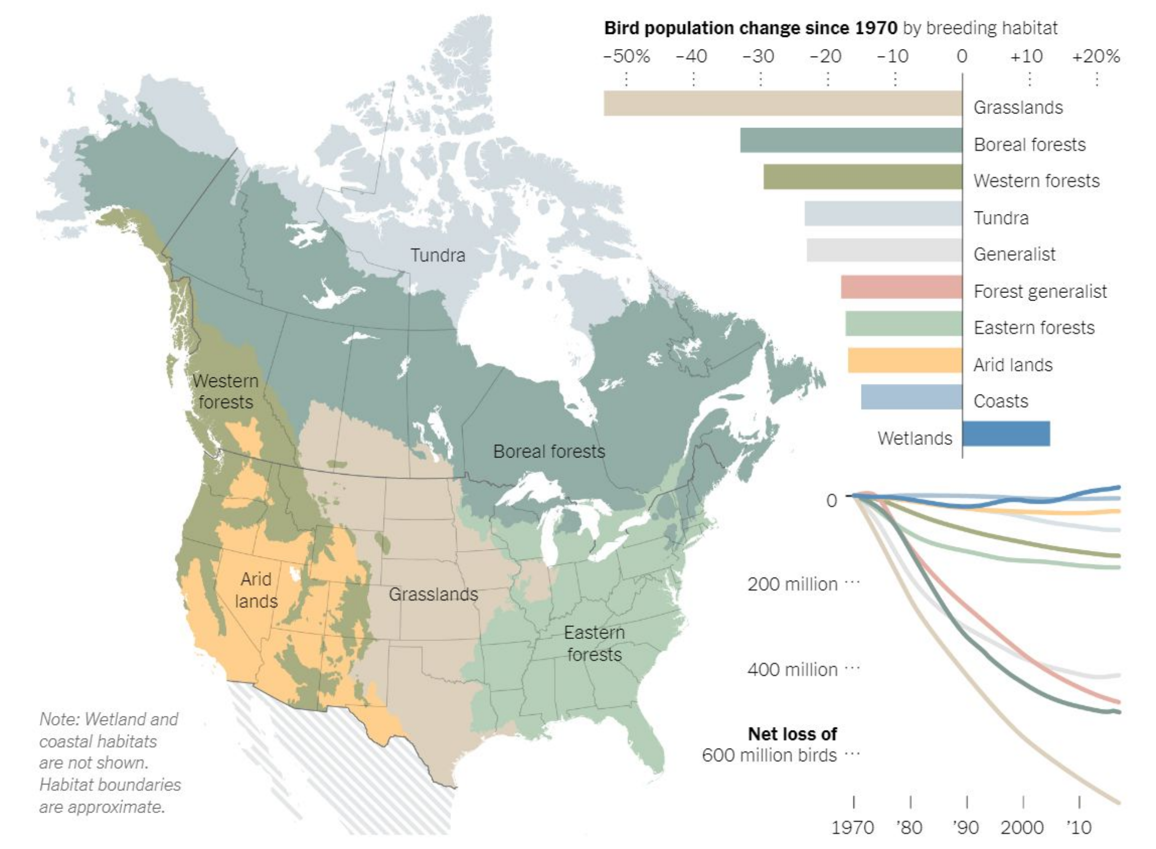
Duck Duck Choose:

Pecking at Pixels with Machine Learning

Sprint 2

By Larissa Huang

Decline in bird populations since 1970



Dataset

Birds 525 SPECIES - Kaggle

- 84,635 training images
- 2,625 test images
- 2,625 validation images

across 525 bird species.

*images sourced from Google

Data dictionary

labels: bird species associated with the image file

scientific label: scientific name for the bird species

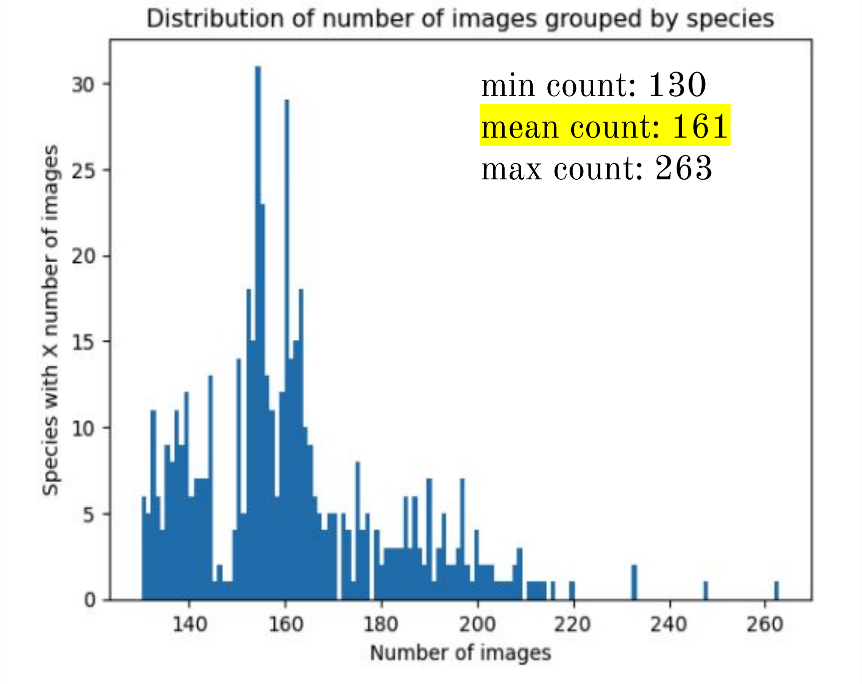
filepaths: the relative file path to an image file

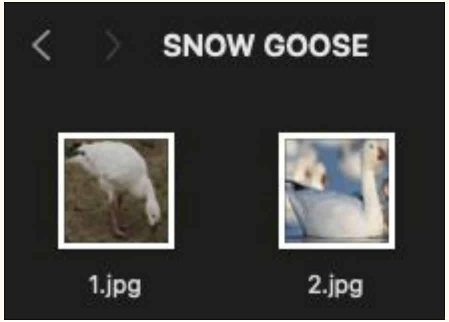
data set: which dataset (train, test or valid) the image filepath belongs to

class_id: the class index value associated with the image file's class

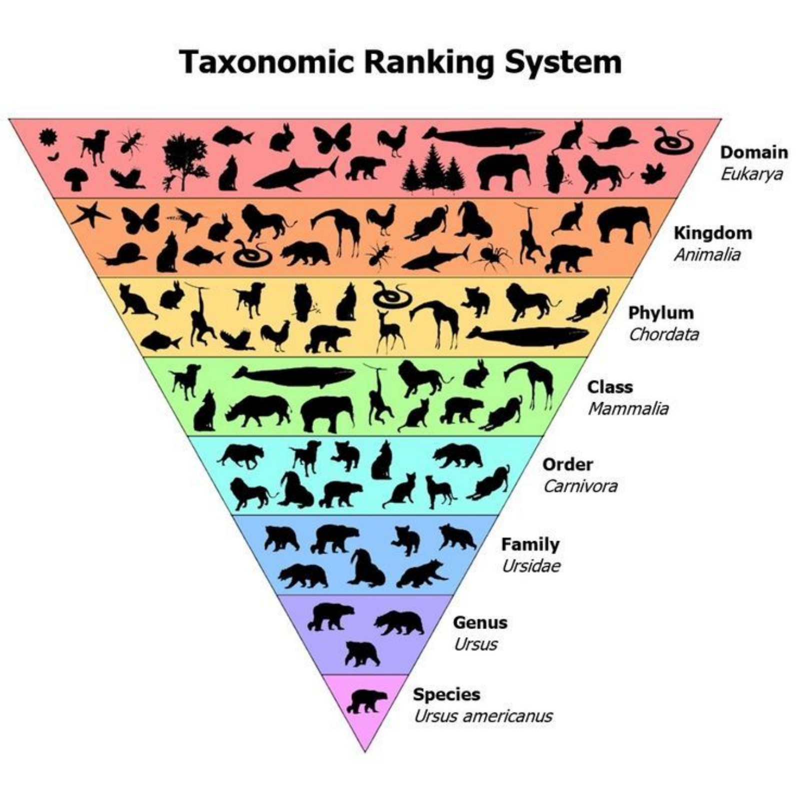
	class id	filepaths	labels	data set	scientific name
0	0.0	train/ABBOTTS BABBLER/001.jpg	ABBOTTS BABBLER	train	MALACOCINCLA ABBOTTI
1	0.0	train/ABBOTTS BABBLER/007.jpg	ABBOTTS BABBLER	train	MALACOCINCLA ABBOTTI
2	0.0	train/ABBOTTS BABBLER/008.jpg	ABBOTTS BABBLER	train	MALACOCINCLA ABBOTTI
3	0.0	train/ABBOTTS BABBLER/009.jpg	ABBOTTS BABBLER	train	MALACOCINCLA ABBOTTI
4	0.0	train/ABBOTTS BABBLER/002.jpg	ABBOTTS BABBLER	train	MALACOCINCLA ABBOTTI

<u>Concern:</u> not enough data, many similar species





Solution: merge multiple species into one genus



How? By getting the last word of a species name

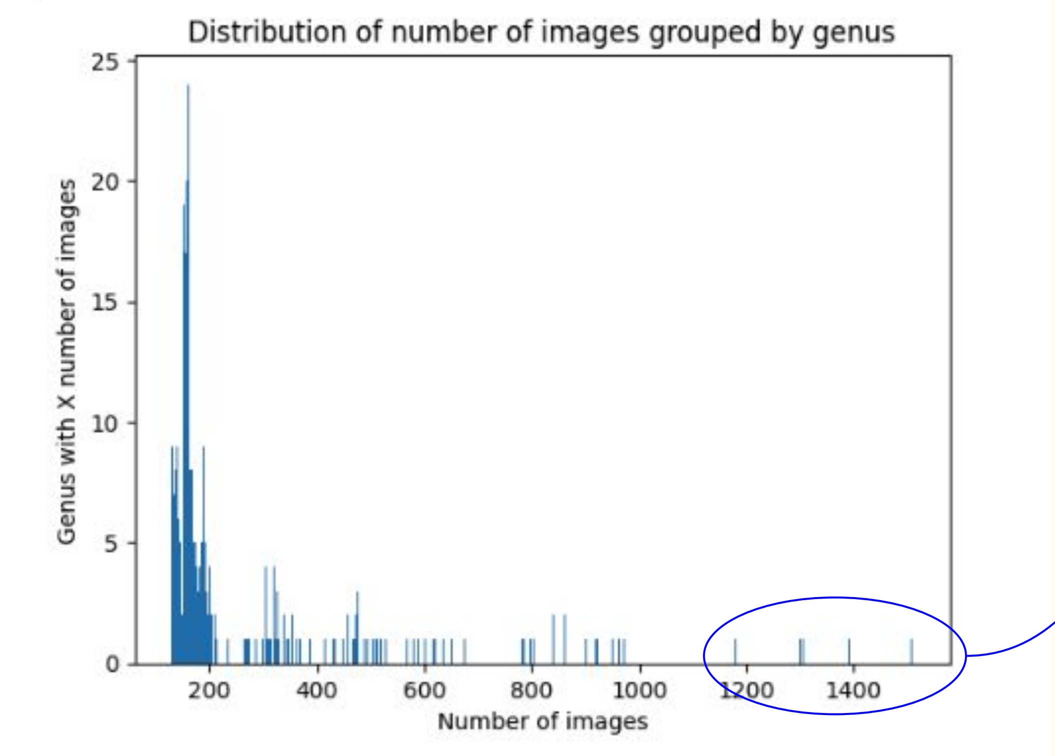
Original folder structure

Unique instances of each last word (312 in total)

0	BABBLER
1	BOOBY
2	HORNBILL
3	CRANE
4	CUCKOO
5	FIREFINCH
6	CATCHER
7	GOOSE
8	ALBATROSS
9	TOWHEE
10	PARAKEET
11	CHOUGH
12	YELLOWTHROAT
13	AVOCET
14	BITTERN
15	COOT
16	FLAMINGO
17	GOLDFINCH
18	KESTREL

The vast majority of species follow this convention and I'm only using the top genera

Checking distribution of images by genus



We can keep the genera with the highest image counts:

	genus	count
91	DUCK	1510
298	WARBLER	1391
217	PHEASANT	1303
161	KINGFISHER	1298
93	EAGLE	1179
102	FINCH	970
123	GOOSE	962
41	BUNTING	952
204	OWL	923
270	TANAGER	921

Preprocessing Steps

- **Array encoding:**
  encode images as arrays representing image data using cv2.imread()

- **Label encoding:**
  {'DUCK': 0, 'EAGLE': 1, 'KINGFISHER': 2, 'PHEASANT': 3, 'WARBLER': 4}

- **Reshaping image arrays:**
  CNN model expects a 4D array

```
Length of train images array: 6681
X_train_images shape: (6681, 224, 224, 3)

Length of valid images array: 205
X_valid_images shape: (205, 224, 224, 3)

Length of test images array: 205
X_test_images shape: (205, 224, 224, 3)
```

Visualize classes - diverse appearances

1.

2.

CNN Base Model performance

Accuracy score of 30% on test data (batch_size =128, epochs = 50)

While this is not good, it's higher than random chance, which would have been 20% with 5 categories.

In the following steps, I will seek to improve this.

Model Limitations

- too simple

- not enough images

- image quality concerns

- class imbalance

DUCK	22.601407
WARBLER	20.820236
PHEASANT	19.503068
KINGFISHER	19.428229
EAGLE	17.647059

Next steps

- Data Augmentation

- Denoising and other image preprocessing steps such as cropping, grayscalling, intensity thresholds, edge detection, colour filters

- Implement Transfer Learning using a pre-trained CNN like EfficientNet