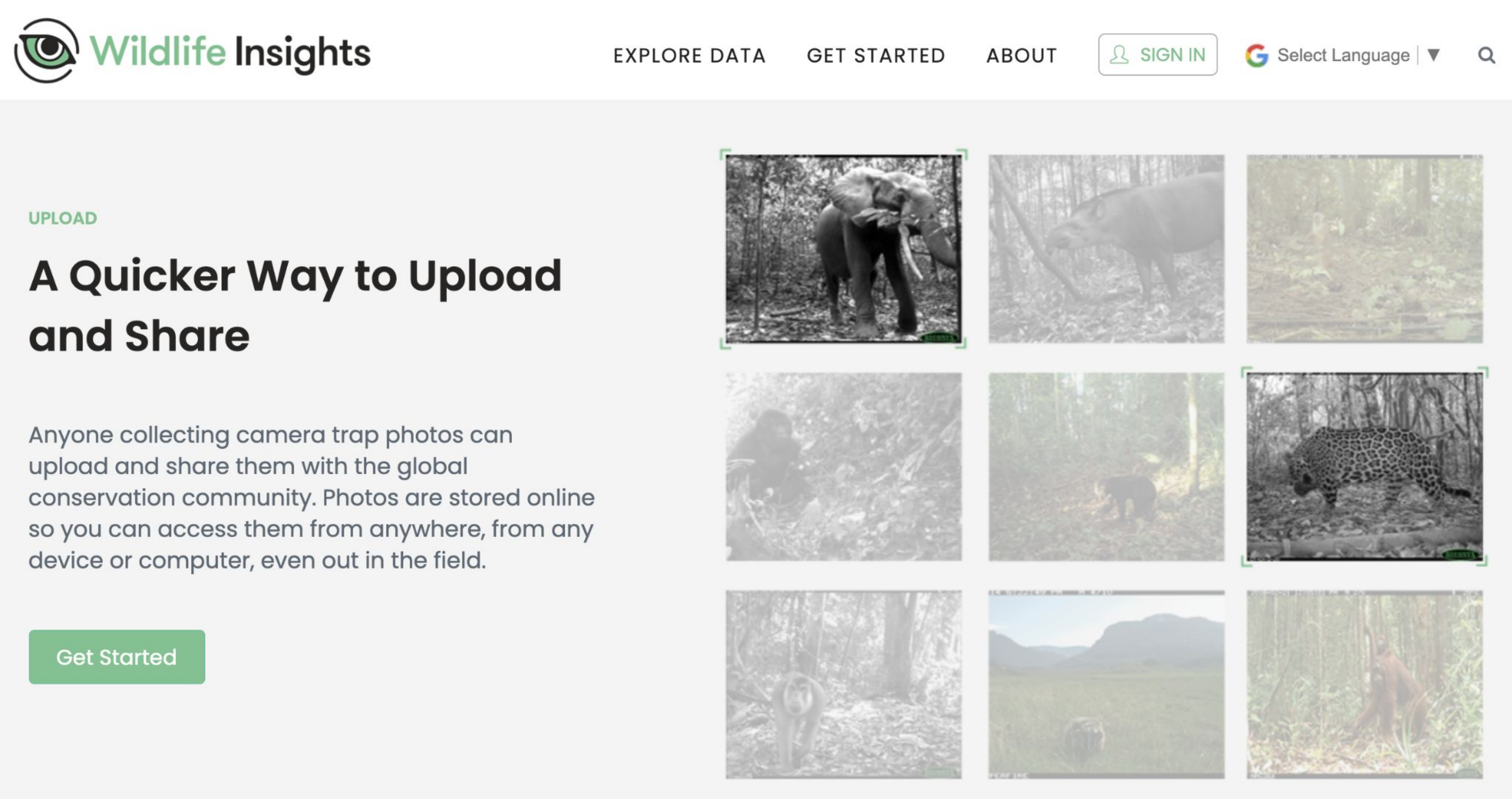
Image classification with Apache Beam and AutoML

Orchestrating simple machine learning pipelines



UPLOAD

A Quicker Way to Upload and Share

Anyone collecting camera trap photos can upload and share them with the global conservation community. Photos are stored online so you can access them from anywhere, from any device or computer, even out in the field.

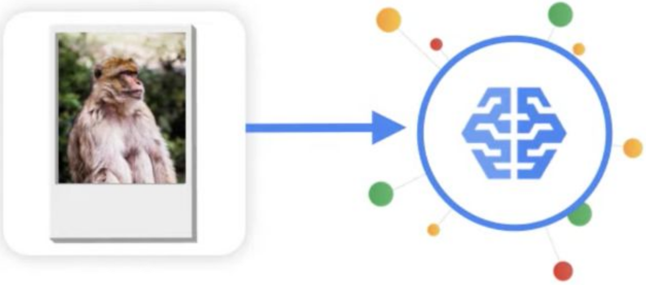
wildlifeinsights.org

Monitor wildlife health via image classification

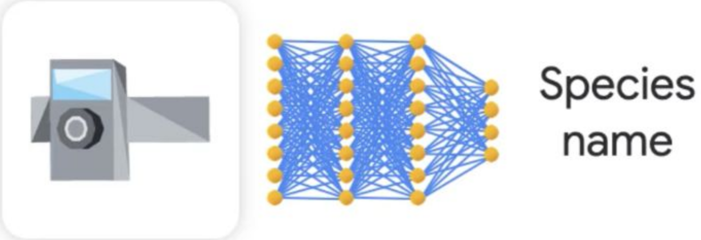
https://youtu.be/hUzODH3uGg0

Open notebook in Colab

Overview



#1 Training: builds an ML model



#2 Prediction: classifies images

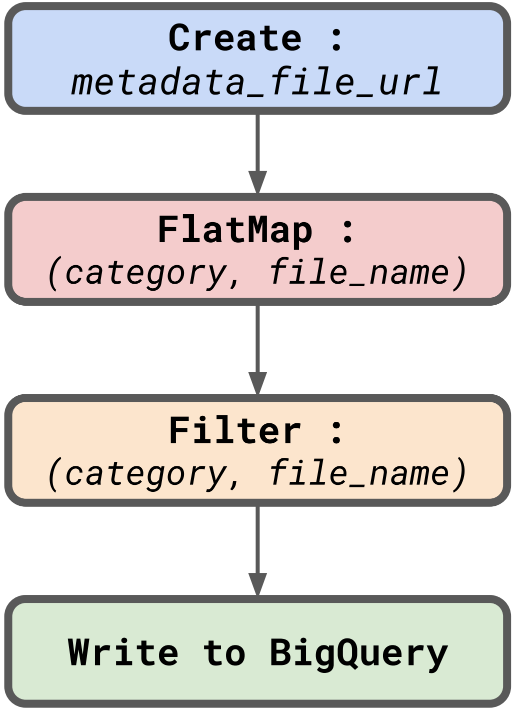
https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/people-and-planet-ai/image-classification

WCS Camera Traps dataset

- Approximately 1.4 million images
- Around 675 species from 12 countries
- More than 560 GB of images total
- Very unbalanced
  - Some species have tens of thousands of images
  - Many species have only a couple images
- Approximately 50% of images are empty
- Image files live in Azure Storage

http://lila.science/datasets/wcscameratraps

Creating the images database



Create : *metadata_file_url*

Singleton metadata URL from LILA science

FlatMap : *(category, file_name)*

Parses the metadata file and yields (category, file_name) pairs

Filter : *(category, file_name)*

Filters out images with invalid categories

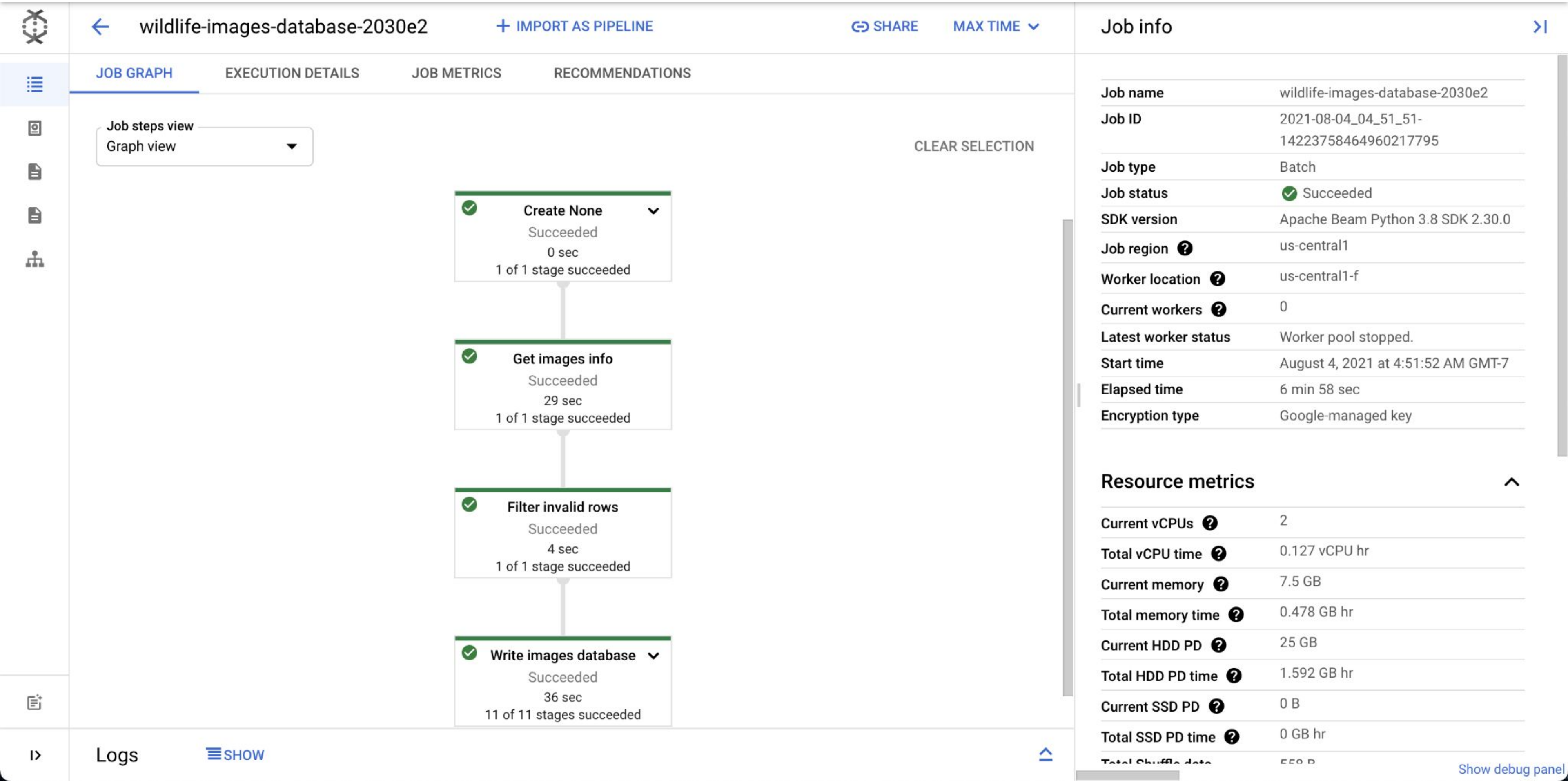
Write to BigQuery

Writes the valid labeled image file names into BigQuery

`create_images_metadata_table.py`

Creating the images database -- job graph

Dataflow



wildlife-images-database-2030e2

Job steps view: Graph view

Create None
Succeeded
0 sec
1 of 1 stage succeeded

Get images info
Succeeded
29 sec
1 of 1 stage succeeded

Filter invalid rows
Succeeded
4 sec
1 of 1 stage succeeded

Write images database
Succeeded
36 sec
11 of 11 stages succeeded

Job info

Job name	wildlife-images-database-2030e2
Job ID	2021-08-04_04_51_51-14223758464960217795
Job type	Batch
Job status	✓ Succeeded
SDK version	Apache Beam Python 3.8 SDK 2.30.0
Job region	us-central1
Worker location	us-central1-f
Current workers	0
Latest worker status	Worker pool stopped.
Start time	August 4, 2021 at 4:51:52 AM GMT-7
Elapsed time	6 min 58 sec
Encryption type	Google-managed key

Resource metrics

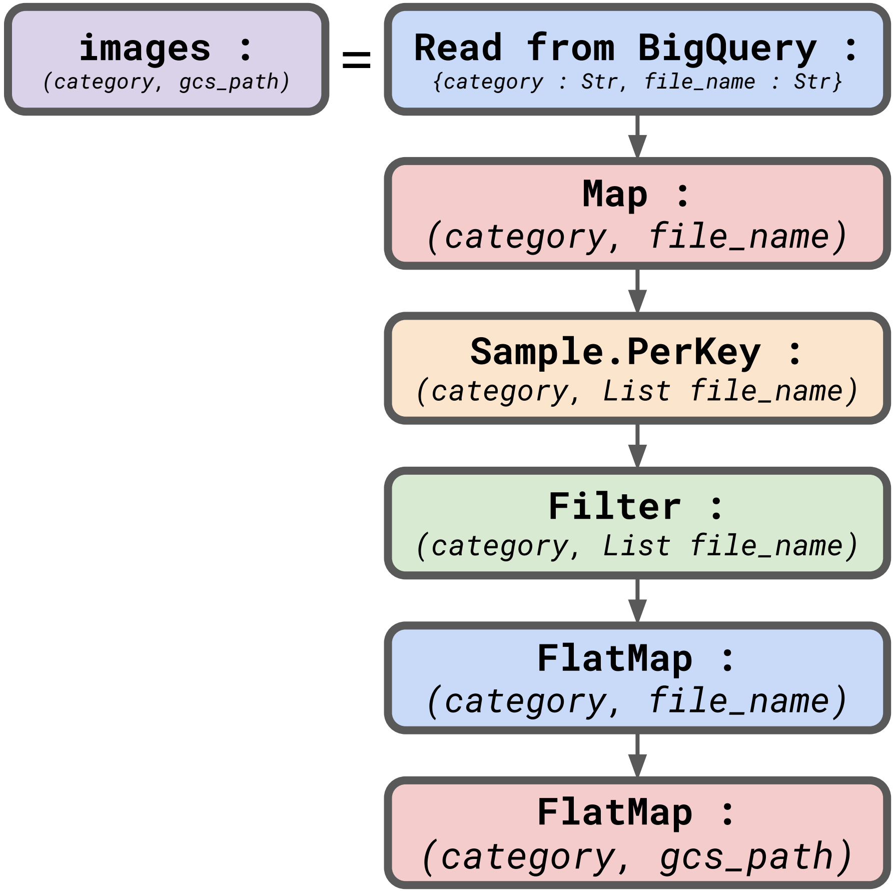
Current vCPUs	2
Total vCPU time	0.127 vCPU hr
Current memory	7.5 GB
Total memory time	0.478 GB hr
Current HDD PD	25 GB
Total HDD PD time	1.592 GB hr
Current SSD PD	0 B
Total SSD PD time	0 GB hr
Total Shuffle data	558 B

Preview of the images metadata table

category	file_name
tapirus bairdii	animals/0597/0707.jpg
equus quagga	animals/0377/1882.jpg
papio anubis	animals/0036/1687.jpg
dicerorhinus sumatrensis	animals/0329/0830.jpg
cephalophus nigrifrons	animals/0331/1215.jpg
tayassu pecari	animals/0174/0182.jpg
cephalophus nigrifrons	animals/0682/1295.jpg
giraffa camelopardalis	animals/0320/1392.jpg
panthera onca	animals/0564/0604.jpg
leopardus pardalis	animals/0576/0243.jpg

Training the model *(part 1) -- balancing the dataset*

beam



images :
(category, gcs_path)
=
Read from BigQuery :
{category : Str, file_name : Str}

Reads the images metadata information from BigQuery

Map :
(category, file_name)

Make `(category, file_name)` pairs

Sample.PerKey :
(category, List file_name)

Get random samples of at most **max_images_per_class**

Filter :
(category, List file_name)

Discard samples smaller than **min_images_per_class**

FlatMap :
(category, file_name)

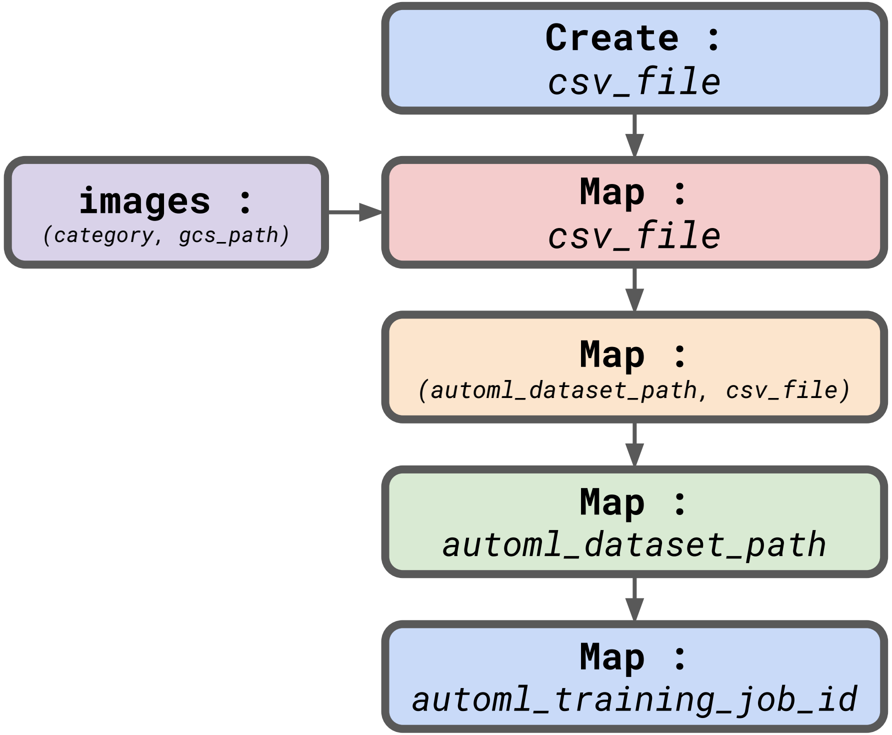
Flatten into `(category, file_name)` pairs

FlatMap :
(category, gcs_path)

Lazily download *only* the images used, skip any errors

Training the model *(part 2) -- preparing for AutoML*

beam



Create : csv_file
images : (category, gcs_path)
Map : csv_file
Map : (automl_dataset_path, csv_file)
Map : automl_dataset_path
Map : automl_training_job_id

Define the file name for a CSV file

Write the labeled images into the CSV file

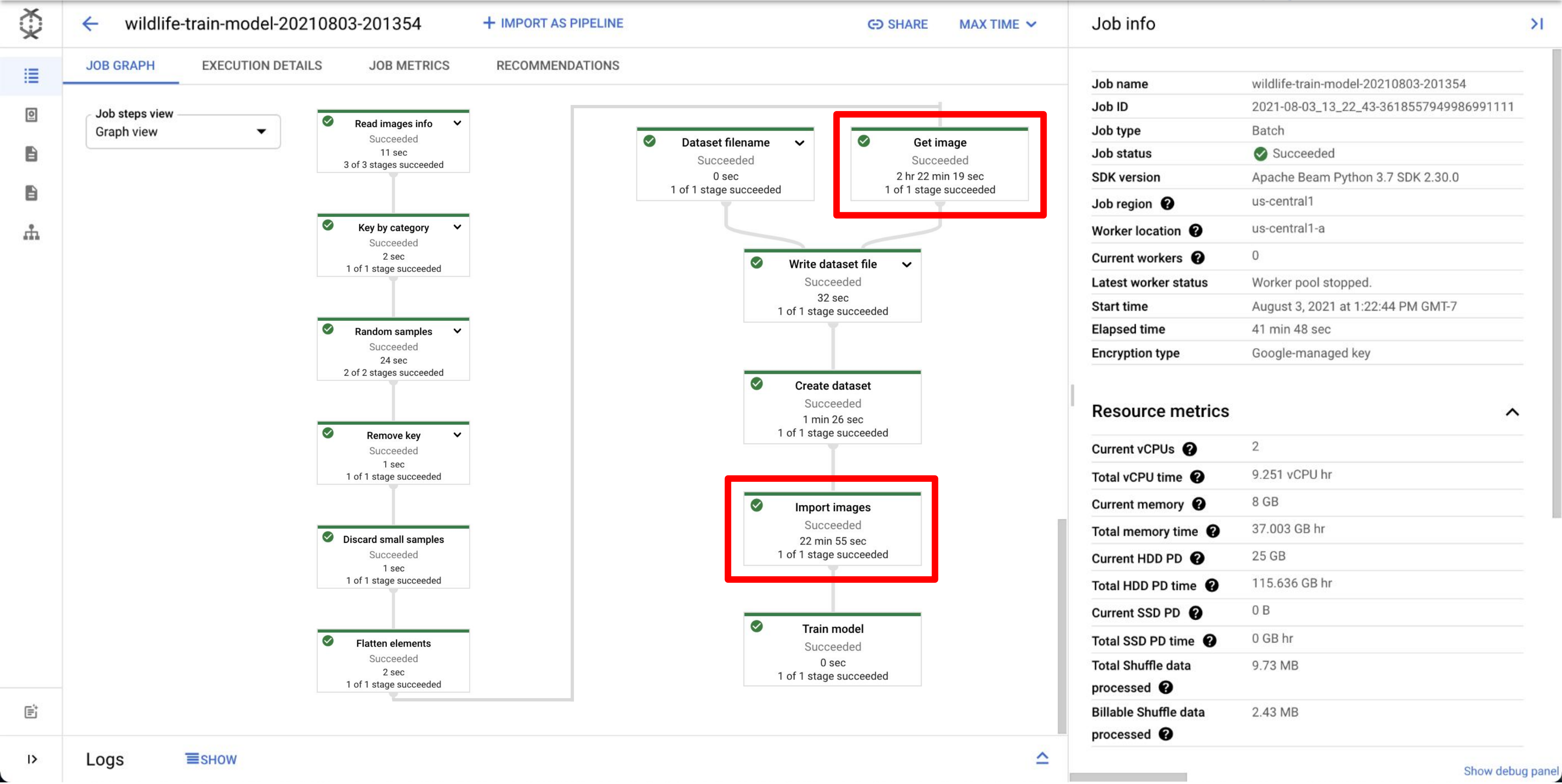
Create an AutoML dataset

Import the CSV entries into the AutoML dataset

Start an AutoML training job

Training the model -- job graph

Dataflow



wildlife-train-model-20210803-201354

Job steps view: Graph view

Read images info
Succeeded
11 sec
3 of 3 stages succeeded

Key by category
Succeeded
2 sec
1 of 1 stage succeeded

Random samples
Succeeded
24 sec
2 of 2 stages succeeded

Remove key
Succeeded
1 sec
1 of 1 stage succeeded

Discard small samples
Succeeded
1 sec
1 of 1 stage succeeded

Flatten elements
Succeeded
2 sec
1 of 1 stage succeeded

Dataset filename
Succeeded
0 sec
1 of 1 stage succeeded

Get image
Succeeded
2 hr 22 min 19 sec
1 of 1 stage succeeded

Write dataset file
Succeeded
32 sec
1 of 1 stage succeeded

Create dataset
Succeeded
1 min 26 sec
1 of 1 stage succeeded

Import images
Succeeded
22 min 55 sec
1 of 1 stage succeeded

Train model
Succeeded
0 sec
1 of 1 stage succeeded

Job info

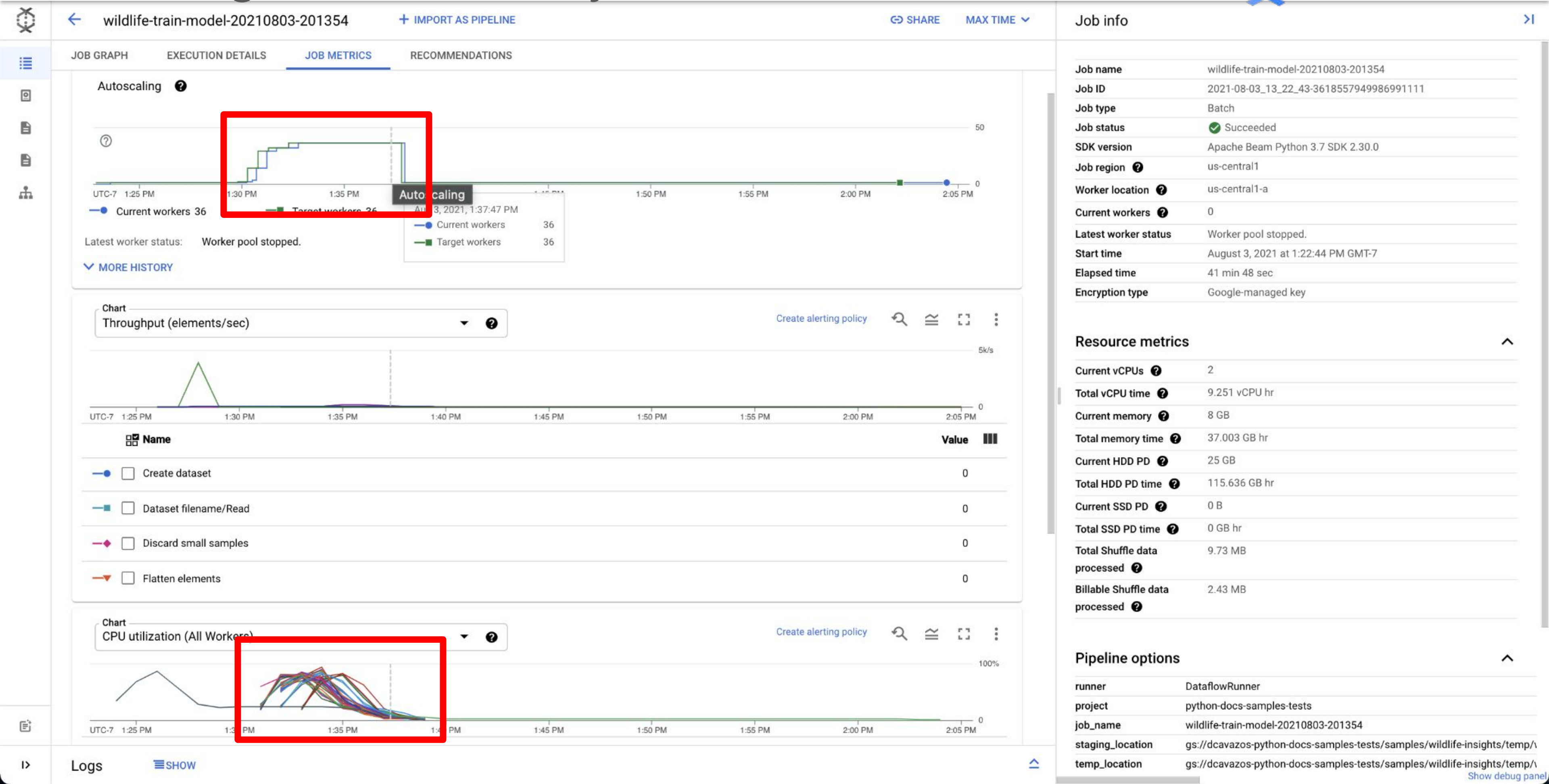
Job name	wildlife-train-model-20210803-201354
Job ID	2021-08-03_13_22_43-3618557949986991111
Job type	Batch
Job status	✓ Succeeded
SDK version	Apache Beam Python 3.7 SDK 2.30.0
Job region ?	us-central1
Worker location ?	us-central1-a
Current workers ?	0
Latest worker status	Worker pool stopped.
Start time	August 3, 2021 at 1:22:44 PM GMT-7
Elapsed time	41 min 48 sec
Encryption type	Google-managed key

Resource metrics

Current vCPUs ?	2
Total vCPU time ?	9.251 vCPU hr
Current memory ?	8 GB
Total memory time ?	37.003 GB hr
Current HDD PD ?	25 GB
Total HDD PD time ?	115.636 GB hr
Current SSD PD ?	0 B
Total SSD PD time ?	0 GB hr
Total Shuffle data processed ?	9.73 MB
Billable Shuffle data processed ?	2.43 MB

Training the model -- job metrics

Dataflow



Current workers 36 Target workers 36

Latest worker status: Worker pool stopped.

Name	Value
□ Create dataset	0
□ Dataset filename/Read	0
□ Discard small samples	0
□ Flatten elements	0

Job info

Job name	wildlife-train-model-20210803-201354
Job ID	2021-08-03_13_22_43-3618557949986991111
Job type	Batch
Job status	Succeeded
SDK version	Apache Beam Python 3.7 SDK 2.30.0
Job region	us-central1
Worker location	us-central1-a
Current workers	0
Latest worker status	Worker pool stopped.
Start time	August 3, 2021 at 1:22:44 PM GMT-7
Elapsed time	41 min 48 sec
Encryption type	Google-managed key

Resource metrics

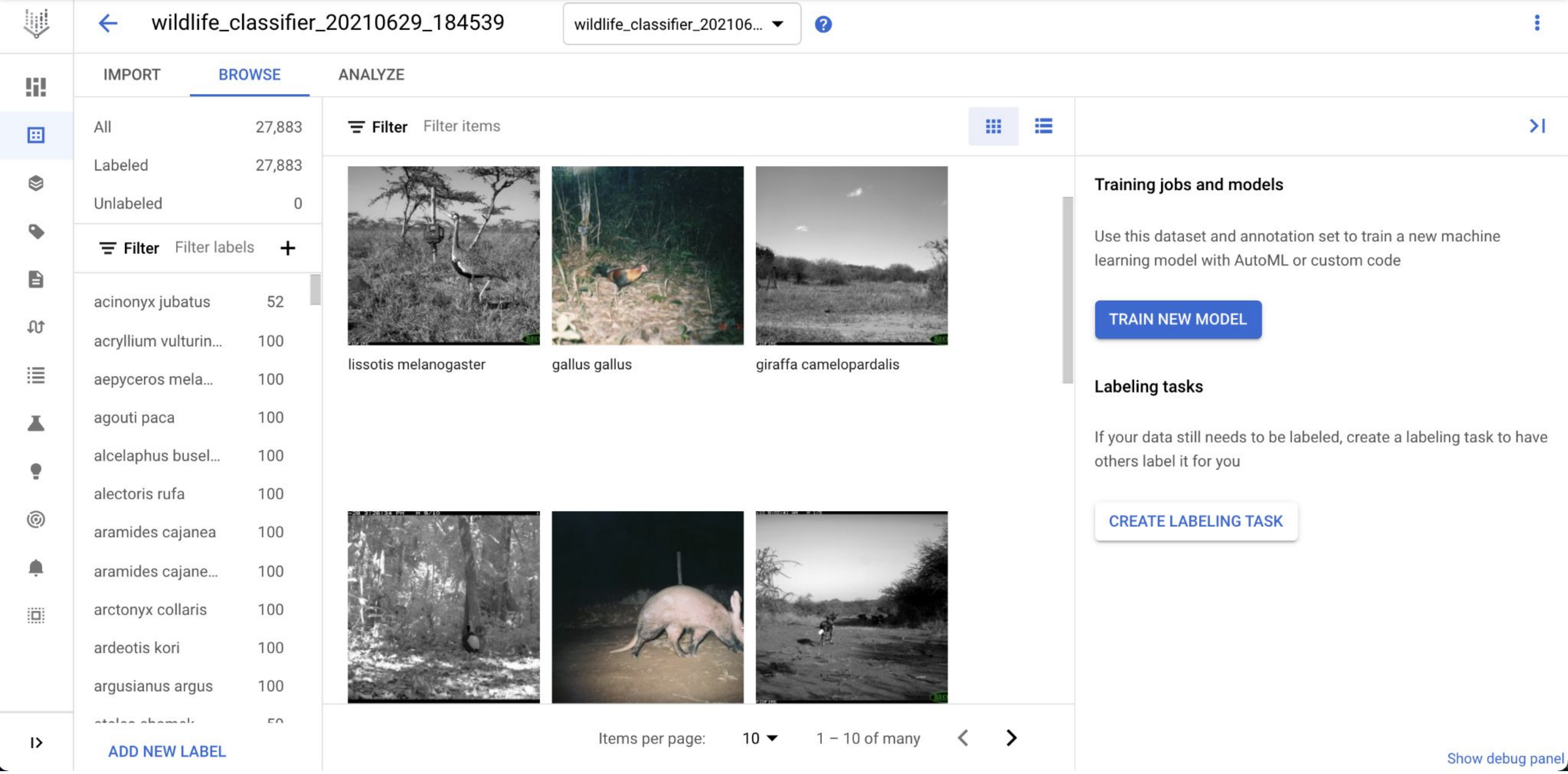
Current vCPUs	2
Total vCPU time	9.251 vCPU hr
Current memory	8 GB
Total memory time	37.003 GB hr
Current HDD PD	25 GB
Total HDD PD time	115.636 GB hr
Current SSD PD	0 B
Total SSD PD time	0 GB hr
Total Shuffle data processed	9.73 MB
Billable Shuffle data processed	2.43 MB

Pipeline options

runner	DataflowRunner
project	python-docs-samples-tests
job_name	wildlife-train-model-20210803-201354
staging_location	gs://dcavazos-python-docs-samples-tests/samples/wildlife-insights/temp/
temp_location	gs://dcavazos-python-docs-samples-tests/samples/wildlife-insights/temp/

Create an AutoML dataset

Vertex AI



wildlife_classifier_20210629_184539

All	27,883
Labeled	27,883
Unlabeled	0

acinonyx jubatus	52
acryllium vulturin...	100
aepyceros mela...	100
agouti paca	100
alcelaphus busel...	100
alectoris rufa	100
aramides cajanea	100
aramides cajane...	100
arctonyx collaris	100
ardeotis kori	100
argusianus argus	100

lissotis melanogaster

gallus gallus

giraffa camelopardalis

Training jobs and models

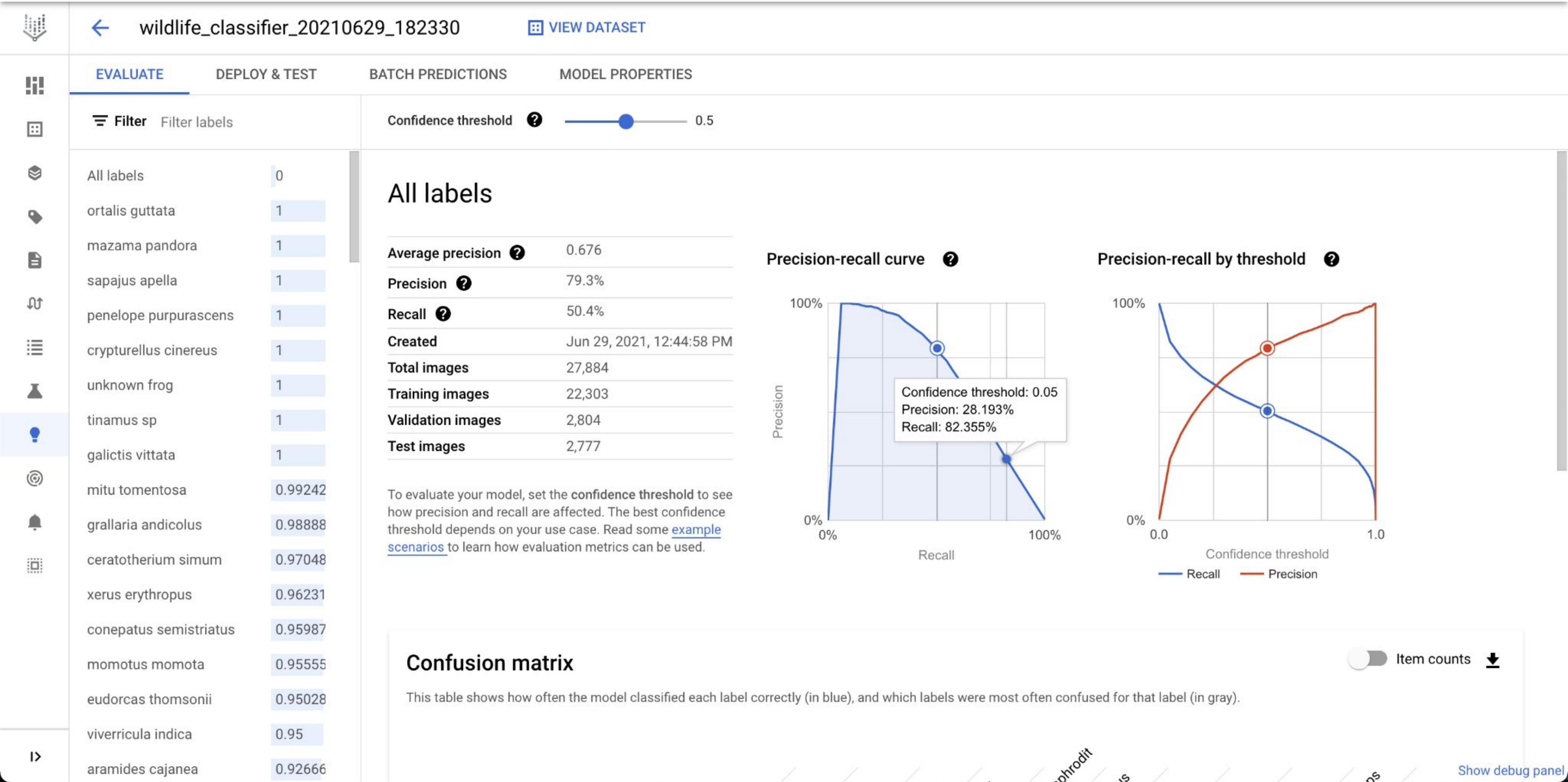
Use this dataset and annotation set to train a new machine learning model with AutoML or custom code

Labeling tasks

If your data still needs to be labeled, create a labeling task to have others label it for you

Training the model -- precision

Vertex AI



wildlife_classifier_20210629_182330

Confidence threshold 0.5

All labels	0
ortalis guttata	1
mazama pandora	1
sapajus apella	1
penelope purpurascens	1
crypturellus cinereus	1
unknown frog	1
tinamus sp	1
galictis vittata	1
mitu tomentosa	0.99242
grallaria andicolus	0.98888
ceratotherium simum	0.97048
xerus erythropus	0.96231
conepatus semistriatus	0.95987
momotus momota	0.95555
eudorcas thomsonii	0.95028
viverricula indica	0.95
aramides cajanea	0.92666

All labels

Average precision	0.676
Precision	79.3%
Recall	50.4%
Created	Jun 29, 2021, 12:44:58 PM
Total images	27,884
Training images	22,303
Validation images	2,804
Test images	2,777

To evaluate your model, set the **confidence threshold** to see how precision and recall are affected. The best confidence threshold depends on your use case. Read some example scenarios to learn how evaluation metrics can be used.

Precision-recall curve

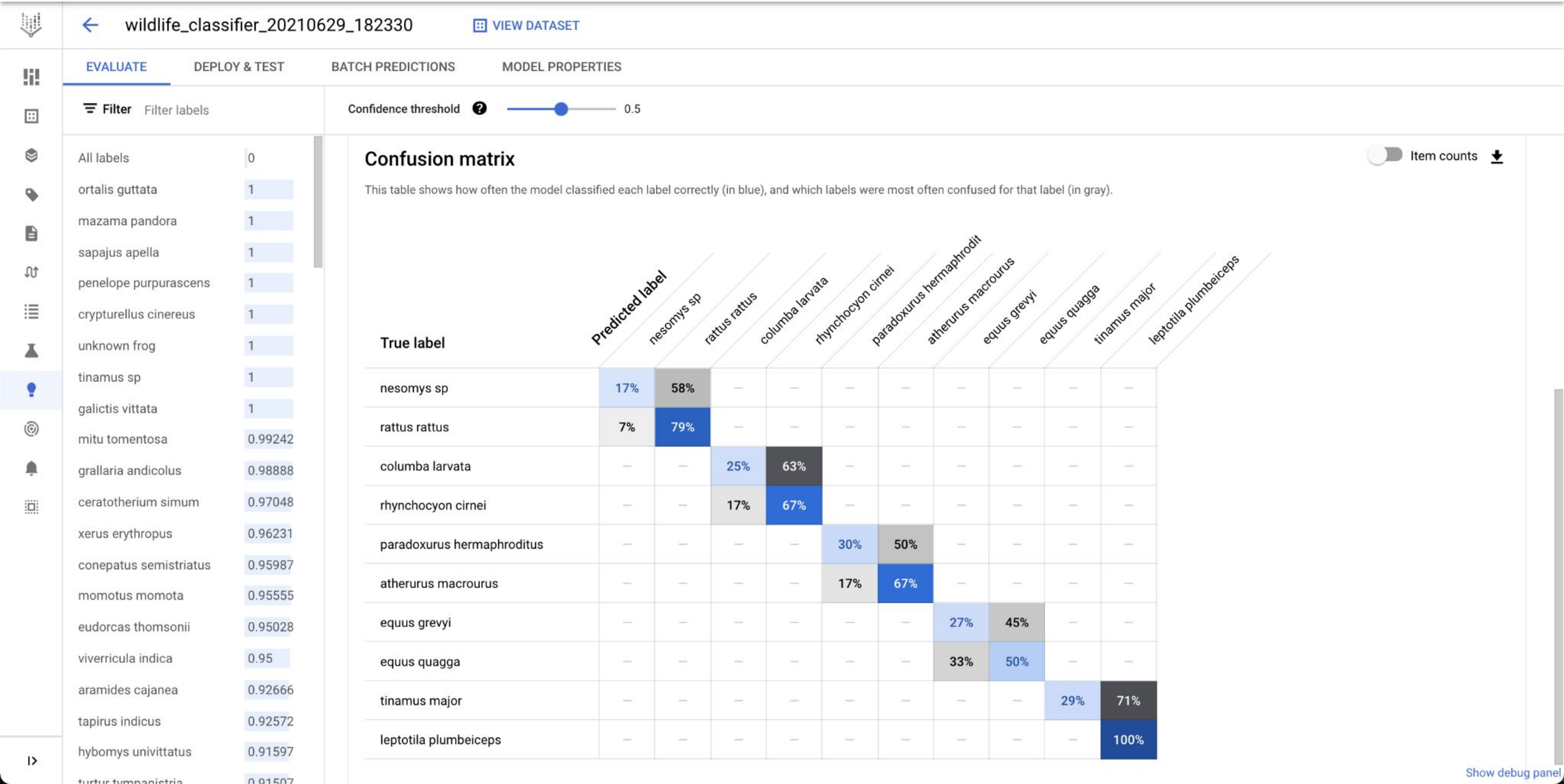
Precision-recall by threshold

Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

Training the model -- confusion matrix

Vertex AI



wildlife_classifier_20210629_182330

Confidence threshold 0.5

All labels	0
ortalis guttata	1
mazama pandora	1
sapajus apella	1
penelope purpurascens	1
crypturellus cinereus	1
unknown frog	1
tinamus sp	1
galictis vittata	1
mitu tomentosa	0.99242
grallaria andicolus	0.98888
ceratotherium simum	0.97048
xerus erythropus	0.96231
conepatus semistriatus	0.95987
momotus momota	0.95555
eudorcas thomsonii	0.95028
viverricula indica	0.95
aramides cajanea	0.92666
tapirus indicus	0.92572
hybomys univittatus	0.91597
turtur tympanistria	0.91507

Confusion matrix

This table shows how often the model classified each label correctly (in blue), and which labels were most often confused for that label (in gray).

True label / Predicted label	nesomys sp	rattus rattus	columba larvata	rhynchocyon cirnei	paradoxurus hermaphrodit	atherurus macrourus	equus grevyi	equus quagga	tinamus major	leptotila plumbeiceps
nesomys sp	17%	58%	–	–	–	–	–	–	–	–
rattus rattus	7%	79%	–	–	–	–	–	–	–	–
columba larvata	–	–	25%	63%	–	–	–	–	–	–
rhynchocyon cirnei	–	–	17%	67%	–	–	–	–	–	–
paradoxurus hermaphroditus	–	–	–	–	30%	50%	–	–	–	–
atherurus macrourus	–	–	–	–	17%	67%	–	–	–	–
equus grevyi	–	–	–	–	–	–	27%	45%	–	–
equus quagga	–	–	–	–	–	–	33%	50%	–	–
tinamus major	–	–	–	–	–	–	–	–	29%	71%
leptotila plumbeiceps	–	–	–	–	–	–	–	–	–	100%

Getting predictions

AutoML Vision

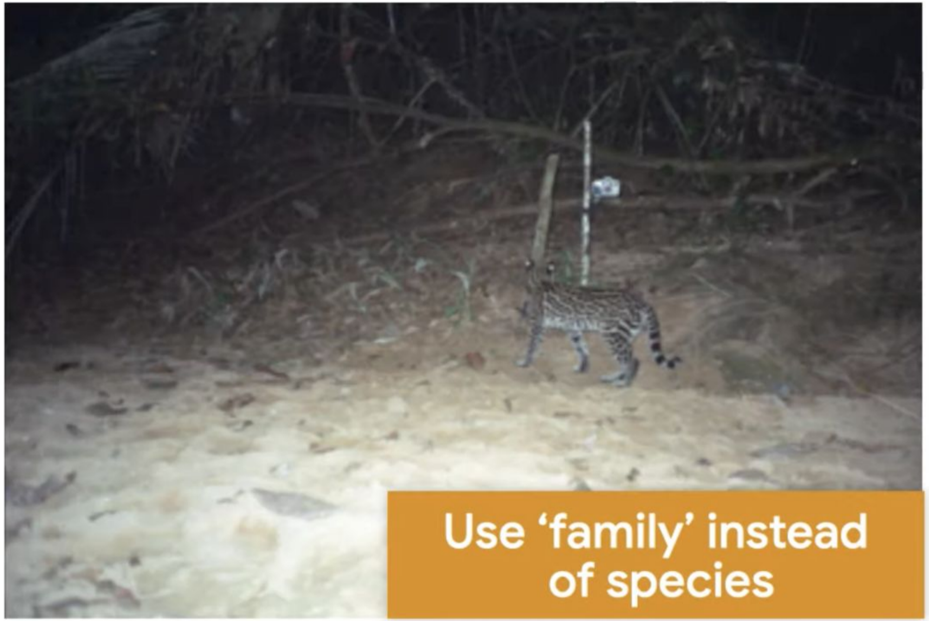
category:
dicerorhinus sumatrensis

file: 'animals/0325/1529.jpg'

prediction:
dicerorhinus sumatrensis:
92.79% confidence

Getting predictions

AutoML Vision


Use ‘family’ instead of species

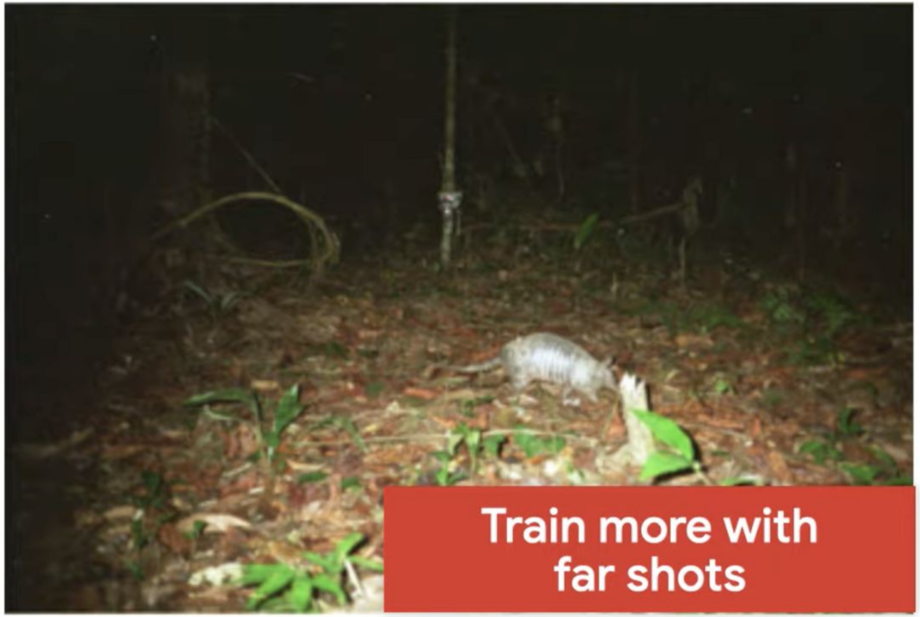
category:
leopardus wiedii

file: 'animals/0000/1705.jpg'

prediction:
leopardus pardalis:
55.56% confidence
leopardus wiedii:
33.45% confidence

Getting predictions

AutoML Vision



Train more with far shots

category:
dasypus novemcinctus

file: 'animals/0000/0425.jpg'

prediction:
procyon cancrivorus:
19.38% confidence

**dasypus novemcinctus:
16.65% confidence**

columba larvata:
10.84% confidence