

## [Cloud AutoML Vision Object Detection](#)

# Edge model creation via the UI quickstart

This quickstart walks you through the process of:

- Using AutoML Vision Object Detection to create your dataset, train a custom AutoML Vision Edge model, and make a prediction.
- Exporting and using your AutoML Vision Edge model.

In this quickstart you use the [user interface \(UI\)](https://console.cloud.google.com/vision) (<https://console.cloud.google.com/vision>) to interact with the Cloud AutoML API and export your AutoML Vision Edge model.

## Set up your project

Before you can use Cloud AutoML Vision Object Detection, you must create a Google Cloud project (one never used with any other AutoML product) and enable Cloud AutoML Vision Object Detection for that project.

1. [Sign in](https://accounts.google.com/Login) (<https://accounts.google.com/Login>) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (<https://accounts.google.com/SignUp>).

2. In the Cloud Console, on the project selector page, select or create a Google Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[GO TO THE PROJECT SELECTOR PAGE](https://console.cloud.google.com/projectselector) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/PROJECTSELECT](https://console.cloud.google.com/projectselector)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](https://cloud.google.com/billing/docs/how-to/modify-project) (<https://cloud.google.com/billing/docs/how-to/modify-project>).
4. Enable the Cloud AutoML and Storage APIs.

[ENABLE THE APIS](https://console.cloud.google.com/flows/enableapi?apiid=storage) ([HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=STORAGE-](https://console.cloud.google.com/flows/enableapi?apiid=storage)

## Beta requirements

1. This Beta feature requires you to use `us-central1` as the specified region.

## Preparing a dataset

In this quickstart you use a dataset created from [Open Images Dataset V4](https://storage.googleapis.com/openimages/web/index.html) (<https://storage.googleapis.com/openimages/web/index.html>). This publicly available Salads dataset is located at `gs://cloud-ml-data/img/openimage/csv/salads_ml_use.csv`.

The CSV format is as follows:

```
TRAIN,gs://cloud-ml-data/img/openimage/3/2520/3916261642_0a504acd60_o.jpg,Salad,0.0,
VALIDATE,gs://cloud-ml-data/img/openimage/3/2520/3916261642_0a504acd60_o.jpg,Seafood,0.0,
TEST,gs://cloud-ml-data/img/openimage/3/2520/3916261642_0a504acd60_o.jpg,Tomato,0.0,
```



3916261642\_0a504acd60\_o.jpg

Each row corresponds to an object localized inside a larger image, with each object specifically designated as test, train, or validation data. The three lines included here indicate **three distinct**

**objects located inside the same image** available at `gs://cloud-ml-data/img/openimage/3/2520/3916261642_0a504acd60_o.jpg`. Each row has a different label: **Salad, Seafood, Tomato**, in addition to other rows with **Baked goods** or **Cheese** labels.

Bounding boxes are specified for each image using the top left and bottom right vertices:

- (0,0) corresponds to the top left-most vertex.
- (1,1) corresponds to the bottom right-most vertex.

For the first row shown above, the (x, y) coordinates for the top left vertex of the **Salad** labeled object are (0.0,0.0954), and the coordinates for the bottom right vertex of the object are (0.977,0.957).

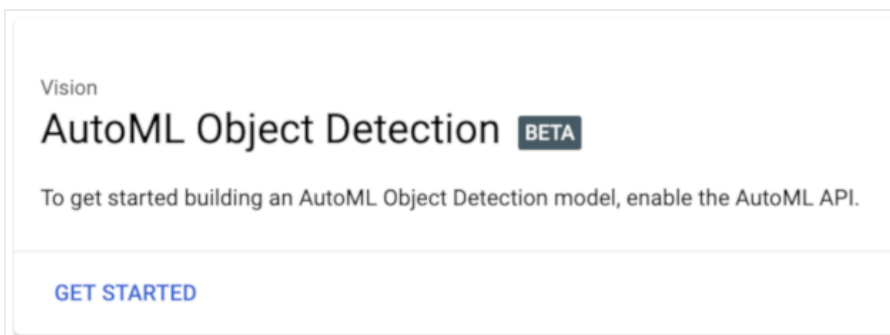
For more detailed information on how to format your CSV file and the minimum requirements for creating a valid dataset, see [Preparing your training data](https://cloud.google.com/vision/automl/object-detection/docs/prepare) (<https://cloud.google.com/vision/automl/object-detection/docs/prepare>).

## Create your dataset

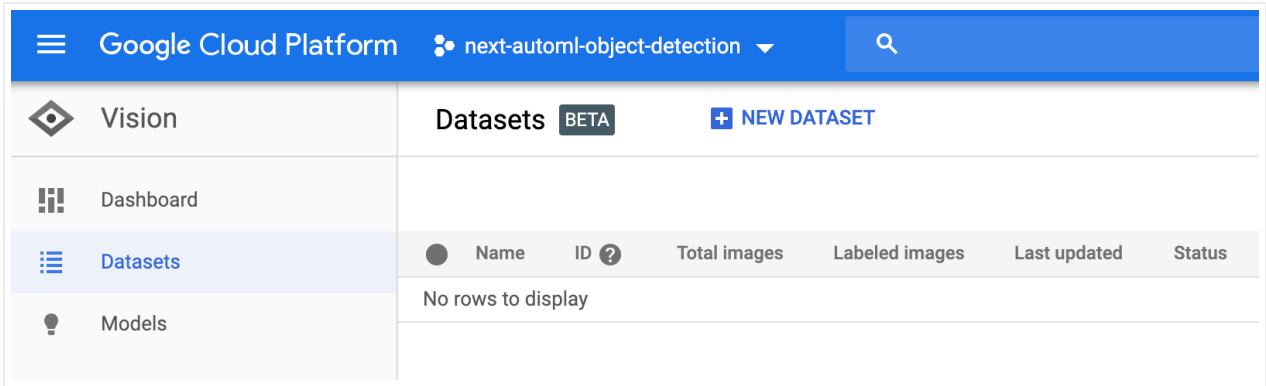
1. Open the [Cloud AutoML Vision Object Detection UI](https://console.cloud.google.com/vision) (<https://console.cloud.google.com/vision>) and select your project from the drop-down list in the title bar.

The first time you open the Cloud AutoML Vision Object Detection UI you will need to **"Enable AutoML API"** if it isn't already enabled.

2. Select **"Get started"** in the pop-up window when prompted.



3. You will be taken to the datasets list page. Create a new dataset by selecting **New Dataset**.



Enter a unique name for your dataset.

## Create new dataset

**Dataset name \***

Use letters, numbers and underscores up to 32 characters.

CANCEL
CREATE DATASET

4. Enter the location of the training data to import in the window that follows.

In the **Select a CSV file on Cloud Storage** text box, enter the path for the sample CSV file (the `gs://` prefix is added automatically):

```
cloud-ml-data/img/openimage/csv/salads_ml_use.csv
```

Alternatively, you can select **Browse** and navigate to the CSV file in one of your Google Cloud Storage buckets.

This quickstart uses sample data staged in a public Google Cloud Storage bucket. The training data are sample JPG images annotated with the bounding box and label of the objects you want the model to learn to identify. To import the training data into the dataset, use a CSV file that points to the image (JPEG, PNG, GIF, BMP, or ICO) files; see [Preparing your training data](#)

(<https://cloud.google.com/vision/automl/object-detection/docs/prepare>) for information about the format and image specifications.

← untitled\_1563557657669 **BETA** **|| LABEL STATS** **EXPORT DATA**

**IMPORT** IMAGES TRAIN EVALUATE TEST & USE

## Import images

AutoML Object Detection uses your images to train a custom machine learning model. [Learn more about preparing your data](#)

- Upload labels in your CSV, or upload un-labeled images, and use our labeling tool.
- If you are uploading images with bounding boxes, each bounding box should be assigned a label.
- At least 100 bounding boxes per label is recommended.

### Select a CSV on Cloud Storage

Each row in your CSV should be a Cloud Storage file path to a training or test image. Images can be in JPG, PNG, or GIF format. [CSV formatting help](#)

Example CSV row:

```
[set, ]image_path[, label, x1, y1, , , x2, y2]
TRAIN,gs://folder/image1.png,car,0.1,0.1,,,0.3,0.3,,
TRAIN,gs://folder/image1.png,bike,.7,.6,,, .8,.9,,
UNASSIGNED,gs://folder/im2.png,car,0.1,0.1,0.2,0.1,0.2,0.3,0.1,0.
TEST,gs://folder/im3.png,,,,, ,
```

gs:// \*

cloud-ml-data/img/openimage/csv/salads\_ml\_use.csv **BROWSE**

**IMPORT**

## 5. Select **Import**.

Your dataset will show a **Status** of **Running:importing images** while your images are being imported. This process takes just a few minutes.

When your training data has been successfully imported, the **Status** column says **Success:Creating dataset** and the UI shows the generated **ID** for the dataset (used when making AutoML API calls) as well as the number of items imported.

Vision		Datasets <span>BETA</span> <a href="#">+ NEW DATASET</a>				
<a href="#">Dashboard</a>						
<a href="#">Datasets</a>						
<a href="#">Models</a>						
Name	Type	Total Images	Labeled Images	Last updated	Status	
<b>untitled_1563557657669</b> IOD6620964353350828032	Object detection	0	0	Jul 19, 2019, 10:38:06 AM	Running: Importing images	
<b>docs_july1_fishfooding</b> IOD8163482410097311744	Object detection	225	225	Jul 18, 2019, 1:55:01 PM	Success: Training model	

If there are any problems importing images you will see **Warning: Importing images** as the status. Select the dataset name and **Details** to see the errors with specific image import.

### Error details

- Error: gs://cloud-ml-data/img/openimage/csv/salads\_ml\_use.csv line 782: The bounding box (0.57, 0.21) - (0.58, 0.22) is too small, 8px \* 8px is the minimum size required.
- Error: gs://cloud-ml-data/img/openimage/csv/salads\_ml\_use.csv line 281: The bounding box (0.86, 0.82) - (0.87, 0.83) is too small, 8px \* 8px is the minimum size required.
- Error: gs://cloud-ml-data/img/openimage/csv/salads\_ml\_use.csv line 1642: The bounding box (0.00, 0.55) - (0.00, 0.61) is too small, 8px \* 8px is the minimum size required.

[CLOSE](#)

## Train your model

1. Once your dataset has been created and processed, select the **Train** tab to initiate model training.


2. Select **Start training** to continue.

Unlabeled images aren't used. Your dataset will be automatically split into [Train, Validation, and Test sets](#).

Ideally, each label should have at least **10 bounding boxes**. Fewer bounding boxes often result in inaccurate precision and recall. You must also have at least **8, 1, 1 bounding boxes** each assigned to your Train, Validation and Test sets.

Labels	Bounding boxes	Train	Validation	Test
Baked Goods	85	66	3	16
Cheese	477	377	49	51
Salad	241	184	24	33
Seafood	148	114	12	22
Tomato	803	604	103	96

This will open a side window with training options.

3. From the training **Define your model** section, change the model name and select  **Edge** as the model type. After selecting to train an Edge model select **Continue**.

## Train new model

---

### 1 Define your model

Model name \*

iod\_edge\_20190718021144

- Cloud hosted  
Host your model on Google Cloud for online predictions
- Edge  
Download your model for offline/mobile use

CONTINUE

### 2 Optimize model for

### 3 Set a node hour budget

START TRAINING

CANCEL

4. In the following **Optimize model for** section, select the default  **Best trade-off** option. After selecting the optimization criteria select **Continue**.



## Train new model

### 1 Define your model

### 2 Optimize model for

	Goal	Package size	Accuracy	Latency for Google Pix
<input type="radio"/>	Higher accuracy	2.8 MB	Higher	360 ms
<input checked="" type="radio"/>	Best trade-off	2.8 MB	Medium	150 ms
<input type="radio"/>	Faster predictions	2.8 MB	Lower	56 ms

Please note that prediction latency estimates are for guidance only. Actual latency will depend on your network connectivity.

CONTINUE

### 3 Set a node hour budget

START TRAINING

CANCEL

5. In the following **Set a node hour budget** section use the recommended node hour budget (24).

## Train new model

### ✓ Define your model

### ✓ Optimize model for

### 3 Set a node hour budget

Specify the maximum number of node hours to spend training your model. If your model stops improving before the, AutoML Vision will stop training and you'll only be charged for the actual node hours used.

For edge models: You can train for 20 compute hours (per billing account) for free. Standard pricing applies afterwards. [Pricing guide](#)

Budget \*  
24

START

✕

Recommended node hours by image count

Image count	Node hours	Wall clock hours
< 1,000	1 - 20	2
1,000 - 10,000	20 - 100	10
> 10,000	100 - 216	24

**!** **Node hour budget:** Recommended training time is calculated based on:

- model learning curves
- training dataset size

If model training converges *before* recommended or custom time selected the system allows for early stopping; this means you are *only charged for the time it takes to train the model*. **You are encouraged to use the recommended amount of hours to avoid using more resources than necessary and reduce billing costs.**

## 6. Select **Start training** to begin model training.

Training is initiated for your model, and should take about an hour. The training might stop earlier than the node hour you selected. The service will email you once training has completed, or if any errors occur.

Once training is complete, you can refer to evaluation metrics, as well as test and use the model.

Select the **Evaluate** tab to get more details on [F1, Precision, and Recall](https://cloud.google.com/vision/automl/object-detection/docs/evaluate) (https://cloud.google.com/vision/automl/object-detection/docs/evaluate) scores.

Select an individual label under **Filter labels** to get details on true positives, false negatives and false positives.

## Make a Prediction

Once training is complete your model must be manually deployed before you can make online predictions with the model.

Select the **Test & use** tab to deploy (and undeploy) your model. The **Test & use** page also has instructions on sending an image to your model for a prediction. You can also refer to [Annotating images](https://cloud.google.com/vision/automl/object-detection/docs/predict) (https://cloud.google.com/vision/automl/object-detection/docs/predict) for examples.

## Export and use the Edge model

The final step in using an AutoML Vision Edge model is to export (optimize and download) and use your model.

There are multiple ways you can export and use your models for prediction on Edge devices.

**Model and export options:** [Tensorflow Lite](https://cloud.google.com/vision/automl/object-detection/docs/deploy#tensorflow_lite_models)

(https://cloud.google.com/vision/automl/object-detection/docs/deploy#tensorflow\_lite\_models) is optimized for mobile phones and ARM (mobile device operating system) devices. It can further be compiled to run on [Edge TPU](https://cloud.google.com/vision/automl/object-detection/docs/deploy#export_to_edge_tpu)

(https://cloud.google.com/vision/automl/object-detection/docs/deploy#export\_to\_edge\_tpu). Another

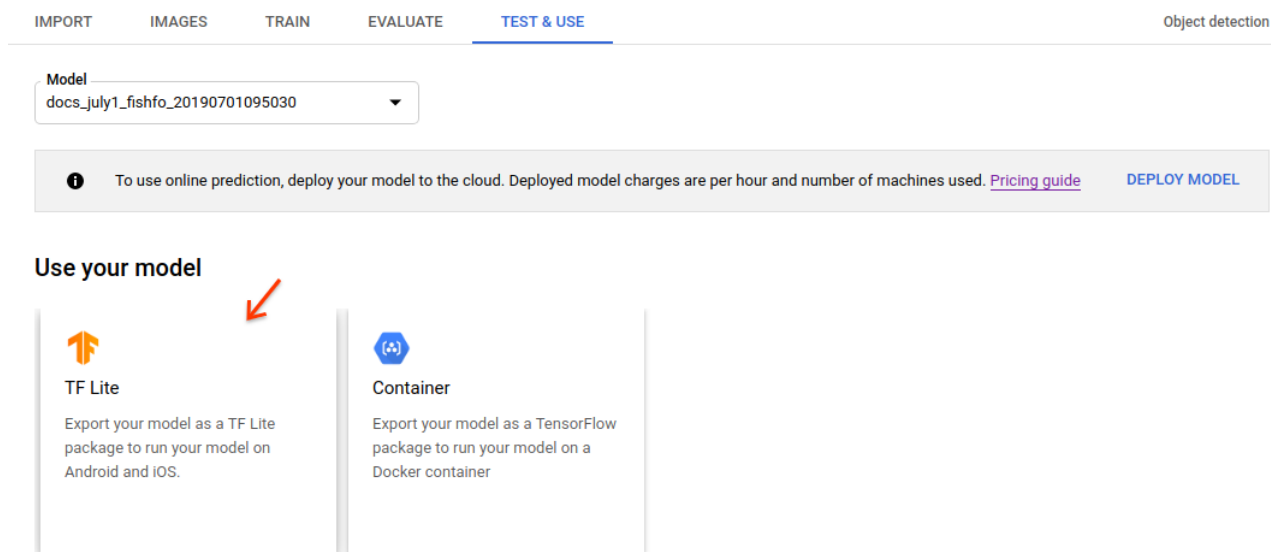
export option - [Tensorflow SavedModel](#)

([https://cloud.google.com/vision/automl/object-detection/docs/deploy#deployment\\_to\\_a\\_container](https://cloud.google.com/vision/automl/object-detection/docs/deploy#deployment_to_a_container)) - can be used in a [Docker container](#)

(<https://cloud.google.com/vision/automl/object-detection/docs/containers-gcs-tutorial>) for serving.

In this quickstart you will use Tensorflow Lite (TF Lite) as an example. TF Lite models are both easy to use and have a wide set of use cases.

### 1. Under **Use your model**, select the **TF Lite** option.



IMPORT IMAGES TRAIN EVALUATE **TEST & USE** Object detection

Model  
docs\_july1\_fishfo\_20190701095030

**i** To use online prediction, deploy your model to the cloud. Deployed model charges are per hour and number of machines used. [Pricing guide](#) [DEPLOY MODEL](#)

#### Use your model

**TF Lite**  
Export your model as a TF Lite package to run your model on Android and iOS.

**Container**  
Export your model as a TensorFlow package to run your model on a Docker container

2. In the side window that opens indicate a Google Cloud Storage location for your exported model. After you have given a storage location, select **Export** to export a TF Lite package into your Cloud Storage storage bucket. The export process typically takes several minutes.

## TF Lite

---

### Use your on-device model

The **Tensorflow Lite (.tflite)** format allows you to run your model on mobile and embedded devices. [Android](#) [iOS](#)

1. Download a sample camera app. It includes instructions for implementing your model on your device
2. Export your model as a TF Lite package

Select a destination

iod\_edge\_export\_bucket BROWSE

EXPORT OPEN IN GCS

```
$ gsutil cp -r gs://iod_edge_export_bucket ./download_dir
```

3. After the export is completed select the "**Open in Google Cloud Storage**" link to go to the Google Cloud Storage destination folder in the Google Cloud Platform Console.

In the Google Cloud Storage destination location you will find a folder named with timestamp and model format, under which you can find the following files:

- a tflite file (`model.tflite`),
- a dictionary file (`dict.txt`)
- a metadata file (`tflite_metadata.json`)

## Cleanup

### ▼ Undeploy & delete instructions

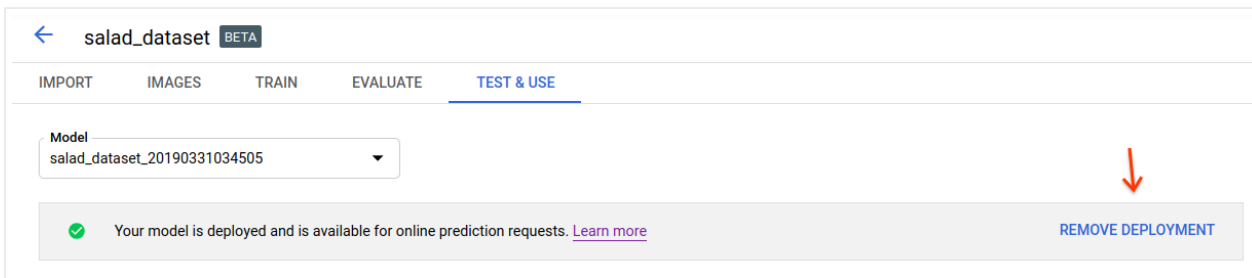
If you no longer need your custom model or dataset, you can delete them.

To avoid unnecessary Google Cloud Platform charges, use the [GCP Console](https://console.cloud.google.com/vision) (<https://console.cloud.google.com/vision>) to delete your project if you do not need it.

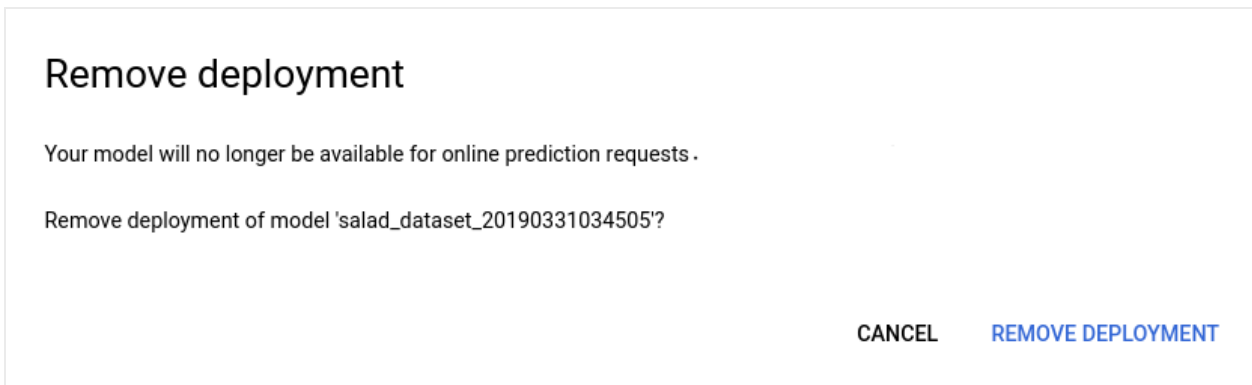
## Undeploy your model

Your model incurs charges while it is deployed.

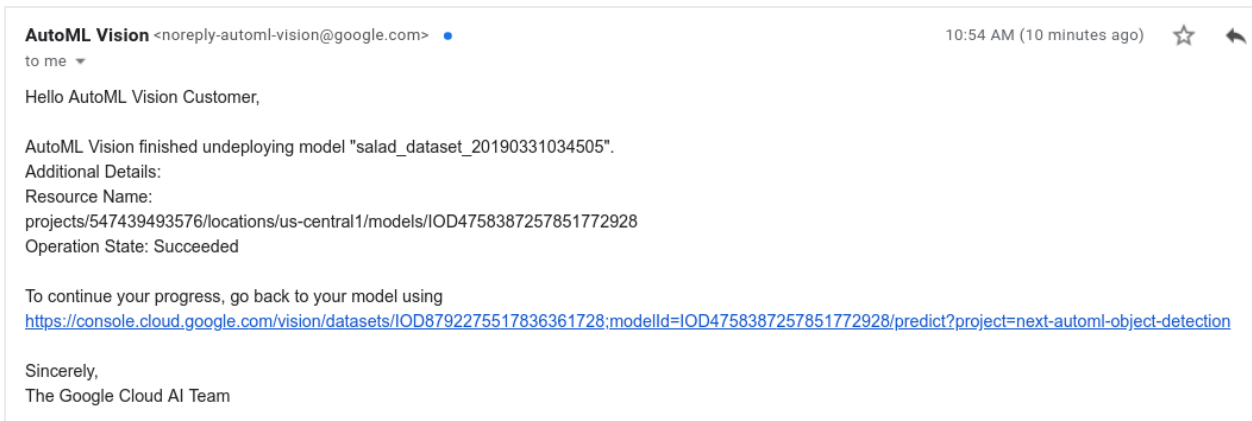
1. Select the **Test & Use** tab just below the title bar.
2. Select **Remove deployment** from the banner beneath your model name to open the undeploy option window.



3. Select **Remove deployment** to undeploy the model.



4. You will receive an email when model undeployment has completed.



## Delete your project (optional)

To avoid unnecessary Google Cloud Platform charges, use the [Cloud Console](https://console.cloud.google.com/) (<https://console.cloud.google.com/>) to delete your project if you do not need it.

## What's Next

With these files, you can follow tutorials to use on [Android devices](https://www.tensorflow.org/lite/models/object_detection/overview) ([https://www.tensorflow.org/lite/models/object\\_detection/overview](https://www.tensorflow.org/lite/models/object_detection/overview)), [iOS devices](https://cloud.google.com/vision/automl/object-detection/docs/tflite-ios-tutorial) (<https://cloud.google.com/vision/automl/object-detection/docs/tflite-ios-tutorial>), or [Raspberry Pi 3](https://www.tensorflow.org/lite/guide/build_rpi) ([https://www.tensorflow.org/lite/guide/build\\_rpi](https://www.tensorflow.org/lite/guide/build_rpi)).

### Other options to use the model:

- You can also export the model as TensorFlow SavedModel and use it with a Docker container in the **Container** tab. See the [container tutorial](https://cloud.google.com/vision/automl/object-detection/docs/containers-gcs-tutorial) (<https://cloud.google.com/vision/automl/object-detection/docs/containers-gcs-tutorial>) on how to export to a container.
- You can export the model for running on Edge TPU in the **Edge devices** tab. You can follow this [Edge TPU official documentation](https://coral.withgoogle.com/docs/) (<https://coral.withgoogle.com/docs/>) to use the model.

---

*Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated November 20, 2019.*