**General Workflow**

This is a simple resume of all the tasks :

* First we label all the images -> generates a csv text file
* Then with format\_annotations we create json files with the annotations and the image url’2
* The record\_maker combines annotations and image files in a big tensor record array which is the input to the train algorithm.
* As a plus record\_maker splits the set in 2 one with 80% of the records for train and the other with 20% for evaluation
* The we create the pipeline.config files for the training
* And just call train

Pipleline.config and training process are explained in the [Algorithmia article](https://blog.algorithmia.com/deep-dive-into-object-detection-with-open-images-using-tensorflow/). Please go to the end part of the article.

Files in Soft :

* Classes-bbox-trainable.txt are the name of the classes
* Label\_map.pbtxt is not used for the moment. It is from the pre-trained
* Format\_annotations.py reads the annotations file generated with the label\_files.py and generates a json file
* Record\_maker.py generates the Tensor Records from the json points file and the image list and the real images.
* In train
  + Pipeline.config is the train set configuration
* In eval
  + Pipeline.config is the eval set configuration
* ssd\_mobilenet\_v2\_coco\_2018\_03\_29 is the pretrained ssd network

**Format\_annotations.py parameters**(please, read the code)

— annotations\_input\_path is the file with annotations

— image\_index\_input\_path is the directory of images

— point\_output\_path is the json point file

— image\_index\_output\_path is the name->url image index

— trainable\_classes\_path is the file with the name of the classes (input)

This program :

- Parses the annotations file

- For each annotation looks for the associated file

- Adds it to the image\_index\_output\_path with the url

- Adds the annotations to the output annotation json file

**Record\_maker.py parameters** (please read the code)

— points\_path is the points json file created in last program

— train\_record\_save\_path is where to save the train record

— eval\_record\_save\_path is where to save the evaluation record

— trainable\_classes\_path is the classes file

— saved\_images\_directory is where the images are

Here we do the following :

- Load the points path

- Shuffle it and split it in 80% trainable 20% eval sets

- Generate 2 tensor record file for train and eva with all the images

**Training**

We need to create the pipeline.config files in eval and train directories.