Dog Breed Classification

**Team: Lyn Wang, Jenny Park,
Rodney Tang, Ryan Castillo,**

Research Question & Motivation

Why

- Dog breed recognition is important for vets & animal shelters
- Even experts struggle with visually similar dog breeds

Goal

- Build an ML model that can identify dog breeds from images

Related Work

Early Works [Traditional Models]

- Introduced the Stanford Dogs dataset
- Used deformable part models and bounding boxes to locate features (ears, snouts)
- Relied on handcrafted features
- Didn't generalize well to messy real-world images

Deep Learning [CNNs & Transfer Learning]

- Used CNNs and various forms of transfer learning
- Improved accuracy vs. handcrafted methods
- Helped with limited data
- Struggled with overfitting

Recent Advances [Vision Transformers]

- Introduced ViTs (Vision Transformers)
- Treated images as patch sequences
- Outperformed CNNs when pretrained on large datasets
- Very resource-intensive

Our Contribution

- Test the different types of methodologies through these eras
- Increase complexity over time to understand the benefits of each method
- Explore weighting generalization vs capacity trade-offs

Datasource: Stanford Dogs Dataset

- 20,580 examples, pre-split into 12,000 training and 8,580 test gathered from ImageNet
- 64x64 RGB images
- 120 total breeds (classes)
- Annotation Data Included

EDA - Class Distribution

Training Set

	label	count
0	Chihuahua	100
1	Japanese_spaniel	100
88	Maltese_dog	100
87	Pekinese	100
86	Shih	100
...	...	...
35	Mexican_hairless	100
34	dingo	100
33	dhole	100
32	Samoyed	100
119	African_hunting_dog	100

120 rows × 2 columns

100 examples per class

Test Set

	label	count
0	Maltese_dog	152
1	Afghan_hound	139
2	Scottish_deerhound	132
3	Pomeranian	119
4	Bernese_mountain_dog	118
...	...	...
108	Doberman	50
106	Welsh_springer_spaniel	50
105	clumber	50
118	Pekinese	49
119	redbone	48

120 rows × 2 columns

~ 50-150 examples per class

EDA - Annotation Data

Before:

After:

EDA - Data Quality

Rhodesian_ridgeback

Leonberg

boxer

Bedlington_terrier

Data Preparation Steps

- Annotation Data Mask
- Stratified random split on test set → validation set
- Prepped labels for consistent formatting and integer labels
- Image Augmentation*

Models

Multiclass Logistic Regression

Model

- Tuned learning rates, batch sizes, optimizers, and initializer settings
- Trained using sparse categorical cross-entropy loss
- Benchmark (Majority Class): 1.77% accuracy

Key Challenge: Underfitting

- Model could not learn from added complexity from data augmentation

Results

Train Accuracy	Val Accuracy	Overfit Gap	Test Accuracy	F1-Score (Weighted-Avg)
8.4%	3.7%	4.7%	3.5%	**0.03**

Fully Connected NN

Model

- Same loss optimization, but applies hidden layers to logistic regression
- Hyperparameter Tuning applied to identify optimal parameters

Key Challenge: Overfitting

1. Data augmentation
2. Early stopping based on validation loss

Results

Train Accuracy	Val Accuracy	Overfit Gap	Test Accuracy	F1-Score (Weighted-Avg)
9.23%	6.97%	2.26%	7.09%	**0.051**

Convolutional Neural Network (CNN)

Model

- Same loss function, custom CNN architecture
- “convolution + max pooling + dropout” block
- Initially tuned with keras, and then manually tuned via grid search for the full model (13 hyperparameters)

Key Challenge 1: Overfitting + Model instability

1. Regularization: L1 + L2 + dropout + batch normalization
2. Applied random augmentation to each epoch

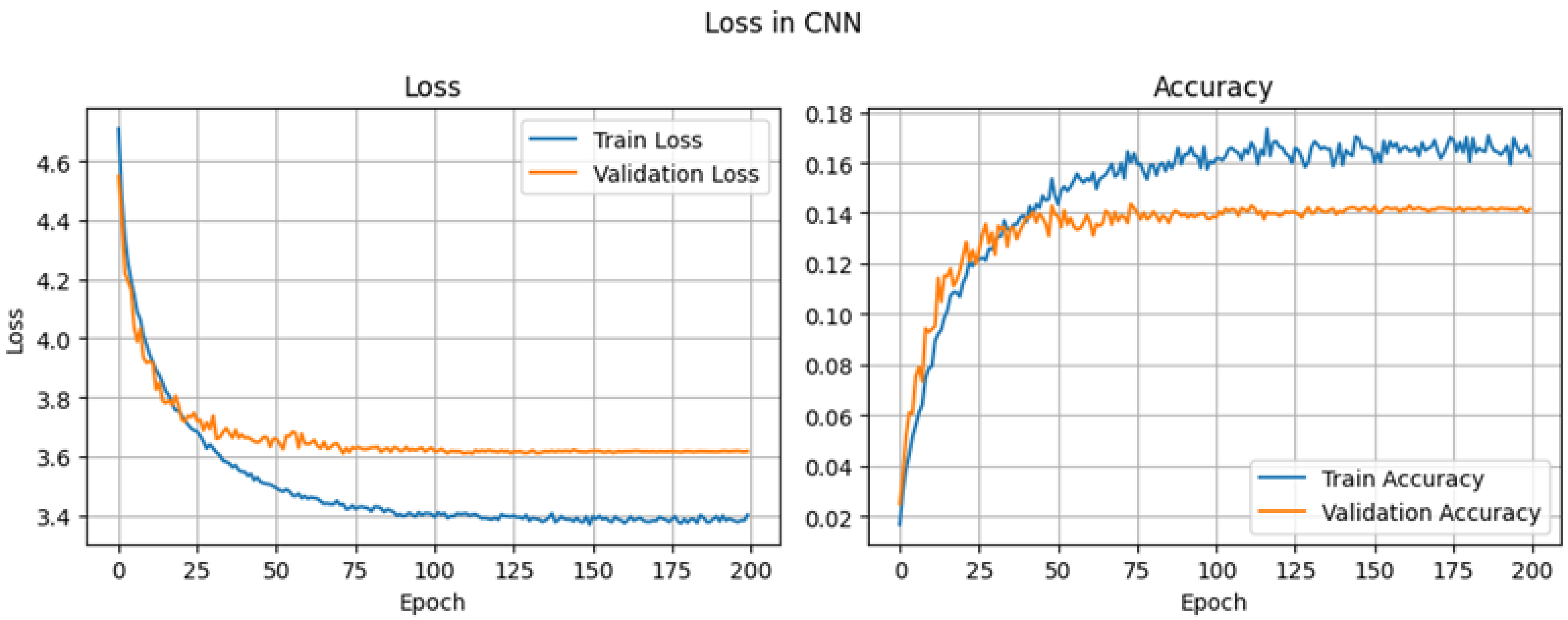
Train val gap is narrowed to within 10 ppts

Key Challenge 2: Scalability to 120-breed

1. Reduce regularization
2. Reduce learning-rate decay and increase batch size

Convolutional Neural Network (CNN)

Model



Results

Train Accuracy	Val Accuracy	Overfit Gap	Test Accuracy	F1-Score (Weighted-Avg)
15.60%	13.66%	1.94%	16.67%	0.13

Transfer Learning (EfficientNet B0)

- Evaluated many pre-trained models: ResNet50, MobileNetV2 EfficientNetB0
- Custom classification head: GlobalAvgPool → BatchNorm → Dense layers
- Fine-tuning: All layers trainable with lower learning rate
- Smart regularization: L2 + Dropout + BatchNorm at each dense layer

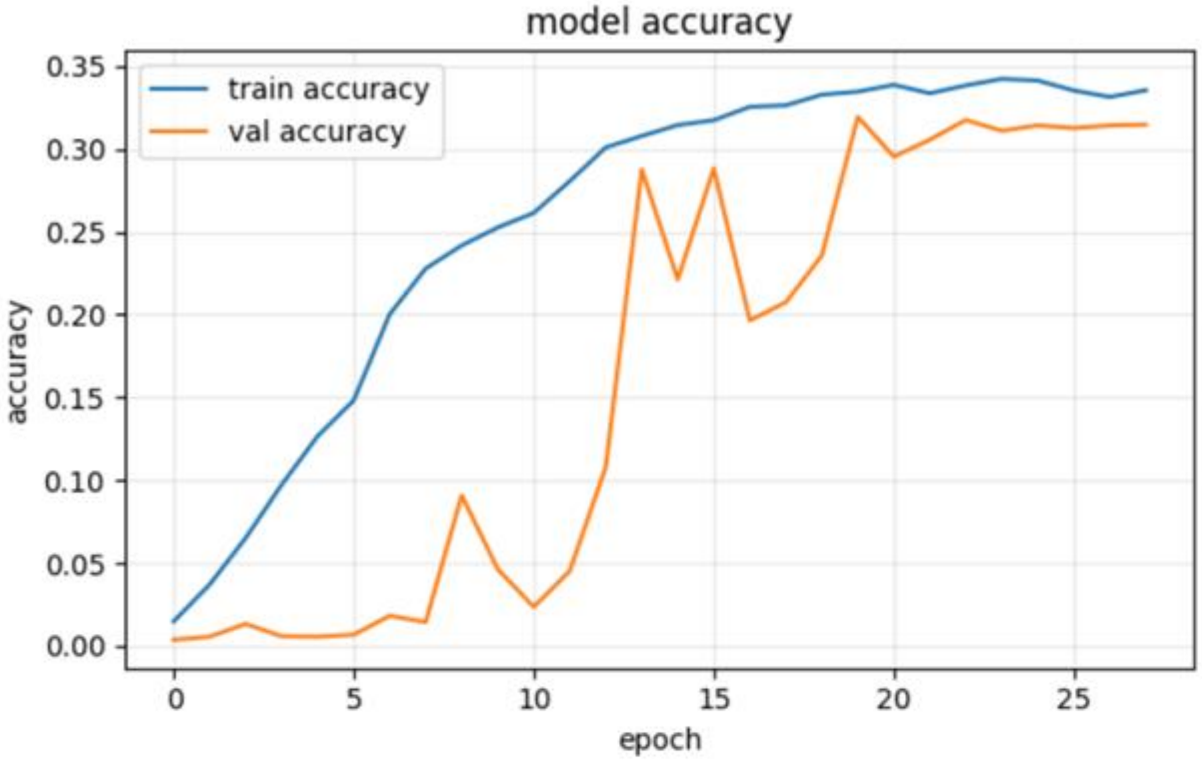
Key Challenge 1: Backbone Selection

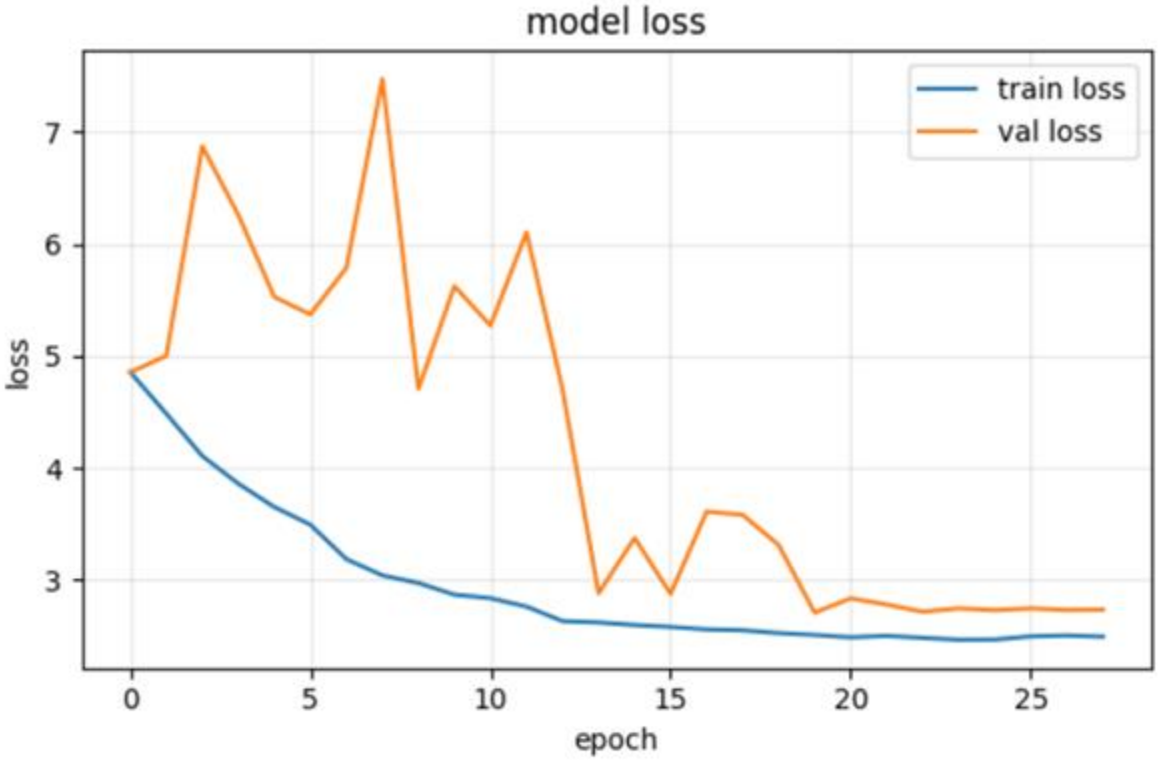
1. ResNet50: Good accuracy but computationally expensive
2. MobileNetV2: Fast but lower feature quality for fine-grained classification
3. EfficientNetB0: Best balance - compound scaling design optimized for efficiency

Key Challenge 2: Overfitting

- Applied various data augmentations
- Multi-layer regularization w/ L2 penalties + dropout + batch normalization
- Early stopping and learning rate reduction based on validation metrics

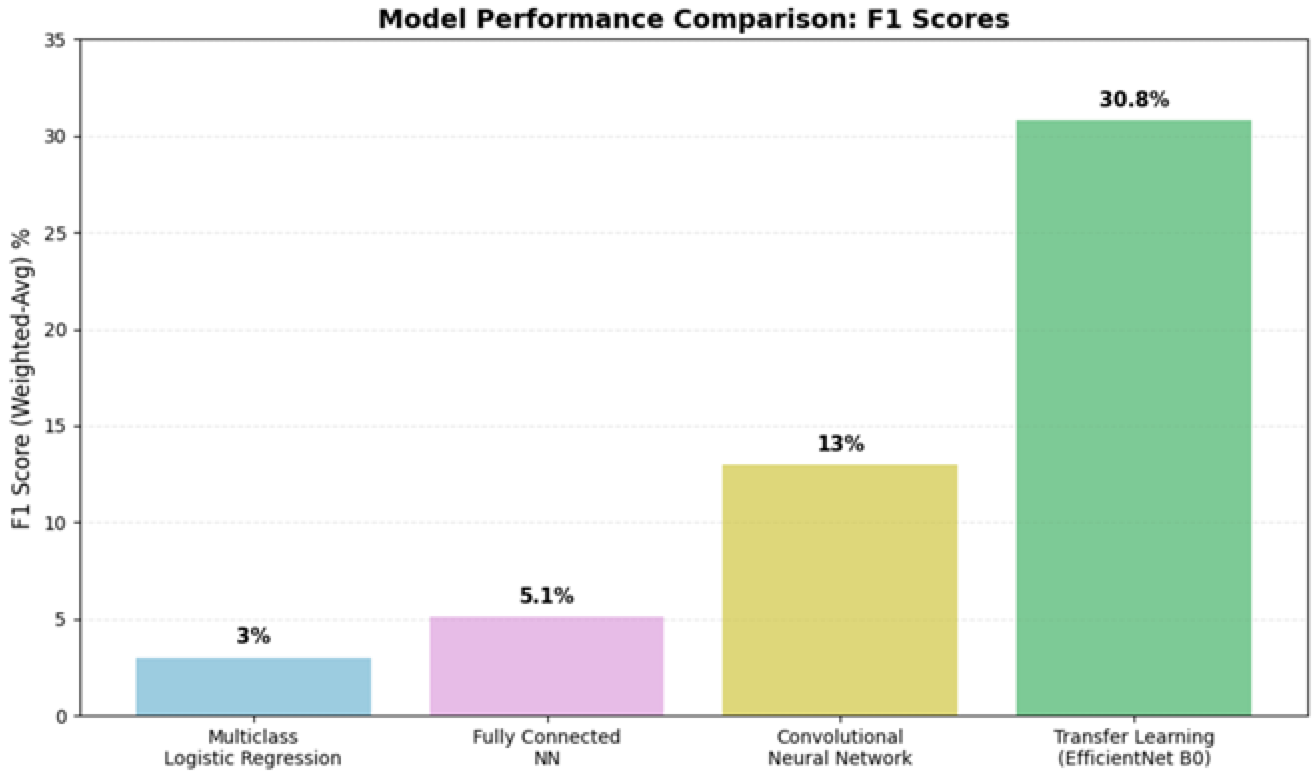
Transfer Learning (EfficientNet B0)





Train Accuracy	Val Accuracy	Overfit Gap	Test Accuracy	F1-Score (Weighted-Avg)
33.5%	31.4%	2.1%	31.5%	0.31

Results & Discussion - Model Comparison



Model	Train Acc	Val Acc	Test Acc	Overfit Gap	F1-Score
Logistic Regression	8.4%	3.7%	3.5%	4.7%	3%
Fully Connected NN	9.23%	6.97%	7.09%	2.26%	5.10%
CNN	15.60%	13.66%	16.67%	1.94%	13%
Transfer Learning	33.5%	31.4%	31.5%	2.1%	30.8%

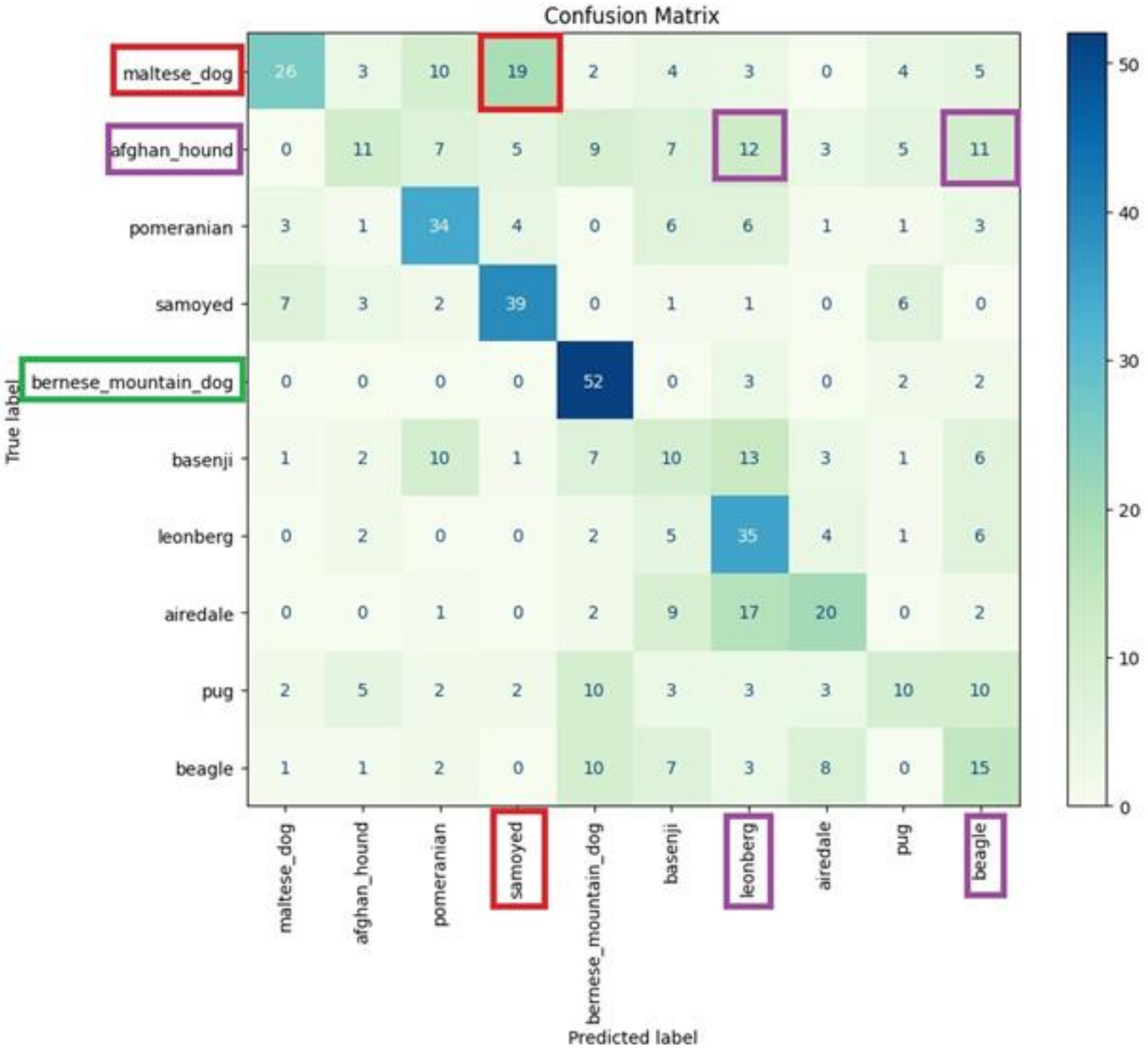
Key Insights

- 10x Performance Jump: Transfer learning achieved 30.8% F1-score vs a 3% baseline
- Overfitting Control: All models maintained <5% train-val gap through proper regularization
- Consistent Generalization: Transfer learning shows tight clustering (31-33%) across train/val/test
- Architecture Matters: CNN (13%) vs NN (5.1%) demonstrates importance of spatial feature learning

Model Evolution

- **Logistic Regression**: Pixel-level classification, insufficient for complex visual patterns
- **Neural Network**: Added non-linearity, modest improvement with better generalization
- **CNN**: Spatial feature extraction, major breakthrough for image classification
- **Transfer Learning**: Leveraged ImageNet knowledge, optimal accuracy-efficiency balance

Results & Discussion - Subgroup Analysis



Conclusion & Discussion

Overall Performance

- Overall, Transfer Learning delivered the best results: 29% F1 and accuracy of 31% (30x improvement over majority class)
- CNN was better able to capture complex spatial hierarchies within the data
- Challenges to the task: limited sample size, large output space (120 distinct breeds)

Opportunities:

1. Reduce class granularity by grouping visually similar breeds
2. Enhance data quality and diversity
3. Vision Transformers

Thank You!

Jenny

Predicted Label:
Toy Poodle

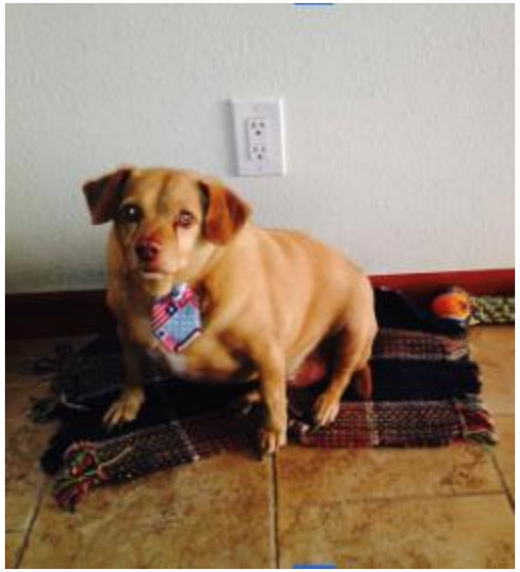
True Label:
mini_schnauzer

Lyn

Predicted Label:
Airedale Terrier

True Label:
mini_schnauzer

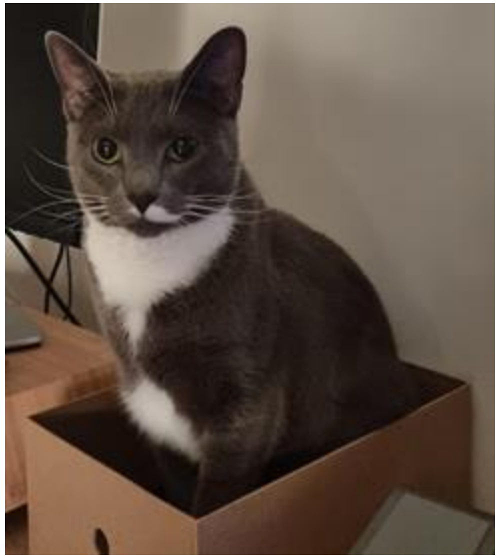
Rodney



Predicted Label:
Boxer

True Label:
dachshund

Ryan



Predicted Label:
Sealyham Terrier

True Label:
cat

Appendix

CNN architecture

Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 64, 64, 3)	0
conv2d (Conv2D)	(32, 64, 64, 8)	1,544
max_pooling2d (MaxPooling2D)	?	0
batch_normalization (BatchNormalization)	(32, 32, 32, 8)	32
conv2d_1 (Conv2D)	(32, 32, 32, 16)	2,064
max_pooling2d_1 (MaxPooling2D)	?	0
batch_normalization_1 (BatchNormalization)	(32, 16, 16, 16)	64
conv2d_2 (Conv2D)	(32, 16, 16, 16)	1,040
conv2d_3 (Conv2D)	(32, 16, 16, 16)	1,040
max_pooling2d_2 (MaxPooling2D)	?	0
flatten (Flatten)	(32, 1024)	0
dense (Dense)	(32, 128)	131,200
dense_1 (Dense)	(32, 128)	16,512
dropout (Dropout)	?	0
dense_2 (Dense)	(32, 120)	15,480

Appendix

Subgroup evaluation / F1 Score

Top accuracy: Fully Connected NN(top left) vs CNN(bottom left)

Breed	F1 Score	Precision	Recall	Support
sealyham_terrier	0.340659	0.236641	0.607843	51.0
entlebucher	0.171717	0.115646	0.333333	51.0
english_foxhound	0.170213	0.121212	0.285714	28.0
old_english_sheepdog	0.168421	0.133333	0.228571	35.0
samoyed	0.160000	0.120690	0.237288	59.0

	precision	recall	f1-score	support
greater_swiss_mountain_dog	0.354167	0.500000	0.414634	34.0
sealyham_terrier	0.352941	0.470588	0.403361	51.0
japanese_spaniel	0.386364	0.395349	0.390805	43.0
kerry_blue_terrier	0.307692	0.307692	0.307692	39.0
pomeranian	0.288136	0.288136	0.288136	59.0

Bottom accuracy: Fully Connected NN (top right) vs CNN(bottom right

Breed	F1 Score	Precision	Recall	Support
american_staffordshire_terrier	0.0	0.0	0.0	32.0
black	0.0	0.0	0.0	29.0
bluetick	0.0	0.0	0.0	36.0
bloodhound	0.0	0.0	0.0	43.0
bouvier_des_flandres	0.0	0.0	0.0	25.0

	precision	recall	f1-score	support
collie	0.0	0.0	0.0	27.0
cocker_spaniel	0.0	0.0	0.0	30.0
cardigan	0.0	0.0	0.0	27.0
chihuahua	0.0	0.0	0.0	26.0
great_dane	0.0	0.0	0.0	28.0