Hindawi
Journal of Engineering
Volume 2024, Article ID 8254130, 13 pages
https://doi.org/10.1155/2024/8254130

Research Article

A New Efficient Classifier for Bird Classification Based on Transfer Learning

Lesia Mochurad and Stanislav Svystovych

Lviv Polytechnic National University, Lviv 79013, Ukraine

Correspondence should be addressed to Lesia Mochurad; lesia.i.mochurad@lpnu.ua

Received 29 November 2023; Revised 12 January 2024; Accepted 29 January 2024; Published 7 February 2024

Academic Editor: Kamran Iqbal

The problem of classification of different species of birds in the images is relevant in the modern world. The essence is to automatically determine the species of birds depicted in the photo using artificial intelligence. It is important for several areas of human life, in particular for nature conservation, environmental research, education, and ecotourism. The development of a classifier, based on deep learning methods, can help to effectively solve the problem of classifying different species of birds, ensuring high accuracy. The work developed a new effective algorithm for the classification of different species of birds in the images. An optimal model architecture was built using the transfer learning approach. A dataset of 525 bird species was analyzed and preprocessed in detail. The model training process was carried out, which includes two phases: only the upper layers are deactivated and the last 92 layers of the pretrained EfficientNetB5 model (not including BatchNormalization layers) and the top layers are activated. As a result, efficacy indicators were obtained: accuracy = 98.86%, precision = 0.99, recall = 0.99, and $F1$ score = 0.99, which showed improvement in comparison with modern research. The developed classifier is best suited for such areas of human life as education and ecotourism because for them the number of different species of birds that the classifier can determine is very important.

1. Introduction

Classification of different bird species in images is important for different areas [1, 2]:

(1) Environmental studies: Identifying bird species in images can assist in monitoring and analyzing birds for ecosystem research, determining changes in populations and studying the effects of climate change and other factors on birds.

(2) Conservation and biodiversity study: Recognition of bird species in images can serve to study and protect bird diversity by helping to identify different species and their migratory paths.

(3) Agricultural research: Automatic bird classification can be used to identify species that may affect agriculture. For example, the allocation of species that can harm agricultural crops or participate in natural processes that are useful for agriculture.

(4) Studying bird behavior: Image analysis can help study the behavior of different bird species, such as their migratory pathways, nesting, feeding, and other aspects of behavior.

(5) Biological research: The classification of birds in images may be important for biological research aimed at understanding the evolutionary and genetic aspects of different bird species.

(6) Environmental pollution monitoring: Changes in bird populations can serve as indicators of environmental pollution. Classification of birds in images can help in assessing the impact of pollutants on the bird world.

The use of deep learning and computer vision methods for automatic classification of birds makes it much easier to process a large amount of data and provide fast and accurate analysis [3, 4].

As it is known [5], classification is a type of problem in the area of artificial intelligence, the essence of which is to assign each object a certain class based on its characteristics. The main purpose of classification is to teach a model to recognize regularities or patterns in data and determine which class a new, previously unknown object belongs to.

The task of classifying different species of birds in images belongs to the area of computer vision.

Computer vision is a field of science and technology that studies and develops systems that provide computers with the ability to "see" and interpret images and videos in the same way that human vision does [6, 7]. The main purpose of computer vision is to give computers the ability to recognize objects, determine their characteristics, and interact with the environment based on visual data.

Problems related to this area are best solved by applying deep learning methods. Deep learning is a sub-branch of machine learning that uses neural networks with a significant number of layers (deep architectures) to automatically identify and research high-level data representations. Deep learning becomes especially powerful when using neural networks with many layers, as it allows models to automatically identify complex dependencies and abstractions in the input data [8].

Eventually, we analyzed the current state of the problem studied in our research work on the classification of different species of birds in the images. At the same time, the main advantages and disadvantages of each of the considered works are highlighted. As a result, in Section 3, we will provide a table of comparison of the best values of the performance indicators of existing models on the test data and the one proposed by us. So, the preprocessing of images, namely, increasing their resolution is described in detail in [9].

The authors introduced a investigate framework in [10] that explores the classification of bird species by integrating deep neural features from visual and audio data through the kernel-based fusion technique. Specifically, the deep neural features are derived from the activation values of the inner layer of the convolutional neural network (CNN). The authors employed multicore learning (MKL) to fuse these features for the ultimate classification. In experimental trials, the proposed CNN + MKL method, which incorporates both types of data, demonstrates superior performance compared to single-modal approaches, certain basic kernel combination methods, and the traditional early fusion approach.

A method for identifying birds based on visual features using convolutional neural networks was proposed by the authors in [11]. Two methods are proposed in this paper. The first is an attention-driven data enhancement. And the second method is a compression model for distilling disjointed knowledge. As a result, a fine-grained bird classification model was created and an accuracy of 87.63% was achieved.

In [12], the authors compared three approaches to solve the problem of classifying different bird species in images. The first approach is the use of a traditional machine learning algorithm (SVM), the second is the use of a deep learning algorithm (ResNet50 model), and the third is the use of a deep learning algorithm in combination with transfer learning (ResNet50 trained model). As a result, it was concluded that the classifier based on deep learning in combination with transfer learning achieved the best results. This experiment completely proved the ineffectiveness of using traditional machine learning methods to classify different species of birds in images. It was also shown that the use of deep learning methods in combination with transfer learning gives a significant advantage to deep learning in the context of solving this problem.

In [13], the authors investigated the use of convolutional neural networks to classify different bird species in images. Four models with different architectures based on transfer learning were used to classify images: ResNet152V2, Inception V3, DenseNet201, and MobileNetV2. The models were trained on the BIRDS 400 SPECIES dataset, containing about 50 thousand images of 400 different species of birds. As a result, models of ResNet152V2 and DenseNet201 had the best results. For ResNet152V2, accuracy = 95.45% and loss (categorical cross-entropy) = 0.8835. At the same time, DenseNet201 resulted worse accuracy = 95.05% but better loss = 0.6854.

In the article [14], the author researched the application of deep learning methods to classify different species of birds in images. For these researches, the BIRDS 400 SPECIES dataset was selected. There were relatively many different deep learning-based models, both simple and complex, trained on the selected dataset. As experiments have shown, the best solution to this problem was done by a complex of models based on transfer learning. In addition, the technique of image augmentation was applied, which had a very positive impact on the final effectiveness of the models. In conclusion, it was determined that the best performance was shown by the pretrained VGG19 model (the first 17 layers were frozen), which was trained on augmented training data. It was able to achieve the best results: loss (categorical cross-entropy) = 0.1426.

In [15], the authors applied deep learning techniques combined with transfer learning to solve the problem of classifying different bird species in images. A number of models were created, previously trained on the ImageNet dataset, based on the following deep architectures: EfficientNetB0, DenseNet201, MobileNetV2, MobileNet, ResNet152V2, VGG16, and VGG19. Next, these models were customized to perform the task of classifying different bird species in images using the BIRDS 400 SPECIES dataset. Experiments have shown that the best solution to this problem coped model EfficientNetB0. After that, all models of the EfficientNet (B0–B7) family were compared in detail. As a result, it was concluded that the EfficientNetB0 model showed the best results on the test data: accuracy = 98.60%.

The research paper [16] conducts an extensive examination of bird detection and species classification, employing the YOLOv5 object detection algorithm and the

EfficientNetB3 deep learning model with retraining. The dataset utilized by the authors corresponds to the one employed in our study. Consequently, in Section 3, we will perform a comparative analysis of the achieved outcomes.

The aim of this work is to develop a new classifier using deep learning methods that would allow for high accuracy and effectively classify as many different bird species as possible in images.

The novelty of this paper is summarized as follows:

(1) A dataset including a large number of bird species was analyzed and preprocessed in detail.

(2) The optimal architecture of the model is proposed, using the approach of transfer learning.

(3) In addition, a new efficient algorithm has been developed to classify different bird species in images based on deep learning.

(4) Through large-scale testing, it was established that on the basis of the proposed algorithm, it was possible to significantly increase the performance indicators of the model: loss, accuracy, precision, recall, and F score.

The rest of the text is built according to the following structure: In Section 2 data preprocessing is described, the optimal architecture of the classification model based on deep learning is built, and model training is discussed in detail. In Section 3, the model training process is carried out based on two phases, and the test results are given. The last section contains conclusions and prospects for further research.

2. Materials and Methods

2.1. Dataset. To train the model in this research work, the BIRDS 525 SPECIES dataset [17] was chosen, which contains about 90 thousand images with 525 different species of birds. Each image is color, has a size of 224×224 pixels, and is stored in jpg format. It is a very high-quality dataset where there is only one bird in each image, and it usually occupies at least 50% of its pixels. Likewise, it should be noted that all images are original and not created by applying augmentation techniques.

The dataset is predivided into three samples: training, validation, and test. The training set contains 84,635 images (94%), while the validation and test sets each contain 2,625 images (3%).

Next, we analyze how well the dataset is balanced. For this purpose, graphs are built with visualization of the number of images for each species of birds. This step was taken for each sample of the dataset separately.

As can be seen in Figure 1, the training sample is quite imbalanced because the number of images for different species of birds ranges from 140 to 260. However, this imbalance is not critical and can be simply ignored. At the same time, Figures 2 and 3 demonstrate that the validation and test datasets are perfectly balanced, as each species of bird in these datasets has exactly five images.

Then it was decided to apply the technique of image augmentation [18]. It provides for applying various transformations to the original images to create new, slightly modified versions. This will increase the invariance of the work of the model, making it more stable and able to classify images in more difficult circumstances, and will contribute to additional regularization due to the introduction of randomness and diversity into the training data, thereby preventing overfitting.

The following transformations were applied:

(1) RandomFlip ("horizontal") horizontally flips the image with a probability of 50%.

(2) RandomTranslation (0.05, 0.05, fill_mode = "nearest") shifts the image vertically and horizontally by a random amount in the range of [−5%, +5%], filling the resulting empty pixels with the value of the nearest pixel from the original image.

(3) RandomRotation (0.05, fill_mode = "nearest") rotates the image by a random amount in the range of [−18°, +18°], filling the resulting empty pixels with the value of the nearest pixel from the original image.

(4) RandomZoom (0.05, fill_mode = "nearest") scales the image by a random amount in the range of [−5%, +5%], filling the resulting empty pixels with the value of the nearest pixel from the original image.

(5) RandomContrast (0.2) adjusts the contrast of the image randomly by the formula $(x - \text{mean}) * \text{factor} + \text{mean}$, where the factor is in the range of [0.8, 1.2].

The result of applying the augmentation technique is shown in Figure 4.

The dataset for model training was also optimized by using the prefetch method [19], which allows data batches to be loaded asynchronously into memory even before the models need them. This approach helps improve workout performance, especially when working with large amounts of data, which is a relevant case for this research work. buffer_size = AUTOTUNE automatically determines the optimal buffer size for maximum performance.

In order to apply transfer learning, we also used the following dataset: ImageNet [20]. The latter is one of the largest and most famous datasets in the field of computer vision (see Figure 5). It includes more than 14 million images that belong to more than 21 thousand classes of completely different kinds. ImageNet has become popular due to its wide variety because it covers a wide range of different objects, from animals and plants to household items and vehicles. This dataset is publicly available and is provided to researchers free of charge for noncommercial use.

2.2. Model Architecture. Further, it was necessary to build the optimal architecture of the classification model based on deep learning. This is an essential task, as it has a significant impact on the final efficiency of the model. It is also laborious since it requires a large number of experiments.

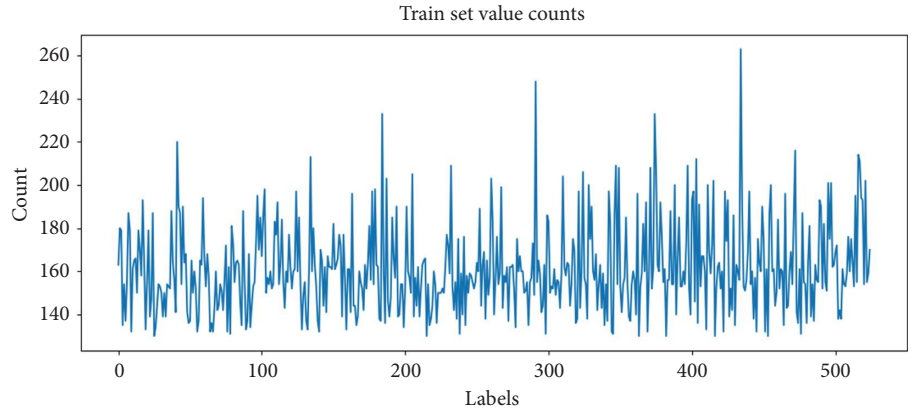

Figure 1: Visualization of the number of images for each bird species in the training sample of the dataset.

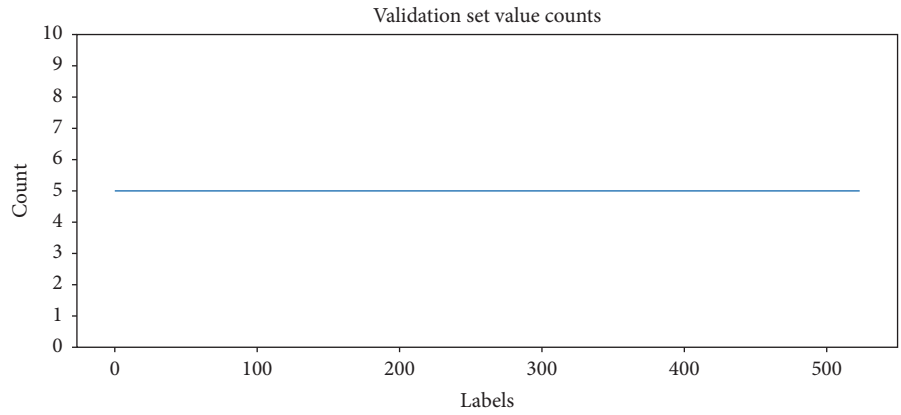

Figure 2: Visualization of the number of images for each bird species in the validation sample of the dataset.

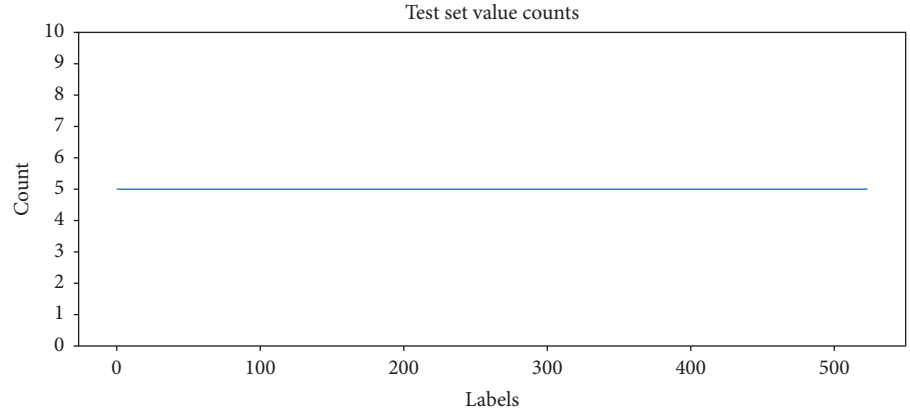

Figure 3: Visualization of the number of images for each bird species in the test sample of the dataset.

It was decided to use the architecture of a convolutional neural network [21], because this is the method of deep learning, which is ideal for solving problems of image classification. The main characteristic of convolutional neural networks is the use of convolutional layers. They apply convolution operations to input images, thereby detecting local patterns (features) on the basis of which classification will take place.

Initially, we used Inception V3 and VGG19 convolutional neural network architectures. As a result, the highest accuracy on the test sample was achieved, which was 92% and 95%, respectively, when training with the Adam optimizer [22]. Eventually, the experiments led to the conclusion that the most optimal solution would be to use transfer training [23]. This is a popular approach in deep learning, the essence of which is that knowledge obtained during the

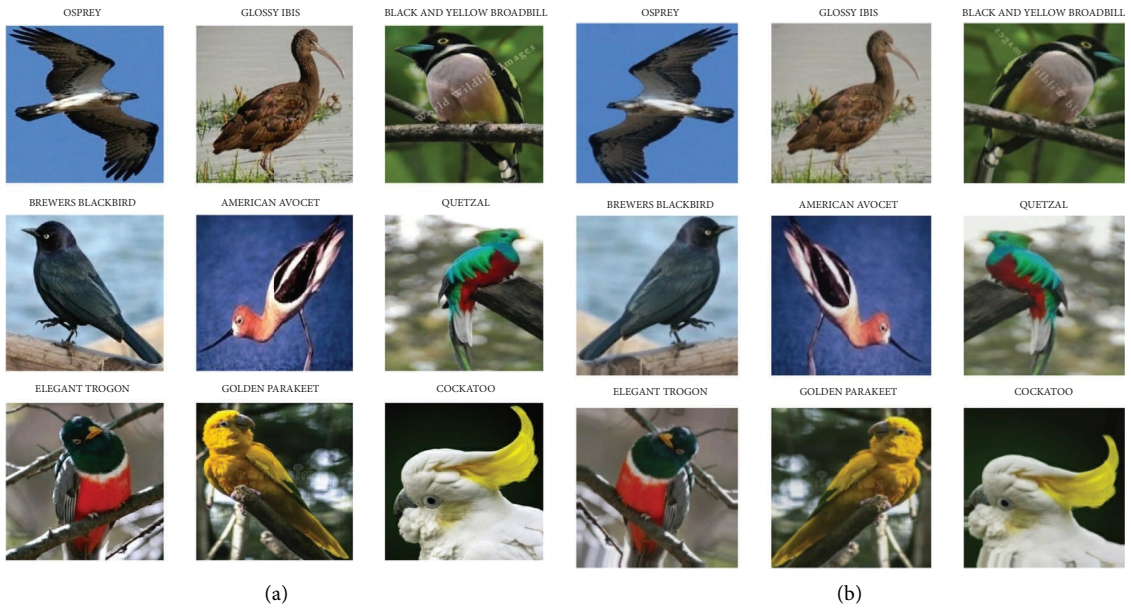

Figure 4: (a) An example from the selected dataset and (b) the same image, but after applying the augmentation technique.

Figure 5: Example of images from the ImageNet dataset.

solution of one problem is reused to solve another. That is, a model previously trained to solve a problem is reused to solve a new one. So, in order to apply this approach in this work, you need to choose a ready-made model of a convolutional neural network, which is pretrained on a suitable dataset.

As a result of an analysis of modern literature [16], it was concluded that the most effective would be the use of the EfficientNetB5 model [24], previously trained on the ImageNet dataset, which was previously described in Section 2.1. The EfficientNetB5 model belongs to the EfficientNet family. The latter belongs to a family of models designed for problems in the field of computer vision, including image classification. These models are characterized by high efficiency with a small number of parameters. The EfficientNet family includes different versions of models, designated from B0 to B7, where B0 is the least powerful model and B7 is the most powerful. EfficientNetB5 differs from smaller versions such as EfficientNetB0, in more options and a deeper architecture. Typically, a deeper architecture allows the model to better adapt to solving complex problems that require a large amount of training data.

In order to load the pretrained model, the keras.applications.EfficientNetB5 function was used. The following parameters were passed to it:

Input_shape = (224, 224, 3): it determines the size of the input data of the model.

Include_top = False: it indicates that the top layer of the model (the one directly responsible for classification based on extracted features) will not be included in the loaded model.

Weights = "imagenet": itindicates that the model will be loaded with the finished weights that it received during training on the ImageNet dataset.

Pooling = "max": it indicates that the last pooling layer in the loaded model architecture will use the maximum pooling operation.

As a result, was loaded a model containing 578 layers.

After the pretrained model was loaded, it was necessary to build the architecture of the final model. To do this, a pretrained model was integrated into it and top layers were added, which already directly perform the classification.

The architecture of the final model is shown in Figure 6 and consists of the following layers:

(1) Input (shape = (224, 224, 3))
(2) Data augmentation
(3) Pretrained model
(4) Dense (1024, activation = "relu")
(5) BatchNormalization ()
(6) Dropout (0.4)
(7) Dense (512, activation = "relu")
(8) BatchNormalization ()
(9) Dropout (0.3)
(10) Dense (256, activation = "relu")
(11) BatchNormalization ()
(12) Dropout (0.2)
(13) Dense (525, activation = "softmax")

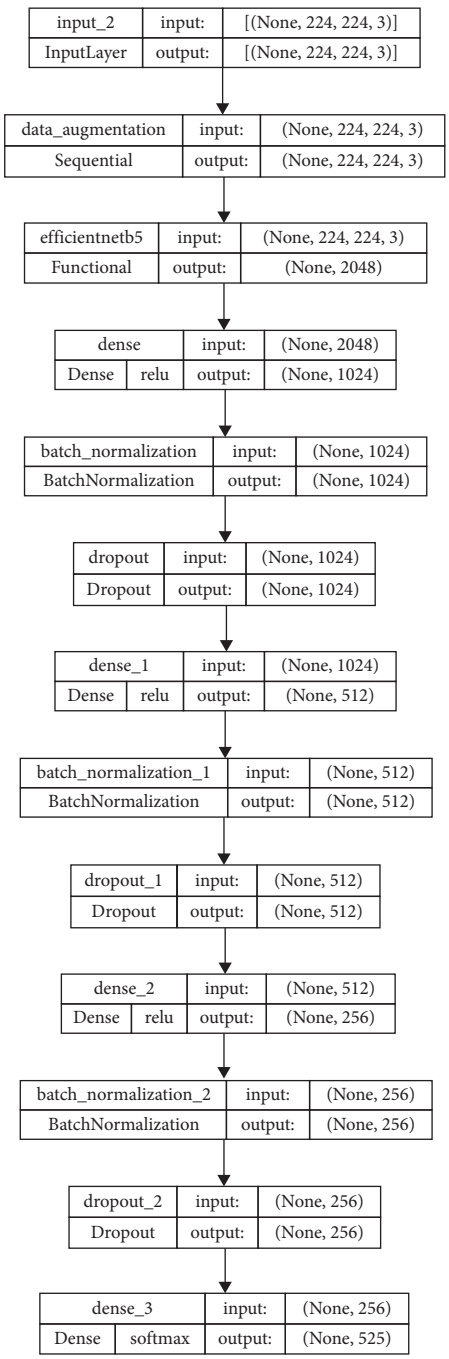

Figure 6: Final model architecture visualization.

Input: it receives a three-dimensional matrix of size (224, 244, 3), which corresponds to a color image of size 224×244 pixels, which coincides with the size of the images of the selected dataset.

Data augmentation: it applies the augmentation technique to the images the model takes as input. This layer is active only during the training of the model because when it is used to solve real problems, augmentation will be completely redundant.

The pretrained model is a pretrained EfficientNetB5 model. Its main work is extracting features from input images.

Dense (fully connected layer): it is used to perform classification based on features that have been extracted by the convolutional layers of the pretrained model.

RELU is an activation function for introducing nonlinearity into a convolutional neural network.

Batch Normalization: it is used to normalize the input data by applying a transformation that makes the average value close to 0 and the standard deviation close to 1.

Dropout: it isused to prevent overtraining of the model, by deliberately losing a certain part of random neurons.

The last dense layer has the same number of neurons as the number of classes in the selected dataset and uses the Softmax activation function to output the vector of the probability of image belonging to a particular class.

2.3. Model Training. After the architecture of the final model was built, it was necessary to proceed to the process of training it. It is necessary in order to fill the model with weights that will be optimal for solving the problem of this work. This stage is key since it is the level of optimality of the weights that determines the effectiveness of the model to a very large extent.

In order to start training the model, it was necessary to determine the optimization algorithm, loss function, and training indicators.

As an optimization algorithm, it was decided to use Adam. This algorithm is an optimization algorithm widely used in deep learning to train models of convolutional neural networks. It combines ideas from other optimization algorithms, such as stochastic gradient descent (SGD) and RMSprop, to enable efficient and adaptive updating of model weights during training. Adam uses stochastic gradient descent to update the model parameters. It also adjusts the learning rate for each parameter separately using the previous gradients and their squares. This adaptability helps the model converge faster and prevents it from getting stuck in local lows. Adam also uses the concept of momentum. Its essence is to accumulate the history of gradients for each parameter of the model during optimization. This story helps stabilize the optimization process, especially under conditions of noise in the data or unstable gradients.

As a function of loss, it was decided to use sparse categorical cross-entropy [25]. This function is used in deep learning to measure the differences between the probability distribution that the model predicts and the authentic class distribution. This loss function is especially useful in multiclass classification problems, where each object can belong to one of the possible classes. Sparse categorical cross-entropy is calculated by comparing the true class distribution with the probability distribution predicted by the model. If the predicted probabilities correspond exactly to the authentic distribution, the loss function equals zero. In other cases, the loss increases, indicating differences between predictions and true values. In order to be able to clearly track the progress of the model training, it was decided to use the accuracy indicator.

It is worth noting that the size of the batches of images the model will take as input for training = 32.

Callbacks for model training were also identified as follows:

> EarlyStopping (monitor = "val_loss," patience = 12); ModelCheckpoint ("model1.h5" monitor = "val_loss," save_best_only = True).
>
> EarlyStopping is used to prematurely stop the model's training process if it has stopped improving [26]. It avoids the unreasonable waste of expensive training resources when the value of the specified indicator of the efficiency of the model ceases to improve over a certain number of eras.
>
> Monitor = "val_loss" indicates that the EarlyStopping callback will observe the value of the loss function on the validation sample of the dataset in order to understand whether the model has stopped improving.
>
> Patience = 12 indicates that to prematurely stop model training, it requires a lack of improvement in the value of the selected indicator for the next 12 epochs from the moment the best value is fixed.

ModelCheckpoint callback is used to automatically store the state of the model (including weights and architecture) during training at certain times [27]. It can be very useful because training on a large amount of data is expensive, and if for some reason it does not end successfully, you can lose all the previously gained weight, which will be a very unpleasant situation.

Also, this callback is useful because it allows you to save exactly the best version of the model, overwriting the old version with it and discarding all the others with worse values of efficiency indicator.

> Filepath = "model1.h5" specifies the file name in which the model will be saved.
>
> Monitor = "val_loss" indicates that the ModelCheckpoint callback will observe the value of the loss function on the validation sample of the dataset. When this value improves, ModelCheckpoint saves the model.
>
> Save_best_only = True indicates that only the best version of the model will be saved, that is, the one for which the value of the selected indicator will be the best.

Then, it was necessary to go directly to the process of training the model.

Since for the pretrained model EfficientNetB5 trainable = false, its layers will be frozen during training (they will not change the weights). The weights will only change at the top layers that were added to the final model after the pretrained one.

The following indicators are used to evaluate the model's performance: loss, accuracy, recall, precision, and $F1$ score.

The following values will be used for the formulas of the following indicators:

True Positives: the number of images that belong to a certain class and have been correctly defined as this class.

False Positives: the number of images that do not belong to a particular class and were mistakenly defined as this class.

True Negatives: number of images that do not belong to a particular class and were correctly defined not as this class.

False Negatives: the number of images that belong to a certain class and were mistakenly defined as not belonging to this class.

Accuracy: it measures the overall correctness of the model classification. It is defined as the ratio of the number of correctly classified images to the total number of images. Accuracy can be useful in the case of a well-balanced dataset:

$$\text{accuracy} = \frac{\text{true negatives} + \text{true positives}}{\text{true positives} + \text{false positives} + \text{true negatives} + \text{false negatives}}. \quad (1)$$

Precision: it measures how much of all images defined by the model as a specific class really belong to this class. It is useful in situations where it is important to maximize the accuracy of the definition of a particular class and avoid erroneous definitions of that class:

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}. \quad (2)$$

Recall: it measures how much of all images belonging to a certain class have been correctly defined by the model as this class. It is useful in situations where it is important to identify as many really positive cases as possible and avoid omissions:

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}. \quad (3)$$

*F*1 score: it measures a generalized assessment of the efficiency of the model by combining precision and recall:

$$\text{F1 score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (4)$$

3. Results

3.1. Phase 1. The model was compiled with a certain optimization algorithm, a loss function, and training indicators. After that, the training process was started using the model.fit function, in which the parameters were transferred to the training and validation samples of the dataset, and the callbacks that were previously defined. It was also determined that the maximum number of training epochs = 100.

The training took place in the Google colab environment, using premium resources (A100 GPU and Colab Pro+), and took several hours. As a result, the model trained the maximum number of epochs (100), and the results achieved are given in Table 1.

The obtained results are additionally visualized in Figures 7 and 8. In particular, the change in the loss indicator is shown in Figure 7, and the change in accuracy during eras is shown in Figure 8.

As can be seen, the accuracy of the model on the validation sample of the dataset was quite high (val_accuracy = 0.9463). This result is satisfactory, but it was decided to conduct additional experiments in order to increase the effectiveness of the model.

3.2. Phase 2. To improve the efficiency of the model, a certain number of layers of the pretrained EfficientNetB5 model were activated (not including BatchNormalization layers, because changing their weights will negatively affect the effectiveness of the model) and continue training the model (start the second phase of training). In general, the trained model kneads 578 layers, and the deeper the layers are, the more complex features they form. That is, it makes sense to activate the last layers, since they are responsible for the formation of high-level features that could be adapted to our task. At the same time, the initial layers form low-level features, and since they will be suitable for our task, the modification of their formation (changing the weights of the initial layers) will not make sense.

The performed experiments demonstrated that the best results were obtained by activating the last 92 layers (not including BatchNormalization layers) of the pretrained model, so this is the number of layers activated for the second phase of training.

In this phase, the learning rate for the Adam optimization algorithm was set to $1e-5$.

Callbacks for model training were also changed:

EarlyStopping (monitor = "val_loss," patience = 13)

ModelCheckpoint ("model2.h5," monitor = "val_loss," save_best_only = True)

ReduceLROnPlateau (monitor = "val_loss," factor = 0.2, patience = 3)

For the EarlyStopping callback, there was a variable parameter patience, now it = 13

For ModelCheckpoint, the filepath parameter has been changed, now it = "model2.h5"

A new ReduceLROnPlateau callback was added.

Table 1: Model training efficiency indicators (phase 1).

Indicators	Values
Loss	0.5515
Accuracy	0.8429
Val_loss	0.1889
Val_accuracy	0.9463

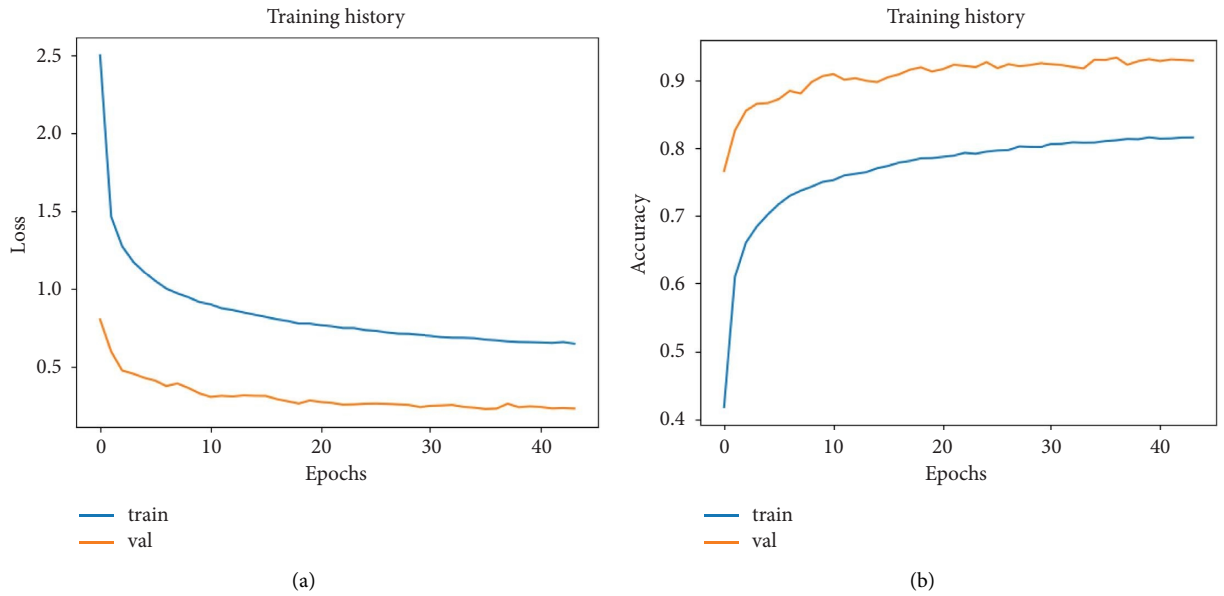

Figure 7: Model training history: loss (phase 1).

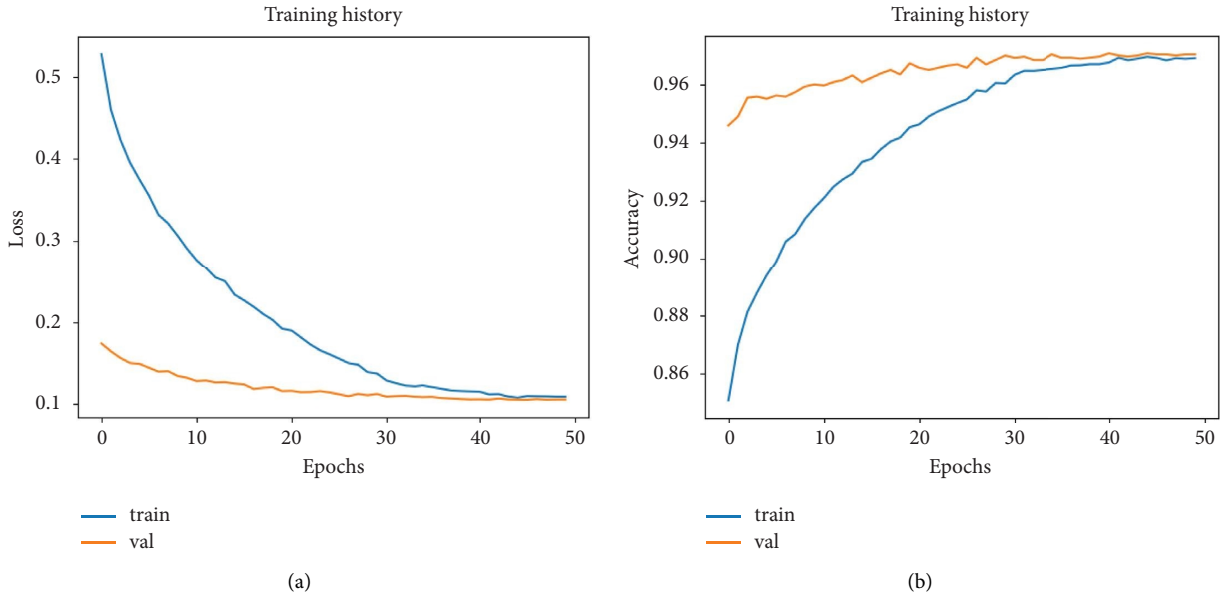

Figure 8: Model training history: accuracy (phase 1).

ReduceLROnPlateau [28] is used to reduce the learning speed after the model stops improving at the previous speed. Its main benefit is that when learning stagnates or sags, reducing speed can help the model increase accuracy and accelerate convergence.

Monitor = "val_loss" indicates that the ReduceLROnPlateau callback will observe the value of the loss function on the validation sample of the dataset in order to understand whether the model has stopped improving.

Table 2: Model training efficiency indicators (phase 2).

Indicators	Values
Loss	0.1212
Accuracy	0.9642
Val_loss	0.1029
Val_accuracy	0.9745

Table 3: Efficiency indicators of the model run on the test sample.

Indicators	Values
Test loss	0.0454
Test accuracy	0.9886
Precision	0.9905
Recall	0.9886
*F*1 score	0.9882

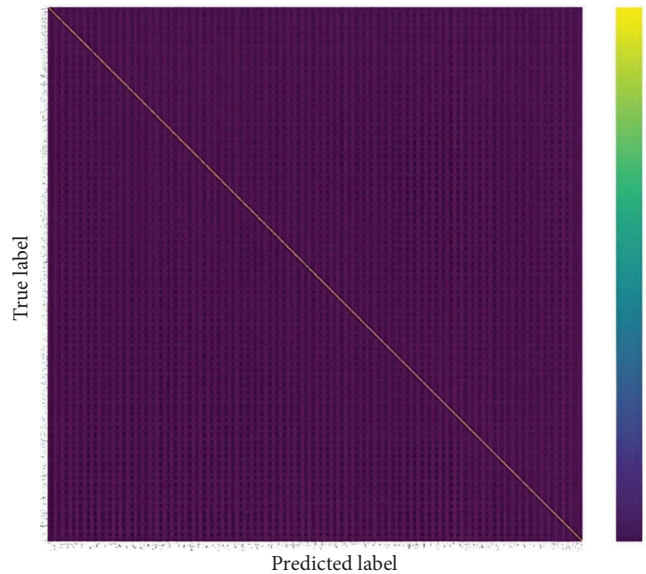

Figure 9: Model error matrix on the test set of data.

Factor = 0.2 indicates that the learning speed will be reduced by 5 times, if there is no improvement in the value of the selected indicator for a certain number of eras.

Patience = 3 indicates that in order for the learning speed to decrease, the selected indicator should not improve over 3 subsequent eras.

The second phase of training once again took place in the Google colab environment, using premium resources (A100 GPU and Colab Pro+), and also took several hours. As a result, the model was trained for 48 epochs, and the results are given in Table 2.

Visualization of the obtained results in the second phase of model training is additionally presented in Figure 8. It was decided to stop the model training process, since the accuracy of the model on the validation sample of the dataset was extremely high (val_accuracy = 0.9745), and there were no more ideas for its further improvement.

After the process of training the model was fully completed, it was necessary to evaluate the effectiveness of its work. The efficiency results after the model was run on the test sample of the dataset are given in Table 3.

The model error matrix on the test sample of the dataset was also visualized (see Figure 9).

In Figure 10, the operation of the model on specific images is demonstrated.

Comparing the model proposed by this research work with those proposed in the analyzed literary sources (see Tables 4 and 5), we can conclude that there are two significant advantages of our model:

(1) High efficiency of the model for the number of bird species equal to 400: in other words, the values of the model efficiency indicators on the test data were lower than in this work–loss = 0.0224, and accuracy = 99.86%.

Figure 10: Demonstration of the operation of the model on specific images.

Table 4: Comparison of models from the relevant literature with the model proposed in this research paper for the number of bird species equals 400.

Sources	Loss	Accuracy (%)
[13]	0.6854	95.05
[14]	0.1426	—
[15]	—	98.60
Our work	0.0224	99.86

Table 5: Comparison of our model with the models from [16] for the BIRDS 525 SPECIES dataset—precision, recall, and *F*1 score.

Models	Precision	Recall	*F*1 score
Inception V3	0.94	0.93	0.93
VGG19	0.96	0.95	0.95
EfficientNetB3	0.98	0.98	0.98
Our work	0.99	0.99	0.99

(2) High efficiency of the model for a large number of different bird species (525) that the model can classify: accuracy = 98.86%, precision = 0.99, recall-= 0.99, and $F1$ score = 0.99.

4. Conclusions

The BIRDS 525 SPECIES dataset which contains 525 species of birds was selected, analyzed, and preprocessed. It was eventually used to train the proposed model. The optimal architecture of the model was also built using the transfer learning approach. For this purpose, the EfficientNetB5 model, which had been previously trained on the ImageNet dataset, was integrated into it.

A new optimization algorithm for the classification of different species of birds in the images was developed. The model training process was carried out, which included two phases: (1) only the upper layers were activated and (2) the last 92 layers of the pretrained EfficientNetB5 model (not including BatchNormalization layers) and the top layers were activated. The efficiency of the model was also evaluated using indicators: loss, accuracy, precision, recall, and F score. An error matrix was visualized so that the model could be analyzed for different classes. As a result, the efficiency indicators accuracy = 98.86%, precision = 0.99, recall = 0.99, and $F1$ score = 0.99 were obtained. A comparative analysis of the obtained indicators with the corresponding indicators obtained by other authors was also carried out. From the analysis of the results obtained, it can be argued that the task of classifying various species of birds was effectively managed while ensuring high accuracy.

As future research, it would be important to optimize the proposed algorithm in order to accelerate model training and the possibility of obtaining a real-time solution [29, 30].

Data Availability

All data are fully accessible without restrictions and links are provided in the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The National Research Foundation of Ukraine funds this study from the state budget of Ukraine within the project no. 2023.03/0029.

References

[1] C. Kwan, G. Mei, X. Zhao et al., "Bird classification algorithms: theory and experimental results," in *Proceedings of the 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, QC, Canada, May 2004.

[2] M. Kumar, A. K. Yadav, M. Kumar, and D. Yadav, "Bird species classification from images using deep learning," in *Computer Vision and Image Processing. CVIP*, D. Gupta, K. Bhurchandi, S. Murala, B. Raman, and S. Kumar, Eds., Springer, Cham, 2023.

[3] L. Mayats-Alpay, "Deep learning methods for automatic classification of medical images and disease detection based on chest X-Ray images," 2022, https://arxiv.org/abs/2211.08244.

[4] L. Mochurad, A. Dereviannyi, and U. Antoniv, "Classification of X-ray images of the chest using convolutional neural networks. IDDM 2021 informatics & data-driven medicine," in *Proceedings of the 4th International Conference on Informatics & Data-Driven Medicine*, pp. 269–282, Valencia, Spain, November 2021.

[5] M. M. Bykov, V. V. Kovtun, A. Smolarz, M. Junisbekov, A. Targeusizova, and M. Satymbekov, "Research of neural network classifier in speaker recognition module for automated system of critical use," in *Proceedings of the Symposium on Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments WILGA*, Wilga, Poland, August 2017.

[6] R. Kaminsky, L. Mochurad, N. Shakhovska, and N. Melnykova, "Calculation of the exact value of the fractal dimension in the time series for the box-counting method," in *Proceedings of the 2019 9th International Conference on Adanced Computer Information Technologies ACSIT'2019*, pp. 248–251, Ceske Budejovice, Czech Republic, June 2019.

[7] F. A. Phang, J. Pusppanathan, N. D. Nawi et al., "Integrating drone technology in service learning for engineering students," *International Journal of Emerging Technologies in Learning*, vol. 16, no. 15, pp. 78–90, 2021.

[8] D. Chumachenko, T. Dudkina, and T. Chumachenko, "Assessing the impact of the Russian war in Ukraine on COVID-19 transmission in Spain: a machine learning-based study," *Radioelectronic and Computer Systems*, vol. 1, pp. 5–22, 2023.

[9] Y. Rashkevych, D. Peleshko, O. Vynokurova, I. Izonin, and N. Lotoshynska, "Single-frame image super-resolution based on singular square matrix operator," in *Proceedings of the 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)*, pp. 944–948, Kyiv, UKraine, June 2017.

[10] N. Bold, C. Zhang, and T. Akashi, "Bird species classification with audio-visual data using CNN and multiple kernel learning," in *Proceedings of the 2019 International Conference on Cyberworlds (CW)*, pp. 85–88, Kyoto, Japan, October 2019.

[11] K. Wang, F. Yang, Z. Chen, Y. Chen, and Y. Zhang, "A fine-grained bird classification method based on attention and decoupled knowledge distillation," *Animals*, vol. 13, no. 2, p. 264, 12.

[12] M. Alswaitti, L. Zihao, W. Alomoush, A. Alrosan, and K. Alissa, "Effective classification of birds' species based on transfer learning," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 4, pp. 4172–4184, 2022.

[13] A. Manna, N. Upasani, S. Jadhav, R. Mane, R. Chaudhari, and V. Chatre, "Bird image classification using convolutional neural network transfer learning architectures," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, 2023.

[14] T. Pal, "400 birds species classification," https://medium.com/mlearning-ai/400-birds-species-classification-f4de768aac4.

[15] A. Reslan and Z. Farou, *Automatic Fine-grained Classification of Bird Species Using Deep Learning*, Budapest, p. 56, 2022.

[16] H.-T. Vo, N. N. Thien, and K. C. Mui, "Bird detection and species classification: using YOLOv5 and deep transfer learning models," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, 2023.

[17] kaggle, "Birds 525 species-image classification," https://www.kaggle.com/datasets/gpiosenka/100-bird-species/code.

[18] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognition*, vol. 137, pp. 109347–109412, 2023.

[19] J. Zhang and B. Jiang, "A kind of metadata prefetch method for distributed file system," in *Proceedings of the 2021 International Conference on Big Data Analysis and Computer Science (BDACS)*, Kunming, China, June 2021.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: a large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, June 2009.

[21] J. Ayeni, "Convolutional neural network (CNN): the architecture and applications," *Applied Journal of Physical Science*, vol. 4, no. 4, pp. 42–50, 2022.

[22] J. Feng, Z. Wang, M. Zha, and X. Cao, "Flower recognition based on transfer learning and Adam deep learning optimization algorithm," in *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence*, pp. 598–604, Shanghai China, September 2019.

[23] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," *J Big Data*, vol. 9, no. 1, p. 102, 2022.

[24] N. K. Chowdhury, M. A. Kabir, M. M. Rahman, and N. Rezoana, "ECOVNet: a highly effective ensemble based deep learning model for detecting COVID-19," *PeerJ Computer Science*, vol. 7, p. e551, 2021.

[25] N. Dousti Mousavi, H. Aldirawi, and J. Yang, "Categorical data analysis for high-dimensional sparse gene expression data," *BioTechnologia*, vol. 12, no. 3, p. 52, 2023.

[26] Y. Bai, E. Yang, B. Han et al., "Understanding and improving early stopping for learning with noisy labels," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24392–24403, 2021.

[27] A. Sagar and J. Dheeba, "On Using Transfer Learning for Plant Disease Detection," 2020, https://www.biorxiv.org/.

[28] M. Abulaish and R. Gulia, "Compact residual learning with frequency-based non-square kernels for small footprint keyword spotting," in *Proceedings of the 19th International Conference on Natural Language Processing (ICON)*, pp. 328–336, Association for Computational Linguistics, New Delhi, India, 2022.

[29] L. Mochurad, K. Shakhovska, and S. Montenegro, "Parallel solving of fredholm integral equations of the first kind by tikhonov regularization method using OpenMP technology," in *Advances in Intelligent Systems and Computing IV. CCSIT 2019*, N. Shakhovska and M. Medykovskyy, Eds., Springer, pp. 25–35, Cham, 2020.

[30] L. Mochurad, "Optimization of regression analysis by conducting parallel calculations," in *Proceedings of the COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems*, pp. 982–996, Kharkiv, Ukraine, April 2021.