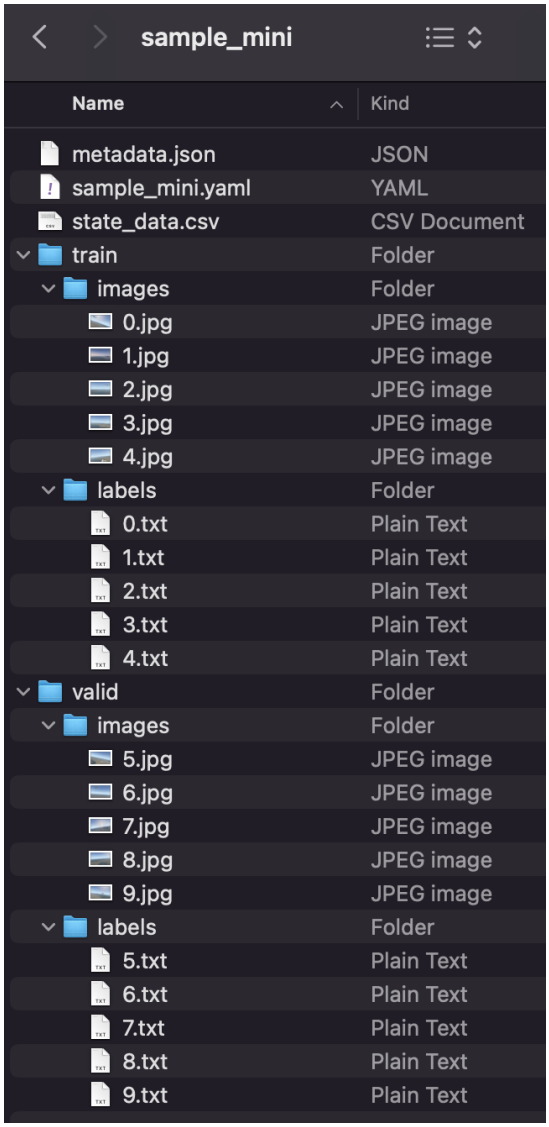
Guide to VisionBasedAircraftDAA Datasets

This document provides an overview of how the datasets generated by this repository are organized and formatted to be compatible with YOLOv5 (YOLOv5 documentation linked here). The same example dataset, called "sample_mini", is used throughout this guide. You may need to download this file to use the hyperlinks.

Dataset Overview

The dataset formatting used in this repository is designed to be compatible with YOLOv5, but some supplemental files are included to document the attributes of each image. All datasets include:

1. **metadata.json:** a supplemental file containing the command line arguments used to generate the dataset
2. **state_data.csv:** a supplemental file containing the position and environment information for each image in the dataset
3. **[DATASET].yaml:** a file containing information about the dataset required by the YOLO model
4. **train:** folder containing the images and labels for the training portion of the dataset
5. **valid:** folder containing the images and labels for the validation portion of the dataset



metadata.json

Each dataset contains a json file that includes the command line arguments used to generate the images in the dataset. For datasets that include data generated by multiple commands (using the '--append' flag), there will be multiple entries in the json structure.

The key for each entry is a string representing the range of indices for images that are represented by that metadata entry. For instance, in the example below, the key "0.9" indicates that the associated metadata is representative of images 0-9, inclusive, in the dataset. There is also a "total_images" entry that contains the total number of images in the dataset which is used for reference purposes in the code.


```
{
   "0.9": {
       "location": "Palo Alto",
       "enrange": 5000.0,
       "urange": 1000.0,
       "weather": 0,
       "daystart": 8.0,
       "dayend": 17.0,
       "num_train": 5,
       "num_valid": 5,
       "append": false,
       "datasetname": "sample_mini",
       "ac": "Cessna Skyhawk",
       "allweather": false,
       "newac": false,
       "own_h": [
           0.0,
           360.0
       ],
       "own_p_max": 30.0,
       "own_r_max": 45.0,
       "intr_h": [
           0.0,
           360.0
       ],
       "vfov": 40.0,
       "hfov": 50.0,
       "outdir": "../../datasets/sample_mini/"
   },
   "total_images": 10
}
```

state_data.csv

Each dataset folder contains a state_data.csv file containing the following information for each image in the dataset:

```
filename,e0,n0,u0,h0,p0,r0,vang,hang,z,e1,n1,u1,h1,p1,r1,intr_x,intr_y,loc,ac,clouds,loca
l_time_sec
```

- **filename:** the numeric index of the image
- **e0, n0, u0:** position of the ownship in meters east, north, and up relative to the origin location specified in constants.py
- **h0, p0, r0:** orientation of the ownship in degrees of heading, pitch, and roll
- **vang, hang:** vertical angle and horizontal angle for the position of the intruder from the perspective of the ownship
- **e1, n1, u1:** position of the intruder in meters east, north, and up relative to the origin location specified in constants.py
- **h1, p1, r1:** orientation of the ownship in degrees of heading, pitch, and roll
- **intr_x, intr_y:** x and y coordinates of the intruder in pixel screen coordinates
- **loc:** origin location to which east, north, and up are relative
- **ac:** intruder aircraft type ('Cessna Skyhawk', 'Boeing 737-800`, or `King Air C90`)
- **clouds:** cloud cover (0 = Clear, 1 = Cirrus, 2 = Scattered, 3 = Broken, 4 = Overcast)
- **local_time_sec:** seconds since midnight local time on January 1st

Here is an example of the data recorded for the image shown below:

```
8, 2355.9042530256856, 3261.6742863566724, 508.6998591008703, 103.9734624716804,
-2.9625173985204767, 3.9581765331022236, -3.2176887991290393, -6.768574808330346,
29.652185687381873, 2385.1923377892804, 3258.091850788876, 505.7627921418019,
263.4528987103189, 8.294201441032358, -1.8456146435640215, 1216.053954289051,
1126.5644622813177, Palo Alto, Cessna Skyhawk, 0, 33057.013030092116
```

[DATASET].yaml

Config file that defines

1. the dataset root directory path and relative paths to image directories and
2. a dictionary of classes that are included in the dataset

Below is an example from a test dataset generated using this repository. (datasets/sample_mini.yaml)

```
train: ../../datasets/sample_mini/train/images
val: ../../datasets/sample_mini/valid/images
names:
  0: aircraft
```

Labels

Labels are outputted in subfolders called “labels” in *.txt files named according to which image the label matches up with. There is one row representing the label for the intruder object, formatted as:

```
class_number x_center y_center width height
```

Coordinates must be in normalized xywh format (from 0-1). If your boxes are in pixels, divide x_center and width by image width, and y_center and height by image height. The class number corresponds to the class of the object as denoted in the yaml file.