**PAPER** • **OPEN ACCESS**

# Automating bird species classification: A deep learning approach with CNNs

To cite this article: Renjun Cai 2023 *J. Phys.: Conf. Ser.* **2664** 012007



Journal of Physics: Conference Series **2664** (2023) 012007 doi:10.1088/1742-6596/2664/1/012007

# Automating bird species classification: A deep learning approach with CNNs

**Renjun Cai**

Nanjing Forestry University, Nanjing, China

renjuncai@njfu.edu.cn

**Abstract.** Bird species identification and classification are challenging yet crucial for research and conservation. Traditional methods are labor-intensive, require specialized expertise, and can be prone to error. Recent advances in deep learning offer an automated solution to this complex problem. This study evaluates a convolutional neural network (CNN) model for classifying images of 525 bird species. The model employs transfer learning using the EfficientNetB0 architecture and was trained on over 84,000 images. With data augmentation, the model achieved 87% validation accuracy and 86.7% test accuracy, demonstrating its ability to overcome limited data and generalize well. Difficulties in obtaining balanced, high-quality data for each species were addressed through transfer learning and augmentation.

**Keywords:** convolutional neural network, bird identification, transfer learning.

## 1. Introduction

Birds are an integral part of ecosystems worldwide, serving important roles as pollinators, seed dispersers, and predators of insects and small mammals. However, many bird populations have declined dramatically due to habitat loss, climate change, and other human activities. Accurate identification and monitoring of bird species is crucial to understanding population changes and implementing effective conservation strategies [1].

Traditional methods of bird identification, such as field observation and specimen examination, require extensive training and can be challenging to scale. It relies on detecting subtle differences in color, pattern, song, and behavior to distinguish between species, which poses difficulties even for experts. As a result, misidentification rates can be high, especially for inexperienced individuals. Therefore, computer vision and deep learning offer an automated solution for identifying and classifying birds more accurately and consistently [2].

Since Convolutional Neural Network (CNN) has achieved remarkable success in image recognition tasks by learning hierarchical feature representations, this research adopts a CNN model, specifically EfficientNetB0, to address the challenging task of multi-class bird image classification and classify images of 525 different species of birds [3-4]. While previous studies have applied machine learning to identify birds, they were limited to a small number of species. Transfer learning and data augmentation techniques have been employed to train the model using a dataset of over 84,000 bird images.

The paper provides a thorough account of the creation and assessment of a CNN model for the classification of multiclass bird images with a description of the model architecture, training process, and evaluation metrics, including accuracy, loss, classification reports, and Gradient-weighted Class Activation Mapping (Grad-CAM) visualizations [5-6].

Research methods used for this study include the collection and preprocessing of the image dataset, the implementation of the EfficientNetB0 architecture with additional dense layers, and the training of the model. The model's ability to classify both seen (validation set) and unseen (test set) bird images has been evaluated to demonstrate its capability for generalized use. A range of visualization techniques, such as accuracy and loss curves, test set predictions, a classification report, and Grad-CAM heatmaps, have been employed to analyze the model's performance, predictions, tendencies, and potential shortcomings.

This research is a notable example of the growing potential of deep learning in the realm of conservation research and environmental applications. Automatic classification of species offers significant advantages, including increasing the scale of available datasets and enabling new research questions. These results promote advancements in conservation policies and actions [7-9].

## 2. EfficientNetB0

EfficientNetB0 is a CNN architecture designed to achieve high accuracy and efficiency in image classification tasks. Tan and Le proposed a novel compound scaling method to scale up CNNs in a principled way [4]. The compound scaling method balances the network width, depth, and resolution, and uses a coefficient to control the available resources for model scaling. EfficientNetB0 is the baseline network that was obtained by searching for neural architecture, which can be scaled up to obtain other variants of EfficientNet, such as B1 to B7, using different coefficients.

EfficientNetB0 exhibits several advantages over extant CNN architectures. Firstly, it attains state-of-the-art accuracy on ImageNet and other transfer learning datasets, while utilizing an order of magnitude fewer parameters and FLOPS (floating-point operations per second). In addition, it is platform-aware and has the adaptability to modify coefficients for different hardware restrictions. Furthermore, EfficientNetB0's main building blocks consist of mobile inverted bottleneck blocks with squeeze-and-excitation optimization, which are highly efficient and efficacious for feature extraction. The inclusion of input preprocessing as part of the model is another advantage, simplifying the data pipeline and mitigating the risk of data mismatch [10].

EfficientNetB0 can be implemented using various deep learning frameworks, such as TensorFlow, Keras, and PyTorch. It can also be loaded with weights pre-trained on ImageNet or other datasets, which can improve the performance of downstream tasks.

## 3. Experiments

### *3.1. Data source*

The dataset utilized in this study was sourced from Kaggle and contains 525 bird species, consisting of 84,635 training images, 2,625 test images (with five images for each species), and 2,625 validation images (also five images for each species). The images were sourced from internet searches based on the name of the species. Additionally, the dataset is organized into three subdirectories, each containing 525 directories that represent each bird species.

Although the training set is imbalanced and the numbers of files per species is different, it is guaranteed to have at least 130 training image files per species. However, one significant shortcoming of the dataset is the unequal ratio between the images of male and female species. Approximately 80% of the images are of males, which tend to be more diverse in color, while females are often more monochromatic. Therefore, since the dataset is dominated by images of male species, the classifier may not perform as well on female species images.

The dataset comprises This dataset is characterized by the presence of only one bird in each image, occupying at least 50% of the pixels. As a result, a moderately complex model can achieve training

and test precisions in the mid-range 90%. All images are in jpg file format, with dimensions of 224 x 224 x 3 [11]. This approach ensures that the processed images contain enough information to develop a highly accurate classifier. Even a moderately robust model will achieve training, validation, and test accuracies in the high 90% range. Each file in the dataset is numbered sequentially, beginning with one for each species, and the test and validation images are named 1.jpg to 5.jpg. The training images are also numbered sequentially with "zeros" padding to ensure file order preservation when used in Python file functions and Keras flow from directories. The preview of the data set is shown in Figure 1.


**Figure 1.** A preview of birds in dataset.

We use Python and Keras framework via NVIDIA RTX3080 on Ubuntu20 to implement the CNN model. The model images shall undergo evaluation by a pre-trained CNN model known as EfficientNetB0. The training process will be monitored using three callbacks: Model Checkpoint, Early Stopping, and Tensorboard callback. The model hyperparameters are summarized below: Batch size of 32, 100 epochs, input shape of (224, 224, 3), and an output layer with 525 classes.

### *3.2. Data preprocessing*

The dataset will be divided into three distinct categories: Training, Validation, and Testing. The training data will be utilized for training the deep learning CNN model, and its parameters will be fine-tuned using the validation data. The model's performance will be evaluated using the Testing data, which represents new, previously unseen data for the model.

Initially, the image dataframe is divided into training and testing sets using sklearn's train test split function. Then, Two ImageDataGenerator objects are defined for training and testing datasets. These generator objects are responsible for loading image data into memory in batches during the CNN model's training. The preprocess input method of the EfficientNet application is applied to the images as they are loaded into the network to ensure that the input images undergo the same preprocessing as the images used to train the EfficientNet model.

The training dataset is divided into two subsets: training and validation, using the flow from dataframe method of the ImageDataGenerator object created earlier. This method reads batches of images from disk and applies data augmentation techniques defined in the generator to modify the original data and increase the dataset's size and variability. The validation data subset assesses the model's performance during training and facilitates selection of the best-performing model based on validation accuracy. The quantity of data across the three datasets is depicted in Table 1.

**Table 1.** The output observed upon inspecting the data.

| Categerires | Quantity of Images | Classes |
|---|---|---|
| training | 54167 | 525 |
| validation | 13541 | 5252 |
| testing | 16927 | 525 |

Finally, we define a sequential model comprising various layers of data augmentation, such as resizing, rescaling, flipping, rotation, zooming, and contrast adjustment. These techniques are applied to the images during training, thereby slightly modifying the input data at each epoch, which enhances the model's generalization.

### *3.3. Training*

The images of the model will be evaluated using the pre-trained CNN model, EfficientNetB0. During the training process, three callbacks will be employed to monitor progress, namely: Model Checkpoint, Early Stopping, and TensorBoard.

*3.3.1. Model initialization.* Initially, we instantiate a pre-trained CNN model, EfficientNetB0, and configure several of its parameters. Specifically, we set its input shape to (224, 224, 3), exclude the final output layer, apply max pooling, and make the model non-trainable by setting its trainable properties to False. This ensures the CNN model's weights remain unaltered during training.

*3.3.2. Training callbacks.* We construct multiple callbacks to employ during the training process. The Model Checkpoint callback stores the optimal model weights during training, allowing us to either resume training from optimal weights or utilize the optimal weights for inference. The EarlyStopping callback prematurely terminates training if the validation loss does not improve over five consecutive epochs. The Reduce LROnPlateau callback decreases the learning rate if the validation loss plateaus for three consecutive epochs.

*3.3.3. Deep learning model construction.* A deep learning model is constructed by adding several layers on top of the pre-trained CNN model. The inputs variable denotes the model's input tensor, which is passed through an augmentation layer, and finally through a series of dense and dropout layers for over fitting prevention. The model concludes after passing through the output layer, which has a softmax activation for multi-classification. The model is compiled utilizing the Adam optimizer, and the categorical cross-entropy is selected as the loss function and accuracy as the metric. Using the previously defined callbacks, the model is trained for 150 epochs, using train images and val images for the training and validation sets, respectively.

### *3.4. Result*

*3.4.1. Training and validation curves.* Figure 2 (a) and (b) illustrate the loss and accuracy curves for the training and validation sets. The model achieves an accuracy of approximately 87% in the validation dataset. Notably, the validation accuracy consistently outperforms the training accuracy during the model training phase. The results demonstrate that the model constructed does not exhibit over fitting to the training set.


**Figure 2.** (a)Loss curves for the training and validation sets.; (b) accuracy curves for the training and validation sets.

*3.4.2. Predictions.* In order to evaluate the performance of a machine learning model, predicting the labels of the test data is an essential step. This process is crucial in determining the model's ability to generalize to new and unseen data. Accurately predicting the labels of the test data demonstrates the model's ability to distinguish between different classes effectively. A high level of accuracy in labeling test data is an indication of a good machine learning model.

To visually present the prediction outcomes, a series of 25 images from the dataset are selected and accompanied by their respective labels (Figure 3). We accomplish this by making use of the random index function, which randomly selects 15 images from the test dataset.


**Figure 3.** 25 random pictures of prediction.

We then plot these images using the 'plt. subplots' function, with 'subplot kw' specifying the x and y ticks. A 'for' loop iterates over all the images, plotting each and displaying relevant information such as true and predicted labels. If the true and predicted labels match, they are shown in green; otherwise, they are colored red.

*3.4.3. Classification reports.* The simplified representation of the classification reports can be seen in Table 2. The model has an overall accuracy of 0.86767, which is decent but not exceptional. The model performs well for some classes, achieving high precision, recall, and F1-score values, while performing poorly for others. For instance, the model has perfect performance for the African crowned crane class, but only achieves a F1-score of 0.58182 for the Abbotts booby class. The macro and weighted averages of precision, recall, and F1-score are also lower than the accuracy, indicating that the model struggles with some classes. Therefore, while the model shows promising results, it still requires further refinement to improve its capabilities for classifying certain bird species.

**Table 2.** The classification reports of the model.

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| ABBOTTS BABBLER | 0.72222 | 0.76471 | 0.74286 | 34.00000 |
| ABBOTTS BOOBY | 0.80000 | 0.45714 | 0.58182 | 35.00000 |
| ABYSSINIAN GROUND HORNBILL | 0.93548 | 0.85294 | 0.89231 | 34.00000 |
| AFRICAN CROWNED CRANE | 1.00000 | 1.00000 | 1.00000 | 26.00000 |
| AFRICAN EMERALD CUCKOO | 0.89655 | 0.72222 | 0.80000 | 36.00000 |
| ... | ... | ... | ... | ... |
| YELLOW HEADED BLACKBIRD | 0.90000 | 1.00000 | 0.94737 | 27.00000 |
| ZEBRA DOVE | 0.96774 | 0.96774 | 0.96774 | 31.00000 |
| Accuracy | 0.86767 | 0.86767 | 0.86767 | 0.86767 |
| Macro Avg | 0.87070 | 0.86932 | 0.86597 | 16927.00000 |
| Weighted Avg | 0.87315 | 0.86767 | 0.86636 | 16927.00000 |

*3.4.4. Grad-CamVisualization.* By using Grad-CAM, one can discern the specific areas in the image that captured the model's attention while making the prediction. The modified version of CAM, Grad-CAM, can be applied to any model that uses a CNN as its underlying architecture [11].

We express two functions designed to produce Grad-CAM visualizations for CNNs, as well as some initialization code. The first function is responsible for loading an image file from a user-specified file path and resizing it to a specified size. The second function generates a heatmap visualization that highlights the most crucial image regions for predicting a particular output class. This involves passing in an image array, a trained CNN model, and the name of a specific convolutional layer within the model. The third function saves the heatmap image to a file and then displays the heatmap overlaid on the original image.

All of these functions applied to a series of test images that have been processed by the CNN model, which is shown in Figure 4. Each image has a corresponding heatmap overlaid on top of it that highlights the regions responsible for the CNN model's classification decision. The objective is to offer a visual illustration of the decision-making procedure of the CNN. This will enable the identification of any potential biases or errors in its output. The layout of the visualizations consists of a set of original images, each with a corresponding heatmap.

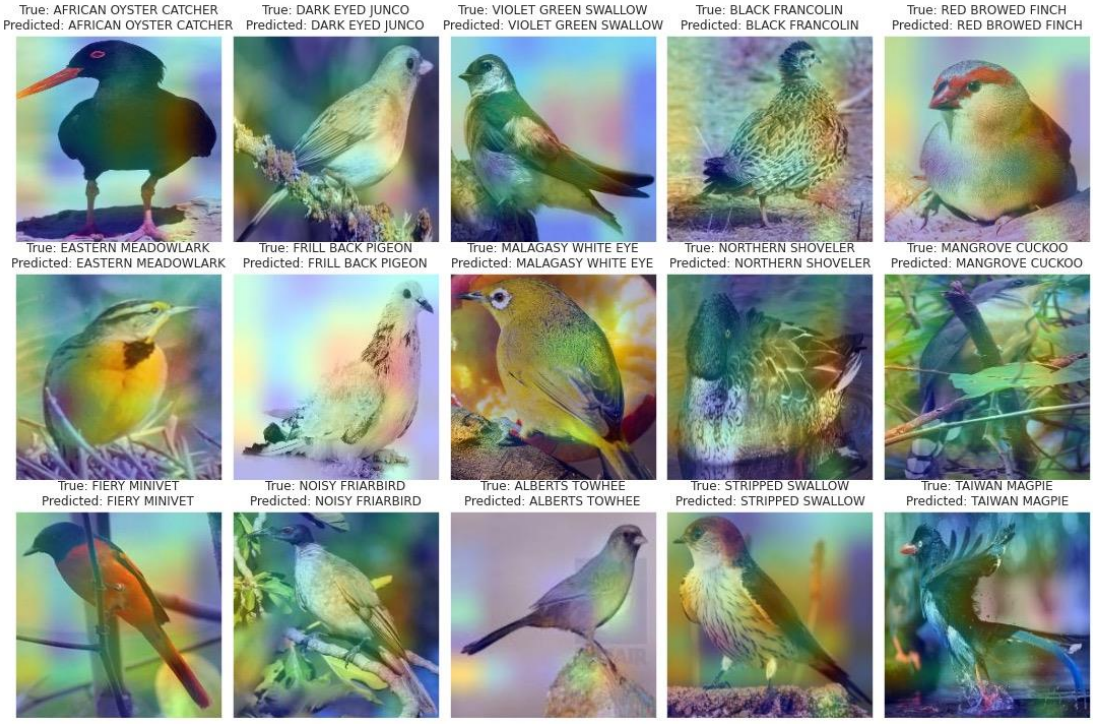


**Figure 4.** Test images with heatmaps.

## 4. Conclusion

This study presents the development and evaluation of a CNN model to classify images from 525 different bird species. The model integrates the pre-trained EfficientNetB0 architecture, and it underwent training on more than 84,000 images. Leveraging transfer learning and data augmentation, the model achieved high accuracy using a modest number of training examples per class. The training and validation accuracy and loss curves demonstrate that the model has learned generalizable features for bird species classification and does not display signs of overfitting. The validation set yielded an accuracy of 87%, while the test set produced 86.7% accuracy, highlighting the model's ability to generalize well to new data. Nonetheless, there is room for further improvement to enhance accuracy.

Future work could focus on several potential areas for improvement, including the collection of additional data, well-balanced datasets for underrepresented species and females,etc. Incorporating additional data augmentation techniques such as cutout, mixup, and generative adversarial networks to generate synthetic training data could be valuable in combating overfitting on species with little data.

Additionally, testing the model on new bird species not present in the original dataset would highlight gaps in the model and inform future investigations. Using the model's convolutional filters, feature maps, and Grad-CAM heatmaps could provide a deeper understanding of its predictions and potential weaknesses. Further analysis could provide insights for model optimization, improving its performance, and creating more explicable predictions. These advancements highlight the growing promise of deep learning models in accurately identifying and classifying birds to promote field studies and conservation efforts.

## References

[1] W¨aldchen J, M¨ader P. Machine learning for image based species identification[J]. Methods in Ecology and Evolution, 2018, 9(11): 2216-2225.

[2] Stowell D, Wood M D, Pamu ¯ la H, et al. Automatic acoustic detection of birds through deep learning: the first bird audio detection challenge[J]. Methods in Ecology and Evolution, 2019, 10(3): 368-380.

[3] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern recognition, 2018, 77: 354-377.
[4] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning. PMLR, 2019: 6105-6114.
[5] Janocha K, Czarnecki W M. On loss functions for deep neural networks in classification[J]. arXiv preprint arXiv:1702.05659, 2017.
[6] Selvaraju R R, Cogswell M, Das A, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization[C]//Proceedings of the IEEE international conference on computer vision. 2017: 618-626.
[7] O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.
[8] Wu J. Introduction to convolutional neural networks[J]. National Key Lab for Novel Software Technology. Nanjing University. China, 2017, 5(23): 495.
[9] O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.
[10] Lei X, Pan H, Huang X. A dilated CNN model for image classification[J]. IEEE Access, 2019, 7: 124087-124095.
[11] BIRDS 525 SPECIES- IMAGE CLASSIFICATION, https://www.kaggle.com/datasets/gpiosenka/100-bird-species. Last accessed May 19, 2023