Cloud AutoML Vision Object Detection

Formatting a training data CSV

After preparing training data (https://cloud.google.com/vision/automl/object-detection/docs/prepare) that is sufficiently representative and uploading those images to Google Cloud Storage you are ready to create a CSV file with bounding boxes and labels for image import into a dataset.

This page describes how you format the CSV file.

CSV formatting guidelines

To use the `importData` method, both the CSV file *and* the images it points to must be in a Google Cloud Storage bucket.

Additionally, the CSV file must also fulfill the following requirements:

- The file can have any filename, but must be in the same bucket as your image file.
- Must be UTF-8 encoded.
- Must end with a `.csv` extension.
- Has one row **for each bounding box** in the set you are uploading, ***or*** one row **for each image with no bounding box** (such as row 4 below).
- Contain **one image per line**; an image with multiple bounding boxes will be repeated on as many rows as there are bounding boxes.

  For example, rows 1 and 2 reference the same image that has 2 annotations (`car,0.1,0.1,,,0.3,0.3,,` and `bike,.7,.6,,,.8,.9,,`). Row 3 refers to an image that has only 1 annotation (`car,0.1,0.1,0.2,0.1,0.2,0.3,0.1,0.3`), while row 4 references an image with no annotations.

Four sample rows:

```
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,0.1,0.1,0.2,0.1,0.2,0.3,0.1,0.3
TEST,gs://folder/im3.png,,,,,,,,,
```

Each row has these columns:

1. **Which set to assign the content in this row to.** This field is required and can be one of these values:
    - `TRAIN` - Use the image to train the model. This set should be the largest, as these images will be used to build your model.
    - `VALIDATE` - Use the image to validate the results that the model returns during training (also known as "dev" datasets). These images will be used by AutoML Vision Object Detection to determine when to stop the model training process.
    - `TEST` - Use the image to quantify the model's results after the model has been trained (also referred to as "holdout" data). These images are used for evaluation after a model has been created using the previous two sets.
    - `UNASSIGNED` - These images are assigned to one of the above three sets by AutoML Vision Object Detection. Use this set tag if you have no preference about which set the images are placed in.

> If you indicate sets in your CSV you must satisfy the requirement that not all images can belong to the same set ***unless*** it is the `UNASSIGNED` set; **if you only specify a single fixed set in your CSV (all images as `TRAIN` images, for example) you encounter an error.** Avoid this error by, *at a minimum*, specifying some images to your target fixed set (`TRAIN`, `VALIDATE`, or `TEST`) AND specifying the remaining images to the `UNASSIGNED` set. This allows AutoML API to split the unassigned images into the two sets not represented in the CSV file.

2. **The content to be annotated.** This field contains Google Cloud Storage URI for the image. Google Cloud Storage URIs (https://cloud.google.com/storage/docs/listing-objects) are case-sensitive.
3. **A label that identifies how the object is categorized.** Labels must start with a letter and only contain letters, numbers, and underscores. AutoML Vision Object Detection also allows you to use labels with white spaces.

    After training image import you can also manually label in the UI (https://cloud.google.com/vision/automl/object-detection/docs/label#annotating_images_via_the_web_ui) , or use Google's Human Labeling (https://cloud.google.com/vision/automl/object-detection/docs/label#human-labeling) service to label training images.
4. **A bounding box for an object in the image.** The bounding box for an object can be specified in two ways:

- either with only 2 vertices (consisting of a set of x,y coordinates) if they are diagonally opposite points of the rectangle (`x_relative_min`, `y_relative_min`,,,`x_relative_max`,`y_relative_max`,,),
- or with all 4 vertices(`x_relative_min`,`y_relative_min`,`x_relative_max`,`y_relative_min`,`x_relative_max`,`y_relative_max`,`x_relative_min`,`y_relative_max`).

Each `vertex` is specified by x, y coordinate values. These coordinates must be a float in the 0 to 1 range, where 0 represents the minimum x or y value, and 1 represents the greatest x or y value.

For example, (0,0) represents the top left corner, and (1,1) represents the bottom right corner; a bounding box for the entire image is expressed as (0,0,,,1,1,,), or (0,0,1,0,1,1,0,1).

AutoML API does not require a specific vertex ordering. Additionally, if 4 specified vertices don't form a rectangle parallel to image edges, AutoML API calculates and uses vertices that do form such a rectangle.

Note: You can use Cloud Vision API's Object Localizer (https://cloud.google.com/vision/docs/object-localizer) feature to help build your dataset by getting more generalized labels and bounding boxes for objects in an image.

For example:

- **Not assigned to a set**: `UNASSIGNED,gs://my-storage-bucket/img/salad_089.jpg,Baked goods,0.56,0.25,,,0.97,0.50,,`
- **Assigned to a set**: `TRAIN,gs://my-storage-bucket/img/salad_089.jpg,Baked goods,0.56,0.25,,,0.97,0.50,,`

The above rows use the following format, since the API assumes exactly two points are rectangular diagonal vertices (top left vertex, bottom right vertex):

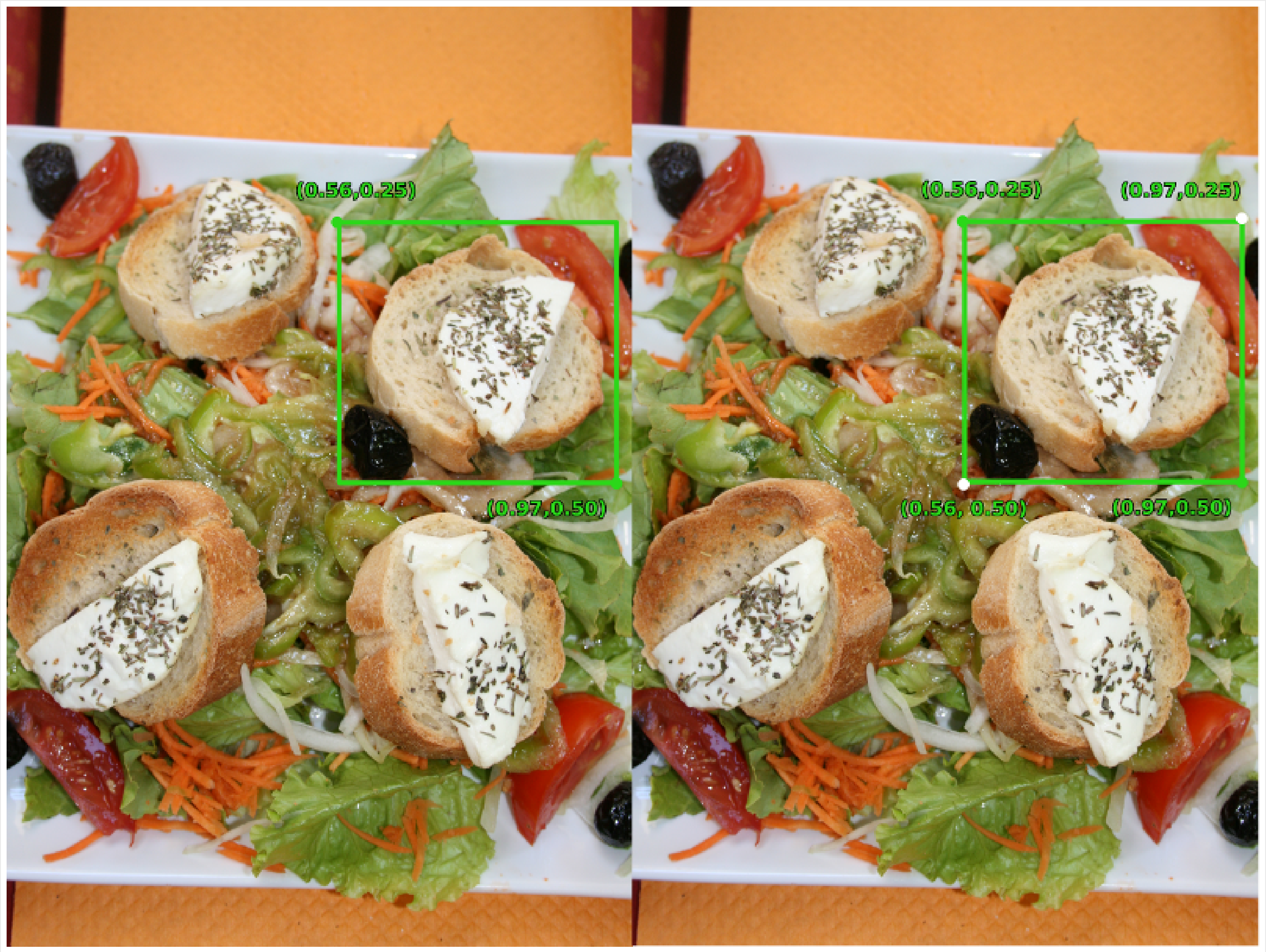
```
set,path,label,x_min,y_min,,,x_max,y_max,,
```

The following format is also valid because it conveys the same information:

```
set,path,label,x_min,y_min,x_max,y_min,x_max,y_max,x_min,y_max
```

This means the "Assigned to a set" row shown above can also be expressed by indicating all 4 vertices:

```
TRAIN,gs://my-storage-bucket/img/salad_089.jpg,Baked goods,0.56,0.25,0.97,0.25,0.97,0.50,0.56,0.50
```



Save the contents as a CSV file in your Google Cloud Storage bucket.

Common errors with CSV

- Using unicode characters in labels. E.g. Japanese characters are not supported.
- Using non-alphanumeric characters in labels.
- Empty lines.
- Incorrect capitalization of Cloud Storage image paths.
- Incorrect access control configured for your image files. Your AutoML service account created when you enabled the API should have read or greater access.
- References to non-image files (such as PDF or PSD files). Likewise, files that are not image files (JPEG, PNG, GIF, BMP, or ICO) but that have been renamed with an image extension will cause an error.
- Non-CSV-formatted files.

Last updated December 5, 2019.