KV260 YOLOv3 Custom Data 開發

A. Training YOLOv3 model (使用 GPU)

Step01: Download dk_yolov3_voc_416_416_65.42G_2.0

download link:
https://www.xilinx.com/bin/public/openDownload?filename=dk_yolov3_voc_416_416_65.42G_2.0.zip

cd dk_yolov3_voc_416_416/

Step02: 下載 darknet

KV260_YOLOv3/dk_yolov3_voc_416_416$ git clone https://github.com/AlexeyAB/darknet

Step03: 修改 yolov3-voc.cfg

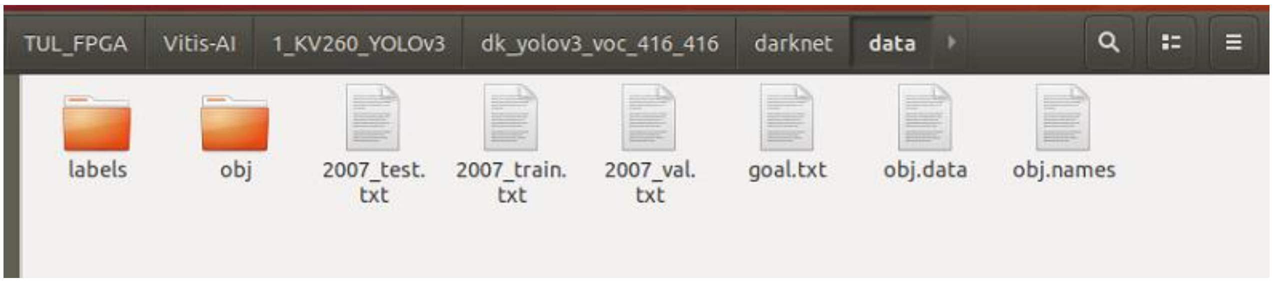
從 darnet/cfg 複製 yolov3-voc.cfg 到 darnet/ 目錄，再進行修改如下：
line3: batch=64
line4: subdivisions=16

line20: max_batches = 4000
line22: steps=3500,3800

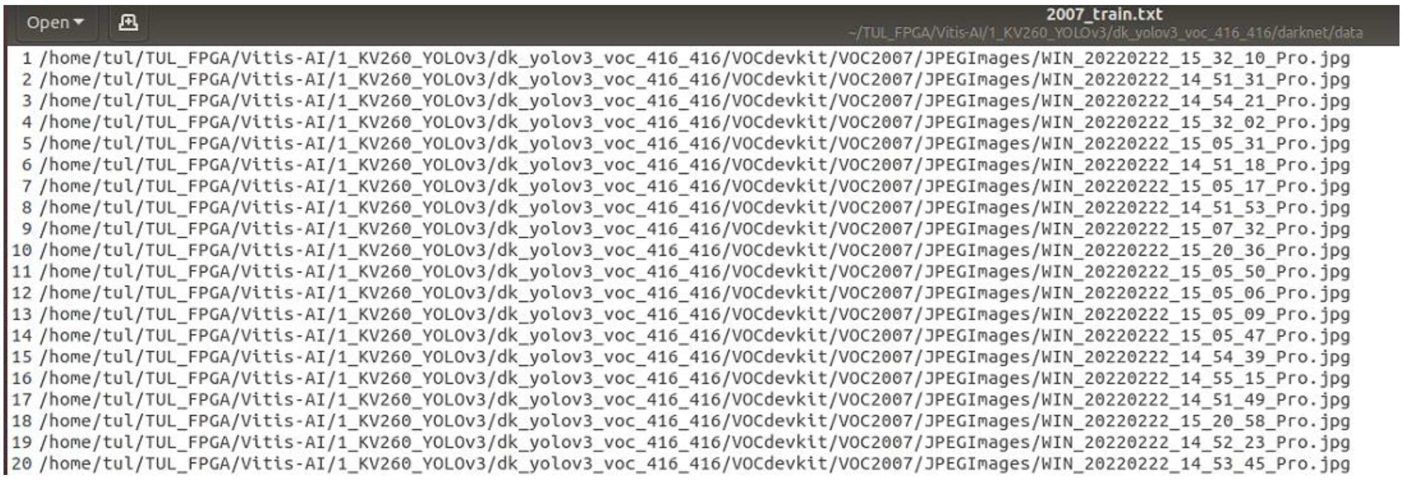
line611: classes=2
line695: classes=2
line779: classes=2

line605: filters=21
line689: filters=21
line773: filters=21

Step04: 準備訓練資料 (VOC 格式)



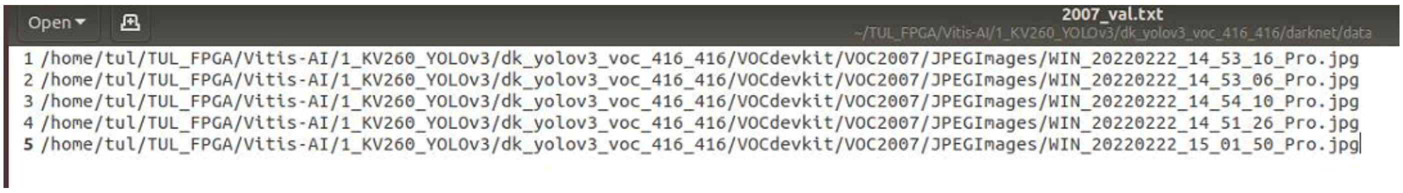
2007_train.txt (68 張)



Open
2007_train.txt
~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/darknet/data

```
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_32_10_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_51_31_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_54_21_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_32_02_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_05_31_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_51_18_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_05_17_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_51_53_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_07_32_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_20_36_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_05_50_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_05_06_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_05_09_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_05_47_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_54_39_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_55_15_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_51_49_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_20_58_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_52_23_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_53_45_Pro.jpg
```

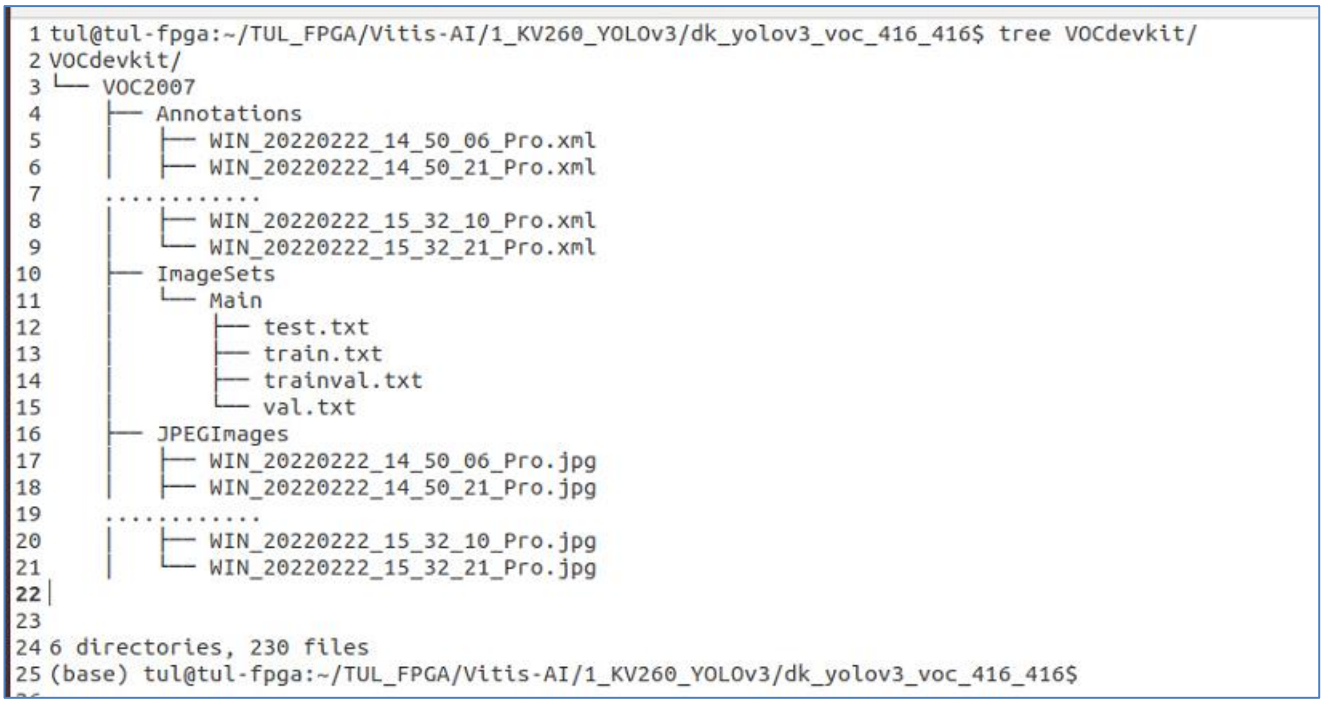
2007_val.txt (5 張)



Open
2007_val.txt
~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/darknet/data

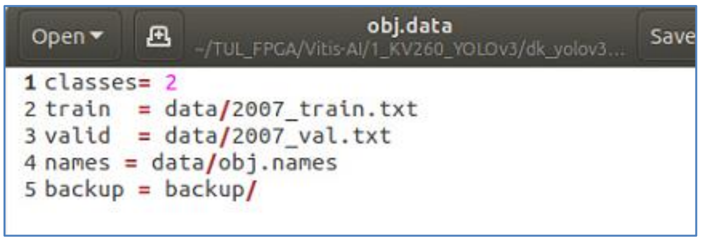
```
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_53_16_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_53_06_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_54_10_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_14_51_26_Pro.jpg
/home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/VOCdevkit/VOC2007/JPEGImages/WIN_20220222_15_01_50_Pro.jpg
```

資料擺放位置:



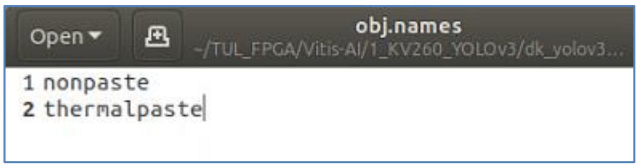
```
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416$ tree VOCdevkit/
VOCdevkit/
└── VOC2007
    ├── Annotations
    │   ├── WIN_20220222_14_50_06_Pro.xml
    │   ├── WIN_20220222_14_50_21_Pro.xml
    ............
    │   ├── WIN_20220222_15_32_10_Pro.xml
    │   └── WIN_20220222_15_32_21_Pro.xml
    ├── ImageSets
    │   └── Main
    │       ├── test.txt
    │       ├── train.txt
    │       ├── trainval.txt
    │       └── val.txt
    ├── JPEGImages
    │   ├── WIN_20220222_14_50_06_Pro.jpg
    │   ├── WIN_20220222_14_50_21_Pro.jpg
    ............
    │   ├── WIN_20220222_15_32_10_Pro.jpg
    │   └── WIN_20220222_15_32_21_Pro.jpg


6 directories, 230 files
(base) tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416$
```

Step05: Create data/obj.data



```
classes= 2
train  = data/2007_train.txt
valid  = data/2007_val.txt
names = data/obj.names
backup = backup/
```

Step06: Create data/obj.names

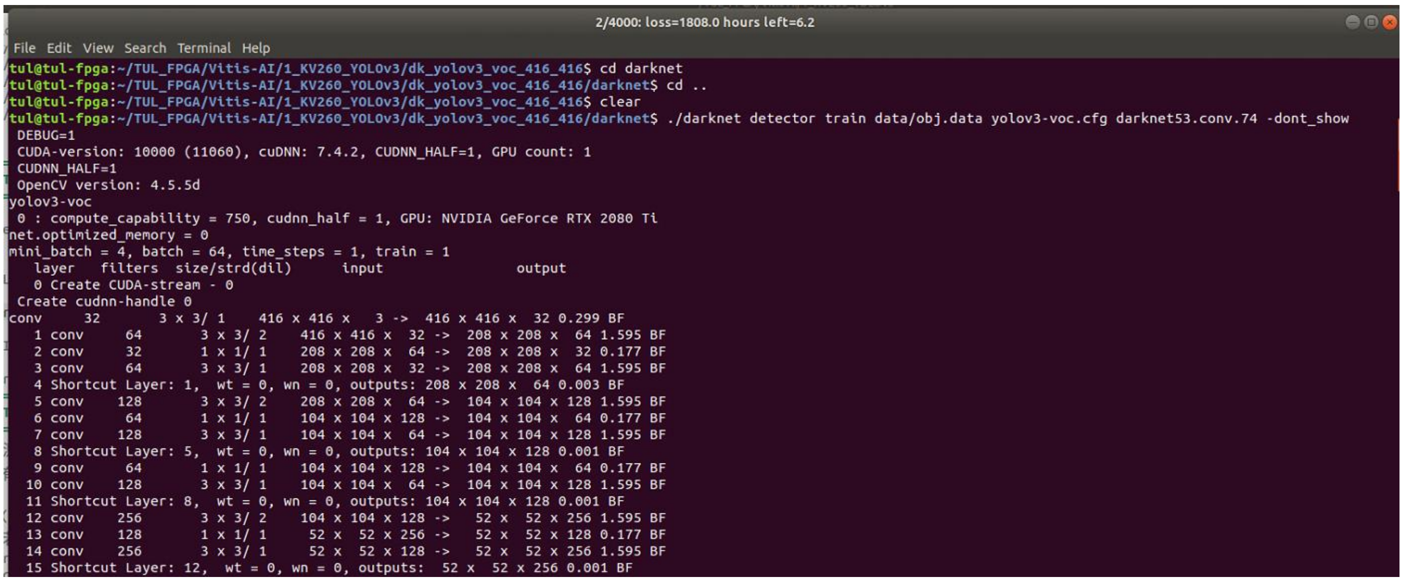


```
nonpaste
thermalpaste
```

Step07: 開始訓練

下載 darknet53.conv.74 預訓練模型
Link: wget https://pjreddie.com/media/files/darknet53.conv.74

./darknet detector train data/obj.data yolov3-voc.cfg darknet53.conv.74 -dont_show

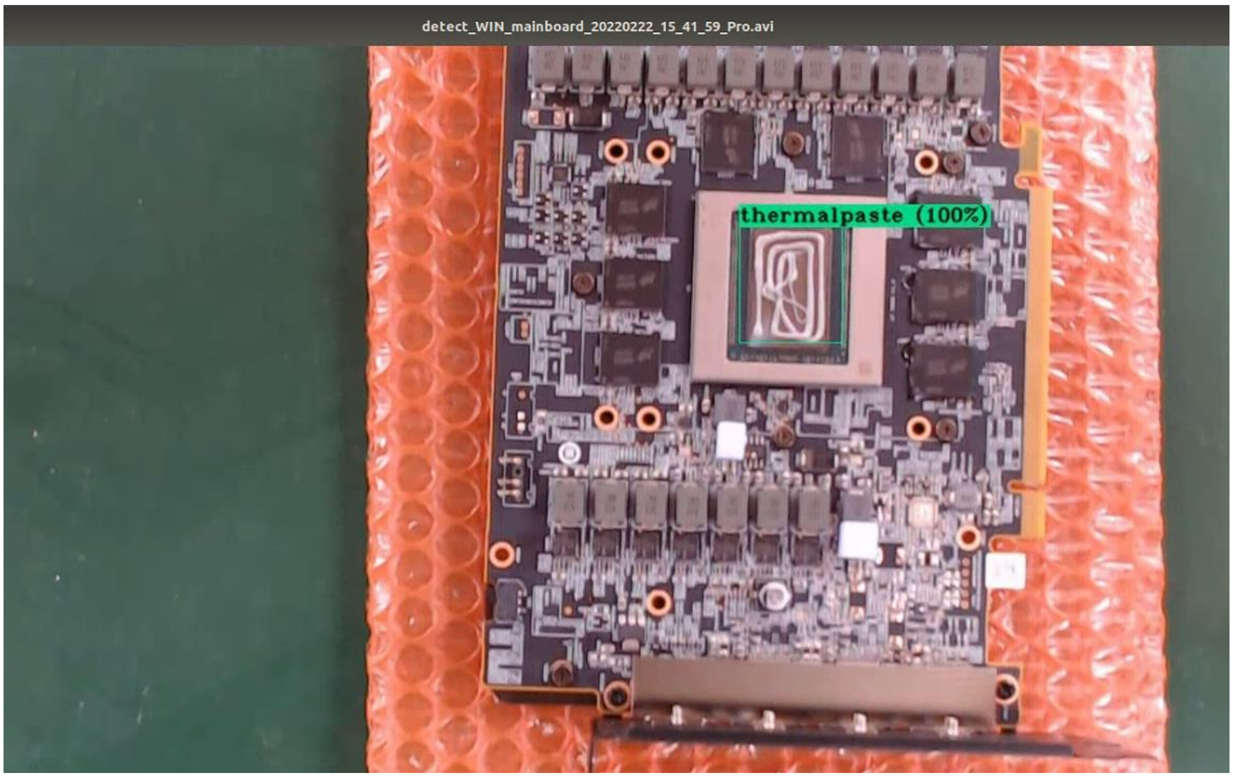

2/4000: loss=1808.0 hours left=6.2
File Edit View Search Terminal Help

```
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416$ cd darknet
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/darknet$ cd ..
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416$ clear
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/darknet$ ./darknet detector train data/obj.data yolov3-voc.cfg darknet53.conv.74 -dont_show
 DEBUG=1
 CUDA-version: 10000 (11060), cuDNN: 7.4.2, CUDNN_HALF=1, GPU count: 1
 CUDNN_HALF=1
 OpenCV version: 4.5.5d
yolov3-voc
 0 : compute_capability = 750, cudnn_half = 1, GPU: NVIDIA GeForce RTX 2080 Ti
net.optimized_memory = 0
mini_batch = 4, batch = 64, time_steps = 1, train = 1
   layer   filters  size/strd(dil)      input                output
   0 Create CUDA-stream - 0
 Create cudnn-handle 0
conv     32       3 x 3/ 1    416 x 416 x   3 ->  416 x 416 x  32 0.299 BF
   1 conv     64       3 x 3/ 2    416 x 416 x  32 ->  208 x 208 x  64 1.595 BF
   2 conv     32       1 x 1/ 1    208 x 208 x  64 ->  208 x 208 x  32 0.177 BF
   3 conv     64       3 x 3/ 1    208 x 208 x  32 ->  208 x 208 x  64 1.595 BF
   4 Shortcut Layer: 1,  wt = 0, wn = 0, outputs: 208 x 208 x  64 0.003 BF
   5 conv    128       3 x 3/ 2    208 x 208 x  64 ->  104 x 104 x 128 1.595 BF
   6 conv     64       1 x 1/ 1    104 x 104 x 128 ->  104 x 104 x  64 0.177 BF
   7 conv    128       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x 128 1.595 BF
   8 Shortcut Layer: 5,  wt = 0, wn = 0, outputs: 104 x 104 x 128 0.001 BF
   9 conv     64       1 x 1/ 1    104 x 104 x 128 ->  104 x 104 x  64 0.177 BF
  10 conv    128       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x 128 1.595 BF
  11 Shortcut Layer: 8,  wt = 0, wn = 0, outputs: 104 x 104 x 128 0.001 BF
  12 conv    256       3 x 3/ 2    104 x 104 x 128 ->   52 x  52 x 256 1.595 BF
  13 conv    128       1 x 1/ 1     52 x  52 x 256 ->   52 x  52 x 128 0.177 BF
  14 conv    256       3 x 3/ 1     52 x  52 x 128 ->   52 x  52 x 256 1.595 BF
  15 Shortcut Layer: 12,  wt = 0, wn = 0, outputs:  52 x  52 x 256 0.001 BF
```

(訓練大約 6 個小時左右)

Step08: 測試模型

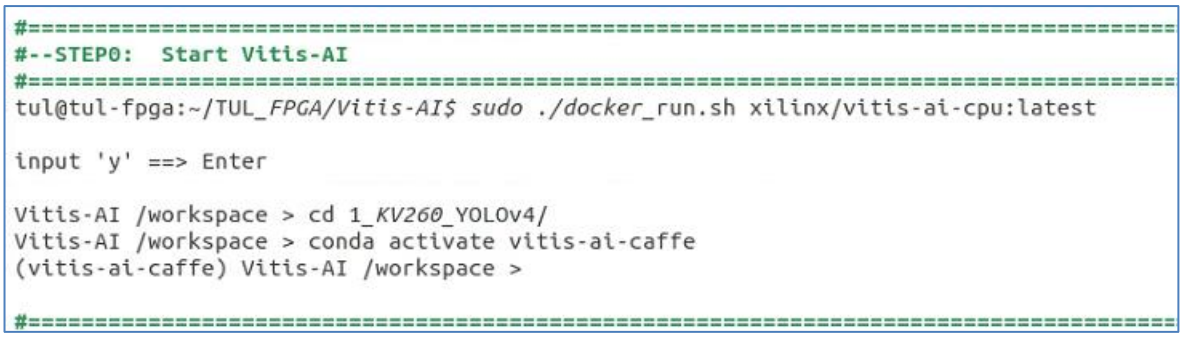
./darknet detector demo data/obj.data yolov3-voc.cfg backup/yolov3-voc_final.weights test_video/WIN_mainboard_20220222_15_41_59_Pro.mp4 -out_filename test_video/detect_WIN_mainboard_20220222_15_41_59_Pro.avi -dont_show



B. 模型轉換

Step01: 啟動 Vitis-AI 環境

選擇 caffe docker, 執行 conda activate vitis-ai-caffe



```
#==========================================================================================
#--STEP0:  Start Vitis-AI
#==========================================================================================
tul@tul-fpga:~/TUL_FPGA/Vitis-AI$ sudo ./docker_run.sh xilinx/vitis-ai-cpu:latest

input 'y' ==> Enter

Vitis-AI /workspace > cd 1_KV260_YOLOv4/
Vitis-AI /workspace > conda activate vitis-ai-caffe
(vitis-ai-caffe) Vitis-AI /workspace >

#==========================================================================================
```

Step02: 轉換 darnet model 成 cafee model

從 https://github.com/Xilinx/Vitis-AI/tree/master/models/AI-Model-Zoo/caffe-xilinx
下載 caffe-xilinx

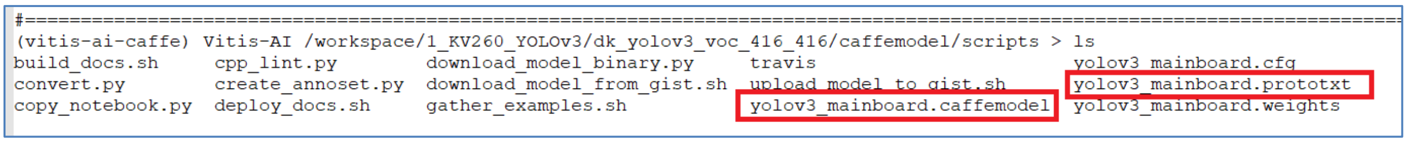
cp -r models/AI-Model-Zoo/caffe-xilinx/   1_KV260_YOLOv3/caffemodel/

cp -r darknet/backup/yolov3-voc_final.weights caffemodel/scripts/yolov3_mainboard.weights

cp -r darknet/yolov3-voc.cfg caffemodel/scripts/yolov3_mainboard.cfg cd caffemodel/scripts/

python convert.py yolov3_mainboard.cfg yolov3_mainboard.weights yolov3_mainboard.prototxt yolov3_mainboard.caffemodel

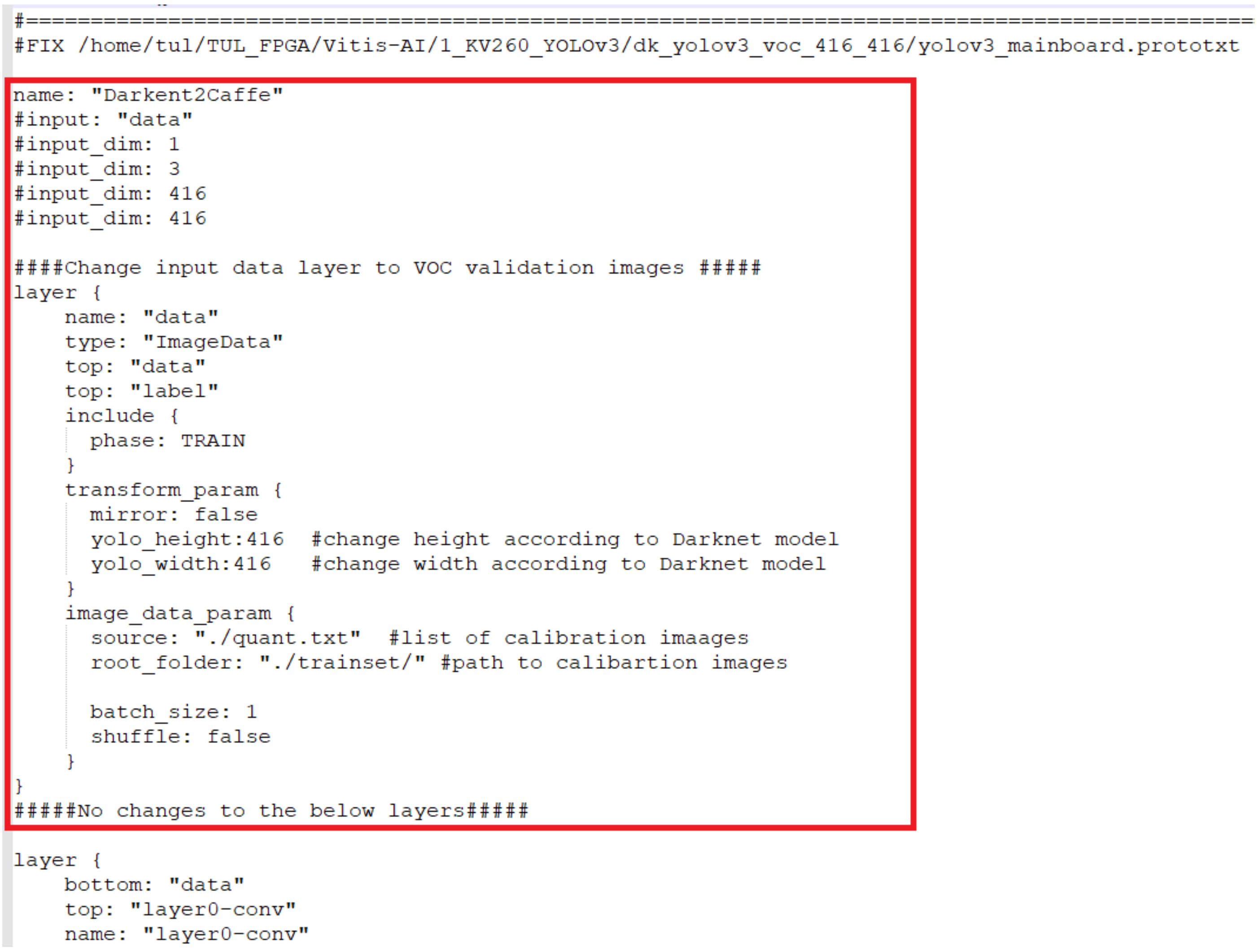
轉換後，得到 caffe 模型如下：



```
#==========================================================================================================================
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416/caffemodel/scripts > ls
build_docs.sh     cpp_lint.py        download_model_binary.py     travis                       yolov3_mainboard.cfg
convert.py        create_annoset.py  download_model_from_gist.sh  upload_model_to_gist.sh      yolov3_mainboard.prototxt
copy_notebook.py  deploy_docs.sh     gather_examples.sh           yolov3_mainboard.caffemodel  yolov3_mainboard.weights
```

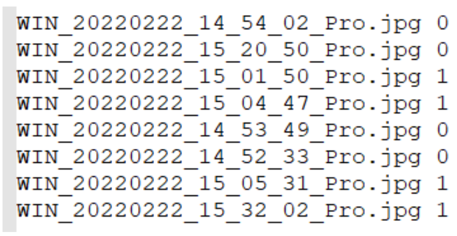
Step03: Quantization 量化模型

➢ 修改 yolov3_mainboard.prototxt



```
#==========================================================================================
#FIX /home/tul/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/yolov3_mainboard.prototxt

name: "Darkent2Caffe"
#input: "data"
#input_dim: 1
#input_dim: 3
#input_dim: 416
#input_dim: 416

####Change input data layer to VOC validation images #####
layer {
    name: "data"
    type: "ImageData"
    top: "data"
    top: "label"
    include {
      phase: TRAIN
    }
    transform_param {
      mirror: false
      yolo_height:416  #change height according to Darknet model
      yolo_width:416   #change width according to Darknet model
    }
    image_data_param {
      source: "./quant.txt"  #list of calibration imaages
      root_folder: "./trainset/" #path to calibartion images

      batch_size: 1
      shuffle: false
    }
}
#####No changes to the below layers#####

layer {
    bottom: "data"
    top: "layer0-conv"
    name: "layer0-conv"
```

➢ 建立 quant.txt (裡面包含影像路徑和類別)



```
WIN_20220222_14_54_02_Pro.jpg 0
WIN_20220222_15_20_50_Pro.jpg 0
WIN_20220222_15_01_50_Pro.jpg 1
WIN_20220222_15_04_47_Pro.jpg 1
WIN_20220222_14_53_49_Pro.jpg 0
WIN_20220222_14_52_33_Pro.jpg 0
WIN_20220222_15_05_31_Pro.jpg 1
WIN_20220222_15_32_02_Pro.jpg 1
```

➢ 量化指令

vai_q_caffe quantize -model yolov3_mainboard.prototxt -keep_fixed_neuron -calib_iter 9 -weights yolov3_mainboard.caffemodel -sigmoided_layers layer81-conv,layer93-conv,layer105-conv -output_dir quantized/ -method 1

```
I0518 22:23:25.441605    91 vai_q.cpp:360] Start Deploy
I0518 22:23:31.734006    91 vai_q.cpp:368] Deploy Done!
----------------
Output Quantized Train&Test Model:   "quantized/quantize_train_test.prototxt"
Output Quantized Train&Test Weights: "quantized/quantize_train_test.caffemodel"
Output Deploy Weights: "quantized/deploy.caffemodel"
Output Deploy Model:   "quantized/deploy.prototxt"
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

量化後，得到 4 個檔案如下：

```
#-------------------------------------------------------------------------------------
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > tree quantized/
quantized/
├── deploy.caffemodel
├── deploy.prototxt
├── quantize_train_test.caffemodel
└── quantize_train_test.prototxt

0 directories, 4 files
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

Step04: Model Compliation

編譯前，需要先修改 quantize_results/deploy.prototxt 如下：

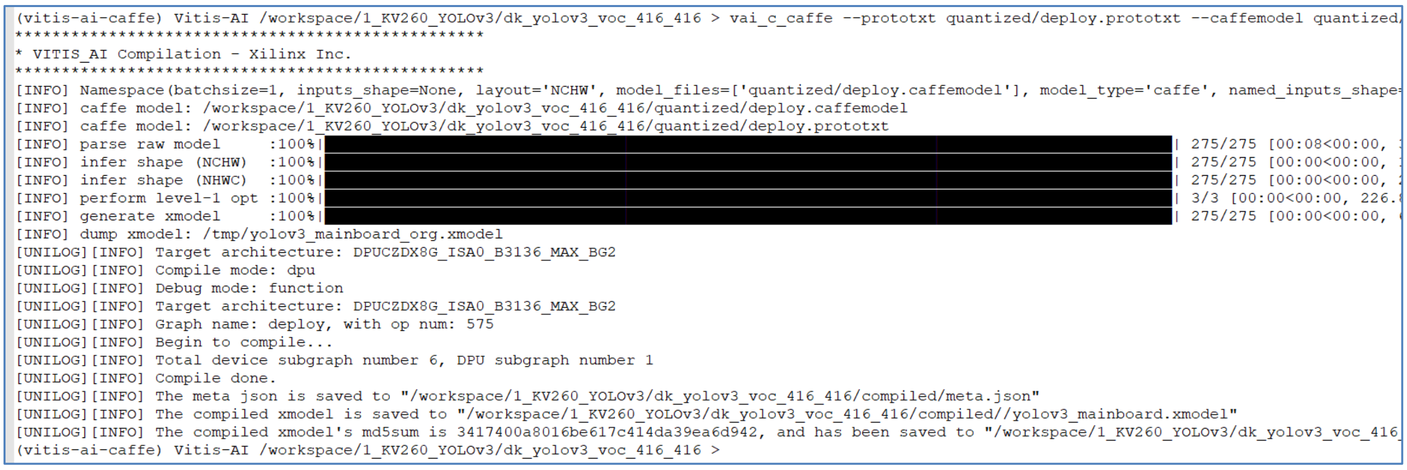
```
layer {
  name: "data"
  type: "Input"
  top: "data"
  input_param {
    shape {
      dim: 1
      dim: 3
      dim: 416
      dim: 416
    }
  }
}
layer {
  name: "layer0-conv"
  type: "Convolution"
  bottom: "data"
  top: "layer0-conv"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  phase: TRAIN
  convolution_param {
    num_output: 32
    bias_term: true
    pad: 1
    kernel_size: 3
    stride: 1
  }
}
...
```

- 建立 compiled 為輸出目錄
- 建立 arch.json 檔案

```
#FIX arch.json
{
    "target": "DPUCZDX8G_ISA0_B3136_MAX_BG2"
}
```

- 執行 compiler

  vai_c_caffe --prototxt quantized/deploy.prototxt --caffemodel quantized/deploy.caffemodel --arch arch.json --output_dir compiled/ --net_name yolov3_mainboard --options "{'mode':'normal','save_kernel':''}"



```
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > vai_c_caffe --prototxt quantized/deploy.prototxt --caffemodel quantized/
**************************************************
* VITIS_AI Compilation - Xilinx Inc.
**************************************************
[INFO] Namespace(batchsize=1, inputs_shape=None, layout='NCHW', model_files=['quantized/deploy.caffemodel'], model_type='caffe', named_inputs_shape=
[INFO] caffe model: /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416/quantized/deploy.caffemodel
[INFO] caffe model: /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416/quantized/deploy.prototxt
[INFO] parse raw model      :100%|██████████████████████████████████████████████████| 275/275 [00:08<00:00,
[INFO] infer shape (NCHW)   :100%|██████████████████████████████████████████████████| 275/275 [00:00<00:00,
[INFO] infer shape (NHWC)   :100%|██████████████████████████████████████████████████| 275/275 [00:00<00:00,
[INFO] perform level-1 opt :100%|██████████████████████████████████████████████████| 3/3 [00:00<00:00, 226.
[INFO] generate xmodel      :100%|██████████████████████████████████████████████████| 275/275 [00:00<00:00,
[INFO] dump xmodel: /tmp/yolov3_mainboard_org.xmodel
[UNILOG][INFO] Target architecture: DPUCZDX8G_ISA0_B3136_MAX_BG2
[UNILOG][INFO] Compile mode: dpu
[UNILOG][INFO] Debug mode: function
[UNILOG][INFO] Target architecture: DPUCZDX8G_ISA0_B3136_MAX_BG2
[UNILOG][INFO] Graph name: deploy, with op num: 575
[UNILOG][INFO] Begin to compile...
[UNILOG][INFO] Total device subgraph number 6, DPU subgraph number 1
[UNILOG][INFO] Compile done.
[UNILOG][INFO] The meta json is saved to "/workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416/compiled/meta.json"
[UNILOG][INFO] The compiled xmodel is saved to "/workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416/compiled//yolov3_mainboard.xmodel"
[UNILOG][INFO] The compiled xmodel's md5sum is 3417400a8016be617c414da39ea6d942, and has been saved to "/workspace/1_KV260_YOLOv3/dk_yolov3_voc_416
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

- 編譯後，得到 xmodel

```
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > tree compiled/
compiled/
├── md5sum.txt
├── meta.json
└── yolov3_mainboard.xmodel

0 directories, 3 files
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

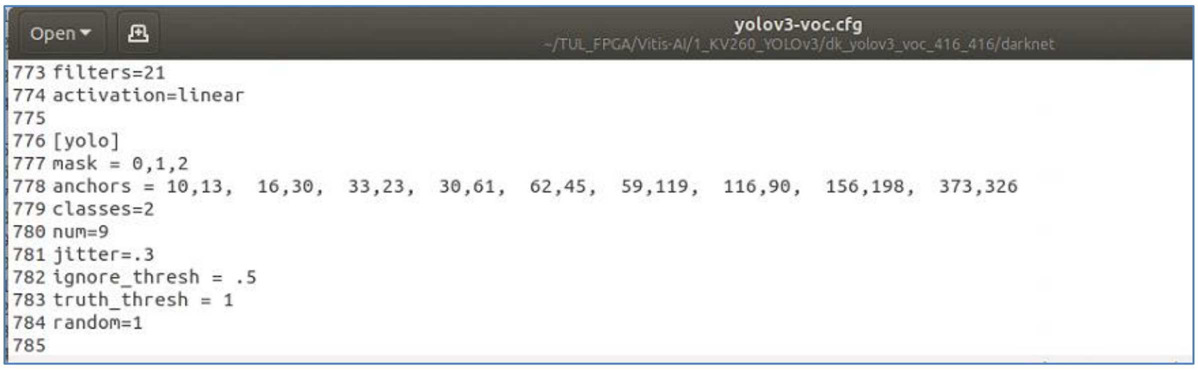
Step05: 建立 Create /compiled/yolov3_mainboard.prototxt

```
model {
  name: "yolov3_mainboard"
  kernel {
     name: "yolov3_mainboard"
     mean: 0.0
     mean: 0.0
     mean: 0.0
     scale: 0.00390625
     scale: 0.00390625
     scale: 0.00390625
  }
  model_type : YOLOv3
  yolo_v3_param {
    num_classes: 2
    anchorCnt: 3
    layer_name: "81"
    layer_name: "93"
    layer_name: "105"
    conf_threshold: 0.3
    nms_threshold: 0.45
    biases: 116
    biases: 90
    biases: 156
    biases: 198
    biases: 373
    biases: 326
    biases: 30
    biases: 61
    biases: 62
    biases: 45
    biases: 59
    biases: 119
    biases: 10
    biases: 13
    biases: 16
    biases: 30
    biases: 33
    biases: 23
    test_mAP: false
   }
}
```

其中：

layer_name 分別為模型 sigmoided_layers 層 81, 93, 105

biases 值，參考 yolov3-voc.cfg anchors 值


Open
yolov3-voc.cfg
~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/darknet

```
filters=21
activation=linear

[yolo]
mask = 0,1,2
anchors = 10,13,  16,30,  33,23,  30,61,  62,45,  59,119,  116,90,  156,198,  373,326
classes=2
num=9
jitter=.3
ignore_thresh = .5
truth_thresh = 1
random=1

```

Step06: 建立 yolov3_mainboard/ 目錄

複製 compiled 所有檔案到 yolov3_mainboard/ 目錄下

```
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > mkdir yolov3_mainboard
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > cp -r compiled/* yolov3_mainboard/
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > tree yolov3_mainboard
yolov3_mainboard
├── md5sum.txt
├── meta.json
├── yolov3_mainboard.prototxt
└── yolov3_mainboard.xmodel

0 directories, 4 files
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

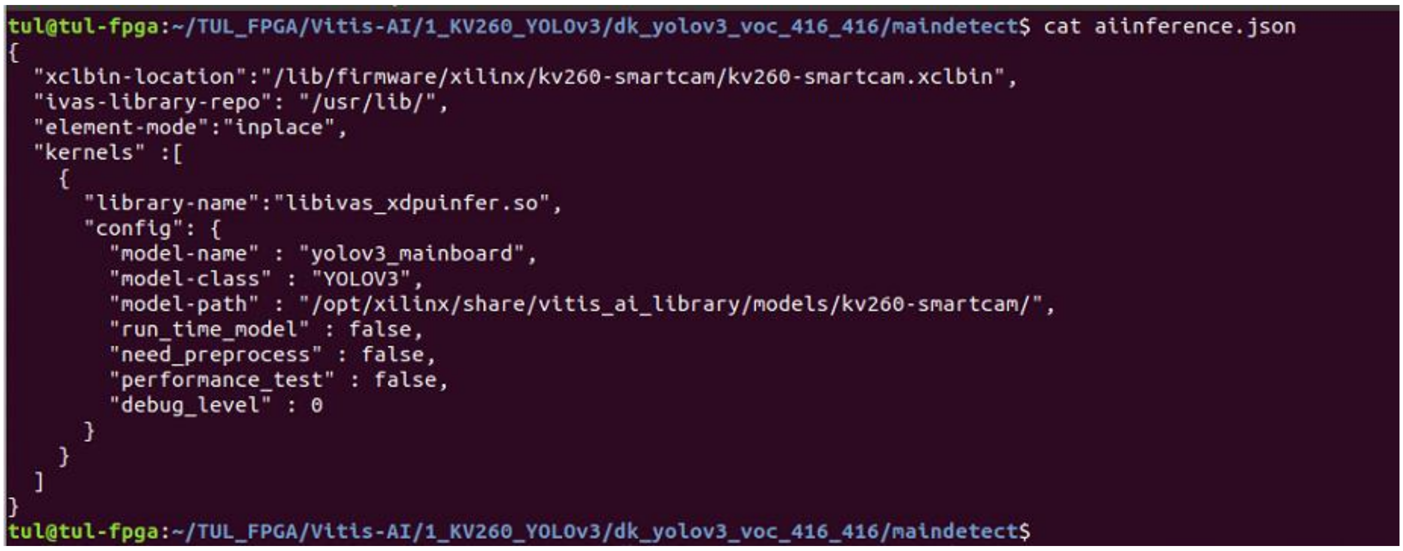
Step07: 建立另一 maindetect/ 目錄

在目錄裡，建立以下 4 個檔案：

```
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > tree maindetect/
maindetect/
├── aiinference.json
├── drawresult.json
├── label.json
└── preprocess.json

0 directories, 4 files
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

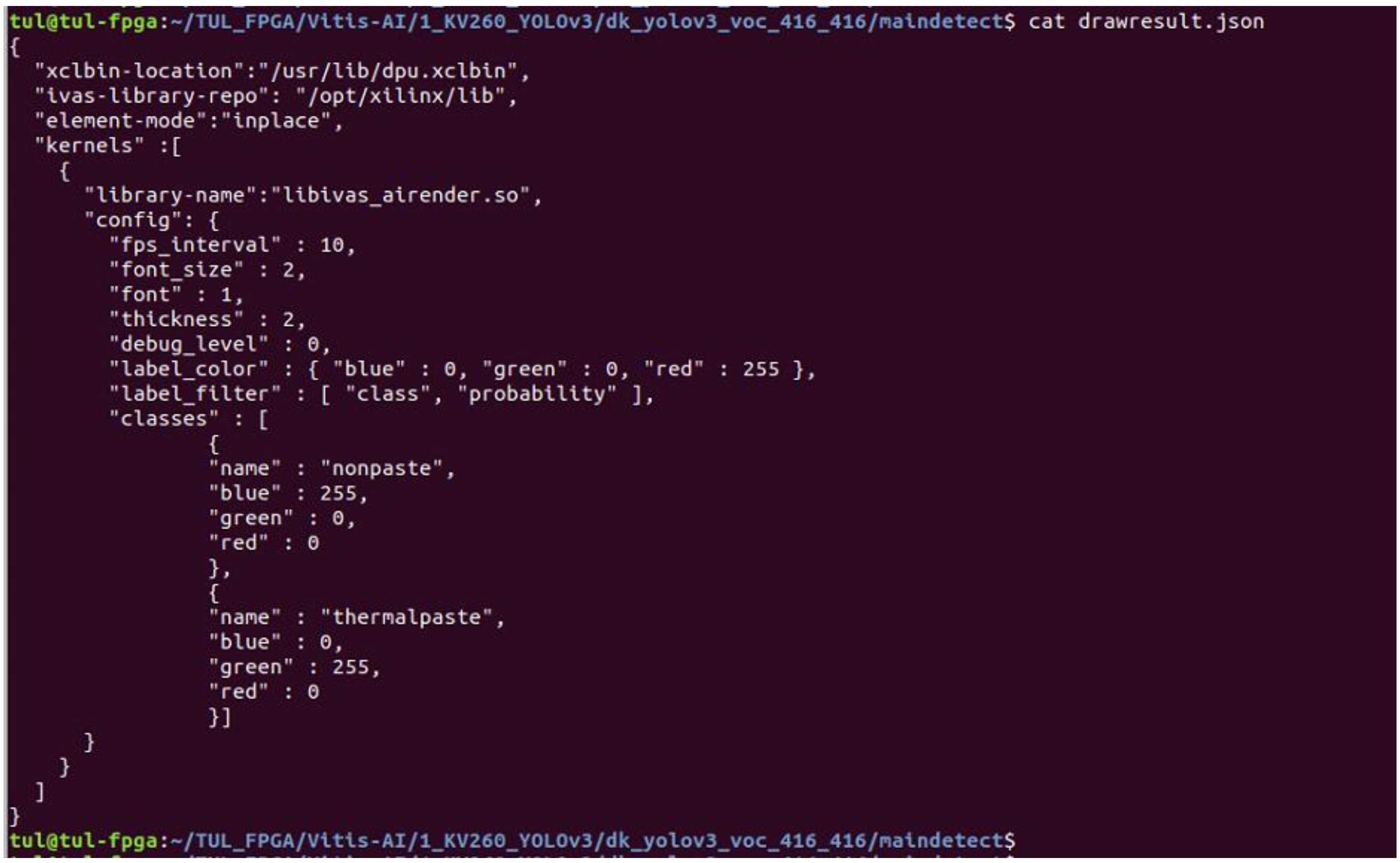
以下為每個檔案內容：

➢ aiinference.json



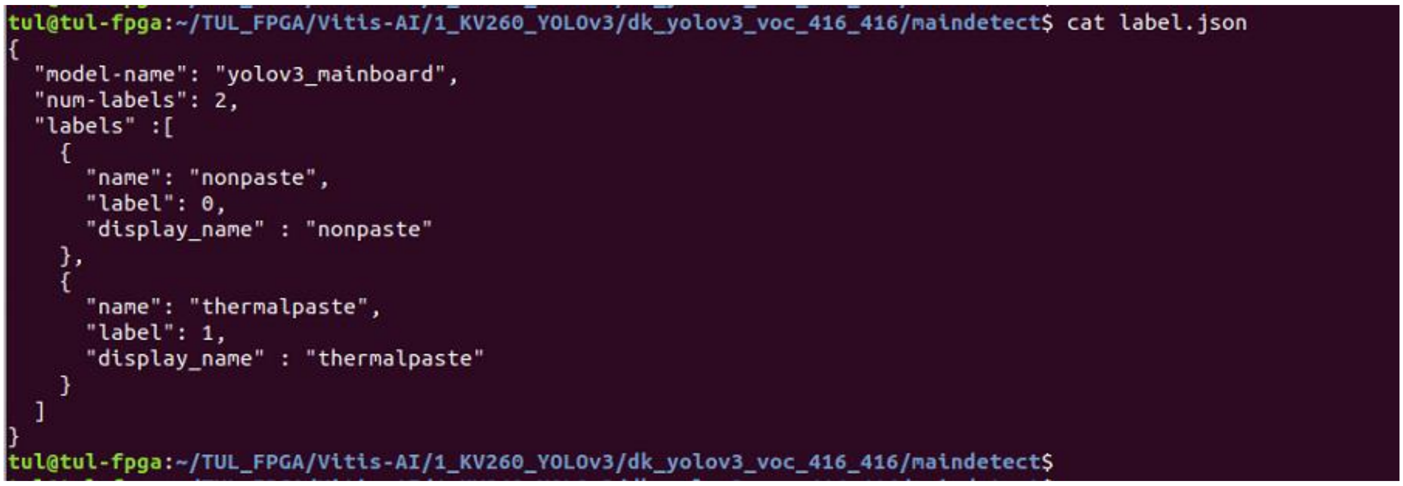
```
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$ cat aiinference.json
{
  "xclbin-location":"/lib/firmware/xilinx/kv260-smartcam/kv260-smartcam.xclbin",
  "ivas-library-repo": "/usr/lib/",
  "element-mode":"inplace",
  "kernels" :[
    {
      "library-name":"libivas_xdpuinfer.so",
      "config": {
        "model-name" : "yolov3_mainboard",
        "model-class" : "YOLOV3",
        "model-path" : "/opt/xilinx/share/vitis_ai_library/models/kv260-smartcam/",
        "run_time_model" : false,
        "need_preprocess" : false,
        "performance_test" : false,
        "debug_level" : 0
      }
    }
  ]
}
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$
```

➢ drawresult.json



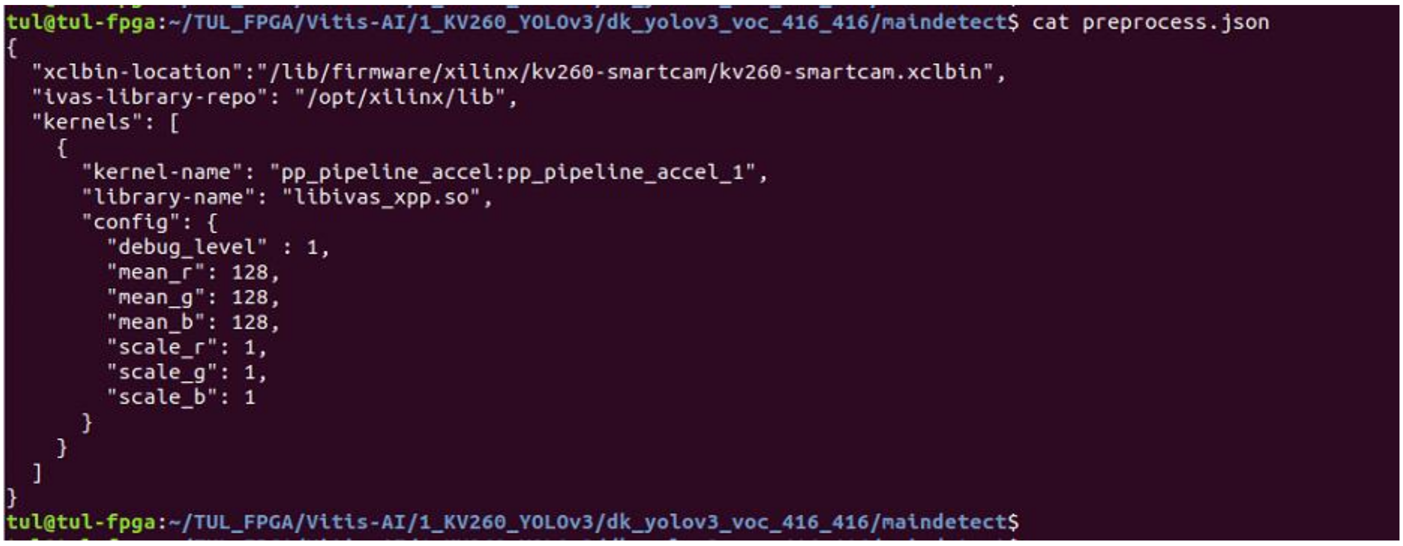
```
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$ cat drawresult.json
{
  "xclbin-location":"/usr/lib/dpu.xclbin",
  "ivas-library-repo": "/opt/xilinx/lib",
  "element-mode":"inplace",
  "kernels" :[
    {
      "library-name":"libivas_airender.so",
      "config": {
        "fps_interval" : 10,
        "font_size" : 2,
        "font" : 1,
        "thickness" : 2,
        "debug_level" : 0,
        "label_color" : { "blue" : 0, "green" : 0, "red" : 255 },
        "label_filter" : [ "class", "probability" ],
        "classes" : [
                {
                "name" : "nonpaste",
                "blue" : 255,
                "green" : 0,
                "red" : 0
                },
                {
                "name" : "thermalpaste",
                "blue" : 0,
                "green" : 255,
                "red" : 0
                }]
      }
    }
  ]
}
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$
```

➢ label.json



```
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$ cat label.json
{
  "model-name": "yolov3_mainboard",
  "num-labels": 2,
  "labels" :[
    {
      "name": "nonpaste",
      "label": 0,
      "display_name" : "nonpaste"
    },
    {
      "name": "thermalpaste",
      "label": 1,
      "display_name" : "thermalpaste"
    }
  ]
}
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$
```

➢ preprocess.json



```
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$ cat preprocess.json
{
  "xclbin-location":"/lib/firmware/xilinx/kv260-smartcam/kv260-smartcam.xclbin",
  "ivas-library-repo": "/opt/xilinx/lib",
  "kernels": [
    {
      "kernel-name": "pp_pipeline_accel:pp_pipeline_accel_1",
      "library-name": "libivas_xpp.so",
      "config": {
        "debug_level" : 1,
        "mean_r": 128,
        "mean_g": 128,
        "mean_b": 128,
        "scale_r": 1,
        "scale_g": 1,
        "scale_b": 1
      }
    }
  ]
}
tul@tul-fpga:~/TUL_FPGA/Vitis-AI/1_KV260_YOLOv3/dk_yolov3_voc_416_416/maindetect$
```

C. KV260 AOI 檢測

Step01: SDCard 設定

參考 Link:
https://xilinx.github.io/kria-apps-docs/main/build/html/docs/smartcamera/docs/app_deployment.html
燒錄官方提供 .img 到 SDCard 後，啟動系統

開機，輸入名稱和密碼
名稱：petalinux
密碼：tul
(密碼第一次開機由使用者設定)

Step02: 取得 application package

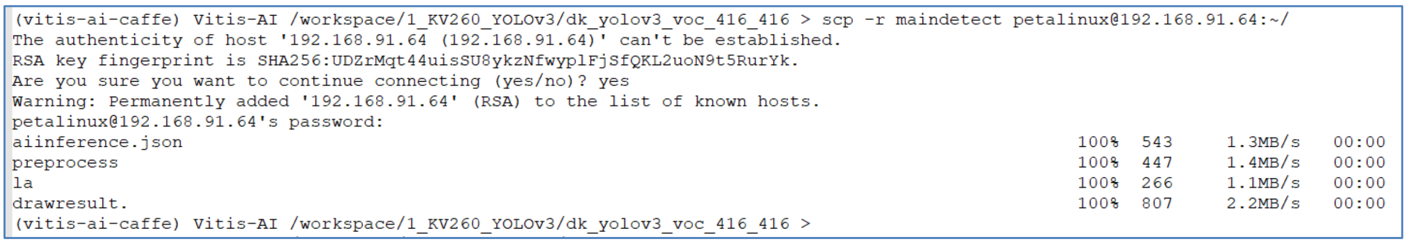
```
sudo xmutil getpkgs
```

Step03: 安裝 kv260 package

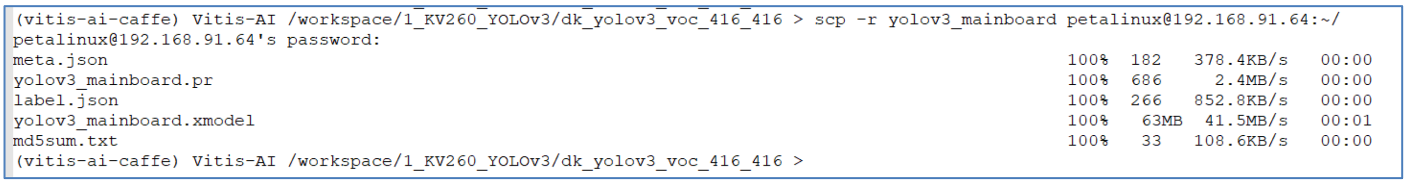
```
sudo dnf install packagegroup-kv260-smartcam.noarch
```

Step04: 複製檔案到 SDCard

使用以下指令：(kv260 SDCard IP 為 192.168.91.64)
scp -r yolov3_mainboard petalinux@192.168.91.64:~/
scp -r maindetect petalinux@192.168.91.64:~/



```
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > scp -r maindetect petalinux@192.168.91.64:~/
The authenticity of host '192.168.91.64 (192.168.91.64)' can't be established.
RSA key fingerprint is SHA256:UDZrMqt44uisSU8ykzNfwyplFjSfQKL2uoN9t5RurYk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.91.64' (RSA) to the list of known hosts.
petalinux@192.168.91.64's password:
aiinference.json                                                                      100%  543     1.3MB/s   00:00
preprocess                                                                            100%  447     1.4MB/s   00:00
la                                                                                    100%  266     1.1MB/s   00:00
drawresult.                                                                           100%  807     2.2MB/s   00:00
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```



```
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 > scp -r yolov3_mainboard petalinux@192.168.91.64:~/
petalinux@192.168.91.64's password:
meta.json                                                                            100%  182   378.4KB/s   00:00
yolov3_mainboard.pr                                                                  100%  686     2.4MB/s   00:00
label.json                                                                           100%  266   852.8KB/s   00:00
yolov3_mainboard.xmodel                                                              100%   63MB  41.5MB/s   00:01
md5sum.txt                                                                           100%   33   108.6KB/s   00:00
(vitis-ai-caffe) Vitis-AI /workspace/1_KV260_YOLOv3/dk_yolov3_voc_416_416 >
```

Step05: 模型配置

執行以下指令：
sudo scp -r maindetect /opt/xilinx/share/ivas/smartcam/
sudo scp -r yolov3_mainboard /opt/xilinx/share/vitis_ai_library/models/kv260-smartcam/

Step06: 啟動 kv260-smartcam application

sudo xmutil unloadapp
sudo xmutil loadapp kv260-smartcam

Step07: 執行程式

```
xilinx-k26-starterkit-2020_2:~$ sudo su -l root
sudo xmutil      unloadapp
sudo xmutil      loadapp kv260-smartcam
```

#FOR video file:

```
sudo smartcam --file /home/petalinux/test.h264 -i h264 -W 1920 -H 1080 -r 30 --target dp --aitask maindetect
```

#FOR mipi camera:

```
sudo smartcam --mipi -W 1920 -H 1080 --target dp --aitask maindetect
```