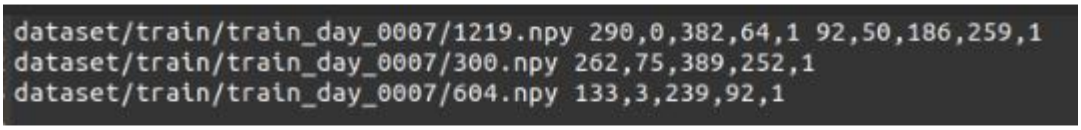
Yolo4_tiny Training in GPU with eTRAM Dataset

1. Clone the repo:
   https://github.com/yss9701/Ultra96-Yolov4-tiny-and-Yolo-Fastest

   Git clone https://github.com/yss9701/Ultra96-Yolov4-tiny-and-Yolo-Fastest

2. Dataset preparation:
   It will create a npy file with labels
   Convert npy and labels to yolovy_tiny



```
dataset/train/train_day_0007/1219.npy 290,0,382,64,1 92,50,186,259,1
dataset/train/train_day_0007/300.npy 262,75,389,252,1
dataset/train/train_day_0007/604.npy 133,3,239,92,1
```

Img_location: x1,y1,x2,y2,class_id

3. Training
   yolov4 _tiny model is trained in npy files

Change def get_random_data() in utils/utils.py

Change the function to read npy and resize to (320,320)

………

……….

……….

Note: change the resize as data preparation

Change the annotation path , cass path and anchor paths, log_dir

```
annotation_path = '2007_train.txt'
classes_path = 'model_data/new_class.txt'
anchors_path = 'model_data/yolo_anchors.txt'
```

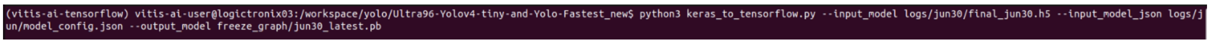
We can create separate annotation for train and val and train the model

4. Train the model in tensorflow2 environment

Cd ……

Python3 train.py

5. Freeze the graph



```
(vitis-ai-tensorflow) vitis-ai-user@logictronix03:/workspace/yolo/Ultra96-Yolov4-tiny-and-Yolo-Fastest_new$ python3 keras_to_tensorflow.py --input_model logs/jun30/final_jun30.h5 --input_model_json logs/j
un/model_config.json --output_model freeze_graph/jun30_latest.pb
```

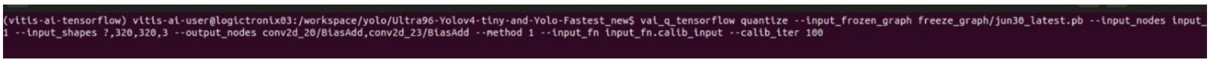
Here is the commands:

Convert h5 model into pb(freeze the model):

```
# python3 keras_to_tensorflow.py --input_model
gpu_model/jun30/ep002-loss4.026-val_loss7.795.h5 --
input_model_json gpu_model/jun30/model_config.json --
output_model freeze_graph/jun30_latest.pb
```

6. Quantize

For Quantization and Compilation, goto “Vitis AI CPU/GPU docker (3.5)” activate the “vitis ai tensorflow” and run:

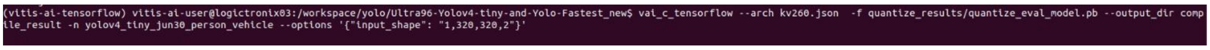


```
(vitis-ai-tensorflow) vitis-ai-user@logictronix03:/workspace/yolo/Ultra96-Yolov4-tiny-and-Yolo-Fastest_new$ vai_q_tensorflow quantize --input_frozen_graph freeze_graph/jun30_latest.pb --input_nodes input_
1 --input_shapes ?,320,320,3 --output_nodes conv2d_20/BiasAdd,conv2d_23/BiasAdd --method 1 --input_fn input_fn.calib_input --calib_iter 100
```

Here is the command:

```
vai_q_tensorflow quantize --input_frozen_graph
freeze_graph/jun30_latest.pb --input_nodes input_1 --
input_shapes ?,320,320,3 --output_nodes
conv2d_20/BiasAdd,conv2d_23/BiasAdd --method 1 --input_fn
input_fn.calib_input --calib_iter 100
```

7. Compile



```
(vitis-ai-tensorflow) vitis-ai-user@logictronix03:/workspace/yolo/Ultra96-Yolov4-tiny-and-Yolo-Fastest_new$ vai_c_tensorflow --arch kv260.json  -f quantize_results/quantize_eval_model.pb --output_dir comp
ile_result -n yolov4_tiny_jun30_person_vehicle --options '{"input_shape": "1,320,320,2"}'
```

Here is the command:

```
vai_c_tensorflow --arch kv260.json  -f
quantize_results/quantize_eval_model.pb --output_dir
compile_result -n yolov4_tiny_jun30_person_vehicle --options
'{"input_shape": "1,320,320,2"}'
```

After compilation completes, you will get the XMODEL inside “`compile_result`” directory.

Datasets Preparation Details

We are using eTRAM dataset from Link: https://eventbasedvision.github.io/eTraM

1. Data preparation :
   We are going to use following github repo for dataset preparation.
   link:https://github.com/eventbasedvision/eTraM/tree/main/rvt_eTram/scripts/genx

   Change nbins: 10 into nbins: 1
   https://github.com/eventbasedvision/eTraM/blob/main/rvt_eTram/scripts/genx/conf_preprocess/representation/stacked_hist.yam

   It will create a h5 file with size height,width, 2

   ```
   NUM_PROCESSES=20  # set to the number of parallel processes to use
   python preprocess_dataset_gen4.py ${DATA_DIR} ${DEST_DIR}
   conf_preprocess/representation/stacked_hist.yaml \
   conf_preprocess/extraction/const_duration.yaml conf_preprocess/filter_gen4.yaml -
   ds gen4 -np ${NUM_PROCESSES}
   ```

   H5 to npy conversion :
   In h5 conversion it creates labels files and index file .
   Scripts to run :
   1:
   2: