**AIDA code documents (Face, Flag, Landmark, ttl file generator)**

Brian Chen

1. **Face: https://github.com/davidsandberg/facenet**
   1. Face detection for image & key frames

CUDA\_VISIBLE\_DEVICES=0 python src/align/align\_dataset\_mtcnn.py \

[input directory of img or key frames] \

datasets/[output dir] \

--image\_size 160 \

--margin 32

* 1. Face Recognition for image & key frames

CUDA\_VISIBLE\_DEVICES=0 python src/classifier.py CLASSIFY datasets/[1.1 output directory] models/facenet/20180402-114759/20180402-114759.pb models/google500\_2\_classifier.pkl [result pickle file name] --batch\_size 1000 > results/[result txt file name].txt

* 1. Store bounding box

python src/bbox.py [1.1 output directory] [bbox pickle name]

1. **Building & Flag**
   1. Object detection for building and flag: <https://github.com/tensorflow/models/tree/master/research/object_detection>
      1. Under the object detection directory in tensorflow, run obj\_preprocess.ipynb

In line 10, change the folder path to img folder.

In line 11, set the saved pickle file name for flag detection.

The last block saves the building\_list.txt

* 1. Flag
     1. Save cropped flag files:

Run save\_flag\_crop.ipynb under object\_detection directory,

line 7 set the output directory of the cropped flags

line 8 is the pickle file stored the object detection result generated from 2.2.1

* + 1. Classify the flags

python src/label\_image.py [output dir of 2.2.1] [output flag result pickle name]

* 1. Building
     1. Feature extraction: <https://github.com/tensorflow/models/tree/master/research/delf>

Under the delf directory in tensorflow (tensorflow/models/research/delf/delf/python/examples), run

CUDA\_VISIBLE\_DEVICES=0 python extract\_features.py \

--config\_path delf\_config\_example.pbtxt \

--list\_images\_path [path to the building\_list.txt created by obj\_preprocess.ipynb] /building\_list.txt \

--output\_dir [output feature directory]

* + 1. Feature matching

python match2.py [output feature directory from 2.3.1] [output result pickle name]

1. **Output ttl files**
   1. Read text entity from RPI ttl files

Run read\_RPI\_entity.ipynb

Block 2 line 12 is the ttl folder fenerated by RPI

Block 2 line 34 is the stored pickle file

* 1. Run create\_ttl\_m18.ipynb (Take m18 as an example)

Need to change the path in the first block when executing

parent\_file: Parent child document relation given by LDC (parent\_children.sorted.tab)

video\_frame\_mapping = video frame id mapping from LDC (masterShotBoundary.msb)

face\_img\_result = Output from 1.2, face classification pickle file for img

face\_frame\_result = Output from 1.2, face classification pickle file for video frame

bbox\_img: Output from 1.3, bbox of image

bbox\_frame = Output from 1.3, bbox of video frame

az\_obj\_graph = Object detection graph from Alireza (rdf\_graphs\_34.pkl)

az\_obj\_jpg = Object detection dictionary for img from Alireza (det\_results\_merged\_34a\_jpg.pkl)

az\_obj\_kf = Object detection dictionary for key frame from Alireza (det\_results\_merged\_34b\_kf.pkl)

Lorelei\_path = External knowledge Lorelei path (entities.tab')

flag\_result = Output from 2.2.2 (flag\_m18\_2.pickle)

landmark\_result = Output from 2.3.2 (result\_dic\_m18\_new.p)

RPI\_entity = Output from 3.1 (PT003\_r1.pickle)

input\_img\_path = jpg file directory path

outputN = Output ttl folder name

The output file will be ttl files in the folder