



Enhancing coffee bean classification: a comparative analysis of pre-trained deep learning models

Esraa Hassan¹

Received: 11 October 2023 / Accepted: 21 February 2024 / Published online: 1 April 2024
© The Author(s) 2024

Abstract

Coffee bean production can encounter challenges due to fluctuations in global coffee prices, impacting the economic stability of some countries that heavily depend on coffee production. The primary objective is to evaluate how effectively various pre-trained models can predict coffee types using advanced deep learning techniques. The selection of an optimal pre-trained model is crucial, given the growing popularity of specialty coffee and the necessity for precise classification. We conducted a comprehensive comparison of several pre-trained models, including AlexNet, LeNet, HRNet, Google Net, Mobile V2 Net, ResNet (50), VGG, Efficient, Darknet, and DenseNet, utilizing a coffee-type dataset. By leveraging transfer learning and fine-tuning, we assess the generalization capabilities of the models for the coffee classification task. Our findings emphasize the substantial impact of the pre-trained model choice on the model's performance, with certain models demonstrating higher accuracy and faster convergence than conventional alternatives. This study offers a thorough evaluation of pre-trained architectural models regarding their effectiveness in coffee classification. Through the evaluation of result metrics, including sensitivity (1.0000), specificity (0.9917), precision (0.9924), negative predictive value (1.0000), accuracy (1.0000), and F1 score (0.9962), our analysis provides nuanced insights into the intricate landscape of pre-trained models.

Keywords Transfer learning · Coffee beans · Computer vision · Pre-trained model

1 Introduction

Coffee beans classification performs a required role in the coffee industry, as it impacts the quality and flavor profile of the final product [1–5]. Accurate classification of coffee beans allows producers to make informed decisions, ensuring an improved product for the end customer. There are farmers' efforts in manual methods for detecting coffee diseases, along with significant financial investment in training for this purpose [6–8]. Coffee production is sensitive to global price fluctuations, impacting economics and stability of countries dependent on this precious commodity. The rise of specialty coffee has transformed coffee consumption into a sensory experience, necessitating

precise classification and differentiation of coffee types to meet the demand of coffee enthusiasts and connoisseurs. Deep learning (DL) is a subfield of artificial intelligence (AI) that has made significant advancements in the computer vision (CV) field, enabling the development of several. The Adam optimizer is applied to pre-trained models [9–17]. Its adaptive learning rate and momentum parameters enable efficient converging to optimal solutions, navigating complex models and enhancing their performance.

The prominent optimizer in the training of pre-trained models plays a crucial role in optimizing the process to adjust learning rates per parameter, is a popular choice for fine-tuning pre-trained models for various applications [18–29].

The versatile and robustness of this tool in optimizing pre-trained architectures make it a valuable tool for deep learning practitioners to utilize pre-trained models for various tasks [30–34]. It is critical to accurately predict coffee types to ensure the quality and consistency of the final product and investigate the impact of pre-trained

✉ Esraa Hassan
esraa.hassan@ai.Kfs.edu.eg

¹ Department of Machine Learning and Information Retrieval,
Faculty of Artificial Intelligence, Kafrelsheikh University,
Kafrelsheikh 33516, Egypt

models on the accuracy and efficiency of predicting coffee types, highlighting the growing demand for specialty coffee and the need for precise classification. We use transfer learning and fine-tuning techniques to evaluate pre-trained models and identify their strengths and weaknesses. The trained methods may lead to incorrect findings, potentially using the wrong pesticides to treat diseases, causing environmental degradation rather than treating the issue [35, 36]. We explore the impact of pre-trained models on coffee type prediction accuracy and efficiency using deep learning techniques, focusing on transfer learning, and fine-tuning [37]. In this study, we investigate the effect of selecting different pre-trained models on the performance of a coffee-type prediction system. We compare various state-of-the-art pre-trained models, such as VGG, ResNet, and MobileNet, and analyze their impact on the overall accuracy, training time, and computational resources required for predicting coffee types [38–42]. We proposed the use of deep learning techniques and transfer learning to predict coffee types and compared the performance of various pre-trained models that evaluate coffee classification models' generalizability and identifies pre-trained models for higher accuracy and convergence, benefiting coffee producers and processors to enhance classification systems and economic stability as shown in Fig. 1 [43–46]. The main study contributions are:

1. Exploring the importance of selecting the right pre-trained deep learning model for accurately classifying specialty coffee types, emphasizing the need for precise classification.
2. Comparing various pre-trained models like AlexNet, LeNet, HRNet, GoogleNet, MobileNetV2, ResNet-50, VGG, EfficientNet, Darknet, and DenseNet to understand their strengths and weaknesses for specific coffee classification tasks.
3. Employing transfer learning and fine-tuning techniques to evaluate the generalizability of models for coffee classification, utilizing pre-existing knowledge from large datasets.
4. providing clear performance findings on pre-trained models for coffee classification, offering recommendations for future coffee-related applications and guiding the selection process.
5. Enhancing the performance of pre-trained models, resulting in more accurate and efficient classification of coffee types depending on Adam optimizer.

The following is how the remaining study is structured. Section 2 commonly presents related works. The methodology is presented in Sect. 3. In Sect. 4, experimental evaluation is offered. In Sect. 5, Conclusion and Discussion is provided.

2 Related works

Pre-trained models, trained on large datasets, may have limitations like inability to handle new data types or certain tasks, and may not be optimal for specific tasks due to their training conditions [47–50]. In this section, we present common related works of pre-trained models for computer vision tasks. Researchers are focusing on pre-trained model types, which are trained on large datasets, saving time and resources are illustrated in Table 1. Esgario et al. [6] suggested a powerful and useful method that could recognize and gauge the level of stress that biotic agents on coffee leaves were causing. A multitask system built on convolutional neural networks makes up the suggested method. They have also investigated using data augmentation techniques to strengthen and improve the system. The accuracy of the suggested system's ResNet50 architecture-based computational trials was 95.24%. When classifying flaws in coffee beans, Chang et al. [7] suggest a technique that has been proven to lessen prejudice. The proposed model had a 95.2% detection accuracy rate. The accuracy rate reached 100% when the model was restricted to defect detection. To develop an automated system, Sorte et al. [8] suggest a technique for employing computational approaches to identify serious diseases in coffee leaves. Expert method to help coffee growers identify diseases in their early phases. Due to the shapelessness of these two diseases, a texture attribute extraction approach for pattern recognition is inspired. Using the suggested layer, Novtahnang et al. [9] describe an ensemble strategy for DL models. Selecting three models that excel at classification and joining them to create an ensemble architecture that is then fed to classifiers to decide the output. To improve the quality and expand the data sample's size, a data pre-processing and augmentation procedure is also used [51, 52]. By achieving 97.31% validation, the suggested ensemble architecture exceeded other cutting-edge neural networks in performance. Velásquez et al. [10] propose an experiment employing a Coffee Leaf Rust development stage diagnostic model in the *Coffea arabica*, Caturra variety, and scale crop using wireless sensor networks, remote sensing, and DL techniques. An F1-score of 0.775 was attained by diagnostic model. Gope et al. [11] propose a deep neural network model to classify green coffee beans with multi-label properties. The model, modified from the EfficientNet-B1 model, uses branches to correspond to each defect after feature extraction layers. This improved overall performance by an f1-score of 0.8229, compared to the single EfficientNet-B1 model. liang et al. [13] aim to develop an automated coffee bean inspection system using YOLOv7 and a convolutional neural network (CNN). The system classifies coffee beans into broken, insect-infested,

The Effect of Choosing a Pre-trained Model to Predict Coffee Type

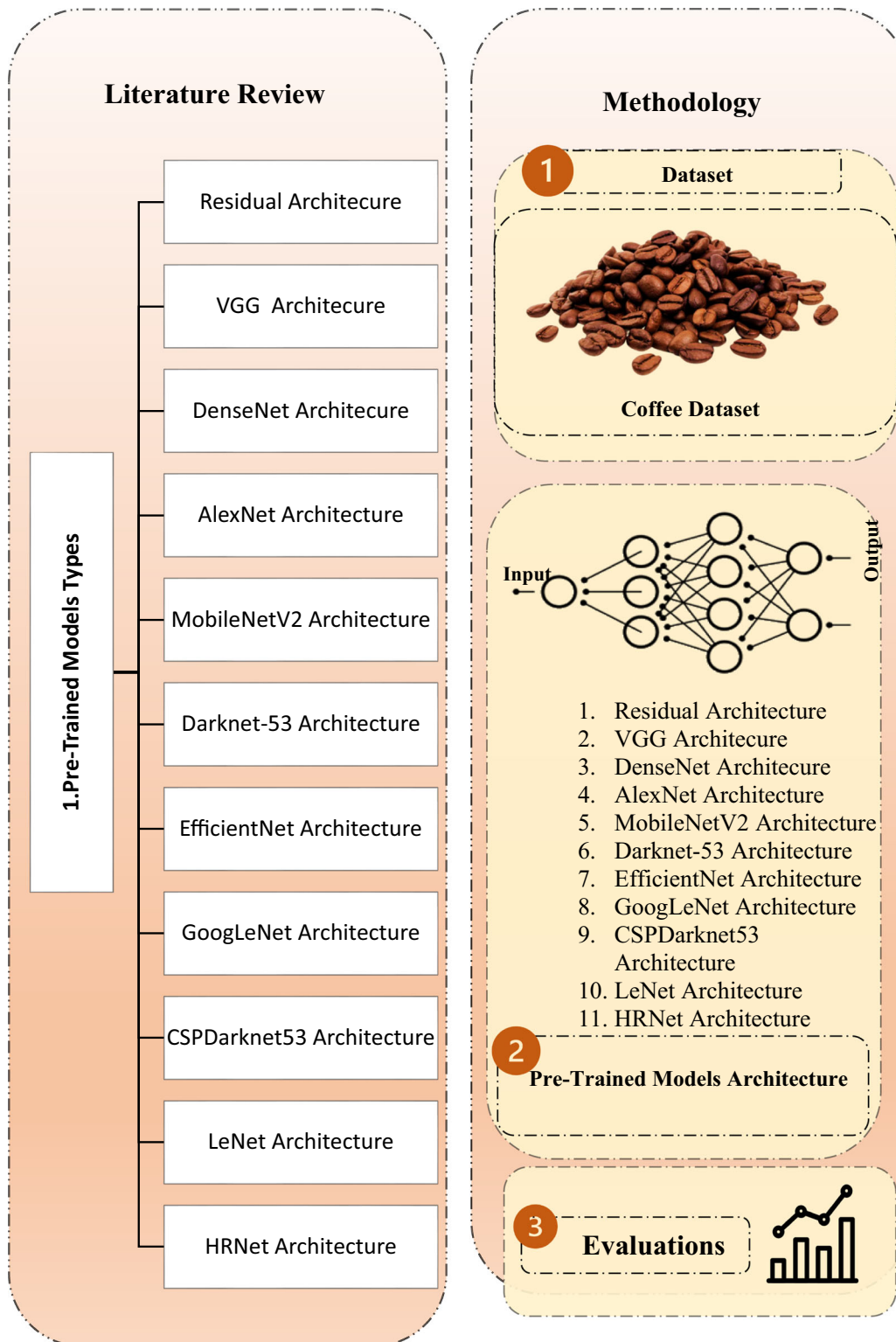


Fig. 1 The general architecture for the study

and mold categories using transfer learning. The YOLOv7 image recognition model processes the captured beans,

determining if they are good or defective. The DenseNet201 model achieves an accuracy of 98.97% in

Table 1 The related work in the ResNet (50) architecture

Year	References	Task	Dataset	Metrics
2023	Hsia et al. [20]	Quality Detection of Green Coffee	Green Coffee Beans	Accuracy: 98.38%, F1 Score: 98.24%
2023	Chen et al. [19]	Coffee Bean Classification	Two Types of Coffee Beans	F1-Score: 97.21%
2023	Ke et al. [18]	Coffee Bean Defect Detection	Coffee Bean Images	Accuracy (F1 Score): 98%
2023	liang et al. [13]	Coffee Bean Inspection	Images of coffee beans	Accuracy: 98.97%
2023	Gope et al. [11]	Distinguishing Coffee Bean Categories	Lab* color model	Recognition Rate: 0.81 (for the test set)
2022	Novtahaning et al. [9]	Ensamble model	Coffee dataset Imagenet	Accuracy = 97.31%
2021	Chang et al. [7]	CNN	Coffee dataset	Accuracy = 90.2%
2020	Esgario et al. [6]	Transfer Learning	Leaf dataset	Accuracy = 95.63%
2020	Velásquez et al. [10]	DL model	Coffee Leaf Database	F1-score = 0.775%
2019	Sorte et al. [8]	DL model	Coffee Leaf Database	Sensitivity = 0.980

classifying defective coffee beans. Ke et al. [18] explore deep convolutional neural networks (DCNNs) are proposed for capturing coffee bean images, with results showing 98% accuracy. The lightweight model, with around 250,000 parameters, is practical due to its low cost. Chen et al. [19] suggest a model architecture that combines semi-supervised learning and attention mechanisms, combining explainable consistency training and a directional attention algorithm to improve prediction ability and achieve an F1-score of 97.21%. Hsia et al. [20] propose a lightweight deep convolutional neural network (LDCNN) for detecting quality in green coffee beans. The model combines DSC, SE block, and skip block frameworks, and includes rectified Adam, lookahead, and gradient centralization to improve efficiency. The model's local interpretable model-agnostic explanations (LIME) model is used to explain predictions. Experimental results show an accuracy rate of 98.38% and F1 score of 98.24%, resulting in lower computing time and parameters.

Esgario et al.'s approach using a multitask system based on convolutional neural networks (CNNs) for stress detection in coffee leaves presents an advantage in its focused application. Nevertheless, a potential drawback lies in its limited scope, as the model may not excel in tasks beyond the detection of stress in biotic agents. Chang et al.'s model for defect detection in coffee beans demonstrates reduced bias and achieves high detection accuracy. However, its drawback lies in its restricted scope, optimized primarily for defect detection and possibly lacking generalization to other tasks [6]. Sorte et al.'s computational approach for disease identification in coffee leaves using expert methods is advantageous. However, it is limited to early-phase diseases, potentially struggling with the identification of late-stage diseases [8]. Novtahaning et al.'s ensemble strategy, combining three models for improved classification, demonstrates enhanced

performance. However, its drawback lies in its increased complexity, which may be computationally demanding [9]. Velásquez et al.'s diagnostic model integrating wireless sensor networks, remote sensing, and deep learning for Coffee Leaf Rust presents advantages in technology integration. However, the limited F1-score achieved indicates a potential need for improved diagnostic accuracy [10]. Gope et al.'s model for multi-label classification of green coffee beans using branches presents improved performance. However, its specificity to green beans may limit its optimality for other types or stages of coffee beans [11]. Liang et al.'s system integrating YOLOv7 and a convolutional neural network for coffee bean classification is advantageous in its integration of image recognition models. However, its limitation lies in focusing on specific defect categories (broken, insect-infested, and mold), potentially overlooking other defects. Sim et al.'s approach using hyperspectral imaging for rapid and non-destructive origin classification offers advantages. However, its dependency on this technology may limit its universal applicability [13]. While pre-trained models offer efficiency gains in various coffee-related applications, their limitations highlight the importance of choosing or adapting models based on the specific requirements of the task at hand.

3 Methodology

In this study, we apply a several pre-trained architectures to classify images of coffee beans by using the Coffee Bean Dataset, which contains images of various types of coffee beans [53–55]. The purpose of this comparison is focusing on the strength and weakness of this architecture. This section describes the details of the CNNs implemented for coffee-type classification.

This study concentrates on finding the most appropriate pre-trained CNN model. The entire procedure is divided into four basic steps: data acquisition, data training, data classification, and data evaluation, which are detailed below.

3.1 Data acquisition

The Coffee Bean Dataset is a collection of information about coffee beans from around the world that is shown in Fig. 2. It includes data on the origin, type, and flavor profile of each bean, as well as information on the growing conditions and processing techniques used to produce them. The images are automatically collected and saved in PNG format, with 4800 images in total, classified into 4 degrees of roasting, with 1200 images under each degree that are illustrated in Table 2. We went with the Coffee Bean Dataset for our study because it's got a solid stash of details about coffee beans from all sorts of places worldwide. We wanted to dig into the specifics of various coffee beans—where they come from, what kind they are, their flavor, how they grow, and how they're processed. The idea is to get down to the nitty-gritty and find patterns and insights that help us grasp what factors play into the unique characteristics of coffee beans [1]. The way the dataset is so neatly organized, with a whopping 4800 images in PNG format, gives us a real treasure trove of visual data. We took it a step further and sorted the beans into four roasting levels, making it a cool 1200 images for each level. This not only lets us dig into how roasting changes the look of the beans but also opens the door for a more detailed analysis of the flavors linked to each roasting level. The Coffee Bean Dataset fits like a glove with what we're

Table 2 Coffee dataset structure

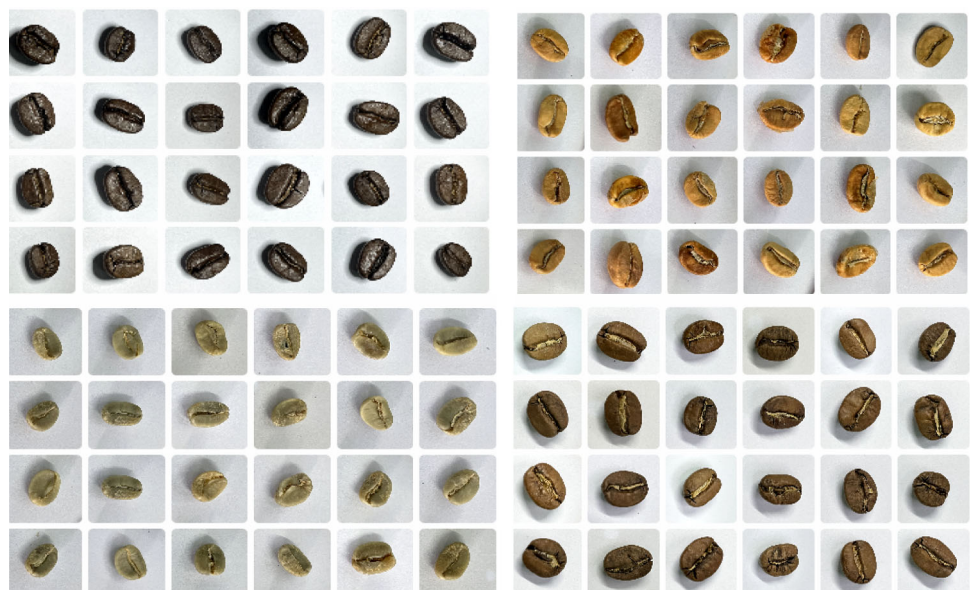
Classes	Folder	Number of Images
Dark	Training	300
Green		300
Light		300
Medium		300
Dark	Testing	100
Green		100
Light		100
Medium		100
Total		1600

trying to do in our study. We're all about exploring how the characteristics of coffee beans connect to how they look, and at the same time, we're checking out how different roasting levels shake up their flavors. This alignment is key—it makes the dataset super relevant, ensuring our study can draw some real-deal conclusions that add to the bigger picture of understanding coffee bean traits [9].

3.2 Data training

In this phase, the training of the CNNs was carried out through the dataset, ImageNet, with the objective of initializing the weights before the training on this coffee dataset [56–58]. On the next stage, we showed the advantage of transfer learning, which aims to transfer knowledge from one or more domains and apply the knowledge to another domain with a different target task [59]. Fine-tuning is a transfer learning concept that consists

Fig. 2 Coffee dataset samples



of replacing the pre-trained output layer with a layer containing the number of classes in the coffee dataset. The main purpose of using pre-trained CNN models is related to the fast and easy training of a CNN using randomly initialized weights, as well as the achievement of lower training errors than ANNs that are not pre-trained. The performance of the following CNN architectures has been evaluated for the coffee plant classification problem with common pre-trained architectures.

3.2.1 AlexNet architecture

AlexNet consists of five convolutional layers and three fully connected layers, with output fed into two fully connected layers and a softmax classifier. The AlexNet model uses the Rectified Linear Unit (ReLU), which is applied to each of the first seven layers with architecture equations [60]. It excels at learning complex features from raw data, resists overfitting, and has a low computational cost, but its main drawback is the high demand for labeled

data. We apply it to coffee classification with the following steps, as shown in Algorithm 1.

3.2.2 CSPDarknet53 architecture

The CSPDarknet53 architecture, developed by Chinese Academy of Sciences researchers, is a CNN based on Darknet53, offering high object detection accuracy and efficient computational resource utilization.

This architecture excels in object detection and image classification, but has limitations like high training data requirements, slow inference speed, and reliance on pre-trained weights from ImageNet, making it unsuitable for real-time applications and new datasets with architecture equations [61]. Algorithm 2 illustrates the detailed steps for coffee dataset with CSPDarknet53.

Algorithm 1 The Alexnet architecture for coffee image dataset

Input Data: The coffee input image as X , and Alexnet uses subscripts to indicate layer indices.

Alexnet architecture

1. Convolutional Layer 1:

$$\text{Convolution: } H_1 = \text{ReLU}(W_1 * X + b_1)$$

- Where:

- W_1 represents the convolutional kernel weights for the first layer.

- b_1 is the bias term.

- ReLU is the rectified linear unit activation function.

2. Max-Pooling Layer 1:

$$\text{Max-Pooling: } H_2 = \text{MaxPool}(H_1)$$

3. Convolutional Layer 2:

$$\text{Convolution: } (H_3 = \text{ReLU}(W_2 * H_2 + b_2))$$

4. Max-Pooling Layer 2:

$$\text{Max-Pooling: } H_4 = \text{MaxPool}(H_3)$$

5. Convolutional Layer 3:

$$\text{Convolution: } H_5 = \text{ReLU}(W_3 * H_4 + b_3)$$

6. Convolutional Layer 4:

$$\text{Convolution: } H_6 = \text{ReLU}(W_4 * H_5 + b_4)$$

7. Convolutional Layer 5:

$$\text{Convolution: } H_7 = \text{ReLU}(W_5 * H_6 + b_5)$$

8. Max-Pooling Layer 3:

$$\text{Max-Pooling: } H_8 = \text{MaxPool}(H_7)$$

9. Fully Connected Layer 1 (Flattening):

$$\text{Linear Transformation: } H_9 = \text{ReLU}(W_6 * H_8 + b_6)$$

10. Fully Connected Layer 2:

$$\text{Linear Transformation: } H_{10} = \text{ReLU}(W_7 * H_9 + b_7)$$

Output Fully Connected Layer 3:

$$\text{Linear Transformation: } Y = W_8 * H_{10} + b_8$$

- Where Y represents the output logits, which can be used for coffee classification.

Algorithm 2 The CSPDarknet53 architecture for coffee image dataset

Input Data: The coffee input image as X

CSPDarknet53 architecture

1. Convolutional Layers: CSPDarknet53 consists of multiple convolutional layers, typically represented as:

$$H_l = \text{Conv}(H_{l-1}, W_l) + b_l$$

where H_l represents the feature maps at layer l ,
Conv denotes the convolution operation with weights W_l and biases b_l

2. Residual Connections: CSPDarknet53 include residual connections that enable the network to learn residual functions:

$$H_l = F(H_{l-1}, W_l) + H_{l-1}$$

where F represents the residual mapping function.

3. Pooling Layers: Max-pooling layers are used for downsampling the feature maps.

Output: Fully Connected Layers: The final layers of the network are typically fully connected layers, which can be represented as:

$$Y = W_{fc} \cdot \text{Flatten}(H_{last}) + b_{fc}$$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights, and biases of the fully connected layer by using the softmax activation function is often applied to the logits to obtain class probabilities:

$$P(\text{Class} = i) = \frac{e^{Y_i}}{\sum_{j=1}^4 e^{Y_j}}$$

3.2.3 Darknet-53 architecture

Darknet-53, a 53-layer deep neural network developed by Joseph Redmon and Ali Farhadi, is a highly accurate real-time object detection method used in various applications.

Its training process is time-consuming and computationally expensive. Algorithm 3 illustrates the detailed steps for coffee dataset with Darknet-53 with architecture equations [62].

Algorithm 3 The Darknet-53 architecture for coffee image dataset

Input Data: The coffee input image as X

Darknet-53 architecture

1. Convolutional Layers: Darknet-53 consists of multiple convolutional layers, typically represented as:

$$H_l = \text{Conv}(H_{l-1}, W_l) + b_l$$

where H_l represents the feature maps at layer l , Conv denotes the convolution operation with weights W_l and biases b_l .

2. Residual Connections: Darknet-53 includes residual connections that enable the network to learn residual functions:

$$H_l = F(H_{l-1}, W_l) + H_{l-1}$$

where F represents the residual mapping function.

3. Pooling Layers: Max-pooling layers are used for down-sampling the feature maps.

Fully Connected Layers (Output): The final layers of the network are typically fully connected layers, which can be represented as: $Y = W_{fc} \cdot \text{Flatten}(H_{last}) + b_{fc}$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer.

3.2.4 DenseNet architecture

DenseNet is a feed-forward convolutional neural network architecture, offering improved parameter efficiency, reduced parameter number, and accuracy, making it ideal for image classification tasks with fewer parameters. It despite its limitations like increased memory usage and slower training times, remains a powerful image classification tool with numerous advantages over traditional CNNs. Algorithm 4 illustrates the detailed steps for coffee dataset with DenseNet with architecture equations [63].

3.2.5 EfficientNet architecture

Google AI's EfficientNet is an advanced CNN architecture with improved accuracy, faster training times, and smaller model sizes. It features AutoML for task-specific hyperparameter search, but its complexity is a disadvantage. EfficientNet, despite its complexity and high computational requirements, is an effective architecture for tasks like image classification, object detection, and natural language processing, offering high accuracy and efficiency, making it an attractive choice for various applications. Algorithm 5 illustrates the detailed steps for coffee dataset with EfficientNet with architecture equations [64].

Algorithm 4 The DenseNet architecture for coffee image dataset

Input Data: The coffee input image as X

DenseNet Architecture

1. Convolutional Layers (Dense Blocks): DenseNet consists of multiple dense blocks, typically represented as:

$$H_l = \text{Concat}(H_{l-1}, \text{Conv}(H_{l-1}, W_l) + b_l)$$

where H_l represents the feature maps at layer l , Conv denotes the convolution operation with weights W_l and biases b_l , and Concat represents concatenation along the channel axis.

2. Transition Layers: Transition layers are used to reduce the spatial dimensions of feature maps and control the number of channels before entering the next dense block. They can be represented as:

$$H_l = \text{Conv}(H_{l-1}, W_l) + b_l$$
$$H_l = \text{AvgPool}(H_l)$$

where AvgPool represents average pooling to down-sample the feature maps.

3. Fully Connected Layers (Output): The final layers of the network are typically fully connected layers, which can be represented as:

$$Y = W_{fc} \cdot \text{Flatten}(H_{last}) + b_{fc}$$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer and Flatten denotes the operation that flattens the feature maps from the last dense block.

Algorithm 5 The EfficientNet architecture for coffee image dataset

Input Data: The coffee input image as X

EfficientNet Architecture

1. Convolutional Blocks: EfficientNet consists of multiple convolutional blocks. Each block typically involves depthwise separable convolutions and non-linear activations like Swish:

$$H_l = \text{DepthwiseSeparableConv}(H_{l-1}, W_l) \cdot \text{Swish}(H_l)$$

where H_l represents the feature maps at layer l , $\text{DepthwiseSeparableConv}$ denotes depthwise separable convolutional operations with weights W_l , and Swish represents the Swish activation function.

2. Global Average Pooling: After the convolutional blocks, global average pooling is applied to reduce the spatial dimensions of the feature maps:

$$H_l = \text{GlobalAvg}(H_l)$$

Fully Connected Layer (Output): The final layers of the network are typically fully connected layers, which can be represented as:

$$Y = W_{fc} \cdot H_{last} + b_{fc}$$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer.

3.2.6 GoogLeNet architecture

GoogLeNet uses an Inception module for multiple filters, improving accuracy in input images. However, it has high computational cost, training difficulties due to large layers, and requires large data. Furthermore, it is not suitable for real-time applications due to its complexity and slow inference time. Algorithm 6 illustrates the detailed steps for coffee dataset with GoogLeNet with architecture equations [65].

Algorithm 6 The GoogLeNet architecture for coffee image dataset

3.2.7 HRNet architecture

HRNet is a DL architecture enhancing image recognition accuracy through hierarchical feature representation, capturing high-level semantic information, robustness to noise, and scalability but faces challenges like large training data requirements and hyperparameter optimization difficulties. It, despite its limitations in handling complex scenes and data augmentation techniques, remains an attractive choice for image recognition tasks due to its high accuracy and scalability. Algorithm 7 illustrates the detailed steps for coffee dataset with HRNet with architecture equations [66].

Input Data: The coffee input image as X

GoogLeNet Architecture

1. Inception Module:

- An inception module typically consists of four parallel paths for feature extraction:

a. 1x1 Convolution Path:

$$H_{1 \times 1} = \text{Conv}_{1 \times 1}(X, W_{1 \times 1}) + b_{1 \times 1}$$

where $\text{Conv}_{1 \times 1}$ denotes 1x1 convolution with weights $W_{1 \times 1}$ and biases $b_{1 \times 1}$.

b. 1x1 Convolution followed by 3x3 Convolution Path:

$$H_{1 \times 1 - 3 \times 3} = \text{Conv}_{1 \times 1}(X, W_{1 \times 1 - 3 \times 3}) + b_{1 \times 1 - 3 \times 3}$$

$$H_{3 \times 3} = \text{Conv}_{3 \times 3}(H_{1 \times 1 - 3 \times 3}, W_{3 \times 3}) + b_{3 \times 3}$$

where $\text{Conv}_{1 \times 1}$ and $\text{Conv}_{3 \times 3}$ are 1x1 and 3x3 convolutions, respectively.

c. Convolution followed by 5x5 Convolution Path:

$$H_{1 \times 1 - 5 \times 5} = \text{Conv}_{1 \times 1}(X, W_{1 \times 1 - 5 \times 5}) + b_{1 \times 1 - 5 \times 5}$$

$$H_{5 \times 5} = \text{Conv}_{5 \times 5}(H_{1 \times 1 - 5 \times 5}, W_{5 \times 5}) + b_{5 \times 5}$$

where $\text{Conv}_{1 \times 1}$ and $\text{Conv}_{5 \times 5}$ are 1x1 and 5x5 convolutions, respectively.

d. Max-Pooling Path: $H_{\text{maxpool}} = \text{MaxPool}(X)$

- After feature extraction from each path, the results are concatenated along the channel axis to form the output of the inception module.

2. Multiple Inception Modules:

- GoogLeNet uses multiple inception modules in sequence to extract hierarchical features at different scales and complexities.

Fully Connected Layer (Output):

- The final output of the inception modules is typically connected to a fully connected layer for classification:

$$Y = W_{fc} \cdot H_{last} + b_{fc}$$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer.

Algorithm 7 The HRNet architecture for coffee image dataset

coffee dataset with Residual Network with architecture equations [67].

Input Data: The coffee input image as X

HRNet Architecture

1. High-Resolution Convolution Blocks:

- The feature maps as H_l for different scales l
- Each convolution block involves a series of operations, including convolution, batch normalization, and activation functions like ReLU. These operations can be represented as:

$$H_l = \text{ReLU}(\text{BN}(\text{Conv}(H_l, w_l) + H_l)$$

where H_l represents the feature maps at scale l , Conv denotes the convolution operation with weights w_l and biases b_l , BN represents batch normalization, and ReLU represents the rectified linear unit activation function.

2. High-Resolution Feature Fusion:

The fusion process can be represented as:

$$H_{\text{fusion}} = \text{concat}(\text{BN} + H_l)$$

where H_{fusion} is the fused feature representation, and concat denotes concatenation along the channel axis.

Fully Connected Layer (Output):

- The fused high-resolution feature representation is typically connected to a fully connected layer for classification:

$$Y = W_{fc} \cdot H_{\text{last}} + b_{fc}$$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer.

3.2.8 Residual network

ResNet-50 is a complex architecture that can learn complex features from large datasets with fewer parameters, but it can be computationally expensive and difficult to interpret. Additionally, ResNet-50 may not be suitable for certain types of tasks such as image generation or natural language processing due to its limited capacity for learning abstract features. Algorithm 8 illustrates the detailed steps for

3.2.9 VGG architecture model

VGG is a widely used technique for object detection, image segmentation, and facial recognition, renowned for its high accuracy in recognizing complex patterns in images. VGG has limitations, including the need for large amounts of data for training and its computational cost due to its numerous parameters. Algorithm 9 illustrates the detailed steps for coffee dataset with VGG Network with architecture equations [68].

Algorithm 8 The Residual architecture for coffee image dataset

Input Data: The input image as X .

VGG Architecture

1. Convolutional Layers (Basic Building Block):

- ResNet-50 consists of multiple basic building, each block can be represented as:

$$H_l = \text{ReLU} = (\text{BN}(\text{Conv}(H_{l-1}, W_l) + b_l)$$

where H_l represents the feature maps at layer l , Conv denotes the convolution operation with weights W_l and biases b_l , BN represents batch normalization.

2. Residual Blocks:

- ResNet-50 introduces residual blocks, which contain multiple basic building blocks in a residual fashion. A residual block can be represented as:

$$H_l = \text{Block}(H_{l-1}, \{W_l\})$$

where Block represents a sequence of basic building blocks, and $\{W_l\}$ denotes the set of weights for the blocks.

3. Pooling Layers:

- Pooling layers are used for down-sampling the feature maps.

Fully Connected Layer (Output):

- The final layers of ResNet-50 are typically fully connected layers, which can be represented as: $Y = W_{fc} \cdot \text{Flatten}(H_{last}) + b_{fc}$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer.

Algorithm 9 The VGG architecture for coffee image dataset

Input Data: The input image as X .

VGG Architecture

1. Convolutional Layers (Basic Building Block):

- ResNet-50 consists of multiple basic building, each block can be represented as:

$$H_l = \text{ReLU} = (\text{BN}(\text{Conv}(H_{l-1}, W_l) + b_l)$$

where H_l represents the feature maps at layer l , Conv denotes the convolution operation with weights W_l and biases b_l , BN represents batch normalization.

2. Residual Blocks:

- ResNet-50 introduces residual blocks, which contain multiple basic building blocks in a residual fashion. A residual block can be represented as:

$$H_l = \text{Block}(H_{l-1}, \{W_l\})$$

where Block represents a sequence of basic building blocks, and $\{W_l\}$ denotes the set of weights for the blocks.

3. Pooling Layers:

- Pooling layers are used for down-sampling the feature maps.

Fully Connected Layer (Output):

- The final layers of ResNet-50 are typically fully connected layers, which can be represented as: $Y = W_{fc} \cdot \text{Flatten}(H_{last}) + b_{fc}$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer.

3.2.10 MobileNetV2 architecture

MobileNetV2 is computationally efficient and accurate model suitable for mobile applications, with smaller datasets, making them ideal for real-world scenarios due to their inverted residuals and linear bottlenecks. MobileNetV1 and MobileNetV2 have limitations in scaling, accuracy, computational efficiency, and tuning, and require more tuning for optimal performance in complex tasks or datasets. Algorithm 9 illustrates the detailed steps for coffee dataset with MobileNetV2 Network with architecture equations [68].

Algorithm 10 The MobileNetV2 architecture for coffee image dataset

standardize the hyper-parameters across the experiments, using the following hyper-parameters, which are described in Table 3. DL has significantly advanced in many research areas.

3.4 Data classification

The number of the classification output layer is equal to the number of the classes. Then, each output has a different probability for the input image because these kinds of models can automatically learn features during the training stage; then, the model picks the highest probability as its prediction of the class. This phase determines which dis-

Input Data: The input image as X .

MobileNetV2 Architecture

1. Depthwise Separable Convolution:

- MobileNetV2 primarily uses depthwise separable convolutions. The operation can be represented as:

$$H_l = \text{DepthwiseConv}(H_{l-1}, W_{\text{depthwise}})$$

where H_l represents the feature maps at layer l , DepthwiseConv denotes depthwise convolution with weights $W_{\text{depthwise}}$.

2. Batch Normalization and ReLU Activation:

- After each depthwise separable convolution, batch normalization and ReLU activation functions are applied:

$$H_l = \text{ReLU}(\text{BN}(H_l))$$

where BN represents batch normalization, and ReLU represents the rectified linear unit activation function.

3. Pointwise Convolution:

- Pointwise convolutions (1x1 convolutions) are used to increase the number of channels in the feature maps:

$$H_l = \text{PointwiseConv}(H_l, W_{\text{Pointwise}})$$

where PointwiseConv denotes 1x1 convolution with weights $W_{\text{Pointwise}}$.

4. Global Average Pooling:

- MobileNetV2 often ends with global average pooling to reduce spatial dimensions:

$$H_{\text{gap}} = \text{GlobalAvgPool}(H_l)$$

Fully Connected Layer (Output):

- The output of global average pooling is connected to a fully connected layer for classification:

$$Y = W_{fc} \cdot \text{Flatten}(H_{\text{last}}) + b_{fc}$$

where Y represents the output logits for the four coffee classes, W_{fc} and b_{fc} are the weights and biases of the fully connected layer and *flatten* denotes the operation that flattens the feature maps.

3.3 CNN settings

The CNN settings usually consist of a series of specific elements, which are the ones that present the variations in the different architectures. To allow a fair comparison between the experiments, an attempt was also made to

ease is present in the leaf using the pre-trained set. Figure 3 shows the general structure for our proposed work.

4 Result and discussion

In this section, we evaluate the performance of various neural network architectures for coffee classification. Alex Net, SPDarknet53, Darknet, DenseNet121, EfficientNet, GoogleNet, HRNet, LeNet, MobileNetV2, ResNet-50, and VGG-19 showed impressive accuracy and efficiency.

The study emphasizes the importance of choosing the right neural network for specific tasks, considering factors like model complexity and computational efficiency. The experiments highlight the diverse capabilities of these architectures.

4.1 The pre-trained architecture

In this section, we compare the proposed work architecture that used for coffee classification.

4.1.1 Alex Net

AlexNet's coffee classification model faces overfitting risk, particularly in smaller datasets, due to its complex architecture and large number of parameters, causing rapid convergence and reduced accuracy in real-world scenarios. Figure 4 shows the common learning curves that appear to be the drawbacks of using it.

4.1.2 LeNet architecture

LeNet's shallow architecture may hinder its ability to capture complex features in coffee images, causing slower convergence and limited pattern learning, potentially impacting classification accuracy, especially with fine-grained coffee varieties. Figure 5 shows the common learning curves that appear to be the drawbacks of using it.

4.1.3 HRNet architecture

HRNet's high-resolution architecture increases computational intensity due to numerous parameters and slow convergence in the learning curve, requiring substantial resources for training and inference. Figure 6 shows the common learning curves that appear to be the drawbacks of using it.

4.1.4 Google net architecture

GoogleNet's intricate architecture, utilizing multiple inception modules with varying kernel sizes, can increase training time and slow convergence in the learning curve

compared to simpler models. Figure 7 shows the common learning curves that appear to be the drawbacks of using it.

4.1.5 Mobile V2 net architecture

MobileNet architectures are lightweight and efficient, reducing parameters for resource-constrained devices. However, this can limit their ability to accurately classify complex tasks like coffee classification. Figure 8 shows the common learning curves that appear to be the drawbacks of using it.

4.1.6 ResNet (50) architecture

ResNet-50's deep architecture demands significant computational resources for training, potentially leading to extended training times and high memory usage, making it less practical for applications with limited computational capabilities. Figure 9 shows the common learning curves that appear to be the drawbacks of using it.

4.1.7 VGG architecture

ResNet-50's deep architecture demands significant computational resources for training, potentially leading to extended training times and high memory usage, making it less practical for applications with limited computational capabilities. Figure 10 shows the common learning curves that appear to be the drawbacks of using it.

4.1.8 Efficient architecture

EfficientNet models, with fewer parameters, can be efficient in resource-constrained environments but may struggle with complex coffee image datasets due to limited learning capacity. Figure 11 shows the common learning curves that appear to be the drawbacks of using it.

4.1.9 Darknet architecture

Darknet's deep architecture and capacity can require significant computational resources for training and inference, making it less practical for resource-constrained environments or real-time applications. Figure 12 shows the common learning curves that appear to be the drawbacks of using it.

Table 3 The general overview comparison between the used pre-trained architecture

Model	Advantages	Drawbacks
AlexNet	High Sensitivity (0.9346) indicates strong ability to identify positive cases. High Precision (0.9615) and Accuracy (0.9575) contribute to reliable performance	Moderate F1 Score (0.9479) suggests a trade-off between precision and recall
LeNet	Simple architecture and relatively lightweight	Suboptimal performance in Sensitivity (0.5556) and Precision (0.6061). Lower F1 Score (0.5797)
HRNet	High Sensitivity (0.9615) and Precision (0.9524) showcase robust performance Well-balanced F1 Score (0.9569) indicates strong classification capabilities	Requires higher computational resources compared to simpler architectures
GoogleNet	Near-perfect scores in Sensitivity (0.9804) and Precision (0.9709) demonstrate excellence. High F1 Score (0.9756) suggests a well-balanced trade-off between precision and recall	Potentially higher computational complexity
MobileNetV2	Exceptional performance across all measures, including Sensitivity (0.9901) and Precision (0.9901). High Accuracy (0.9925) and F1 Score (0.9901) highlight its efficiency	Required fine-tuning for image classification task
ResNet (50)	Wide applicability and success in various computer vision tasks	Lower performance in Sensitivity (0.5435) and Precision (0.5714) Moderate F1 Score (0.5571) indicates a trade-off between precision and recall
VGG	Perfect scores in Specificity (0.9917), Precision (0.9924), and Accuracy (1.0000) showcase reliability. Exceptionally high F1 Score (0.9962) indicates superior classification capabilities	Potentially higher computational complexity
EfficientNet	Solid performance with high Sensitivity (0.9630) and Precision (0.9559). Balanced F1 Score (0.9594) and high Accuracy (0.9579) contribute to effectiveness	Required additional resources for training
Darknet	Consistent high scores across all measures, with strength in Specificity (0.9836) and Precision (0.9848). High F1 Score (0.9811) indicates robust performance	Potentially higher computational complexity
DenseNet	Notable scores in Sensitivity (0.9701), Specificity (0.9836), and Precision (0.9848) Balanced F1 Score (0.9774) suggests strong classification capabilities	required fine-tuning for image classification task

4.1.10 DenseNet architecture

DenseNet's architecture, featuring dense skip connections, can be challenging to understand and optimize due to complex hyperparameter configuration and debugging, and show a slower learning curve. Figure 13 shows the common learning curves that appear to be the drawbacks of using it.

The evaluation of deep learning models for coffee classification reveals AlexNet, LeNet, HRNet, GoogleNet, MobileNetV2, ResNet, VGG, EfficientNet, Darknet, and DenseNet as robust models with high sensitivity, precision, and accuracy, but with moderate F1 Scores and potential computational complexity. These insights help in selecting suitable models for coffee classification, as illustrated in Table 3 and Table 4.

The evaluation results provide a nuanced understanding of each architecture's advantages and drawbacks. While some models excel in specific aspects, considerations of computational complexity and task-specific requirements are crucial for informed model selection. These insights contribute valuable guidance for practitioners seeking optimal deep learning models for classification tasks.

5 Conclusion

The global coffee industry, a vital sector for numerous nations, is currently grappling with economic instability due to fluctuations in global coffee prices. The study utilized deep learning techniques, specifically pre-trained models, to accurately predict coffee types. The selection of

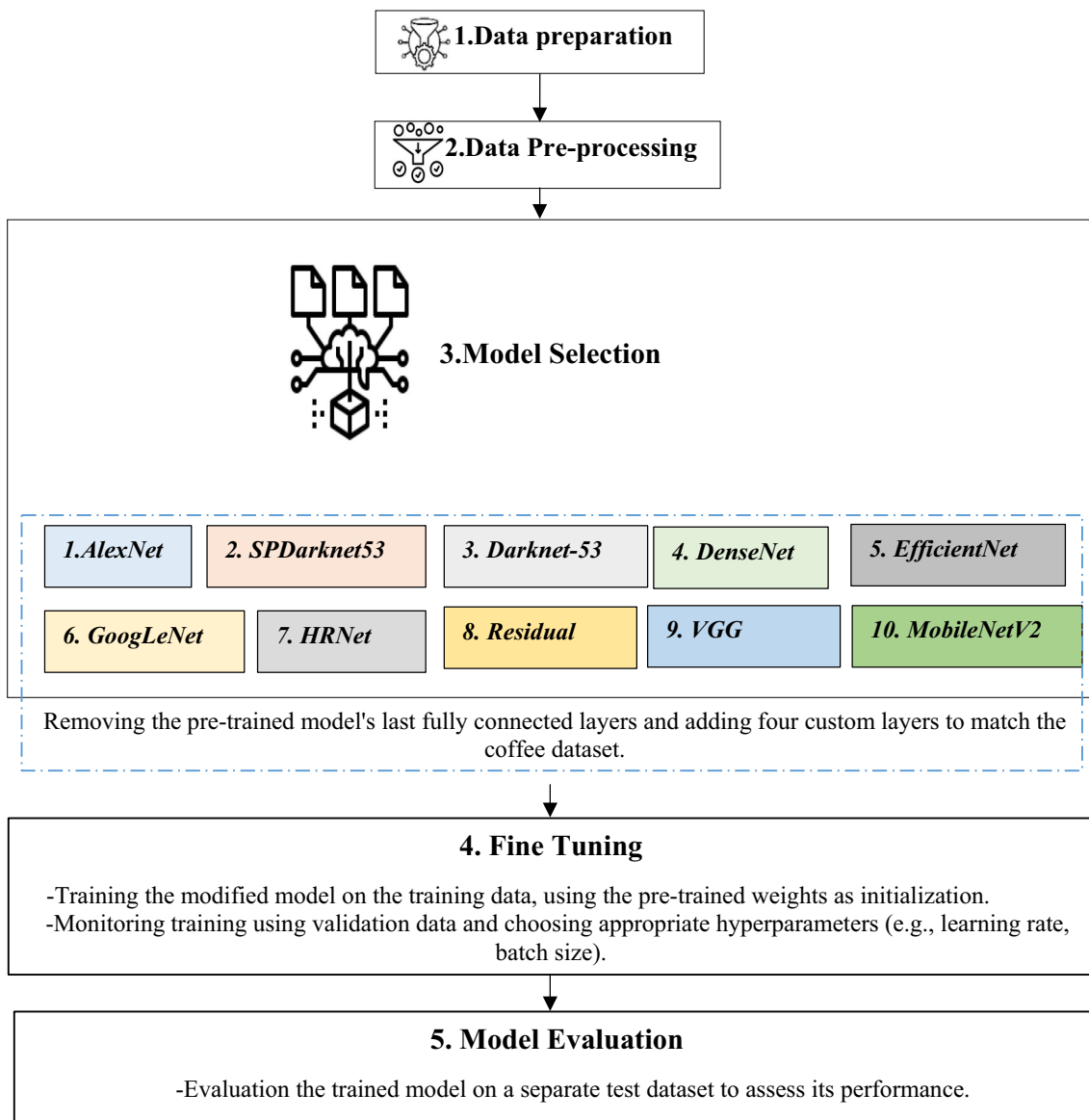


Fig. 3 The main structure for our proposed work

the best pre-trained model is crucial due to the growing demand for specialty coffee and the need for precise classification. In this study, we focus on evaluating the effectiveness of various pre-trained models through deep learning techniques. The motivation behind opting for transfer learning lies in the recognition that leveraging knowledge gained from models trained on large datasets can significantly boost the performance of models trained on smaller datasets. The increasing popularity of specialty coffee amplifies the need for accurate classification, making the selection of an optimal pre-trained model crucial. In our comprehensive comparison, we assess several well-

known pre-trained models, including AlexNet, LeNet, HRNet, Google Net, Mobile V2 Net, ResNet (50), VGG, Efficient, Darknet, and DenseNet. Through transfer learning and fine-tuning, we gauge the models’ ability to generalize to the coffee classification task illustrated in Table 5 and Fig. 14. We reveal the pivotal role of the pre-trained model choice in influencing performance, with specific models demonstrating higher accuracy and faster convergence than conventional alternatives. By employing key evaluation metrics such as sensitivity, specificity, precision, negative predictive value, accuracy, and F1 score, we provide nuanced insights into the complex landscape of

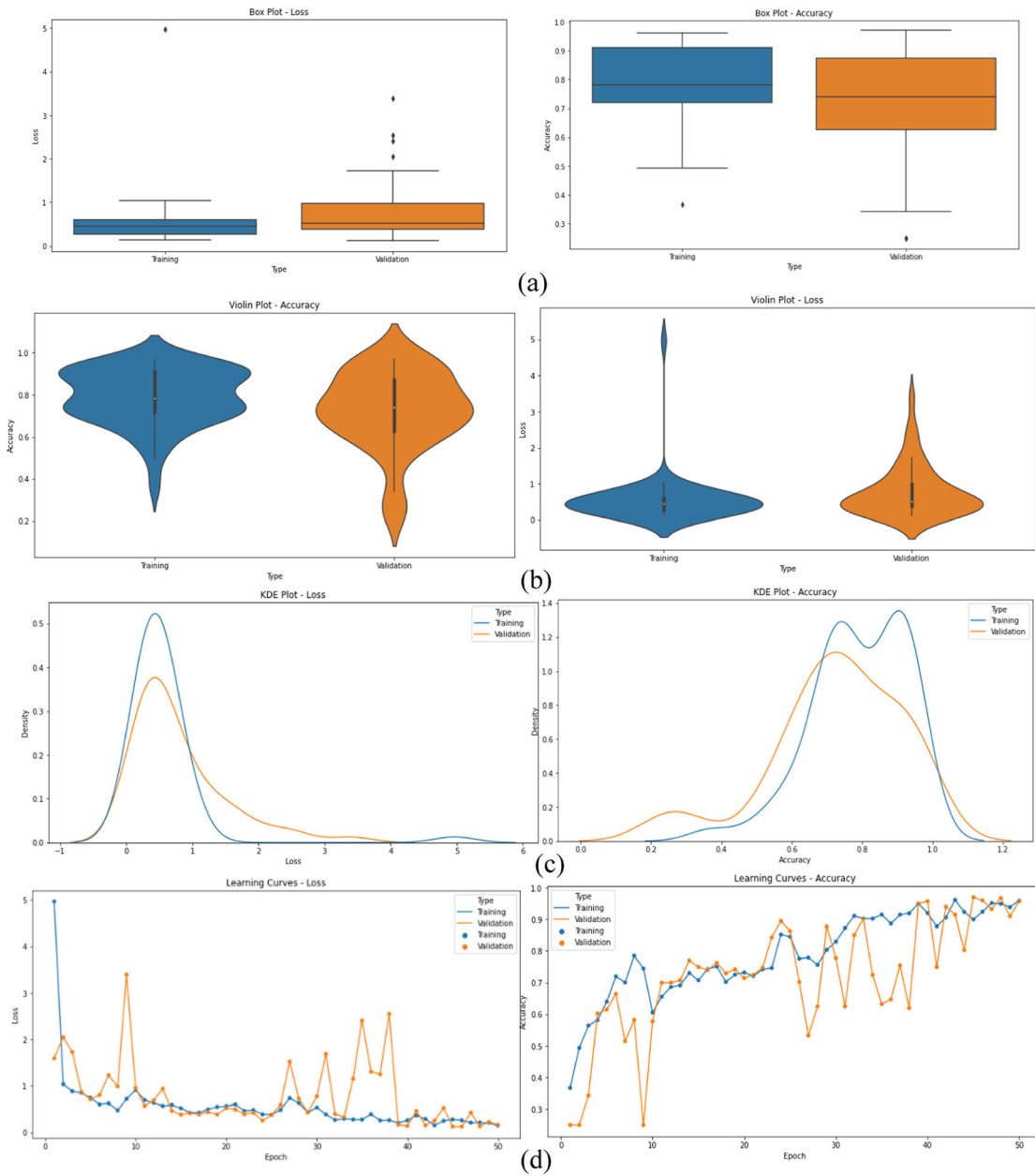


Fig. 4 a BoxPlot, b violin curve, c KDI curve, d learning curve

Fig. 5 **a** BoxPlot, **b** violin curve, **c** KDI curve, **d** learning curve

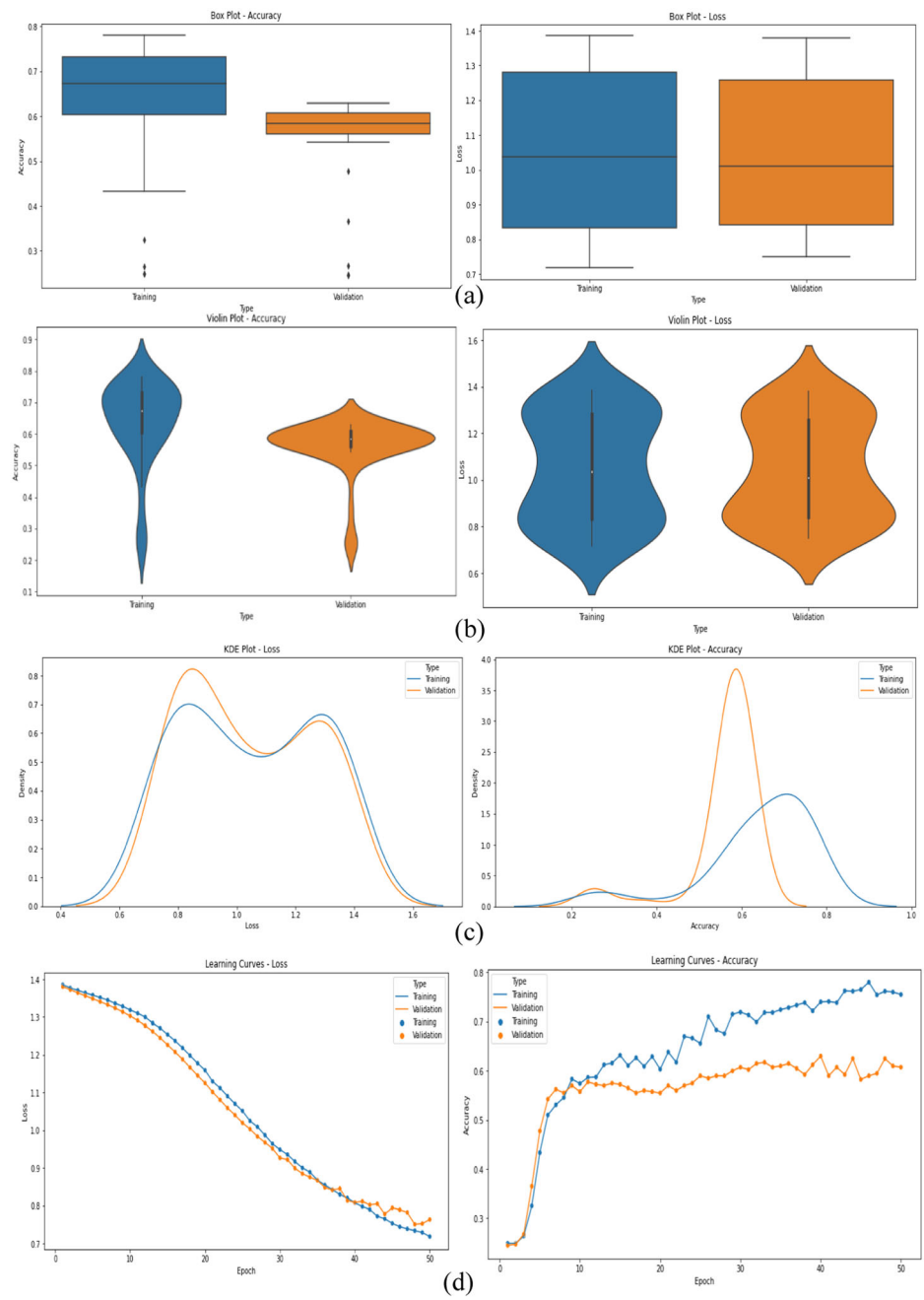
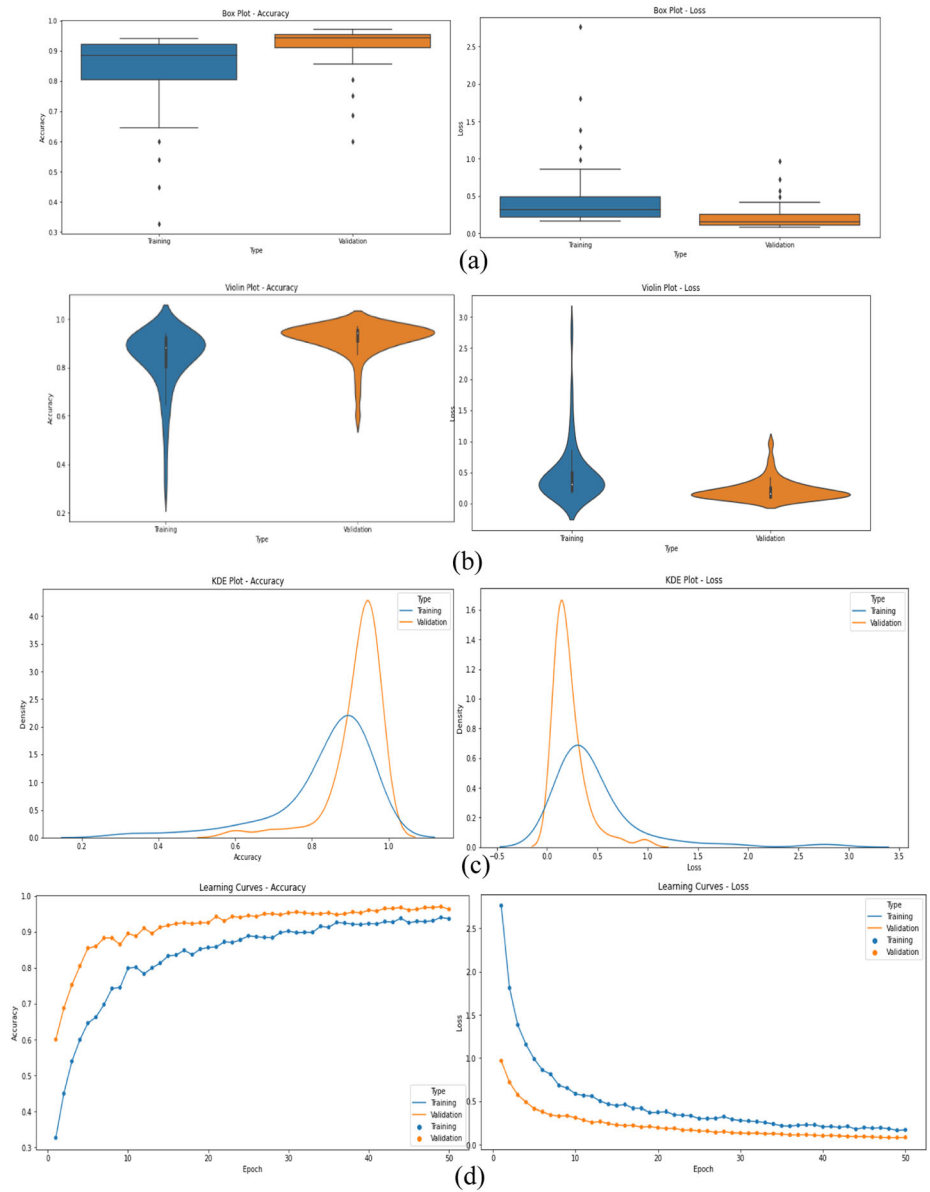


Fig. 6 **a** BoxPlot, **b** violin curve, **c** KDI curve, **d** learning curve



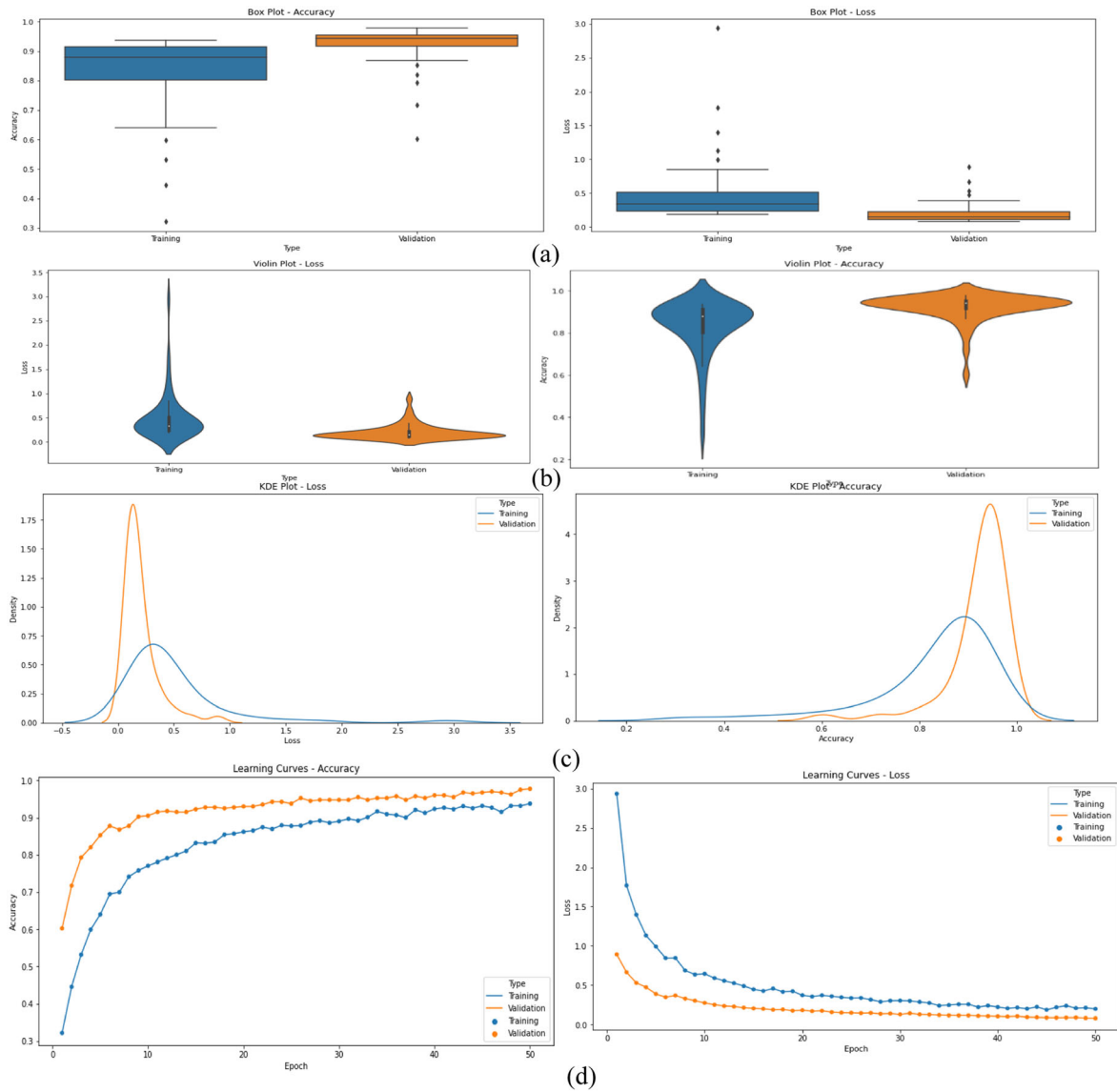


Fig. 7 a BoxPlot, b violin curve, c KDI curve, d learning curve

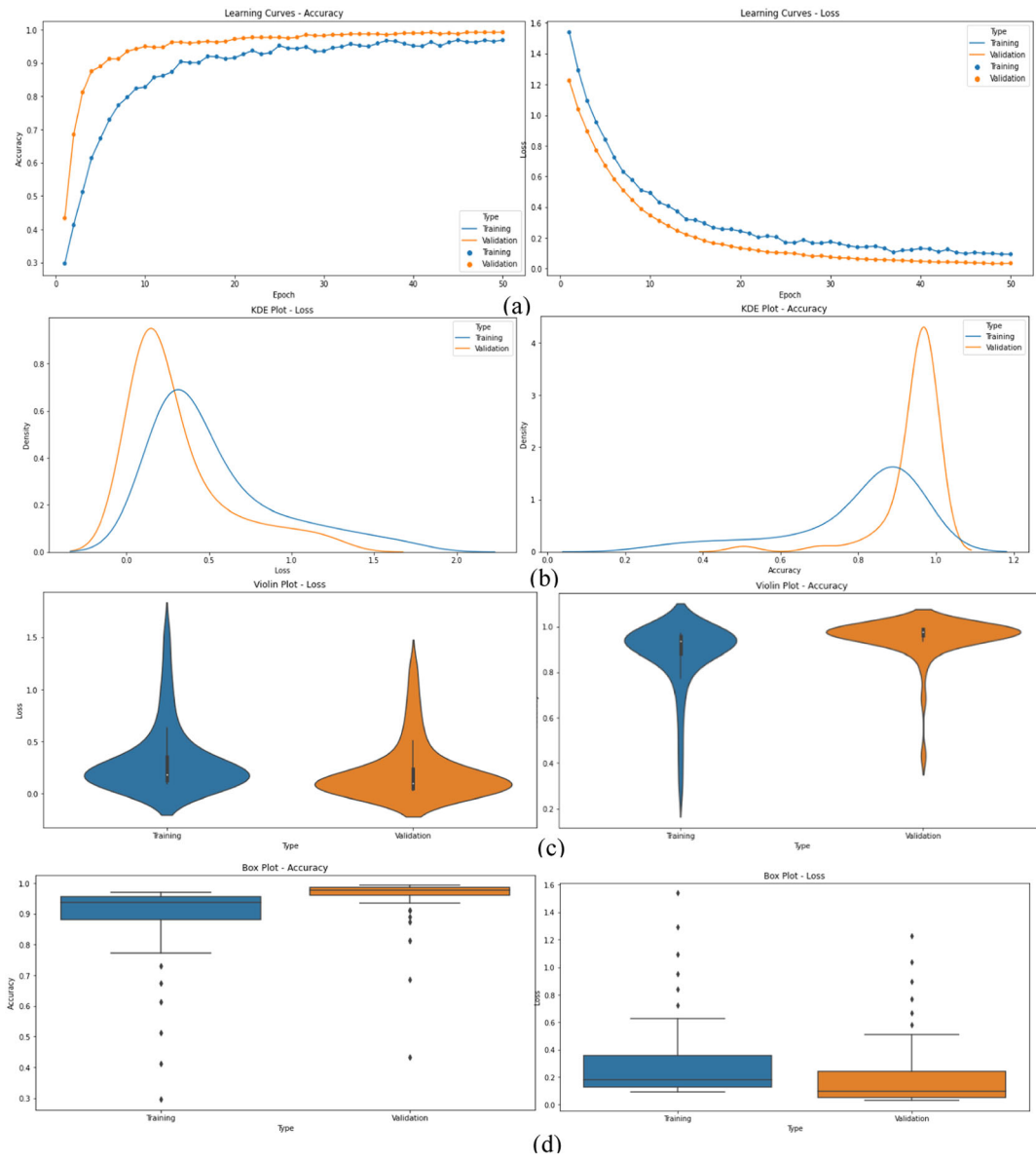


Fig. 8 a BoxPlot, b violin curve, c KDI curve, d learning curve

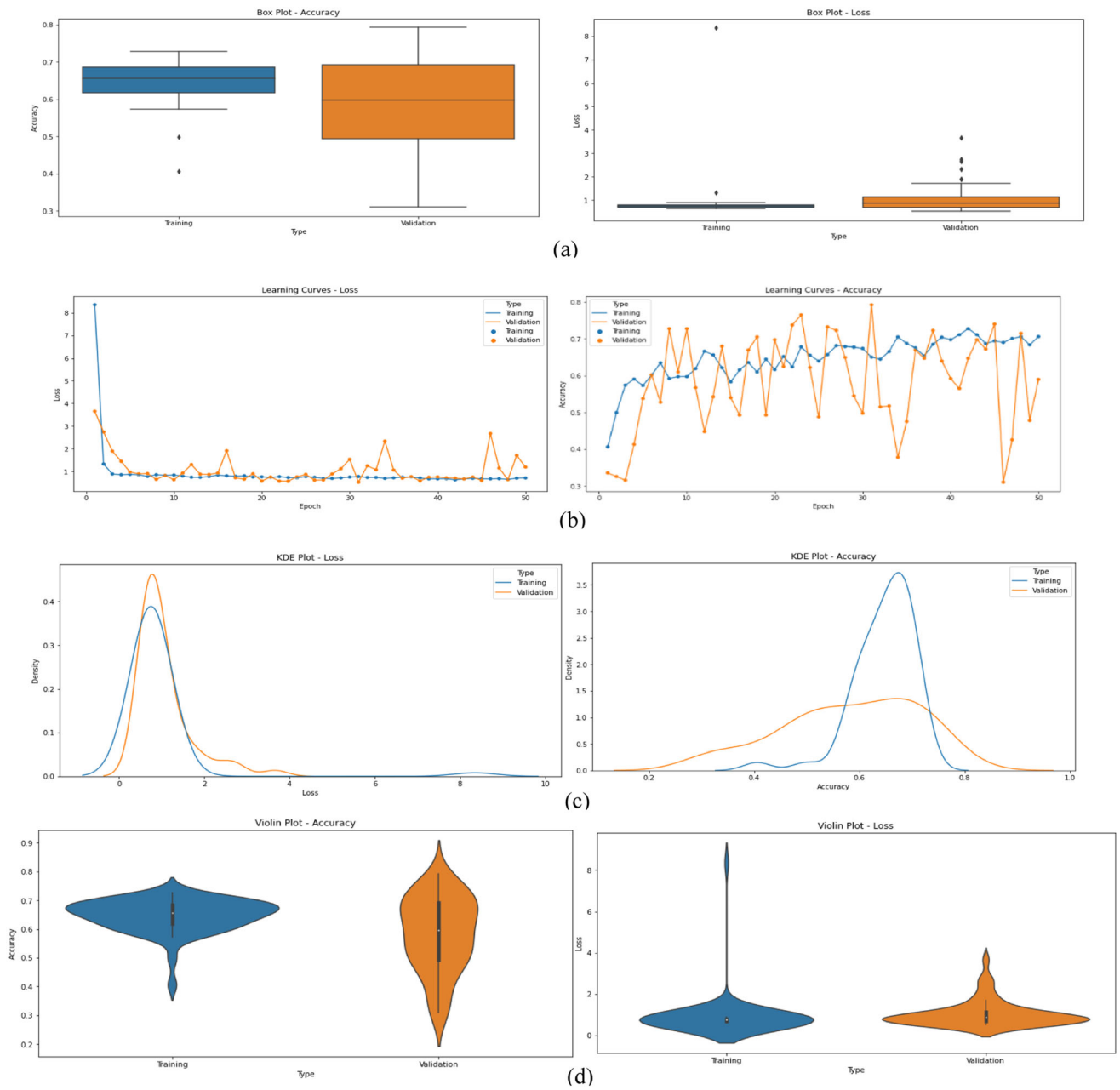


Fig. 9 a BoxPlot, b violin curve, c KDI curve, d learning curve

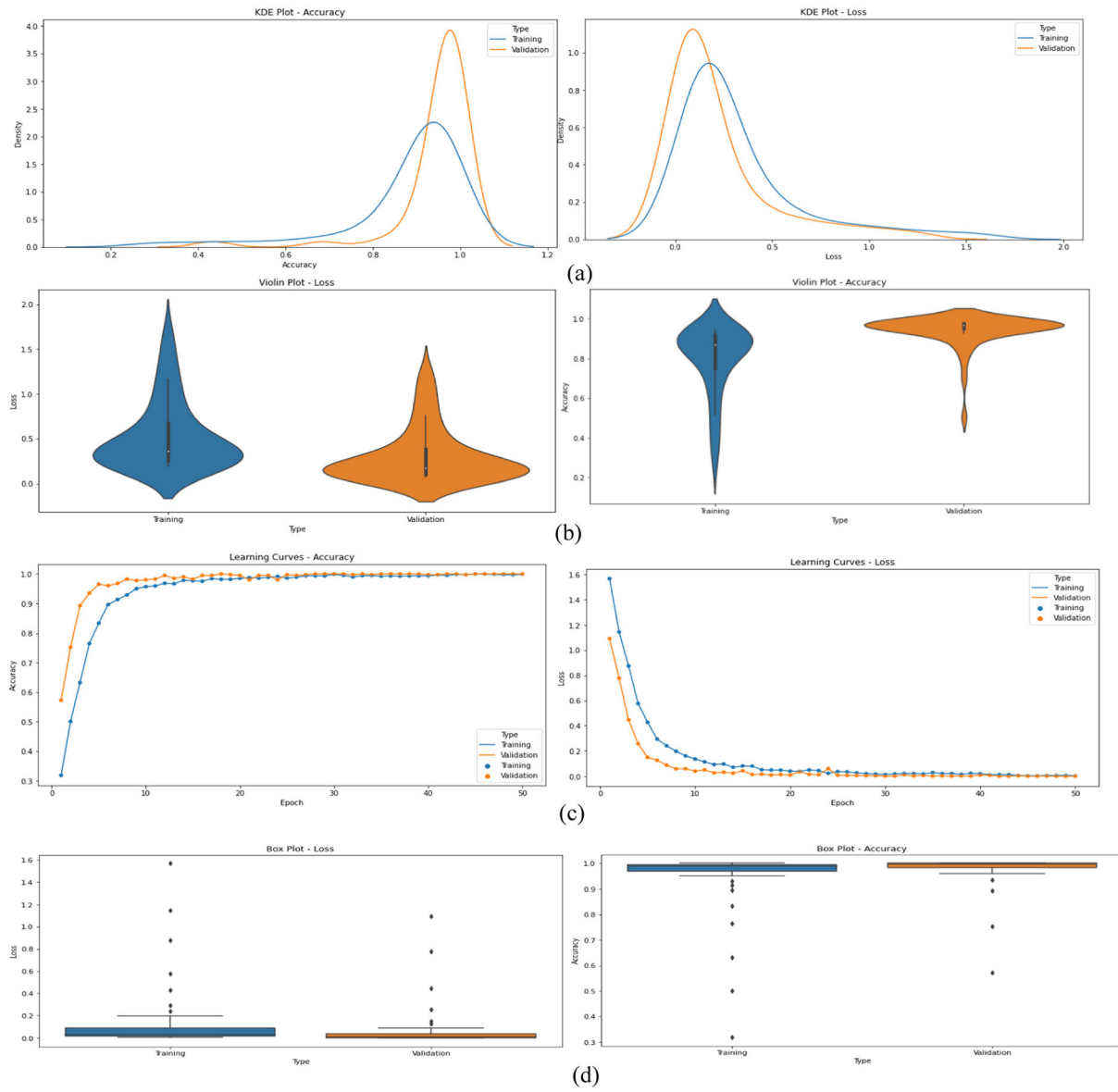


Fig. 10 a BoxPlot, b violin curve, c KDI curve, d learning curve

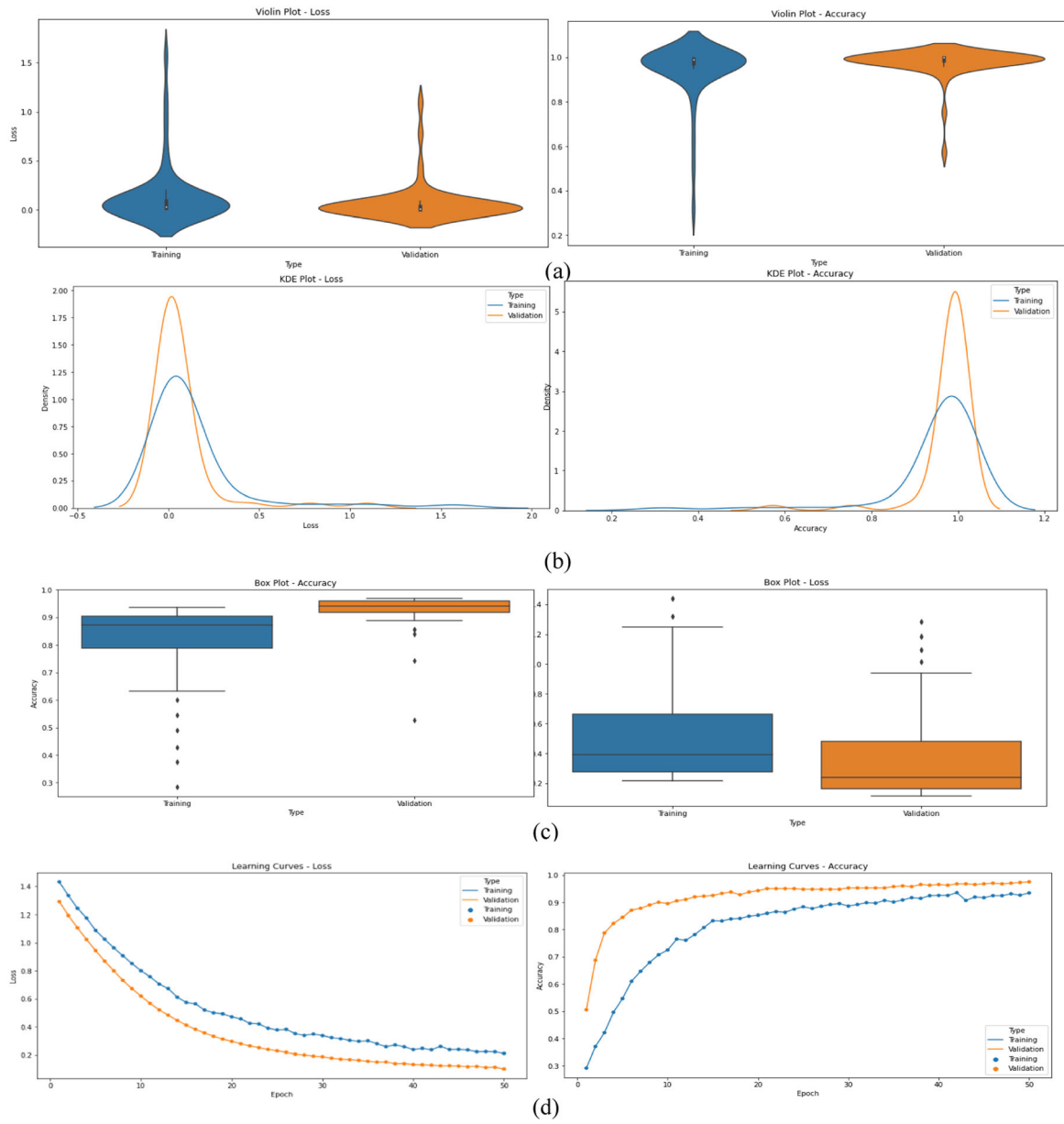


Fig. 11 a BoxPlot, b violin curve, c KDI curve, d learning curve

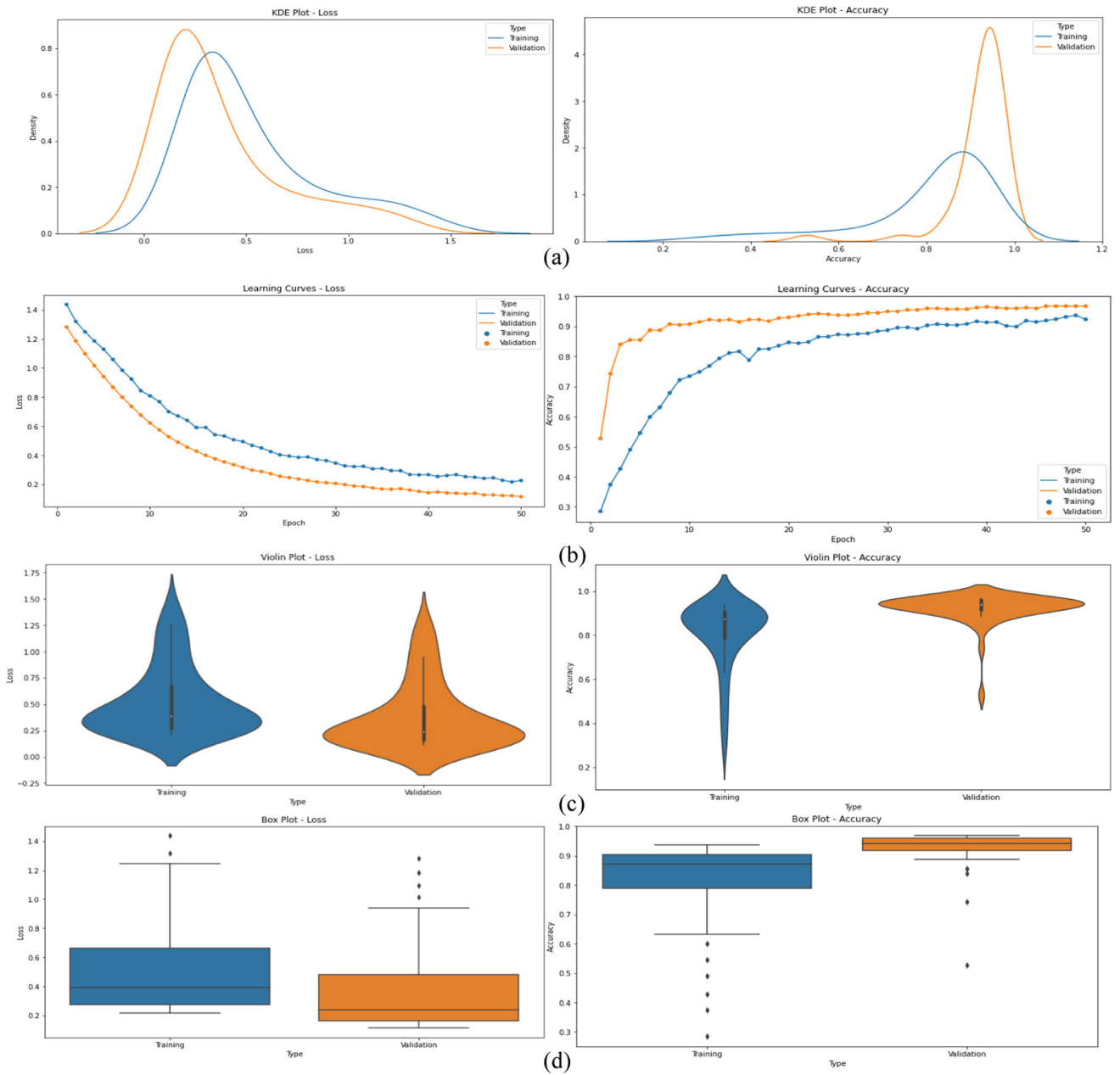


Fig. 12 a BoxPlot, b violin curve, c KDI curve, d learning curve

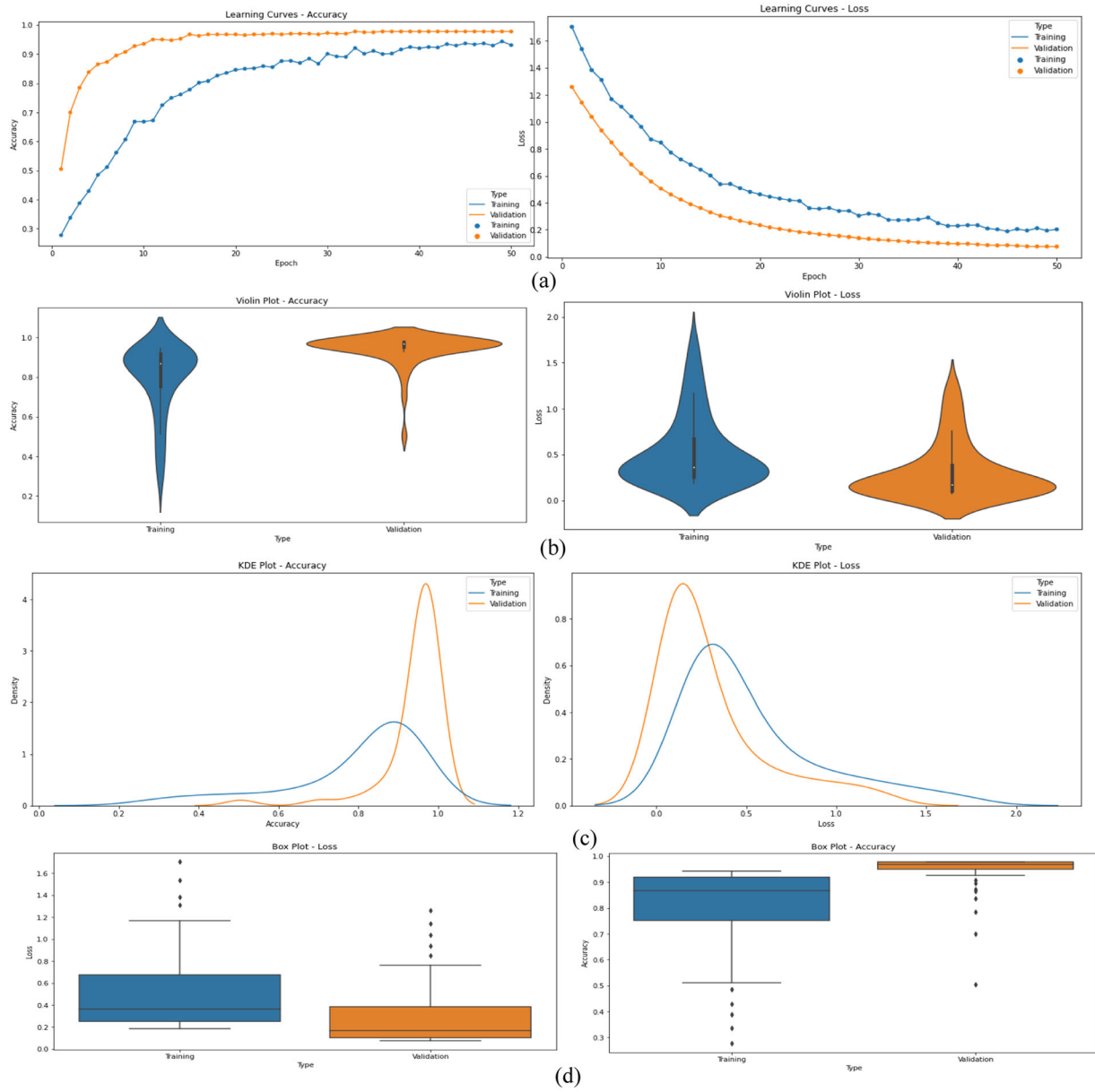


Fig. 13 a BoxPlot, b violin curve, c KDI curve, d learning curve

Table 4 Comparison of Hyperparameters for the used pre-trained architecture

Model	Learning Rate	Batch Size	Optimizer	Activation Function	Dropout Rate	Number of Layers	Kernel Size	Pooling
AlexNet	0.001	64	SGD	ReLU	0.5	8	11 × 11	Max
LeNet	0.01	32	Adam	Sigmoid	0.25	5	5 × 5	Average
HRNet	0.001	128	RMSProp	ReLU	0.3	Varies	3 × 3	Max
GoogleNet	0.001	32	Adadelata	ReLU	0.4	22	1 × 1, 3 × 3	Average
MobileNetV2	0.001	32	Adam	ReLU	0.25	53	3 × 3	Average
ResNet (50)	0.001	64	SGD	ReLU	0.3	50	7 × 7	Average
VGG	0.001	32	Adagrad	ReLU	0.5	16	3 × 3	Max
EfficientNet	0.001	64	Adam	Swish	0.2	Varies	3 × 3	Average
Darknet	0.0001	16	SGD	Leaky ReLU	0.4	19	3 × 3	Max
DenseNet	0.001	64	Adam	ReLU	0.3	121	3 × 3	Average

Table 5 The state-of-the-art comparison between the proposed work models

Architecture	Evaluation Measures					
	Sensitivity	Specificity	Precision	Negative Predictive Value	Accuracy	F1 Score
AlexNet	0.9346	0.9677	0.9615	0.9449	0.9575	0.9479
LeNet	0.5556	0.6486	0.6061	0.6000	0.6075	0.5797
HRNet	0.9615	0.9600	0.9524	0.9677	0.9650	0.9569
Google Net	0.9804	0.9756	0.9709	0.9836	0.9775	0.9756
Mobile V2 Net	0.9901	0.9917	0.9901	0.9917	0.9925	0.9901
ResNet (50)	0.5435	0.6154	0.5714	0.5882	0.5805	0.5571
VGG	1.0000	0.9917	0.9924	1.0000	1.0000	0.9962
Efficient	0.9630	0.9524	0.9559	0.9600	0.9579	0.9594
Darknet	0.9774	0.9836	0.9848	0.9756	0.9825	0.9811
DenseNet	0.9701	0.9836	0.9848	0.9677	0.9775	0.9774
Hsia et al. [20]	0.9789	N/A	0.9860	N/A	0.9838%	0.9824
Chen et al. [19]	0.9716	N/A	0.9738	N/A	N/A	0.97.21
Ke et al. [18]	0.9753	N/A	0.9648	N/A	N/A	0.98

pre-trained models. This strategic use of transfer learning and fine-tuning not only enhances the accuracy of coffee classification but also contributes to addressing economic challenges associated with global price fluctuations in coffee bean production. In addition to advancing our understanding of coffee bean production dynamics, our study has tangible implications for real-world problem-solving.

The challenges faced by coffee production in the face of global price fluctuations directly impact the economic

stability of countries reliant on this industry. As the specialty coffee market continues to grow, precise classification becomes paramount. Our comprehensive evaluation, leveraging transfer learning and fine-tuning, ensures that the selected pre-trained models can be practically applied in the coffee industry for more accurate and efficient classification. In essence, our study transcends the theoretical realm by offering a valuable resource for coffee producers, processors, and distributors. The nuances uncovered through our result evaluation metrics, including

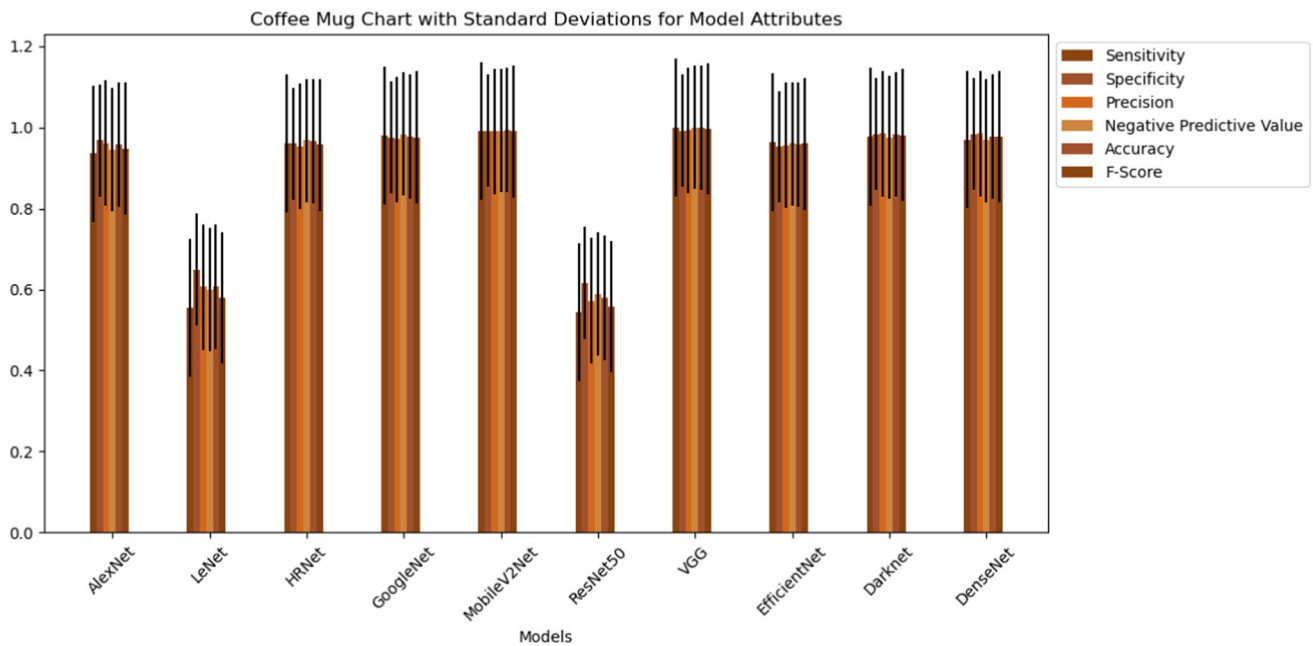


Fig. 14 The chart for with standard deviations for proposed model attributes

sensitivity, specificity, precision, negative predictive value, accuracy, and F1 score, provide actionable insights into the practical use of pre-trained models in addressing challenges faced by the coffee industry. This research not only enhances our understanding of the intricate landscape of pre-trained models but also contributes to the development of tools that can make a real impact in the coffee production sector. In the future, we will use it for the real-time classification of coffee bean images. This could potentially lead to improved efficiency in sorting and categorizing coffee beans for commercial purposes.

Funding Open access funding provided by The Science, Technology & Innovation Funding Authority (STDF) in cooperation with The Egyptian Knowledge Bank (EKB). Egyptian Knowledge Bank (EKB).

Data availability <https://www.kaggle.com/datasets/gpiosenka/coffee-bean-dataset-resized-224-x-224>.

Declarations

Conflict of interest There is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended

use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Faisal M, Leu J-S, Darmawan JT (2023) Model selection of hybrid feature fusion for coffee leaf disease classification. IEEE Access
- Salinas P, Rosaura N et al. (2021) Automated machine learning for satellite data: integrating remote sensing pre-trained models into AutoML systems. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, Cham
- Hassan E, Shams M, Hikal NA, Elmougy S (2021) Plant seedlings classification using transfer, pp 3–4
- Hassan E, Shams MY, Hikal NA, Elmougy S (2022) The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. *Multimed Tools Appl.* <https://doi.org/10.1007/s11042-022-13820-0>
- Hassan E, Shams MY, Hikal NA, Elmougy S (2022) A novel convolutional neural network model for malaria cell images classification. *Comput Mater Continua* 72(3):5889–5907. <https://doi.org/10.32604/cmc.2022.025629>
- Pradana-López S et al (2021) Deep transfer learning to verify quality and safety of ground coffee. *Food Control* 122:107801
- Chang SJ, Huang CY (2021) Deep learning model for the inspection of coffee bean defects. *Appl Sci (Switzerland)* 11(17):8226. <https://doi.org/10.3390/app11178226>
- Boa-Sorte LX, Ferraz CT, Fambrini F, Goulart RDR, Saito JH (2019) Coffee leaf disease recognition based on deep learning and texture attributes. *Procedia Comput Sci* 159:135–144. <https://doi.org/10.1016/j.procs.2019.09.168>
- Novtahaning D, Shah HA, Kang J-M (2022) Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease. *Agriculture* 12(11):1909. <https://doi.org/10.3390/agriculture12111909>

10. Velásquez D, Sánchez A, Sarmiento S, Toro M, Maiza M, Sierra B (2020) A method for detecting coffee leaf rust through wireless sensor networks, remote sensing, and deep learning: case study of the Caturra variety in Colombia. *Applied Sciences* (Switzerland) 10(2):697. <https://doi.org/10.3390/app10020697>
11. Gope HL, Fukai H, Aoki R (2022) Multi-label classification of defective green coffee bean images using efficientnet deep learning model. *Trans Asian J Sci Technol* 5
12. Ramamurthy K et al (2023) A novel deep learning architecture for disease classification in Arabica coffee plants. *Concurr Comput Pract Exp* 35(8):e7625
13. Liang C-S, Xu Z-Y, Zhou J-Y, Yang C-M, Chen J-Y (2023) Automated detection of coffee bean defects using multi-deep learning models. In: 2023 VTS Asia Pacific Wireless Communications Symposium (APWCS), Tainan city, Taiwan, pp 1–5. <https://doi.org/10.1109/APWCS60142.2023.10234059>
14. Arif MS, Mukheimer A, Asif D (2023) Enhancing the early detection of chronic kidney disease: a robust machine learning model. *Big Data Cogn Comput* 7:144. <https://doi.org/10.3390/bdcc7030144>
15. Asif D, Bibi M, Arif MS, Mukheimer A (2023) Enhancing heart disease prediction through ensemble learning techniques with hyperparameter optimization. *Algorithms* 16(6):308. <https://doi.org/10.3390/a16060308>
16. Nawaz Y, Arif MS, Shatanawi W, Nazeer A (2021) An explicit fourth-order compact numerical scheme for heat transfer of boundary layer flow. *Energies* 14(12):3396. <https://doi.org/10.3390/en14123396>
17. Nawaz Y, Arif MS, Abodayeh K (2022) An explicit-implicit numerical scheme for time fractional boundary layer flows. *Int J Numer Meth Fluids* 94(7):920–940
18. Ke LY, Chen E, Hsia CH (2023) Green coffee bean defect detection using shift-invariant features and non-local block. In: 2023 IEEE 6th international conference on knowledge innovation and invention (ICKII). IEEE, pp 430–431
19. Chen PH, Jhong SY, Hsia CH (2022) Semi-supervised learning with attention-based CNN for classification of coffee beans defect. In: 2022 IEEE international conference on consumer electronics-Taiwan. IEEE, pp 411–412
20. Hsia CH, Lee YH, Lai CF (2022) An explainable and lightweight deep convolutional neural network for quality detection of green coffee beans. *Appl Sci* 12(21):10966
21. Chavarro AF, Renza D, Ballesteros DM (2023) Influence of hyperparameters in deep learning models for coffee rust detection. *Appl Sci* 13(7):4565
22. Shibu George G, Raj Mishra P, Sinha P, Ranjan Prusty M (2023) COVID-19 detection on chest X-ray images using Homomorphic Transformation and VGG inspired deep convolutional neural network. *Biocybern Biomed Eng* 43(1):1–16. <https://doi.org/10.1016/j.bbe.2022.11.003>
23. Hassan E, Talaat FM, Hassan Z, El-Rashidy N (2023) Breast cancer detection: a survey. *Artificial intelligence for disease diagnosis and prognosis in smart healthcare*. CRC Press, Cambridge, pp 169–176
24. Hassan E, Talaat FM, Adel S, Abdelrazek S, Aziz A, Nam Y, El-Rashidy N (2023) Robust deep learning model for black fungus detection based on Gabor filter and transfer learning. *Comput Syst Sci Eng* 47(2):1507–1525
25. Hassan E, Hossain MS, Saber A, Elmougy S, Ghoneim A, Muhammad G (2024) A quantum convolutional network and ResNet (50)-based classification architecture for the MNIST medical dataset. *Biomed Signal Process Control* 87:105560
26. Hassan E, Elmougy S, Ibraheem MR, Hossain MS, AlMutib K, Ghoneim A, AlQahtani SA, Talaat FM (2023) Enhanced deep learning model for classification of retinal optical coherence tomography images. *Sensors* 23(12):5393. <https://doi.org/10.3390/s23125393>
27. Gamel SA, Hassan E, El-Rashidy N, Talaat FM (2023) Exploring the effects of pandemics on transportation through correlations and deep learning techniques. *Multimedia Tools Appl* 1–22
28. Elmougy S, Hikal NA, Hassan E (2021) An efficient technique for CT scan images classification of COVID-19. *J Intell Fuzzy Syst* 40(3):5225–5238
29. Hassan E, Shams MY, Hikal NA, Elmougy S (2023) COVID-19 diagnosis-based deep learning approaches for COVIDx dataset: a preliminary survey. In: *Artificial intelligence for disease diagnosis and prognosis in smart healthcare*, p 107
30. Hu Z, Wang Z, Jin Y, Hou W (2023) VGG-TSwinformer: Transformer-based deep learning model for early Alzheimer's disease prediction. *Comput Methods Programs Biomed* 229:107291. <https://doi.org/10.1016/J.CMPB.2022.107291>
31. Rodrigues LF et al (2022) Optimizing a deep residual neural network with genetic algorithm for acute lymphoblastic leukemia classification. *J Digital Imaging* 35(3):623–637
32. Pan X et al (2021) AFINet: attentive feature integration networks for image classification. <http://arxiv.org/abs/2105.04354>
33. Wang S et al (2022) Improved single shot detection using DenseNet for tiny target detection. *Concurr Comput Pract Exp* 35(2):e7491
34. Kumar A, Sangwan KS, Dhiraj (2021) A computer vision-based approach for driver distraction recognition using deep learning and genetic algorithm based ensemble. https://doi.org/10.1007/978-3-030-87897-9_5
35. Jia J, Sun M, Wu G, Qiu W (2023) DeepDN_iGlu: prediction of lysine glutarylation sites based on attention residual learning method and DenseNet. *Math Biosci Eng* 20(2):2815–2830
36. Hendrawan Y et al (2022) Deep Learning to detect and classify the purity level of Luwak Coffee green beans. *Pertanika J Sci Technol* 30(1):1–18
37. Barantsov IA, Pnev AB, Koshelev KI, Tynchenko VS, Nelyub VA, Borodulin AS (2023) Classification of acoustic influences registered with phase-sensitive OTDR using pattern recognition methods. *Sensors* 23(2):582. <https://doi.org/10.3390/s23020582>
38. Yang C-HH, Tsai Y-Y, Chen P-Y (2021) Voice2series: Reprogramming acoustic models for time series classification. In: *International conference on machine learning*. PMLR
39. Arunkumar JR, berihun Mengist T (2020) Developing Ethiopian Yirgacheffe coffee grading model using a deep learning classifier. *Int J Innov Technol Exploring Eng* 9(4):3303–3309
40. Ong P, Teo KS, Sia CK (2023) UAV-based weed detection in Chinese cabbage using deep learning. *Smart Agricult Technol* 4:100181. <https://doi.org/10.1016/j.atech.2023.100181>
41. Kashiparekh K et al (2019) ConvtimeNet: A pre-trained deep convolutional neural network for time series classification. In: 2019 international joint conference on neural networks (IJCNN). IEEE
42. Mu T et al (2023) TSC-AutoML: meta-learning for automatic time series classification algorithm selection. In: 2023 IEEE 39th international conference on data engineering (ICDE). IEEE
43. Novtahaning D, Shah HA, Kang J-M (2022) Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease. *Agriculture* 12(11): 1909
44. Mridha K et al (2023) Explainable deep learning for coffee leaf disease classification in smart agriculture: a visual approach. In: 2023 international conference on distributed computing and electrical circuits and electronics (ICDCECE). IEEE
45. Hassan E, El-Rashidy N, Talaat FM (2022) Review: mask R-CNN models. <https://njccs.journals.ekb.eg>
46. Annrose J et al (2022) A cloud-based platform for soybean plant disease classification using archimedes optimization-based hybrid deep learning model. *Wirel Pers Commun* 122(4):2995–3017

47. Karthik R, Joshua Alfred J, Joel Kennedy J (2023) Inception-based global context attention network for the classification of coffee leaf diseases. *Ecol Inform* 77:102213
48. Todeschini G, Kheta K, Giannetti C (2022) An image-based deep transfer learning approach to classify power quality disturbances. *Electric Power Syst Res* 213:108795
49. Syamsuri B, Putra Kusuma G (2019) Plant disease classification using Lite pretrained deep convolutional neural network on Android mobile device. *Int J Innov Technol Explor Eng* 9(2):2796–2804
50. Ding B (2023) LENet: Lightweight and efficient LiDAR semantic segmentation using multi-scale convolution attention. <http://arxiv.org/abs/2301.04275>
51. Montalbo FJP, Hernandez AA (2020) Classifying Barako coffee leaf diseases using deep convolutional models. *Int J Adv Intell Inform* 6(2):197–209
52. Lu X et al (2022) A hybrid model of ghost-convolution enlightened transformer for effective diagnosis of grape leaf disease and pest. *J King Saud Univ Comput Inform Sci* 34(5):1755–1767
53. Waldamichael FG, Debelee TG, Ayano YM (2022) Coffee disease detection using a robust HSV color-based segmentation and transfer learning for use on smartphones. *Int J Intell Syst* 37(8):4967–4993
54. Wang N et al. COFFEE: counterfactual fairness for personalized text generation in explainable recommendation. arXiv preprint [arXiv:2210.15500](https://arxiv.org/abs/2210.15500) (2022)
55. Rani AA et al (2023) Classification for crop pest on U-SegNet. In: 2023 7th international conference on computing methodologies and communication (ICCMC). IEEE
56. Leonard F, Akbar H (2022) Coffee grind size detection by using convolutional neural network (CNN) architecture. *J Appl Sci Eng Technol Educ* 4(1):133–145
57. Suryana DH, Raharja WK (2023) Applying artificial intelligence to classify the maturity level of coffee beans during roasting. *Int J Eng Sci Inf Technol* 3(2):97–105
58. Wallelign S et al (2019) Coffee grading with convolutional neural networks using small datasets with high variance
59. Li L, Zhang S, Wang B (2021) Plant disease detection and classification by deep learning—a review. *IEEE Access* 9:56683–56698
60. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. *Adv Neural Info Process Syst* 25
61. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. arXiv preprint [arXiv:2004.10934](https://arxiv.org/abs/2004.10934)
62. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)
63. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
64. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR, pp 6105–6114
65. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
66. Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X, Liu W, Xiao B (2020) Deep high-resolution representation learning for visual recognition. *IEEE Trans Pattern Anal Mach Intell* 43(10):3349–3364
67. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
68. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.