Date: 5/2/2019

Author: Eric Bianchi

=====================================================================================

Description of the Contents in COCO-Bridge.zip

=====================================================================================

*Folder* “Original\_COCO-Birdge\_Dataset”:

This folder contains all the image files and bounding box information necessary to develop the neural network used in COCO-Bridge. It includes all the annotations. The images have been re-scaled.

* *Folder* “Annotations”
  + Bbox\_info\_CSV\_Evaluation.csv
  + Bbox\_info\_CSV\_Training.csv
* *Folder* “Evaluation\_Photos”
  + 1\_E.jpg
  + ….
  + 55\_E.jpg
* *Folder* “Training\_Photos”
  + 1\_T.jpg
  + …
  + 719\_T.jpg

*Folder* “Original\_COCO-Birdge-Python\_Package”:

This folder contains all the necessary classes and methods for re-creating the work done in COCO-Bridge.

* bbox.py
* class\_list.py
* csv\_info.py
* eval\_bbox.py
* eval\_con.py
* im\_aug.py
* image.py
* mAP\_main.py
* metrics\_Eval\_Build.py
* test.py

*Folder* “Original\_COCO-Birdge-Modelfile”:

This folder contains all the necessary files to pre-process and post-process the raw data. The finished neural network model is in the pre-processing folder. The mAP and qualitative results from the control model on the evaluation data are in the post-processing folder.

* Folder “Post-Processing”
  + Folder “Metrics\_Eval\_txt
    - 1\_E.jpg.txt
    - …
    - 55\_E.jpg.txt
  + Folder “Results\_Output”
    - Folder “mAP\_results\_IOU\_0.5\_Thres\_10%”
      * Folder “Classes”
        + Bearing.png
        + Cover Plate Termination.png
        + Gusset Plate Connection.png
        + Out of Plane Stiffener.png
      * Folder “Images”
        + Folder “single\_predictions”

(contains all predictions on each photo)

* + - * + 1\_E.jpg
        + …
        + 55\_E.jpg
      * Ground-Truth Info.png
      * mAP.png
      * Predicted Objects Info.png
      * Results.txt
    - Folder “mAP\_results\_IOU\_0.5\_Thres\_25%”
      * (Typical; same as above)
    - Folder “mAP\_results\_IOU\_0.5\_Thres\_50%”
    - Folder “mAP\_results\_IOU\_0.5\_Thres\_75%”
    - Folder “mAP\_results\_IOU\_0.35\_Thres\_1%”
    - Folder “mAP\_results\_IOU\_0.35\_Thres\_10%”
    - Folder “mAP\_results\_IOU\_0.35\_Thres\_25%”
    - Folder “mAP\_results\_IOU\_0.35\_Thres\_50%”
    - Folder “mAP\_results\_IOU\_0.35\_Thres\_75%”
    - Folder “mAP\_results\_IOU\_0.75\_Thres\_10%”
    - Folder “mAP\_results\_IOU\_0.75\_Thres\_25%”
    - Folder “mAP\_results\_IOU\_0.75\_Thres\_50%”
    - Folder “mAP\_results\_IOU\_0.75\_Thres\_75%”
  + Folder “Threshold\_%”
    - 1%\_Metrics\_Detect\_txt
      * 1\_E.jpg.txt
      * …
      * 55\_E.jpg.txt
    - 10%\_Metrics\_Detect\_txt
    - 25%\_Metrics\_Detect\_txt
    - 50%\_Metrics\_Detect\_txt
    - 75%\_Metrics\_Detect\_txt
  + README\_PostProcessing.txt
  + Run\_test.py
    - (The run file for the evaluation tests, a very powerful file as it can easily facilitate many tests from one file)
  + Test\_videoFeed.py
    - (This may be used to run in the background over a video-feed)
* Folder “Pre-Processing”
  + Folder “inference\_graph”
    - Folder “saved\_model”
      * Folder “variables”
      * Saved\_model.pb
    - Checkpoint
    - Frozen\_inference\_graph.pb
    - Model.ckpt-00000-of-00001
    - Model.ckpt.index
    - Model.ckpt.meta
    - Pipeline.config
  + Folder “ssd\_inception\_v2\_coco\_2017\_11\_17”
    - Folder “saved\_model”
      * Folder “variables”
      * Saved\_model.pb
    - Checkpoint
    - Frozen\_inference\_graph.pb
    - Model.ckpt-00000-of-00001
    - Model.ckpt.index
    - Model.ckpt.meta
  + Folder “Toolkit”
    - (Begin the processing with the “Data\_photo” folder)
    - Folder “\_\_pycache\_\_”
    - Folder “Data\_photo”
      * (To pre-process the images the following steps are taken: (1) create a folder called Original Files. (2) create a folder called "Eval" and "Train". Put your XML/Image pairs in those folders, splitting up the training and evaluation how you desire. (3) Extract only the Images from the "Train" folder in "Original Files" and put into the "Train\_Files" folder. (4) Extract only the Images from the "Eval" folder in "Orininal Files" and put into the "Evaluation\_Files" folder - The rest of folder may be left blank. (5) The next step is to run XML\_to\_CSV.py)
      * Evaluation\_Files
      * Finished\_Files
      * Flip\_R0
      * Mirror\_R0
      * Mirror\_R90
      * Mirror\_R180
      * Mirror\_R270
      * Rotate\_0
      * Rotate\_90
      * Rotate\_180
      * Rotate\_270
      * Train\_Files
    - Augmentation\_Number\_Evaluation.csv
      * (Determines how many times the image will be augmented, i.e flipped/mirrored. Numbers 1-8, where 1 is not augmented, and 8 is completed flipped and rotated every way)
    - Augmentation\_Number\_Train.csv
      * (Determines how many times the image will be augmented, i.e flipped/mirrored)
    - Data Augmentation for Image Media Resizing.py
      * (Re-size the images after running the List\_of\_image\_objects\_for\_augmentation.py)
    - List of Image Objects for Augmentation.py
      * (Run this file after the xml\_to\_csv.py)
    - Xml\_to\_csv.py
      * (run this file after entering in all the photos and XML files into “Data\_photo”)
  + Folder “training\_data”
    - Bbox\_info\_CSV\_output\_Evaluation.csv
      * (file used for generating the eval.tfrecord)
    - Bbox\_info\_CSV\_output\_Train.csv
      * (file used for generating the train.tfrecord)
    - Checkpoint
    - Eval.tfrecord
    - Eval\_labels.csv
      * (original bbox data, used to create the Bbox\_info\_CSV\_output\_Evaluation.csv)
    - Events.out.tfevents.151108601.ca217
    - Graph.pbxt
    - Mode.ckpt-0.data-00000-of-00001
    - Model.ckpt-0.index
    - Model.ckpt-0.meta
    - Model.ckpt-1874.data-00000-of-00001
    - Model.ckpt-1874.index
    - Model.ckpt-1874.meta
    - Model.ckpt-3811.data-00000-of-00001
    - Model.ckpt-3811.index
    - Model.ckpt-3811.meta
    - Model.ckpt-5000.data-00000-of-00001
    - Model.ckpt-5000.index
    - Model.ckpt-5000.meta
    - Pipeline.config
    - Train.tfrecord
    - Train\_labels.csv
      * (original bbox data, used to create the Bbox\_info\_CSV\_output\_Train.csv)
  + Export\_inference\_graph.py
    - (Function to take the training data and turn it into a usable format for running the model)
  + Generate\_tfrecords.py
    - (Function to create the filetypes which may be read by the TensorFlow API to build the neural network)
  + Label\_map.pbtxt
    - (This is the file which contains the IDs for all the classes or structural details. This is an important file which much be updated as new structural details/classes are added)
  + Pipeline.config
    - (configuration file)
  + Ssd\_inception\_v2\_coco.config
    - (The file which contains the training parameters)
  + Train.py
    - (The file to run and begin the training process)