Course:	CS6242 - Data and Visual Analytic
Title:	Project Final Report
Name:	Team 133

1 INTRODUCTION

Many existing bird species image identification applications are limited by factors such as bird species geography or being available exclusively on mobile app stores. The purpose of our team's project is to provide a quick and easy-to-use local website interface through which users can upload a bird image to determine the likely bird species of the image provided. The image identification will be done through a back-end, trained Convolution Neural Network (CNN). This tool will be free to use and "plug and play". Interested parties, such as bird photography enthusiasts/hobbyists and researchers, may find this tool useful for purposes such as personal enjoyment, building bird species image/classification databases, and research development. Accurate identification of bird species has applications in different types of entities such as academia, non-profit/conservation organizations, and general photography hobbyists.

2 PROBLEM DEFINITION

Our project's problem definition consists of two major parts. The first part is the actual process of bird species image identification. Using a bird image training data set, we built a model that takes a bird image and accurately identifies the bird species. The second part is the model visualization. In this part, we deployed the model through an interactive and intuitive web page user interface, where the user can quickly upload a locally saved bird image and identify the bird species. In addition to providing the user with the three most probable bird species, the application also gives the user a short summary of each of the returned bird species, as well as the likelihood of the bird matching each of the returned bird species based on the confidence of the model.

3 SURVEY

Ref	Main Idea	Usefulness	Potential Shortcomings
[1]	Image classification of 50 Philippine bird species using a CNN	Introduced to the idea of Transfer Learning to fine-tune a pre-trained model for image classification.	Image classification is limited to Philippine bird species only.
[2]	Comparing Faster R-CNN and YOLOv5 for bird identification of specific Indian region	Multiple ML methods we can potentially use. Baseline accuracy for comparison	Smaller dataset, only in certain areas. Had to identify bounding boxes
[3]	Used a CNN and OpenCV to identify bird species. Also utilized data augmentation.	Gives an outline for image recognition algorithm, and various ML techniques.	Testing accuracy was much lower than training, so their model was probably overfitting.
[4]	Used a Deep CNN and Tensorflow to identify bird species	Gives an overview of DCNN and using Tensorflow, and an accuracy benchmark	Their accuracy is low (80 percent), and they converted to grayscale.
[5]	Simplify image edge detection without using conventional methods (binary decision labels).	Introduced to potential methods (Histogram of Oriented Gradients and Support Vector Machine) to determine bird edge in images.	Not for bird species classification, only for bird image edge detection.

[6]	Compared different deep learning models on the classification of images of endemic bird species in Taiwan. Found that Inception-ResNet-V2 performed best.	Re-configured the model to include a 'swapping model' to achieve improved model results.	Image classification is limited to Taiwanese bird species only.
[7]	Describes theory and implementation of convolutional neural networks.	CNNs are a common and effective way to go about image classification.	Does not specifically address image classification of bird species.
[8]	Surveys the efficacy of different CNN approaches with regard to fine-grained object classification.	The CNN used will need to accomplish fine-grained object classification.	Trying every approach suggested would require a lot of time and processing power.
[9]	Demonstrates increased model accuracy when the pose of the bird is estimated beforehand.	May allow us to boost the accuracy of our model.	Implementation would be challenging.
[10]	Collects high-quality datasets containing 48,562 bird images with the help of crowd annotators. This outperformed Mechanical Turk.	High-quality images are important to measure the performance of image recognition.	Collecting data from crowd annotators may take a long time and effort to sort through them.
[11]	Auto-collected data using Raspberry Pi cameras and trained deep-learning CNN by separating bird images from the background.	An explanation of training CNN can be useful to understand the model deployment.	Trained with a limited dataset containing only three species of small birds, while our dataset contains 450 species.
[12]	Discussed deep learning pre-trained model VGG-19 of CNN, and PCA to extract features from bird images with various machine learning classifiers.	The added feature of PCA on layers 6 and 7 reduced the training and testing time and improved the accuracy of identifying birds.	The study was conducted with only 4340 images and may need more analysis to improve the results.
[13]	Implemented different image processing techniques and training an ANN learner	Great explanation on how different image processing techniques can be used	It is for recognition, not classification and it uses ANN while we intend to use CNN.
[14]	Identifying bird species using Deep learning algorithm (CNN) for 200 species and 11,788 samples.	The final model is connected to a user-friendly website that is similar to the interactive interface we will make.	The number of species is lower than the one we will be dealing with, and the data set is very small.
[15]	Using CNN model with Tensorflow back-end to classify birds images. The result is visually presented along with the prediction accuracy.	The methodology is very well-explained and contains the CNN architecture.	The experiments and testing are not discussed in depth as the methodology.

4 PROPOSED METHOD

4.1 INNOVATIONS

The innovations that have been incorporated into our project are as follows:

1. Utilizing transfer learning from a pre-trained base model.
2. Introducing noise into the training data to increase model generalization. This is accomplished both by pre-processing the data to include sheared, rotated, and zoomed-in versions of the images and by incorporating dropout layers into the model.
3. Utilizing Google Colab's "GPU hardware accelerator" option to increase speed.
4. Visualizing the results on a browser.

4.2 INTUITION AND DESCRIPTION

4.2.1 Model

Classification of bird species presents a fine-grained object classification problem since it aims to distinguish classes that simultaneously have high intra-class variance (training images of a single species of bird will consist of many different backgrounds, poses, and levels of noise) and low inter-class variance (different bird species resemble each other, sometimes to the extent that even a human cannot distinguish between them). [8]

By using a pre-trained ResNet50 model as the base layer of our model, we were able to build upon previous learning and achieve state-of-the-art accuracy (96.8% on unseen data) classifying 450 bird species. The intuition behind transfer learning is that knowledge gained while learning to recognize distinguishing features of objects would apply to a specific learning task such as bird classification.

Additionally, we introduced noise into the training data to increase model generalization and avoid overfitting. This was accomplished both by pre-processing the data to include sheared, rotated, and zoomed-in versions of the images and by incorporating dropout layers into the model, which add noise by temporarily removing units from the network throughout the training cycle.

In a Google Colab notebook, training, validation, and test datasets were downloaded using kaggle.com's API. The data set consisted of 450 bird species over 70,626 training images, 22500 test images(5 images per species), and 2250 validation images (5 images per species). All images had been resized to the same shape, 224 x 224 x 3. The ResNet50 base model was imported, and sheared, zoomed, rotated, and horizontally flipped versions of the images were made using Keras' ImageDataGenerator function. Shearing occurred over a range of 0.2, zoom over 0.15, and rotation over 5 degrees. The Google Colab notebook's runtime type was changed to GPU in order to increase the speed of training.

A Keras sequential class model was built as follows:

1. the pre-trained ResNet50 model
2. a flatten layer
3. a 2048-node deep dense layer (relu activation, he_normal kernel initializer)
4. a dropout layer (with a dropout rate of 0.2)
5. a 2048-node deep dense layer (relu activation, he_normal kernel initializer)
6. a dropout layer (with a dropout rate of 0.2)
7. a 450-node deep dense layer (softmax activation)

The layers in the base model were frozen (their parameters remained unchanged) for the first part of the training process. This was to avoid sacrificing any previous learning achieved by the pre-trained base model while training the upper layers.

Using Adam optimization with a learning rate of 0.0001, the model was trained until it achieved over 80 percent accuracy. This took three epochs (iterations) and resulted in an accuracy of 82.6%.

Thereafter, the base model's layers were unfrozen, and training continued with a reduced learning rate of 0.00005 until validation accuracy reached 96.0%. The learning rate was lowered in order to avoid convergence to a sub-optimal solution or introducing divergent behavior in the loss function. At this point, the model achieved 96.8% accuracy on the test set of images.

The entire training process took 13 epochs (iterations). Each epoch took approximately 830 seconds, resulting in a total training time of about 3 hours, less than what we expected considering the model had over 234 million parameters. The trained model was saved using Keras' model.save() function, resulting in a 2.6 GB file.

A function was made to take an input image, resize it to 224 x 224 x 3, and output the saved model's top three species predictions along with their corresponding model-assigned probability. Testing the model with various images of birds found online, we found that the model consistently gave correct predictions with high confidence.

4.2.2 Data from Wikipedia

Our goal in the visualization was to provide a summary of each bird, along with an example photo. This information is not provided in our Kaggle dataset. Therefore, we decided to create our own data set and collected the summary data and sample images from Wikipedia. We scraped Wikipedia for the short summary about each species and a web path for an example photo. This data was stored in CSV file and fed directly to our visualization script.

4.2.3 Visualization

The visualization of our project was done through the Python Flask environment. The trained TensorFlow model is loaded, and then Html scripts are called which ask for the user's input image. The team tested creating user interfaces using React.js and Flask. The React.js Html templates were more aesthetically pleasing and provided a cleaner user experience. We ended up taking the Html templates created in React.js and re-purposing the templates to work in Flask since the model and model outputs were already accessible in Flask. The result is an aesthetically pleasing web interface integrated with the TensorFlow model, providing the user with quick predictions.

As stated in our proposal, we provide three species 'guesses' each time an image is submitted. From our Wiki scraper, we show an example image of each species, along with a summary. Our model and visualization also yield the 'confidence' for each guess. This number can be slightly misleading, due to the limited number of species in the dataset. For example, if a picture of a species that's not in the dataset is submitted, the model will still provide the result for the species that most closely matches the image. Since all confidence values across the 450 species need to add up to 100 percent, the model could say there was a >90 percent confidence of being correct, even though it was very wrong.

The visualization also includes a button to go back to the upload stage to guess another image, or to the homepage.

4.3 RESULTS

The model achieved 96.8% accuracy on an unseen test set of images.

The entire training process took 13 epochs (iterations). The layers of the base model were unfrozen after the third epoch since the model had achieved 82.6% accuracy by that point. The remaining 10 epochs were trained with Adam optimization with a learning rate of 0.00005.
Each epoch took approximately 830 seconds, resulting in a total training time of about 3 hours, less than what we expected considering the model had over 234 million parameters. The trained model was saved using Keras' model.save() function, resulting in a 2.6 GB file.

5 EXPERIMENTS/EVALUATION

As mentioned in the Results Section above, we tested the image recognition model on novel bird images. The model performed very well and achieved 96.8% accuracy on our test images. Of course, the model is limited to identifying the 450 bird species that were included in the training images. However, given that many existing bird identification applications are only able to identify a portion of those that are covered by our model, we consider our project to be successful in addressing our first problem definition (building a model that takes a bird image and accurately identifies the bird species). On top of accurately identifying the bird species, we are able to provide the user with some quick facts that they may find useful about the bird species and a link to the bird species Wikipedia page.

6 CONCLUSION AND DISCUSSION

In this project, we successfully deployed an easy and accessible bird identification tool using a pre-trained deep-learning model. The database of this project contains over 70,000 images of 450 bird species obtained from Kaggle. The CNN model correctly predicted the test set of bird images with an accuracy of 96.8%. Overall, our model has performed very well given a large training data set. Thus, this model along with the intuitive and simple-to-use interface, and added description of bird species with a confidence level can be a practical tool for anyone interested in observing and studying birds.

All team members contributed a similar amount of effort.

Bibliography

[1] C.R Raquel, K.M.A Alarcon, and L.L Figueroa. *Image classification of Philippine bird species using deep learning*. Taylor & Francis, London, U.K., 2019.

[2] Saloni Saxena, Sneh Thorat, Prachi Jain, Rupal Mohanty, and Trupti Baraskar. Analysis of object detection techniques for bird species identification. In *International Conference on Electronic Circuits and Signalling Technologies*, 2022.

[3] Anisha Singh, Akarshita Jain, and Bipin Kumar Rai. Image based bird species identification. *International Journal of Research in Engineering, IT and Social Sciences*, 10(4):17–243, 2020.

[4] Pralhad Gavali, Prachi Abhijeet Mhetre, Neha Chandrakhant Patil, Nikita Suresh Bamane, and Harshal Dipak Buva. Bird species identification using deep learning. *International Journal of Research & Technology*, 8(4), 2019.

[5] Ajay Kumar Rahul Kumar and Arnav Bhavsar. Bird region detection in images with multi-scale hog features and svm scoring. In *Proc. of 2nd International Conference on Computer Vision and Image Processing*, Advances in Intelligent Systems and Computing, pages 353–364, Singapore, 2018. Springer.

[6] Yo-Ping Huang and Haobijam Basanta. Recognition of endemic bird species using deep learning models. *IEEE Access*, 9:102975–102984, 2021.

[7] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc., 2022.

[8] Bo Zhao et al. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.

[9] Steve Branson et al. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint*, 1406(2952), 2014.

[10] Grant Van Horn et al. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] André C. Ferreira et al. Deep learning-based methods for individual recognition in small birds. *Methods in Ecology and Evolution*, 11(9):1072–1085, 07 2020.

[12] Suleyman Al-Showarah and Sohyb Qbailat. Birds identification system using deep learning. *International Journal of Advanced Computer Science and Applications*, 12:2021, 05 2021.

[13] Uma D Nadimpalli, Randy R Price, Steven G Hall, and Pallavi Bomma. A comparison of image processing techniques for bird recognition. *Biotechnology progress*, 22(1):9–13, 2006.

[14] Pralhad Gavali and J.Saira Banu. Bird species identification using deep learning on gpu platform. *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020.

[15] Satyam Raj, Saiaditya Garyali, Sanu Kumar, and Sushila Shidnal. Image based bird species identification using convolutional neural network, Jun 2020.