Cloud AutoML Vision Object Detection

Preparing your training data

Preparing your images

	General image requirements
Supported file types	• JPEG • PNG • GIF • BMP • ICO
Types of images	AutoML Vision Object Detection models are optimized for photographs of objects in the real world. ★ **Note:** AutoML Vision Object Detection does not currently satisfy the requirements to be considered compliant with the Health Insurance Portability and Accountability Act (HIPAA).
Training image file size (MB)	30MB maximum size.
Prediction image file* size (MB)	1.5MB maximum size.
Image size (pixels)	1024 pixels by 1024 pixels suggested maximum. For images much larger than 1024 pixels by 1024 pixels some image quality may be lost during AutoML Vision Object Detection's image normalization process.

* **Note:** The AutoML API currently only supports sending base64-encoded image content to the `predict` method. For an example, see Make a prediction (https://cloud.google.com/vision/automl/object-detection/docs/predict). For general information on encoding an image, see the base64 encode (https://cloud.google.com/vision/automl/object-detection/docs/base64) how-to topic.

	Labels and bounding box requirements
Label instances for training	10 annotations (instances) minimum.
Annotation requirements	For each label you must have *at least* 10 images, each with *at least* one annotation (bounding bo However, for model training purposes it's recommended you use about **100 annotations per labe** have the better your model will perform.
Label ratio (most common label to least common label):	The model works best when there are at most 100x more images for the most common label tha For model performance, it is recommended that you remove very low frequency labels.
Bounding box edge length	At least **0.01 * length of a side** of an image. For example, a 1000 * 900 pixel image would require
Bounding box size (pixels)	8 pixels by 8 pixels minimum.
Bounding boxes per distinct image	500 maximum.
Bounding boxes returned from a prediction request	100 (default), 500 maximum. You can specify this value as part of the `predict` request in the `params.max_bounding_box_c` (https://cloud.google.com/automl/docs/reference/rest/v1/projects.locations.models/predict#

	Training data and dataset requirements
Training image characteristics	The training data should be as close as possible to the data on which predictions are to be made. For example, if your use case involves blurry and low-resolution images (such as from a security camera), your training data should be composed of blurry, low-resolution images. In

	general, you should also consider providing multiple angles, resolutions, and backgrounds for your training images. AutoML Vision Object Detection models can't generally predict labels that humans can't assign. So, if a human can't be trained to assign labels by looking at the image for 1-2 seconds, the model likely can't be trained to do it either.
Images in each dataset	150,000 maximum
Total annotated bounding boxes in each dataset	1,000,000 maximum
Number of labels in each dataset	1 minimum, 1,000 maximum

Best practice guide

What kind of image data can you use?

- Supported image file formats: JPEG, PNG, GIF, BMP, or ICO.
- The training data should be *as close as possible* to the data on which predictions are to be made. For example, if your use case involves low-resolution images from cellphone cameras, your training data should be composed of low-resolution images. In general, you should also consider providing multiple angles, resolutions, and backgrounds for your training images. Another example is if you want to detect regions in high resolution images, do not train the model with cropped images.

What are the minimal conditions on data?

- The labels you use must be valid strings (no comma inside). The comma is only an issue in CSV-based importing. A way to address this issue is: `"file_comma,path","label,comma",0,0,,,1,1,,`.
- The bounding boxes should be larger than 8 by 8 pixels in all cases. Bounding boxes that are smaller than this will be filtered.

- Images may exceed 1024*1024 pixels, but those images will be downscaled automatically, possibly leading to a loss in image quality. For this reason we recommend a maximum image size of 1024*1024 pixels. Images *smaller* than these dimensions will not be upscaled.
- All the bounding boxes should be *inside* images.
- The bounding boxes should be exhaustively labeled: if there are two cars in an image, all of them should be labeled.

How big does the dataset need to be?

- The more, the better. This is almost always true. However, an exception to this would be if adding more samples leads to imbalance or leakage (see below (#common-issues)).
- The amount of data needed to train a good model depends on different factors:
  - **The amount of classes**. The more unique classes you have, the more samples per class are needed.
  - **Complexity/diversity of classes**. It's similar to humans: A human can probably quickly learn to distinguish between beer and wine, with just a few samples. He will have to try different wines quite more often to distinguish between 5-6 different sorts of red wines, and for many humans it will be challenging to learn to distinguish between 50 different flavors of red wines. At least one will have to practice a lot. Similarly, neural networks would quickly be able to distinguish between elephants and cats, but they would need many more samples to classify 30 different animals.
- As a rule of thumb, we recommend to have *at least* 100 training samples per class if you have distinctive and few classes, and more than 200 training samples if the classes are more nuanced and you have more than 50 different classes.

Training vs. evaluation datasets

When training machine learning models you typically divide the dataset usually into three separate datasets:

1. a training dataset
2. a validation dataset

3. a test dataset

Test, train, and validation sets are often conceptually referred to as "dataset splits".

A training dataset is used to build a model. The model being trained tries multiple hyper-parameters while searching for patterns in the training data. During the process of pattern identification, AutoML Vision Object Detection uses the *validation dataset* to test the hyperparameters of the model. AutoML Vision Object Detection chooses the best-performing algorithms and patterns from all options identified during the training stage.

After the best performing algorithms and patterns have been identified they are tested for error rate, quality, and accuracy using the *test dataset*. Customers must have a separate test dataset that they can use to test the model independently. This test dataset is either specified in the training set by the user or chosen automatically at training time.

Both a validation and a test dataset are used in order to avoid bias in the model. During the validation stage, optimal model parameters are used. Using these optimal model parameters can result in biased metrics. Using the *test dataset* to assess the quality of the model *after* the validation stage provides the training process with an unbiased assessment of the quality of the model.

If you manually choose dataset samples, you should structure datasets in a way that represents the same population. Similarly, you should create dataset splits that have similar images, all with a similar distribution of labels.

Manual and automatic dataset splits

You can manually specify (https://cloud.google.com/vision/automl/object-detection/docs/csv-format) the split of training, validation, and test when importing datasets in a CSV file.

If you do not specify it, AutoML Vision Object Detection will randomly split your data. Splits are created in the following manner:

- 80% of images are used for training.
- 10% of images are used for hyper-parameter tuning and/or to decide when to stop training.
- 10% of images are used for evaluating the model. These images are not used in training.

The maximum size of a test dataset is 50,000 images, even if 10% of the total dataset exceeds that maximum.

Common issues

- **Imbalanced data**: In many cases the number of samples per class (label) is not equal. Minor imbalances don't generally create issues, but larger discrepancies between classes can cause an issue. When there is a bigger imbalance, eg. some classes are represented more than 10 times that of other classes, this becomes problematic for model building. While there are approaches to counteract class imbalances, it's not an ideal configuration for model training. When possible, try to avoid model training with highly imbalanced data.

  As a general rule, keep the ratio between the *most* common and the *least* common classes below 2 to 1.

- **Bad splits**: When you provide training data, AutoML Vision Object Detection can automatically split it into training, validation, and testing datasets. You can also assign the train split labels yourself as well.

  There is no guarantee that you will get the same split if you import the same data multiple times.

  The training, validation and testing data should not have strong correlation. For example, a common bad case is that when the images are from videos, leading many images to be very similar to each other. If you let the system randomly split the dataset for you it will be very likely that you will get highly similar images in both training and validation/testing data. This will lead to an incorrectly obtained high accuracy on testing data.

- **Data leakage**: Data leakage is a significant problem that can bias models. Data leakage happens when the algorithm is able to use information during model training that it should not, and that will not be available during future predictions. It leads to overly optimistic results on train, validation, and potentially test datasets. However, this performance might not be as good on some future unseen data. This happens often unintentionally, and requires special care during data preparation.

  Leakage examples: Positives and negatives from different image sources, or viewing angles.

Last updated December 5, 2019.