Project Write Up: Dog Breed Reveal

By Melanie Ackerman

Abstract:

Can you tell a dog's breed just by looking at them? With so many breeds out there, it can even be hard to pinpoint the breed of a pure-bred dog, let alone mutts. Wouldn't it be fun to snap a picture of a dog and immediately know it's breed and typical traits? Such a tool would be especially helpful to dog adoption organizations (or a regular dog enthusiast who likes meeting dogs off the street, like me!). In this project I use over 10,000 dog photos spanning 120 breeds to create a classifier to determine a dog's breed from an image. During this process I define three CNN models that contain the same layers, but have different numbers of classes. I first define a 12-breed classifier, then increase to 30 breeds, and finally all 120 breeds. Unfortunately, due to time and processing constraints, the 120-breed model could not be finished. However, the 30-breed model performed well and is the focus of the project.

Design:

This project takes images of dogs and classifies them by breed using a convolutional neural network. The dataset is from a Kaggle competition found at https://www.kaggle.com/competitions/dog-breed-identification/data. The dataset contains over 10,000 unique photos of dogs spanning 120 breeds. I took samples from this larger dataset so as to only include 12 breeds, and later 30 breeds, balanced the classes for the 30-breed model, and followed an image augmentation workflow to create the CNN model. I took advantage of transfer learning using ResNet50V2 layers, then added global average pooling and batch normalization layers before the fully connected layer. Finally, I evaluated the model with the classification metrics report from the sklearn library.

Data Description:

The dataset is from a Kaggle competition found at https://www.kaggle.com/competitions/dog-breed-identification/data. The dataset contains over 10,000 unique photos of dogs spanning 120 breeds.

Algorithms:

I split the data 80/20 into train and test sets, and further split the train set 75/25 into train and validation sets. The test set was not used until the very end. I use image augmentation and resize images to 150x150 when creating image generators and Resnet50V2 layers for transfer learning. I add global average pooling and batch normalization layers before the fully connected dense layer. The 12- and 30-breed models have dropout set to .5, while the 120-breed model sets dropout to .3 in the global average pooling layer and to .2 in the fully connected layer. Though this final model could not finish running, training accuracy was converging around 70%, compared to around 80% in the other models.

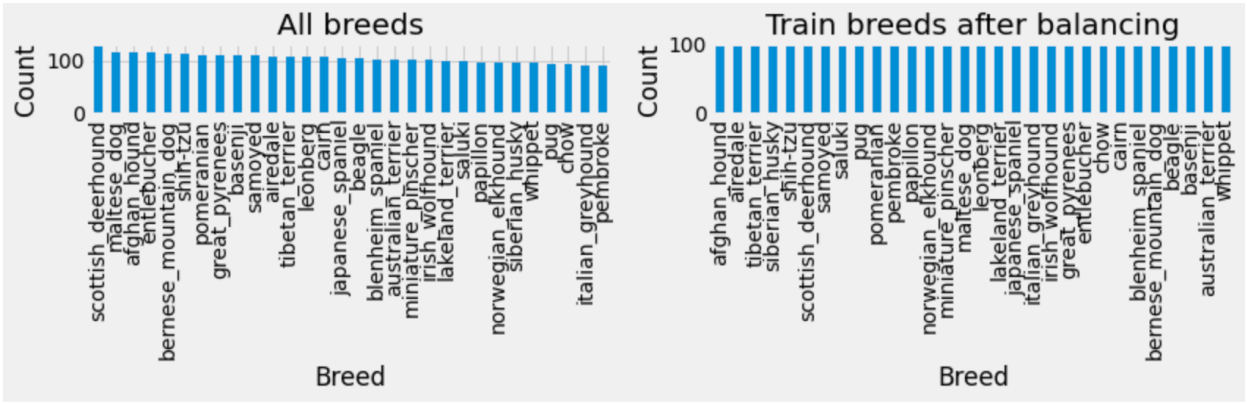
The 30-breed model evaluation on test data is pictured at the bottom of this document.

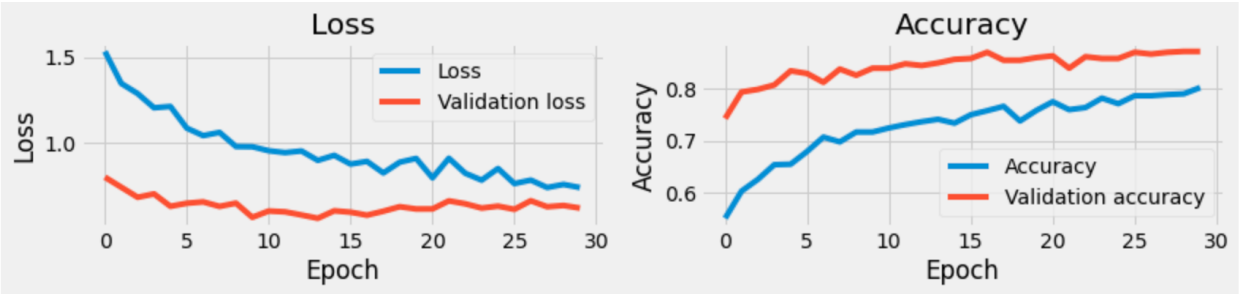
Tools:

NumPy and Pandas for data manipulation
Tensorflow/Keras for modeling
Sklearn for evaluation
Matplotlib for visualization

Communication:

I produced slides containing the following visuals:





	precision	recall	f1-score	support
afghan_hound	0.74	0.81	0.78	32
airedale	0.78	0.69	0.73	26
australian_terrier	0.79	0.76	0.78	25
basenji	0.60	0.76	0.67	33
beagle	0.82	0.77	0.79	30
bernese_mountain_dog	0.96	0.86	0.91	28
blenheim_spaniel	0.79	0.73	0.76	26
cairn	0.80	0.77	0.78	26
chow	0.80	0.89	0.84	27
entlebucher	0.80	0.95	0.87	21
great_pyrenees	0.61	0.65	0.63	26
irish_wolfhound	0.38	0.29	0.32	21
italian_greyhound	0.52	0.50	0.51	30
japanese_spaniel	0.70	0.81	0.75	26
lakeland_terrier	0.47	0.50	0.48	16
leonberg	0.67	0.89	0.76	27
maltese_dog	0.81	0.59	0.68	29
miniature_pinscher	0.72	0.84	0.78	25
norwegian_elkhound	0.78	0.70	0.74	20
papillon	0.81	0.85	0.83	26
pembroke	0.91	0.80	0.85	25
pomeranian	0.80	0.80	0.80	30
pug	0.62	0.69	0.65	26
saluki	0.74	0.52	0.61	33
samoyed	0.84	0.84	0.84	25
scottish_deerhound	0.78	0.70	0.74	30
shih-tzu	0.50	0.33	0.40	27
siberian_husky	0.71	0.62	0.67	24
tibetan_terrier	0.49	0.74	0.59	27
whippet	0.50	0.47	0.48	17
accuracy			0.71	784
macro avg	0.71	0.70	0.70	784
weighted avg	0.71	0.71	0.71	784