BIODIVERSITAS
Volume 23, Number 5, May 2022
Pages: 2561-2569

ISSN: 1412-033X
E-ISSN: 2085-4722
DOI: 10.13057/biodiv/d230537

Short Communication: Development of a protected birds identification system using a convolutional neural network

IRMAN HERMADI, WULANDARI♥, DHANA DHIRA
Department of Computer Science, Faculty of Mathematics and Natural Sciences, Institut Pertanian Bogor. Jl. Meranti, Kampus IPB Dramaga, Bogor 16680, West Java, Indonesia. Tel.: +62-251-8625584, ♥email: wulandari.ilkom@apps.ipb.ac.id

Manuscript received: 15 December 2022. Revision accepted: 26 April 2022.

Abstract. *Hermadi I, Wulandari, Dhira D. 2022. Short Communication: Development of a protected birds identification system using a convolutional neural network. Biodiversitas 23: 2561-2569.* The protected animals are the animals having small populations, a sharp decline in the number of individuals in the wild, or endemic. The government has banned owning, keeping, or trading these animals. The first step in conserving these animals is identification. The government of the Republic of Indonesia has defined 564 species of bird as protected. This issue becomes a challenge to bird species identification. This study aims to develop a web application that implements a convolutional neural network (CNN) model for image-based protected bird species identification. This study uses the images of ten protected bird species in Indonesia as the research subject. This study consists of the stages viz. data collection, data preprocessing, data splitting, CNN model development, model evaluation, and web development using the Prototyping method. This study has successfully developed a model that gained 97% accuracy, 98% precision, and 97% recall on testing data. The study utilized HTML, CSS, Javascript, and Tensorflow.js for Web development. The black-box testing result shows that the prototype is acceptable.

Keywords: Bird conservation, bird species identification, convolutional neural network, prototyping

Abbreviations: CNN: Convolutional Neural Network, FP: False Positive, FN: False-Negative, TN: True Negative, TP: True Positive

INTRODUCTION

Indonesia has been widely known for its mega biodiversity of faunas. Among those diverse faunas, some species have been valued high for their features such as their appearance, sound, organs, or some other values that make them high-demand subjects such as pets, trade, and hunting. These actions may cause population decline of those species and their extinction in the long term. Therefore, those acts need to be stopped to conserve wildlife and prevent wildlife extinction. Indonesia has put efforts into protecting wildlife by creating some legal umbrellas. The Government Regulation (PP) Number 7 of 1999 regulates the preservation of floras and faunas to prevent extinction by classifying the species with small populations, a sharp decline in the number of individuals in the wild, or endemic as protected species. The Government Regulation (PP) Number 5 of 1990 rules about the prohibition of keeping, owning, or trading protected species. The protected species list gets updated over time until the latest list was declared in Minister of Environment and Forestry of Republic Indonesia's Regulation No. P.106/ MENLHK / SETJEN / KUM.1 /6/2018 concerning protected species of plants and animals. This regulation lists the 919 protected floras and faunas; 564 are birds.

Identifying many species of birds found in the market or the field is a challenging task. However, identification is the first step in conserving the protected bird species. The ability of staff at the Natural Resources Conservation Center (KSDA), a government agency with authority in this field, and the citizens, to identify animal species is still limited as it is difficult to memorize all of the varying features of that large number of species. The officers need to identify and categorize the wildlife in the port as soon as possible. Meanwhile, the printed guidebook is relatively not easy to find a particular species. Therefore, we need tools to overcome this problem. Several applications to help with animal species identification are already available, such as iNaturalist, SmartScan, Merlin, and WildScan. iNaturalist is an application for naturalists, citizen scientists, or biologists to share their observations of nature across the globe. It has a concept of mapping its biodiversity observations, tagging the observed animal, and asking users' opinions about a particular animal. Smart scan, an Android application, identifies the animals and plants based on a picture. It also provided the detail of captured plants or animals. However, those two applications cover animals in general. On the other hand, a mobile application called Merlin Bird ID developed by Cornell Lab can detect bird species by image. However, to help the officers identify the animal as soon as possible, it needs an application focused on protected wildlife. WildScan is an application that provides an identification and reporting system to combat illegal treatments toward wildlife by utilizing the questionnaire given to the users (World Bank Group 2018). This application is built to aid law enforcement in West Africa in their work to combat the illegal wildlife trade. This tool needs a relatively long time to identify the species because the users need to identify the characteristics manually. However, the identifying application mainly focuses on Indonesia's protected wildlife is still not developed. In order to help the Indonesian officers identify the protected wildlife based on Regulation No. P.106/MENLHK/SETJEN/KUM.1/6/2018 faster, we aim to build an identifying application using the image. In this research, we focused on the bird species classification due to its similar features. For several years, the bird species classification topic has received much attention in computer vision as a promising application in studying the environment and biology.

Before this study, some studies had demonstrated the performance of convolutional neural network (CNN) in this area. Huang and Basanta (2019) demonstrated that CNN has a good performance in identifying the endemic birds' species in Taiwan with 93.98% accuracy than the support vector machine algorithm with 89% accuracy. Biswas et al. (2021) have done a comparative study among some CNN-based architectures such as MobileNet V2, DenseNet201, InceptionResNetV2, ResNet50, ResNet152V2, and Xception, which showed that MobileNet V2 performed better than other architectures on identifying the local bird species in Bangladesh with 96.71% accuracy. In Indonesia, Harjoseputro et al. (2020) demonstrated that MobileNet could perform better than the standard CNN on identifying protected birds in Indonesia with 70% accuracy by considering the performance and complexity tradeoffs. We aim to develop a web application that implements CNN with the sophisticated MobileNet V2 architecture to recognize the protected bird species in Indonesia.

MATERIALS AND METHODS

Data collection and preprocessing

This study uses data collected manually from April 2021 to June 2021. The dataset consists of 400 images of ten protected birds understudy in this initial stage. The images are distributed uniformly across the classes by the amount of 40 images for each class. Those ten species were selected because they were categorized as protected wildlife based on the regulation No P.106/2018, and their number of image available were sufficient for open access license. The images are collected from sources such as eBird and Oriental Bird Images, which have open access licenses and free-to-use for any education and research purposes. The species used in this study are based on the Minister of Environment and Forestry of Republic Indonesia's Regulation No. P.106/2018 (Table 1;

Figure 1).

Data collection was done manually to ensure that the collected images were relevant to the species and satisfied other requirements. The image resolution is at least 224 × 224 pixels. The displayed bird in the image must be clear as the main object. The image preferably does not have much noise and distortion. Varying poses and conditions of the birds and their environments can generally increase the CNN model's ability. During preprocessing, the images were cropped to focus more on the bird as the main object. The file size was reduced, and the aspect ratio was adjusted to 1:1 with a pixel size of 224 × 224, which would satisfy the prerequisites of MobileNet V2 (

B

A

Figure 2).

After the image was edited, the images were saved into separate directories with their respective species names. These images in the folder were then divided into three partitions: training, validation, and testing set randomly, with the proportion of 80% for the training set, 10% for the validation set, and the last 10% for the testing set. The dataset contained three sets with ten folders named with the ten species used in this study when the partition was done. CNN requires many images in its training process to reduce the overfitting risk. This study uses image augmentation to increase the number of images used in model training by transforming the images in the dataset (Zhang et al. 2020). The image augmentation was applied to the training set to increase the number of the training set. Several transformations, such as shifting, shearing, zooming with the intensity up to 20%, brightness adjustments in the range [0.3, 0.7], and flipping, were done to the training set on the augmentation phase. Then, the pixel values of the images get normalized to the range [0,1]. The image augmentation and pixel normalization were done by utilizing the function ImageDataGenerator provided by TensorFlow.

Table 1. List of the protected bird species and references of their sample image in Figure 1

English	Bahasa Indonesia	Scientific name	Image in Figure 1	Image source
Black-necked stork	Bangau leher-hitam	*Ephippiorhynchus asiaticus*	A	https://ebird.org/species/blnsto1
Bali Myna	Curik Bali	*Leucopsar rothschildi*	B	https://ebird.org/species/balmyn1
Javan green magpie	Ekek geling	*Cissa thalassina*	C	https://ebird.org/species/shtmag1
Java sparrow	Gelatik Jawa	*Lonchura oryzivora*	D	https://ebirdr.com/bird/java-sparrow
Palm cockatoo	Kakatua raja	*Probosciger aterrimus*	E	https://ebird.org/species/palcoc1
Red-headed trogon	Luntur harimau	*Harpactes erythrocephalus*	F	https://ebird.org/species/rehtro1

Green peafowl	Merak hijau	*Pavo muticus*	G	https://ebird.org/species/grepea1
Fairy pitta	Paok bidadari	*Pitta nympha*	H	https://ebird.org/species/faipit1
Blue-banded kingfisher	Raja udang kalung-biru Jawa	*Alcedo euryzona*	I	https://ebird.org/species/blbkin2
White-tailed flycatcher	Sikatan besar	*Cyornis concretus*	J	https://ebird.org/species/whtfly2

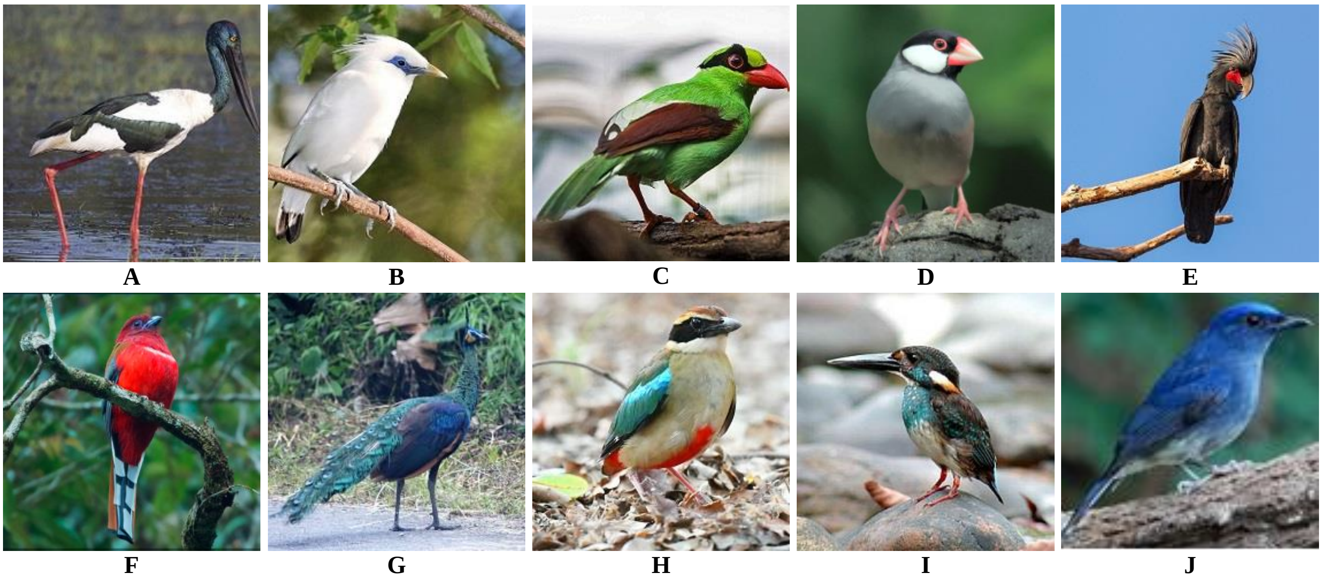

Figure 1. Sample images for each species that is being used in this study

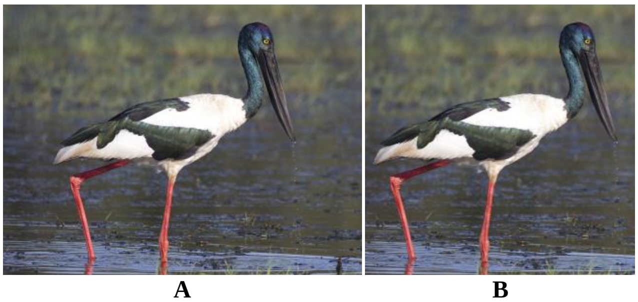

Figure 2. The image on editing process: A. Before editing; B. Edited image (https://ebird.org/species/blnsto1)

CNN model building, training, testing, and evaluation

Convolutional Neural Network (CNN)

CNN is a kind of artificial neural network that can process data with 2D patterns and is designated to learn the spatial hierarchy from the low-level image features to the high-level image features automatically. Therefore, CNN is suitable for machine learning using images as input (Yamashita et al. 2018). CNN consists of an input layer; convolution layers, which produce the feature map from the image; pooling layers, which reduce the feature map's dimension; fully connected layers, which learn the features and map them to the output layer; and an output layer (Alzubaidi et al. 2021) (Figure 3). There are several sophisticated CNN-based machine learning algorithms such as MobileNet, ResNet, DenseNet, Inception, InceptionResNet, and Xception.

BIODIVERSITAS
Volume 23, Number 5, May 2022
Pages: 2561-2569

ISSN: 1412-033X
E-ISSN: 2085-4722
DOI: 10.13057/biodiv/d230537

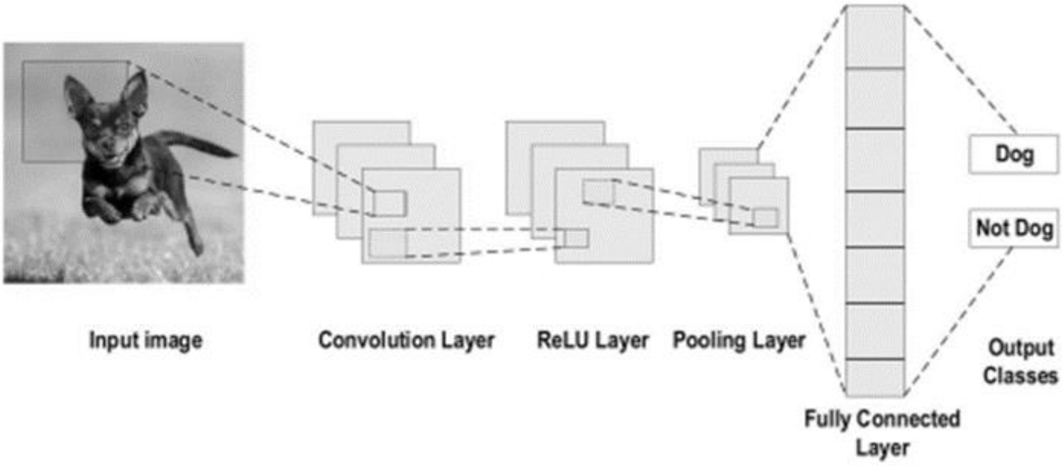

Figure 3. An example of the standard CNN

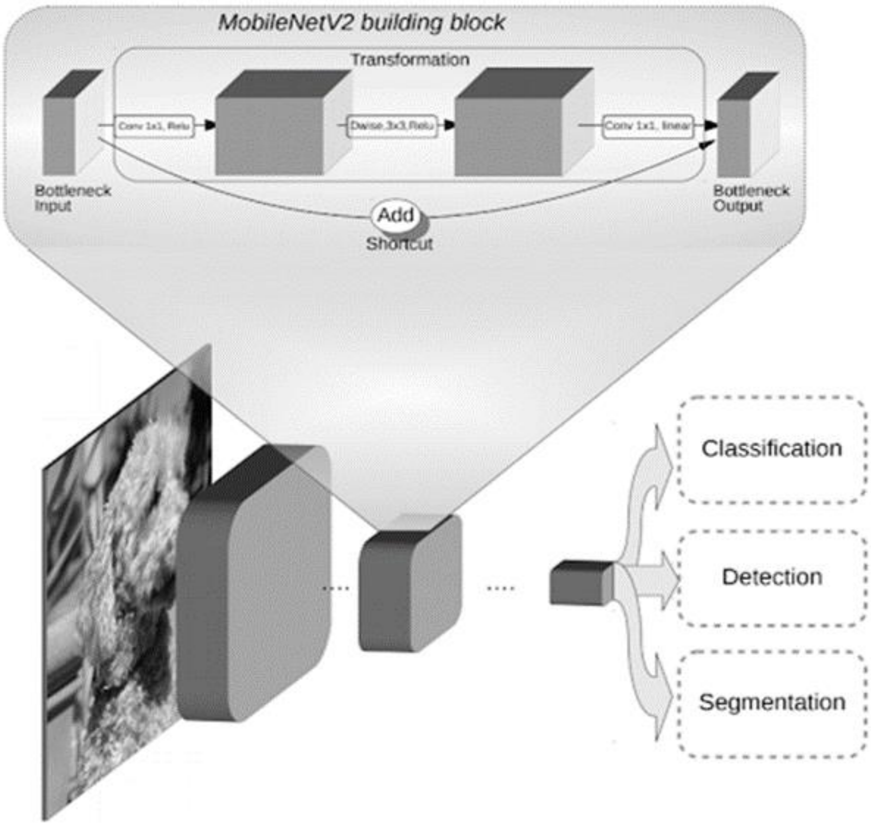

Figure 4. The architecture of MobileNet V2 (Sandler et al. 2018)

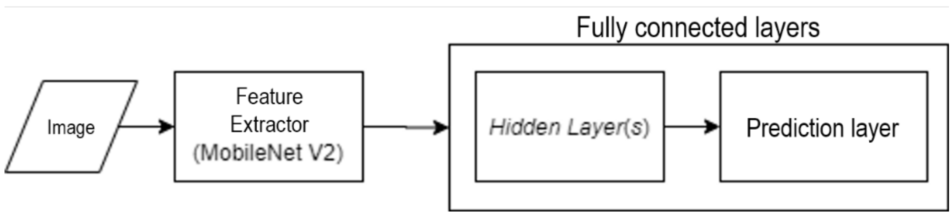

Figure 5. The proposed CNN model architecture

MobileNet V2

MobileNet V2 is the improved model of the MobileNet that focuses on reducing computational cost and information loss (Sandler et al. 2018). MobileNet V2 used linear bottlenecks and shortcuts inside the bottlenecks to achieve its goal. As a result, it claimed to be able to perform with a computational cost nine times smaller than the standard CNN. Furthermore, MobileNet V2 applies the depth-wise separable convolution inside every bottleneck and can do the classification, detection, and segmentation tasks (Figure 4). Depth-wise separable convolution consists of two types of convolutions: depth-wise convolution, which filters a single convolution per channel, and point-wise convolution, which maps the resulted features of depth-wise convolution with the linear combination (Howard et al. 2017).

Model building

This study proposes a MobileNet V2 as a feature extractor and a simple classification layer (fully connected). It consists of a hidden layer with 64 neurons to classify the images based on the output from the feature extractor, and one output layer that applies softmax function with ten neurons, in line with the focal identification system for ten species of protected birds in Indonesia (

Figure 5). The softmax function is commonly used in multiclass classification because its characteristics produce the probability of the image belonging to every class in the dataset (Nwankpa et al. 2018) (Figure 6).

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

Figure 6. Softmax function

The number of epochs for the training process was decided with the early stopping strategy. This strategy can also be used to reduce the overfitting risk and the unnecessary training epochs (Ying 2019). The number of epochs in this strategy is not explicitly set before the training but during the training process based on the stopping criterion that was set (Goodfellow et al. 2016). This strategy stores the values in a certain epoch until the model with the better metric value (validation accuracy) appears in the following *n* epochs. In this strategy, *n* is called patience (Prechelt 1998).

Model testing and evaluation

The CNN model produced from training would be tested by testing set. The testing result would be evaluated by accuracy, recall, and precision metrics that could be obtained from the confusion matrix. A confusion matrix is a table that summarizes the detailed information on the performance of a classifier in classifying the testing data (Ting 2017). There are four values in the confusion matrix: true positive (TP), true negative (TN), false positive (FP), and false-negative (FN) (Table 2. **Example of confusion matrix of multiclass classification for class K2**Table 2).

Accuracy measures the degree to which the predictions of a model match the reality being modeled (Sammut and Webb 2017). Precision refers to the proportion of predicted items as positive, was correct. Recall refers to the proportion of actual positives that were identified correctly (Salmi and Rustam 2019) (Figure 8).

The web application development

Tensorflow.js

TensorFlow.js is a library that enables the implementation and execution of machine learning models in Javascript, either on the web browser or Node.js (Smilkov et al. 2019). Furthermore, this library allows the models produced with Tensorflow on python to be implemented in Javascript by converting the saved model to the suitable format of Tensorflow.js, which Tensorflow.js also provide.

Web development using prototyping

Prototyping is a software development method that consists of five steps: communication, quick plan, modeling, quick design, prototype construction, deployment delivery, and feedback (Pressman and Maxim 2015) (

Figure **7**).

First, the communication was done by communicating with the stakeholders and resulted in a user story that contains a brief description of user needs (Pandey et al. 2018). Then, a quick plan of the web application features was done based on the user story, resulting in a use case diagram describing the system's behavior (Putra and Andriani 2019). After that, the activity diagram that defines the user's activity flow was constructed based on the user story (Wibawa 2018). Next, the web application that implements the requirements collected from the previous steps was constructed using HTML, CSS, and Javascript. Finally, the web application was deployed to the server for testing with the defined scenario (Table 3).

Table 2. Example of confusion matrix of multiclass classification for class K2

Actual	Predicted		
	K1	**K2**	**K3**
K1	**TN**	FP	**TN**
K2	FN	**TP**	FN
K3	**TN**	FP	**TN**

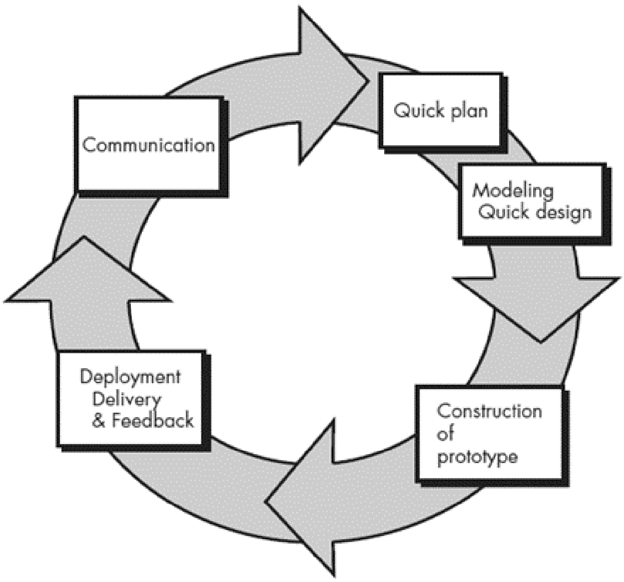

Figure 7. Flow diagram of prototyping (Pressman and Maxim 2015)

A $\text{Accuracy} = \dfrac{\text{TP+TN}}{\text{TP + FP + TN + FN}}$

B $\text{Precision} = \dfrac{TP}{TP + FP}$

C $\text{Recall} = \dfrac{TP}{TP + FN}$

Figure 8. The formula of metrics: A. Accuracy; B. Precision; C. Recall

Table 3. Black-box testing scenario

Function	Scenario
Upload, then display the uploaded image	The user clicks on the button "Pilih Gambar", then chooses the image that would be identified
Bird species identification	The user uploads an image, then clicks on the button "Identifikasi". After that, the system will display the top three most relevant species based on that image.

RESULTS AND DISCUSSION

Model building and training

The CNN model was built by adopting the convolutional basis of MobileNet V2 to be the feature extractor. The basis convolutional layers were frozen; therefore, the weights of these layers would not be trained in this study. The extracted features from these MobileNet V2 layers would get the dimension reduced in the Flatten layer before entering the classification process. The classification layer (fully connected) consists of two layers: a layer for learning the features and the output layer (Figure 9). After that, the model was compiled using the parameters of optimizer and learning rate, loss function, metric for the training process, initial epoch, and the callback parameters for the early stopping strategy (Table 4). By the early stopping strategy, the model training was done through 28 epochs, as the model could not achieve better accuracy on the validation set than the model achieved at the 18th epoch. The best accuracy for the validation set was obtained at the 18th epoch on the training with 97% accuracy (Figure 10).

Table 4. Model parameters

Parameter	Value
Optimizer	Adam
Learning rate	0.001
Epoch	100
Loss function	Categorical cross-entropy
Metric	Accuracy
Callback	Early stopping (metric: Validation accuracy, patience: 10)

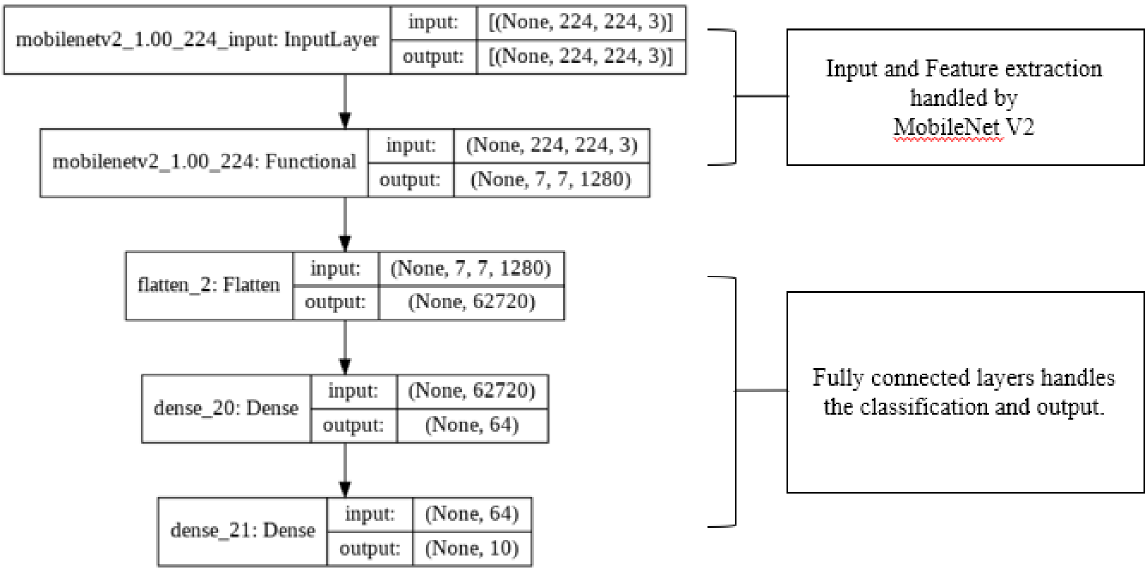

Figure 9. The CNN model architecture that implements MobileNet V2

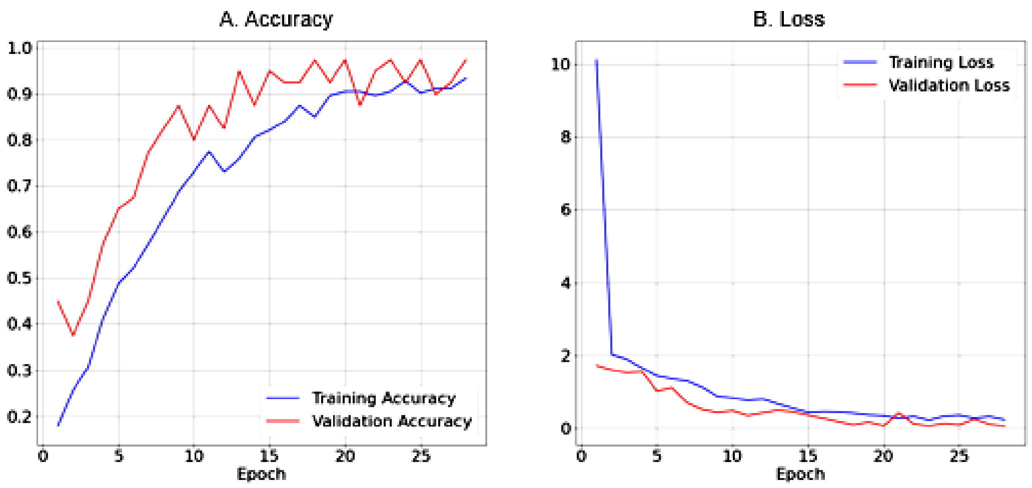

Figure 10. The plot of the metric and loss values during model training

Model testing and evaluation

The obtained model from the training process was then tested with the testing set. The testing result was evaluated with the metrics, namely accuracy, precision, and recall, that could be derived from the confusion matrix. The confusion matrix showed how many images were classified by the model, either correctly or mistakenly. Based on the testing result described by the confusion matrix, the model obtained 97% accuracy, with one false negative case at class White-tailed flycatcher and one false positive case at Blue-banded kingfisher on the testing set (Figure 11). The misclassification occurred due to its high similarity features, such as color and body shape. Then, model performance on each of the ten classes was obtained by calculating the precision and recall for each class (Table 5). The higher precision indicates the lower false-positive, the higher recall indicates the higher probability for the correct image classification in the model. As for overall performance, the model obtained 98% average precision and 97% average recall due to its misclassification of the White-tailed flycatcher as a Blue-banded kingfisher.

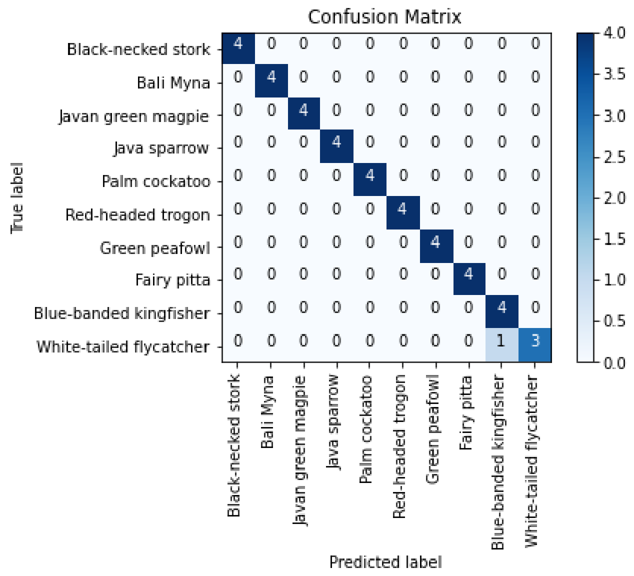

Figure 11. The confusion matrix, a summarized result of model testing

Table 5. The precision and recall scores of each class.

Species	Precision	Recall
Black-necked stork	1	1
Bali Myna	1	1
Javan green magpie	1	1
Java sparrow	1	1
Palm cockatoo	1	1
Red-headed trogon	1	1
Green peafowl	1	1
Fairy pitta	1	1
Blue-banded kingfisher	0.8	1
White-tailed flycatcher	1	0.75

The web application development

The first step of the Prototyping method is communication, which was done through the online discussion with the stakeholders to obtain the main requirements. This step produced a user story that contained actors and tasks that would be involved in this system (Table 6). A use case diagram was designed in the quick plan (Figure 12). This use case diagram not only consists of one actor (Observer) and two tasks as described in the user story but also describes the dependency of the tasks. For example, the task "Checking the bird species description" can be done when the observer has finished the task "Upload the bird image which will be identified."

The input image is expected to have a focus on birds. The quality of the image itself will affect the classification result, such as when the bird object is not clearly visible, have much noise, or the bird object is overlapped with another object.

The development then continues to model quick design, which builds an activity diagram that models the flow and the system's features based on the defined use cases in the use case diagram. In this system, the flow starts when the observer clicks on the "Pilih Gambar" button, then chooses an image containing a bird to be identified. After that, the web application would display the uploaded image. The flow continues when the observer clicks on the button "Identifikasi", the image is received by the model to run the identification. The result will be shown on the webpage. If the observer wants to check the species' description, they will click the link "Klik untuk lihat deskripsi" to get redirected to the description page. Otherwise, the flow had been stopped when the result was displayed on the webpage (Figure 13).

Table 6. The user story produced from the communication stage
.

Actor	User story	Task
Observer	As an observer, I want to identify the species of found protected bird.	Uploads an image of the bird to the system
	As the observer, I want to read a short description of the result of the bird species identification.	Checking the description of the identification result.

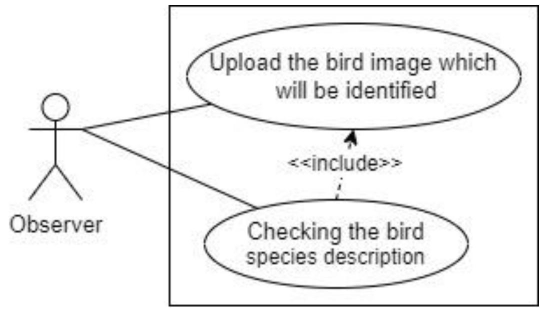

Figure 12. The use case diagram

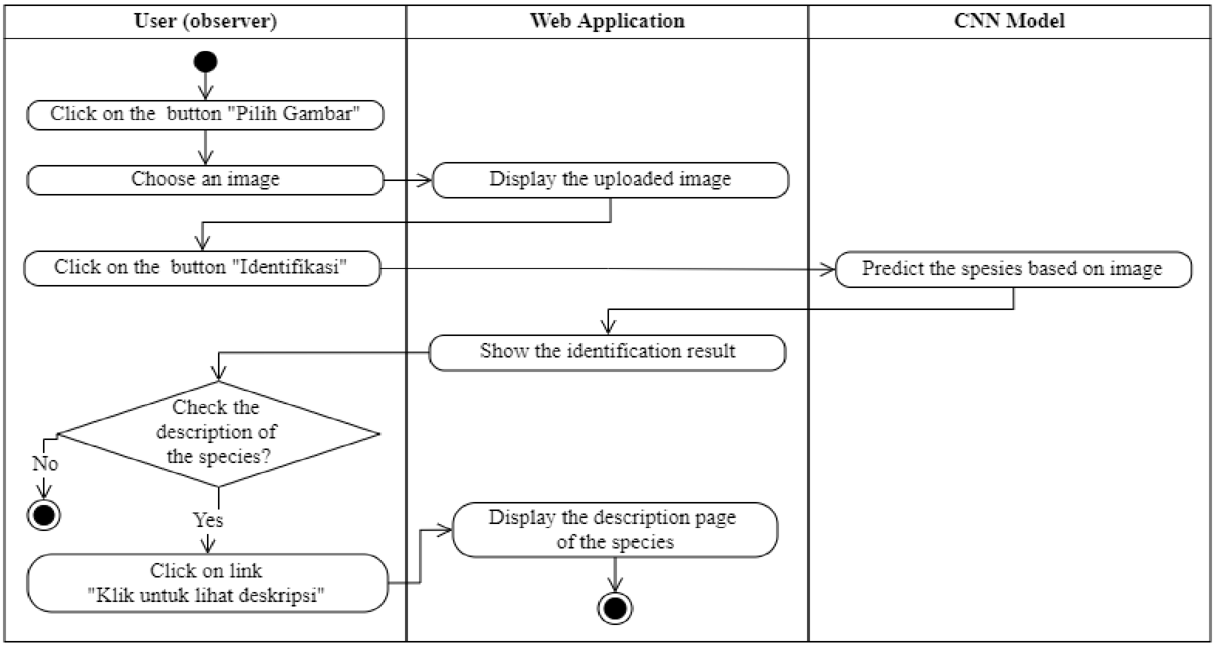

Figure 13. The activity diagram

The next step in the Prototyping method is the construction of the prototype. The web application was constructed by applying all things obtained from the previous steps. As the model was built and trained with TensorFlow in python, it was required to convert its saved model to a compatible format on Javascript, which was done by utilizing the script provided by TensorFlow. After the model got converted, the construction of the web application was done using HTML, CSS to set the layout and appearance of the webpage, and Javascript to do the business process and the model implementation for the web application (Figure 14). Finally, the web application was deployed to the server to undergo the black-box testing with the predefined scenario. A test case would be regarded as a success if there is no problem and the output from the system would be the same as expected in the scenario during the testing. The testing result showed that all functions on the web application were executed as the expected scenario. Thus, it can be said that the overall web application passed the testing (

Table 7).

This study shows the process of building a classification model using CNN to identify birds categorized in regulation No. 106/2018. Based on the model testing and evaluation results, it can be concluded that the MobileNet V2 architecture can provide an efficient model that can perform with 97% accuracy. As the model can be implemented in the web-based application, it allows the system's portability as one connects to the internet and can be accessed everywhere. For future works, increasing the number and variety, such as age and gender, in the dataset possibly enhance the accuracy. Implementing a progressive web application and portability of mobile applications, which can be used anywhere and anytime, can aid field officers in combating illegal wildlife trading and have more impact on wildlife conservation in Indonesia.

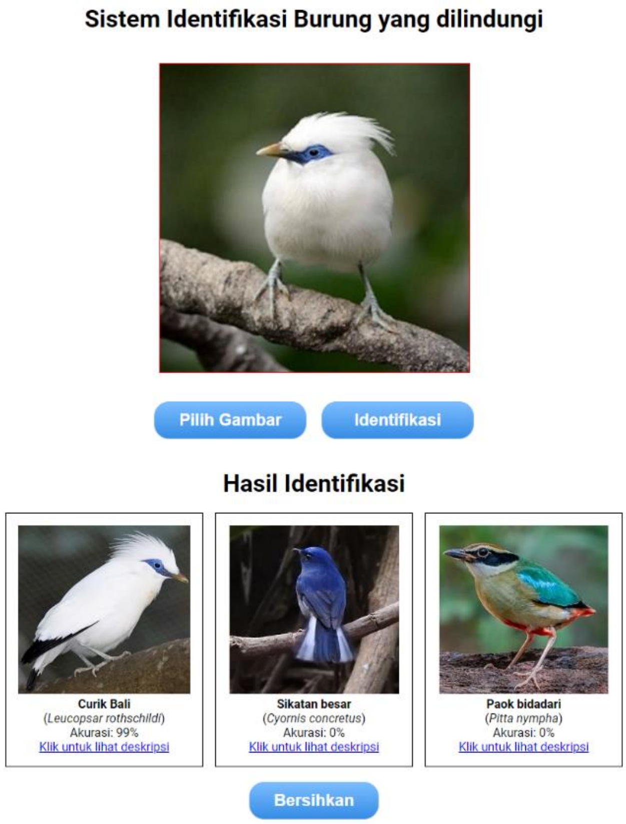

Figure 14. The appearance of the web application

Table 7. The result of the black-box testing

Function	Scenario	Result
Upload, then display the uploaded image	The user clicks on the button "Pilih Gambar", then chooses the image that would be identified.	Success
Bird species identification	The user uploads an image, then clicks on the button "Identifikasi". After that, the system will display the top three most relevant species based on that image.	Success

ACKNOWLEDGEMENTS

The authors' gratitude goes to the Department of Computer Science, Institut Pertanian Bogor, Bogor, Indonesia, for funding this research and attending the WECMIC 2021 Conference.

REFERENCES

Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, Santamaría J, Fadhel MA, Al-Amidie M, Farhan L. 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. J Big Data 8: 53. DOI: 10.1186/s40537-021-00444-8.

Biswas AA, Rahman Md. M, Rajbongshi A, Majumder A. 2021. Recognition of Local Birds using Different CNN Architectures with Transfer Learning. In: International Conference on Computer Communication and Informatics (ICCCI), 27-29 January 2021, Coimbatore, India. DOI: 10.1109/ICCCI50826.2021.9402686.

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

Harjoseputro Y, Yuda IP, Danukusumo KP. 2020. MobileNets: Efficient convolutional neural network for identification of protected birds. Intl J Adv Sci Eng Inf Technol 6: 2290-2296. DOI: 10.18517/ijaseit.10.6.10948.

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H. 2017. MobileNets: Efficient convolutional neural networks for mobile vision applications. Computing Research Repository (CoRR). arXiv. DOI: 10.48550/arXiv.1704.04861

Huang YP, Basanta H. 2019. Bird image retrieval and recognition using a deep learning platform. IEEE Access 7: 66980-66989. DOI: 10.1109/ACCESS.2019.2918274.

Nwankpa C, Ijomah W, Gachagan A, Marshall S. 2018. Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. Computing Research Repository (CoRR). DOI: 10.48550/arXiv.1811.03378

Pandey H, Kumar S, Singh VK, Ahmad W. 2018. Comparison of usage of crowdsourcing in traditional and agile software development methodologies on the basis of effectiveness. Intl J Comput Eng Technol 9: 271-284.

Prechelt L. 1998. Automatic early stopping using cross validation: Quantifying the criteria. Neural Netw 11: 761-767. DOI: 10.1016/S0893-6080(98)00010-0.

Pressman RS, Maxim BR. 2015. Software Engineering: A Practitioner's Approach-8th ed. McGraw-Hill, New York.

Putra DWT, Andriani R. 2019. Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD. J Teknoif 7: 32-39. DOI: 10.21063/jtif.2019.V7.1.32-39.

Salmi N, Rustam Z. 2019. Naïve bayes classifier models for predicting the colon cancer. IOP Conf Ser Mater Sci Eng 546: 052068. DOI: 10.1088/1757-899X/546/5/052068.

Sammut C, Webb GI. 2017. Accuracy. In: Sammut C, Webb GI (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA. DOI: 10.1007/978-1-4899-7687-1_3.

Sandler M, Howard A, Zhu ML, Zhmoginov A, Chen LC. 2018. MobileNetV2: Inverted Residuals and Linear bottlenecks. Computing Research Repository (CoRR). arXiv: DOI: 10.1109/CVPR.2018.00474.

Smilkov D, Thorat N, Assogba Y, Yuan A, Kreeger N, Yu P, Zhang K, Cai S, Nielsen E, Soergel D, Bileschi S, Terry M, Nicholson C, Gupta SN, Sirajuddin S, Sculley D, Monga R, Corrado G, Viégas FB, Wattenberg M. 2019. TensorFlow.js: Machine Learning for the Web and Beyond. Computing Research Repository (CoRR). arXiv: DOI: 10.48550/arxiv.1901.05350

Ting KM. 2017. Confusion matrix. In: Sammut C, Webb GI (eds). Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA. DOI: 10.1007/978-1-4899-7687-1_50.

Wibawa DY. 2018. Development of Front-End Module for E-Commerce in Digital Tani Mobile Application [Thesis]. Institut Pertanian Bogor, Bogor. [Indonesian].

World Bank Group. 2018. Tools and Resources to Combat Illegal Wildlife Trade. World Bank, Washington DC (WA). Retrieved from https://openknowledge.worldbank.org/handle/10986/29542, accessed on 16.03.2022

Yamashita R, Nishio M, Do RKG, Togashi K. 2018. Convolutional neural networks: An overview and application in radiology. Insights into Imaging 9: 611-629. DOI: 10.1007/s13244-018-0639-9.

Ying X. 2019. An Overview of Overfitting and its Solutions. J Phys Conf Ser 1168 (2): 022022. DOI: 10.1088/1742-6596/1168/2/022022.

Zhang A, Lipton ZC, Li M, Smola AJ. 2020. Dive into Deep Learning. https://d2l.ai