Karbala International Journal of Modern Science

Manuscript 3379

Comparison of Efficient Deep Learning Architectures for Lactobacillus Species Identification

Dea Aisyah Rusmawati

Ishak Ariawan

Afrinal Firmanda

Comparison of Efficient Deep Learning Architectures for Lactobacillus Species Identification

Abstract

Identifying microorganism species, such as Lactobacillus, is essential in ensuring the food products' quality and safety. Traditional laboratory practice requires expert knowledge and experience, but the method is expensive and time-consuming due to complex sample preparation. Faster, more accurate, and cheaper computational methods, such as transfer learning technology, are needed for the Lactobacillus species classification. The technique has been effective in a variety of image recognition contexts. Deep learning architecture can also be applied as an innovative strategy for digital image-based identification. Therefore, this research aims to compare several deep-learning architectures in classifying bacterial strains of Lactobacillus. The four architectures (Inception V3, MNASNet, RegNet, and Xception) used have excellent performance with accuracy above 95%. Among those architectures, the mobile neural architecture search network (MNASNet) exhibits the most potential, with 99.15% accuracy, 99.09% precision, 99.14% sensitivity, and 99.11% F1 score. This performance is particularly notable given that MNASNet operates with 3.1 million parameters. Although RegNet has a slightly lower parameter count at 2.68 million, it achieves the accuracy, precision, sensitivity, and F1 score of 98.73%, 98.73%, 98.77%, and 98.75%, respectively. Xception, with over 20 million parameters, attains 98.99% accuracy, 98.71% precision, 98.78% sensitivity, and 98.74% F1 score. In addition, the MNASNet architecture is highly efficient, making it very suitable to be implemented or embedded on mobile devices. The model is promising regarding the deep learning architecture prospects for classifying microscopic images of Lactobacillus species.

Keywords

Classification of Lactobacillus; Deep Learning; Identification; Inception V3; MNASNet; RegNet; Xception

Creative Commons License

RESEARCH PAPER

Comparison of Efficient Deep Learning Architecture for *Lactobacillus* Species Identification

Dea A. Rusmawati [a,*], Ishak Ariawan [b], Afrinal Firmanda [c]

[a] Department of Food Technology, University of Sultan Ageng Tirtayasa, Serang, Banten, Indonesia
[b] Marine Information System, Universitas Pendidikan Indonesia, West Java 40154, Indonesia
[c] Department of Chemical Engineering, Faculty of Engineering, University of Indonesia, Depok, 16424, Indonesia

Abstract

Identifying microorganism species, such as *Lactobacillus*, is essential in ensuring the food products' quality and safety. Traditional laboratory practice requires expert knowledge and experience, but the method is expensive and time-consuming due to complex sample preparation. Faster, more accurate, and cheaper computational methods, such as transfer learning technology, are needed for the *Lactobacillus* species classification. The technique has been effective in a variety of image recognition contexts. Deep learning architecture can also be applied as an innovative strategy for digital image-based identification. Therefore, this research aims to compare several deep-learning architectures in classifying bacterial strains of *Lactobacillus*. The four architectures (Inception V3, MNASNet, RegNet, and Xception) used have excellent performance with accuracy above 95 %. Among those architectures, the mobile neural architecture search network (MNASNet) exhibits the most potential, with 99.15 % accuracy, 99.09 % precision, 99.14 % sensitivity, and 99.11 % F1 score. This performance is particularly notable given that MNASNet operates with 3.1 million parameters. Although RegNet has a slightly lower parameter count at 2.68 million, it achieves the accuracy, precision, sensitivity, and F1 score of 98.73 %, 98.73 %, 98.77 %, and 98.75 %, respectively. Xception, with over 20 million parameters, attains 98.99 % accuracy, 98.71 % precision, 98.78 % sensitivity, and 98.74 % F1 score. In addition, the MNASNet architecture is highly efficient, making it very suitable to be implemented or embedded on mobile devices. The model is promising regarding the deep learning architecture prospects for classifying microscopic images of *Lactobacillus* species.

Keywords: Classification of *Lactobacillus*, Deep learning, Identification, Inception V3, MNASNet, RegNet, Xception

1. Introduction

The digital revolution has arrived, and development experts have a long way to go to solve the challenge of global food security. Nowadays, people not only need food to fulfill basic nutritional requirements but also look for additional functions for health [1]. Hence, functional foods are trending and answer to the demand [2]. One of the most important lactic acid bacteria, *Lactobacillus* [3]. This bacteria also has nutrition and is beneficial for health [2]. *Lactobacillus* play a vital role in the fermentation of several dairy products [4]. It is used commercially in producing multiple products, like yogurt and kefir, partially due to its ability to release lactic acid and improve specific sensory characteristics [5].

Beyond food production, *Lactobacillus* species are being explored for their potential in various industries, e.g., pharmaceutical, nutraceutical, and agriculture. In the pharmacy, these bacteria are used to develop probiotic supplements to prevent and treat various gastrointestinal disorders [6]. In agriculture, *Lactobacillus* enhance plant growth and protect crops against pathogens by producing bioactive compounds [7]. Their multifunctionality highlighted the importance of accurately identifying *Lactobacillus* species to ensure the quality and safety

Received 1 July 2024; revised 15 October 2024; accepted 17 October 2024.
Available online 9 November 2024

* Corresponding author.
E-mail address: dea@untirta.ac.id (D.A. Rusmawati).

Peer review under responsibility of University of Kerbala.

https://doi.org/10.33640/2405-609X.3379

of food products and the efficacy of probiotic treatments and agricultural applications [8].

Traditional laboratory methods for identifying the species require experts with experience and knowledge. Conversely, most automatic and rapid microbiological identification methods are based on biochemical or modular biological technologies [9,10]. The methods are expensive and time-consuming due to the requirements of complex samples [11]. Using digital images as information opens new opportunities to optimize the identification of *Lactobacillus* species.

Automating the identification process is very promising in bioimage informatics. The strategy is reasonable for particular image analysis, e.g., motion analysis, morphometric features, or object detection [12]. Most image analysis algorithms have been expanded for particular biological assays [13]. More precise, less expensive, and faster computational methods for classifying species are needed. Therefore, accurate and efficient *Lactobacillus* classification ensures quality and food sustainability.

Transfer learning technology using deep architecture has been proven effective in various image recognition [14]. The technology can be applied as an innovative method in the digital image-based identification of *Lactobacillus* species. Transfer learning can improve accuracy and efficiency by leveraging existing knowledge from models trained on image datasets [15]. Identifying *Lactobacillus* species is based on microscopic images.

Several research have adopted deep learning architecture to classify bacterial species using microscopic images. Zielinski et al. [16] used a deep-learning method for classifying bacterial colonies. Convolutional neural networks (CNN) were used as a feature extractor, support vector machine, and random forest classifier. In 2018, Nasip and Kenan [17] used another method to solve digital image of bacterial species (DIBaS) classification tasks. The two different architectures classified bacteria using a VGGNet [18] and AlexNet [19] to achieve accuracies of 98.25 % and 97.53 %, respectively. Subsequently, three works using the DIBaS dataset were published in 2019. The first achieved 98.22 % classification accuracy with a deep CNN and a data augmentation method known as "deep bacteria" [20]. In the second publication, "deep bacteria" was exceeded by achieving 99.2 % classification accuracy and retuning the trained model from ResNet-50 [21]. For the third publication, Rujichan et al. [22] achieved a classification accuracy of up to 95.09 % by returning a previously trained version of the MobileNet V2 architecture Sandler et al. [23], and adopting several augmentations.

A classification accuracy of 98.68 % was achieved using fractional sequential orthogonal moments to extract features and propose a new selection method called salp swarm algorithm + teaching-based learning optimization (SSATLBO) [24]. Meanwhile, Satoto et al. [25] also achieved 98.59 % accuracy using a custom CNN topology and data augmentation methods. García et al. [15] adopted 4 different architectures using data augmentation methods, namely shufflenet, MobileNet, Squeezenet, and Efficientnet. Each classification has an accuracy of 96.35 % (shufflenet_v2_x1_0), 97.38 % (mobilenet_v3_large), 91.36 % (squeezenet1_1), and 97.20 % (efficient net-b0). Reddy et al. [26] used semi-automatic labeling and semantic segmentation (ResUNet++) to achieve an average accuracy of 95 %. Additionally, Singh et al. [27] used AlexNet and GoogleNet to achieve an accuracy of 97.14 % and 98.67 %, respectively. Amri et al. [28] reported accuracy of 91.67 % and 94.44 % for ResNet and VGGNet, respectively. According to Kristensenet et al. [29], DIBas is first subjected to a feature extraction process before the classification stage is carried out. The extracted features are color and morphological, with a classification accuracy of 99 % using the random forest algorithm. Jamshidi et al. [30] used 2722 data collected with the same number of bacteria. The EfficientNetV2 architecture formed a classification accuracy of up to 99.33 %.

Despite the significant progress achieved in previous studies, several challenges and gaps remain in applying CNN architectures within microbiology, particularly concerning the generalization of models across datasets and circumstances. One of the critical challenges is the limited exploration of the latest architectures that may offer superior performance or efficiency in classifying *Lactobacillus* species. Existing researches mainly focus on the broader classification of bacterial species, leaving a gap in the specific application of *Lactobacillus* strain classification methods.

Recent studies have begun to address these challenges and gaps in current methodologies. For instance, García (2022) [15] highlighted the limitations of current CNN-based approaches in differentiating closely related bacterial species under various imaging conditions. The findings highlight the necessity for further research into more robust architectures, training, strategies, and the standardization of experimental data. Therefore, it is necessary to improve the accuracy and reliability of CNN models in identifying *Lactobacillus* species.

This study compares several deep-learning architectures in classifying bacterial strains of *Lactobacillus*. The experiments were carried out with an architecture proposed as an efficient solution for image classification with low computing cost regarding memory and number of calculations. The architecture was initialized by pre-training the model with the ImageNet dataset [31]. In addition, the architecture was evaluated using a confusion matrix with the augmented DIBas dataset. The contributions of this study include:

- The first research uses the *Lactobacillus* species dataset as a classification target on MNASNet and RegNet.
- The performance is close to state-of-the-art using a network architecture with less than 5 million parameters.
- The network's efficient and low-resource nature increases the ease of adapting and implementing on a mobile device.

2. Materials and method

This research developed and tested 4 deep learning architectures with several parameters of less and more than 10 million (Table 1). The architecture was trained using imagenet, and the model was transferred by adjusting the dataset to be processed, including the number of adjusted targets, as shown in Fig. 1.

The selection of deep learning architectures plays a critical role in determining the success of classification and identification *Lactobacillus* species based on microscopic images. This study chooses four architectures (MNASNet, RegNet Xception, and InceptionV3) due to their unique characteristics and compatibility with the dataset. This selection is based on proven performance in general image classification tasks and the classification of *Lactobacillus* species.

MNASNet is an architecture developed focusing on efficiency, especially in its application to mobile and embedded systems. The architecture is designed with principles that optimize computational resources without significantly compromising accuracy. However, the efficiency advantage also comes with challenges. MNASNet, despite its efficiency, may struggle when applied to highly complex datasets. This is caused by its more minimalist architecture than the more significant and deeper models [32]. RegNet is a stand-out architecture that offers robust regulation through its flexible and scalable design. RegNet can reduce the risk of overfitting, a common problem often in classification tasks with limited or particular datasets. The strong regulation ensures the model learns from the training data and generalizes well to unseen data. However, it needs for carefully tuned to achieve optimal performance. The process requires significant time and computational resources, especially if the dataset is highly complex or has highly varied [33]. Xception, for extreme inception, is an architecture that leverages depthwise separable convolutions to enhance model efficiency and accuracy. The approach allows Xception to capture fine details in images more efficiently, making it a strong choice for complex image recognition tasks. In *Lactobacillus* species classification, Xception's ability to capture fine details is crucial. Microscopic images often have tiny variations between one species and another, and models that can capture the subtle differences will have a clear advantage in accuracy. However, like RegNet, Xception also requires significant computational resources, mainly when applied to large or highly complex datasets [34]. InceptionV3 is an architecture known for its Inception modules, which allow multi-scale information processing within a single layer. The multi-scale approach is beneficial in analyzing complex images, such as microscopic images, where details at various levels of resolution can provide different and essential information for classification tasks. The main advantage of InceptionV3 lies in its flexibility in handling multi-scale information, making it highly useful in classifying *Lactobacillus* species where morphological details can vary greatly. However, the challenge faced when using InceptionV3 is its optimization and implementation complexity. Its modular structure, while highly flexible, can also be more challenging to optimize than simpler architectures [35]. The stages of the architecture use Python 3.10.12 with Pytorch 2.0.0 library.

Table 1. Number of parameters for each architecture.

Deep Learning Architecture	Parameter
MNASNet1_0	3,116,403
RegNetx002	2,684,792
Xception	22,855,952
InceptionV3	27,161,264

2.1. Data acquisition and preparation

This research adopted data from DIBaS [16]. DIBaS contains 660 microscopic images evenly distributed among 32 species of bacteria and 1 fungus and is available as a public database. The dataset was selected using only 11 species of *Lactobacillus*; hence, the dataset contained 250 images. In Poland, Jagiellonian University collected all the images to obtain DIBaS data using an Olympus CX31 upright biological microscope equipped with an SC30 camera (Olympus Corporation, Japan). A typical example of *Lactobacillus* species is presented in Fig. 2.

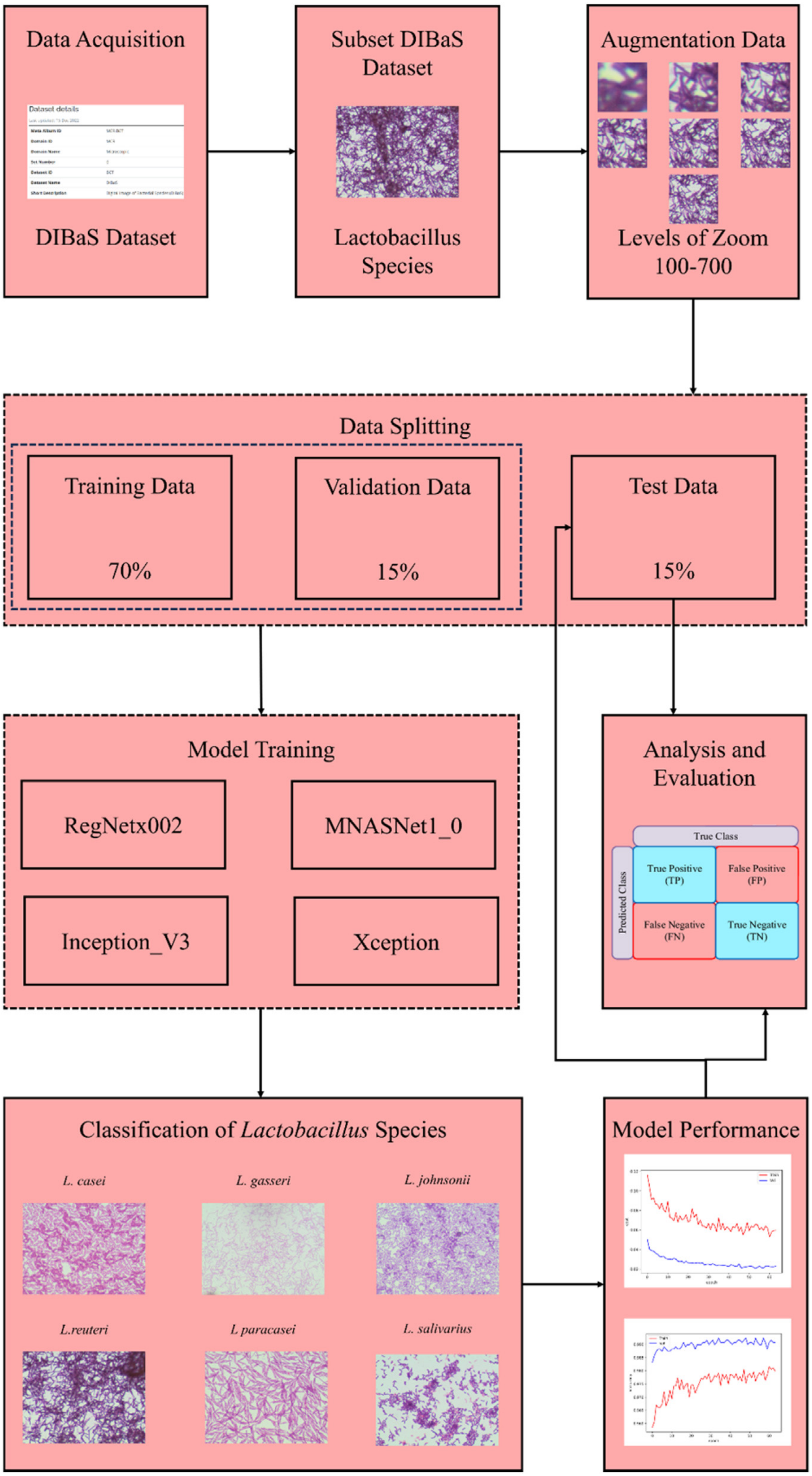

Fig. 1. Research stages.

The data augmentation (250 data) is carried out before being used in the model creation process. The strategy aimed at increasing the amount of data, where an original image can produce an extra 35. This study used a simulation of various zoom levels for the bacteria obtained by cutting multiple regions of different sizes from the original image [15]. Each original image produces 5 cuts and is zoomed into

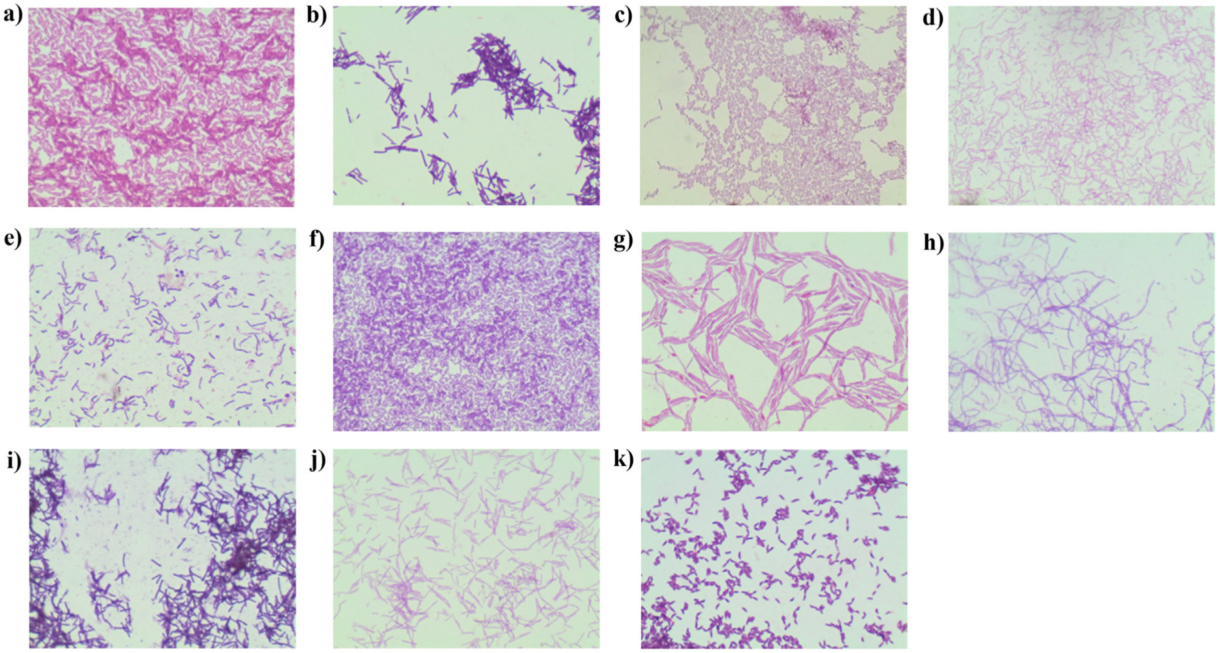

Fig. 2. Samples of Lactobacillus *species, a) L. casei, b) L. crispatus, c) L. delbrueckii, d) L. gasseri, e) L. jehnsenii, f) L. johnsonii, g) L. paracasei, h) L. plantarum, i) L. reuteri, j) L. rhamnosus, and k) L. salivarius.*

seven sizes from 100 to 700. Data augmentation has 7290 images, as can be seen in Algorithm 1. All proposed architectures are designed to process inputs of 224 × 224 pixels; hence, after data augmentation, we resized every sample to the input size using Lanczos interpolation. The number of samples for each species and a visualization of the augmentation process are available in Fig. 3 and Table 2, respectively. The data augmentation method is based on the approach of García et al. [15], who demonstrated that this augmentation technique could improve accuracy by up to 2 % with fewer epochs than data that was not augmented. Additionally, data augmentation can avoid fitting issues, ensuring that the model generalizes better to unseen data.

The 7290 images were grouped into 70 % train, 15 % validation, and 15 % test data by maintaining the same class balance in the division. Additionally, the division produced training and test data for 5544 and 1188 images, respectively. The distribution of data splitting for each species is presented in Table 2. Train and validation data were utilized to train four deep learning architectures to result in models. Meanwhile, the test data were used to evaluate the model performance.

2.2. Deep learning models

2.2.1. MNASNet

MNASNet uses depthwise separable convolutions to balance performance and efficiency [32]. The architecture adopts 2 convolution layers for each block. First, a depthwise convolution layer with 3 × 3 or 5 × 5 kernels is responsible for extracting spatial features. Second, a pointwise convolution layer with a 1 × 1 kernel combines information from previously extracted features. The architecture implements the channel expansion-contraction transformation across the network. The transformation includes dynamically changing the number of channels to optimize computational efficiency and network representation capabilities. In addition, the process is carried out to minimize the computational load on specific layers. MNASNet also uses harmonious bottleneck layers to reduce the computing load [36], as presented in Fig. 4.

Algorithm 1: Data augmentation process.

```
Result: An augmented version of the DIBaS dataset.
for Species in DIBaS do
    for Sample in Species do
        for Shape in [100,200,300,400,500,600,700] do
            L=Obtain 5 crops of Sample with size Shape;
            resize all images in L to 224x224 px;
        end
        add all images in L to augmented folder;
        resize Sample to 224x224px;
        add Sample to augmented folder;
    end
end
```

The layer focuses on spatial and channel transformations, resulting in a slimmer feature map size. By introducing the layer, MNASNet balances model

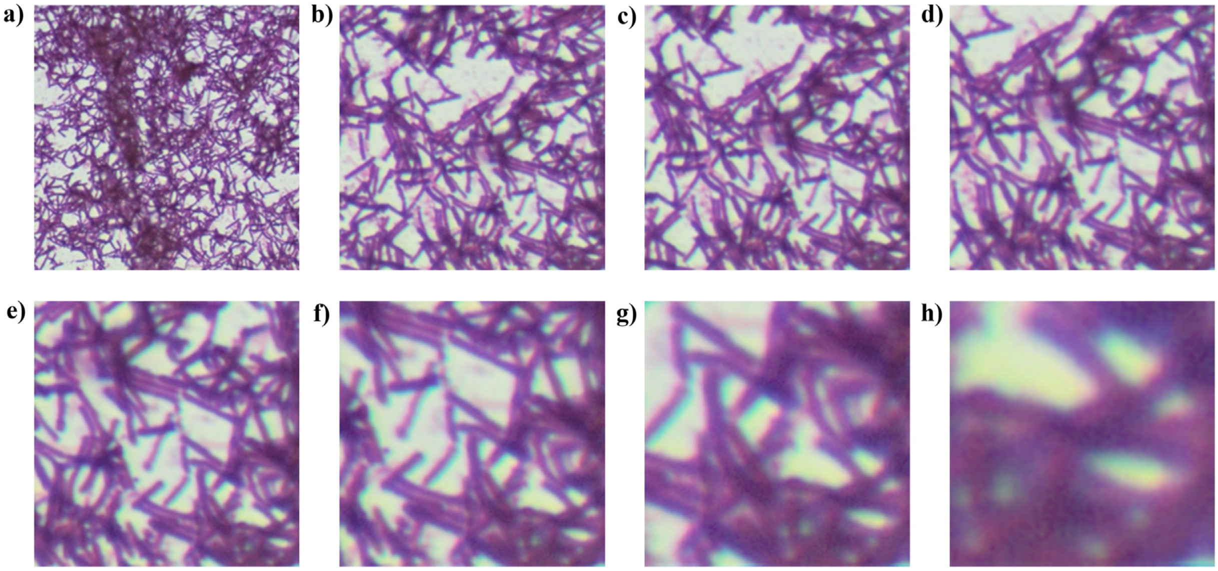

Fig. 3. L. reuteri augmentation process a) original, b) Crops 700 × 700, c) Crops 600 × 600, d) Crops 500 × 500, e) Crops 400 × 400, f) Crops 300 × 300, g) Crops 200 × 200, h) Crops 100 × 100.

accuracy and computational efficiency. Meanwhile, the harmonious bottleneck layer has spatial and channel dimension expansions and contractions [37]. The method forms an improved convolutional layers efficient model accuracy. The comparison of depthwise separable convolution blocks and harmonious bottleneck layers is shown in Equations (1) and (2). The spatial size of the input and output feature map, channel, and kernel size is $H \times W$, $C1/C2$, and $K \times K$ [32]. Meanwhile, B is the block's computational cost between the contraction and spatial expansion operations. The total cost of depthwise separable convolution is:

$$(H \times W \times C1 \times K \times K) + (H \times W \times C1 \times C2) \quad (1)$$

The total cost of the harmonious bottleneck layer is:

$$\frac{B}{s^2}\left(\frac{H}{s} \times \frac{W}{s} \times C1 + H \times W \times C2\right) \times K^2\Big) \quad (2)$$

Table 2. The amount of species data and the allocation of train, validation, and test data.

No	Species of *Lactobacillus*	Train set	Validation	Test set	Count
1	*L. casei*	504	111	105	720
2	*L. crispatus*	485	125	110	720
3	*L. delbrueckii*	500	104	116	720
4	*L. gasseri*	524	92	104	720
5	*L. jehnsenii*	491	107	122	720
6	*L. johnsonii*	497	132	91	720
7	*L. paracasei*	507	107	106	720
8	*L. plantarum*	533	94	93	720
9	*L. reuteri*	489	118	113	720
10	*L. rhamnosus*	512	111	97	720
11	*L. salivarius*	502	87	131	720
	Total	**5544**	**1188**	**1188**	**7920**

2.2.2. RegNet

RegNet is a neural network design paradigm utilized to simplify the training of deeper networks by adopting a regularization method [33]. The method aims to design neural networks with a regularization-focused method to tackle the training of deeper models more efficiently. In addition, RegNet is used as the basis of design, allowing better learning of deep representations through shortcut connections. The method focuses on channel expansion-contraction transformation to simplify network design. RegNet includes adaptively changing the number of channels to increase efficiency and reduce complexity. The architecture has 3 main parts: stem, body, and head. Stem is responsible for preprocessing with strides 2 and 32 and a 3 × 3 convolution kernel. The body performs feature extraction, and each stage operates at a decreasing resolution. Meanwhile, the head contains an average pooling and a fully connecting layer.

The modules in RegNet have linear residual connections except the first and last. The residual connections help in more profound network training. Moreover, RegNet concentrates on designing individual network instances and devises a design space for parameterizing a population. This allows for regularization in the overall network design [38]. RegNet also introduces harmonious bottleneck layers to reduce the computational load.

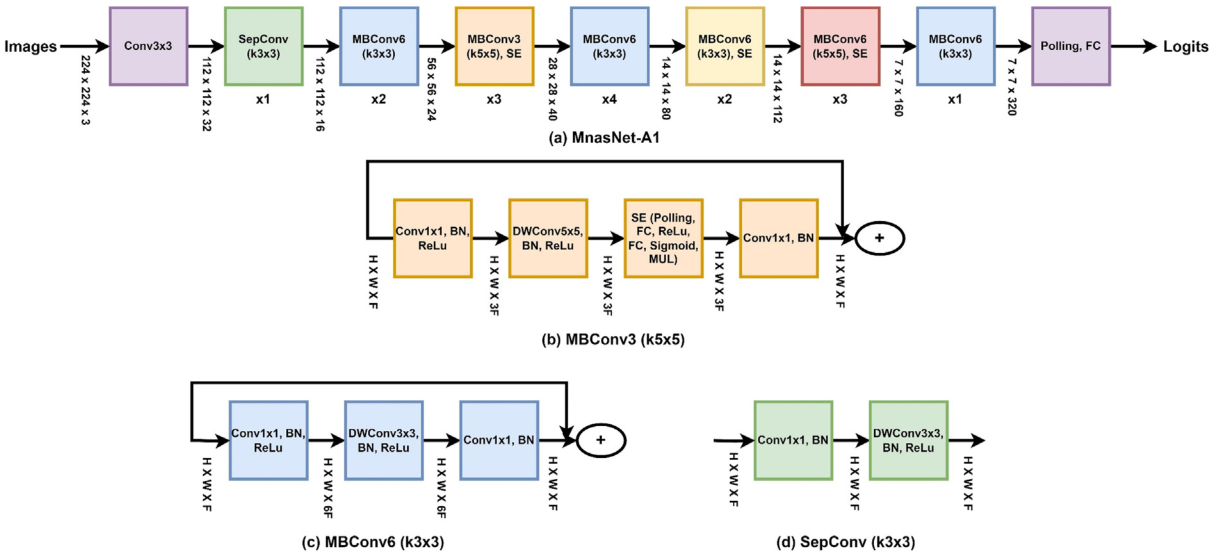

Fig. 4. MNASNet architecture.

The layer focuses on spatial and channel transformations to balance representational capabilities and efficiency. The architecture is designed to improve network training efficiency and provide strong generalization capabilities across different settings, as shown in Fig. 5.

2.2.3. Xception

Xception is a CNN architecture introduced by the developer of Keras, François Chollet [34]. The architecture takes an extreme method, focusing on depthwise separable convolutions. This is achieved based on the Inception architecture, with significant improvements. Deep separation is a core principle that includes depthwise and pointwise convolution, where the channel information is separated and combined. Xception uses 1 × 1 convolution on each depthwise channel to ensure extreme separation. The process enables better feature extraction at different abstraction levels.

The architecture is divided into 3 flow structures, namely, inbound, middle, and outbound. Each structure contains 14 modules with 36 convolutional layers. There are residual connections except the first and last modules of the incoming and outgoing flow. Xception architecture starts with an inbound flow, which contains 4 modules, and each module has 2 convolution layers. In the first module, convolution is carried out using a size of 3 × 3 with 32 and 64 filters. In the other three modules, separate convolutions are realized with a size of 3 × 3 using 128, 256, and 728 filters. The incoming flow accepts inputs at 3 × 299 × 299 image and creates an output with a 19 × 19 × 728 feature map [39].

A total of 3 convolution processes with 728 filters in 3 × 3 are repeated 8 times in the middle flow, creating a 19 × 19 × 728 map in the output. The feature map, which is the middle output, is provided as input to the exit flow with 2 modules. In the first module, separate convolution is performed with 728

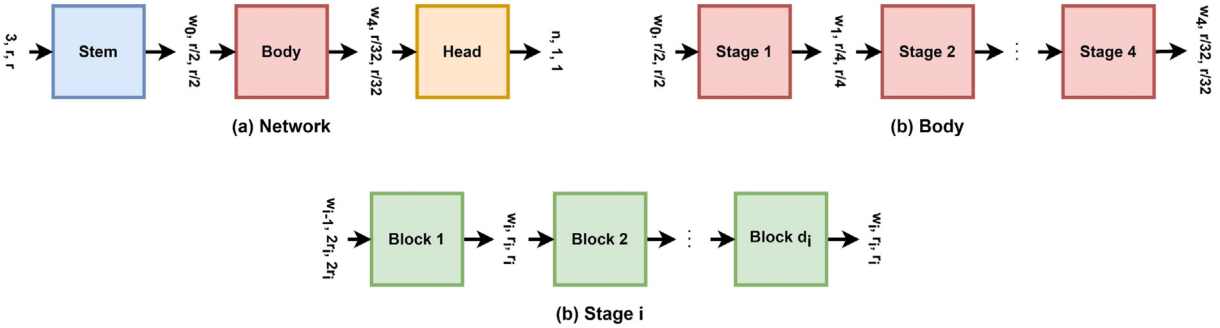

Fig. 5. RegNet architecture.

and 1024 filters at a size of 3 × 3. Meanwhile, the last module performs the process with 1536 and 2048 filters. The architecture ends with adding a fully connected layer, as shown in Fig. 6.

As with ResNet, Xception uses linear residual connections around splitting blocks. The process trains deeper models and overcomes the risk of vanishing gradients. The architecture is essentially a linear stack of deeply separated convolutions for interpretation and modification. Meanwhile, the feature extraction module consists of some linear separation blocks to make the model easy to understand.

2.2.4. Inception V3

InceptionV3 is a CNN architecture widely known for image recognition and classification tasks [35]. The Google Research team developed the architecture to overcome model complexity and improve accuracy. In addition, a computationally efficient architecture is designed with high representation capabilities for understanding complex image features. The method leverages a multi-path, where varying filter sizes are used in a single layer to extract spatial information at multiple scales.

The prominent architecture of InceptionV3 is built using inception modules, which consist of several convolution paths with different filter sizes. The design helps in capturing features at various levels of abstraction. Other models can be simplified by reducing the channel dimensions using 1 × 1 convolution. However, the process reduces the computational load and model parameters. InceptionV3 also uses auxiliary classifiers to stabilize training, prevent vanishing gradients, and produce better convergence, as shown in Fig. 7.

2.3. Models evaluation

This research uses a confusion matrix to evaluate classification models, namely accuracy, precision, sensitivity, and F1 Score [40]. Additionally, classification models can be evaluated through categorical cross-entropy, and the equations used are shown in Table 3.

3. Result and discussion

3.1. Training details for model selection

The training phase was applied using Python 3.10.12 with Pytorch 2.0.0 library. The code is written and executed in a Kaggle Notebook environment with graphic processing unit (GPU) accelerators such as Nvidia P100 (https://www.kaggle.com/). The computational setup consisted of the Windows 11 (64-bit) operating system, powered by a 12th-generation Intel i5 processor and supported by 16GB of RAM.

In addition, different optimizers are explored while keeping other factors uniform. The Adam algorithm also achieves good results on training and validation sets [43], with the learning rates used in the adaptation and fine-tuning phases at 0.001 and 0.00001. Stable convergence of the training loss function is also reported with a batch size 32. Finally, the number of epochs is set to 20 and 100 in the adaptation and fine-tuning phases, respectively. A callback function is defined to monitor validation accuracy and stop training when there is no improvement for 5 and 10 consecutive epochs in the adaptation and fine-tuning phase. Meanwhile, training and validation at the fine-tuning stage show good performance, including MNASNet, RegNet, Xception, and InceptionV3 at an accuracy of 99.22 %, 99.07 %, 99.66 %, and 97.31 %, as well as a loss of 0.0208, 0.0278, 0.0133, and 0.1483, respectively (Fig. 8). The first three models (MNASNet, RegNet, and Xception) exhibited exceptionally high accuracy, exceeding 99 %, with no statistically significant differences observed. It indicates that the models are highly stable and effective during the training

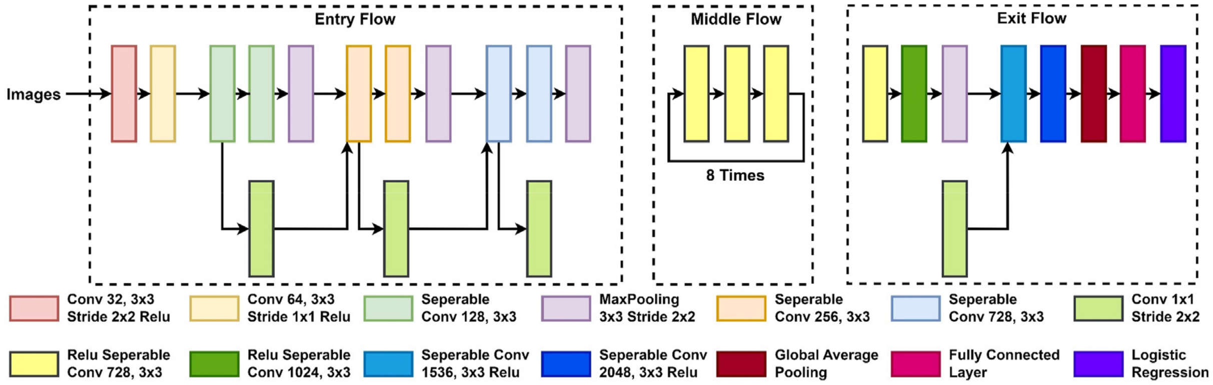
Entry Flow
Middle Flow
Exit Flow
Images
8 Times
Conv 32, 3x3 Stride 2x2 Relu
Conv 64, 3x3 Stride 1x1 Relu
Seperable Conv 128, 3x3
MaxPooling 3x3 Stride 2x2
Seperable Conv 256, 3x3
Seperable Conv 728, 3x3
Conv 1x1 Stride 2x2
Relu Seperable Conv 728, 3x3
Relu Seperable Conv 1024, 3x3
Seperable Conv 1536, 3x3 Relu
Seperable Conv 2048, 3x3 Relu
Global Average Pooling
Fully Connected Layer
Logistic Regression

Fig. 6. Xception architecture [41].

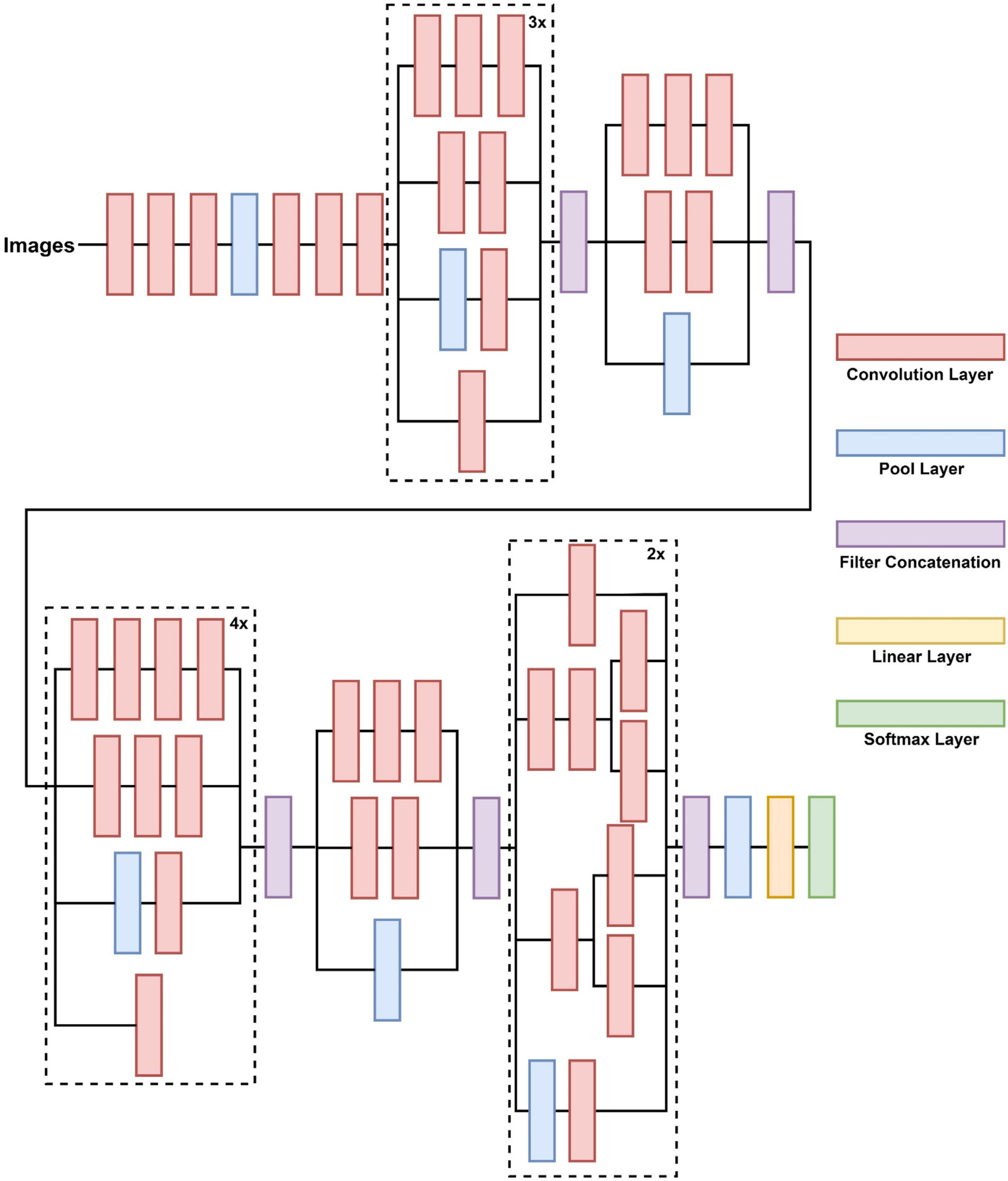

Fig. 7. Inception V3 architecture [42].

and validation phases as strong candidates for further evaluation in the testing phase. Since their accuracies are not significantly different, the three algorithms will be used in the testing phase to assess generalization capabilities on previously unseen data. Although inceptionV3 also performs well, its slightly lower accuracy than the other three models renders it less prioritized for subsequent testing.

3.2. Testing models

The three models with the lowest loss and highest accuracy during the training and validation were

Table 3. The equation for evaluating four classification models.

Evaluation	Equations
Accuracy	$\frac{(TP+TN)}{(TP+FP+FN+TN)}$
Precision	$\frac{TP}{(TP+FP)}$
Sensitivity	$\frac{TP}{(TP+FN)}$
F1 Score	$\frac{2 \times (Precision \times Sensitivity)}{(Precision + Sensitivity)}$
Categorical Cross – Entropy	$-\sum_{i=1}^{n} y_i \cdot \log(f(X_i))$

tested. The evaluation used the confusion matrix method: accuracy, precision, sensitivity, and F1 score, as visualized in Fig. 9, and the details are shown in Table 4. Meanwhile, the data in the model testing has yet to be included in the training and validation mechanism.

Based on Table 4, MNASNet has the sensitivity, highest precision, and F1 score. This is followed by Xception on accuracy and sensitivity and RegNet on precision and F1 score. The architecture also experienced a decrease in accuracy compared to the model performance in the fine-tuning phase. Meanwhile, Xception on validation data has a high accuracy of 99.66 % compared to other models. The testing phase, which used test data, experienced a decrease of 0.67 %. The value differed from MNASNet and Regnet at 0.07 % and 0.34 %, respectively. MNASNet, with only 3.1 million parameters, performed better than Xception at 2.8 million. Similarly, RegNet had only 2.6 million parameters with better values in precision and F1-score.

The three models did not experience overfitting or underfitting. It can be evident from the evaluation of precision and sensitivity, where performance remains consistent without significant differences. Based on the confusion matrix in Fig. 9, the models generally performed well in classifying 11 *Lactobacillus* species, although there were some significant misclassifications. Further analysis of the MNASNet model revealed that most of these errors—including two *L. reuteri* samples and one *L. gasseri* sample misclassified as *L. jehnsenii*, one *L. casei* sample misclassified as *L. reuteri*, three *L. plantarum* samples misclassified as *L. delbrueckii*, and three *L. rhamnosus* samples misclassified as *L. crispatus*—were caused by suboptimal image quality during testing. The images were more focused on the background color rather than the bacterial morphological features themselves (Fig. 10). This issue primarily arises due to data augmentation techniques with zooming up to 100 × 100, which distorted the original bacterial images. Therefore, critical features for classification become less transparent and more complicated for the model to interpret.

3.3. Discussion

*Special description of the title (dispensable).

The low sample size in the original dataset can lead to fitting issues in specific architectures. Research by García et al. [15] demonstrated examples of architectures that experienced underfitting even when using transfer learning, such as SqueezeNet1_0 and SqueezeNet1_1. Other architectures, like ShuffleNet_v2_x1_5 and ShuffleNet_v2_x2_0, could not resolve the issue from the outset, requiring further testing to determine whether transfer learning could yield better results on the original dataset. Based on this perspective, the main

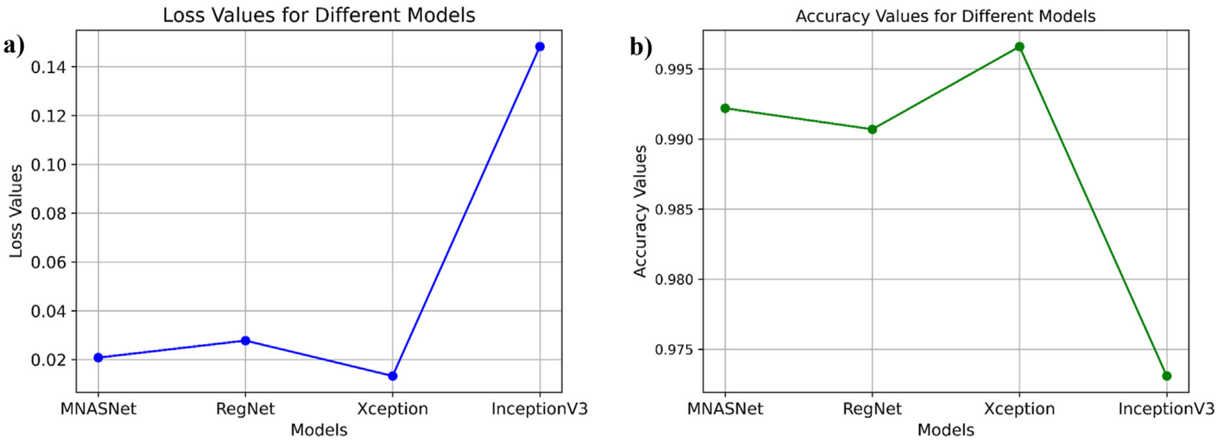

Fig. 8. The validation results in the fine-tuning phase for models (MNASNet, RegNet, Xception, and InceptionV3). a) The graph presents the loss value models. b) The graph presents the accuracy values models. As depicted, InceptionV3 exhibits the highest loss and the lowest accuracy value, indicating a poorer fit to the data than the other models.

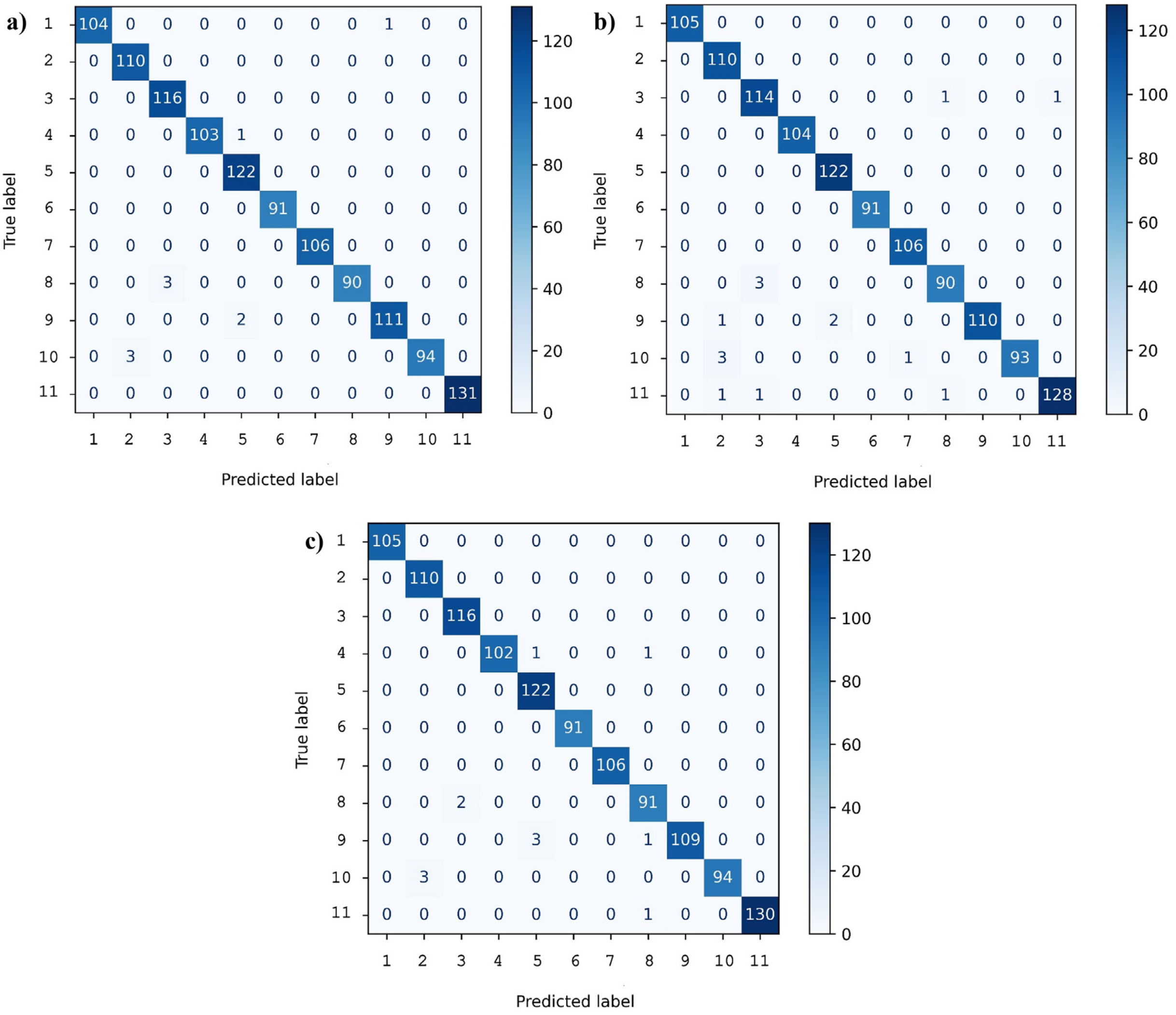

Fig. 9. Testing models confusion matrix, a) MNASNet, b) RegNet, and c) Xception.

problem lies in the lack of training data, as evidenced by the significant gap between scores obtained using the original data and those achieved after data augmentation. Data augmentation has also shown promising results, such as the study of Khalifa et al. [20], who used several augmentation techniques like zooming, Gaussian noise, and salt-and-pepper noise. Consistent with these findings, MNASNet demonstrated excellent performance using the zooming augmentation technique. The zooming augmentation produced smaller-sized images that contribute to MNASNet's strong performance, as it is designed to generate lightweight and efficient models, which are crucial in micro-image-based classification tasks [32]. Table 5 compares the architectures with parameters less than 5 million. Moreover, MNASNet exhibits better generalization capabilities, as the architecture search process involves extensive exploration of variations within the training data. In *Lactobacillus* classification, where significant morphological variation between species may occur, MNASNet maintains high performance by capturing essential features in images that other architectures might overlook. Based on Fig. 8 and Table 4, MNASNet's accuracy decline of 0.07 % compared to RegNet's 0.34 % and Xception's 0.67 % is attributed to MNASNet's flexible architecture design, which allows it to adapt to changes in data distribution and the addition of new data.

Table 4. Accuracy, precision, sensitivity, and f1 score of the three models.

Models	Accuracy	Precision	Sensitivity	F1 Score
MNASNet	**99.15 %**	**99.09 %**	**99.14 %**	**99.11 %**
RegNet	98.73 %	98.73 %	98.77 %	98.75 %
Xception	98.99 %	98.71 %	98.78 %	98.74 %

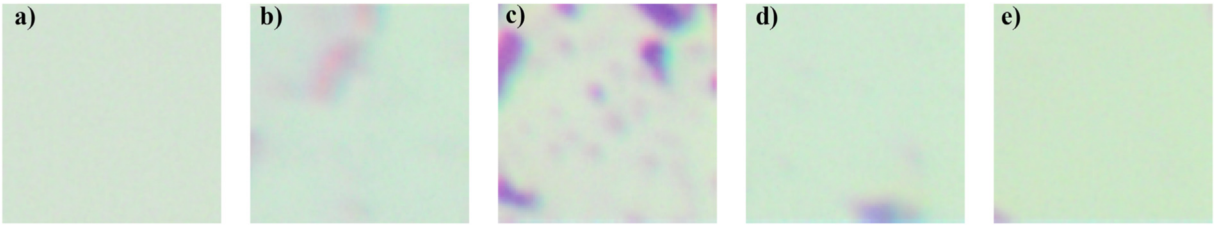

Fig. 10. Illustrates examples of misclassifications made by the MNASNet model in identifying Lactobacillus *species. a) L. casei, b) L. gaseri, c) L. reuteri, d) L. plantarum., and e) L. rhamnosus.*

Table 5. Compares the results of architectures with parameters less than 5000 M.

Method	Description	Parameters	Accuracy
Khalifa et al. [20]	Custom CNN	0.541 M	98.22 %
Rujichan et al. [22]	MobileNet v2	3.904 M	95.09 %
García et al. [15]	MobileNet v3	4.243 M	97.38 %
	Shufflenet_v2	1.286 M	96.35 %
	Squeezenet1	0.738 M	91.36 %
	Efficient net-b0	4.048 M	97.20 %
Proposed method	MNASNet	3.116 M	**99.15 %**

As shown in Fig. 11, the average execution time per epoch for the MNASNet model indicates a significant training speed, with an average execution time of 19.59 s per epoch. The speed surpasses other models, such as Xception and InceptionV3, which require average times of 96.61 s and 55.95 s per epoch.

The higher average per-epoch speed of MNASNet is primarily due to its lighter architectural design, which allows for faster backpropagation and optimization processes without compromising accuracy. It is crucial in practical applications, especially when working with large datasets or adopting fine-tuning approaches requiring multiple iterations.

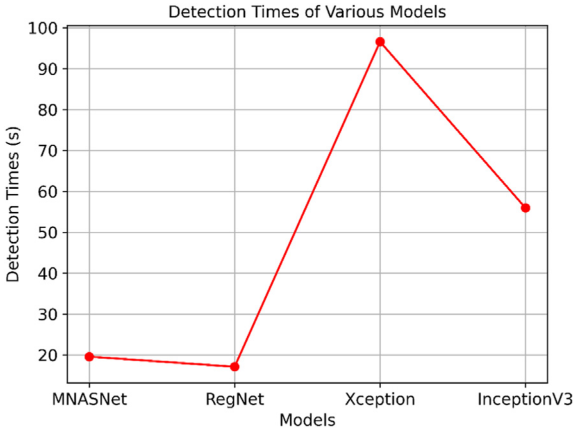

Fig. 11. The line plot illustrates the detection time (in seconds) of four different models: MNASNet, RegNet, Xception, and InceptionV3.

Although MNASNet achieves faster average per-epoch speeds, this advantage comes with additional costs during the architecture search phase, which may require more time and resources. Once the optimal architecture is found, the high average per-epoch speed makes the model more effective for deployment in various use cases. These findings confirm that MNASNet offers significant advantages in terms of training efficiency and, therefore, is more suitable for applications that require fast training cycles and optimal resource utilization.

4. Conclusions

Lactobacillus, a beneficial bacterium, plays a significant role in the health and food industries, particularly in fermentation products like yogurt and kefir. This study reveals the potential of deep learning architectures in classifying *Lactobacillus* species from microscopic images, achieving accuracies above 95 %. Among the tested architectures, MNASNet stands out with its optimal balance of performance and computational efficiency, achieving an accuracy of 99.15 % with only 3.1 million parameters. Moreover, MNASNet demonstrated a high computational efficiency, with an average computation time of 19.59 s per epoch. It makes MNASNet more efficient than other models, which typically have over 20 million parameters and require longer computation times. Therefore, MNASNet is highly suitable for implementation on mobile and embedded devices with limited resources.

The implications of this research extend beyond mere classification accuracy. The ability to deploy such an efficient model in resource-constrained environments opens new avenues for real-time monitoring and quality control in the food industry. Additionally, as the number of known *Lactobacillus* species has expanded from 11 to over 200, the scalability and adaptability of this approach have become crucial. It suggests that further development could extend the model to accommodate a broader range of species, enhancing its utility in both industrial and research settings.

Future work could explore further integrating additional data sources or refining model architectures to enhance classification accuracy and efficiency. The promising results underscore the potential of deep learning in microbiology, paving the way for developing more sophisticated and accessible tools for bacterial identification.

Ethics information

We confirm that our research did not involve any ethical concerns requiring formal review, as it adheres to standard research protocols.

Funding

Additionally, this study did not receive any external funding and was self-funded.

Acknowledgements

No external funding was used for this study.

References

[1] M.A. Augustin, M. Riley, R. Stockmann, L. Bennett, A. Kahl, T. Lockett, M. Osmond, P. Sanguansri, W. Stonehouse, I. Zajac, L. Cobiac, Role of food processing in food and nutrition security, Trends Food Sci Technol 56 (2016) 115−125, https://doi.org/10.1016/j.tifs.2016.08.005.

[2] D. Granato, F.J. Barba, D.B. Kovačević, J.M. Lorenzo, A.G. Cruz, P. Putnik, Functional foods: product development, technological trends, efficacy testing, and safety, Annu Rev Food Sci Technol 11 (2020) 93−118, https://doi.org/10.1146/annurev-food-032519051708.

[3] M.M. Abedin, R. Chourasia, L.C. Phukon, P. Sarkar, R.C. Ray, S.P. Singh, A.K. Rai, Lactic acid bacteria in the functional food industry: biotechnological properties and potential applications, Crit Rev Food Sci Nutr 63 (2023) 1−19, https://doi.org/10.1080/10408398.2023.2227896.

[4] Bintsis, Lactic acid bacteria: their applications in foods, J Bacteriol Mycol Open Access 6 (2018) 89−94, https://doi.org/10.15406/jbmoa.2018.06.00182.

[5] T. Gemechu, Review on lactic acid bacteria function in milk fermentation and preservation, Afr J Food Sci 9 (2015) 170−175, https://doi.org/10.5897/AJFS2015.1276.

[6] J. Ji, H. Yang, Using probiotics as supplementation for helicobacter pylori antibiotic therapy, Int J Mol Sci 21 (2020) 1−15, https://doi.org/10.3390/ijms21031136.

[7] B. Hamid, M. Zaman, S. Farooq, S. Fatima, R.Z. Sayyed, Z.A. Baba, T.A. Sheikh, M.S. Reddy, H.E. Enshasy, A. Gafur, N.L. Suriani, Bacterial plant biostimulants: a sustainable way towards improving growth, productivity, and health of crops, Sustainability 13 (2021) 1−24, https://doi.org/10.3390/su13052856.

[8] S.S. Behera, R.C. Ray, N. Zdolec, Lactobacillus plantarum with functional properties: an approach to increase safety and shelf-life of fermented foods, BioMed Res Int 2018 (2018) 1−18, https://doi.org/10.1155/2018/9361614.

[9] S.A. Jenkins, D.B. Drucker, M.G.L. Keaney, L.A. Ganguli, Evaluation of the rapid id 32a system for the identification of bacteroides fragilis and related organisms, J Appl Bacteriol 71 (1991) 360−365, https://doi.org/10.1111/j.1365-2672.1991.tb03801.x.

[10] D.A. Spratt, Significance of bacterial identification by molecular biology methods, Endod Top. 9 (2004) 5−14, https://doi.org/10.1111/j.1601-1546.2004.00106.x.

[11] W.M. Ahmed, B. Bayraktar, A.K. Bhunia, E.D. Hirleman, J.P. Robinson, B. Rajwa, Classification of bacterial contamination using image processing and distributed computing, IEEE J Biomed Health Inf 17 (2013) 232−239, https://doi.org/10.1109/TITB.2012.2222654.

[12] C. Sommer, D.W. Gerlich, Machine learning in cell biology−teaching computers to recognize phenotypes, J Cell Sci 126 (2013) 5529−5539, https://doi.org/10.1242/jcs.123604.

[13] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, D.V. Valen, Deep learning for cellular image analysis, Nat Methods 16 (2019) 1233−1246, https://doi.org/10.1038/s41592-019-0403-1.

[14] W. Zhu, B. Braun, L.H. Chiang, J.A. Romagnoli, Investigation of transfer learning for image classification and impact on training sample size, Chemom Intell Lab Syst. 211 (2021) 1−9, https://doi.org/10.1016/j.chemolab.2021.104269.

[15] R.G. García, S.J. Rodríguez, B.B. Martínez, C.H. Gracidas, R. M. Torres, Efficient deep learning architectures for fast identification of bacterial strains in resource-constrained devices, Multimed Tools Appl 81 (2022) 39915−39944, https://doi.org/10.1007/s11042022-13022-8.

[16] B. Zieliński, A. Plichta, K. Misztal, P. Spurek, M.B. Włoch, D. Ochońska, Deep learning approach to bacterial colony classification, PLoS ONE 12 (2017) 1−14, https://doi.org/10.1371/journal.pone.0184554.

[17] O.F. Nasip, K. Zengin, Deep learning based bacteria classification, in: 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), IEEE. 2018, pp. 1−5, https://doi.org/10.1109/ISMSIT.2018.8566685.

[18] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: 2015 3rd International Conference on Learning Representations (ICLR), 2015, pp. 1−14, https://doi.org/10.48550/ARXIV.1409.1556, arXiv.

[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun ACM 60 (2017) 84−90, https://doi.org/10.1145/3065386.

[20] N.E.M. Khalifa, M.H.N. Taha, A.E. Hassanien, Deep bacteria: robust deep learning data augmentation design for limited bacterial colony dataset, Int J Reason-Based Intell Syst. 11 (2019) 256−264, https://doi.org/10.1504/IJRIS.2019.102610.

[21] M. Talo, An automated deep learning approach for bacterial image classification, in: 2019 2nd International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES), 2019, pp. 1−5, https://doi.org/10.48550/ARXIV.1912.08765, arXiv.

[22] C. Rujichan, N. Vongserewattana, P. Phasukkit, Bacteria classification using image processing and deep convolutional neural network, in: 2019 12th Biomedical Engineering International Conference (BMEiCON), IEEE. 2019, pp. 1−4, https://doi.org/10.1109/BMEiCON4.7515.2019.8990270.

[23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.C. Chen, Mobilenetv2: inverted residuals and linear bottlenecks, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. 2018, pp. 4510−4520, https://doi.org/10.1109/CVPR.2018.00474.

[24] M.A. Elaziz, K.M. Hosny, A.A. Hemedan, M.M. Darwish, Improved recognition of bacterial species using novel fractional-order orthogonal descriptors, Appl Soft Comput 95 (2020) 1−18, https://doi.org/10.1016/j.asoc.2020.106504.

[25] B.D. Satoto, M.I. Utoyo, R. Rulaningtyas, E.B. Koendhori, An auto contrast custom convolutional neural network to identifying gram-negative bacteria, in: 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), IEEE. 2020, pp. 70−75, https://doi.org/10.1109/CENIM51130.2020.9297964.

[26] C. Reddy, P.A. Reddy, V.R. Kanabur, D. Vijayasenan, S.S. David, S. Govindan, Semi-automatic labeling and semantic segmentation of gram-stained microscopic images from dibas dataset, in: 2023 2nd International Conference on Computational Systems and Communication (ICCSC), IEEE. 2023, pp. 1−6, https://doi.org/10.1109/ICCSC56913.2023.10142976.

[27] A. Singh, A. Kumar, H.M. Salman, N. Rawat, S.K. Jain, A.T. Rao, Transfer learning approach on bacteria classification from microscopic images, in: 2022 5th International Conference on Contemporary Computing and Informatics (IC3I), IEEE. 2022, pp. 982−987, https://doi.org/10.1109/IC3I56241.2022.10072818.

[28] M.F. Amri, A.R. Yuliani, D.E. Kusumandari, A.I. Simbolon, M.I. Rizqyawan, U. Nadiya, Bacterial classification using deep structured convolutional neural network for low resource data, J Elektron Telekomun 23 (2023) 47−54, https://doi.org/10.55981/jet.533.

[29] K. Kristensen, L.M. Ward, M.L. Mogensen, S.L. Cichosz, Using image processing and automated classification models to classify microscopic gram stain images, Comput Methods Programs Biomed Update. 3 (2023) 1−7, https://doi.org/10.1016/j.cmpbup.2022.100091.

[30] M.B. Jamshidi, S. Sargolzaei, S. Foorginezhad, O. Moztarzadeh, Metaverse and microorganism digital twins: a deep transfer learning approach, Appl Soft Comput 147 (2023) 1−16, https://doi.org/10.1016/j.asoc.2023.110798.

[31] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, F.F. Li, Imagenet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2009, pp. 248−255, https://doi.org/10.1109/CVPR.2009.5206848.

[32] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, Q.V. Le, MnasNet: platform-aware neural architecture search for mobile, in: 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 2815−2823, https://doi.org/10.1109/CVPR.2019.00293.

[33] I. Radosavovic, R.P. Kosaraju, R. Girshick, K. He, P. Dollar, Designing network design spaces, in: 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 10425−10433, https://doi.org/10.1109/CVPR42600.2020.01044.

[34] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 1800−1807, https://doi.org/10.1109/CVPR.2017.195.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 2818−2826, https://doi.org/10.1109/CVPR.2016.308.

[36] P. Shah, M.E. Sharkawy, A-mnasnet: augmented mnasnet for computer vision, in: 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE. 2020, pp. 1044−1047, https://doi.org/10.1109/MWSCAS48704.2020.9184619.

[37] S. Aghera, H. Gajera, S.K. Mitra, Mnasnet based lightweight cnn for facial expression recognition, in: 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC), IEEE. 2020, pp. 1−6, https://doi.org/10.1109/iSSSC.50941.2020.9358903.

[38] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, Z. Xu, Regnet: self-regulated network for image classification, IEEE Trans Neural Network Learn Syst. 34 (2023) 9562−9567, https://doi.org/10.1109/TNNLS.2022.3158966.

[39] Ö. Polat, Detection of covid-19 from chest ct images using xception architecture: a deep transfer learning based approach, Sakarya Univ J Comput Inform Sci 25 (2021) 800−810, https://doi.org/10.16984/saufenbilder.903886.

[40] A. Tharwat, Classification assessment methods, Appl Comput Inf. 17 (2021) 168−192, https://doi.org/10.1016/j.aci.2018.08.003.

[41] M.N. Noor, M. Nazir, S.A. Khan, I. Ashraf, O.Y. Song, Localization and classification of gastrointestinal tract disorders using explainable ai from endoscopic images, Appl Sci 13 (2023) 1−19, https://doi.org/10.3390/app1315.9031.

[42] L.C.R. Gómez, H.L. García, J.M.C. Padilla, C.B. Huidobro, H. G. Rosales, R.S. Robles, J.G.A. Olague, J.I.G. Tejada, C.E.G. Tejada, D. Rondon, K.O.V. Condori, Detection of pedestrians in reverse camera using multimodal convolutional neural networks, Sensors 23 (2023) 1−21, https://doi.org/10.3390/s23177559.

[43] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: 2019 3rd International Conference on Learning Representations (ICLR), arXiv, 2015, pp. 1−15, https://doi.org/10.48550/ARXIV.1412.6980.