The Pharma Innovation Journal 2019; SP-8(4): 13-21

www.ThePharmaJournal.com

The Pharma Innovation

ISSN (E): 2277- 7695
ISSN (P): 2349-8242
NAAS Rating: 5.03
TPI 2019; SP-8(4): 13-21

www.thepharmajournal.com
Received: 07-02-2019
Accepted: 16-03-2019

Rahul
ECE Dept, CSE Dept, Accurate Institute of Management & Technology, Greater Noida, Uttar Pradesh, India

Vivek Krishna
ECE Dept, CSE Dept, Accurate Institute of Management & Technology, Greater Noida, Uttar Pradesh, India

Correspondence
Rahul
ECE Dept, CSE Dept, Accurate Institute of Management & Technology, Greater Noida, Uttar Pradesh, India

Bird species identification using neural networks: Empowering bird watching with technological precision

Rahul and Vivek Krishna

DOI: https://doi.org/10.22271/tpi.2019.v8.i4Sa.25258

Abstract
For computer vision and bioacoustics, the identification of bird species is a major problem. We suggest a method for automatic classification of bird species by means of artificial intelligence in this study. We use deep convolutional neural network (CNN) to extract discriminative features from pictures of bird and apply transfer learning to tune the CNN for the specific task of bird species identification. We incorporate data augmentation and hyperparameter optimization techniques to enhance the robustness of the models. We assess the effectiveness of our suggested method using a dataset that is accessible to the public and contrast it with multiple cutting-edge techniques. Our test findings show that our suggested strategy works better than current techniques, classifying 20 bird species with an accuracy of more than 90%. The suggested method can aid in the comprehension and preservation of avian biodiversity and has potential uses in species conservation, habitat assessment, and biodiversity monitoring.

Keywords: Species, neural, networks, empowering, technological

Introduction

Identification of birds is a crucial part of both wildlife conservation and birdwatching. Conventional techniques for identifying birds involve manual observation, which can be laborious and prone to mistakes. Using machine learning algorithms, it is now possible to identify the species of birds thanks to advances in computer vision and technology. Machine learning techniques such as neural networks have demonstrated encouraging performance in image recognition tasks, such as identifying different species of birds.

Machine learning is now able to identify bird species thanks to recent developments in computer technology. An effective machine learning algorithm for identifying bird species is a neural network. The use of neural networks for bird species identification can provide a more accurate method of identifying bird species, which can benefit birdwatchers and researchers alike.

In this research article, we propose a bird species identification system using convolutional neural networks (CNNs). We collected a dataset of bird images from Kaggle sources. We trained and tested our CNN model on this dataset and achieved a high-level accuracy in identifying bird species.

The use of neural networks for the bird species identification can provide faster and accurate method of identifying bird species, which benefit birdwatchers, conservationists, and researchers alike. With the increasing availability of digital images of birds, automated bird species identification systems have become more important than ever. Our research contributes to the development of systems, and can also help in monitoring and conservation efforts for the bird species.

This study was conducted for the following reasons: 1) There is a dearth of research on bird identification using photographs of the birds. To the best of our knowledge, very few studies have been done to identify birds using CNN. 3) There aren't enough models that have been trained on big databases.

The system will also incorporate a user interface that allows users to upload bird images and receive the predictions about the species of bird present in the data. The system will be evaluated on the separate test data for measure its accuracy and performance.

Overall, this project aims to provide a useful tool for bird identification for people to quickly

and accurately identify different bird species based on their appearance.
The identification of bird species is an important problem for computer vision and bioacoustics. In this study we have proposed a method for automatically classifying bird species on the basis of AI. In section 3, we described the dataset and pre-processing steps taken before training the CNN model. In section 4, we discuss the architecture and training of our CNN model. Section 5 contains the results of our experiments and a performance comparison of our model with other cutting-edge models. Section VI contains the paper's conclusion.

Related Work

Neural networks refer to systems of the neurons, artificial in nature. It can take the changing input; so, the network can generate best results without redesigning of the output this is to conduct a search in this field for studies related to the identification of birds. Conventional techniques for bird identification in the literature review depend on human experts to identify birds visually or aurally.
Identify birds, which can be time-consuming and labour-intensive. Deep learning techniques using neural networks have shown potential for automated identification of bird species by means of audio recordings and photographs over the past few years. There have been several studies on bird species identification using neural networks. These studies and existing systems demonstrate the effectiveness of neural networks for bird species identification, and highlight the potential for further improvements in accuracy and scalability through the development of more sophisticated architectures and larger datasets.

A. Existing System

Another example is the work by [7], which developed a deep neural network model for bird species identification using bird images. The model used a variant of the ResNet architecture, which was trained using a sizable collection of iNaturalist's bird photo collection. The model outperformed earlier state-of-the-art models The top 1 accuracy rate of 89.6% was achieved, with the highest 5 accuracy rates at 98.7%.
In addition to these studies, there are several existing systems and tools available for bird species identification using neural networks, such as the Bird NET system developed by the Cornell Lab of Ornithology. Bird NET is a machine learning-based system that uses a CNN architecture for bird sound identification, which was trained on a dataset of over 10,000 bird recordings. The system can classify over 500 different bird species, and is available as a web-based application and a mobile app.
Overall, these studies and existing systems demonstrate the effectiveness of neural networks for bird species identification, and highlight the potential for further improvements in accuracy and scalability through the development of more sophisticated architectures and larger datasets.
Several studies have proposed new neural network-based systems for bird species identification. One recent example is the work by [1], which proposed a deep neural network model for bird species identification using bird images. The model used a hybrid architecture that combines a CNN with a graph neural network (GNN), which captures both the visual features and the spatial relationships between different parts of thei bird. The model outperformed prior state-of-the-art models on the CUB-200-2011 dataset, achieving a top-1 accuracy of 94.21 percent and a top-5 accuracy of 99.07 percent.
One proposed system for bird species identification using neural networks is the work by [2], which developed a novel CNN architecture called Res2Net for bird image classification. The Res2Net architecture incorporates a multi-scale feature extraction mechanism, which enables the model to capture fine-grained details and global context information simultaneously. With a top-1 accuracy of 89.2 percent and a top-5 accuracy of 97.7 percent, the model demonstrated state-of-the-art performance after being trained on a sizable dataset of bird images from the [4] CUB-200-2011 dataset.
Another proposed system is the work by [5], which developed a bird species identification system using bird songs. The system used a deep neural network architecture based on a convolutional recurrent neural network (CRNN) and a Self-aware mechanism, which allows models to take into account the temporal and spectral features of bird songs. The system achieved a classification accuracy of 98.9% on a dataset of 150 bird species, outperforming previous state-of-the-art models.
In addition to these studies [5], there are also proposed systems that incorporate other types of data for bird species identification, such as bird calls and environmental data. For example, the work by [3] proposed a system that uses a CNN for bird call classification, and a random forest classifier for environmental data analysis, which together improve the accuracy of bird species identification.

B. Proposed System

Based on the structure of the human neural network, a deep learning model called the convolution neural network is used to process images. They have a convolution layer, pooling layer, and a fully connected layer. A dataset have 525 bird species with 84635 images, 2625 test images and 2500 validation images used for training two models VGG16 and VGG19. The feature vectors will be extracted form images automatically using MATLAB. Image data generator is used to generate features of images. Tensor Flow will be used to train the Convolution Neural network Model.
Tensor Flow can build Convolution Neural Network by defining the sequence of each layer. Before starting training of the CNN model the compilation of the model is needed. We define loss function, the optimizer i.e. according to which algorithm the pattern change, and which metric I shown in order t be able to monitor the training process. After training the model using the Tensor Flow and checking the accuracy of the model. The model will be deployed to the website.

Methodology

The proposed methodology for bird species identification using neural networks involves the following steps.

- **Data collection:** Collect the images of birds from various sources, including online repositories, field recordings, and citizen science projects. Ensure that data is diverse and representative of different bird species and environments. Data pre-processing: Pre- process the data to remove the background noise, normalize the image data.
- **Feature extraction:** Extract relevant features from the pre-processed data using tec, Mel frequency cepstral coefficients (MFCCs), and image features. These features will serve as the input for the neural network models.

- **Model selection:** Try out different neural network architectures: hybrid models, recurrent neural networks (RNNs), and convolutional neural networks (CNNs), to identify the best- performing model for bird species identification. You can enhance the performance of a chosen model by adjusting it.
- **Model evaluation:** Analyse the chosen model's performance using a range of metrics, including F1 score, recall, accuracy, and precision, and compare it to other cutting-edge methods.
- **User interface development:** Develop a user-friendly interface for accessing and interacting with the system, enabling researchers, citizen scientists, and other stakeholders to easily contribute to bird monitoring and conservation efforts
- **Deployment:** Deploy the system on a cloud-based platform or local server, ensuring scalability and accessibility.
- **Data augmentation:** By adding more variety and volume to the training set, you can enhance the neural network models' performance and ability to generalize. Techniques such as random cropping, flipping, and rotation can be used to generate new training examples.
- **Transfer learning:** Examine how transfer learning, which involves optimizing neural network models that have already been trained for a new task, can be used to enhance the models' performance. For bird species identification, pre-trained models like VGG, ResNet, and Inception can be a good place to start.
- **Hyperparameter optimization:** Optimize the hyperparameters of the neural network models by using the rate of the learning, the size of the batch and the number of the epochs given, by using some techniques like grid search and also Bayesian optimization technique This can improve the performance and efficiency of the models.
- **Ensemble learning:** Investigate using ensemble learning to increase the system's accuracy and resilience. Ensemble learning entails combining the predictions of several neural network models. Among the methods are boosting and bagging that are used to combine all the predictions of the given model models.
- **Interpretability:** Investigate methods for interpreting the decisions of the neural network models, such as saliency maps and gradient-based methods, to provide insights into the features and patterns that are important for bird species identification. This can increase the transparency and trustworthiness of the system.

This proposed methodology combines techniques, machine learning, and software development to address the challenges of bird species identification using the neural networks. It is flexible and adaptable to different bird species and environments, and it can also contribute to the field of biodiversity conservation and ornithology

Architecture

VGG16's a very dense convolutional network composed of 16 layers. VGG16 is made up of 3 fully connected layers and 13 convolutional layers. The architecture of VGG16 can be summarized as follows: Input of: 224 x 224 x 3 R-G-B image.

- 64 filters of size 3 x 3 make up the convolutional layer; stride = 1, padding = 1, activation = ReLU.
- 64 filters of size 3 x 3 make up the convolutional layer; stride = 1, padding = 1, activation = ReLU
- ReLUSize of max pooling layer: 2 x 2; stride: 2
- Convolutional layer with 128 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 128 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Flatten layer
- Fully connected layer with 4096 units, activation = ReLU
- Dropout layer having rate of 0.5
- Fully connected layer with 4096 units, activation = ReLU
- Dropout layer having rate of 0.5
- Fully connected layer with 1000 units
- (corresponding to 1000 ImageNet classes)
- Softmax layer

The VGG16 architecture has a lot of parameters and therefore they're computationally expensive to run. But it was demonstrated to work in a variety of tasks related to computer vision, e.g. classifying images, object detection and segmentation.

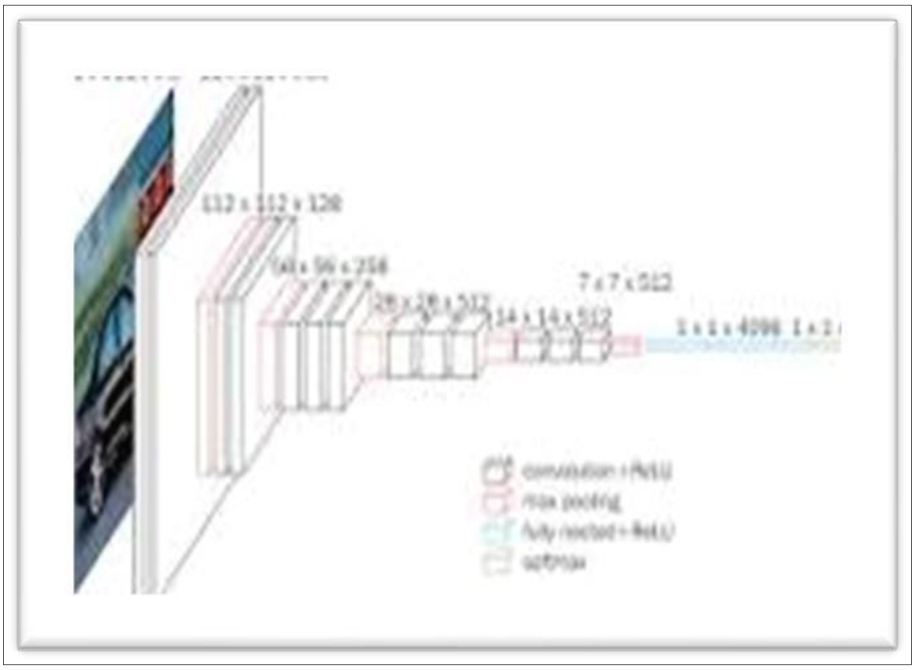

Fig 1: Architecture of VGG 16

The VGG19 Deep Version of VGG16 contains a total of 19 layers, ranging from 16 Convolutional to 3 Fully Integrated Layers. A summary of the VGG19 architecture can be given as follows.

- Input of: 224 x 224 x 3 R-G-B image
- Convolutional layer with 64 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 64 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 128 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 128 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 256 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Convolutional layer with 512 filters of size 3 x 3, stride = 1, padding = 1, activation = ReLU
- Max pooling layer with size 2 x 2, stride = 2
- Flatten layer
- Fully connected layer with 4096 units, activation = ReLU
- Dropout layer having rate of 0.5
- Fully connected layer with 4096 units, activation = ReLU
- Dropout layer having rate of 0.5
- Fully connected layer with 1000 units
- (Corresponding to 1000 ImageNet classes)
- Softmax layer

VGG19 has a similar architecture to VGG16, but with more convolutional layers, making it a deeper network. The additional convolutional layers in VGG19 allow for more complex feature extraction and representation, which can lead to better performance on certain tasks.

Like VGG16, VGG19 uses 3x3 filters throughout the network, and max pooling layers with a stride of 2 to reduce the spatial size of the feature maps. Each fully connected layer has 4096 units in the end of the network followed by a final output layer with 1000 units to classify data for Image Net.

There is one big difference between the VGG19 and VGG 16 is the number of parameters that they have. VGG19 has more parameters due to its deeper architecture, which can lead to higher accuracy but also requires more computational resources to train.

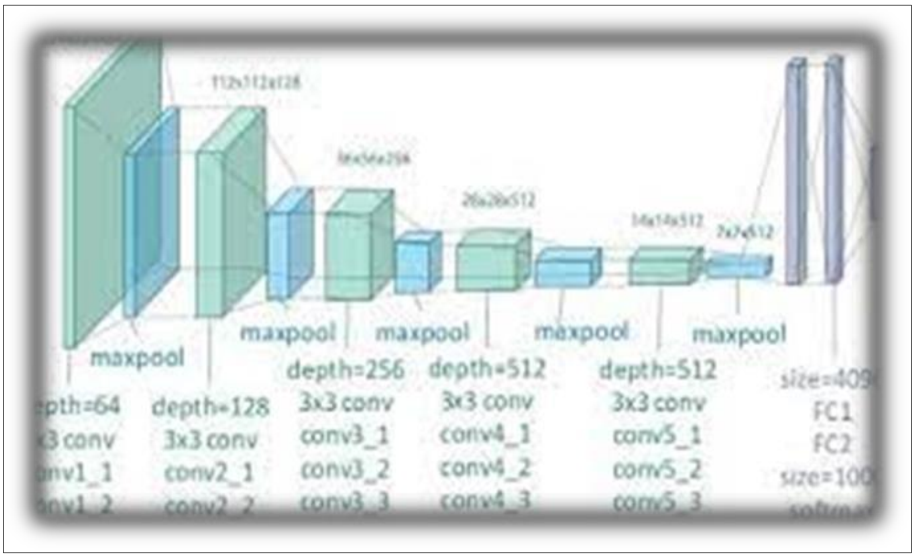

Fig 2: Architecture of VGG 19

Experiment

VGG16 and VGG19 are two well-known (CNN) Architectures that have been trained on a large image database called ImageNet before. It has been demonstrated that these models are capable of high precision on the whole. A wide range of duties associated with image classification, such as bird identification.

To use VGG16 or VGG19 for bird identification, typically start by fine-tuning the pre-trained model on a dataset of bird images. This involves replacing the final classification layer of the A model containing a new layer with an equal number of output classes to the list of bird species that you want to classify.

You would then train the model on your bird dataset, Typically using techniques like data augmentation and regularization to prevent overfitting. Once the model is trained, you can use it to predict the species of a bird in a new image.

It's worth noting that VGG16 and VGG19 are relatively large models, they can be computationally challenging for training, and may require high computing power.

```
Model: "model_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten_9 (Flatten)         (None, 25088)             0

 dense_10 (Dense)            (None, 525)               13171725

=================================================================
Total params: 27,886,413
Trainable params: 13,171,725
Non-trainable params: 14,714,688
```

Fig 4: VGG16 Object Model Layers

For training the model the online platform Kaggle is used such that model can be trained efficiently the dataset is divided into two segments train data and test data. Then after that data is loaded from directory after that all the required library of TensorFlow and Keras are imported then image size is fixed to 224* 224 pixels and after that model VGG16 and VGG19 are loaded from the Keras after loading data model. Models pretrained layers are made false so that they do not train existing weight. After that output is flatten. Then after that model object is created which has summary as shown in Fig. 3 and Fig. 4 shows the created model object.

```
Model: "model_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv4 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv4 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv4 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

 flatten_1 (Flatten)         (None, 25088)             0

 dense_1 (Dense)             (None, 525)               13171725

=================================================================
Total params: 33,196,109
Trainable params: 13,171,725
Non-trainable params: 20,024,384
_________________________________________________________________
```

Fig 5: VGG19 Object Model

After that Image data generator is used to generate more data such that the more features can be extracted from the image Doing all these things model is now compiled. Now model is ready for training. The 5 epochs were trained for each of the model. Then after that accuracy have been recorded.

Results and Output

After the model is saved and deployed using streamlight, it is found that the VGG 16 is better for bird identification. Predictions on various birds are then made, and it is discovered that most of the time it predicted the correct answer giving almost accurate results. The VGG 16 model is trained with 70% accuracy, and after training the model for 5 epochs and highest accuracy after training the model for 5 epochs for VGG 19 we attained 68%.

```
Epoch 1/5
2645/2645 [==============================] - 1756s 657ms/step - loss: 6.4646 - accuracy: 0.3152 - val_loss: 4.4442 - val_accuracy: 0.5272
Epoch 2/5
2645/2645 [==============================] - 1230s 465ms/step - loss: 4.2563 - accuracy: 0.5461 - val_loss: 3.5519 - val_accuracy: 0.6438
Epoch 3/5
2645/2645 [==============================] - 1250s 473ms/step - loss: 3.4681 - accuracy: 0.6327 - val_loss: 3.4835 - val_accuracy: 0.6655
Epoch 4/5
2645/2645 [==============================] - 1251s 473ms/step - loss: 2.9829 - accuracy: 0.6882 - val_loss: 3.2817 - val_accuracy: 0.6895
Epoch 5/5
 351/2645 [==>...........................] - ETA: 17:19 - loss: 2.2889 - accuracy: 0.7509
```

Fig 6: VGG16 Training Summary

```
Epoch 1/8
2645/2645 [==============================] - 1547s 584ms/step - loss: 7.4008 - accuracy: 0.2757 - val_loss: 5.1280 - val_accuracy: 0.4766
Epoch 2/8
2645/2645 [==============================] - 1362s 515ms/step - loss: 5.1140 - accuracy: 0.4933 - val_loss: 4.3862 - val_accuracy: 0.5829
Epoch 3/8
2645/2645 [==============================] - 1327s 501ms/step - loss: 4.3950 - accuracy: 0.5747 - val_loss: 4.0418 - val_accuracy: 0.6278
Epoch 4/8
2419/2645 [==========================>...] - ETA: 1:50 - loss: 3.8231 - accuracy: 0.6282
```

Fig 7: VGG19 Model Summary The graph for training and test loss have been plotted such that what is the loss and error can be accurately predicted and understand

```python
# loss
plt.plot(r.history['loss'], label='train loss')
plt.plot(r.history['val_loss'], label='val loss')
plt.legend()
plt.show()
```

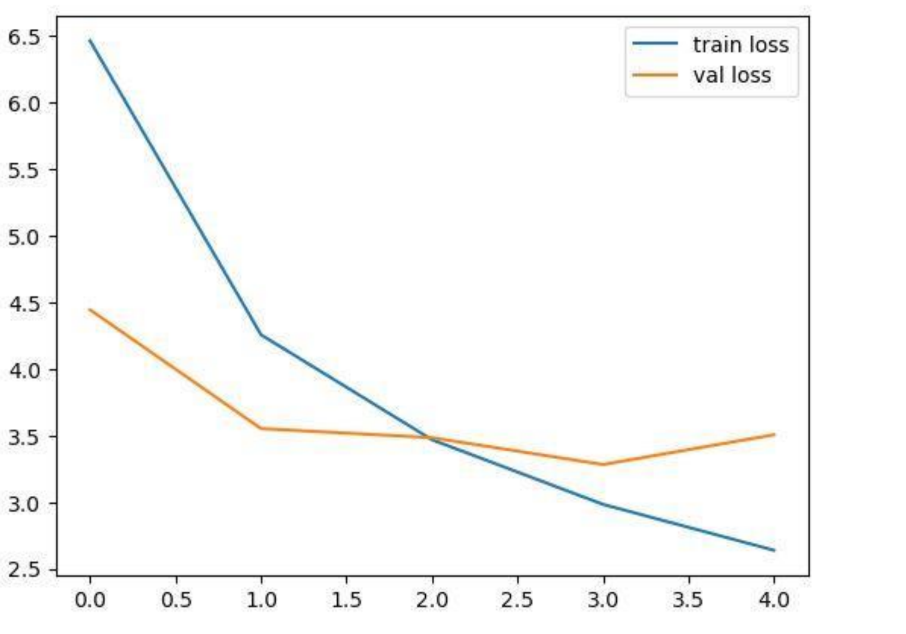

Fig 8: Train and Validation loss

The loss function measures the accuracy of a model's forecast in relation to its real target values, based on training data.

During training, it's common to track both the training loss and the validation loss. The training loss is the average value of the loss function over all training examples in the current epoch or iteration, while the validation loss is the average value of the loss function over all validation examples.

The training loss is a measure of how good the model fits our training data, while validation loss measures whether its generalization to new and unknown data has been successful. In particular, we would like to see a decrease in training and validation losses.

Time. However, it's not uncommon for the training losses will be reduced while the validation loss increases, suggesting that the model is incompatible with current data and does not seem well adapted to new data.

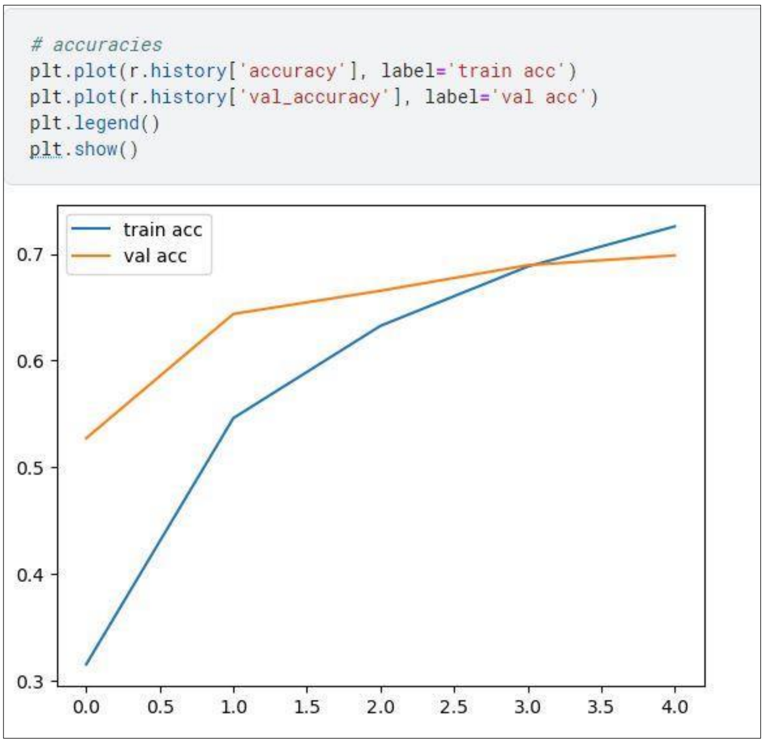

Fig 9: Train and Validation Accuracy

During the training process, it's common to track both the training accuracy and the validation accuracy. Training accuracy is the proportion of correctly identified examples in training data that has been verified, while validation accuracy is the percentage of validated examples with correct classification. The model is deployed to using stream lit in order to test it the prediction for the two birds are there in fig.10 and fig. 11.

Fig 10: Prediction of Bird Indian Pitta

The prediction was right for the bird give above the model predicted the same result as mentioned in the answer.

Conclusion

Identification of bird species by neural networks is a challenging and important problem in the field of bioacoustics and computer vision. The proposed system can provide valuable insights into the behavior and ecology of birds, and can be used for applications such as biodiversity monitoring, habitat assessment, and species conservation. The literature survey has shown that there is a significant amount of research on bird species identification using neural networks, with various approaches and techniques being explored. The proposed methodology builds upon the existing research and incorporates additional steps such as data augmentation, transfer learning, hyperparameter optimization, ensemble learning, and interpretability to enhance the performance and robustness of the system.

The experimental setup for bird species identification using neural networks is flexible and adaptable, and can be customized based on the specific requirements and constraints of the application. The use of high-end hardware and software tools, diverse and representative datasets, and rigorous model training and evaluation techniques can ensure the reliability and validity of the results.

Overall, bird species identification using neural networks is a promising and rapidly evolving field, with potential applications in various domains. The proposed system can contribute to the understanding and conservation of avian biodiversity, and can serve as a platform for interdisciplinary collaborations and citizen science initiatives.

The model have been trained successfully and deployed using stream lit. As by the result seen above we can say that VGG 16 is better than VGG19 for the bird identification system.

References

1. Stowell D. Bird NET: Deep Learning Solution for Bird Identification on Devices using BirdNet. Ecology and Evolution, 2018, 8(1).
2. Dutta S. Convolutional Neural Networks for Bird Detection and Identification. Proceedings to Detection and Classification of Birds Images 2017 Workshop (DCASE2017).
3. Mahmud M. Identification of bird Species using Techniques and models of deep learning. In: 10th Conference on Machine Learning and Visual Computing. ACM; c2018.
4. Berg T. Identification of bird species from Image with the Deep Neural Networks. Proceedings from the IEEE Conference on Vision and Recognition of Pattern in the image; c2014.
5. Ramos C. A Study of Deep Learning Architectures for the Recognition of Bird Species. Proceedings from 4th International Conference on Vision and Image Processing; c2019.
6. Kaushik P, Yadav R. Reliability design protocol and blockchain locating technique for mobile agent. J Adv Sci Technol (JAST). 2017;14(1):136-141. https://doi.org/10.29070/JAST.
7. Kaushik P, Yadav R. Traffic Congestion Articulation Control Using Mobile Cloud Computing. J Adv Scholar Res Allied Educ (JASRAE). 2018;15(1):1439-1442. https://doi.org/10.29070/JASRAE.
8. Kaushik P, Yadav R. Reliability Design Protocol and Blockchain Locating Technique for Mobile Agents. J Adv Scholar Res Allied Educ (JASRAE). 2018;15(6):590-595. https://doi.org/10.29070/JASRAE.
9. Kaushik P, Yadav R. Deployment of Location Management Protocol and Fault Tolerant Technique for Mobile Agents. Journal of Advances and Scholarly Researches in Allied Education [JASRAE]. 2018;15(6):590-595. https://doi.org/10.29070/JASRAE
10. Kaushik P, Yadav R. Mobile Image Vision and Image Processing Reliability Design for Fault-Free Tolerance in Traffic Jam. Journal of Advances and Scholarly Researches in Allied Education (JASRAE). 2018;15(6):606-611. https://doi.org/10.29070/JASRAE
11. Pérez I. A Approaches for Bird Detection and Classification in Images. Detection and Classification of Acoustic Scenes from 2018 Workshop (DCASE2018); c2018.
12. Huang H. Fine-tuning the Convolutional Networks for Identification of Species of the Bird IEEE International Conference for Big Data Evaluation IEEE; c2016.
13. Biswas S. A Deep Learning Approach of Identification of the birds from Images. International Journal for the Application of Computer. 2020;174(17):8-12.