WORK TRACKER

# JULY 2019

## **17th JULY**

### WORK DONE :

* Use Fatkun Batch downloader to download license plate images in batches from Google Image Search
* Gathered around 500 images
* As it required multiple searches, and multiple batch downloads, images names could conflict which could create problems while copying from downloads folder(target folder) to destination folder
* Wrote a python code “rename.py” (inside “data” folder) for renaming images in the form “i1.jpg/png/jpeg”, “i2.\_” ..... where the digits after i started from L+1 where L = number of images already there in the destination folder
* The python code also removed any files of the format “.webp” or any other form other than jpg,png or jpeg

NEXT CHALLENGE:

* Collect more images
* Remove possible duplicate images with different names and sizes

## **18th JULY**

### SOLUTIONS TO PREVIOUS ISSUES :

* Keep using Fatkun Batch downloader to download more images
* Use code as given in <https://medium.com/@urvisoni/removing-duplicate-images-through-python-23c5fdc7479e> for removing duplicates

ISSUES:

* Code in above link requires all files to be in same format

### SOLUTIONS TO ISSUES :

* Convert all files to same format using PIL library code as in “convert.py” inside data folder

### WORK DONE :

* Made a copy of the image data folder somewhere else and try the below operations, if successful, replace original folder
* Converted all files into “.png” format

## **21st JULY**

## WORK DONE:

* Ran the duplicate removal technique
* Copied the modified images into the original folder

## NEXT CHALLENGE:

* Annotate the images using microsoft VOTT tool
* Convert the annoted file into the required format
* Run self training YOLO on these annoted files

## **23rd-24th JULY**

## WORK DONE:

* Annoted the images files using Microsoft vott

## NEXT CHALLENGE:

* Convert them to required format for running darknet YOLO
* Run darknet Yolo

## **27th JULY**

## ISSUES:

## Vott exported in json format, while conversion code for darknett expects csv

## SOLUTION:

## Modify the settings in vott to export in csv

## WORK DONE:

* Exported the labelled annotations in csv format
* Ran the code

TrainSelfYolo/1\_Image\_Annotation/Convert\_to\_YOLO\_format.py

to convert the vott export annotations file to a txt file in the darknet YOLO format

* Ran the code to download the weights as in TrainSelfYolo/2\_Training/Download\_and\_Convert\_YOLO\_weights.py
* Uploaded the whole LicensePlateDetection folder to Google Drive to do further work of training in colab

## NEXT CHALLENGE:

## Train YOLO by running Train\_YOLO.py in TrainSelfYolo/2\_Training/

## (Working in Google Colab)

## **28th JULY**

## ISSUES:

## Keras backend error module 'keras.backend' has no attribute 'control\_flow\_ops'

## SOLUTION:

* Use “ !pip install q keras==1.2.2 ” inside google colab to replace higher version of keras with lower version

## ISSUES:

## Error

File "/content/gdrive/My Drive/License\_Plate\_Detection/TrainSelfYolo/2\_Training/src/keras\_yolo3/yolo3/model.py", line 8, in <module>

from keras.layers import (

ImportError: cannot import name 'Add'

## SOLUTION:

* Checked keras version used in TrainSelfYolo/requirements.txt. Keras version == 2.2.4
* Use “ !pip install q keras==2.2.4 ” inside google colab to replace keras with appropriate version
* Tensorflow version also causing problems. Use !pip install -r "/content/gdrive/My Drive/License\_Plate\_Detection/TrainSelfYolo/requirements.txt"

to install all required versions of the libraries

## ISSUES:

## Error

Traceback (most recent call last):

File "Train\_YOLO.py", line 190, in <module>

lines = ChangeToOtherMachine(lines, remote\_machine="")

File "/content/gdrive/My Drive/License\_Plate\_Detection/TrainSelfYolo/Utils/Train\_Utils.py", line 238, in ChangeToOtherMachine

suffix = (file.split(repo))[1]

IndexError: list index out of range

## The YOLO annotations file “data\_train .txt” generated via

## “ Ran the code to convert the vott export annotations file to a txt file in the darknet YOLO format ”

## was generated in local machine so contained all pathnames as per local machine path.

## SOLUTIONS:

* Regenarate the “data\_train.txt” by again running TrainSelfYOLO/1\_Image\_Annotation/Convert\_to\_YOLO\_format.py

inside colab

## ISSUES:

## Error

Traceback (most recent call last):

File "Train\_YOLO.py", line 190, in <module>

lines = ChangeToOtherMachine(lines, remote\_machine="")

File "/content/gdrive/My Drive/License\_Plate\_Detection/TrainSelfYolo/Utils/Train\_Utils.py", line 238, in ChangeToOtherMachine

suffix = (file.split(repo))[1]

IndexError: list index out of range

## Still same error. Have to debug inside Train\_Utils.py

## SOLUTION:

* Inside Train\_Utils.py , line number 215 approx

replace

“ def ChangeToOtherMachine(filelist, repo="TrainYourOwnYOLO", remote\_machine=""): ”

with

“ def ChangeToOtherMachine(filelist, repo="TrainSelfYolo", remote\_machine=""): ”

* This was causing problem as original repository name was TrainYourOwnYOLO, while we had renamed the folder to TrainSelfYolo

## **29th JULY**

## WORK DONE:

* Made modifications in Train\_YOLO.py to incorporate flags to allow running only initial training phase or running only the latter fine-tuning phase.
* Also made modifications to have differentvalues for epochs of first and second phase with default values of 30 and 10 respectively. Value kept low due to resource limitations
* Training YOLO (initial training phase with layers freezed). Initial phase weights saved in License\_Plate\_Detection/TrainSelfYolo/Data/Model\_Weights/trained\_weights\_stage\_1.h5

## NEXT WORK:

* Run only second part (fine tuning) by first loading model from the above initial weights file (Check for correctness of code in Train\_Yolo.py)
* The loss for the initial epoch of second phase should be low (around 100)
* If loss in not correctly, try fixing any problems in code in Train\_Yolo.py
* Else run the code without the flags (i.e running both the phases)

## **30th JULY**

## WORK DONE:

* Made modifications in Detector.py to incorporate testing based on model with intermediate weights

## NEXT WORK:

* Run only second part (fine tuning) by first loading model from the above initial weights file (Check for correctness of code in Train\_Yolo.py)
* The loss for the initial epoch of second phase should be low (around 100)
* If loss in not correctly, try fixing any problems in code in Train\_Yolo.py
* Else run the code without the flags (i.e running both the phases)

## **1st AUGUST**

## WORK DONE:

* Tried to fix in Train\_YOLO.py to incorporate running fine\_tune from loaded weights but failed
* Ran detector.py on intermediate weights (not being able to detect bounding boxes)

## NEXT WORK:

* Try and run the whole training process

## **2nd AUGUST**

## WORK DONE:

* Code to re-size the entire original set of images to 200 \* 200
* Started reannoting the new resized image set

## NEXT WORK:

* Repeat the entire process with these new set of images

## **3rd AUGUST**

## WORK DONE:

* Finished annotation of the new resized images

## NEXT WORK:

* Repeat the entire process with these new set of images

## **4th–5th AUGUST**

## WORK DONE:

* Made a new set of images taking only those with height and width more than 416,416 (YOLO model size)
* Annoted these images
* Trained model on these images
* Got satisfactory results

## NEXT WORK:

* Find random images from net (high res like those in training set), to test model further
* Try to think of next step (Maybe OCR)