SPOTTED!

AC215 • Fall 2023 • Midterm Presentation

GROUP MEMBERS

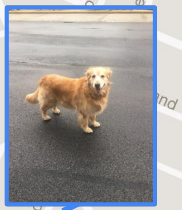
Alex Coward • Olga Leushina • Jonathan Sessa

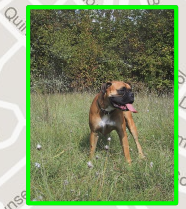
PROBLEM STATEMENT

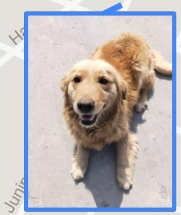
When a pet goes missing, there currently exists a significant **information disconnect** between **lost pet owners** looking for them and **spotters** who find them.

PROPOSED SOLUTION

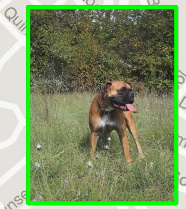
PROPOSED SOLUTION

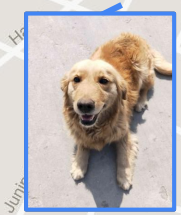


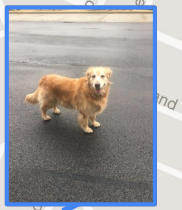


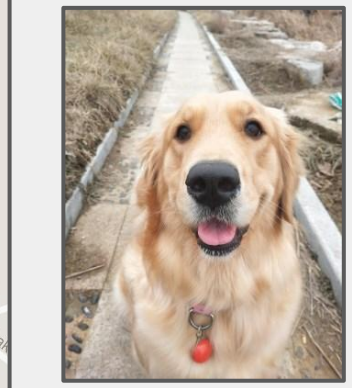


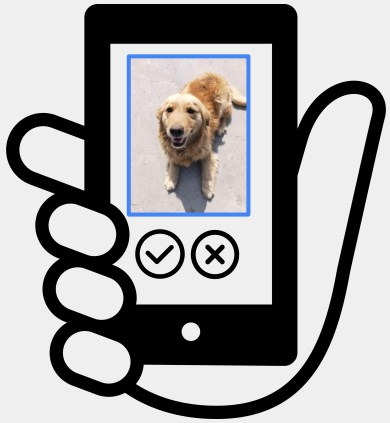
PROPOSED SOLUTION



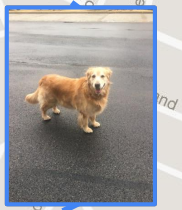


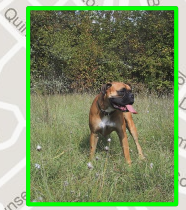


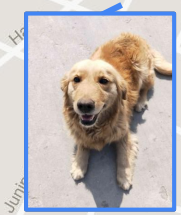




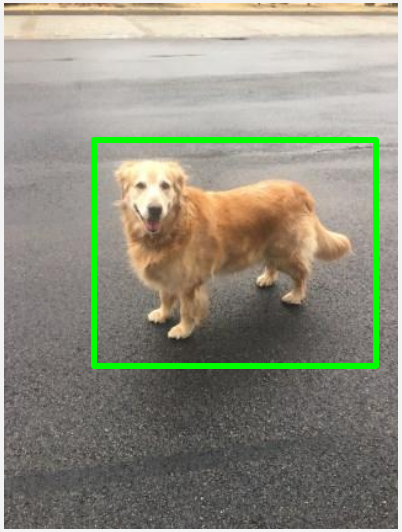
PROPOSED SOLUTION







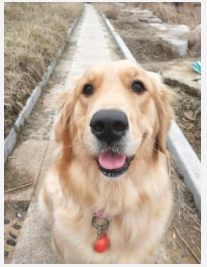
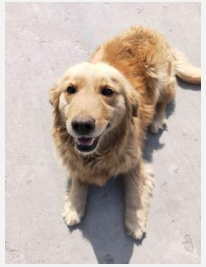
PROPOSED SOLUTION



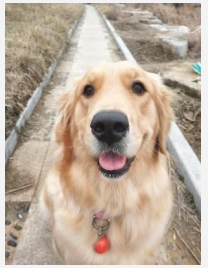
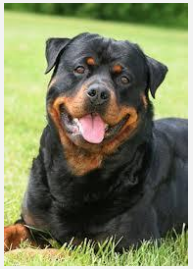
Object Detection

Golden Retriever 72.3%
Labrador Retriever 12.3%
Chesapeake Bay 8.3%
Great Pyrenees 5.8%

Breed Classification

Match Likelihood: 76.3%

Match Likelihood: 18.7%

Matching

DATA

DATA: BY THE NUMBERS

96,000
DOG BREED IMAGES

149
REPRESENTED DOG BREEDS

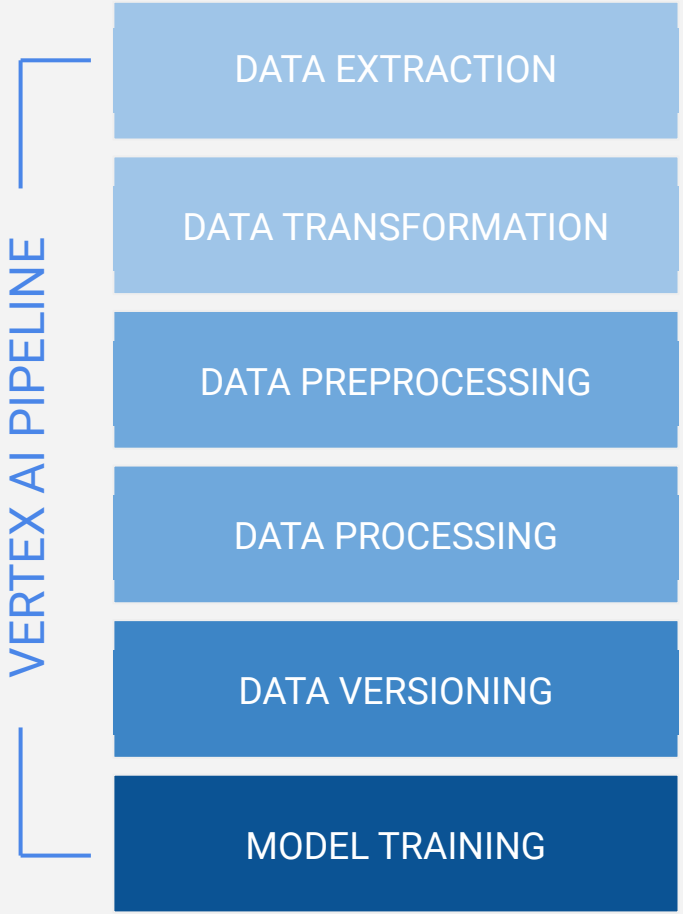
95%
INCLUDE BOUNDING BOXES

161,000 TOTAL IMAGES

65,000
SHELTER DOG IMAGES

14,000
INDIVIDUAL DOGS

DATA: PIPELINE



Download compressed images and annotations

Extract data and upload using **GCSFuse** and **Dask/Multiprocessing**

Extract labels and rename files using **GCSFuse** and **Multiprocessing**

Convert annotations to JSON and common schema

Resize and pad images using **GCSFuse** and **Multiprocessing**

Create train/validation/test split stratified on breed

Convert data and annotations to **TFRecords** (96,000 Files -> 17 Files)

Create shards using **GCSFuse** and **Multiprocessing**

Create **DVC** files for new datasets

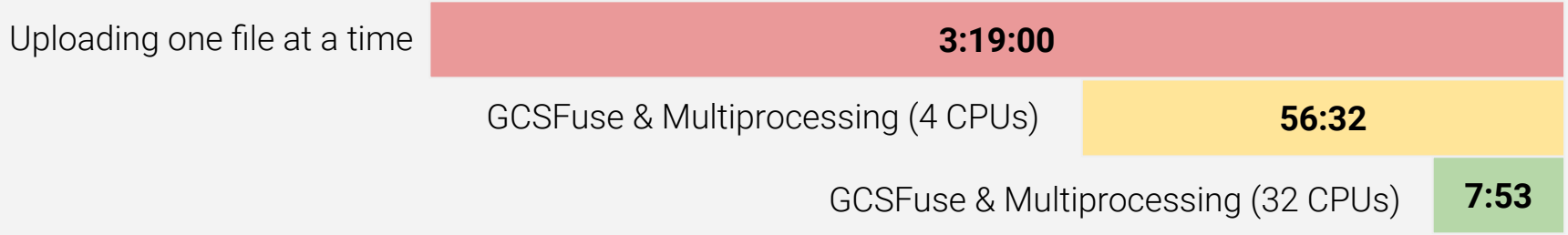
Update **GitHub** with next version number

Select appropriate dataset from GCS depending on model

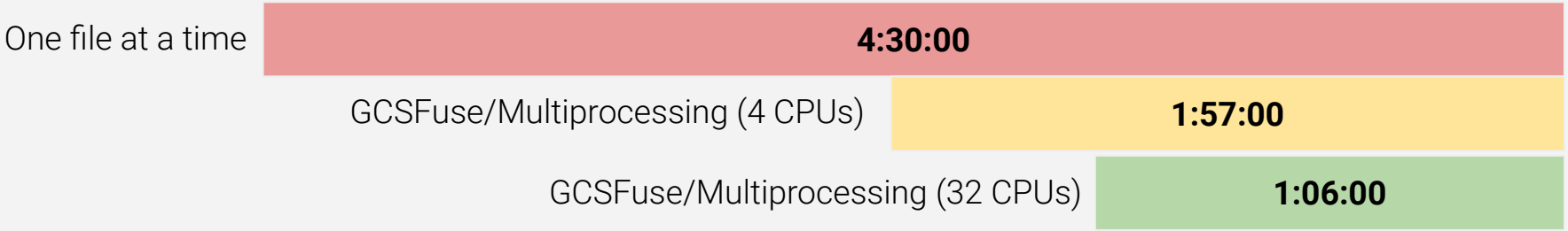
Train model using **Vertex AI** and save to **GCS**

DATA: OPTIMIZATION

DATA EXTRACTION: Copying 20K Images & 20K Annotations



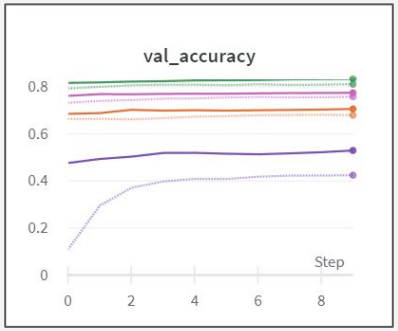
DATA TRANSFORMATION: Resizing 20K Images & Writing 20K Annotations

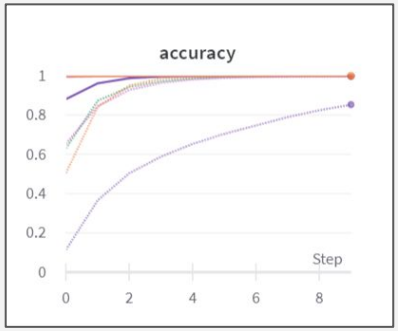


MODELING

MODELING: BREED CLASSIFICATION

Performance on Stanford Dogs Dataset (20K images, 120 Breeds, Colab 10 epochs)

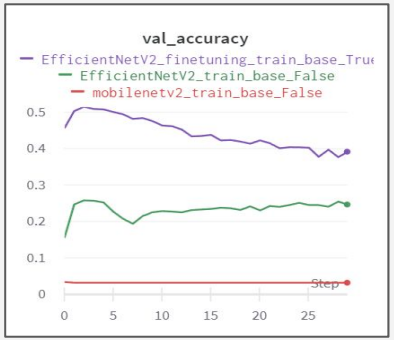


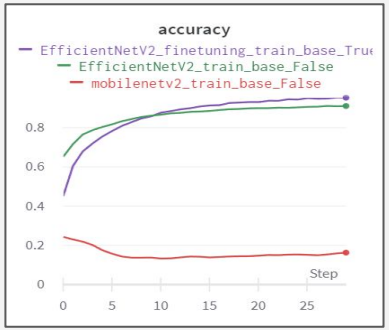


0.82

TEST ACCURACY

Performance on Full Dataset (96K images, 149 Breeds, serverless 30 epochs)



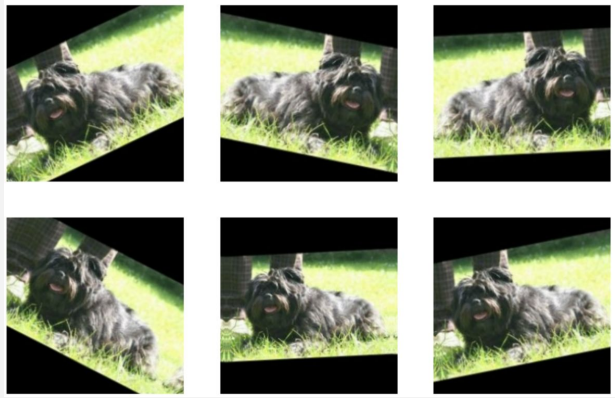


0.39

TEST ACCURACY

Best results: EfficientNetV2_B7 after fine-tuning

MODELING: BREED CLASSIFICATION



To prevent overfitting we will proceed with:

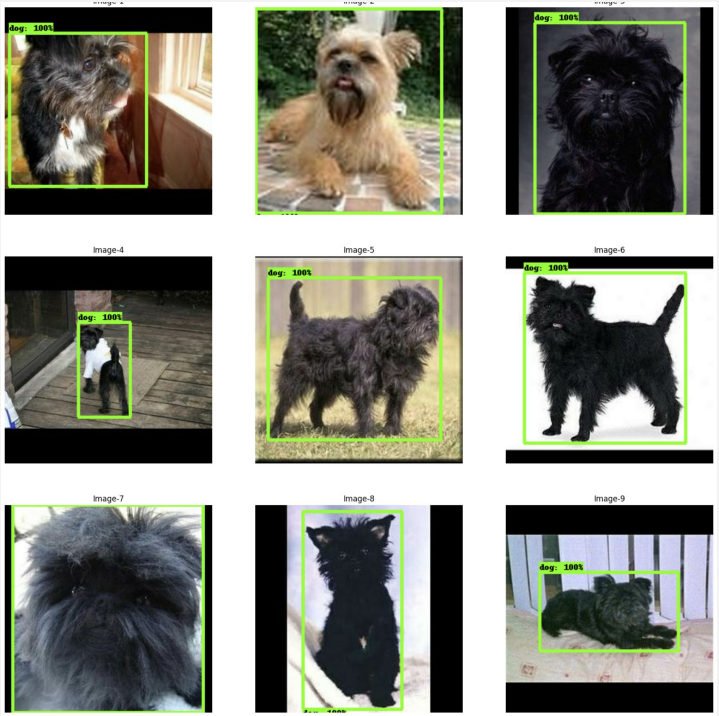
- Data augmentation
- Regularization, play with dropout, optimizers, other models
- Dimensionality reduction (breeds)
- Incorporate object detection

Lesson learned: for model distillation Student has to be more comparable to the Teacher

trainable_parameters	execution_time	loss	accuracy	model_size	learning_rate	batch_size	epochs	optimizer	name
6,014,984	9.87 mins	0.64	83.50%	73.440 MB	0.00001	64	10	Adam	EfficientNetV2_train_base
377,832	6.99 mins	14.19	3.06%	1.532 MB	0.00100	64	35	Adam	student_distill
377,832	0.17 mins	6.88	2.67%	4.575 MB	0.00100	64	5	Adam	student_scratch

NEXT STEPS

NEXT STEPS: OBJECT DETECTION

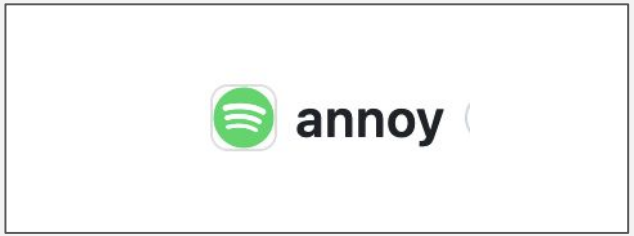


Ground Truth

Model Inference

NEXT STEPS: MATCHING MODEL

VECTOR LIBRARIES



VECTOR DATABASES

NEXT STEPS: WEB APPLICATION

Model Improvements

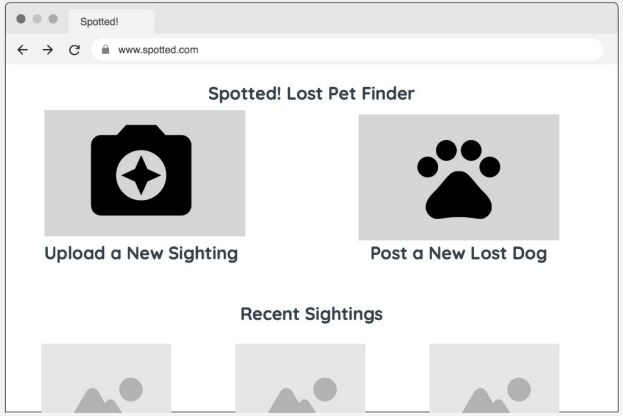
- Additional training and fine-tuning
- Model compression and quantization

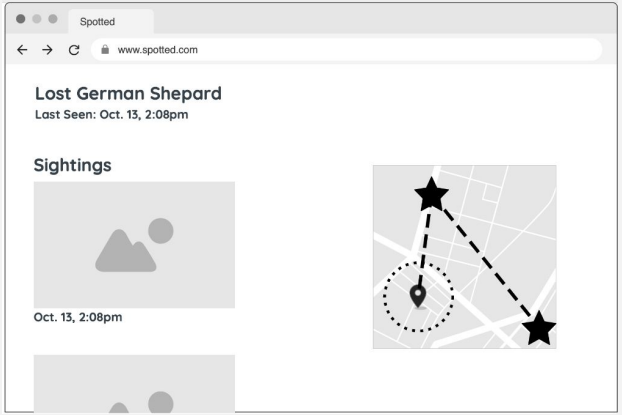
Application Development

- Build and design web application
- Test and deploy final application

Model Monitoring

- Incorporate user data into workflow
- Monitor models for data drift





THANK YOU