mustafanowaid@uoanbar.edu.iq;
ahmed.a.n@uoanbar.edu.iq;

Abstract

Accurate identification of bacterial species from culture plates is essential for microbiological analysis, yet conventional identification procedures can be time-consuming and dependent on specialized expertise. This study investigates the feasibility of automated multiclass bacterial identification directly from complete culture-plate images and compares convolutional and transformer-based deep-learning architectures under a unified experimental protocol. A balanced dataset comprising 950 high-resolution images representing 19 clinically relevant bacterial species was used. ResNet50, DenseNet121, and ViT-B/16 were fine-tuned using

identical input resolution, training-only data augmentation, optimization settings, and stratified five-fold cross-validation. Performance was evaluated using accuracy, macro-precision, macro-recall, macro F1-score, Matthews correlation coefficient, Cohen’s kappa, class-wise performance, and aggregated out-of-fold error analysis. ViT-B/16 achieved the best overall performance, with a mean accuracy of $99.05 \pm 0.86\%$ and macro F1-score of $99.05 \pm 0.86\%$, followed by DenseNet121 with $97.58 \pm 1.69\%$ accuracy and ResNet50 with $94.21 \pm 1.75\%$. ViT-B/16 also achieved MCC and Cohen’s kappa values of approximately 0.99 and produced only nine misclassifications across the aggregated out-of-fold predictions. Class-wise analysis showed consistently high performance across most bacterial species, with errors concentrated in a limited number of visually challenging class pairs. These findings demonstrate the potential of deep-learning models, particularly Vision Transformers, for accurate whole-plate bacterial species classification without prior extraction of individual colonies. The study provides a reproducible comparative baseline for CNN- and transformer-based bacterial image classification and motivates future strain-independent and multi-centre validation.

Keywords: Bacterial species classification, Culture-plate imaging, Deep learning, Vision Transformer, Microbial identification

1 Introduction

1.1 General Background and Motivation

The morphological classification of microbial cultures and colonies relies heavily on specialized expertise and practical experience, which has limited the extent of automation in this area. This study investigates the potential for automating the identification of pathogenic microbial species using images of colonies obtained from different cultured microorganisms. Colony morphology of microorganisms such as size, color, edge, elevation, and texture, as observed on culture media in clinical laboratories, can be used to visually discriminate different bacteria [1]. Rapid and accurate identification of pathogens is crucial for early infection diagnosis and appropriate antimicrobial treatment, with colony morphology on the agar serving as an important initial screening method [2]. However, the process requires specialized expertise and is time-consuming, leading to inconsistent interpretations among clinical technologists [3].

1.2 Problem Statement

This study aims to classify the species of pathogenic bacteria to support subsequent clinical decision-making and antimicrobial susceptibility testing. The classical methods for bacterial identification are time-consuming, labor-intensive, and require specialized expertise [4]. Identification of pathogenic bacteria is very important to help us select or suggest suitable antibiotics/drugs. Bacteria play a remarkable and important role in healthcare, medical, ecological, agricultural, and industrial processes; thus, it is important to know and identify bacterial species [5]. However, this study aims to

develop an automated, cost-effective, and scalable bacterial identification framework, reducing diagnostic time and effort while maintaining high precision using systematic Deep Learning (DL).

1.3 Related previous works and their limitations

Artificial Intelligence (AI)-based image analysis techniques were used to identify pathogenic bacteria using characteristics of colony morphology automatically [6]. Recently, different Machine Learning approaches and models were applied to identify species of bacteria in the cultural media, according to colony morphology, staining of bacteria by Gram stain [7], and Support Vector Machine (SVM) [8]. Most colony-based studies have examined only a few microbial species under controlled conditions that may not reflect the morphological diversity found in routine clinical practice [9]. Microbial species like bacteria and fungi were studied by AI-based image recognition *Pseudomonas aeruginosa*, *Staphylococcus aureus*, *Escherichia coli*, *Enterococcus faecium* [1], *Klebsiella pneumoniae* [9] and *Aspergillus* species [10]. DL-Based Rapid identification was done using various models to identify numerous pathogenic bacteria depending on characteristics in the culture medium, such as Convolutional Neural Network (CNN) [11]. Hybrid model (MobileNet-XGBoost) [1], Hybrid model 2D Nano-material Cross-Response Sensing Array + Linear Discriminant Analysis (LDA) [12], GoogLeNet, [13] and You Only Look Once version 12 (YOLOv12) model [9]. Despite promising results, existing studies remain limited via small numbers of bacterial species, restricted datasets, single train-test splits, and reliance on isolated colony crops. These limitations highlight the need for a systematic and reproducible comparison of multiple DL architectures using complete culture-plate images under consistent experimental conditions.

1.4 Contributions and Key Results

The main contributions of this study can be summarized as follows:

- Whole-plate classification of 19 bacterial species using 950 high-resolution culture-plate images obtained from 151 clinical strains.
- A unified experimental comparison of ResNet50, DenseNet121, and ViT-B/16 under identical preprocessing, training-only data augmentation, optimization, cross-validation, and evaluation settings.
- A direct comparison of three distinct feature-learning paradigms: residual CNN learning through ResNet50, dense feature reuse through DenseNet121, and transformer-based global self-attention through ViT-B/16 for bacterial species classification.
- Evaluation using stratified five-fold cross-validation to provide a consistent and reproducible assessment of all three architectures.
- A comprehensive performance analysis using overall, class-wise, agreement-based, stability, and error-based evaluation measures, including accuracy, precision, recall, F1-score, Matthews correlation coefficient, Cohen's kappa, confusion matrices, and five-fold variability.

- The experimental results showed that ViT-B/16 achieved the highest overall performance, with a mean five-fold accuracy of $99.05 \pm 0.86\%$ and a macro F1-score of $99.05 \pm 0.86\%$, followed by DenseNet121 with $97.58 \pm 1.69\%$ accuracy and ResNet50 with $94.21 \pm 1.75\%$ accuracy.

The remainder of this paper is organized as follows. Section 2 presents the dataset, preprocessing pipeline, data augmentation, comparative model architectures, training configuration, and evaluation metrics. Section 3 discusses the experimental results, while Section 4 concludes the paper and outlines future research directions.

2 Methodology

The proposed methodology outlines a systematic DL pipeline for multiclass bacterial species classification, as shown in Figure 1. The process commences with the collection and preparation of complete bacterial culture-plate images, followed by RGB conversion, image resizing, tensor conversion, intensity scaling, and model-specific normalization. Stratified five-fold cross-validation is subsequently implemented, with data augmentation applied exclusively to the training subset of each fold. The processed images are independently classified using ResNet50, DenseNet121, and ViT-B/16 under unified training conditions. Finally, the models are evaluated using class-specific and overall performance metrics, and their five-fold results are aggregated to provide a reliable comparative analysis.

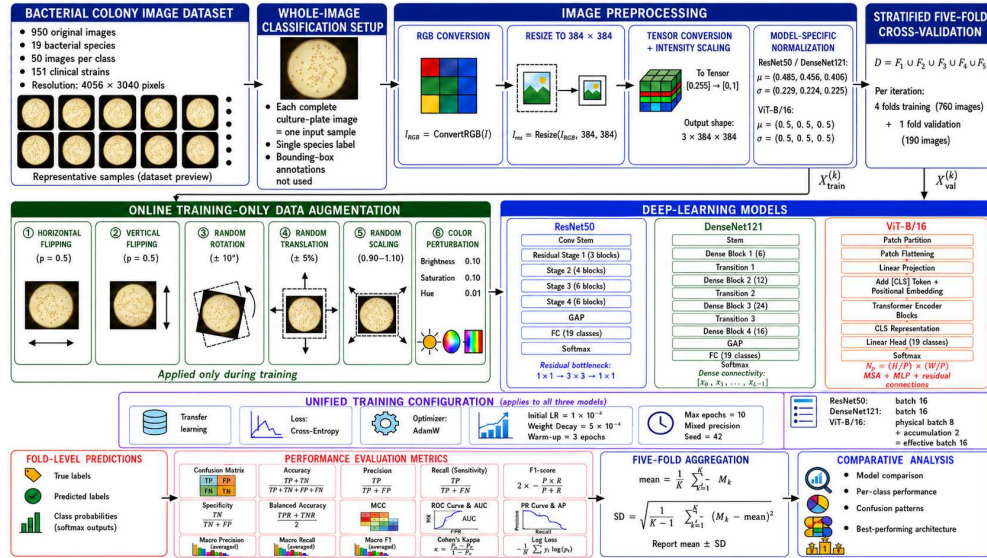


Fig. 1 Overall workflow of the proposed DL framework for multiclass bacterial species classification.

2.1 Dataset Description

The presented approach employed the high-resolution colony images of the clinically isolated bacteria dataset, developed for automated bacterial colony analysis and DL applications [14]. The dataset contains 950 original high-resolution Petri-dish images representing 19 clinically important bacterial species and 151 clinical strains. Each species contains exactly 50 images, resulting in a completely balanced multiclass dataset. The number of strains per species ranges from 4 to 14, allowing the dataset to capture phenotypic variation among clinical isolates of the same species. The bacterial species and their corresponding class codes are summarized in Table 1, while representative image samples from the dataset are illustrated in Figure 2. The clinical isolates were collected between August 2023 and May 2025 from respiratory and urinary tract specimens, blood, and other clinical specimens. The bacterial isolates were initially identified using the Vitek-2 Compact system and subsequently confirmed using matrix-assisted laser desorption/ionization time-of-flight mass spectrometry (MALDI-TOF MS). Depending on the bacterial species, colonies were cultured on Blood Agar Plate, MacConkey agar, or Salmonella–Shigella agar and incubated at 35°C for approximately 24–48 hours.

The images were acquired using the Micro-Chronos automated imaging system in a fully enclosed and light-proof environment. Each inverted Petri dish was photographed from directly above by a fixed camera in such a way as to minimize geometric distortion and variations in the angle of view. The imaging system employed dual-source lighting: the transmitted light highlighted the boundaries of the colony and its base projections, whereas the annular reflected light improved the surface texture, contrast, and color features. Those images in which the number of visible colonies was inadequate or in which the image quality was poor were excluded manually. The final images were saved in high-quality JPG format at an original spatial resolution of 4056×3040 pixels.

Table 1 Bacterial species included in the dataset.

Code	Bacterial species	Code	Bacterial species
aba	<i>Acinetobacter baumannii</i>	ppu	<i>Pseudomonas putida</i>
bce	<i>Bacillus cereus</i>	sag	<i>Streptococcus agalactiae</i>
bcp	<i>Burkholderia cepacia</i>	sau	<i>Staphylococcus aureus</i>
cst	<i>Corynebacterium striatum</i>	sep	<i>Staphylococcus epidermidis</i>
eca	<i>Enterococcus casseliflavus</i>	ses	<i>Salmonella enterica</i> subsp. <i>enterica</i>
eco	<i>Escherichia coli</i>	sma	<i>Serratia marcescens</i>
efa	<i>Enterococcus faecium</i>	spn	<i>Streptococcus pneumoniae</i>
enc	<i>Enterobacter cloacae</i>	spy	<i>Streptococcus pyogenes</i>
kpn	<i>Klebsiella pneumoniae</i>	stm	<i>Stenotrophomonas maltophilia</i>
mno	<i>Morganella morganii</i>		

The complete bacterial image dataset is defined in Equation 1:

$$\mathcal{D} = \{(I_i, y_i)\}_{i=1}^N, \quad (1)$$

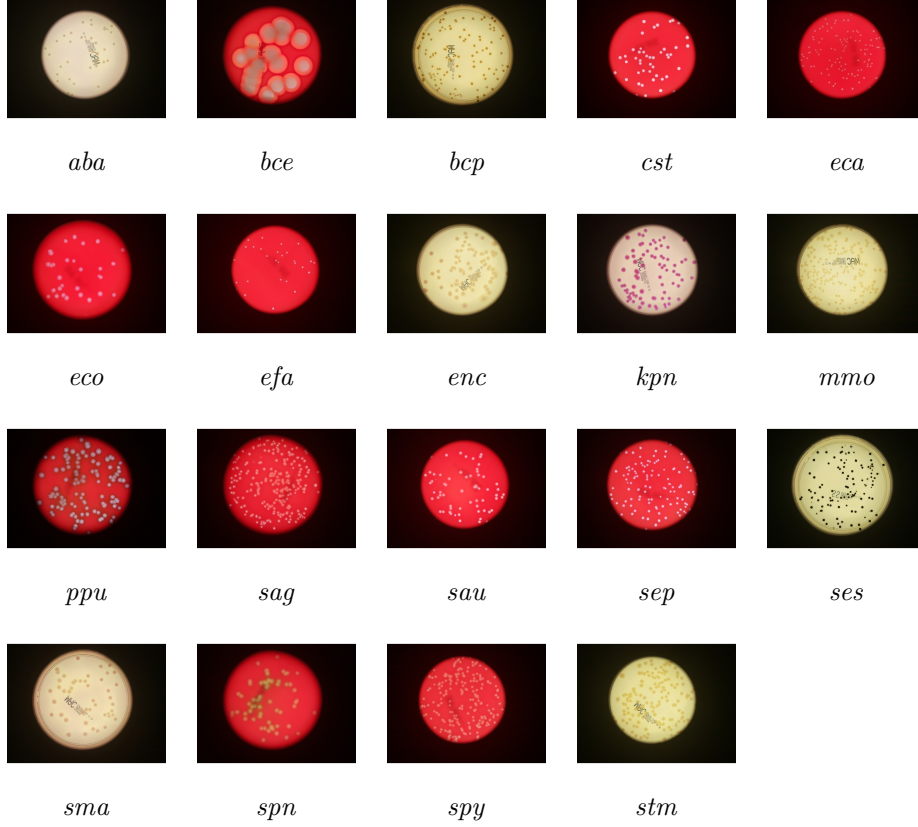


Fig. 2 Dataset sample.

where I_i denotes the i th bacterial colony image, $y_i \in \{1, 2, \dots, C\}$ denotes its corresponding class label, $N = 950$ is the total number of images, and $C = 19$ is the total number of bacterial species.

2.2 Image Preprocessing

The same preprocessing pipeline was used for all the bacterial colony images before model training in order to ensure compatibility with ResNet50, DenseNet121 and ViT-B/16, and this preprocessing pipeline was also consistently applied in all the cross-validation folds and models.

2.2.1 RGB Color Conversion

Before being fed into the classification models, each bacterial colony image was converted into a three-channel Red–Green–Blue (RGB) representation. This step ensured a consistent input format for ResNet50, DenseNet121, and ViT-B/16, all of which require three-channel images. The RGB conversion is defined in Equation 2[15]:

$$I_{\text{RGB}} = \text{ConvertRGB}(I), \quad (2)$$

where I denotes the original image, I_{RGB} represents the resulting RGB image, and $\text{ConvertRGB}(\cdot)$ denotes the color-conversion function.

2.2.2 Image Resizing

The original bacterial colony images had a spatial resolution of 4056×3040 pixels. To provide a uniform input size for ResNet50, DenseNet121, and ViT-B/16, each RGB image was resized to 384×384 pixels. The resizing operation is defined in Equation 3 [16]:

$$I_{\text{res}} = \text{Resize}(I_{\text{RGB}}, 384, 384), \quad (3)$$

where I_{RGB} denotes the original three-channel RGB image, $\text{Resize}(\cdot)$ represents the spatial resizing operation, and $I_{\text{res}} \in \mathbb{R}^{384 \times 384 \times 3}$ denotes the resized RGB image. Bilinear interpolation was employed to estimate pixel intensities at the new spatial coordinates. The same input resolution was used for all three architectures to ensure a consistent and fair comparative evaluation.

2.2.3 Tensor Conversion and Intensity Scaling

Each resized RGB image was converted from a conventional image array into a floating-point tensor suitable for deep-learning computation, as defined in Equation 4 [17]:

$$I_{\text{ten}} = \text{ToTensor}(I_{\text{res}}), \quad (4)$$

where I_{ten} denotes the resulting floating-point image tensor and $\text{ToTensor}(\cdot)$ represents the transformation that converts the image representation from height–width–channel ($H \times W \times C$) format to channel–height–width ($C \times H \times W$) format.

During this transformation, the original 8-bit pixel intensities were simultaneously scaled from $[0, 255]$ to $[0, 1]$, as defined in Equation 5:

$$I_{\text{ten}}^{(ch)}(x, y) = \frac{I_{\text{res}}^{(ch)}(x, y)}{255}, \quad (5)$$

where ch denotes the color-channel index and (x, y) represents the spatial coordinates. Consequently, the resulting tensor satisfies $I_{\text{ten}} \in \mathbb{R}^{3 \times 384 \times 384}$, with pixel intensities restricted to the interval $[0, 1]$. This step produced scaled floating-point inputs suitable for subsequent model-specific normalization and processing by ResNet50, DenseNet121, and ViT-B/16.

2.2.4 Model-Specific Normalization

Each image tensor was normalized using the channel-wise mean and standard deviation associated with the pretrained configuration of the corresponding architecture. This step aligns the input distribution with that used during model pretraining and supports stable transfer learning. The normalization operation is defined in Equation 6 [18]:

$$I_{\text{norm}}^{(ch)}(x, y) = \frac{I_{\text{ten}}^{(ch)}(x, y) - \mu_{ch}}{\sigma_{ch}}, \quad (6)$$

where $I_{\text{ten}}^{(ch)}(x, y)$ denotes the scaled pixel intensity at spatial position (x, y) in channel ch , μ_{ch} and σ_{ch} denote the corresponding channel-wise mean and standard deviation, respectively, and $I_{\text{norm}}^{(ch)}(x, y)$ represents the resulting normalized pixel value.

For ResNet50 and DenseNet121, the ImageNet normalization parameters associated with their pretrained configurations are defined in Equations 7 and 8:

$$\boldsymbol{\mu}_{\text{CNN}} = (0.485, 0.456, 0.406), \quad (7)$$

$$\boldsymbol{\sigma}_{\text{CNN}} = (0.229, 0.224, 0.225). \quad (8)$$

For ViT-B/16, the normalization parameters specified by its pretrained configuration are defined in Equations 9 and 10 [19]:

$$\boldsymbol{\mu}_{\text{ViT}} = (0.5, 0.5, 0.5), \quad (9)$$

$$\boldsymbol{\sigma}_{\text{ViT}} = (0.5, 0.5, 0.5). \quad (10)$$

The three elements of each vector correspond to the red, green, and blue channels, respectively. Model-specific normalization ensured that each architecture received inputs consistent with its original pretraining distribution, thereby facilitating stable fine-tuning and maintaining a consistent comparative framework across the three models.

2.3 Stratified Five-Fold Cross-Validation

A stratified five-fold cross-validation strategy was employed to evaluate ResNet50, DenseNet121, and ViT-B/16 under identical data partitions.

The dataset was divided into five mutually exclusive subsets, as defined in Equation 11 [20]:

$$\mathcal{D} = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3 \cup \mathcal{F}_4 \cup \mathcal{F}_5, \quad \mathcal{F}_i \cap \mathcal{F}_j = \emptyset, \quad i \neq j \quad (11)$$

The validation set $\mathcal{D}_{\text{val}}^{(k)}$ for the k th cross-validation iteration is defined in Equation 12:

$$\mathcal{D}_{\text{val}}^{(k)} = \mathcal{F}_k \quad (12)$$

The training set $\mathcal{D}_{\text{train}}^{(k)}$ for the k th cross-validation iteration is defined in Equation 13:

$$\mathcal{D}_{\text{train}}^{(k)} = \bigcup_{\substack{j=1 \\ j \neq k}}^5 \mathcal{F}_j \quad (13)$$

where $k \in \{1, 2, 3, 4, 5\}$ denotes the current cross-validation iteration.

Folds were generated using shuffled stratification with a fixed random seed of 42. Identical fold assignments were applied to ResNet50, DenseNet121, and ViT-B/16 to facilitate a fair comparative analysis.

2.4 Data Augmentation

Data augmentation was applied exclusively to the training subset of each cross-validation fold to increase data diversity and improve model robustness. No augmented images were permanently stored, and a new random transformation was generated whenever an image was loaded during training [21]. The validation subset remained unchanged and was processed only through the deterministic preprocessing pipeline without any stochastic augmentation.

The augmented image I_{aug} is defined in Equation 14:

$$I_{\text{aug}} = \text{Aug}(I) \quad (14)$$

where $\text{Aug}(\cdot)$ represents the composition of stochastic geometric and color transformations.

2.4.1 Horizontal Flipping

Horizontal flipping was applied randomly to each training image with a probability of 0.5. This transformation reverses the image along the vertical axis while preserving its height, width, color channels, and bacterial colony characteristics. Horizontal flipping is defined in Equation 15 [21]:

$$I_H = \begin{cases} \text{Flip}_H(I), & r_H < 0.5, \\ I, & r_H \geq 0.5, \end{cases} \quad (15)$$

where I denotes the input training image, I_H is the resulting image, $\text{Flip}_H(\cdot)$ represents the horizontal-flipping operation, and $r_H \sim U(0, 1)$ is a randomly generated value from a uniform distribution.

The horizontal flipping at the pixel level can be expressed using Equation 16:

$$I_H(x, y, ch) = I(x, W - 1 - y, ch), \quad (16)$$

where x and y denote the vertical and horizontal pixel coordinates, respectively, ch represents the color channel, and W is the image width.

2.4.2 Vertical Flipping

Vertical flipping was randomly applied to each training image with a probability of 0.5. This transformation reverses the image along the horizontal axis while preserving its dimensions, color channels, and colony morphology. Vertical flipping is defined in Equation 17 [22]:

$$I_{HV} = \begin{cases} \text{Flip}_V(I_H), & r_V < 0.5, \\ I_H, & r_V \geq 0.5, \end{cases} \quad (17)$$

where I_{HV} is the resulting image, $\text{Flip}_V(\cdot)$ denotes the vertical-flipping operation, and $r_V \sim U(0, 1)$ is a random value sampled from a uniform distribution.

The vertical flipping at the pixel level can be expressed using Equation 18:

$$I_{HV}(x, y, ch) = I_H(H - 1 - x, y, ch), \quad (18)$$

where H is the image height.

2.4.3 Random Rotation

Random rotation was applied to each training image to simulate small variations in Petri-dish orientation. The rotation angle was sampled uniformly from the interval defined in Equation 19 [22]:

$$\theta \sim U(-10^\circ, 10^\circ), \quad (19)$$

where θ denotes the randomly selected rotation angle. The transformed image is defined in Equation 20:

$$I_R = \text{Rotate}(I_{HV}, \theta), \quad (20)$$

where I_R is the resulting rotated image, and $\text{Rotate}(\cdot)$ represents the rotation operation.

For a pixel located at (x, y) , the rotated coordinates (x', y') around the image center (x_c, y_c) are calculated as expressed in Equation 21:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix}, \quad (21)$$

where (x, y) denotes the original pixel coordinates, (x', y') represents the transformed coordinates, and (x_c, y_c) is the center of rotation.

2.4.4 Random Translation

Random translation was applied to the training images to simulate minor changes in the Petri dish's position within the image frame. The horizontal and vertical translation distances were independently sampled within 5% of the image width and height, as expressed in Equations 22 and 23 [23]:

$$t_x \sim U(-0.05W, 0.05W), \quad (22)$$

$$t_y \sim U(-0.05H, 0.05H), \quad (23)$$

where W and H denote the image width and height, respectively, and t_x and t_y represent the horizontal and vertical translation distances.

The translated image is defined in Equation 24:

$$I_T = \text{Translate}(I_R, t_x, t_y), \quad (24)$$

where I_T is the resulting translated image, and $\text{Translate}(\cdot)$ denotes the spatial translation operation.

The translated coordinates (x', y') for a pixel located at (x, y) are calculated as shown in Equations 25 and 26:

$$x' = x + t_x, \quad (25)$$

$$y' = y + t_y, \quad (26)$$

Equivalently, the transformation can be represented in homogeneous coordinates as shown in Equation 27:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (27)$$

2.4.5 Random Scaling

Random scaling was applied to the training images to simulate small variations in the apparent size of the Petri dish and bacterial colonies. A scaling factor was randomly sampled from the interval defined in Equation 28 [24]:

$$s_f \sim U(0.90, 1.10), \quad (28)$$

where s_f denotes the selected scaling factor. Values of $s_f < 1$ produce a zoom-out effect, whereas values of $s_f > 1$ produce a zoom-in effect.

The scaled image is defined in Equation 29:

$$I_S = \text{Scale}(I_T, s_f), \quad (29)$$

where I_S is the resulting scaled image, and $\text{Scale}(\cdot)$ represents the scaling operation.

The scaling around the image center (x_c, y_c) for a pixel located at (x, y) is expressed using Equations 30 and 31:

$$x' = s_f(x - x_c) + x_c, \quad (30)$$

$$y' = s_f(y - y_c) + y_c, \quad (31)$$

The equivalent homogeneous transformation is expressed in Equation 32:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_f & 0 & (1 - s_f)x_c \\ 0 & s_f & (1 - s_f)y_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (32)$$

Since rotation, translation, and scaling were jointly implemented using a random affine transformation, the final geometrically augmented image is defined in Equation 33:

$$I_A = \text{Affine}(I_{HV}, \theta, t_x, t_y, s_f), \quad (33)$$

where I_A represents the final image after applying the combined geometric transformations, I_{HV} denotes the image obtained after the sequential horizontal- and vertical-flipping stages described in section 2.4.1 and 2.4.2, θ is the rotation angle, t_x and t_y are the horizontal and vertical translation values, and s_f is the scaling factor.

2.4.6 Color Perturbation

Color perturbation was applied to the training images to improve robustness against minor variations in illumination and colony appearance. Brightness, saturation, and hue were randomly adjusted, as defined in Equation 34 [25]:

$$I_C = \text{ColorJitter}(I_A, b_f, s_c, h_f), \quad (34)$$

where I_A is the geometrically augmented image, I_C is the color-perturbed output, and $\text{ColorJitter}(\cdot)$ denotes the stochastic color-transformation function, b_f presents the brightness factor, which is sampled as $b_f \sim U(0.90, 1.10)$,

s_c presents the saturation factor, which is sampled as $s_c \sim U(0.90, 1.10)$, and h_f shifted by a randomly sampled value, which is sampled as $h_f \sim U(-0.01, 0.01)$.

The adopted color-augmentation settings were brightness= 0.10, saturation= 0.10, hue= 0.01.

The slight changes made kept the basic shape of the bacterial colony unchanged while at the same time decreasing the model's sensitivity to small variations in color and lighting.

2.5 Comparative Deep-Learning Models

Three pretrained DL architectures (ResNet50, DenseNet121, and ViT-B/16) were independently employed to classify the bacterial colony images into 19 species. The selected models represent two major feature-learning paradigms: CNN, represented by ResNet50 and DenseNet121, and transformer-based learning, represented by ViT-B/16. This selection enabled a comparative assessment of residual feature learning, dense feature reuse, and global self-attention for bacterial colony classification.

The models received two types of inputs for each cross-validation fold. During training, each model processed augmented and preprocessed images from the training subset, as presented in Equation 35:

$$\mathbf{X}_{\text{train}}^{(k)} \in \mathbb{R}^{B \times 3 \times 384 \times 384}, \quad (35)$$

where $\mathbf{X}_{\text{train}}^{(k)}$ denotes the training subset of the k th fold, B is the batch size, 3 represents the RGB channels, and 384×384 is the input resolution. During evaluation, the models processed images from the held-out validation fold using deterministic preprocessing without stochastic augmentation as expressed in Equation 36:

$$\mathbf{X}_{\text{val}}^{(k)} \in \mathbb{R}^{B \times 3 \times 384 \times 384}, \quad (36)$$

where $\mathbf{X}_{\text{val}}^{(k)}$ denotes the validation subset of the k th fold.

The classification process for each model can be presented in Equation 37:

$$\mathbf{z}_m = f_m(\mathbf{X}; \boldsymbol{\theta}_m) \in \mathbb{R}^{19}, \quad (37)$$

where m is the model's name $m \in \{\text{ResNet50}, \text{DenseNet121}, \text{ViT-B/16}\}$, $f_m(\cdot)$ denotes the corresponding architecture, $\boldsymbol{\theta}_m$ represents its learnable parameters, and $\mathbf{z}_m$ is the vector of 19 class logits. The original classification head of each pretrained model was replaced with a task-specific layer containing 19 output neurons.

The logits were transformed into class probabilities using the softmax function as expressed in Equation 38:

$$p_m(c \mid \mathbf{X}) = \frac{\exp(z_{m,c})}{\sum_{j=1}^{19} \exp(z_{m,j})}, \quad (38)$$

where $p_m(c \mid \mathbf{X})$ is the probability assigned by model m to bacterial class c , and $z_{m,c}$ is the corresponding logit. The predicted bacterial species was then determined using Equation 39:

$$\hat{y}_m = \arg \max_{c \in \{1, \dots, 19\}} p_m(c \mid \mathbf{X}). \quad (39)$$

An independent test set was not utilized in the experimental protocol. Model performance was instead assessed using stratified five-fold cross-validation. In each iteration, four folds consisting of 760 images were allocated for model training, and the remaining fold containing 190 images was designated as the validation set for performance evaluation. For a rigorous and unbiased comparative analysis, ResNet50, DenseNet121, and ViT-B/16 were trained and evaluated with identical fold assignments, input resolutions, augmentation strategies applied only during training, effective batch sizes, optimization parameters, and evaluation metrics.

2.5.1 ResNet50

ResNet50 was employed as the first convolutional baseline for bacterial colony image classification [26]. The model was initialized using pretrained ImageNet weights and subsequently fine-tuned on the 19-class bacterial dataset. Figure 3 shows the ResNet50 architecture.

For each cross-validation fold, ResNet50 received complete preprocessed Petri-dish images represented as tensors of size as expressed in Equation 40:

$$\mathbf{X} \in \mathbb{R}^{B \times 3 \times 384 \times 384}, \quad (40)$$

ResNet50 addresses the degradation problem in deep convolutional networks through residual learning. Instead of directly learning the desired mapping $\mathcal{H}(\mathbf{x})$, each residual block learns a residual function $\mathcal{F}(\mathbf{x}; \boldsymbol{\theta}_R)$, as expressed in Equation 41:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}; \boldsymbol{\theta}_R) + \mathbf{x}, \quad (41)$$

where $\mathbf{x}$ is the input feature map, $\boldsymbol{\theta}_R$ represents the learnable parameters of the residual branch, and $\mathbf{y}$ is the block output. The identity shortcut connection enables the original input to bypass the nonlinear transformations, thereby supporting stable gradient propagation through the deep network.

ResNet50 uses bottleneck residual blocks composed of three convolutional operations using Equation 42:

$$\mathcal{F}(\mathbf{x}) = \text{Conv}_{1 \times 1}(\text{Conv}_{3 \times 3}(\text{Conv}_{1 \times 1}(\mathbf{x}))), \quad (42)$$

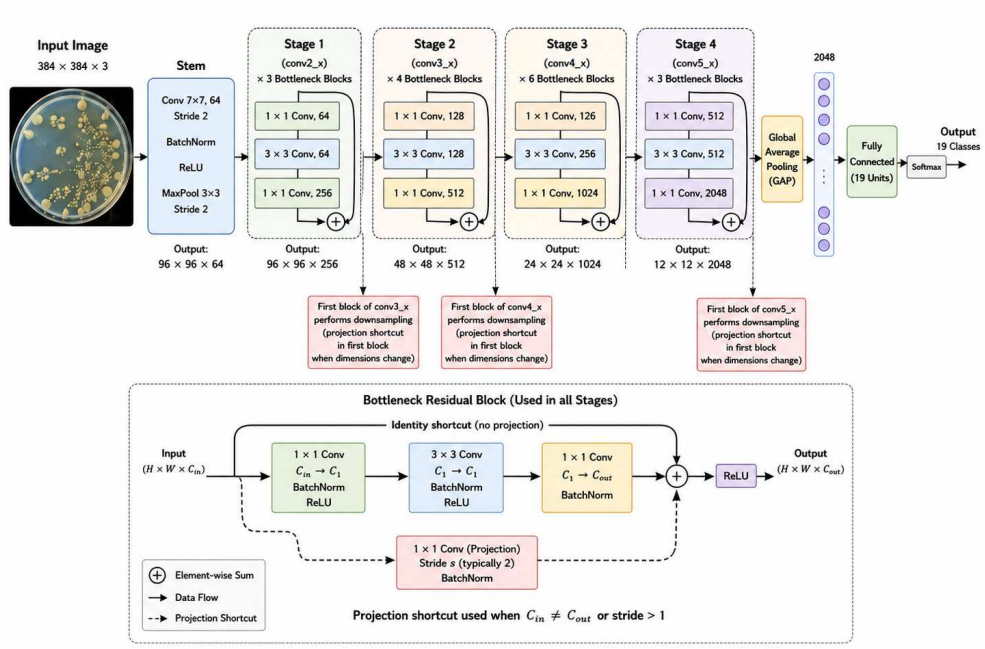


Fig. 3 ResNet50 architecture.

The first 1×1 convolution reduces the channel dimensionality, the 3×3 convolution extracts spatial features, and the final 1×1 convolution restores the feature dimensionality. Batch normalization and nonlinear activation are applied within the residual blocks.

The network consists of an initial convolutional stem followed by four residual stages containing 3, 4, 6, and 3 bottleneck blocks, respectively. As the image propagates through these stages, its spatial resolution is progressively reduced while the number and abstraction level of the feature maps increase. The extracted deep representation is denoted in Equation 43:

$$\mathbf{F}_R = f_{\text{ResNet50}}(\mathbf{X}; \boldsymbol{\theta}_R), \quad (43)$$

where $f_{\text{ResNet50}}(\cdot)$ represents the ResNet50 feature extractor, $\boldsymbol{\theta}_R$ denotes its learnable parameters, and $\mathbf{F}_R$ is the resulting high-level feature map.

Global Average Pooling (GAP) was subsequently applied to convert the spatial feature maps into a one-dimensional feature vector using Equation 44:

$$g_{R,c} = \frac{1}{H_f W_f} \sum_{i=1}^{H_f} \sum_{j=1}^{W_f} F_R(i, j, c), \quad (44)$$

where H_f and W_f are the height and width of the final feature map, c is the channel index, and $\mathbf{g}_R$ is the pooled feature representation.

The original pretrained classification layer was replaced with a task-specific fully connected layer containing 19 output neurons as expressed in Equation 45:

$$\mathbf{z}_R = \mathbf{W}_R \mathbf{g}_R + \mathbf{b}_R, \quad \mathbf{z}_R \in \mathbb{R}^{19}, \quad (45)$$

where $\mathbf{W}_R$ and $\mathbf{b}_R$ are the learnable classifier weights and biases, respectively, and $\mathbf{z}_R$ is the vector of class logits of ResNet50 model.

The class probabilities were obtained using the softmax function as expressed in Equation 46:

$$p_R(c | \mathbf{X}) = \frac{\exp(z_{R,c})}{\sum_{j=1}^{19} \exp(z_{R,j})}, \quad (46)$$

where $p_R(c | \mathbf{X})$ is the probability assigned by ResNet50 model to the bacterial class c .

The predicted bacterial species was then determined using Equation 47:

$$\hat{y}_R = \arg \max_{c \in \{1, \dots, 19\}} p_R(c | \mathbf{X}), \quad (47)$$

where $\hat{y}_R$ denotes the final prediction of the ResNet50 model.

2.5.2 DenseNet121

DenseNet121 was employed as the second convolutional architecture to classify complete bacterial culture plate images into 19 species. The model was initialized using pretrained ImageNet weights and fine-tuned on the bacterial image dataset. Figure 4 shows the DenseNet121 architecture [27].

For each cross-validation fold, the model received preprocessed image batches represented as $\mathbf{X} \in \mathbb{R}^{B \times 3 \times 384 \times 384}$.

Unlike conventional convolutional networks, DenseNet121 establishes direct connections between each layer and all preceding layers within the same dense block. Therefore, the input to layer l is formed by concatenating the feature maps generated by layers 0 through $l - 1$ as expressed in Equation 48 [28]:

$$\mathbf{x}_l = \mathcal{H}_l([\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{l-1}]), \quad (48)$$

where $\mathbf{x}_l$ is the output of layer l , $\mathcal{H}_l(\cdot)$ denotes the nonlinear transformation performed by that layer, and $[\cdot]$ represents channel-wise concatenation.

Each dense layer applies Batch Normalization (BN), Rectified Linear Unit (ReLU) activation, a 1×1 convolution, and a 3×3 convolution as expressed in Equation 49:

$$\mathcal{H}_l(\mathbf{x}) = \text{Conv}_{3 \times 3}(\text{ReLU}(\text{BN}(\text{Conv}_{1 \times 1}(\text{ReLU}(\text{BN}(\mathbf{x}))))), \quad (49)$$

The 1×1 convolution acts as a bottleneck that reduces the channel dimensionality, whereas the 3×3 convolution extracts local spatial features. Each layer contributes

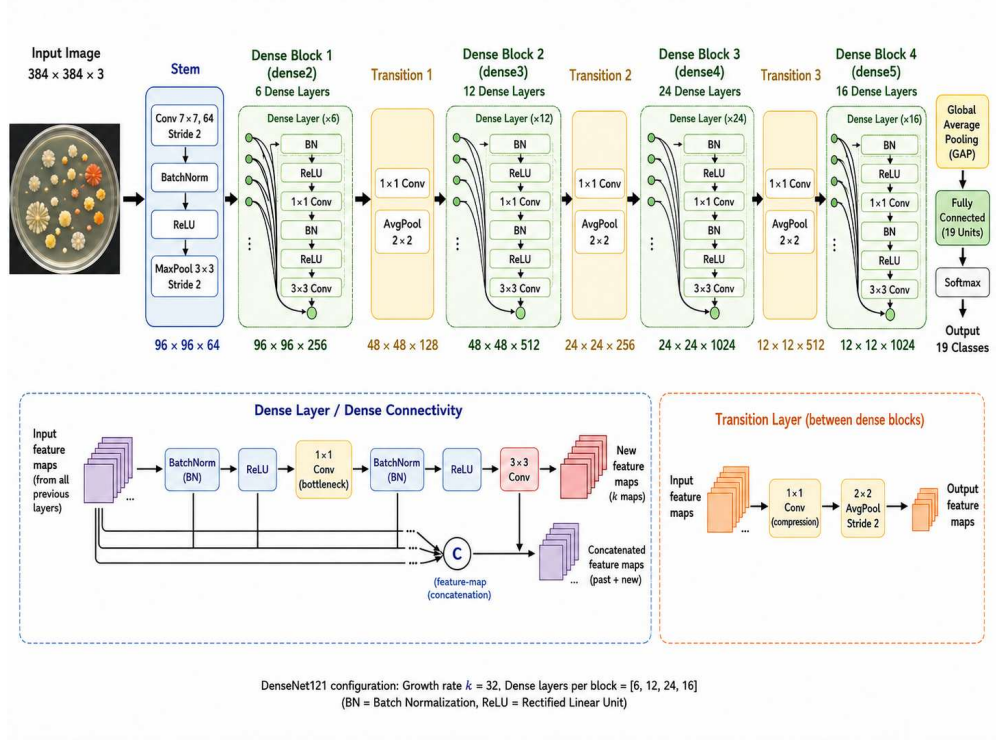


Fig. 4 DenseNet121 architecture.

a fixed number of new feature maps, referred to as the growth rate k . Thus, after l layers, the number of channels is expressed in Equation 50:

$$C_l = C_0 + k l, \quad (50)$$

where C_0 is the initial number of channels and k is the growth rate.

DenseNet121 contains four dense blocks with 6, 12, 24, and 16 dense layers, respectively. Transition layers are inserted between consecutive dense blocks to reduce the spatial dimensions and compress the number of feature maps. A transition layer $\mathbf{x}_{tr}$ is represented using Equation 51 [29]:

$$\mathbf{x}_{tr} = \text{AvgPool}_{2 \times 2}(\text{Conv}_{1 \times 1}(\mathbf{x})), \quad (51)$$

The complete feature-extraction process is defined in Equation 52:

$$\mathbf{F}_D = f_{\text{DenseNet121}}(\mathbf{X}; \boldsymbol{\theta}_D), \quad (52)$$

where $\mathbf{F}_D$ is the final high-level feature map of DenseNet121 model, $f_{\text{DenseNet121}}(\cdot)$ denotes the DenseNet121 feature extractor, and $\boldsymbol{\theta}_D$ represents its learnable parameters.

GAP was then applied to aggregate each feature channel using Equation 53:

$$g_{D,c} = \frac{1}{H_f W_f} \sum_{i=1}^{H_f} \sum_{j=1}^{W_f} F_D(i, j, c), \quad (53)$$

where $\mathbf{g}_D$ is the pooled feature representation of DenseNet121 model.

The original pretrained classifier was replaced with a task-specific fully connected layer containing 19 output neurons using Equation 54:

$$\mathbf{z}_D = \mathbf{W}_D \mathbf{g}_D + \mathbf{b}_D, \quad \mathbf{z}_D \in \mathbb{R}^{19}, \quad (54)$$

where $\mathbf{W}_D$ and $\mathbf{b}_D$ are the learnable classifier weights and biases, respectively, and $\mathbf{z}_D$ is the vector of class logits of DenseNet121 model.

The class probabilities were obtained using the softmax function using Equation 55:

$$p_D(c | \mathbf{X}) = \frac{\exp(z_{D,c})}{\sum_{j=1}^{19} \exp(z_{D,j})}, \quad (55)$$

where $p_D(c | \mathbf{X})$ is the probability assigned by DenseNet121 model to the bacterial class c .

The predicted bacterial species was then determined as shown in Equation 56:

$$\hat{y}_D = \arg \max_{c \in \{1, \dots, 19\}} p_D(c | \mathbf{X}), \quad (56)$$

where $\hat{y}_D$ denotes the final prediction of the DenseNet121 model.

2.5.3 Vision Transformer Base/16 (ViT-B/16)

ViT-B/16 was employed as the transformer-based architecture for classifying complete bacterial culture-plate images into 19 species. The model was initialized using the pre-trained configuration and subsequently fine-tuned on the bacterial image dataset [30]. Figure 5 shows the ViT-B/16 architecture.

For each cross-validation fold, ViT-B/16 received preprocessed image batches represented as $\mathbf{X} \in \mathbb{R}^{B \times 3 \times 384 \times 384}$.

Unlike CNN, ViT-B/16 divides each input image into a sequence of non-overlapping square patches. The total number of patches is expressed in Equation 57 [31]:

$$N_p = \frac{H}{P} \times \frac{W}{P}, \quad (57)$$

where N_p denotes the number of image patches and P represents the patch size.

Each patch is flattened into a one-dimensional vector, as expressed in Equation 58 [32]:

$$\mathbf{x}_p^{(i)} \in \mathbb{R}^{P^2 C}, \quad (58)$$

where $\mathbf{x}_p^{(i)}$ denotes the i th image patch and C is the number of image channels.

The flattened patches are mapped into a D -dimensional embedding space using a learnable linear projection using Equation 59:

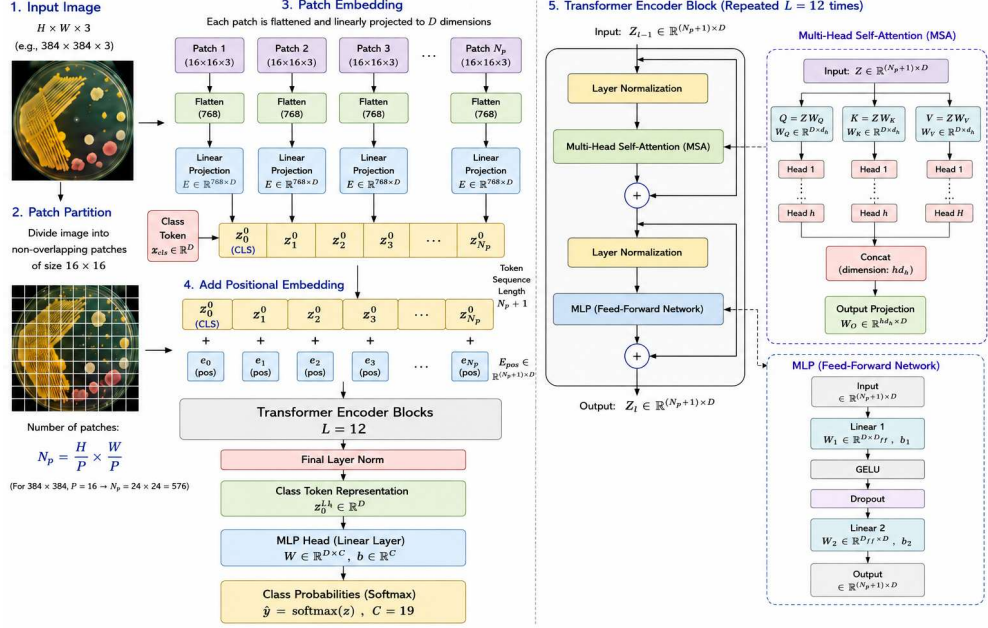


Fig. 5 ViT-B/16 architecture.

$$\mathbf{e}_i = \mathbf{x}_p^{(i)} \mathbf{E}, \quad (59)$$

where $\mathbf{E} \in \mathbb{R}^{P^2 C \times D}$ is the learnable patch-projection matrix and $\mathbf{e}_i \in \mathbb{R}^D$ is the embedding of the i th patch, and D denotes the dimensionality of the patch embedding space.

A learnable classification token, denoted by $\mathbf{x}_{\text{cls}}$, is prepended to the patch embeddings. Positional embeddings are then added to preserve the spatial arrangement of the patches. The initial transformer sequence is defined in Equation 60:

$$\mathbf{Z}_0 = [\mathbf{x}_{\text{cls}}; \mathbf{x}_p^{(1)} \mathbf{E}; \mathbf{x}_p^{(2)} \mathbf{E}; \dots; \mathbf{x}_p^{(N_p)} \mathbf{E}] + \mathbf{E}_{\text{pos}}, \quad (60)$$

where $\mathbf{Z}_0$ is the initial token sequence provided to the first transformer encoder, $\mathbf{x}_{\text{cls}}$ is a learnable classification token used to summarize the global image information, and $\mathbf{E}_{\text{pos}}$ is the learnable positional embedding matrix added to preserve the spatial order of the classification token and image patches.

The token sequence is processed through a stack of transformer encoder blocks. Each encoder block contains Layer Normalization (LN), Multi-head Self-Attention (MSA), a multilayer perceptron, and residual connections. For encoder block l , the attention stage is expressed in Equation 61:

$$\mathbf{Z}'_l = \text{MSA}(\text{LN}(\mathbf{Z}_{l-1})) + \mathbf{Z}_{l-1}, \quad (61)$$

where $\mathbf{Z}'_l$ is the intermediate token representation produced after the attention stage, $\mathbf{Z}_{l-1}$ denotes the input token sequence received from the previous transformer encoder block, $\text{LN}(\cdot)$ represents layer normalization applied before the attention operation, and $\text{MSA}(\cdot)$ denotes the multi-head self-attention mechanism.

followed by the feed-forward stage using Equation 62 [32]:

$$\mathbf{Z}_l = \text{MLP}(\text{LN}(\mathbf{Z}'_l)) + \mathbf{Z}'_l, \quad (62)$$

where $\mathbf{Z}_l$ is the final output sequence of the l th transformer encoder block, and $\text{MLP}(\cdot)$ represents the position-wise multilayer perceptron applied independently to each token.

Within each attention head, the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ matrices are generated using Equations 63, 64, and 65, respectively [33]:

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}_Q, \quad (63)$$

$$\mathbf{K} = \mathbf{Z}\mathbf{W}_K, \quad (64)$$

$$\mathbf{V} = \mathbf{Z}\mathbf{W}_V, \quad (65)$$

where $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ are learnable projection matrices.

The scaled dot-product attention is calculated using Equation 66:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \mathbf{V}, \quad (66)$$

where $\mathbf{K}^T$ denotes the transpose of the key matrix, $\mathbf{Q}\mathbf{K}^T$ computes the similarity scores between every query and all keys, $\sqrt{d_k}$ is a scaling factor that prevents the similarity scores from becoming excessively large, and $\text{Softmax}(\cdot)$ converts the scaled similarity scores into normalized attention weights.

The outputs of the individual attention heads are concatenated and projected using Equation 67 [34]:

$$\text{MSA}(\mathbf{Z}) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \mathbf{W}_O, \quad (67)$$

where $\text{MSA}(\mathbf{Z})$ is the final multi-head attention representation, head_i is the output of the i th attention head, h is the total number of attention heads, $\text{Concat}(\cdot)$ denotes concatenation of all head outputs along the feature dimension, and $\mathbf{W}_O$ is a learnable output-projection matrix.

After the final transformer encoder, the representation associated with the classification token is extracted using Equation 68:

$$\mathbf{g}_V = \text{LN}\left(\mathbf{Z}_L^{(\text{cls})}\right), \quad (68)$$

where L denotes the final encoder block and $\mathbf{g}_V$ represents the global image feature vector.

The original classification head was replaced with a task-specific fully connected layer containing 19 output neurons using Equation 69:

$$\mathbf{z}_V = \mathbf{W}_V^{\text{cls}} \mathbf{g}_V + \mathbf{b}_V, \quad \mathbf{z}_V \in \mathbb{R}^{19}, \quad (69)$$

where $\mathbf{W}_V^{\text{cls}}$ and $\mathbf{b}_V$ are the learnable classifier weights and biases, respectively, and $\mathbf{z}_V$ is the vector of class logits of ViT-B/16 model.

The class probabilities were obtained using the softmax function using Equation 70:

$$p_V(c | \mathbf{X}) = \frac{\exp(z_{V,c})}{\sum_{j=1}^{19} \exp(z_{V,j})}, \quad (70)$$

where $p_V(c | \mathbf{X})$ is the probability assigned by ViT-B/16 model to the bacterial class c .

The predicted bacterial species was then determined using Equation 71:

$$\hat{y}_V = \arg \max_{c \in \{1, \dots, 19\}} p_V(c | \mathbf{X}), \quad (71)$$

where $\hat{y}_V$ denotes the final prediction of the ViT-B/16 model.

2.6 Model Training Configuration

ResNet50, DenseNet121, and ViT-B/16 were trained under a unified experimental protocol to ensure a fair comparative analysis. The models were started off with pretrained weights, tuned to generate 19-class outputs, and all of them were optimized by means of the same loss function, optimizer, learning-rate strategy, number of epochs, and stratified five-fold cross-validation approach. For the ViT-B/16, small changes were made to the batch size since it demanded more memory, and gradient accumulation was used to maintain the same effective batch size as that used by the convolutional models. Table 2 shows the training configuration of the comparative models.

2.7 Performance Evaluation Metrics

After each cross-validation fold, the performance of ResNet50, DenseNet121, and ViT-B/16 was assessed by means of a full range of multiclass classification metrics. Accuracy, precision, recall, the F1-score, and the confusion matrices were computed, and macro- and weighted-averaged values were also given; the final performance of each model was then summarised by means of the mean and standard deviation obtained from the five folds.

A multiclass Confusion Matrix (CM) was generated for each cross-validation fold to summarize the classification performance across the bacterial species:

$$\mathbf{CM} \in \mathbb{N}^{C \times C}.$$

Each matrix element is defined in Equation 72 [35]:

$$CM_{i,j} = \sum_{n=1}^N \mathbb{I}(y_n = i \wedge \hat{y}_n = j), \quad (72)$$

Table 2 Training configuration of the comparative models.

Training parameter	ResNet50	DenseNet121	ViT-B/16
Input size	384×384	384×384	384×384
Number of classes	19	19	19
Pretrained weights	ImageNet	ImageNet	ImageNet-21K / ImageNet-1K
Cross-validation	Stratified 5-fold	Stratified 5-fold	Stratified 5-fold
Number of epochs	10	10	10
Physical batch size	16	16	16
Gradient accumulation steps	1	1	2
Effective batch size	16	16	16
Loss function	Cross-entropy	Cross-entropy	Cross-entropy
Optimizer	AdamW	AdamW	AdamW
Initial learning rate	1×10^{-4}	1×10^{-4}	1×10^{-4}
Final learning-rate factor	0.01	0.01	0.01
Weight decay	5×10^{-4}	5×10^{-4}	5×10^{-4}
Warm-up epochs	3	3	3
Mixed-precision training	Enabled	Enabled	Enabled
Random seed	42	42	42
Data-loading workers	2	2	2

where y_n is the true label of the n th image, $\hat{y}_n$ is its predicted label, N is the total number of evaluated images, and $\mathbb{I}(\cdot)$ is an indicator function that returns 1 when the stated condition is true and 0 otherwise.

The rows of the CM represent the actual bacterial classes, while the columns represent the predicted classes. The diagonal elements $CM_{i,i}$ indicate correct classifications, whereas off-diagonal elements represent misclassifications between different bacterial species.

The CM components for an individual class c , are derived using Equations 73, 74, 75, and 76:

$$TP_c = CM_{c,c}, \quad (73)$$

$$FN_c = \sum_{\substack{j=1 \\ j \neq c}}^C CM_{c,j}, \quad (74)$$

$$FP_c = \sum_{\substack{i=1 \\ i \neq c}}^C CM_{i,c}, \quad (75)$$

$$TN_c = \sum_{\substack{i=1 \\ i \neq c}}^C \sum_{\substack{j=1 \\ j \neq c}}^C CM_{i,j}. \quad (76)$$

where TP_c , FN_c , FP_c , and TN_c denote the true positives, false negatives, false positives, and true negatives for class c , respectively. These quantities were subsequently used to calculate the class-specific and overall evaluation metrics. Table 3 shows the evaluation metrics and their formulas [36–38].

Table 3 The evaluation metrics and their formulas.

Metric	Mathematical formula
Accuracy	$\text{Accuracy} = \frac{\sum_{c=1}^C TP_c}{N}$
Precision	$P_c = \frac{TP_c}{TP_c + FP_c}$
Recall	$R_c = \frac{TP_c}{TP_c + FN_c}$
F1-Score	$F1_c = 2 \frac{P_c R_c}{P_c + R_c}$
Macro Precision	$P_{\text{macro}} = \frac{1}{C} \sum_{c=1}^C P_c$
Macro Recall	$R_{\text{macro}} = \frac{1}{C} \sum_{c=1}^C R_c$
Macro F1-Score	$F1_{\text{macro}} = \frac{1}{C} \sum_{c=1}^C F1_c$
Weighted Precision	$P_{\text{weighted}} = \sum_{c=1}^C \frac{n_c}{N} P_c$
Weighted Recall	$R_{\text{weighted}} = \sum_{c=1}^C \frac{n_c}{N} R_c$
Weighted F1-Score	$F1_{\text{weighted}} = \sum_{c=1}^C \frac{n_c}{N} F1_c$
Five-Fold Mean	$\bar{M} = \frac{1}{K} \sum_{k=1}^K M_k$
Five-Fold Standard Deviation	$\sigma_M = \sqrt{\frac{1}{K-1} \sum_{k=1}^K (M_k - \bar{M})^2}$

where C is the number of classes, N is the total number of evaluated images, n_c is the number of images belonging to class c , K denotes the total number of cross-validation folds, and M_k represents the value of metric M obtained from the k th fold.

3 Results and Discussion

3.1 Overall Cross-Validation Performance

The performance of ResNet50, DenseNet121, and ViT-B/16 was evaluated using stratified five-fold cross-validation. In each iteration, four folds were used for training and the remaining fold was held out for evaluation. The same fold assignments were used for the three models to ensure a consistent comparison. Table 4 summarizes the mean and standard deviation of the principal evaluation metrics across the five folds.

Because the dataset contained an equal number of images for each bacterial species and stratification preserved the class proportions across folds, the macro- and weighted-averaged metrics were nearly identical. Therefore, only the macro-averaged measures are reported to avoid redundant information.

Table 4 Overall five-fold cross-validation performance of the evaluated models (mean $\pm$ SD).

Model	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)	MCC	Cohen's κ
ResNet50	94.21 $\pm$ 1.75	95.11 $\pm$ 1.61	94.21 $\pm$ 1.75	94.19 $\pm$ 1.75	0.9395 $\pm$ 0.0183	0.9389 $\pm$ 0.0184
DenseNet121	97.58 $\pm$ 1.69	97.97 $\pm$ 1.43	97.58 $\pm$ 1.69	97.56 $\pm$ 1.68	0.9747 $\pm$ 0.0177	0.9744 $\pm$ 0.0178
ViT-B/16	99.05 $\pm$ 0.86	99.17 $\pm$ 0.75	99.05 $\pm$ 0.86	99.05 $\pm$ 0.86	0.9901 $\pm$ 0.0091	0.9900 $\pm$ 0.0091

ViT-B/16 achieved the best overall performance, with a mean accuracy of 99.05% and a macro F1-score of 99.05%. DenseNet121 ranked second, achieving 97.58% accuracy and a macro F1-score of 97.56%, whereas ResNet50 achieved 94.21% accuracy and a macro F1-score of 94.19%.

The difference in mean accuracy between ViT-B/16 and DenseNet121 was 1.47 percentage points, while ViT-B/16 exceeded ResNet50 by 4.84 percentage points. DenseNet121 also outperformed ResNet50 by 3.37 percentage points. A similar ranking was observed for precision, recall, F1-score, MCC, and Cohen's κ , indicating that the performance differences were consistent across multiple evaluation criteria rather than being restricted to accuracy alone.

The fact that ViT-B/16 achieved high MCC and κ values, both about 0.99, once again shows that there was strong agreement between the predicted and the actual bacterial species; DenseNet121 also demonstrated strong agreement, while the lower MCC and κ values of ResNet50 were in line with its worse classification performance.

3.2 Fold-Wise Performance and Stability

The variation across individual folds was examined to evaluate model stability. Table 5 presents the accuracy obtained in each cross-validation fold.

Table 5 Fold-wise classification accuracy (%) of the three models.

Fold	ResNet50	DenseNet121	ViT-B/16
Fold 1	96.84	99.47	99.47
Fold 2	92.63	97.37	100.00
Fold 3	94.74	98.95	98.42
Fold 4	94.21	96.84	99.47
Fold 5	92.63	95.26	97.89
Mean $\pm$ SD	94.21 $\pm$ 1.75	97.58 $\pm$ 1.69	99.05 $\pm$ 0.86

ViT-B/16 demonstrated the lowest variability across folds, achieving accuracy values between 97.89% and 100.00%, with a standard deviation of 0.86%. In contrast,

DenseNet121 and ResNet50 exhibited standard deviations of 1.69% and 1.75%, respectively. ViT-B/16 achieved perfect classification in Fold 2 and demonstrated the highest accuracy in Folds 2, 4, and 5. In Fold 1, ViT-B/16 and DenseNet121 both achieved an accuracy of 99.47%. DenseNet121 outperformed ViT-B/16 only in Fold 3, with accuracies of 98.95% and 98.42%, respectively.

The fact that a smaller variation was seen with ViT-B/16 indicates that its performance was less dependent on the specific way the training and validation sets were divided. DenseNet121 likewise demonstrated consistently good results and stayed above 95% accuracy in all five folds. ResNet50 kept its accuracy above 92% in each fold but always showed lower performance than the other two architectures.

3.3 Exploratory Statistical Comparison

An exploratory statistical analysis was performed to assess whether the differences observed across the five folds exceeded the fold-to-fold variation. The three models were first compared using the Friedman test, and subsequent pairwise comparisons were conducted using paired t-tests. The standardized paired effect size was quantified using Cohen’s d_z .

The Friedman test indicated an overall difference among the three models ($\chi^2(2) = 8.32$, $p = 0.016$). The mean ranks were 1.30 for ViT-B/16, 1.70 for DenseNet121, and 3.00 for ResNet50, where a lower rank indicates better performance. Given that three pairwise comparisons were conducted, the Holm adjustment was applied to the corresponding p-values.

Table 6 Exploratory pairwise comparison of fold-wise classification accuracy.

Comparison	Mean difference (pp)	t	Raw p -value	Holm-adjusted p -value	Cohen’s d_z
ViT-B/16 vs. ResNet50	+4.84	6.01	0.004	0.008	2.69
DenseNet121 vs. ResNet50	+3.37	7.34	0.002	0.006	3.28
ViT-B/16 vs. DenseNet121	+1.47	2.06	0.108	0.108	0.92

Both ViT-B/16 and DenseNet121 showed significantly higher fold-wise accuracy than ResNet50 after adjustment for multiple pairwise comparisons. The associated effect sizes were also large.

In contrast, no statistically significant difference was detected between ViT-B/16 and DenseNet121 ($p = 0.108$), despite ViT-B/16 achieving a higher mean accuracy and lower standard deviation. Therefore, the present analysis supports a clear difference between ResNet50 and the two higher-performing architectures, whereas the difference between ViT-B/16 and DenseNet121 remains inconclusive.

These inferential findings should nevertheless be interpreted cautiously. Only five paired fold estimates were available, and cross-validation folds are not fully independent observations. Consequently, the statistical tests are used here as supportive analyses rather than definitive evidence of superiority.

3.4 Error and Confusion Analysis

To examine the nature of the classification errors, the out-of-fold predictions from the five validation folds were aggregated. As a result, each of the 950 images was used exactly once in the final error analysis for each model.

ResNet50 misclassified 55 images, resulting in an error rate of 5.79%. DenseNet121 decreased the number of errors to 23 (2.42%), whereas ViT-B/16 yielded only 9 incorrect predictions (0.95%).

Table 7 Dominant misclassification patterns aggregated across the five folds.

Model	Total errors / 950	Dominant confusions (true → predicted : number of cases)
ResNet50	55	sep → cst: 9; aba → sma: 8; enc → mmo: 7; eca → spy: 6; sep → efa: 5; sau → cst: 4; spy → eca: 3
DenseNet121	23	sep → cst: 8; aba → sma: 5; sau → sep: 3
ViT-B/16	9	aba → sma: 3; all other confusion pairs occurred once or less

Two notable patterns emerged. The misclassification of *Acinetobacter baumannii* as *Serratia marcescens* (aba → sma) was observed across all three models. Specifically, this error occurred eight times with ResNet50, five times with DenseNet121, and three times with ViT-B/16. The persistence of the same confusion across different architectures may indicate overlapping visual characteristics under the culture and imaging conditions used in this dataset. However, determining the underlying biological or morphological reason would require a separate targeted analysis.

Second, the sep → cst confusion was prominent in both convolutional models, occurring nine times with ResNet50 and eight times with DenseNet121, but it was not observed among the ViT-B/16 errors. This finding suggests that ViT-B/16 was better able to distinguish these two classes in the present dataset. One possible explanation is that the transformer’s global representation may incorporate discriminative information distributed across different regions of the plate. Nevertheless, the present experiments do not directly isolate self-attention as the causal mechanism, and this interpretation should therefore remain tentative.

3.5 Comparison of the Convolutional and Transformer Architectures

The experimental results showed a consistent descriptive ranking of ViT-B/16, DenseNet121, and ResNet50. Among the convolutional architectures, DenseNet121 consistently outperformed ResNet50. Dense connectivity allows earlier feature representations to be reused throughout the network, which may help retain fine-grained texture and morphological information. In the present experiments, this architecture achieved a 3.37-percentage-point improvement in mean accuracy over ResNet50.

ViT-B/16 achieved the highest mean accuracy, macro F1-score, MCC, and κ , while also producing the lowest cross-fold variability. Unlike the convolutional models, ViT-B/16 divides the input image into patches and models relationships among these patches using self-attention. This representation may be advantageous for complete Petri-dish images because discriminative information is not necessarily confined to a single local region. Colony distribution, texture, size, color, boundaries, and overall plate appearance may jointly contribute to species recognition.

The confusion analysis was consistent with this interpretation, particularly for the *sep* $\rightarrow$ *cst* class pair. However, the results alone do not establish that self-attention is directly responsible for the improvement. Differences in model capacity, pretrained representations, and optimization behavior may also contribute.

Another practical point to consider is computational complexity. Since ViT-B/16 has a much larger number of parameters than DenseNet121 and ResNet50, it follows that even though it achieves the best predictive performance, DenseNet121 could still be a worthwhile alternative in cases where memory usage, inference speed, or computational resources are limited.

3.6 Comparison with Previous Studies

Table 8 presents a comparison among the current study and representative previous approaches for bacterial and microbial classification. While several studies have reported high accuracies, direct comparisons require caution due to variations in the number of classes, image types, dataset characteristics, and validation protocols.

The present study achieved competitive performance compared with the selected literature. ViT-B/16 obtained a mean accuracy of 99.05%, while DenseNet121 and ResNet50 achieved 97.58% and 94.21%, respectively. Earlier approaches based on SVM, hybrid MobileNet–XGBoost, and sensing-array-based classification reported accuracies ranging from approximately 84% to 93.3% [1, 8, 12], whereas the CNN-based multiclass approach of Khalifa et al. [11] achieved 98.22%.

More recent studies have reported accuracy close to 99%. Sarmis et al. [9] achieved 99% using YOLOv12; however, their task involved discrimination between only *E. coli* and *K. pneumoniae* and did not employ cross-validation. Ahmed et al. [39] reported a slightly higher accuracy of 99.34% using ShuffleNetV2, but their evaluation was based on isolated colony crops and a fixed 70/15/15 data split. In contrast, the present study directly analyzes complete culture-plate images from 19 bacterial species and evaluates all three architectures using the same stratified five-fold cross-validation protocol.

Therefore, the main strength of the present study is not solely the achieved accuracy of 99.05%, but the combination of a relatively broad 19-class whole-plate classification task, systematic five-fold evaluation, and detailed analysis across overall, class-wise, and error-based performance measures. These results demonstrate the potential of ViT-B/16 for multiclass bacterial identification while also showing that DenseNet121 provides a strong convolutional alternative under the same experimental conditions.

Table 8 Comparison with representative previous studies.

Authors	Year	Microbes Name	Models	Accuracy
Rodrigues et al. [1]	2023	<i>Pseudomonas aeruginosa</i> vs. <i>Staphylococcus aureus</i>	Hybrid model	92%
		<i>Escherichia coli</i> vs. <i>S. aureus</i>	MobileNet-XGBoost	91%
		<i>Escherichia coli</i> vs. <i>P. aeruginosa</i>		84%
Kim et al. [8]	2021	Five species: <i>E. coli</i> , <i>S. aureus</i> , <i>Salmonella enterica</i> Serovar Typhimurium, <i>Enterococcus faecium</i> , and <i>P. aeruginosa</i>	SVM	93.3%
Khalifa al. [11]	2019	33 bacterial classes	CNN	98.22%
Zhijun Li [12]	2022	Eight microbial classes including <i>E. coli</i> , MRSA, <i>P. aeruginosa</i> , <i>Enterococcus faecalis</i> , <i>Klebsiella pneumoniae</i> , and <i>Candida albicans</i>	2D nanomaterial sensing array + LDA	92%
Sarmis al. [9]	2026	<i>E. coli</i> and <i>K. pneumoniae</i>	YOLOv12 model	99%
Naito al. [2]	2026	10 microbial species; colony crops + culture-plate tiles	ResNet-50	92%
Ahmed al. [39]	2026	Microbial species classification from individual AGAR colony crops	ShuffleNetV2	99.34%
Proposed study	2026	19 clinically important bacterial species using complete culture-plate images	ResNet50, DenseNet121, and ViT-B/16	99.05% (ViT-B/16)

3.7 Limitations

Several limitations should be considered when interpreting the results. First, cross-validation was performed at the image level, so images from the same clinical strain may appear in both training and held-out folds. Therefore, the reported performance does not necessarily reflect generalization to unseen strains, and future work should consider strain-grouped validation.

Second, all images were acquired under standardized laboratory and imaging conditions; external validation across different devices, laboratories, and culture settings is still required. Third, the study benchmarks established pretrained architectures rather than proposing a new network design. Finally, the statistical analysis was based on only five fold-level observations, which limits inferential power. Future work should therefore focus on strain-independent evaluation, external datasets, and broader computational-efficiency analysis.

4 Conclusion

This study evaluated three pretrained DL architectures (ResNet50, DenseNet121, and ViT-B/16) for multiclass identification of bacterial species from complete culture-plate

images. The experiments were conducted on a dataset of 950 images representing 19 clinically relevant bacterial species utilizing stratified five-fold cross-validation under a unified training and evaluation protocol. Among the evaluated models, ViT-B/16 achieved the strongest overall performance, with a mean accuracy and macro F1-score of 99.05%, together with MCC and Cohen’s kappa values close to 0.99. DenseNet121 achieved 97.58% accuracy, while ResNet50 reached 94.21%. ViT-B/16 also showed the lowest cross-fold variability and produced only nine misclassifications across the aggregated out-of-fold predictions. Class-wise analysis further showed that most bacterial species were identified reliably, while the remaining errors were concentrated in a limited number of visually challenging class pairs. Overall, the findings demonstrate that complete culture-plate images can support highly accurate automated bacterial classification without requiring prior extraction of individual colonies. The results also highlight the potential of transformer-based representations for capturing discriminative plate-level visual patterns. Future work should directly address the current study limitations by adopting strain-grouped or leave-strain-out validation, evaluating the models on independent multi-centre datasets acquired with different imaging systems and laboratory conditions, and conducting a more detailed analysis of computational cost, inference time, and model efficiency. Further investigation of recurrent confusion patterns may also help identify the morphological features responsible for the remaining classification errors.

Declarations

Funding. None.

Conflict of interest. The authors declare that they have no conflict of interest.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. The dataset used in this study is publicly available through the Zenodo repository at <https://doi.org/10.5281/zenodo.18256864>.

Materials availability. Not applicable.

Code availability. The code used in this study is available from the corresponding author upon reasonable request.

Author contributions. A.A.A. contributed to conceptualization, investigation, and supervision. H.S.A. contributed to methodology, validation, and interpretation of the microbiological aspects of the study. A.K.K. contributed to conceptualization, methodology, software implementation, model development, data analysis, visualization, and writing of the original draft. M.N.O. contributed to data curation, validation, formal analysis, and manuscript review and editing. A.A.N. contributed to methodology, result interpretation, validation, and manuscript review and editing. All authors reviewed and approved the final version of the manuscript.

References

- [1] Rodrigues, P.M., Ribeiro, P., Tavaría, F.K.: Distinction of different colony types by a smart-data-driven tool. *Bioengineering* **10**(1) (2023) <https://doi.org/10.3390/bioengineering10010026>
- [2] Naito, T., Takei, S., Misawa, S., Kuribara, M., Tabe, Y.: Dual-input deep learning system for microbial identification from blood agar plates. *PLoS One* **21**(7), 0353761 (2026) <https://doi.org/10.1371/journal.pone.0353761>
- [3] Leber, A.L., Peterson, E., Dien Bard, J.: Personnel standards and workforce subcommittee, american society for microbiology. the hidden crisis in the times of COVID-19: Critical shortages of medical laboratory professionals in clinical microbiology. *J. Clin. Microbiol.* **60**(8), 0024122 (2022) <https://doi.org/10.1128/jcm.00241-22>
- [4] Krishnamurthy, P., Shetty, R., Isaac, K.P., Unni, S.N., Rao, S.S., Parthasarathy, K.: Differentiation of bacterial species in liquid culture using laser speckle contrast imaging. *J. Biophotonics* **18**, 202400565 (2025) <https://doi.org/10.1002/jbio.202400565>
- [5] Sunanda, B., Ramesh Babu, D.R., Arnepalli, Y., Mohith, B.K., Durga Bhavesh, C.H., Jha, P.: AI-driven bacterial identification: A hybrid deep learning and machine learning framework. In: *Proceedings of 2025 International Conference on Emerging Technologies in Computing and Communication (ETCC 2025)* (2025). <https://doi.org/10.1109/ETCC65847.2025.11108526>
- [6] Wu, Y., Gadsden, S.A.: Machine learning algorithms in microbial classification: a comparative analysis. *Front. Artif. Intell.* **6**, 1200994 (2023) <https://doi.org/10.3389/frai.2023.1200994>
- [7] Ebisawa, K.F., *et al.*: Comparison of gram stain interpretation accuracy between a computer-aided diagnosis app and microbiology specialists in blood culture samples. *Microbiol. Spectr.* (2026) <https://doi.org/10.1128/spectrum.00081-26>
- [8] Kim, S., *et al.*: Human sensor-inspired supervised machine learning of smartphone-based paper microfluidic analysis for bacterial species classification. *Biosens. Bioelectron.* **188**, 113335 (2021) <https://doi.org/10.1016/j.bios.2021.113335>
- [9] Sarmis, A., Ustebay, S., Mutlu, M.A., Canbaz, F.A., Kaya, G.K.: Deep learning-based rapid identification of escherichia coli and klebsiella pneumoniae from chromogenic agar urine cultures using YOLOv12. *Risk Manag. Healthc. Policy* **19**, 1–14 (2026) <https://doi.org/10.2147/RMHP.S561761>
- [10] Tsang, C.C., *et al.*: Automatic identification of clinically important aspergillus species by artificial intelligence-based image recognition: proof-of-concept study.

- Emerg. Microbes Infect. **14**(1) (2025) <https://doi.org/10.1080/22221751.2024.2434573>
- [11] Khalifa, N.E.M., Taha, M.H.N., Hassanien, A.E., Hemedan, A.A.: Deep bacteria: Robust deep learning data augmentation design for limited bacterial colony dataset. *Int. J. Reason.-Based Intell. Syst.* **11**, 256 (2019) <https://doi.org/10.1504/IJRIS.2019.102610>
 - [12] Li, Z., *et al.*: 2d nanomaterial sensing array using machine learning for differential profiling of pathogenic microbial taxonomic identification. *Microchim. Acta* **189**, 273 (2022) <https://doi.org/10.1007/s00604-022-05368-5>
 - [13] Kong, F., *et al.*: Implementation of convolutional neural networks for microbial colony recognition. *Microbiol. Spectr.* **13**(9) (2025) <https://doi.org/10.1128/spectrum.02885-24>
 - [14] Du, J., *et al.*: High-resolution colony images of clinically isolated bacteria for automated detection and deep learning. *Scientific Data* **13**(1) (2026) <https://doi.org/10.1038/s41597-026-07095-5>
 - [15] Huang, G.H., *et al.*: Deep convolutional neural networks on multiclass classification of three-dimensional brain images for parkinson’s disease stage prediction. *Journal of Imaging Informatics in Medicine* **38**(5), 2934–2950 (2025) <https://doi.org/10.1007/s10278-025-01402-z>
 - [16] Lee, J.C., Byeon, K., Song, B., Kim, K., Kwak, J.T.: DIOR-ViT: Differential ordinal learning vision transformer for cancer classification in pathology images. *Med. Image Anal.* **105** (2025) <https://doi.org/10.1016/j.media.2025.103708>
 - [17] Morales, J., *et al.*: Visual analysis of deep learning semantic segmentation applied to petrographic thin sections. *Sci. Rep.* **15**(1) (2025) <https://doi.org/10.1038/s41598-025-99767-2>
 - [18] Öznacar, T., Varol Kayapunar, N.: Advanced skin cancer prediction with medical image data using MobileNetV2 deep learning and optimized techniques. *Sci. Rep.* **15**(1) (2025) <https://doi.org/10.1038/s41598-025-14963-4>
 - [19] Meva, D.T., Popat, K., Kukadiya, H., Jariwala, N.: Deep learning-based breast cancer detection with customized ensemble attention. *Sci. Rep.* **16**(1) (2026) <https://doi.org/10.1038/s41598-026-50563-6>
 - [20] Eisentraut, L., Mai, C., Hosch, J., Benecke, A., Penava, P., Buettner, R.: Deep learning-based detection of tuberculosis using a gaussian chest x-ray image filter as a software lens. *IEEE Access* **13**, 36065–36081 (2025) <https://doi.org/10.1109/ACCESS.2025.3544923>
 - [21] Islam, T., Hafiz, M.S., Jim, J.R., Kabir, M.M., Mridha, M.F.: A systematic review

- of deep learning data augmentation in medical imaging: Recent advances and future research directions. *Healthcare Analytics* **5**, 100340 (2024) <https://doi.org/10.1016/j.health.2024.100340>
- [22] Aurnob, A.R., Shrestha, T.E., Al Huda, M.S., Tamim, S.A., Arman, R.A., Ali, M.A.: DeepResVit: A hybrid deep learning approach for ovarian cancer classification with XAI. In: 2024 2nd International Conference on Information and Communication Technology (ICICT), pp. 229–233 (2024). <https://doi.org/10.1109/ICICT64387.2024.10839719>
 - [23] Zhou, Z., Zhang, H., Effatparvar, M.: Improved sports image classification using deep neural network and novel tuna swarm optimization. *Sci. Rep.* **14**(1) (2024) <https://doi.org/10.1038/s41598-024-64826-7>
 - [24] Wang, S.: Evaluation of impact of image augmentation techniques on two tasks: Window detection and window states detection. *Results in Engineering* **24** (2024) <https://doi.org/10.1016/j.rineng.2024.103571>
 - [25] Ktena, I., *et al.*: Generative models improve fairness of medical classifiers under distribution shifts. *Nat. Med.* **30**(4), 1166–1173 (2024) <https://doi.org/10.1038/s41591-024-02838-6>
 - [26] Xu, W., Fu, Y.-L., Zhu, D.: ResNet and its application to medical image processing: Research progress and challenges. *Comput. Methods Programs Biomed.* **240**, 107660 (2023) <https://doi.org/10.1016/j.cmpb.2023.107660>
 - [27] Dai, H., Yang, Y., Yue, X., Chen, S.: Improving retinal OCT image classification accuracy using medical pre-training and sample replication methods. *Biomed. Signal Process. Control* **91**, 106019 (2024) <https://doi.org/10.1016/j.bspc.2024.106019>
 - [28] Kumar, Y., *et al.*: Automating cancer diagnosis using advanced deep learning techniques for multi-cancer image classification. *Sci. Rep.* **14**(1) (2024) <https://doi.org/10.1038/s41598-024-75876-2>
 - [29] Rahimnezhad, A., Rezaei, S.M., Bayat, M.P., Heydarheydari, S.: Hybrid deep learning models for automatic segmentation and classification of breast lesions in ultrasound images. *Egyptian Journal of Radiology and Nuclear Medicine* **56**(1) (2025) <https://doi.org/10.1186/s43055-025-01659-y>
 - [30] Kehkashan, T., *et al.*: Gastroenterological disease detection using transformer-based medical imaging for sustainable healthcare. *Sci. Rep.* **16**(1) (2026) <https://doi.org/10.1038/s41598-026-45222-9>
 - [31] Kittichai, V., *et al.*: Enhance fashion classification of mosquito vector species via self-supervised vision transformer. *Sci. Rep.* **14**(1) (2024) <https://doi.org/10.1038/s41598-024-83358-8>

- [32] Papa, L., Russo, P., Amerini, I., Zhou, L.: A survey on efficient vision transformers: Algorithms, techniques, and performance benchmarking. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(12), 7682–7700 (2024) <https://doi.org/10.1109/TPAMI.2024.3392941>
- [33] Sun, W., Ma, Y., Wang, R.: k-NN attention-based video vision transformer for action recognition. *Neurocomputing* **574**, 127256 (2024) <https://doi.org/10.1016/j.neucom.2024.127256>
- [34] Hu, X., Zhu, S., Peng, T.: Hierarchical attention vision transformer for fine-grained visual classification. *J. Vis. Commun. Image Represent.* **91**, 103755 (2023) <https://doi.org/10.1016/j.jvcir.2023.103755>
- [35] Krstinić, D., Skelin, A.K., Slapničar, I., Braović, M.: Multi-label confusion tensor. *IEEE Access* **12**, 9860–9870 (2024) <https://doi.org/10.1109/ACCESS.2024.3353050>
- [36] Rainio, O., Teuho, J., Klén, R.: Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **14**(1) (2024) <https://doi.org/10.1038/s41598-024-56706-x>
- [37] Tharwat, A., Jaster, B., Schenck, W., Kohlhase, M.: Active learning evaluation metrics for classification and regression frameworks. *Engineering Applications of Artificial Intelligence* (2026) <https://doi.org/10.1016/j.engappai.2026.114295>
- [38] Kocak, B., *et al.*: Evaluation metrics in medical imaging AI: fundamentals, pitfalls, misapplications, and recommendations. *European Journal of Radiology Artificial Intelligence* **3**, 100030 (2025) <https://doi.org/10.1016/j.ejrai.2025.100030>
- [39] Ahmed, M.A., Alenazy, R., AbdelMoety, A., Kamel, H.A., Shafik, S.M.: Comparative evaluation of deep learning architectures for microbial colony classification in microbiological imaging. *BMC Microbiol.* **26**(1) (2026) <https://doi.org/10.1186/s12866-026-05028-1>