A Novel Convolutional Neural Network Architecture for Pollen-Bearing Honeybee Recognition

Thi-Nhung Le[1], Thi-Minh-Thuy Le[2], Thi-Thu-Hong Phan[3]*, Huu-Du Nguyen[4], Thi-Lan Le[5]
Faculty of Information Technology, Vietnam National University of Agriculture, Hanoi, Vietnam[1,2]
School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam[1,5]
Department of Artificial Intelligence, FPT University, Danang, Vietnam[3]
School of Applied Mathematics and Informatics, Hanoi University of Sciences and Technology, Hanoi, Vietnam[4]

***Abstract*—Monitoring the pollen foraging behavior of honeybees is an important task that is beneficial to beekeepers, allowing them to understand the health status of their honeybee colonies. To perform this task, monitoring systems should have the ability to automatically recognize images of pollen-bearing honeybees extracted from videos recorded at the beehive entrance. In this paper, a novel convolutional neural network architecture is proposed for recognizing pollen-bearing and non-pollen-bearing honeybees from their images. The performance of the proposed model is illustrated based on a real dataset and the obtained results show that it performs better than some other state-of-the-art deep learning architectures like VGG16, VGG19, or Resnet50 in terms of both accuracy and execution time. Thus, the proposed model can be considered an effective algorithm for designing automatic honeybee colony monitoring systems.**

***Keywords*—Pollen-bearing honeybee; image classification; convolutional neural network; honeybee monitoring system; pollen dataset**

I. Introduction

Honeybees bring great benefits to human life. Products from honeybees such as honey, propolis, and pollen bring high economic efficiency. They are both popular components for daily consumption and an important source of raw materials in the production of medicines and beauty care. The global honey market was valued at 8.9 billion dollars by the year 2022, and it is predicted to reach 12.6 billion dollars in the year 2030, according to a report in VANTAGEMarketResearch[1]. In addition, honeybees are known as the most common and effective pollinators [1]. A large-scale survey in [2] has shown that honeybee pollination has contributed to increased yields and improved quality for many crops worldwide. For example, in the US, honeybees have increased fruit setting by 60% and seed yield by 20% for almonds; in Argentina, honeybees simultaneously increase fruit setting (by 15%) and the content of fruit sugar for apples, thereby increasing profits by 70%; and in Brazil, honeybees increased soybean yield by 18.9%. Besides that, the pollination of honeybees also contributes to the preservation of the diversity of plant ecosystems.

To take care of honeybee colonies, beekeepers must monitor the health of the colonies regularly. This task requires gathering information about the activities, status, and behaviors of honeybees in the colony, including pollen foraging behavior. In fact, pollen is the leading food of honeybees. It provides proteins, lipids, vitamins, and minerals necessary for the growth and reproduction of honeybees [3]. Information about the foraging behavior of honeybees can bring valuable understanding about the pollen source status in the habitat, the need for food, the increase of individuals, and the health of the whole honeybee colony. As a result, it allows beekeepers to understand the status of their honeybee colonies and detect unusual problems in the colonies for timely intervention. Recognizing honeybees bringing pollen back to the hive is then an effective solution for monitoring the beehives.

In recent years, thanks to the application of IoT (Internet of Things) technologies, several automatic honeybee monitoring systems have been deployed. These systems use surveillance cameras to record the activities of honeybees at the beehive entrance, then use different techniques to extract and analyze information from the recorded images [4]. Due to its powerful ability in image data processing, the Convolutional Neural Network (CNN) is perhaps the most widely used technique in these systems. In the context of recognizing pollen-bearing honeybee images, many CNN-based models have been designed. For example, several well-known CNN-based models like VGG16, VGG19, Resnet50, and DarkNet53 have been applied in [5] towards precise recognition of pollen-bearing honeybees. Rodriguez et al. [6] tested with several types of CNN architectures and showed that shallow-CNN architecture gives higher recognition accuracy than machine learning methods such as SVM (Support Vector Machine), Naive Bayes, or K-nearest neighbors. The authors also provided a real dataset of pollen-bearing and non-pollen-bearing honeybee images. However, we have found that there are a few mislabeled samples in this dataset where some images of non-pollen-bearing honeybees were assigned as pollen-bearing honeybees and vice versa. In addition, the use of complex structures for these CNN-based models requires a large number of samples for training and testing, leading to a significant cost of calculation and resources for the execution. To provide an effective model for designing automatic beehive monitoring systems, in this study, we propose a novel CNN architecture for recognizing pollen-bearing honeybee images. Here, we aim to find a model that is better than existing models in terms of both efficiency and execution cost. The performance of the proposed model is validated by comparing it with several other complex models like VGG16, VGG19 [7], and Resnet50 [8] architectures using the same dataset.

In summary, the main contributions of the study are as follows:

[1]https://www.vantagemarketresearch.com/industry-report/honey-market-2138

- We adjust a dataset published in a previous study by assigning correct labels to some previously mislabeled samples.
- We propose a novel structure for a CNN-based model to classify the images of pollen-bearing and non-pollen-bearing honeybees. We also suggest a data augmentation technique to handle the cases where there are few observed samples.
- We verify the efficacy and cost of the proposed method by extensive experiments, achieving an absolute accuracy for pollen-bearing honeybee recognition on the testing set.

The rest of the paper is organized as follows. Section II is to present the recent related works in the literature. In Section III, we describe the dataset considered in the study and the proposed CNN architecture. The experiments and the obtained results are presented in Section IV. Finally, Section V is for some concluding remarks.

II. Related Work

In this section, we briefly discuss recent studies related to the classification models of pollen-bearing and non-pollen-bearing honeybees.

For pollen-bearing honeybee image recognition, many studies relied on image processing techniques and conventional machine learning algorithms [9]. Babic et al. [10] applied background subtraction using a Mixture of Gaussian for the segmentation of honeybees. Then, based on the difference in color variance and eccentricity features between pollen-bearing and non-pollen-bearing honeybees, the authors used the Nearest Mean Classifier to classify them. The accuracy achieved by the classifier is 88.7%. However, the classification accuracy depends on the results of background subtraction and the light source in the video recording area. Yang and Collins [11] used color thresholding and the Mixture of Gaussian to detect and extract images of individual honeybees in frames captured from video recorded at the beehive entrance and track them using Kalman filter and Hungarian algorithm. The bee blob analysis method from the binary image of each frame was applied to remove the main body of the honeybee and retain only the pollen blobs. The two main features of the pollen blobs, including the area of the pollen blobs and the location of them relative to the bee's body, will be used to remove noise blobs. Finally, the pollen sacs detection results are combined with the previous honeybee detection and tracking model to identify if a honeybee bears pollen sacs. The test results with several videos show that the pollen measurement model has the highest sensitivity of 76%. In [12], the authors conducted experiments using two methods to segment honeybee images: the k-means algorithm and the algorithm that only considers the b component of the CIE LAB color space. Then, the SVM classifier with Gaussian kernel was used to classify the pollen-bearing and non-pollen-bearing honeybee images based on the Dense SIFT (Dense Scale Invariant Feature Transform) descriptors and the VLAD (Vector of Locally Aggregated Descriptors) encoder. The test results show that the method that combines the k-means segmentation algorithm and the classifier based on the descriptors on the decorrelated channels gives the highest value of the area under the ROC curve (AUC-ROC) at 0.915. In [6], authors performed the classification of pollen-bearing and non-pollen-bearing honeybees with three traditional methods: K-Nearest Neighbor algorithm, Naive Bayes statistical algorithm, and Support Vector Machines with linear and non-linear kernel functions. The results show that the SVM RBF method (Support Vector Machine with Radial Basis Function) with PCA (Principal Component Analysis) preprocessing technique and using the Gaussian feature map gives the highest accuracy at 91.16%.

In machine learning models, the important features were often selected from the inputs manually and subjectively. This can greatly affect the performance of the model once the key elements are not considered. To overcome this disadvantage, recent research suggest using models based on deep learning, more specifically, different CNN architectures. Rodriguez et al. [6]conducted experiments with 1-layer and 2-layer Shallow-CNN models, VGG16, VGG19, and Resnet50. The results show that all these models achieve high accuracy in which Shallow-CNN with small step size gives the highest accuracy of 96.4%, followed by VGG19, VGG16, and Resnet50 with an accuracy of 90.2%, 87.2%, 61.7%, respectively. Sledevič [13] investigated different CNN architectures with different numbers of hidden layers. After several experiments, the author stated that the architecture consisting of three hidden layers 7-7, 5-5, and 3-3 is the most suitable for classifying pollen-bearing and non-pollen-bearing honeybees, achieving a 94% accuracy. In [14], a pollen sac detection model on an individual honeybee image is used to classify a honeybee image as a pollen-bearing or non-pollen-bearing honeybee. This detection model uses Faster R-CNN architecture with the core for classification as VGG16. When a pollen sac is detected on an individual honeybee image, it is marked by a bounding box labeled "pollen" and a numerical value that is the confidence score of the detection. When the confidence score is greater than or equal to a predefined threshold, it is counted as a pollen sac, and the individual honeybee image is counted as a pollen-bearing honeybee image. This model has a pollen detection accuracy of 81.5%. Ngo et al. [15] relied on the YOLOv3-tiny model to detect and classify objects. Since YOLO-v3 treats the object classification problem as a regression problem, whereby an input image is divided into grid cells, each grid cell is responsible for detecting a target honeybee. This model allows the simultaneous detection of multiple objects on a frame belonging to one of two classes of pollen-bearing and non-pollen-bearing honeybees. The obtained results from this study showed that the proposed model gives a classification accuracy of 94%. In another research, nine different pre-trained CNN models, including VGG16, VGG19, Resnet50, ResNet101, Inception V2, Inception V3, Xception, DenseNet201, and DarkNet53 have been explored [5]. The authors also considered the influence of color by applying some image preprocessing techniques to the input dataset. The experimental results showed that the DarkNet53 and VGG16 architectures attained higher recognition accuracy than the others. In [16], the authors first tested the image classification using the transfer learning method with seven pre-trained Deep Neural Networks (DNNs) including AlexNet, DenseNet201, GoogLeNet, ResNet101, ResNet18, VGG16, and VGG19. After that, the authors continued to experiment with the SVM classifier using shallow features, deep features, and shallow+deep features extracted from the DNNs. Three different standard datasets were used for the training and evaluation of models. The experimental results showed no significant difference in performance between them. For the pollen-bearing honeybee image dataset, the transfer learning method with pre-trained DNN yielded the highest accuracy of 99.07%.

III. Materials and Methods

A. Data Description

In this study, the Pollen dataset published in [6] has been considered. The dataset is available and can be accessed publicly at GitHub[2]. The authors stated that at the beginning, there were 810 honeybee images extracted and manually annotated from videos taken at the beehive entrance under natural light conditions. However, after being curated, many have been removed due to misclassification. Thus, the final data downloaded from the source above contains only 714 images. The photo was built by fixing the size of the cropping rectangle to 180×300 pixels, containing a fully visible image of a single honeybee. The images are also adjusted to make sure that the honeybees are facing upward in all images. Each photo was then labeled with pollen or non-pollen. Out of 714 images, 369 are labeled as pollen (P) and the rest 345 are non-pollen (NP). Fig. 1 presents some images of pollen-bearing and non-pollen-bearing honeybees in this dataset.

Fig. 1. Images of pollen-bearing honeybees (a) and non-pollen-bearing honeybees (b).

Based on this dataset, we have conducted several experiments to investigate the performance of the proposed model. The obtained results have shown that our model has misidentified some images from pollen-bearing honeybees to non-pollen-bearing ones and vice versa. We then carried out a thorough analysis of these misidentified images and found that some of them were mislabeled. According to our knowledge, images *NP24865-145r* and *NP27452-203r* are images of honeybees that bear pollen but are annotated as non-pollen-bearing ones (NP). Meanwhile, images *P7660-97r, P7776-99r, P11440-32r*, and *P11762-35r* in the dataset are images of honeybees that do not bear pollen but are annotated as pollen-bearing honeybees (P). Fig. 2 shows these mislabeled images.

[2]https://github.com/piperod/PollenDataset

TABLE I. The Structure of the Pollen Dataset

Dataset	Class label	Number of images
Original	Pollen	369
	Non-pollen	345
Corrected	Pollen	367
	Non-pollen	347

We have relabeled these images and used the corrected dataset to train and test the performance of the proposed model (as well as control models). The structure of the original Pollen dataset and the relabeled one is summarized in Table I.

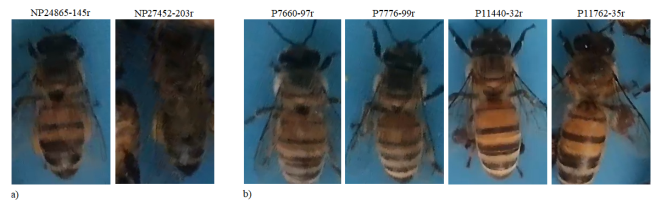

Fig. 2. Mislabeled images in the Pollen dataset, from Pollen to Non-Pollen (a) and vice versa (b).

B. Proposed Method

1) Convolutional neural networks: Convolutional Neural Networks (CNNs) refer to a well-known deep learning algorithm specialized in handling image data. The basic architecture of a CNN model consists of three main types of layers, as displayed in Fig. 3.

The functions of each type of layer are as follows:

- The first type of layer of a basic CNN architecture is Convolution, the core of a CNN model used to extract various features from the input. The mathematical convolutional operation is performed in this layer, between the input and a filter. The dot product is taken between the filter and the parts of the input by sliding the filter over the image. The output from each layer containing information about the image like corners and edges is then fed to the next layer to learn other input features.
- Following the convolutional layers are the Pooling layers. These layers summarize the features extracted from the previous convolution layers, aiming to decrease the size of the obtained feature map and reduce computational costs. Several types of pooling operations can be used in a CNN model depending on the specific situation, such as Max Pooling and Average Pooling.
- The last Fully connected layers perform the classification task based on the features extracted from previous layers, mapping the representation between the input and the output. They generate scores for each class, then use them for the final classification.

In general, there is no universal optimal model for all datasets. For different problems, one should design different models with different structures to achieve the best performance. Choosing the right model is the key to solving many

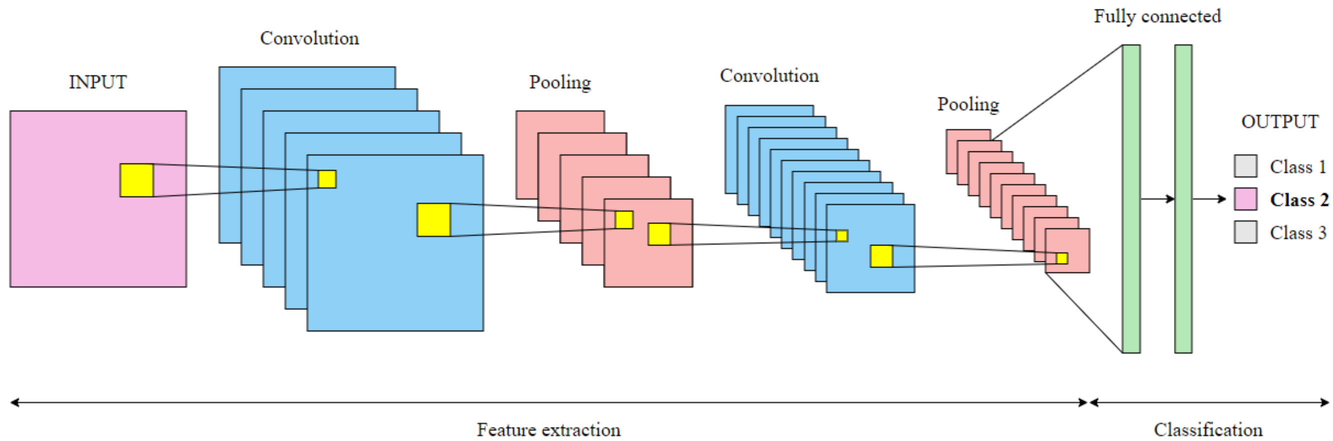

Fig. 3. Architecture of a basic CNN model.

problems in practice. More introduction and discussion about the architectures of CNN-based deep learning model can be seen in [17] and [18].

2) The proposed CNN model: As discussed above, the CNN architectures have been widely applied to the problem of recognizing pollen-bearing honeybees, some of them achieved up to about 99% accuracy. Several transfer learning models whereby different CNN architectures such as VGG16, VGG19, Resnet50, and Resnet101, etc. are pre-trained and some SVM classification models are based on features extracted from CNN architectures, were presented in [5] and [16]. However, the use of these models requires significant costs due to their complex architectures. To overcome this problem and aim for simplicity and efficiency in use, we thought about using a basic CNN model. This idea can be verified by investigating the basic CNN structures with different hyperparameters. In particular, using the Grid search method, we have figured out an optimal architecture for the CNN model in classifying pollen-bearing and non-pollen-bearing honeybees. The proposed architecture comprises:

- 4 convolutional layers equipped with a ReLU (Rectified Linear Unit) activation function,
- 5 max-pooling layers,
- 1 flatten layer, a dense layer with a ReLU activation function, and a dense layer with a Sigmoid activation function.

The use of this simple architecture obviously makes the model lighter than other pre-trained deep-learning models like VGG16, VGG19, and Resnet50 which contain more layers and parameters. In addition, in this study, we use some data augmentation techniques such as image rescaling, random rotating, shifting (horizontally and vertically), shearing, random zooming, random flipping, and nearest filling. These techniques enrich the data by generating different variations from the original images which will be used in different epochs of the model training process, thereby improving the performance of the classification model. Fig. 4 shows a visualization of several images obtained after applying data augmentation techniques to a honeybee image.

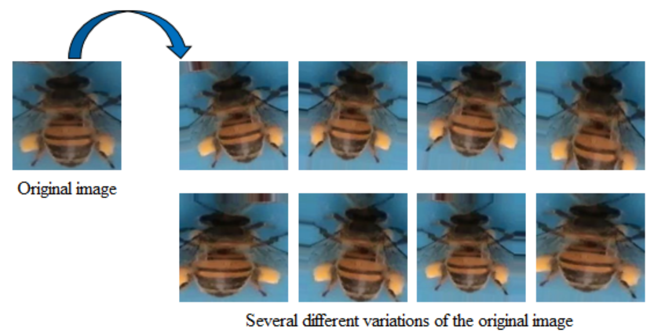

Fig. 4. An example of data augmentation.

The architecture of the proposed model is illustrated in Fig. 5. Each input is a 224×224 RGB image containing the image of an individual honeybee. After passing through the convolutional layers and the max-pooling layers to extract the important features, it is fed to fully connected layers. A predefined threshold is used to classify whether the honeybee image is a pollen-bearing honeybee or not. The performance of the proposed method will be discussed in the sequel.

IV. Experiments and Results

A. Experimental Setup

In this study, two schemes of splitting the Pollen dataset are considered. By the first scheme, as in several previous studies, we randomly divide the corrected Pollen dataset into three subsets, i.e., the training set, the validation set, and the testing set at a ratio of 6:1:3. Accordingly, 60% of the samples corresponding to 428 images (which include 221 images of pollen-bearing honeybees and 207 images of non-pollen-bearing honeybees) are for model training, 10% of the samples corresponding to 70 images (which include 36 images of pollen-bearing honeybees and 34 images of non-pollen-bearing honeybees) are for model validation, while the remaining 60% of the samples corresponding to 216 images (which include 110 images of pollen-bearing honeybees and 106 images of non-pollen-bearing honeybees) are for model testing. Moreover, to investigate the effect of partitioning data

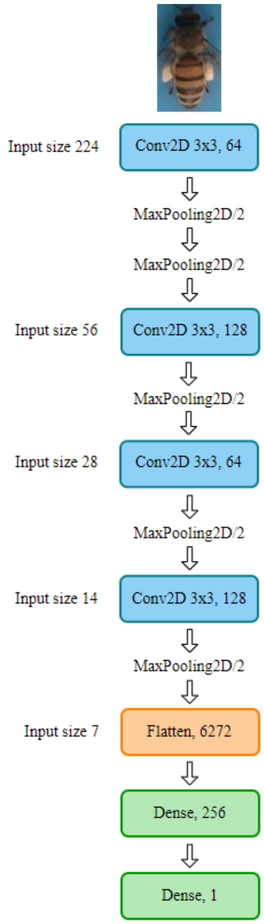

Fig. 5. The proposed CNN architecture.

TABLE II. Two Schemes of Splitting the Pollen Dataset

Scheme	Class label	Training set	Validation set	Testing set
1	Pollen	221	36	110
	Non-pollen	207	34	106
2	Pollen	183	36	148
	Non-pollen	173	34	140

into training, validation, and testing sets on the model performance, we design the second scheme where the Pollen dataset is split into the three subsets at a ratio of 5:1:4, namely fewer samples for training and more samples for testing compared to the first scheme. The details of the number of images in each subset of each scheme are presented in Table II.

The experiments were performed on Google Colab, using Python 3 Google Compute Engine backend (GPU) with a system RAM of 83.5 GB, GPU RAM of 40 GB, and Disk of 166.8 GB.

After hyperparameters tuning, hyperparameters are set up for model training as follows: batch size is 4, the number of epochs is 25, the initial learning rate is 0.001, and the optimizer is Adam.

B. Evaluation Metrics

To evaluate the performance of the proposed pollen-bearing honeybee recognizing model comprehensively, in this study, Precision, Recall, F1-score, Accuracy, Loss, and AUC-ROC metrics have been utilized. These are widely used metrics to assess the performance of the classification models.

- Precision, Recall, F1-score, and Accuracy are computed as follows:

$$\text{Precision}\& = \&\frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall}\& = \&\frac{TP}{TP+FN} \tag{2}$$

$$\text{F1-score}\& = \&\frac{2*Recall*Precision}{Recall+Precision} \tag{3}$$

$$\text{Accuracy}\& = \&\frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

  where
  - TP: number of pollen-bearing honeybees images that are properly classified as pollen-bearing honeybees images;
  - FP: number of non-pollen-bearing honeybees images that are misclassified as pollen-bearing honeybees images;
  - FN: number of pollen-bearing honeybees images that are misclassified as non-pollen-bearing honeybees images;
  - TN: number of non-pollen-bearing honeybees images that are properly classified as non-pollen-bearing honeybees images.

- Loss (Binary Cross Entropy) is calculated as follows:

$$\text{Loss} = \frac{-1}{N}\sum_{i=1}^{N}\left(y_i log_e p_i + (1-y_i)\, log_e\,(1-p_i)\right) \tag{5}$$

  where
  - N is the number of images;
  - y_i is the real label of the ith image ($y_i = 1$ if the $i-th$ image is a pollen-bearing honeybee image; $y_i = 0$ if the $i-th$ image is a non-pollen-bearing honeybee image);
  - p_i is the probability of the event predicting the $i-th$ image as a pollen-bearing honeybee image ($1-p_i$ is the probability of the event predicting the $i-th$ image as a non-pollen-bearing honeybee image).

- AUC-ROC: one of the most important evaluation metrics for checking any classification model's performance is calculated as the area under the ROC (Receiver Operating Characteristics) curve.

From the above definitions, the larger the values of Precision, Recall, F1-score, Accuracy, and AUC-ROC, and the smaller the value of Loss, the better the model is at classifying classes in a dataset.

TABLE III. THE PERFORMANCE OF CNN-BASED MODELS ON THE FIRST SCHEME OF SPLITTING THE POLLEN DATASET

Method	Precision	Recall	F1-score	Accuracy	Loss	AUC-ROC
VGG16 ([6])	-	-	-	87.20	-	-
VGG16 ([5])	-	-	-	94.80	-	-
VGG16 (ours)	98.67	98.58	98.61	98.61	10.86	0.9994
VGG19 ([6])	-	-	-	90.20	-	-
VGG19 ([5])	-	-	-	98.20	-	-
VGG19 (ours)	96.48	96.24	96.29	96.30	14.99	0.9961
Resnet50 ([6])	-	-	-	61.70	-	-
Resnet50 ([5])	-	-	-	86.60	-	-
Resnet50 (ours)	99.53	99.55	99.54	99.54	4.91	0.9955
Proposed CNN	**100.00**	**100.00**	**100.00**	**100.00**	**1.32**	**1.0000**

TABLE IV. THE EXECUTION TIME OF CNN-BASED MODELS ON THE FIRST SCHEME OF SPLITTING THE POLLEN DATASET

Method	Training time (s)	Testing time (s)
VGG16	272.830	133
VGG19	265.359	167
Resnet50	7471.634	39
Proposed CNN	**150.543**	**6**

C. Experimental Results and Discussion

The performance of the proposed method and the corresponding execution time on the first scheme of splitting the Pollen dataset is presented in Table III and Table IV. For the purpose of comparison, we also show the performance and the execution time of other CNN-based transfer learning methods using the same dataset in the literature.

Several important remarks can be drawn from these tables as follows.

- The proposed CNN model provides the best performance in terms of all the metrics. Although the use of other CNN-based models results in quite an impressive efficiency (for instance, an accuracy of 99.54% with Resnet50, and 98.61% with VGG16), our proposed method can still achieve higher performance, with an absolute efficiency of 100% for the metrics of Precision, Recall, F1-score, and Accuracy, and the maximum value is 1 for the metric of AUC-ROC. It also leads to the smallest value of the Loss of 1.32. This means the proposed model can accurately recognize all pollen-bearing and non-pollen-bearing honeybee images in the Pollen dataset.
- After correcting the mislabeled images, the accuracy of other CNN-based models is generally improved. For example, based on the original Pollen dataset, the Resnet50 model in [6] and [5] provided an accuracy of 61.70% and 86.60%, respectively. Meanwhile, on the corrected dataset, it can reach an accuracy of 99.54%.
- Thanks to its simple architecture with fewer layers than some other CNN architectures such as VGG16, VGG19, and Resnet50, the proposed model also reduces significantly the execution time for both training and testing processes. Indeed, it took only 150.543 seconds for training and 6 seconds for testing. Meanwhile, the second-fastest models asked for about 265.359 seconds for training (VGG19) and 39 seconds for testing (Resnet50), which are significantly slower than the proposed model, as can be seen in Table IV. This finding has a practical meaning as it allows the designing of efficient real-time recognition systems of pollen-bearing honeybees.

Fig. 6 and Fig. 7 show the curves of the training and validation accuracy, and the training and validation loss of the models compared. As can be seen from these figures, the proposed CNN model gives a higher performance than the other models. This is accordant with the results discussed above.

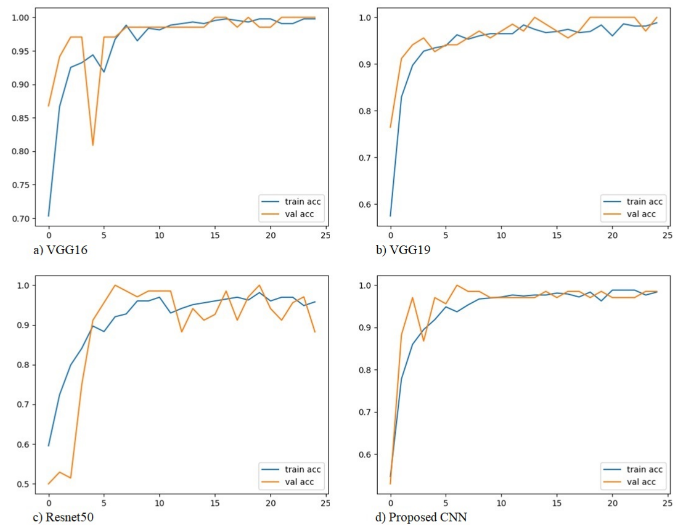

Fig. 6. Training and validation accuracy curve.

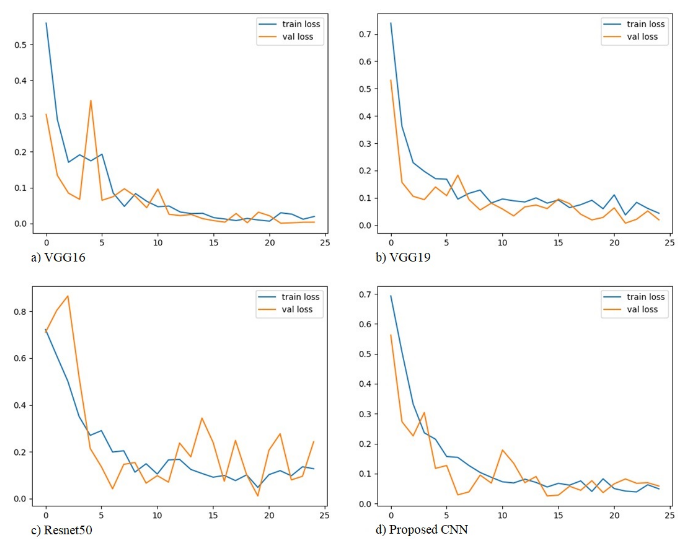

Fig. 7. Training and validation loss curve.

In Tables V and VI, we present the experimental results obtained from using the second scheme of splitting the Pollen dataset at the ratio 5:1:4. Since the scheme has not been considered in previous studies, we present the performance of our experiments only. The same result as the first scheme can also be witnessed in these two tables where our proposed method still brings the best Precision, Recall, F1-score, Accuracy, and Loss in the fastest processing time. However, the performance of all models, in this case, has been reduced a bit compared

TABLE V. THE PERFORMANCE OF CNN-BASED MODELS ON THE SECOND SCHEME OF SPLITTING THE POLLEN DATASET

Method	Precision	Recall	F1-score	Accuracy	Loss	AUC-ROC
VGG16	98.27	98.25	98.26	98.26	9.24	0.9961
VGG19	97.61	97.54	97.57	97.57	13.03	0.9941
Resnet50	95.73	95.40	95.47	95.49	9.58	**0.9966**
Proposed CNN	**98.95**	**98.99**	**98.96**	**98.96**	**8.02**	0.9921

TABLE VI. THE EXECUTION TIME OF CNN-BASED MODELS ON THE SECOND SCHEME OF SPLITTING THE POLLEN DATASET

Method	Training time (s)	Testing time (s)
VGG16	217.666	179
VGG19	241.568	223
Resnet50	6295.647	49
Proposed CNN	**123.874**	**10**

to the first scheme. For example, the proposed model does not achieve an absolute accuracy as in the first scheme, instead, it decreases to 98.96%. This result can be explained by the reduced number of samples in the training set. As a result, the model learns less from the training set, resulting in reduced classification performance.

V. CONCLUSION

In this paper, we have proposed a novel convolutional neural network model for classifying pollen-bearing and non-pollen-bearing honeybee images. Rather than using complex and pre-trained CNN models, we design a basic CNN architecture with a few layers, leading to a lighter and also more efficient model. We have also corrected some mislabeled samples from a widely used dataset in the literature. The performance of the proposed CNN model has been investigated and compared with other models based on this corrected dataset. The obtained results have shown that our method leads to the best performance in terms of both accuracy and execution time. In particular, it could identify correctly 100% all the pollen-bearing and non-pollen-bearing honeybee images from the testing set in the shortest time.

There are still several limitations that should be considered before deploying the use of the proposed model in designing automated systems to recognize pollen-bearing honeybees in practice. For example, the efficiency of the model was verified based on a small dataset that contains 714 images only. Its performance should be validated on other datasets with larger sizes. In addition, the choice of hyperparameters of the proposed architecture is suitable for the current dataset, but may not be for other datasets. Therefore, it would be better to have another method to find hyperparameters that are optimal for each dataset. From this point of view, some optimization algorithms such as Random search or Bayesian optimization could be applied for future work. In addition, the model can be applied to process honeybee images for some other related tasks, like counting the number of pollen-bearing honeybees (for the purpose of measuring the amount of pollen carried by honeybees to the hive), classifying pollen, or recognizing disease-carrying honeybees. However, its performance needs to be verified for each specific situation.

ACKNOWLEDGMENT

The authors would like to thank the Vietnam National University of Agriculture for sponsoring this research through the University-level science and technology project with the title "Research on deep learning algorithms of convolutional neural network and their applications in recognizing pollen-bearing honeybee images" and number "T2022-10-39".

REFERENCES

[1] U. Joshi, K. Kothiyal, Y. Kumar, and R. Bhatt, "Role of honeybees in horticultural crop productivity enhancement," *International Journal of Agricultural Sciences*, vol. 17, no. AAEBSSD, pp. 314–320, 2021.

[2] S. A. M. Khalifa, E. H. Elshafiey, A. A. Shetaia, A. A. A. El-Wahed, A. F. Algethami, S. G. Musharraf, M. F. AlAjmi, C. Zhao, S. H. D. Masry, M. M. Abdel-Daim, M. F. Halabi, G. Kai, Y. A. Naggar, M. Bishr, M. A. M. Diab, and H. R. El-Seedi, "Overview of bee pollination and its economic value for crop production," *Insects*, vol. 12, no. 8, 2021.

[3] K. A. Stoner, H. P. Hendriksma, and S. Tosi, "Pollen as food for bees: Diversity, nutrition, and contamination," *Frontiers in Sustainable Food Systems*, vol. 6, no. 1129358, 2023.

[4] I. F. Rodriguez, J. Chan, M. Alvarez Rios, K. Branson, J. L. Agosto-Rivera, T. Giray, and R. Mégret, "Automated video monitoring of unmarked and marked honey bees at the hive entrance," *Frontiers in Computer Science*, vol. 3, no. 769338, 2022.

[5] F. C. Monteiro, C. M. Pinto, and J. Rufino, "Towards precise recognition of pollen bearing bees by convolutional neural networks," *In Iberoamerican Congress on Pattern Recognition*, pp. 217–226, 2021, May.

[6] I. F. Rodriguez, R. Megret, E. Acuna, J. L. Agosto-Rivera, and T. Giray, "Recognition of pollen-bearing bees from video using convolutional neural network," in *In 2018 IEEE winter conference on applications of computer vision (WACV)*, 2018, March, pp. 314–322.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] S. Bilik, O. Bostik, L. Kratochvila, A. Ligocki, M. Poncak, T. Zemcik, M. Richter, I. Janakova, H. P., and K. Horak, "Machine learning and computer vision techniques in bee monitoring applications," *arXiv preprint arXiv:2208.00085*, 2022.

[10] Z. Babic, R. Pilipovic, V. Risojevic, and G. Mirjanic, "Pollen bearing honey bee detection in hive entrance video recorded by remote embedded system for pollination monitoring," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 3, pp. 51–57, 2016.

[11] C. Yang and J. Collins, "Improvement of honey bee tracking on 2d video with hough transform and kalman filter," *Journal of Signal Processing Systems*, vol. 90, pp. 1639–1650, 2018.

[12] V. Stojnić, V. Risojević, and R. Pilipović, "Detection of pollen bearing honey bees in hive entrance images," in *In 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2018, March, pp. 1–4.

[13] T. Sledevič, "The application of convolutional neural network for pollen bearing bee classification," in *In 2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 2018, November, pp. 1–4.

[14] C. Yang and J. Collins, "Deep learning for pollen sac detection and measurement on honeybee monitoring video," in *In 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2019, December, pp. 1–6.

[15] T. N. Ngo, D. J. A. Rustia, E. C. Yang, and T. T. Lin, "Automated monitoring and analyses of honey bee pollen foraging behavior using a deep learning-based imaging system," *Computers and Electronics in Agriculture*, vol. 187, no. 106239, 2021.

[16] S. K. Berkaya, E. S. Gunal, and S. Gunal, "Deep learning-based classification models for beehive monitoring," *Ecological Informatics*, vol. 64, no. 101353, 2021.

[17] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.