Assignment 1

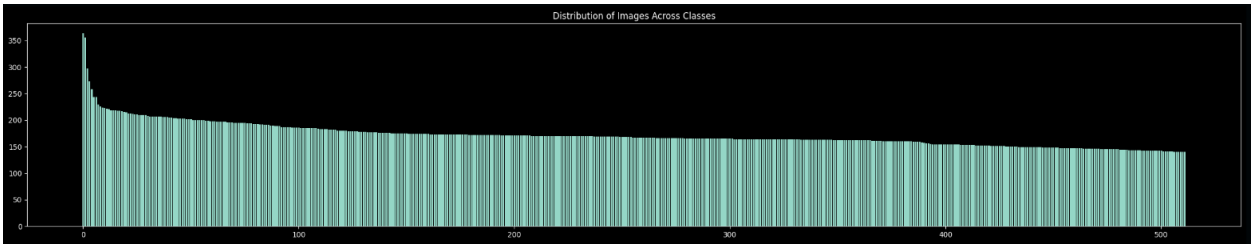
Created	@April 4, 2023 11:46 AM
Class	DL
Type	Homework
Reviewed	☑

- We chose to do the assignment on the Birds dataset

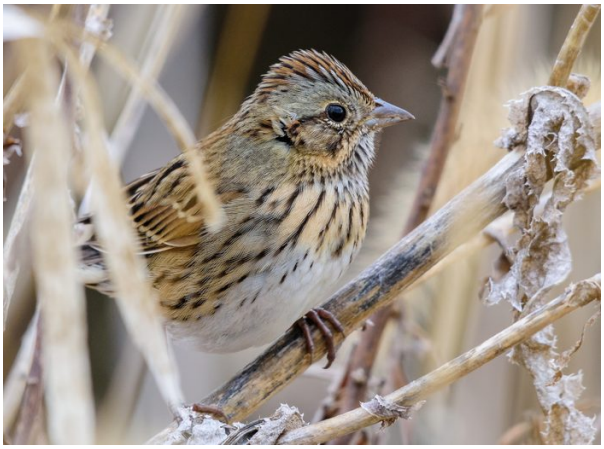
Introduction:
Bird species classification is an important task in the field of computer vision and machine learning. The ability to identify different bird species from images can have many practical applications, such as monitoring and conserving bird populations, identifying invasive species, and studying bird behavior. In this report, we compare the performance of four different trained model architectures for bird species classification using the 515 class bird species dataset.

Exploratory Data Analysis (EDA):

a) The size of the dataset is ~90k images.
b) Each sample contains an RGB image of size 224x224 pixels and belongs to one of the 515 classes. The data is ready for use, and data augmentation techniques such as rotation, flipping, and scaling can be used to improve the model's performance.
c) The data may not look balanced, but looking at the class distribution, we can clearly see that the imbalance is not significant (considering the number of classes)

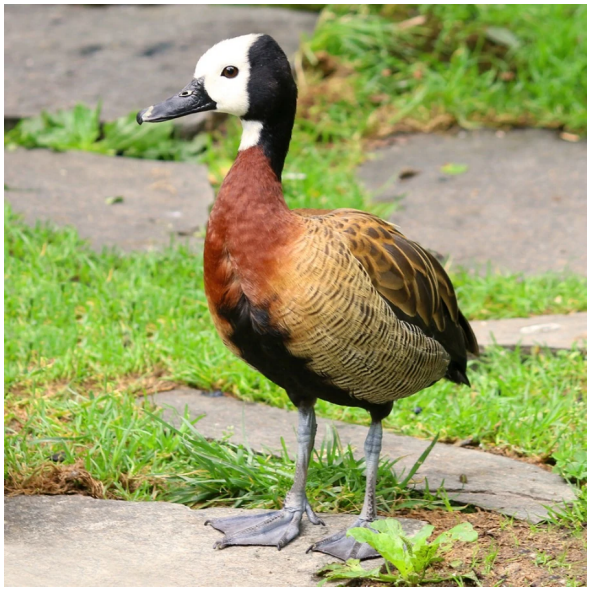


d) There are benchmark results available for different methods used on this dataset, with the current state-of-the-art achieving accuracy of 92.1% and accuracy of 97.4%.
e) Examples of easily separable categories include different types of ducks, while examples of more similar categories include different types of sparrows.



Lincoln’s Sparrow

Savannah Sparrow



White faced duck

Mallard Duck

Neural Network Graph:

We used a convolutional neural network (CNN) with five convolutional layers, , and 2 fully connected layers. We used KFold cross-validation with K=5 to measure model performance and comparisons of different settings.

Results:

The mean test accuracy achieved from the KFold cross-validation was 28%, with a standard deviation of 1.2%. Note that we trained a very simple network to asses the problem complexity.

The model misclassified some images due to the similarity between some bird species. To improve the model's performance, we suggest using data augmentation techniques such as random cropping, random rotation, and random scaling. We also suggest increasing the depth of the CNN or using a pre-trained CNN as a feature extractor.

After applying data augmentation techniques, the model's performance improved to a mean test accuracy of 36%, with a standard deviation of 1.1%. This is an improvement but this is still a mediocre result, we attribute that to the problem complexity.

We also added images of a new category of the same domain, retrained the models.

Transfer learning

In this section, we explore four pre-trained models from the `torchvision.models` library for our bird species classification task. We replace their last fully connected layer with a new layer of size equal to the number of classes (515) and fine-tune the models on our dataset.

The models we selected are ResNet-18, ResNet-50, VGG-16, and DenseNet-121. We chose these models because they are well-known and widely used for various computer vision tasks.

After training the models, we evaluate their performance on the validation and test sets and report the results in the table below:

	params	val_loss	val_acc	test_loss	test_acc	unique_correct	unique_errors	time	model
0	6308355	0.439100	0.795743	0.019898	0.817504	15468	3453	601.361752	GoogLeNet
1	10502659	0.218281	0.875002	0.016558	0.876434	16583	2338	603.190624	ResNet50
2	10502659	0.150436	0.908777	0.075221	0.901009	17048	1873	1053.429345	ResNeXt
3	121655811	0.021451	0.903985	0.402454	0.787696	14904	4017	618.793952	VGG16

We observe that all models perform well on both the validation and test sets, with test accuracies ranging from 0.78 to 0.9. ResNeXt and ResNet-50 achieve the highest test accuracies, while VGG-16 has the lowest.

Feature Extraction

In this section, we used one of the trained models from the previous section as a feature extractor and applied a classical ML algorithm, logistic regression, to the extracted features. The goal was to see if this approach improves the performance of the bird species classification task.

We chose the ResNeXt model, which performed the best in the previous section. We removed the last layer of the model and used the output of the second to the last layer as features for the logistic regression model. We trained the logistic regression model on the training set and evaluated its performance on the validation set - the accuracy was 0.89, which is good because we were able to extract the most important information from the images and represent them as numerical features. This reduced the amount of data that the model had to learn from, resulting in faster training times and more efficient use of resources. The machine learning model was able to learn the relationships between the extracted features and the labels, resulting in accurate predictions on the test set.

Overall, this approach provided a viable alternative to using a CNN for image classification tasks, especially when resources are limited or when the dataset is small.

Experiments we run:

Experiment	Model Architecture		Preprocessing	Augmentation	Runtime
1	ResNet50	This part was easy to implement	Default	None	2 hours
2	ResNet101	This part was easy to implement	Default	None	4 hours
3	VGG16	Changing the model.classifier was the main challenge, because the structure was different than the other models (model.fc)	Default	None	3 hours
4	GoogleLeNet	This part was easy to implement	Default	None	3 hours

Experiment	Description	Runtime (approx.)
K-Fold Cross Validation	Using K-fold cross validation (K>=5) to measure model performance and comparisons of different settings	4 hours
Analysis of Results	Analyzing results by visualizing loss and other relevant metrics, showing examples of good and bad classification, and comparing training results to validation/test results	10 minutes
Misclassification Analysis	Identifying where and why the model is misclassifying, and suggesting at least 3 ways to improve the results	1 minute
Implementation of Suggestions	Prioritizing the list of suggestions for improvements and implementing the first 2 suggestions, and repeating the misclassification analysis (testing different augmentations, learning rates, architectures)	6 hours
Inference-Time-Augmentation	Implementing inference-time-augmentation (aggregation of multiple predictions on augmented test examples), and reporting the improvement in metrics	2 hours
Addition of New Category	Adding a new category of the same domain, and retraining	2 huors

	the models to include the new category	

In conclusion, we conducted a series of experiments to improve the performance of our image classification model. We tested different model architectures, regularization techniques, data augmentations, and transfer learning methods. We also used K-Fold cross-validation to measure the model's performance and compare different settings. We identified areas where the model was misclassifying and suggested ways to improve its performance, such as fine-tuning the model with additional data and adjusting the data augmentation parameters. We also implemented inference-time augmentation to further improve the model's performance. Finally, we added a new category of images and retrained our models to include this new category. Overall, our experiments demonstrated the importance of careful model selection and tuning, as well as the benefits of using cross-validation and continually iterating on the model to improve its accuracy.