Object Detection with Vertex AI and AutoML

Daniel Bank

https://github.com/danielbank/object-detection

Scope of This Talk

- Brief history of object detection solutions

- Build an object detection model using Vertex AI

- Run object detection on an Android device (no network calls)

- A little TensorFlow Lite along the way

TL;DR:
It is incredibly easy* to build an object detection model with Vertex AI and deploy it on Android

* assuming you are willing to spend time labeling your data and money training a model

What is Object Detection?

○ **Image classification (Single-label)**

Predict the one correct label that you want assigned to an image.

○ **Image classification (Multi-label)**

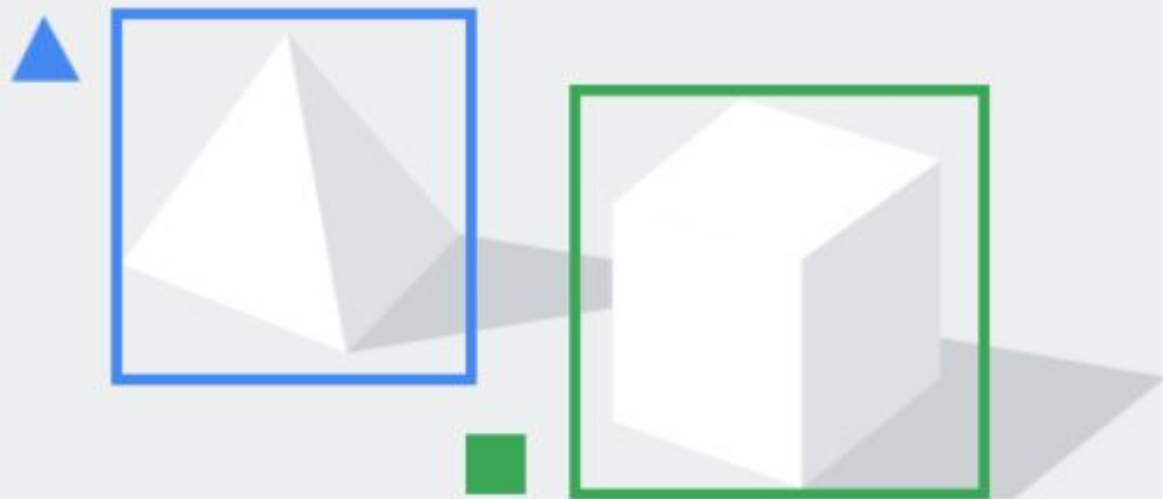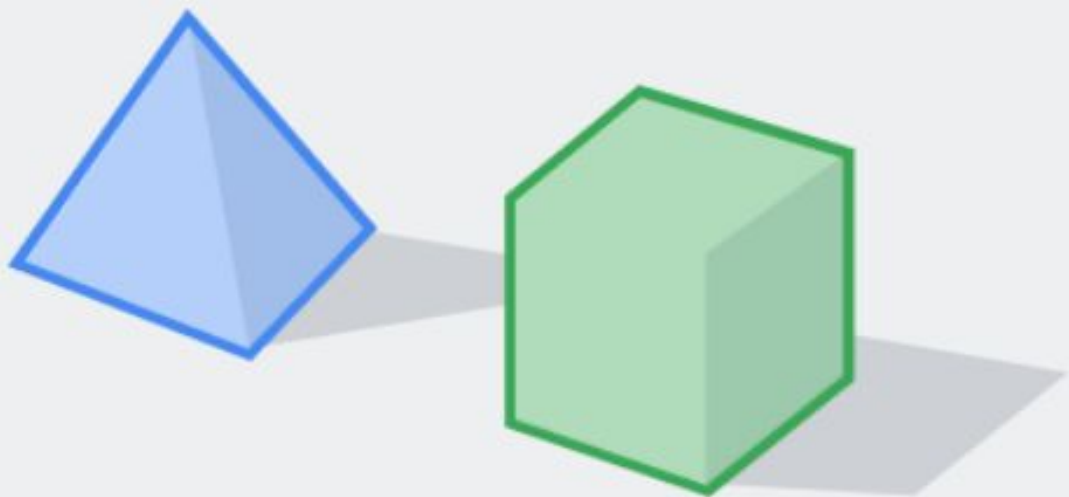Predict all the correct labels that you want assigned to an image.

Image object detection

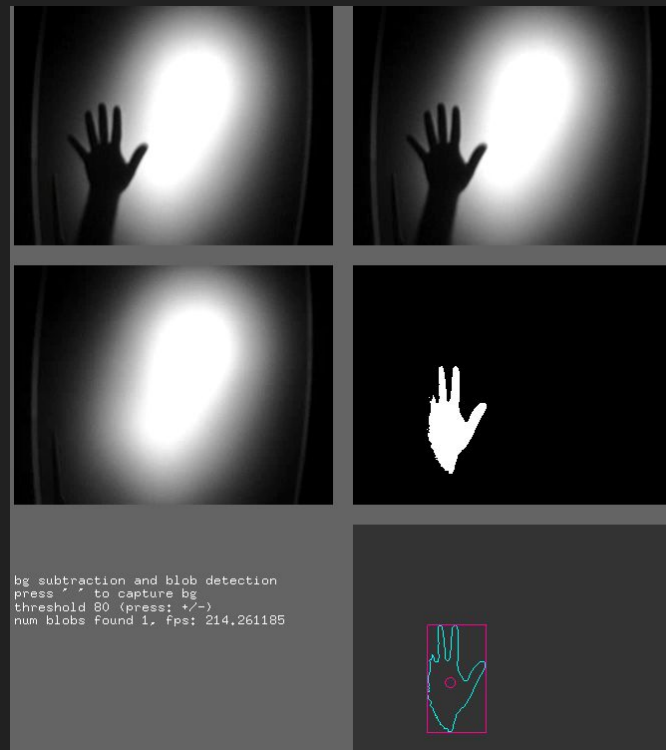Predict all the locations of objects that you're interested in.
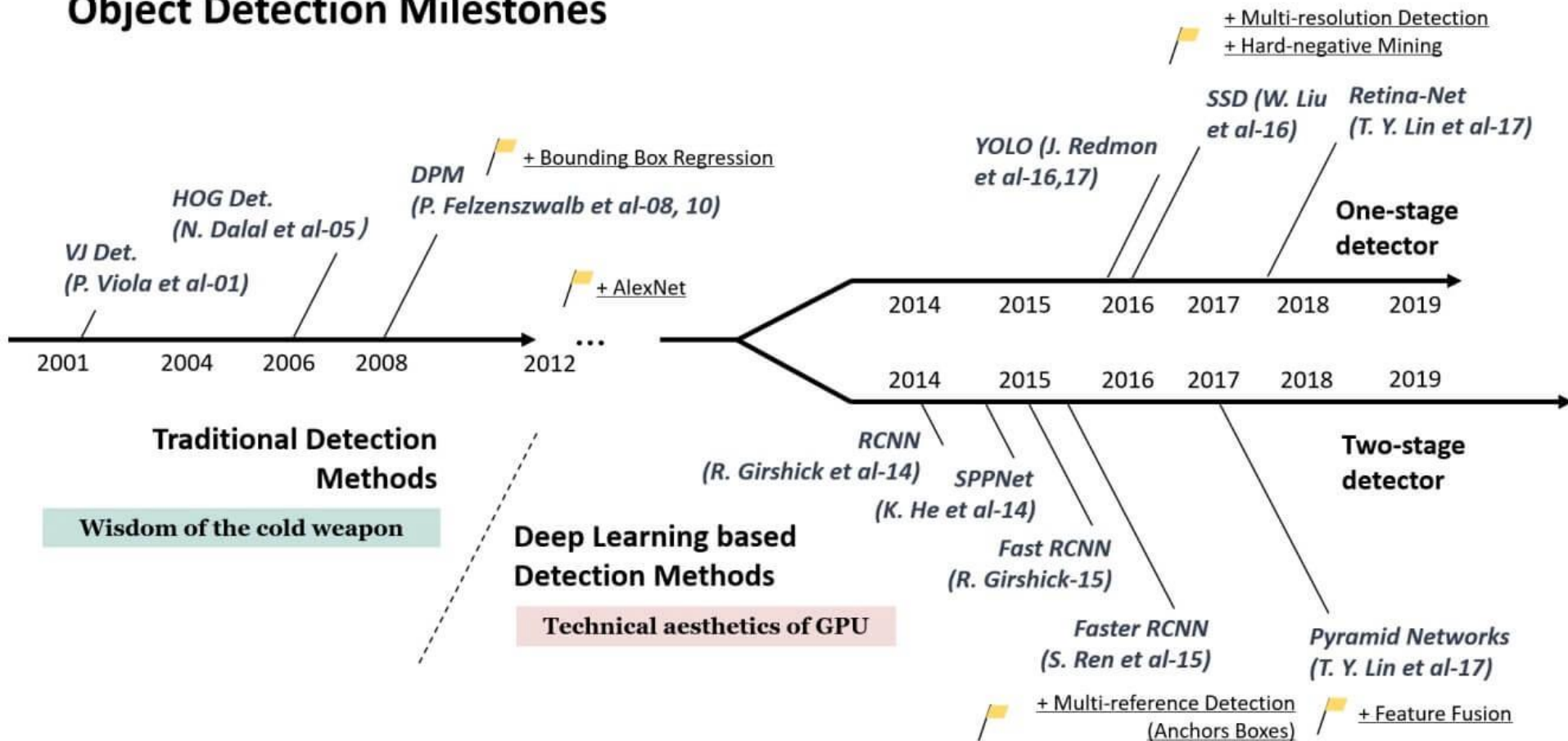
○ **Image segmentation**

Predict per-pixel areas of an image with a label.

OpenCV

- C++ Library for computer vision

- Initially released in **2000**

- Image processing, video capture, and analysis

- DNN Module (Deep Neural Network) in **2015**


```
bg subtraction and blob detection
press ' ' to capture bg
threshold 80 (press: +/-)
num blobs found 1, fps: 214.261185
```

Object Detection Milestones

+ Multi-resolution Detection
+ Hard-negative Mining

SSD (W. Liu et al-16)   Retina-Net (T. Y. Lin et al-17)

YOLO (J. Redmon et al-16,17)

+ Bounding Box Regression

DPM (P. Felzenszwalb et al-08, 10)

HOG Det. (N. Dalal et al-05)

VJ Det. (P. Viola et al-01)

+ AlexNet

One-stage detector

2001   2004   2006   2008   2012   ...

2014   2015   2016   2017   2018   2019

2014   2015   2016   2017   2018   2019

Traditional Detection Methods

Wisdom of the cold weapon

Deep Learning based Detection Methods

Technical aesthetics of GPU

RCNN (R. Girshick et al-14)

SPPNet (K. He et al-14)

Fast RCNN (R. Girshick-15)

Faster RCNN (S. Ren et al-15)

Two-stage detector

Pyramid Networks (T. Y. Lin et al-17)

+ Multi-reference Detection (Anchors Boxes)
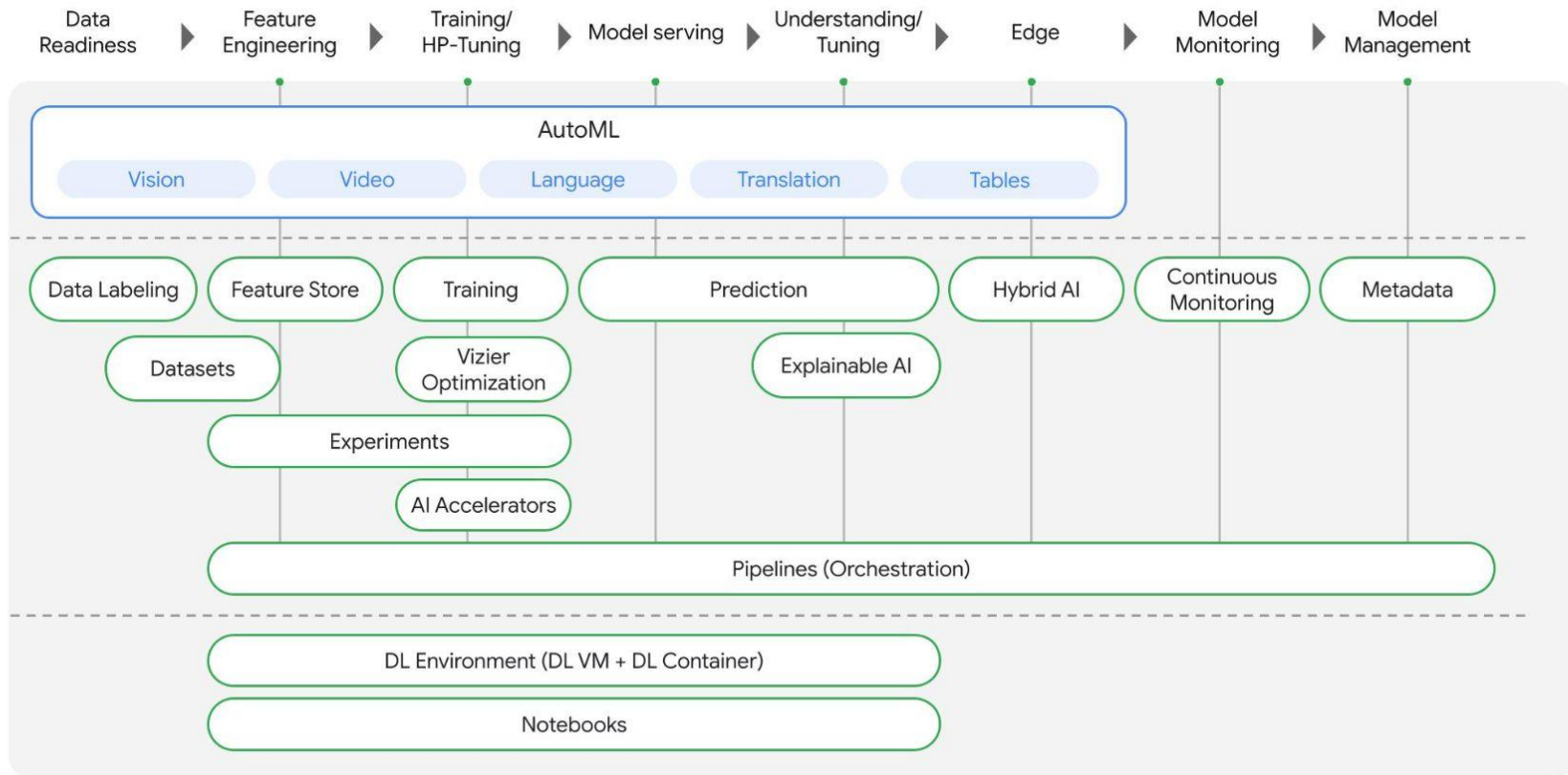
+ Feature Fusion

TensorFlow Object Detection API

- An open source framework built on top of TensorFlow that makes it easy to construct, train and deploy object detection models

- Released in **2017**

- Includes a [Detection Model Zoo](#), a collection of pre-trained models

  - Model Zoo only has two-stage detectors

Vertex AI and AutoML

- Vertex AI

  - Google's Managed ML Platform

  - Launched in **2021**

- Neural Architecture Search (NAS) a.k.a AutoML

  - AI that generates other Neural Networks

  - Developed by Google Brain in **2017**

  - Vertex AI NAS

    - Available for Qualcomm Technologies Neural Processing SDK, optimized for Snapdragon 8

What's included in Vertex AI?

Data Readiness ▶ Feature Engineering ▶ Training/HP-Tuning ▶ Model serving ▶ Understanding/Tuning ▶ Edge ▶ Model Monitoring ▶ Model Management

AutoML

| Vision | Video | Language | Translation | Tables |

Data Labeling | Feature Store | Training | Prediction | Hybrid AI | Continuous Monitoring | Metadata

Datasets

Vizier Optimization

Explainable AI

Experiments

AI Accelerators

Pipelines (Orchestration)

DL Environment (DL VM + DL Container)

Notebooks

Building a Dataset

Detecting Dice

- Six-sided Dice Dataset: https://www.kaggle.com/nellbyler/d6-dice

    - 250 images

    - 1-25 six-sided dice per image

    - Images and annotations (YOLO format)

- Dice Detection Tutorial: https://github.com/nell-byler/dice_detection

Search   Products, resources, docs (/)

Vertex AI

Dashboard

Datasets

Feature Store

Labeling tasks

Workbench

Pipelines

Training

Experiments

Model Registry

Endpoints

Batch predictions

Metadata

Matching Engine

Marketplace

← **Create dataset**

○ **Image classification (Single-label)**
Predict the one correct label that you want assigned to an image.

○ **Image classification (Multi-label)**
Predict all the correct labels that you want assigned to an image.
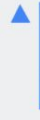
◉ **Image object detection**
Predict all the locations of objects that you're interested in.
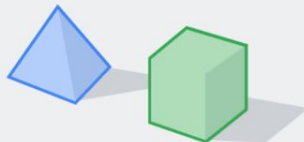
○ **Image segmentation**
Predict per-pixel areas of an image with a label.

Region
us-central1 (Iowa)

∨ **ADVANCED OPTIONS**

You can use this dataset for other image-based objectives later by creating an annotation set. Learn more

Vertex AI

← d6_dice_1662878099415   d6_dice_1662878099415...

IMPORT   BROWSE   ANALYZE

TOOLS

Dashboard

Workbench

Pipelines

DATA

Feature Store

Datasets

Labeling tasks

MODEL DEVELOPMENT

DEPLOY AND USE

Marketplace

You're ready to train

Your dataset meets the recommendation of 100 images assigned to each label. Learn more

Labels	Images	
1		258
2		272
3		274
4		253
5		280
6		241

Custom data split

You currently don't have any items assigned to Training, Validation, or Test sets. Don't worry, when you train a model, your items will be randomly divided into these sets, using recommended machine learning practices. Learn more
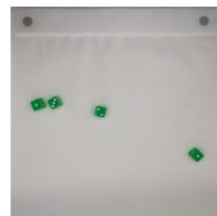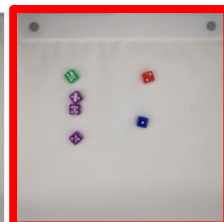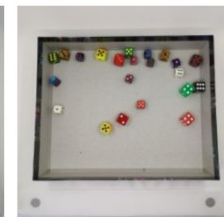
Training jobs and models

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

TRAIN NEW MODEL

Labeling tasks

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

CREATE LABELING TASK

Training a Model

Vertex AI

TOOLS

- Dashboard
- Workbench
- Pipelines

DATA

- Feature Store
- Datasets
- Labeling tasks

MODEL DEVELOPMENT

DEPLOY AND USE

- Marketplace

Train new model

1. Training method
2. Model details
3. Training options
4. Compute and pricing

START TRAINING   CANCEL

Dataset
d6_dice_1662878099415

Annotation set
d6_dice_1662878099415_iod

Objective
Image object detection

Please refer to the pricing guide for more details (and available deployment options) for each method.

Model training method

○ AutoML
Train high-quality models with minimal effort and machine learning expertise. Just specify how long you want to train. Learn more

● AutoML Edge
Train a model that can be exported for on-prem/on-device use. Typically has lower accuracy. Learn more

○ Custom training (advanced)
Run your TensorFlow, scikit-learn, and XGBoost training applications in the cloud. Train with one of Google Cloud's pre-built containers or use your own. Learn more

CONTINUE

☰

Vertex AI

TOOLS

🔲 Dashboard

👤 Workbench

⟋⟍ Pipelines

DATA ⌃

◆ Feature Store

▦ Datasets

🏷 Labeling tasks

MODEL DEVELOPMENT ⌄

DEPLOY AND USE ⌄

🛒 Marketplace

◁

← IMP

All

Label

Unlab

▽ F

Image

1

2

3

4

5

6

ADD

Train new model

✓ Training method

✓ Model details

③ Training options

④ Compute and pricing

START TRAINING   CANCEL

	Goal	Package size	Accuracy	Latency on iPhone X ▾
○	Higher accuracy	5.6 MB	Higher	34ms
◉	Best trade-off	3.1 MB	Medium	23ms
○	Faster predictions	557 KB	Lower	8ms

Please note that prediction latency estimates are for guidance only. Actual latency depends on your network connectivity. Edge TPU predictions typically will have lower latency.

CONTINUE

Vertex AI

TOOLS

Dashboard

Workbench

Pipelines

DATA

Feature Store

Datasets

Labeling tasks

MODEL DEVELOPMENT

DEPLOY AND USE

Marketplace

IMP

All

Label

Unlab

Image

1

2

3

4

5

6

ADI

Train new model

✓ Training method

✓ Model details

✓ Training options

④ Compute and pricing

START TRAINING   CANCEL

Enter the **maximum** number of node hours you want to spend training your model.

You can train for as little as 1 node hour. You may also be eligible to train with free node hours. Pricing guide

Budget *
2   Maximum node hours

Estimated completion date: Sep 18, 2022 4 PM GMT-7

◉ **Enable early stopping**
Ends model training when no more improvements can be made and refunds leftover training budget. If early stopping is disabled, training continues until the budget is exhausted.

Vertex AI

Training   + CREATE

⟳ REFRESH   💡 LEARN

TOOLS

- Dashboard
- Workbench
- Pipelines

DATA

MODEL DEVELOPMENT

- Training
- Experiments
- Metadata

DEPLOY AND USE

- Endpoints
- Model Registry
- Batch predictions
- Matching Engine
- Marketplace

TRAINING PIPELINES | CUSTOM JOBS | HYPERPARAMETER TUNING JOBS

Training pipelines are the primary model training workflow in Vertex AI. You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding a dataset or uploading the model to Vertex AI for prediction serving. Learn More

Region
us-central1 (Iowa)   ▼   ❓

⊞ Filter   Enter a property name   ❓   ▯▯

Name	ID	Status	Job type	Model type	Created	Elapsed time	Labels
d6_dice_edge	3563755504767336448	⟳ Training	Training pipeline	🖼 Image object detection	Sep 18, 2022, 1:38:12 PM	28 min 34 sec	—	⋮

Vertex AI

Training ➕ CREATE

REFRESH LEARN

TRAINING PIPELINES   CUSTOM JOBS   HYPERPARAMETER TUNING JOBS

Training pipelines are the primary model training workflow in Vertex AI. You can use training pipelines to create an AutoML-trained model or a custom-trained model. For custom-trained models, training pipelines orchestrate custom training jobs and hyperparameter tuning with additional steps like adding a dataset or uploading the model to Vertex AI for prediction serving. Learn More

Region
us-central1 (Iowa) ▾ ❓

☰ Filter   Enter a property name

Name	ID	Status	Job type	Model type	Created	Elapsed time	Labels
d6_dice_edge	3563755504767336448	✅ Finished	Training pipeline	🖼 Image object detection	Sep 18, 2022, 1:38:12 PM	2 hr 18 min	—

TOOLS
- Dashboard
- Workbench
- Pipelines

DATA ▾

MODEL DEVELOPMENT ▲
- Training
- Experiments
- Metadata

DEPLOY AND USE ▾

Marketplace

Vertex AI

← d6_dice_edge > Version 1 ▾   ▣ VIEW DATASET   ⬇ EXPORT

EVALUATE   DEPLOY & TEST   BATCH PREDICT   VERSION DETAILS

← untitled_4497850143629901824 ▾   ⇄ COMPARE   ⊞ CREATE EVALUATION

TOOLS

▤ Dashboard

▦ Workbench

⬡ Pipelines

DATA ⌄

MODEL DEVELOPMENT ⌃

◉ Training

⚗ Experiments

▦ Metadata

DEPLOY AND USE ⌃

◉ Endpoints

◆ Model Registry

▣ Batch predictions

⬢ Matching Engine

▦ Marketplace

⟨

▤ Filter   Filter labels

| Confidence threshold ❔ | ────⬤──── | 0.34 |
| IoU threshold ❔ | ───────⬤── | 0.44 |

All labels	0
6	0.773
1	0.765
3	0.746
2	0.693
5	0.677
4	0.623

Validation images
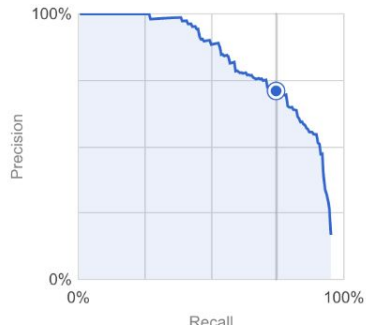
| Test images | 26 |

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.
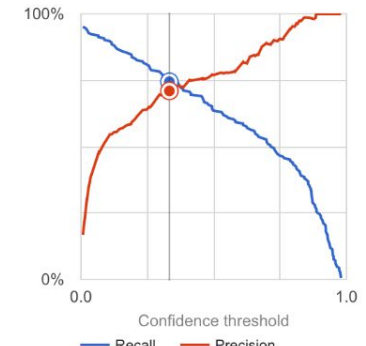
Precision-recall curve ❔

Precision-recall by threshold ❔

─── Recall   ─── Precision

Vertex AI

← d6_dice_edge > Version 1 ▼   ⊞ VIEW DATASET   ⬆ EXPORT   💡 LEARN

EVALUATE   DEPLOY & TEST   BATCH PREDICT   VERSION DETAILS

TOOLS

Dashboard

Workbench

Pipelines

DATA   ⌄

MODEL DEVELOPMENT   ⌃

Training

Experiments

Metadata

DEPLOY AND USE   ⌃

Endpoints

Model Registry

Batch predictions

Matching Engine

Marketplace

Use your edge-optimized model

TF Lite

Export your model as a TF Lite package to run your model on edge or mobile devices.

Container

Export your model as a TF Saved Model to run on a Docker container.

TensorFlow.js

Export your model as a TensorFlow.js package to run your model in the browser and in Node.js.

Deploy your model

Endpoints are machine learning models made available for online prediction requests. Endpoints are useful for timely predictions from many users (for example, in response to an application request). You can also request batch predictions if you don't need immediate results.

DEPLOY TO ENDPOINT

Name	ID	Status	Models	Region	Monitoring	Most recent monitoring job	Most recent alerts	Last updated ↓	API	Notification	Labels ❓
No active endpoints containing this model											

Test your model PREVIEW

Vertex AI

TOOLS

Dashboard

Workbench

Pipelines

DATA

MODEL DEVELOPMENT

Training

Experiments

Metadata

DEPLOY AND USE

Endpoints

Model Registry

Batch predictions

Matching Engine

Marketplace

← d6_dice_edge › Version 1 ▾   ⊞ VIEW DATASET

EVALUATE   **DEPLOY & TEST**   BATCH PREDICT   VERSIO

TF Lite

Export your model as a TF Lite package to run your model on edge or mobile devices.

Container

Export your model as a TF S Model to run on a Docker c

Deploy your model

Endpoints are machine learning models made available for online prediction reques
are useful for timely predictions from many users (for example, in response to an a
request). You can also request batch predictions if you don't need immediate result

DEPLOY TO ENDPOINT

Name	ID	Status	Models	Region	Monitoring
No active endpoints containing this model					

Test your model PREVIEW

ⓘ   In order to test your model, you will need to deploy it first. Pricing

---

Export model   🔗 LEARN

The **TensorFlow Lite (.tflite)** format allows you to run your model on mobile and embedded devices.

1. Export your model as a TF Lite package.

   ┌───┐
   │ Destination folder on Cloud Storage *   │
   │ ✅ gs:// d6_dice_edge   BROWSE   │
   └───┘

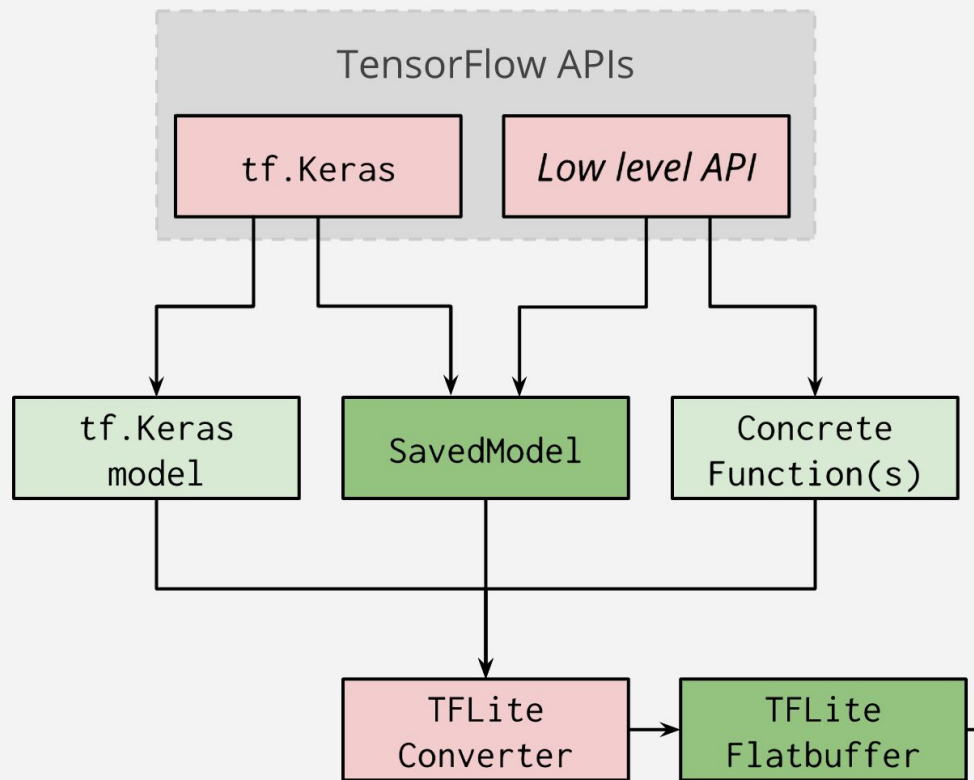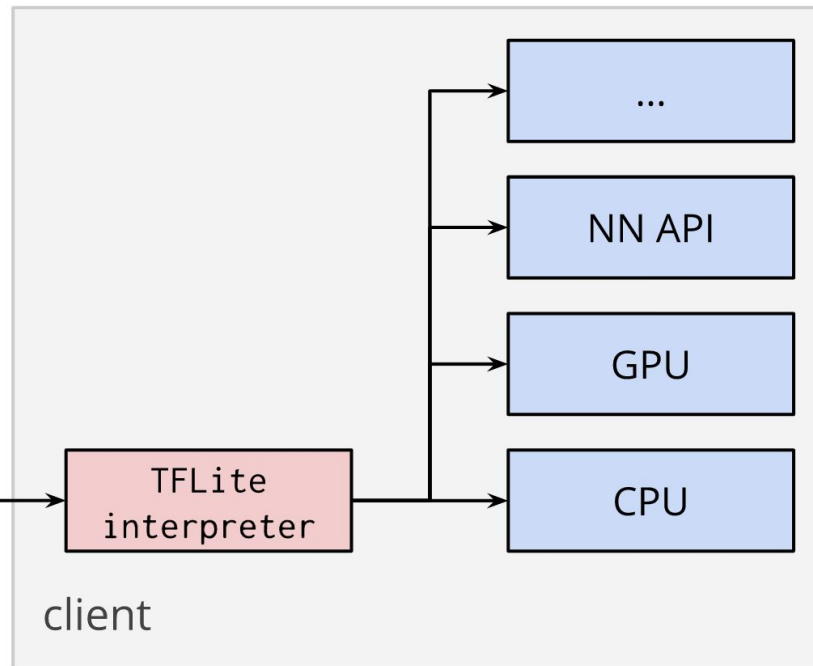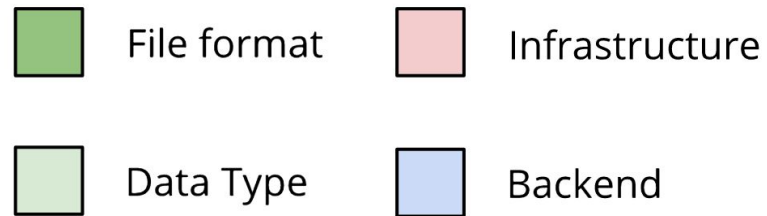   **EXPORT**   VIEW FOLDER 🔗

2. Model export takes a couple of minutes. After exporting is finished, copy the package to your computer using the following command:

   ```
   $ gsutil cp -r gs://d6_dice_edge ./download_dir
   ```
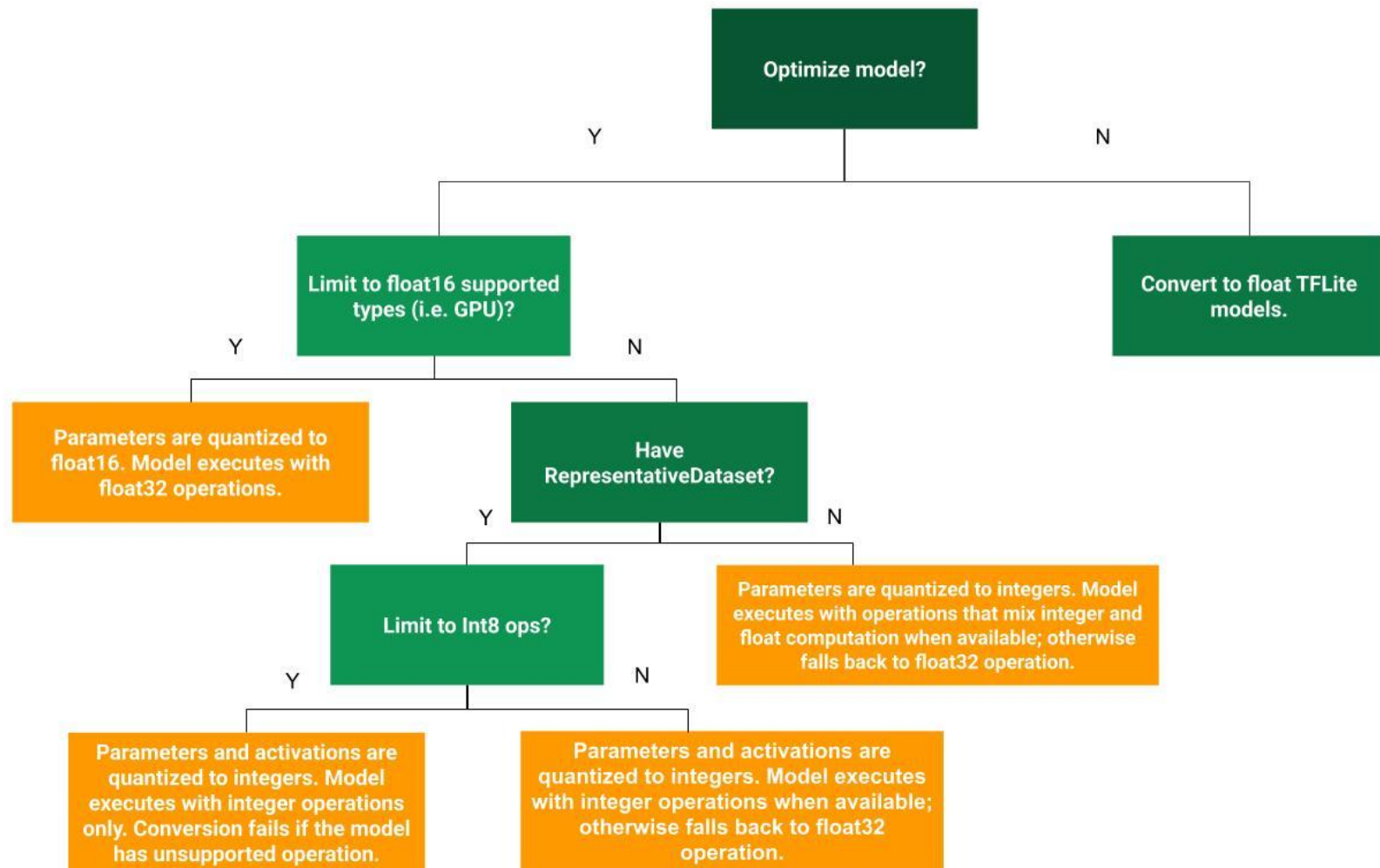
CLOSE

Post-Training Quantization Options

| TF Lite Option	Technique Used	Benefits	Hardware
OPTIMIZE_FOR_SIZE	"Hybrid operations"	4x smaller, 2-3x speedup, accuracy	CPU
DEFAULT	Integer Quantization	4x smaller, More speedup	CPU, Edge TPU, etc.

THE LABELS ARE IN THE MODEL FILE

TFLite Android Example App

TensorFlow Lite Object Detection Android Demo

https://github.com/tensorflow/examples/tree/master/lite/examples/object_detection/android
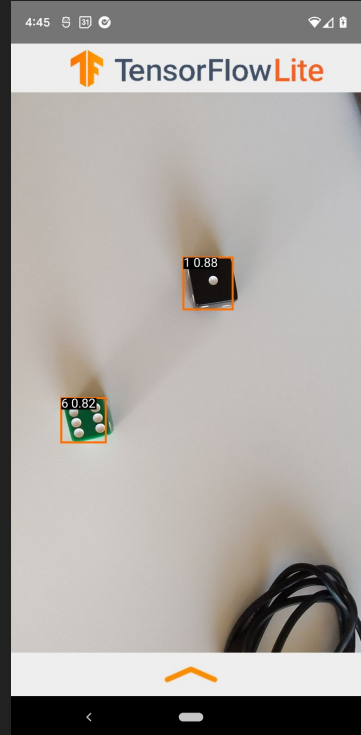
```diff
+++ b/android/app/src/main/java/org/tensorflow/lite/examples/objectdetection/ObjectDetectorHelper.kt
@@ -82,14 +82,7 @@ class ObjectDetectorHelper(

         optionsBuilder.setBaseOptions(baseOptionsBuilder.build())

-        val modelName =
-            when (currentModel) {
-                MODEL_MOBILENETV1 -> "mobilenetv1.tflite"
-                MODEL_EFFICIENTDETV0 -> "efficientdet-lite0.tflite"
-                MODEL_EFFICIENTDETV1 -> "efficientdet-lite1.tflite"
-                MODEL_EFFICIENTDETV2 -> "efficientdet-lite2.tflite"
-                else -> "mobilenetv1.tflite"
-            }
+        val modelName = "model.tflite"
```
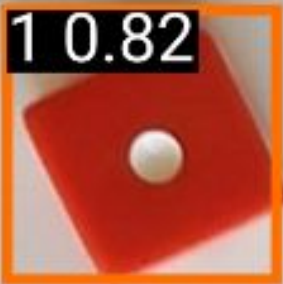
Links

- Object Detection
    - [Object Detection in 20 Years: A Survey](#)
    - [Object Detection using YOLOv5 and OpenCV DNN in C++ and Python](#)
    - [YOLO v5](#)
    - [YOLO Algorithm and YOLO Object Detection](#)
    - [COCO Dataset](#)

- [Vertex AI](#)
    - [Vertex AI NAS Announcement 11/30/2021](#)
    - [AutoML Beginner's Guide](#)

- TensorFlow and TensorFlow Lite
    - [TensorFlow Lite Object Detection Android Demo](#)
    - [Easier object detection on mobile with TensorFlow Lite](#)
    - [TensorFlow post-training quantization](#)

- Dice Detection
    - [Six-sided Dice Dataset](#)
    - [Real-time dice detection and classification](#)

Thank you!