AFSSAAE

Advances in Food Science, Sustainable Agriculture and Agroindustrial Engineering 2023, 6(4), 382-398
ISSN 2622-5921

ORIGINAL RESEARCH Open Access

Enzyme dosage detection to degrade feathers in edible bird's nests: A comparative convolutional neural networks study

Verianti Liana[1*], Rizal Arifiandika[2], Bagas Rohmatulloh[2], Riris Waladatun Nafi'ah[1], Arif Hidayat[1], Yusuf Hendrawan[2], Dimas Firmanda Al-Riza[2], Tunjung Mahatmanto[3], Hermawan Nugroho[4]

[1.] Department of Agro-Industrial Technology, Faculty of Agricultural Technology, Universitas Brawijaya, Malang, East Java, Indonesia
[2.] Department of Biosystem Engineering, Faculty of Agricultural Technology, Universitas Brawijaya, Malang, East Java, Indonesia
[3.] Department of Food Science and Biotechnology, Faculty of Agricultural Technology, Universitas Brawijaya, Malang, East Java, Indonesia
[4.] Department of Electrical and Electronic Engineering, University of Nottingham in Malaysia, Selangor, Semenyih, Malaysia

KEYWORDS

Convolutional neural networks
Edible bird's nest
Serine protease enzyme
Swiftlet feathers
Transfer learning

ABSTRACT

Edible Bird's Nest (EBN), a costly food product made from swiftlet's saliva, has encountered a longstanding problem of plucking the swiftlet's feather from the nests. The destructive and inefficient manual process of plucking the feathers can be substituted with a serine protease enzyme alternative. Accurate detection of enzyme dosage is crucial for ensuring efficient feather degradation with cost-effective enzyme usage. This research employed the transfer learning method using pretrained Convolutional Neural Networks (Pt-CNN) to detect enzyme dosage based on EBN's images. This study aimed to compare the image classification mechanisms, architectures, and performance of three Pt-CNN: Resnet50, InceptionResnetV2, and EfficientNetV2S. InceptionResnetV2, using parallel convolutions and residual networks, significantly contributes to learning rich informative features. Consequently, the InceptionResnetV2 model achieved the highest accuracy of 96.18%, while Resnet50 and EfficientNetV2S attained only 30.44% and 17.82%, respectively. The differences in architecture complexity, parameter count, dataset characteristics, and image resolution also play a role in the performance disparities among the models. The study's findings aid future researchers in streamlining model selection when facing limited datasets by understanding the reasons for the model's performance and contributing to a non-destructive and quick solution for EBN's cleaning process.

Introduction

Edible Bird's Nest (EBN), a valuable product made from the solidified saliva of swiftlet birds (*Aerodramus fuciphagus*), is known as the "Caviar of the East" (Looi and Omar, 2016; Daud et al., 2019). Indonesia is the world's largest exporter of EBN, with an annual export volume of approximately 2000 tons, contributing 0.5% to Indonesia's Gross Domestic Product (Ito et al., 2021). The price of EBN ranges from $1,000-$15,000/kg, depending on its grade, shape, origin, and type (Looi and Omar, 2016; Chan, 2018). The quality of EBN is primarily determined by its color and cleanliness, as whiter and cleaner nests fetch higher prices. To meet these standards, swiftlet breeders use meticulous EBN cleaning to remove attached feathers. However, the current manual feather-plucking process is time-consuming and costly regarding labor, equipment, and water usage (Jong et al., 2013). The common working time for the trained workers need 8 hours to clean 10 EBNs daily, an average of 48 minutes per EBN (Meng et al., 2017). Moreover, feather plucking with a tweezer is challenging due to the fine feathers firmly attached to the nests and can be destructive,

*Corresponding author
E-mail address: veriantiliana@gmail.com

Received on 18 July 2023, Revised on 17 December 2023, Accepted on 27 December 2023

reducing the EBN's weight and lowering its market value.

The feathers found in EBN are composed of 90% keratin, a water-insoluble protein with a high disulfide bond content (Gopinath et al., 2015). Utomo et al. (2018) proposed using keratinase enzyme to degrade these feathers, eliminating the need for manual plucking. However, the information on the required enzyme dosage for feather degradation in EBN was not stated. This gap was addressed by the findings of Navone and Speight (2018), who used serine protease enzyme to degrade more than 90% of the bird feathers, with a dosage of 2 KU/mL for every 0.01 gram of feathers. Considering that manual feather plucking takes around 30 mins/nest, applying serine protease enzymes would require approximately 10 hours to degrade feathers in any number of EBNs, which is more efficient and effective (Navone and Speight, 2018).

Accurate determination of the required dosage for feather degradation in EBN relies on prior knowledge of the feather's weight, which can be predicted non-destructively based on the EBN image. On the other hand, accurate enzyme dosage detection is crucial for effective feather degradation and cost-efficient enzyme usage, thereby optimizing EBN production. One highly successful deep learning algorithm that can classify images accurately is the Convolutional Neural Network (CNN) (Liu et al., 2022). The main advantage of CNN is its ability to automatically identify relevant features without human supervision (Alzubaidi et al., 2021), eliminating the need for manual texture descriptor design (Ponzio et al., 2019). Sungsiri et al. (2022) used CNN to classify different grades of EBN with an accuracy of 99.34%. Other CNN architectures like U-Net have also been proven to detect dirt in EBN with a high True Positive rate of 96.69% (Yeo and Yen, 2021). However, no existing CNN research is used for detecting enzyme dosage based on EBN images, which is the novelty of this study. Most enzyme studies using CNNs focused on the classification of enzyme commission numbers with 3D spatial structure input (Amidi et al., 2018) or protein sequence input (Ryu et al., 2019) and the classification of enzymes or non-enzymes using DDE matrices (Dipeptide Deviation from Expected Mean) (Sikander et al., 2021). Therefore, the findings of Pt-CNN's ability to classify an exact number of enzyme dosages based on raw EBN images become the pioneer in exploring the potential of more complex CNNs applications in predicting enzyme dosage for preserving EBN product quality (agro-industries) or other fields like pharmaceuticals and healthcare.

This research utilized transfer learning to address the challenge of large data requirements in CNN training (Alzubaidi et al., 2021; Ponzio et al., 2019). Transfer learning leverages knowledge from large-scale tasks and datasets like ImageNet, adapting model parameters to smaller or task-specific datasets (Alzubaidi et al., 2021; Iman et al., 2023). The comparative analysis in this study focuses on three Pt-CNN: Resnet50, InceptionResnetV2, and EfficientNetV2S. While no studies have directly compared their performance for the same classification task, individual studies have demonstrated their superiority over other models in various domains. For instance, Resnet50 achieved 94.15% accuracy for moss water stress classification (Hendrawan et al., 2021), InceptionResnetV2 achieved 92.68% accuracy for rice leaf disease detection (Islam et al., 2021), and EfficientNetV2S achieved 95.59% and 96.44% accuracy for disease detection in cardamom and grape plants, respectively (Sunil et al., 2022).

However, most studies utilizing Pt-CNN do not explain comprehensively the reasons for the accuracy differences obtained from different model implementations. Understanding the factors behind accuracy disparities among Pt-CNN models is crucial for optimizing computation time and selecting the best model. Therefore, this study provides insights into the classification mechanisms, architectures, and accuracy-influencing factors of Pt-CNN, shedding light on the relationship between input data and output accuracy and going beyond the black-box nature of the models.

Research and Methods

The research was conducted from March to June 2023. The EBN images dataset was collected from Prima Walet Indonesia (Malang City, East Java Province, Indonesia) and UD Anugerah Walet Sejahtera (Lamongan Regency, East Java Province, Indonesia). Subsequently, the assembly of the photobox was performed at the Laboratory of Mechatronics Equipment and Agroindustrial Machinery, Faculty of Agricultural Technology, Universitas Brawijaya. The model training and testing process of the Pt-CNN models was conducted at the Laboratory of Computational and Systems Analysis, Faculty of Agricultural Technology, Universitas Brawijaya.

Tools and materials

The tools and materials in this research were divided into two categories, namely, for constructing a photobox used for capturing images of EBN and developing and testing Pt-CNN models. The photobox was made of 5 mm thick white acrylic, constructed with 40 × 25 × 25 cm dimensions. Inside the photobox, the LED strip was used as the light source, a Logitech C270 HD webcam for capturing images of EBNs, a bowl-shaped EBN as the object of the image capture, a USB cable as the connection between the LED and power supply, and the connection between the webcam and laptop. Photobox can be seen in Figure 1. Next, the image preprocessing process, training, and testing of Pt-CNN models were performed on a Dell Inspiron 3505 laptop (AMD Ryzen 5 3500U processor with Radeon Vega Mobile Gfx, 8.00 GB RAM, Windows 11 Home Single Language). The image preprocessing process, training, and testing of the models were conducted using Jupyter Notebook. Three Pt-CNN models, Resnet50, InceptionResnetV2, and EfficientNetV2S, were transferred from the Tensorflow module.

EBN dataset acquisition

Uncleaned EBN (with feathers intact) was placed inside the photobox with green paper as the background. The Logitech C270 webcam, connected to a Dell Inspiron 3505 laptop, captured images in JPEG format with a resolution of 2500 × 700 pixels in RGB color mode. A total of 72 EBNs were captured from the front and back, clearly showing the location of the swiftlet feathers. Each front and back side of the EBN were combined into a single image, resulting in 72 pairs of images, as depicted in Figure 2.

Dataset labeling

Each EBN image was labelled with the dosage of serine protease enzyme required to degrade all the feathers in that particular nest. Labelling enzyme dosage was performed by plucking the feathers from the photographed EBN and weighing the feathers. Subsequently, the weight of the feathers was substituted into the following formula:

$$\text{Serine protease enzyme dosage (KU/mL)} = \frac{\text{swiftlets feathers weight}}{0.01\ \text{gram}} \times 2\ \text{KU/mL} \quad \text{.....(1)}$$

The formula was derived from the research conducted by Navone and Speight (2018), which utilized 2 KU/mL of a serine protease enzyme that successfully degraded more than 90% of feathers. One KU (Keratin Unit) is defined as the increment of 0.1 absorbance at 595 nm after 30 minutes of incubation of 0.01 gram of keratin azure in 100 μ L enzyme dilution with 2.4 mL of 100 mM Tris-HCl buffer pH 8 or 10 (Navone and Speight, 2018). Several EBNs exhibited marginally differing or equally weighted feathers, resulting in the assignment of the same enzyme dosage class after undergoing calculation using Equation 1. Consequently, from the 72 uncleaned EBNs, 30 classes of enzyme dosage were identified within the range of 3-59 KU/mL. These enzyme dosage numbers were then chosen as the folder labels to represent distinct classes of images within the observed range, specifically 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 16, 17, 18, 19, 20, 21, 22, 23, 24, 27, 29, 30, 31, 32, 40, 41, 42, 47, and 59.

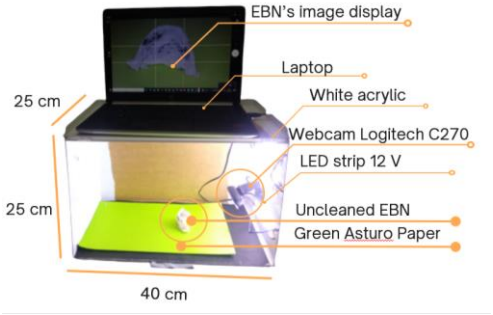


Figure 1. Photobox prototype

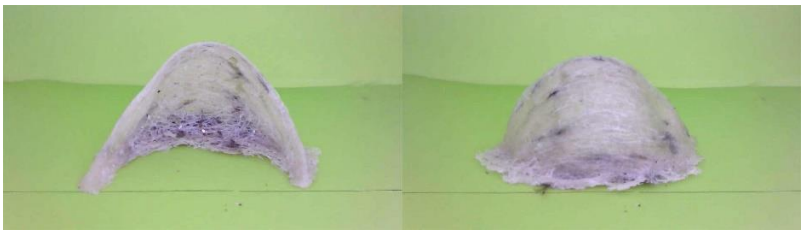


Figure 2. EBN's dataset sample

Dataset augmentation

In this study, augmentation was performed on 72 EBN images using rotation (90°, 180°, and 270°), flipping, contrast adjustment, and brightness adjustment, resulting in 2880 images. The image data generator tool facilitated the image augmentation process, which enables real-time image augmentation through the tensorflow.keras.preprocessing.image module.

Separation of training and validation datasets

From the 2,880 images, the data were randomly divided into training and validation sets in a proportion of 70:30, resulting in 2016 training data and 864 validation data. The 70:30 dataset split ratio was chosen as it has been proven to demonstrate better performance of CNN compared to the ratios of 80:20 and 60:40 in Distributed Denial of Service classification (Gadze et al., 2021). The training data was used to train the model to classify images into 30 enzyme classes, while the validation data was used to evaluate the accuracy of the trained model in classifying unseen images during training (Ho et al., 2020; BaniMustafa et al., 2023).

Dataset preprocessing

The image dataset preprocessing was performed by dividing each pixel value of the image by 255 using the image data generator, as the image data is in RGB format with pixel values ranging from 0 to 255. After rescaling, the images were resized directly to the target input size of each Pt-CNN model, where Resnet50 and EfficientNetV2S require an input size of 224 × 224 pixels, while InceptionResnetV2 requires an input size of 299 × 299 pixels.

Training pretrained convolutional neural networks (Pt-CNN) models

The Pt-CNN models were trained using transfer learning, utilizing pre-trained models available in the tensorflow.keras.applications module. All model programming was performed in Python using Jupyter Notebook and the Tensorflow modules. A total of 2016 training data samples were used to build the Pt-CNN models for detecting serine protease enzyme dosages. After model construction, the performance was evaluated using 864 validation data samples, categorized into 30 enzyme dosage classes. Each Pt-CNN model (Resnet50, InceptionResnetV2, and EfficientNetV2S) utilized the following hyperparameters: batch size = 32, epochs = 20, patience = 5, optimizer = Adam, learning rate = 0.001, and loss function = sparse categorical cross-entropy. The programming implemented an early stopping technique, which terminated the model training early based on predefined criteria. In this case, it monitored the validation loss metric. The training process would stop if the validation loss did not improve for five consecutive epochs (patience = 5).

Assessing the performance of models

After completing the training and validation process of the Pt-CNN models, four performance metrics would be presented as graphs: train loss, train accuracy, validation loss, and validation accuracy. Train loss is a metric that measures the error during the training phase, aiming to optimize the updated network parameters for each epoch. The loss function employed in this study is Sparse Categorical Cross Entropy (SCCE) since it utilizes integer labels for multi-class classification (Darafsh et al., 2021). The formula for the SCCE loss function is given by Equation 2.

$$\text{SCCE}(y, \hat{y}) = -\sum_{i=1}^{C} yi \log(\hat{y}i) \quad \text{.....(2)}$$

Where:
y is the target vector that contains the true labels in the one hot encoding form (binary vector where the element corresponding to the category is set to 1, while the other elements are set to 0)
ŷ is the prediction vector that contains the prediction probabilities for each class
C is the number of classification classes
i is the index of the class

Train accuracy is a metric used to measure the model's performance in correctly classifying the training data. The formula for train accuracy is shown in the Equation 3 below:

$$\text{Train accuracy} = \frac{\text{number of correct classified training data}}{\text{total training data}} \quad \text{.....(3)}$$

A high training accuracy does not guarantee that the model will perform well on unseen data (Hammad and El-Sankary, 2019); thus, validation accuracy becomes the determining metric for the model's ability to generalize patterns in the data. This is due to the role of validation accuracy, which evaluates the model's performance in classifying the unseen data during the training process, with the formula, as shown in Equation 4. Therefore, among the four-performance metrics, the

validation accuracy was prioritized for selecting the best model since the ultimate goal of machine learning is to produce a model that can perform well and accurately classify unseen data, making it more robust and relevant to real-world scenarios (Pawar et al., 2023). Validation loss is a metric used to evaluate the model's performance in minimizing errors or loss on the validation data. The formula for validation loss is the same as Equation 2, which utilizes SCCE.

Validation accuracy

$$= \frac{\text{number of correct classified validation data}}{\text{total training data}} \quad \text{.....(4)}$$

Results and Discussion

Dataset classification mechanism with resnet50

Detecting serine protease enzyme doses in EBN images using ResNet50 included a feature extraction process with convolutional layers, followed by a classification layer to determine the image class. The convolutional layers extracted features from 224 × 224 pixel images, generating a series of feature maps capturing various image aspects (Rani et al., 2023). ResNet50 utilized a residual learning framework at the high-level abstraction to address the vanishing gradient problem in deep neural networks (He et al., 2015). This framework introduced shortcut connections between network layers, allowing the network to learn residual features that captured the difference between the input and output of a specific layer. The output of each residual block was then passed through pooling layers to reduce the spatial dimension of the feature map. Subsequently, the output of the pooling layers was flattened and passed through several fully connected layers that performed the final classification task (Tsalera et al., 2022). The output of the fully connected layers was passed through a softmax layer, which transformed the output into a probability distribution over the predetermined classes (El-Magd et al., 2022). Finally, the ResNet50 classifier predicted the class label of the input image by selecting the class with the highest probability from the softmax output.

Resnet50 architecture

ResNet50 has 50 layers of a residual network consisting of 49 convolutional layers and one fully connected layer (He et al., 2015). These 49 layers are divided into five sections, with the first convolution layer serving as input preprocessing and the remaining four sections as bottleneck building blocks. Each bottleneck block comprises several convolutional layers, batch normalizations, rectified linear unit activation functions, and shortcuts (Zhang et al., 2022). Detailed information about the layers in the ResNet50 architecture can be found in Table 1 (He et al., 2015).

The first layer of this network is a convolutional layer with 64 filters of size 7×7 and a stride of 2, which aims to reduce the spatial dimensions of the input image (He et al., 2015). The subsequent layers consist of a series of bottleneck residual blocks, a variation of the residual block that uses 1×1 convolution to create a different bottleneck compared to the previous ResNet34 network. The bottleneck is intended to reduce the number of parameters and matrix multiplications, thereby decreasing the input dimension to the subsequent 3×3 convolutional layer (He et al., 2015).

Table 1. ResNet50 architecture

Layers	Output Size	50-Layer
conv1	112×112	7×7, 64, *stride* 2
		3×3 *max pool*, *stride* 2
conv2_x	56×56	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
conv4_x	14×14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
conv5_x	7×7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	Average pool, 30-d fully connected, softmax

Each block consists of multiple convolutional layers followed by shortcut connections, allowing the network to learn residual features. Residual feature (residual function = F(x)) refers to the difference between the input (x) and the output of the residual block (H(x)), formulated as F(x) = H(x) − x (He et al., 2015). Therefore, the expected output (H(x)) can be represented as F(x) + x, which can be achieved through a feedforward neural network with shortcut connections (He et al., 2015). The purpose of the operation H(x) = F(x) + x is to enable the network to identify areas for parameter updates. If F(x) effectively maps the input (x) to the desired output (H(x)), the value of F(x) will approach 0, causing H(x) to approximate x. This means that when the H(x) value deviates significantly from x, the network needs to update the parameters to make the output approach the input. Shortcut or residual connections, acting as identity mappings (x), pass through several convolutional layers and add the previous layer's output to the stacked output (He et al., 2015). Therefore, if a layer reduces accuracy, it will be eliminated (Tsalera et al., 2022). This prevents the gradient vanishing problem and improves the efficiency of network training (Zhao et al., 2022).

After the last residual block, the feature representation's spatial dimension is reduced using average pooling (Sabri et al., 2020). The resulting feature map with lower dimensions is then connected to a fully connected layer with 30 neurons, serving as the final classification layer (Galanis et al., 2022). In this study, the original ResNet50 architecture's 1000 neurons were converted to 30 neurons to match the 30 enzyme dosage classes. The output of the fully connected layer is passed through a softmax activation function to produce probabilities for each class (Maeda-Gutiérrez et al., 2020). The class with the highest probability is considered the network's final prediction.

Dataset classification mechanism with inceptionResnetV2

In InceptionResnetV2, input images of size 299 × 299 pixels (Ryu, 2023) undergo convolutional layers to extract features (Wan et al., 2019). Inception modules capture multi-scale features with different filter sizes (Nazir et al., 2019), followed by a filter expansion layer (1 × 1 convolution without activation) to match input depth (Ryu, 2022). Output feature maps are merged through filter concatenation. InceptionResNetV2 also includes residual connections, aid in propagating residual information efficiently, mitigating the gradient vanishing problem (Merino et al., 2021; He et al., 2015). Afterwards, the generated feature maps are pooled via average pooling, resulting in small-sized maps with smoother values (Galanis et al., 2022). Finally, a fully connected layer with softmax function predicts input image probabilities for predefined categories (Liu et al., 2022; Maeda-Gutiérrez et al., 2020).

InceptionResnetV2 architecture

InceptionResNetV2 has 164 layers of 4 max-pooling layers and 160 convolutional layers, with approximately 55 million parameters (Shazia et al., 2021; Elharrouss et al., 2022). The architecture of InceptionResnetV2 consists of blocks including input block (229 × 229 pixels), stem, 5 × Inception-Resnet-A, Reduction-A, 10 × Inception-Resnet-B, Reduction-B, 5 × Inception-Resnet-C, average pooling, dropout, and softmax (Demir and Yilmaz, 2020). The details of the architecture can be found in Table 2. The stem scheme in InceptionResNetV2 includes convolutional layers, filter concatenation, and max pooling. Convolutional layers consist of convolutional filters or kernels to extract features (Yamashita et al., 2018), and the weights of kernels are adjusted during training to learn significant features (Alzubaidi et al., 2021). Max pooling is employed as a sub-sampling method to reduce the size of feature maps while preserving dominant feature information (Alzubaidi et al., 2021).

After passing through the stem component, the training process is continued with the Inception-Resnet-A section. The Inception-Resnet blocks are followed by Reduction blocks, except Inception-Resnet-C. The notable distinction between Inception-Resnet-A, Inception-Resnet-B, and Inception-Resnet-C lies in the number of filters and feature maps output (Szegedy et al., 2016). In Inception-ResNet-A, the final convolution process involves 1 × 1 Conv (384 linear) filters, resulting in 384 feature maps (Szegedy et al., 2016). As the blocks are traversed, the number of filters and output feature maps progressively increase, where Inception-Resnet-B and Inception-Resnet-C generate 1154 and 2048 linear filters and output (Szegedy et al., 2016). This increasing trend indicates that the model captures more complex image features and patterns as the training progresses.

Residual connections are implemented in each Inception-Resnet block to sum the input with the output of parallel convolution branches, allowing a direct flow of residual information and addressing the vanishing gradient problem (Yue et al., 2018). This resulting sum is passed through the ReLU

activation function, which activates positive values and sets negative values to 0 (Chieng et al., 2018). ReLU introduces non-linearity, enabling the network to learn complex representations (Kulathunga et al., 2021). After five iterations of the Inception-Resnet-A block, its output is connected to the Reduction-A block. Subsequently, the output of Reduction-A undergoes ten more iterations in the Inception-Resnet-B block. It is then connected to the Reduction-B block. Both Reduction-A and Reduction-B reduce the dimensionality of the feature maps while retaining important image information by applying MaxPool layers (Alzubaidi et al., 2021). This is evident from the decline of output dimension in Reduction blocks compared to the Inception-Resnet blocks (Table 2): Inception-Resnet-A goes from $35 \times 35 \times 256$ to $17 \times 17 \times 896$ in Reduction A, Inception-Resnet-B goes from $17 \times 17 \times 896$ to $8 \times 8 \times 1792$ in Reduction-B. This shows InceptionResnet's effectiveness in extracting informative features with reduced complexity (Szegedy et al., 2016). After passing through the Inception-Resnet-C block, the feature maps undergo average pooling. Dropout is then applied as a regularization technique during training to randomly set hidden unit activations to zero, preventing overfitting (Liu et al., 2022). In this architecture, the dropout rate is specified as "keep 0.8", indicating an 80% retention probability for each unit during training (Srivastava et al., 2014). Finally, the softmax function is utilized in the last layer to produce probability values for each class between 0 and 1 (Alzubaidi et al., 2021).

Dataset classification mechanism with efficientNetV2S

EfficientNetV2S resizes the input image to 224×224 pixels and applies a 3×3 convolutional layer with a stride of 2. MBConv (Mobile Inverted Residual Bottleneck Convolution) and Fused-MBConv blocks extract features at different scales with fewer parameters than traditional convolution layers (Chollet, 2017; Tan and Le, 2021). Depthwise convolutions in MBConv reduce parameters by using separate filters for each input channel (Chollet, 2017). While Fused-MBConv replaces the 3×3 depthwise and 1×1 expansion convolutions in MBConv with regular 3×3 convolutions to further reduce parameters (Tan and Le, 2021). Both blocks incorporate the Squeeze and Excitation (SE) block to enhance feature learning and the block's output is passed through the Swish activation function (Tan and Le, 2021), a non-linear activation function that produces smoother gradients compared to the ReLU function (Ramachandran et al., 2017). EfficientNetV2S utilizes progressive learning that applies weak regularizations to smaller images and progressively transitions to stronger regularizations for larger images to address overfitting and enhance accuracy and training speed (Tan and Le, 2021). Regularization techniques include dropout, RandAugment, and Mixup (Srivastava et al., 2014; Cubuk et al., 2020; Tan and Le, 2021). The final MBConv layer produces a feature map subjected to Global Average Pooling (GAP), maintaining a constant output size by averaging value per channel (Patel and Wang, 2022). The resulting single value from GAP is concatenated into a vector and processed through fully connected layers (Alzubaidi et al., 2021), contributing to the generation of a probability distribution for image classes.

The architecture of efficientNetV2S

The EfficientNetV2S architecture begins with a stem layer, followed by seven stages, each containing multiple MBConv and Fused-MBConv blocks. The stem layer initiates image processing with a 3×3 convolutional layer generating 24 channels and using a stride of 2 (Tan and Le, 2021), effectively reducing the output's spatial dimensions by half (Goodfellow et al., 2016). The stem output is fed into the Fused-MBConv and MBConv layers (Table 3). Fused-MBConv is applied in stages 1-3 for faster training with more efficient FLOPs and parameters, which are determined through an automated architecture search (Tan and Le, 2021).

Table 2. The architecture of inceptionResnetV2

Layer	Output Size	Channels
Input	299×299	3
Stem	35×35	256
5×Inception-Resnet-A	35×35	256
Reduction-A	17×17	896
10×Inception-Resnet-B	17×17	896
Reduction-B	8×8	1792
5×Inception-Resnet-C	8×8	1792
Average Pooling	1×1	1792
Dropout (keep 0.8)	1×1	1792
Softmax	1×1	30

Table 3. The architecture of efficientNetV2S

Stage	Operator	Stride	Total Channels	Total Layers
0	Conv3×3	2	24	1
1	Fused-MBConv1, k3×3	1	24	2
2	Fused-MBConv4, k3×3	2	48	4
3	Fused-MBConv4, k3×3	2	64	4
4	MBConv4, k3×3, SE0.25	2	128	6
5	MBConv6, k3×3, SE0.25	1	160	9
6	MBConv6, k3×3, SE0.25	2	256	15
7	Conv1×1 & Pooling & Fully connected	-	1280	1

The differences between the Fused-MBConv layers and the MBConv layers are in terms of stride, channels, and layers, as shown in Table 3. The term $k3\times3$ notation denotes 3×3 filter size, updated during training through backpropagation. In the MBConv layers, the term SE0.25 represents Squeeze and Excitation with a 25% reduction ratio of parameters (Tan and Le, 2021). The SE block recalibrates feature weights to enhance representation by assigning larger weights to essential features and smaller weights to less critical features, thereby reducing parameters and emphasizing informative features (Hu et al., 2020). In the final stage, a Conv1×1, Pooling, and Fully connected layer employs a 1×1 convolution to reduce spatial dimensions and enhance complex feature learning by increasing channels to 1280 (Sharma and Foroosh, 2020; Tan and Le, 2021). This is followed by GAP for one-dimensional feature representation, connected to a fully connected layer for classification (Zhu et al., 2022; Lin et al., 2014; Alzubaidi et al., 2021).

Performance result of resnet50 model

The process of ResNet50 training to classify 30 classes of serine protease enzymes based on EBN data stopped at epoch 19 due to the application of early stopping. The detailed accuracy and loss results during the training and validation process are shown in Figure 3a. To determine the trend of improvement or degradation for each metric, each metric was calculated with:

$$\text{Percentage Change in Metric per Epoch} = \frac{\text{metric value of epoch}-(n) - \text{metric value of epoch}-(n-1)}{\text{metric value of epoch } (n-1)} \times 100\% \quad \text{.....(5)}$$

If the result of Equation 5 was negative, the color appeared orange in Difference (%) column in Figure 3a, meaning that the corresponding metric value in that epoch has decreased compared to the previous epoch. Conversely, if the result was positive, meaning that the metric value has improved compared to the previous epoch. Negative values or decreases in value are expected in the train and validation loss metrics, while positive values or improvements in value are expected in the train and validation accuracy metrics. This analysis of the trend formula was applied to all Pt-CNN models in this study.

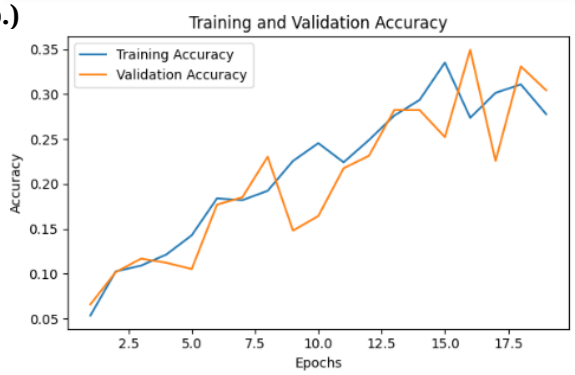
The distribution of columns with orange color in Figure 3a, indicating a decrease in the metric value, shows varied patterns without significant trends. In the train loss metric, a significant decrease of 61.14% occurred from epoch 1 to epoch 2, followed by a slight 2.28% decrease from epoch 2 to epoch 3. Train loss increased in the last four epochs, with the most significant increase of 12.83% in epoch 19. Train accuracy showed a trend of increasing values up to epoch 15 and decreased in epochs 16 and 19, as shown in Figure 3b. Train accuracy decreased, and train loss increased from epoch 16 to 19, indicating the model failed to achieve stable or optimal performance during training. Validation metrics exhibited instability, with no clear decreasing trend in validation loss and fluctuating validation accuracy. The training stopped at epoch 19 due to a lack of a significant decrease in validation loss (Barry-Straume et al., 2018). Validation loss in epoch 19 (3.6663) was not better than in epoch 15 (3.3873). Notably, a decrease in validation loss does not always correspond to an increase in validation accuracy, as seen in Figure 3a and 3c. For example, epoch 16 had a high validation accuracy of 34.95% but an increased validation loss of 3.5215 compared to the previous epoch (3.3873).

The validation accuracy values fluctuated highly, indicating instability and limited ability to classify images correctly. The validation loss also had instability, as shown in Figure 3c, where the increase in validation loss reflected high uncertainty in classifying enzyme dosages, leading to errors and potential wastage. Overall, the unstable patterns of accuracy and loss indicated that the Resnet50 has not effectively learned the EBN features, and the validation accuracy of the last epoch showed that only 30.44% of the validation data were correctly classified.

a.)

Epoch (n)	Train Loss	Difference (%)	Train Accuracy	Difference (%)	Validation Loss	Difference (%)	Validation Accuracy	Difference (%)
1	11,8269		0,0536		5,4461		0,0660	
2	4,5962	-61,14	0,1027	91,60	3,9964	-26,62	0,1019	54,39
3	4,4916	-2,28	0,1091	6,23	4,7561	19,01	0,1169	14,72
4	4,6638	3,83	0,1215	11,37	5,5298	16,27	0,1123	-3,93
5	4,6658	0,04	0,1429	17,61	5,5371	0,13	0,1053	-6,23
6	4,0733	-12,70	0,1840	28,76	3,9790	-28,14	0,1771	68,19
7	3,8435	-5,64	0,182	-1,09	5,3505	34,47	0,1852	4,57
8	3,9268	2,17	0,1925	5,77	4,1938	-21,62	0,2303	24,35
9	4,0181	2,33	0,2257	17,25	4,9616	18,31	0,1481	-35,69
10	3,8071	-5,25	0,2455	8,77	3,7187	-25,05	0,1644	11,01
11	3,8759	1,81	0,2242	-8,68	4,4217	18,90	0,2176	32,36
12	3,5091	-9,46	0,2490	11,06	3,5254	-20,27	0,2315	6,39
13	3,5445	1,01	0,2763	10,96	3,5555	0,85	0,2824	21,99
14	3,1568	-10,94	0,2937	6,30	3,0544	-14,09	0,2824	0,00
15	3,0793	-2,46	0,3353	14,16	3,3873	10,90	0,2523	-10,66
16	3,3751	9,61	0,2738	-18,34	3,5215	3,96	0,3495	38,53
17	3,4143	1,16	0,3016	10,15	4,0498	15,00	0,2257	-35,42
18	3,4955	2,38	0,3110	3,12	4,0753	0,63	0,3310	46,65
19	3,9438	12,83	0,2778	-10,68	3,6663	-10,04	0,3044	-8,04

b.)



c.)

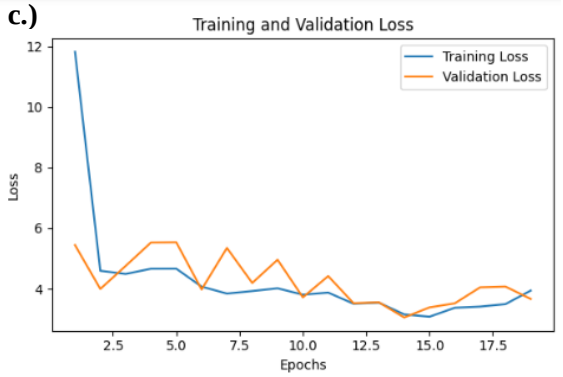


Figure 3. Visualization of resnet50 performance metrics, including: a) The training and validation history with percentage change per epoch, b) Training and validation accuracy graph, and c) Training and validation loss graph

Performance result of inceptionResnetV2 model

The InceptionResnetV2's training stopped at epoch 19 using early stopping, as there was no improvement in the validation loss over the last five epochs. Figure 4a provides the detailed performance metric values and the value differences (increment/reduction), calculated similarly to the prior explanations of Resnet50. The train loss metric was dominated by the orange color in Figure 4a, indicating the model's tendency to minimize prediction errors in the training data. The highest train loss value was found at epoch 1, and it was successfully reduced by 95.74% in the subsequent epoch. This indicated that the model has successfully learned from the training data, updated its parameters, and optimized the loss function. After epoch 4, the train loss value increased in epochs 5, 7, 10, 13, and 15. For each time the train loss increased, it subsequently decreased in the next epoch, indicating that the model quickly learned to improve its performance whenever there was an increase in prediction errors in the training data. Epochs 18 and 19 exhibited notable increases in train loss, indicating potential challenges in generalizing patterns in the data, but it required further validation metric evaluation.

Similar to the train loss metric, the train accuracy values showed significant changes in epoch 2, where the train accuracy increased by 106.47% from 45.14% to 93.20%. This means that the model successfully reduced train loss and improved the accuracy of the training data predictions. This can be clearly seen in Figure 4b and 4c, where the graph shows a sharp increase in training and validation accuracy and a sharp decrease in training and validation loss from epoch 1 to epoch 2. While there were minor fluctuations in train loss from epoch 4 to 7, train accuracy remained above 90%, overall, relatively stable. In epochs 18 and 19, despite a slight decrease in train accuracy when train loss increased (around 1.20%), the accuracy values remained above 95%.

InceptionResnetV2 consistently exhibited the lowest train loss values (after the first epoch) among the compared models, demonstrating its ability to learn patterns and make accurate predictions. The validation loss showed a downward trend until epoch 14 but had a significant increase at epoch 10 (1.1011).

However, the InceptionResnetV2 model improved its performance, decreasing validation loss to 0.6722 in epoch 11. Starting from epoch 15, the validation loss increased significantly, triggering the early stopping algorithm. There was no significant improvement in validation loss in the last five epochs, leading to training termination at epoch 19. The use of early stopping prevents overfitting and ensures model performance (Bonet et al., 2021). The InceptionResnetV2 model achieved a validation accuracy of 83.91% in the first epoch, increasing until epoch 3. Insignificant fluctuations in validation accuracy were observed in Figure 4b. From epochs 11 to 19, the validation accuracy consistently surpassed 95%, demonstrating the model's high accuracy rate. The close alignment of the lines representing train and validation loss and accuracy in Figure 4b and 4c. Further illustrated the successful application of learned features during training by the InceptionResnetV2 model to the validation data. This robust performance highlights the model's ability to accurately predict unseen data, making it well-suited for real-life applications.

Performance result of efficientNetV2S

The training process of the EfficientNetV2S model did not stop early and continued until the maximum specified epoch, which was epoch 20. The detailed metrics of the performance results of the EfficientNetV2S model training with the same calculation approach as the previous models can be found in Figure 5a. The EfficientNetV2S model demonstrated a predominant decreasing trend in train loss, shown by the dominant, orange-colored columns. However, there were occasional insignificant fluctuations with slight increases. Regarding train accuracy, there were fluctuating but overall increasing values, as seen in Figure 5b.

a.)

Epoch (n)	Train Loss	Difference (%)	Train Accuracy	Difference (%)	Validation Loss	Difference (%)	Validation Accuracy	Difference (%)
1	18,1583		0,4514		2,2933		0,8391	
2	0,7728	-95,74	0,9320	106,47	0,5615	-75,52	0,9329	11,18
3	0,2333	-69,81	0,9737	4,47	0,5969	6,30	0,9583	2,72
4	0,1935	-17,06	0,9722	-0,15	0,9395	57,40	0,9479	-1,09
5	0,2550	31,78	0,9692	-0,31	0,4961	-47,20	0,9572	0,98
6	0,2096	-17,80	0,9688	-0,04	0,4230	-14,73	0,9479	-0,97
7	0,7825	273,33	0,9315	-3,85	1,1159	163,81	0,9190	-3,05
8	0,4227	-45,98	0,9668	3,79	0,5429	-51,35	0,9630	4,79
9	0,2346	-44,50	0,9797	1,33	0,3831	-29,43	0,9664	0,35
10	0,5354	128,22	0,9697	-1,02	1,1011	187,42	0,9410	-2,63
11	0,2919	-45,48	0,9712	0,15	0,6722	-38,95	0,9583	1,84
12	0,2125	-27,20	0,9846	1,38	0,7274	8,21	0,9688	1,10
13	0,4089	92,42	0,9762	-0,85	0,6001	-17,50	0,9734	0,47
14	0,1083	-73,51	0,9881	1,22	0,2255	-62,42	0,9826	0,95
15	0,2864	164,45	0,9856	-0,25	0,6045	168,07	0,9560	-2,71
16	0,0998	-65,15	0,9911	0,56	0,6188	2,37	0,9525	-0,37
17	0,0920	-7,82	0,9936	0,25	0,2457	-60,29	0,9838	3,29
18	0,1214	31,96	0,9911	-0,25	0,3839	56,25	0,9815	-0,23
19	0,4153	242,09	0,9792	-1,20	0,5960	55,25	0,9618	-2,01

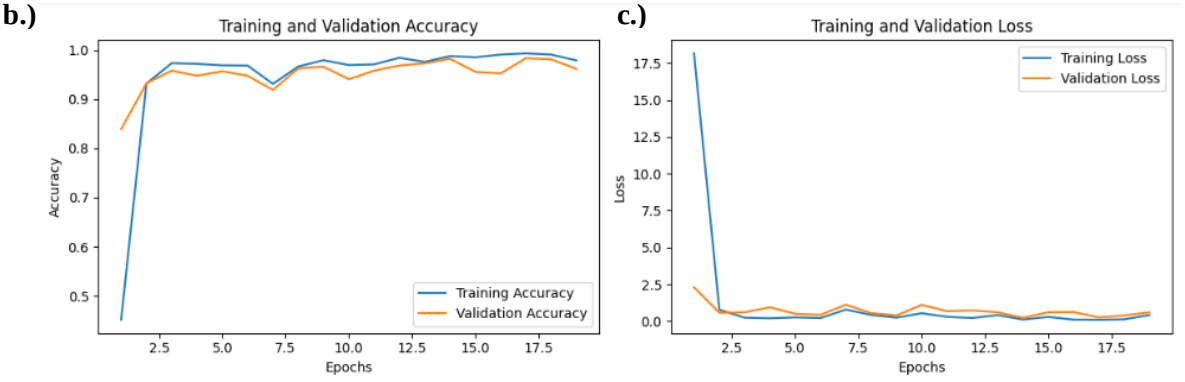


Figure 4. Visualization of inceptionResnetV2 performance metrics, including: a) The training and validation history with percentage change per epoch, b) Training and validation accuracy graph, and c) Training and validation loss graph

a.)

Epoch (n)	Train Loss	Difference (%)	Train Accuracy	Difference (%)	Validation Loss	Difference (%)	Validation Accuracy	Difference (%)
1	3,2251		0,1131		3,1280		0,1447	
2	3,1316	-2,90	0,1255	10,96	3,0677	-1,93	0,1586	9,61
3	3,1160	-0,50	0,1305	3,98	3,0382	-0,96	0,1285	-18,98
4	3,0561	-1,92	0,1349	3,37	3,0114	-0,88	0,1076	-16,26
5	3,0115	-1,46	0,1260	-6,60	2,9815	-0,99	0,1481	37,64
6	3,0208	0,31	0,1458	15,71	2,9402	-1,39	0,1551	4,73
7	2,9713	-1,64	0,1409	-3,36	2,9206	-0,67	0,1736	11,93
8	2,9335	-1,27	0,1528	8,45	2,9437	0,79	0,1331	-23,33
9	2,9526	0,65	0,1538	0,65	2,8881	-1,89	0,1586	19,16
10	2,9487	-0,13	0,1493	-2,93	2,8892	0,04	0,1644	3,66
11	2,9266	-0,75	0,1528	2,34	2,8694	-0,69	0,1736	5,60
12	2,9198	-0,23	0,1478	-3,27	2,8504	-0,66	0,1447	-16,65
13	2,9132	-0,23	0,1523	3,04	2,8352	-0,53	0,1817	25,57
14	2,8807	-1,12	0,1627	6,83	2,8308	-0,16	0,1771	-2,53
15	2,9146	1,18	0,1597	-1,84	2,8188	-0,42	0,1609	-9,15
16	2,8903	-0,83	0,1711	7,14	2,8083	-0,37	0,1644	2,18
17	2,8763	-0,48	0,1741	1,75	2,8029	-0,19	0,1678	2,07
18	2,8559	-0,71	0,1582	-9,13	2,8083	0,19	0,1725	2,80
19	2,8534	-0,09	0,1766	11,63	2,7776	-1,09	0,1910	10,72
20	2,8551	0,06	0,1667	-5,61	2,7805	0,10	0,1782	-6,70

b.)

c.)

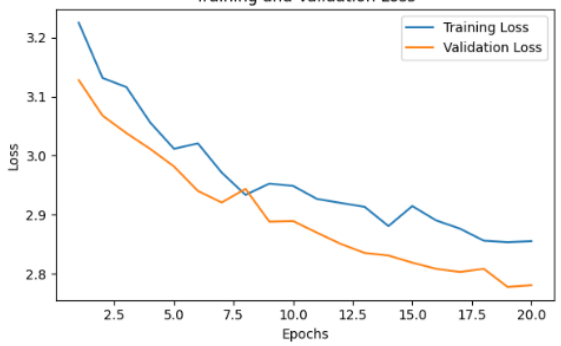


Figure 5. Visualization of efficientNetV2S performance metrics, including: a) The training and validation history with percentage change per epoch, b) Training and validation accuracy graph, and c) Training and validation loss graph

The validation loss metric predominantly decreased, with only 4 out of 20 values showing insignificant increases. The graph in Figure 5c clearly illustrates the decreasing training and validation loss trend. In contrast, the validation accuracy metric exhibited variable increases and was not consistently predictable, as there were sharp drops after some instances of improvement. This indicated that the trained model was unstable in generalizing to unseen validation data in particular epochs. Since the model training continued until the maximum specified epoch (epoch 20), the early stopping algorithm did not detect a significant downward trend in the validation loss metric. Although there was a minor increase of 0.10% in the validation loss during the last epoch, it was not substantial, and the decreasing trend of the validation loss suggests the potential for further decline if training were to continue. Similarly, the validation accuracy metric has the potential to improve with extended training, but the lack of significant improvement in its trend suggests limited enhancements with more epochs. Moreover, prolonged training increases the risk of overfitting, as the model may memorize non-significant features that do not contribute significantly to validation accuracy performance.

The top performing model selection

Based on the analysis of metrics, particularly the validation accuracy, among these three Pt-CNN models, the InceptionResnetV2 model was the most robust and accurate in detecting the serine protease enzyme dosage in the EBN images. A summary of the values for the validation accuracy and other metrics at the last epoch for the three Pt-CNN models is presented in Table 4.

InceptionResnetV2 achieved the highest validation accuracy of 96.18%, outperforming Resnet50 (30.44%) and EfficientNetV2S (17.82%) (Table 4). Its lower train loss and validation loss values indicated effective learning and reduced prediction errors. The train accuracy demonstrated that 97.92% of the 2016 training samples were accurately classified. This high train accuracy, which only differed by 1.77% from the validation accuracy, indicated that the InceptionResnetV2 model had an effective training outcome and was

not overfitting, making it suitable for new images. Furthermore, InceptionResnetV2 exhibited a high validation accuracy of 83.91% in the first epoch, surpassing Resnet50 (6.60%) and EfficientNetV2S (14.47%). The findings of this study differed from the hypothesis, which predicted that the EfficientNetV2S would outperform the other two Pt-CNN due to its higher Top1-accuracy when trained on the ImageNet dataset. Several scientific reasons for the highest validation accuracy of InceptionResnetV2 and being the best model for detecting enzyme dosage for EBN images are as follows:

a. InceptionResnetV2 has a more complex and deep architecture than Resnet50 and EfficientNetV2S, combining the strengths of Inception and Residual Network models. This architecture allows InceptionResnetV2 to effectively handle scale and pattern complexities in the data (Nazir et al., 2019; Hu et al., 2020) and address deep training information with residual blocks (Alaeddine and Jihene, 2021), resulting in high validation accuracy for classifying enzyme dosage from EBN images. The Pt-CNN models used in the study were pretrained on ImageNet and accessed through the Tensorflow module. InceptionResnetV2 has a higher number of both trainable and non-trainable layers amount (782) compared to Resnet50 (177) and EfficientNetV2S (516), indicating its greater capacity to learn complex data patterns and extract high-level features from images.
b. The implementation of innovations, specifically different backbones in each model, had a significant impact on their performance in image classification. Both InceptionResnetV2 and Resnet50, which had the first and second highest accuracy, utilized residual connections to prevent the gradient vanishing problem (He et al., 2015; Yue et al., 2018). The Inception backbone of the InceptionResnetV2 was assumed to contribute to its high accuracy since it extracted the complex and informative image features (Wang et al., 2023). The excellent performance of InceptionResnetV2 result aligned with the findings of Chun et al. (2022) which stated that InceptionResnetV2 outperformed Resnet50 in extracting food image features. The EfficientNetV2S backbone consisted of MBConv and Fused-MBConv blocks, which reduced the number of parameters while maintaining accuracy (Chollet, 2017; Tan and Le, 2021). Overall, these backbone innovations played a crucial role in improving the performance of the Pt-CNN models (Table 5).
c. The InceptionResnetV2 model has a higher number of parameters compared to the other two models. Based on the parameter data (trainable and non-trainable parameters), InceptionResnetV2 had a total of 57,285,886 parameters, making it the model with the highest number of parameters among Resnet50 and EfficientNetV2S, which had 26,598,302 and 20,369,790 parameters, respectively. Parameters in a CNN, namely weights and biases (Kumar and Garg, 2018), affected the model's ability to capture the relationship between input and output variables. The more parameters there are, the more complex and flexible the model is in learning complex data patterns and producing representations of the input-output relationship. With more relationship representations, the model has more options to select the best accuracy resulting in parameters (Lu et al., 2022) and can adapt to a broader range of variations to achieve higher accuracy values.
d. The EBN dataset's unique characteristics align well with the image features learned by the Pt-InceptionResnetV2 model during its previous training on the ImageNet dataset (Taye, 2023). Through the ImageNet training process, the model's large number of parameters (weights) enabled it to establish accurate relationships between inputs and outputs through backpropagation (Laraba et al., 2019). When processing EBN images, the Pt-InceptionResnetV2 model transfers pretrained weights with fixed values at the start of the feature extraction process. The high validation accuracy of 83.91% at the beginning of training indicates a strong match between the EBN image features and the learned parameters from ImageNet (Bemporad and Piga, 2021). These pretrained weights, tailored to the characteristics of the input image features, provide an advantageous starting point, bringing the Pt-CNN model close to the global optimum. Leveraging these appropriate pretrained weight values enables the InceptionResnetV2 model to achieve high accuracy faster during training, optimizing time and resource requirements (Bemporad and Piga, 2021; Li et al., 2022).

Table 4. Comparison of training results for the three models

Pt-CNN Model	Train loss	Train accuracy	Validation loss	Validation accuracy
Resnet50	3.9438	0.2778	3.6663	0.3044
InceptionResnetV2	0.4153	0.9792	0.5960	0.9618
EfficientNetV2S	2.8551	0.1667	2.7805	0.1782

Table 5. Summary of innovations among the three Pt-CNN

Pt-CNN	The Applied Innovations	The Benefits of Applied Innovations
Resnet50	Bottleneck building block convolution 1×1 with shortcut connections	- Reducing the number of parameters and matrix multiplications in the network, thereby reducing the spatial dimension of the output to improve computational efficiency. - Allowing the network to learn residual features through shortcut connections to prevent the problem of gradient vanishing in deep neural networks.
Inception-ResNetV2	The combination of *Inception* module (convolutional layer with different filter size) and residual connection	- Able to extract features at different spatial scales, resulting in richer and more complex image information. - Allowing the network to propagate residual information more efficiently, speeding up network training and preventing gradient vanishing issues.
Efficient-NetV2S	The application of progressive learning involves adjusting regularization and image size through MBConv and Fused-MBConv layers.	- The MBConv and Fused-MBConv blocks are capable of extracting features in a progressive manner while preserving information and reducing the number of parameters, thus accelerating the training process. - The MBConv and Fused-MBConv blocks also maintain higher accuracy results and lower Floating Point Operations (FLOPs) compared to conventional convolution layers. - With the implementation of progressive learning, the computational load is reduced due to the adjustment of image size and applied regularization.

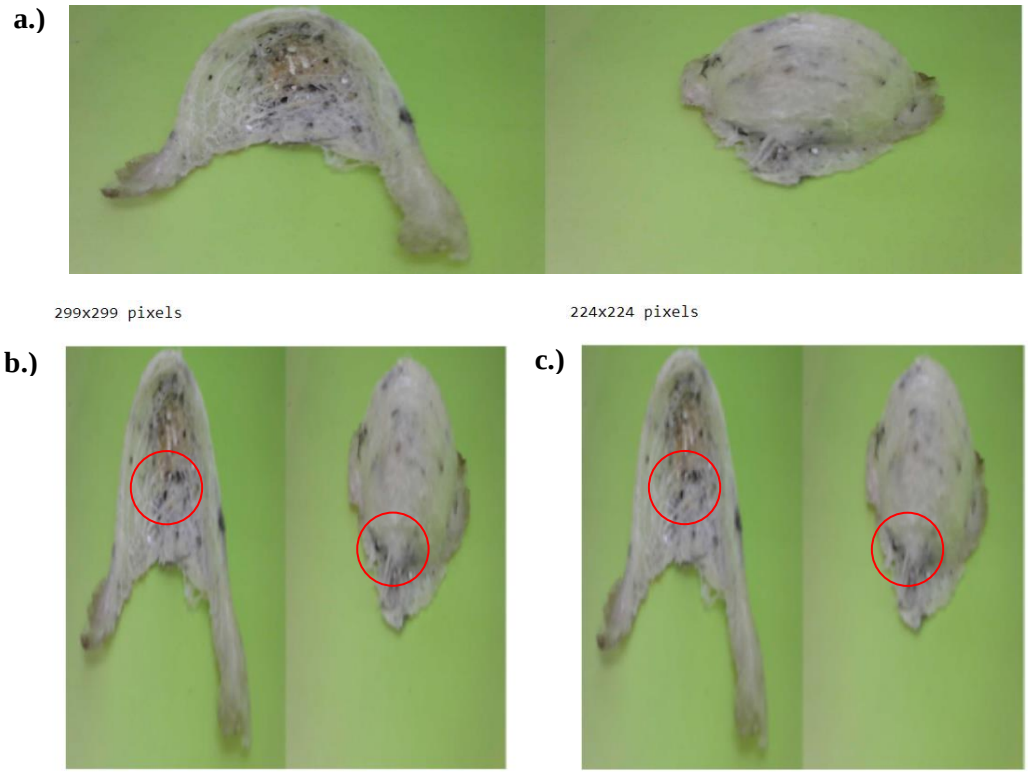


Figure 6. Input images of EBN before being trained by the Pt-CNN model categorized as follows: a) Original image, b) Preprocessed to 299 × 299 pixels, and c) Preprocessed to 224 × 224 pixels, with red circles indicating areas of significant image differences.

e. The different input preprocessing influences the model's performance in extracting important features. The result of the preprocessed input resolution of the InceptionResnetV2 model is higher than the Resnet50 and EfficientNetV2S models. Before being trained by the model, all EBN images were internally preprocessed to 299 × 299 pixels for InceptionResnetV2 model and 224 × 224 pixels for the Resnet50 and EfficientNetV2S models. The black color intensity of swiftlet feathers is more prominent in the 299 × 299 pixels, compared to the 224 × 224 pixels size, as indicated by the red circle in the image (Figure 6). The difference in color intensity between high and lower-resolution images affected the pixel values in the RGB input matrix processed by the convolutional layers. Darker color intensities had lower pixel values, with more apparent black color intensity serving finer details. This enabled the InceptionResnetV2 model to extract subtle features not visible in lower-resolution images, accessing more detailed visual information. Logically, a larger area with a high black color intensity indicates a higher density of swiftlet feathers, requiring a higher enzyme dosage. Thus, input images with higher resolutions significantly contribute to a more focused area of swiftlet feathers, resulting in higher accuracy performance of the model. This research finding is also consistent with several studies that have demonstrated improved model performance with higher image resolutions (Kannojia and Jaiswal, 2018; Sabottke and Spieler, 2020; Thambawita et al., 2021).

Conclusions

The Pt-CNN employed feature extraction through convolutional and pooling layers, followed by a final classification stage to determine the enzyme dosage class. Resnet50 used bottleneck blocks with residual connections, InceptionResnetV2 combined Inception modules with residuals, and EfficientNetV2S employed MBConv and Fused-MBConv layers. InceptionResnetV2 exhibited the best performance, achieving a validation accuracy of 96.18%, validation loss of 0.5960, train accuracy of 97.92%, and train loss of 0.4153. Further research is needed to study the application of serine protease enzyme dosage to EBN and test the model performance in testing datasets to ensure the models function accurately in the new datasets.

Acknowledgment

The authors would like to thank the Indonesia Ministry of Education, Culture, Research and Technology for providing the funding during the 2021 National Student Creativity scheme. Also, the team (Rizal Arifiandika, Bagas Rohmatulloh, and Riris Waladatun Nafi'ah) for the assistance in collecting datasets.

Declarations

Conflict of interests The authors declare no competing interests.

References

Alaeddine, H., and Jihene, M. (2021) 'Deep residual network in network', *Computational Intelligence and Neuroscience*, 2021, pp. 1–9

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., and Farhan, L. (2021) 'Review of deep learning: concepts, CNN architectures, challenges, applications, future directions', *Journal of Big Data*, 8(53), pp. 1-74

Amidi, A., Amidi, S., Vlachakis, D., Megalooikonomou, V., Paragios, N., and Zacharaki, E. I. (2018) 'EnzyNet: enzyme classification using 3d convolutional neural networks on spatial representation', *PeerJ*, 2018(5), pp. 1–11

BaniMustafa, A., Qattous, H., Ghabeish, I., and Karajeh, M. (2023) 'A machine learning hybrid approach for diagnosing plants bacterial and fungal diseases', *International Journal of Advanced Computer Science and Applications*, 14(1), pp. 912–921

Barry-Straume, J., Tschannen, A., Engels, D. W. and Fine, E. (2018) 'An evaluation of training size impact on validation accuracy for optimized convolutional neural networks', *SMU Data Science Review*, 1(4), pp. 1–17

Bemporad, A., and Piga, D. (2021) 'Global optimization based on active preference learning with radial basis functions', *Machine Learning*, 110, pp. 417-448

Bonet, D., Ortega, A., Ruiz-Hidalgo, J., and Shekkizhar, S. (2021) 'Channel-wise early stopping without a

validation set via nnk polytope interpolation', *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2021 - Proceedings*, pp. 351–358.

Chan, G. K. (2018) 'Searching for active ingredients in edible bird's nest', *Journal of Complementary Medicine & Alternative Healthcare*, 6(2), pp. 1–5

Chieng, H. H., Wahid, N., Pauline, O., and Perla, S. R. K. (2018) 'Flatten-t swish: A thresholded relu-swish-like activation function for deep learning', *International Journal of Advances in Intelligent Informatics*, 4(2), pp. 76–86

Chollet, F. (2017) 'Xception: Deep learning with depthwise separable convolutions', *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, pp. 1251–1258.

Chun, M., Jeong, H., Lee, H., Yoo, T., and Jung, H. (2022) 'Development of korean food image classification model using public food image dataset and deep learning methods', *IEEE Access*, 10, pp. 128732–128741

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020) 'RandAugment: Practical automated data augmentation with a reduced search space', *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 1–10.

Darafsh, S., Ghidary, S. S., and Zamani, M.S. (2021) 'Real-Time activity recognition and intention recognition using a vision-based embedded system', *Computer Vision and Pattern Recognition*, pp. 1–16.

Daud, N., Yusop, S. M., Babji, A. S., Lim, S. J., Sarbini, S. R., and Yan, T. H. (2019) 'Edible bird's nest: Physicochemical properties, production, and application of bioactive extracts and glycopeptides', *Food Reviews International*, 37(2), pp. 177–196

Demir, A., and Yilmaz, F. (2020) 'Inception-resnet-v2 with leakyrelu and averagepooling for more reliable and accurate classification of chest x-ray images', *TIPTEKNO 2020 - Tip Teknolojileri Kongresi - 2020 Medical Technologies Congress, TIPTEKNO 2020*, pp. 1–4.

El-Magd, L. M. A., Elsonbaty, A. A., and Elbelkasy, M. S. A. (2022) 'Enhanced ct-image for covid-19 classification using resnet 50', *Journal of Theoretical and Applied Information Technology*, 100(12), pp. 3830–3840

Elharrouss, O., Akbari, Y., Almaadeed, N., and Al-Maadeed, S. (2022) 'Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches', *Computer Vision and Pattern Recognition*, pp. 1–23.

Gadze, J. D., Bamfo-Asante, A. A., Agyemang, J. O., Nunoo-Mensah, H., and Opare, K. A. B. (2021) 'An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers', *Technologies*, 9(1), pp. 1–22

Galanis, N.I., Vafiadis, P., Mirzaev, K.G. and Papakostas, G.A. (2022) 'Convolutional neural networks: A roundup and benchmark of their pooling layer variants', *Algorithms*, 15(11), pp. 1–19

Goodfellow, I., Bengio, Y., and Courville, A. (2016) *Deep Learning*. Cambridge: MIT Press.

Gopinath, S. C. B., Anbu, P., Lakshmipriya, T., Tang, T. H., Chen, Y., Hashim, U., Ruslinda, A. R., and Arshad, M. K. M. (2015) 'Biotechnological aspects and perspective of microbial Keratinase production', *BioMed Research International*, 2015, pp. 1–10

Hammad, I., and El-Sankary, K. (2019) 'Practical considerations for accuracy evaluation in sensor-based machine learning and deep learning', *Sensors (Switzerland)*, 19, pp. 1–13

He, K., Zhang, X., Ren, S., and Sun, J. (2015) 'Deep residual learning for image recognition', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1–12.

Hendrawan, Y., Damayanti, R., Riza, D. F. A., and Hermanto, M. B. (2021) 'Classification of water stress in cultured Sunagoke moss using deep learning', *Telkomnika (Telecommunication Computing Electronics and Control)*, 19(5), pp. 1594–1604

Ho, S. Y., Phua, K., Wong, L., and Bin Goh, W. W. (2020) 'Extensions of the external validation for checking learned model interpretability and generalizability', *Patterns*, 1(8), pp. 1–9

Hu, J., Shen, L., Albanie, S., Sun, G., and Wu, E. (2020) 'Squeeze-and-excitation networks', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(8), pp. 2011–2023

Iman, M., Arabnia, H. R. and Rasheed, K. (2023) 'A Review of deep transfer learning and recent advancements', *Technologies*, 11(2), pp. 1–14

Islam, M. A., Shuvo, M. N. R., Shamsojjaman, M., Hasan, S., Hossain, S., and Khatun, T. (2021) 'An Automated convolutional neural network based approach for paddy leaf disease detection', *International Journal of Advanced Computer Science and Applications*, 12(1), pp. 280–288

Ito, Y., Matsumoto, K., Usup, A. and Yamamoto, Y. (2021) 'A sustainable way of agricultural livelihood: edible bird's nests in Indonesia', *Ecosystem Health and Sustainability*, 7(1), pp. 1–10

Jong, C. H., Tay, K. M., and Lim, C. P. (2013) 'Application of the fuzzy failure mode and effect analysis methodology to edible bird nest processing', *Computers and Electronics in Agriculture*, 96, pp. 90–108

Kannojia, S. P., and Jaiswal, G. (2018) 'Effects of varying resolution on performance of cnn based image classification an experimental study', *International Journal of Computer Sciences and Engineering*, 6(9), pp. 451–456

Kulathunga, N., Ranasinghe, N. R., Vrinceanu, D., Kinsman, Z., Huang, L., and Wang, Y. (2021) 'Effects of nonlinearity and network architecture on the performance of supervised neural networks', *Algorithms*, 14(2), pp. 1–17

Kumar, V., and Garg, M.. (2018) 'Deep learning as a frontier of machine learning: A Review', *International Journal of Computer Applications*, 182(1), pp. 22–30

Laraba, S., Tilmanne, J., and Dutoit, T. (2019) 'Leveraging pre-trained CNN models for skeleton-based action recognition', *Computer Vision Systems: 12th International Conference, ICVS 2019*, pp. 612–626.

Li, S., Yuan, S., Liu, S., Wen, J., and Huang, Q. (2022) 'Research on an accuracy optimization algorithm of kriging model based on a multipoint filling criterion', *Mathematics*, 10, pp. 1-11

Lin, M., Chen, Q., and Yan, S. (2014) 'Network in network', *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, pp. 1–10.

Liu, D., Zhang, L., Lai, X., and Liu, H. (2022) 'Image feature selection embedded distribution differences between classes for convolutional neural network', *Applied Soft Computing*, 131, pp. 1–12

Liu, M., Xie, T., Cheng, X., Deng, J., Yang, M., Wang, X., and Liu, M. (2022) 'Focused dropout for convolutional neural network', *Applied Sciences (Switzerland)*, 12(15), pp. 1–14

Looi, H. Q., and Omar, A. R. (2016) 'Swiftlets and edible bird's nest industry in Asia', *Pertanika Journal of Scholarly Research Reviews*, 2(1), pp. 32–48

Lu, Y., Huo, Y., Yang, Z., Niu, Y., Zhao, M., Bosiakov, S., and Li, L. (2022) 'Influence of the parameters of the convolutional neural network model in predicting the effective compressive modulus of porous structure', *Frontiers in Bioengineering and Biotechnology*, 10, pp. 1–11

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C. A., and Olvera-Olvera, C. A. (2020) 'Comparison of convolutional neural network architectures for classification of tomato plant diseases', *Applied Sciences (Switzerland)*, 10(4), pp. 1–15

Meng, G. K., Kin, L. W., Han, T. P., Koe, D., and Keen Raymond, W. J. (2017) 'Size characterisation of edible bird nest impurities: A preliminary study', *Procedia Computer Science*, 112, pp. 1072–1081

Merino, I., Azpiazu, J., Remazeilles, A., and Sierra, B. (2021) '3D convolutional neural networks initialized from pretrained 2D convolutional neural networks for classification of industrial parts', *Sensors (Switzerland)*, 21(4), pp. 1–18

Navone, L., and Speight, R. (2018) 'Understanding the dynamics of keratin weakening and hydrolysis by proteases', *PLoS ONE*, 13(8), pp. 1–21

Nazir, U., Khurshid, N., Bhimra, M. A., and Taj, M. (2019) 'Tiny-Inception-resNet-v2: Using deep learning for eliminating bonded labors of brick kilns in South Asia', *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019*, pp. 1–6.

Patel, K., and Wang, G. (2022) 'A discriminative channel diversification network for image classification', *Pattern Recognition Letters*, 153, pp. 176–182

Pawar, A., Singh, M., Jadhav, S., Kumbhar, V., Singh, T., and Shah, S. (2023) 'Different crop leaf disease detection using convolutional neural network', *Proceedings of the International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022)*, pp. 966-979.

Ponzio, F., Urgese, G., Ficarra, E., and Di Cataldo, S. (2019) 'Dealing with lack of training data for convolutional neural networks: The case of digital pathology', *Electronics (Switzerland)*, 8(3), pp. 1–21

Ramachandran, P., Zoph, B., and Le, Q. V (2017) 'Swish: A self-gated activation function', *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, pp. 1–12.

Rani, N. S., Akshatha, K., and Koushik, K. (2023) 'Quality assessment model for handwritten photo document images', *Procedia Computer Science*, 218, pp. 133–142

Ryu, J. (2022) 'A visual saliency-based neural network architecture for no-reference image quality assessment', *Applied Sciences (Switzerland)*, 12(19), pp. 1–9

Ryu, J. (2023) 'Improved image quality assessment by utilizing pre-trained architecture features with unified learning mechanism', *Applied Sciences (Switzerland)*, 13(4), pp. 1–10

Ryu, J. Y., Kim, H. U., and Lee, S. Y. (2019) 'Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers', *Proceedings of the National Academy of Sciences of the United States of America*, 116(28), pp. 13996–14001

Sabottke, C. F., and Spieler, B. M. (2020) 'The effect of image resolution on deep learning in radiography', *Radiology: Artificial Intelligence*, 2(1), pp. 1–7

Sharma, A. K., and Foroosh, H. (2020) 'Slim-CNN: A light-weight CNN for face attribute prediction', *Proceedings - 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2020*, pp. 329–335.

Shazia, A., Xuan, T. Z., Chuah, J. H., Usman, J., Qian, P., and Lai, K. W. (2021) 'A comparative study of multiple neural network for detection of COVID-19 on chest X-ray', *Eurasip Journal on Advances in Signal Processing*, 2021(50), pp. 1–16

Sikander, R., Wang, Y., Ghulam, A., and Wu, X. (2021) 'Identification of enzymes-specific protein domain based on DDE, and convolutional neural network', *Frontiers in Genetics*, 12, pp. 1–14

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014) 'Dropout: A simple way to prevent neural networks from overfitting', *Journal of Machine Learning Research*, 15, pp. 1929–1958

Sungsiri, A., Nonsiri, S., and Monsakul, A. (2022) 'The classification of edible-nest swiftlets using deep learning', *6th International Conference on Information Technology, InCIT 2022*, pp. 404–409.

Sunil, C. K., Jaidhar, C. D., and Patil, N. (2022) 'Cardamom plant disease detection approach using efficientNetV2', *IEEE Access*, 10, pp. 789–804

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2016) 'Inception-v4, inception-ResNet and the impact of residual connections on learning', *31st AAAI Conference on Artificial Intelligence*, pp. 1–12.

Tan, M., and Le, Q. V. (2021) 'EfficientNetV2: Smaller models and faster training', *International Conference on Machine Learning, 2021*, pp. 1–11.

Taye, M. M. (2023) 'Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions', *Computation*, 11(3), pp. 1–23

Thambawita, V., Strümke, I., Hicks, S. A., Halvorsen, P., Parasa, S., and Riegler, M. A. (2021) 'Impact of image resolution on deep learning performance in endoscopy image classification: An experimental study using a large dataset of endoscopic images', *Diagnostics*, 11(12), pp. 1–9

Tsalera, E., Papadakis, A., Samarakou, M., and Voyiatzis, I. (2022) 'Feature extraction with handcrafted methods and convolutional neural networks for facial emotion recognition', *Applied Sciences (Switzerland)*, 12(17), pp. 1–20

Utomo, B., Rosyidi, D., Radiati, L. E., Tri Puspaningsih, N. N., and Proborini, W. D. (2018) 'Use of keratinase to maintain pre-washing glycoprotein profiles of edible bird's nest', *Drug Invention Today*, 10(2), pp. 2986–2990

Wan, X., Ren, F., and Yong, D. (2019) 'Using inception-resnet V2 for face-based age recognition in scenic spots', *Proceedings of 2019 6th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2019*, pp. 159–163.

Wang, Q., Zhang, Y., Ge, H., Jiang, Y., and Qin, Y. (2023) 'Identification of Rice Freshness Using Terahertz Imaging and Deep Learning', *Photonics*, 10(5), pp. 1–12

Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. (2018) 'Convolutional neural networks: An overview and application in radiology', *Insights into Imaging*, 9(4), pp. 611–629

Yeo, Y. H., and Yen, K. S. (2021) 'Impurities detection in intensity inhomogeneous edible bird's nest (EBN) using a u-net deep learning model', *International Journal of Engineering and Technology Innovation*, 11(2), pp. 135–145

Yue, B., Fu, J., and Liang, J. (2018) 'Residual recurrent neural networks for learning sequential representations', *Information (Switzerland)*, 9(3), pp. 1–14

Zhang, H., Wu, C., Zhang, Zhongyue, Zhu, Y., Lin, H., Zhang, Zhi, Sun, Y., He, T., Mueller, J., Manmatha, R., Li, M., and Smola, A. (2022) 'ResNeSt: split-attention networks', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2736–2746.

Zhao, Y., Han, S., Meng, Y., Feng, H., Li, Z., Chen, J., Song, X., Zhu, Y., and Yang, G. (2022) 'Transfer-learning-based approach for yield prediction of winter wheat from planet data and SAFY model', *Remote Sensing*, 14(21), pp. 1–17

Zhu, X., Yu, Y., Zheng, Y., Su, S., and Chen, F. (2022) 'Bilinear attention network for image-based fine-grained recognition of oil tea (*Camellia oleifera Abel.*) cultivars', *Agronomy*, 12(8), pp. 1–15