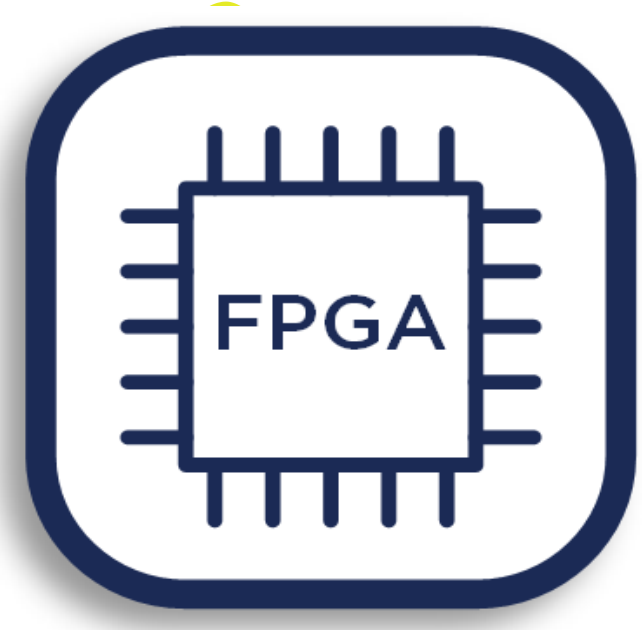
Multi-Video Analytics System Based on FPGA

Name: Sabol Socare
ID. : 62050104

FPGA Infographics



01 Flexible

02 Parallelism

03 Reliability

04 Cost Efficient

05 Better Performance

06 Programmability

Applications on FPGA

Some FPGA applications

- **Medical** — Diagnostic and monitoring purposes
- **Server & *Cloud*** — Data analytics and metadata and data processing
- **Defense & Space** — Used for high-resolution optical data processing and radar imaging
- **Video & Image Processing** — Accelerate image and video processing applications while meeting low-power constraints

Multi-Video Analytics System Based on FPGA

Project's objectives

Customized AI model

01

Compile all 8 models which are supported VVAS framework

Multi-Video Analytics

02

Develop multi-video analytics system for Detection and Classification in security system

Retrain AI models

03

Utilize AI model for detection and classification for testing performance on CPU,GPU,FPGA

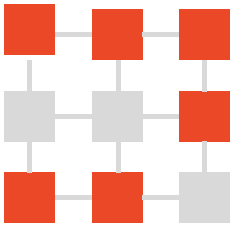
Multi-Video Analytics System Based on FPGA

01

Customized AI models

8 out of 16 models which are supported VVAS

02



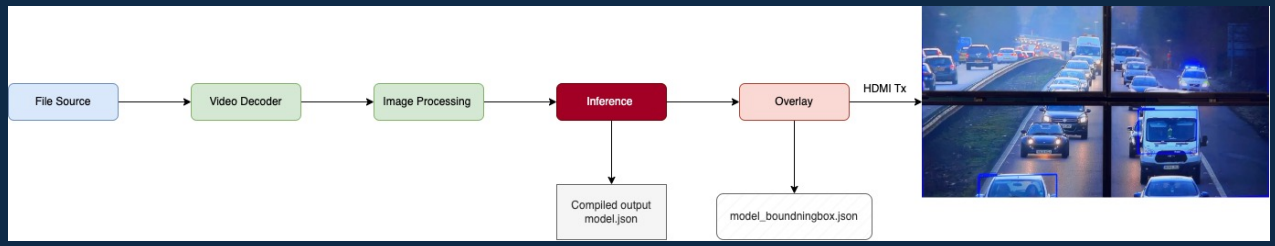
Multichannel Rendering Video

Single Stage ML

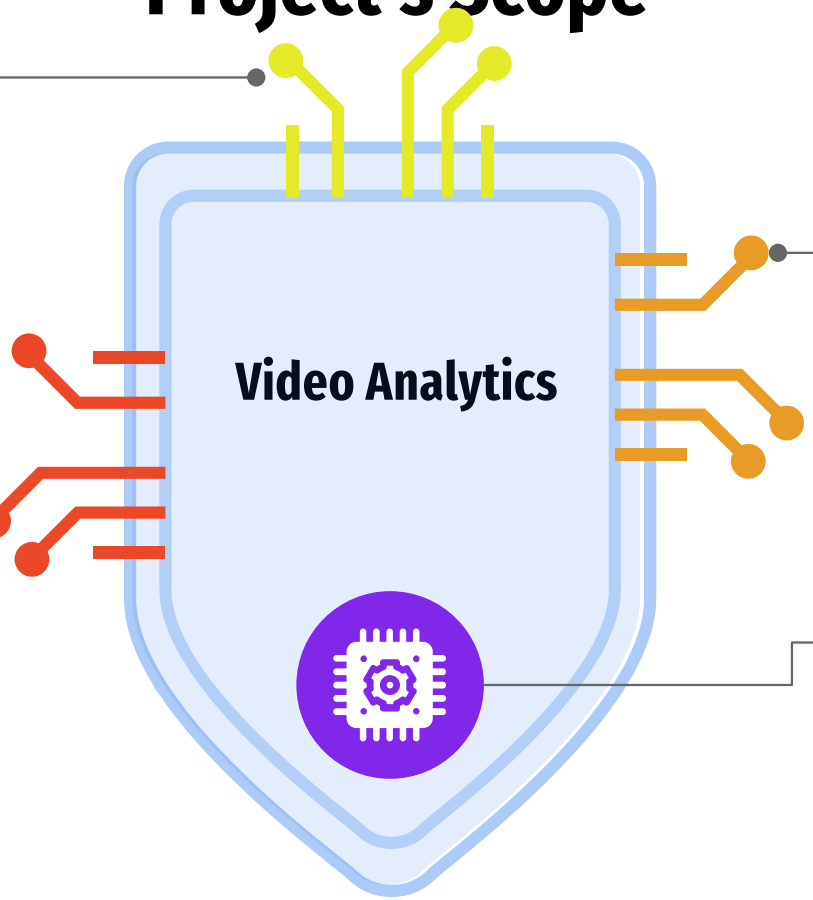
What is Single stage ML ?

Single Stage ML (Machine Learning) is ML operation that ivolve only one ML operation on the input image.

So we can scale up to build 4 channels video analytics by process 4 streams in parallel.



Project's Scope



01

Be able to process on FPGA which is ZCU104 Board

02

Customized AI model for classification and detection application

03

Compile AI for using on ZCU104 board

04

Train AI models for testing performance on CPU, GPU and FPGA

Project Equipment

01 ZCU104 Evaluation Board rev 1.0

02 Micro USB wire

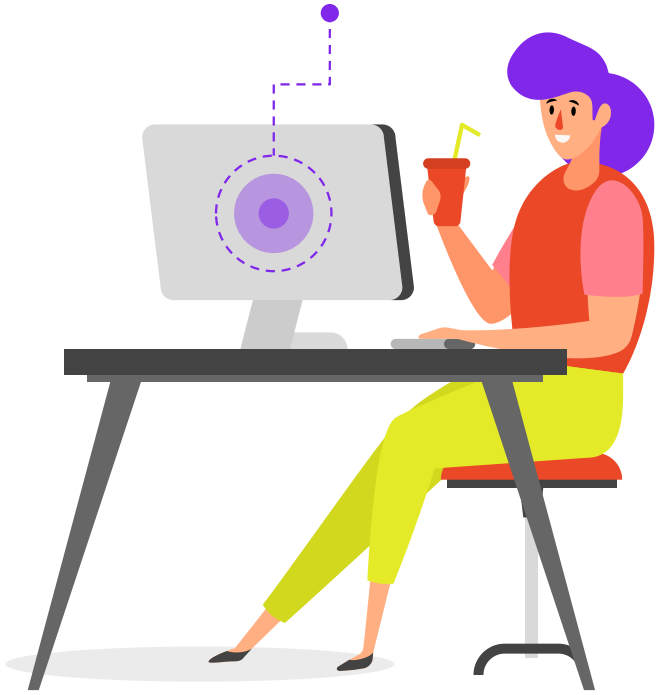
03 HDMI 2.0

04 HDMI 2.0 Wire

05 MicroSD Card

AI

System configuration



Vitis Unified Software platform version 2022.2 01

PetaLinux tool version 2022.2 02

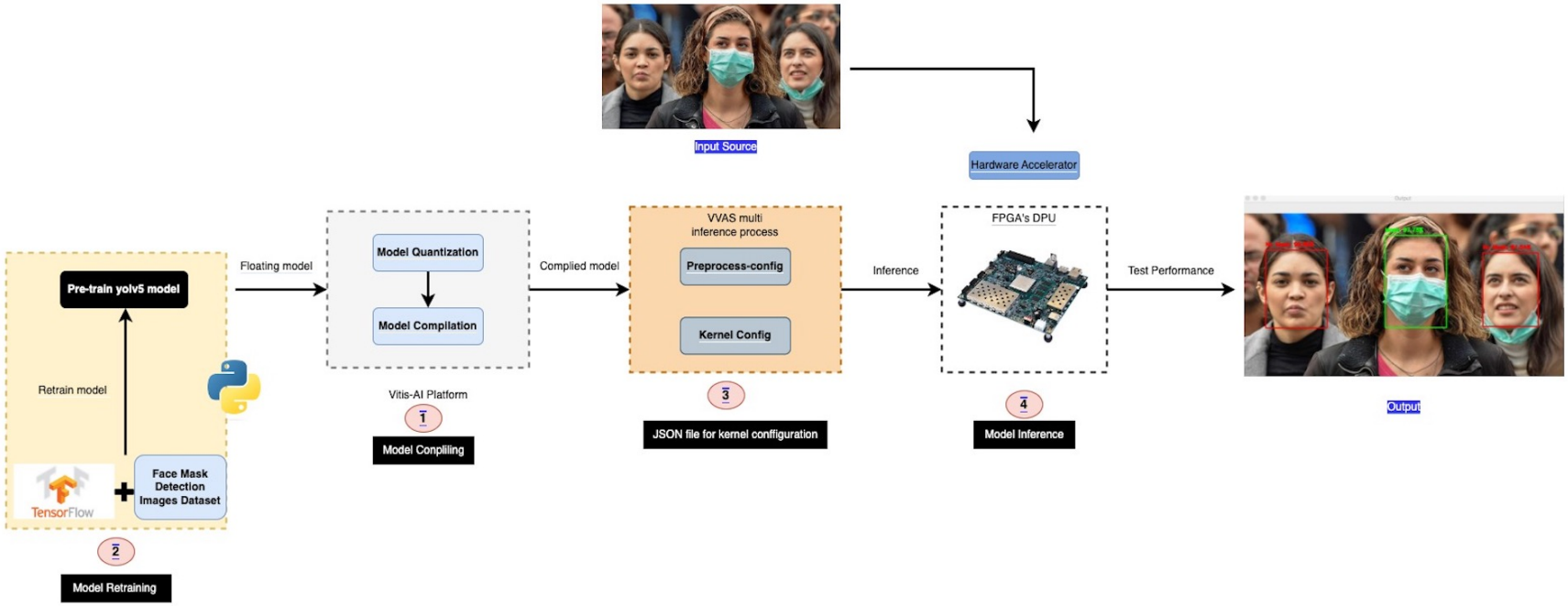
Serial terminal emulator 03

Git 04

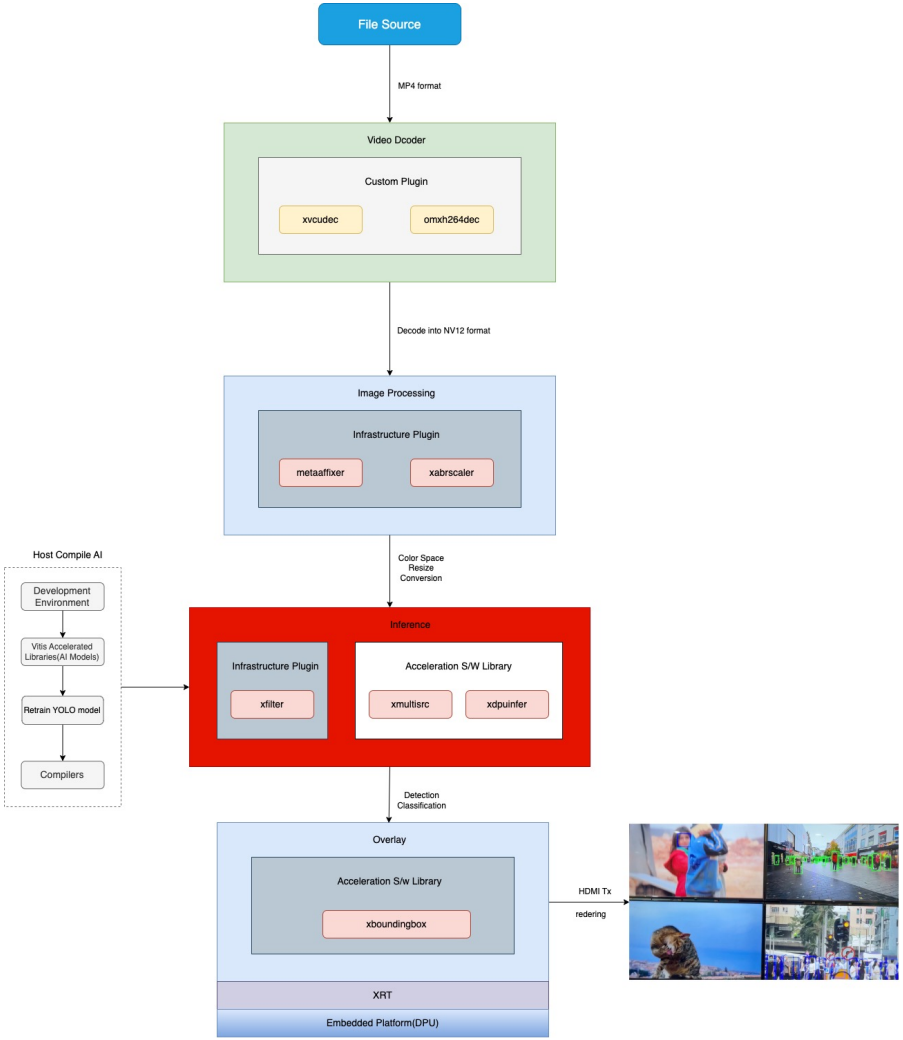
MacOS/Ubuntu 18.04/20.04 05

Balena Etcher Flashing tool 06

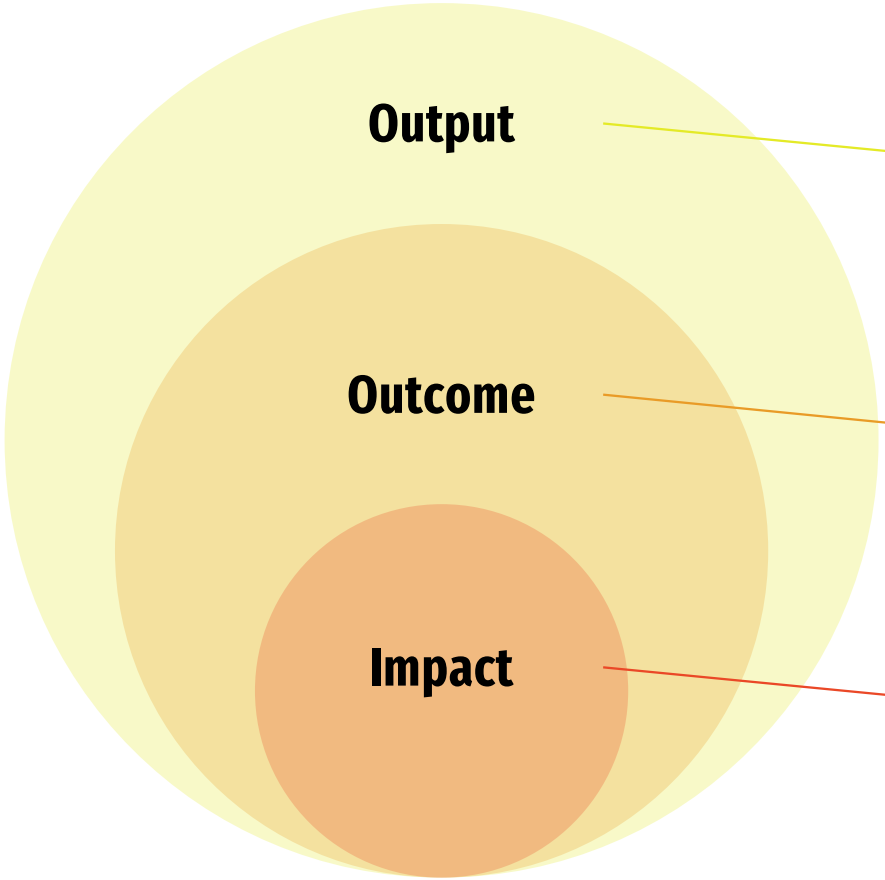
Project Overall's diagram

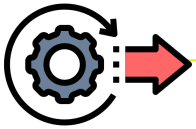


AIC's Infrastructure



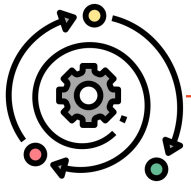
Output/Outcome/Impact





Multi-model processing on FPGA architecture

Multi-channel AI processing with high-speed processing on specialized architecture



Deployment of AI applications on edge computing with low latency

Experiment test

01

Compile AI models

Compiling AI model from model Zoo, Xilinx

02

Multi-video analytics

Analyze multi-video on FPGA

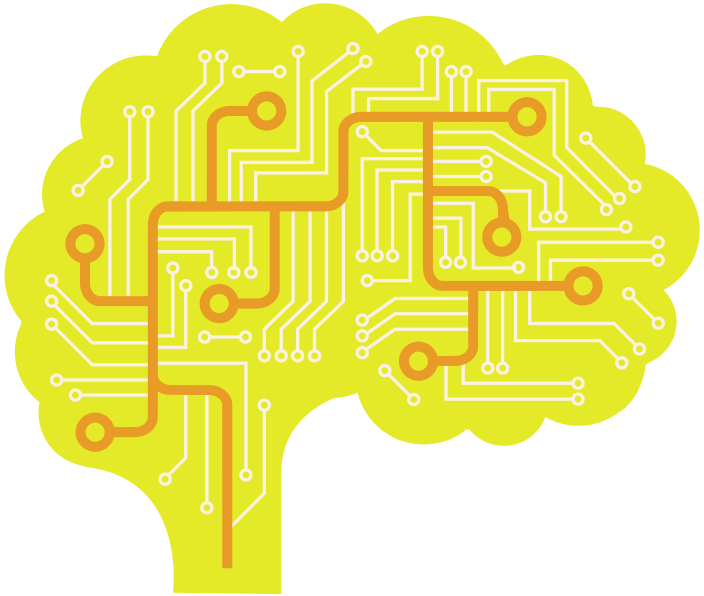
03

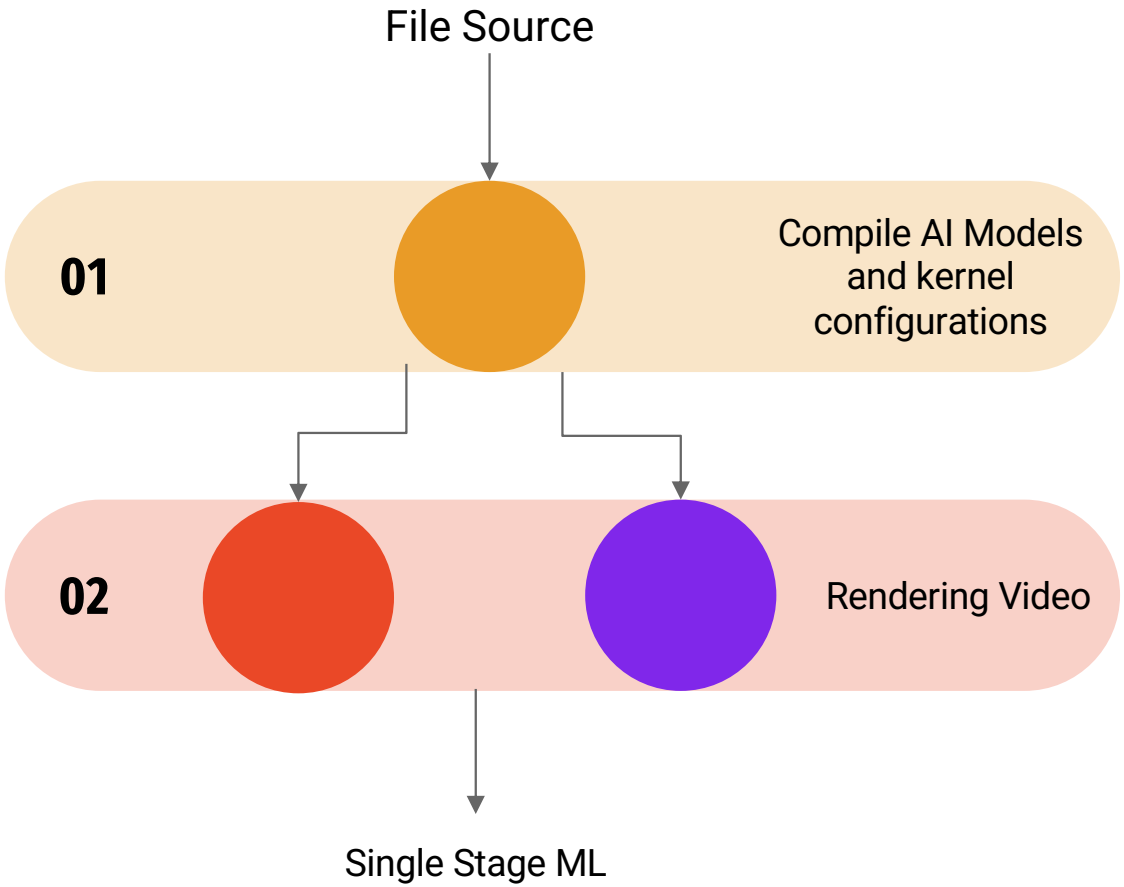
Train AI models

Train and test performance with AI models

Experiment Result

Test result





1. Compile AI model

Cross compile

```
(vitis-ai-caffe) Vitis-AI /workspace/vvas_model > source compile_cf_model.sh ssd_traffic_pruned_0_9  cf_
ssdtraffic_360_480_0.9_11.6G_2.0
**************************************************
* VITIS_AI Compilation - Xilinx Inc.
**************************************************
[INFO] Namespace(batchsize=1, inputs_shape=None, layout='NCHW', model_files=['../models/AI-Model-Zoo/caf
fe_model/cf_ssdtraffic_360_480_0.9_11.6G_2.0/quantized/deploy.caffemodel'], model_type='caffe', named_in
puts_shape=None, out_filename='/tmp/ssd_traffic_pruned_0_9_org.xmodel', proto='../models/AI-Model-Zoo/ca
ffe_model/cf_ssdtraffic_360_480_0.9_11.6G_2.0/quantized/deploy.prototxt')
[INFO] caffe model: /workspace/models/AI-Model-Zoo/caffe_model/cf_ssdtraffic_360_480_0.9_11.6G_2.0/quant
ized/deploy.caffemodel
[INFO] caffe model: /workspace/models/AI-Model-Zoo/caffe_model/cf_ssdtraffic_360_480_0.9_11.6G_2.0/quant
ized/deploy.prototxt
[INFO] parse raw model     :100%|████████████████████| 123/123 [00:00<00:00, 332.52it/s]
[INFO] infer shape (NCHW)  :100%|████████████████████| 123/123 [00:00<00:00, 1423.88it/s]
[INFO] infer shape (NHWC)  :100%|████████████████████| 123/123 [00:00<00:00, 1619.24it/s]
[INFO] perform level-1 opt :100%|████████████████████| 3/3 [00:00<00:00, 206.10it/s]
[INFO] generate xmodel     :100%|████████████████████| 123/123 [00:00<00:00, 2264.00it/s]
[INFO] dump xmodel: /tmp/ssd_traffic_pruned_0_9_org.xmodel
[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: 263
[UNILOG][INFO] Begin to compile...
[UNILOG][INFO] Total device subgraph number 5, DPU subgraph number 1
[UNILOG][INFO] Compile done.
[UNILOG][INFO] The meta json is saved to "/workspace/vvas_model/./compiled_output/cf_ssdtraffic_360_480_
0.9_11.6G_2.0/meta.json"
[UNILOG][INFO] The compiled xmodel is saved to "/workspace/vvas_model/./compiled_output/cf_ssdtraffic_36
0_480_0.9_11.6G_2.0/ssd_traffic_pruned_0_9.xmodel"
[UNILOG][INFO] The compiled xmodel's md5sum is f91e919eb6beba63de3fb34f42e5a907, and has been saved to "
/workspace/vvas_model/./compiled_output/cf_ssdtraffic_360_480_0.9_11.6G_2.0/md5sum.txt"
(vitis-ai-caffe) Vitis-AI /workspace/vvas_model >
```

Test jpec

```
root@xlnx-zcu104-vDec-hdmiTx-2021-2:~/Vitis-AI-Library/samples/ssd# ./test_jpeg_ssd ssd_traffic_pruned_0_9 sample_ssd.jpg
WARNING: Logging before InitGoogleLogging() is written to STDERR
I0919 07:30:46.808861  2247 demo.hpp:1183] batch: 0     image: sample_ssd.jpg
I0919 07:30:46.809010  2247 process_result.hpp:43] RESULT: 1    460.372 319.02  518.265 360.097 0.964248
I0919 07:30:46.809141  2247 process_result.hpp:43] RESULT: 1    549.768 330.249 649.877 383.833 0.953907
I0919 07:30:46.809193  2247 process_result.hpp:43] RESULT: 1    627.717 310.577 797.316 543.705 0.953907
I0919 07:30:46.809262  2247 process_result.hpp:43] RESULT: 1    499.001 330.743 561.403 368.852 0.951322
I0919 07:30:46.809309  2247 process_result.hpp:43] RESULT: 1    8.27212 307.261 204.645 529.78  0.77812
I0919 07:30:46.809372  2247 process_result.hpp:43] RESULT: 1    569.494 297.068 737.173 365.71  0.6773
I0919 07:30:46.809427  2247 process_result.hpp:43] RESULT: 1    175.256 320.477 296.409 413.135 0.62172

root@xlnx-zcu104-vDec-hdmiTx-2021-2:~/Vitis-AI-Library/samples/ssd#
```

Test Performance

```
root@xlnx-zcu104-vDec-hdmiTx-2021-2:~/Vitis-AI-Library/samples/ssd# ./test_performance_ssd ssd_traffic_pruned_0_9 ./test_performance_ssd.list
WARNING: Logging before InitGoogleLogging() is written to STDERR
I0928 02:46:15.769080  3112 benchmark.hpp:184] writing report to <STDOUT>
I0928 02:46:15.769455  3112 benchmark.hpp:211] waiting for 0/30 seconds, 1 threads running
I0928 02:46:25.769587  3112 benchmark.hpp:211] waiting for 10/30 seconds, 1 threads running
I0928 02:46:35.769747  3112 benchmark.hpp:211] waiting for 20/30 seconds, 1 threads running
I0928 02:46:45.770006  3112 benchmark.hpp:219] waiting for threads terminated
FPS=46.5669
E2E_MEAN=21454.4
DPU_MEAN=17348.1
```

** Compiling AI model (Model Zoo)

Compile AI model

ตารางโมเดลที่หลังจากปรับแต่งทดสอบประสิทธิภาพของแต่ละโมเดล

ลำดับที่	ชื่อโมเดล	ทดสอบภาพ	ทดสอบประสิทธิภาพ
1	resnet18	ผ่าน	FPS= 68.8985
2	mobilenet_v2	ผ่าน	FPS= 239.659
3	inception_v1	ผ่าน	FPS= 148.714
4	ssd_adas_pruned_0_95	ผ่าน	FPS= 77.8735
5	ssd_mobilenet_v2	ผ่าน	FPS= 22.9255
6	ssd_pedestrian_pruned_0_97	ผ่าน	FPS= 78.0783
7	yolov3_voc_tf	ผ่าน	FPS= 10.0492
8	densebox_640_360	ผ่าน	FPS= 179.286

2. Multi-video analytics

Customize JSON file for rendering video

Kernels configuration

01

Preprocess-config

Is the core interface between xinfer plug-in and DPU

02

kernels-config

Is the core interface between xfilter plug-in and DPU

Customize JSON file

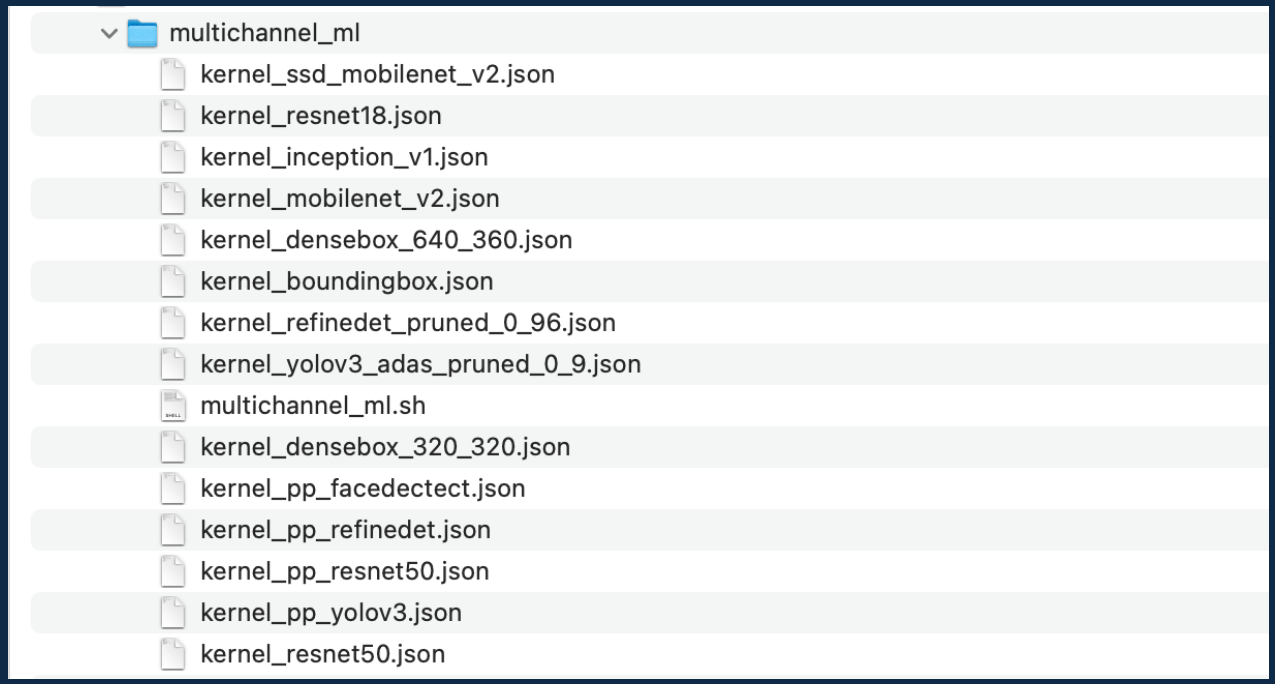
```
{
  "vvas-library-repo": "/usr/lib/",
  "inference-level": 1,
  "attach-ppe-outbuf": false,
  "kernel" : {
     "library-name":"libvvas_xdpuinfer.so",
     "config": {
        "batch-size" : 0,
        "model-name" : "densebox_320_320",
        "model-class" : "FACEDETECT",
        "model-format" : "BGR",
        "model-path" : "/usr/share/vitis_ai_library/models/",
        "run_time_model" : false,
        "need_preprocess" : false,
        "performance_test" : false,
        "debug_level" : 1,
        "max-objects":3
     }
  }
}
```

Preprocess-config

```
{
  "ivas-library-repo": "/usr/lib/",
  "element-mode":"inplace",
  "kernels" :[
    {
      "library-name":"libvvas_xboundingbox.so",
      "config": {
        "model-name" : "densebox_320_320",
        "display_output" : 1,
        "font_size" : 0.5,
        "font" : 3,
        "thickness" : 3,
        "debug_level" : 0,
        "label_color" : { "blue" : 0, "green" : 0, "red" : 0 },
        "label_filter" : [ "class", "probability" ],
        "classes" : [
        ]
      }
    }
  ]
}
```

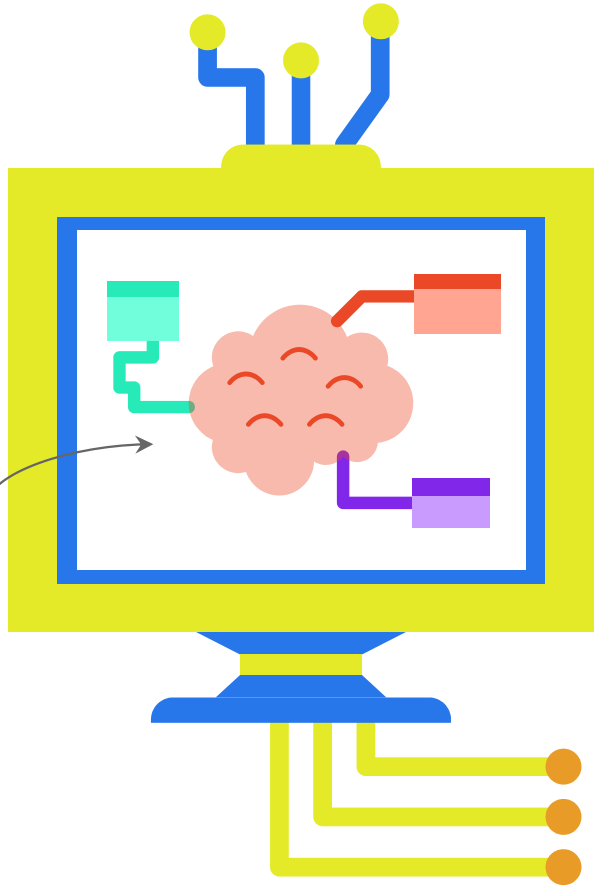
kernels-config

Compile AI model

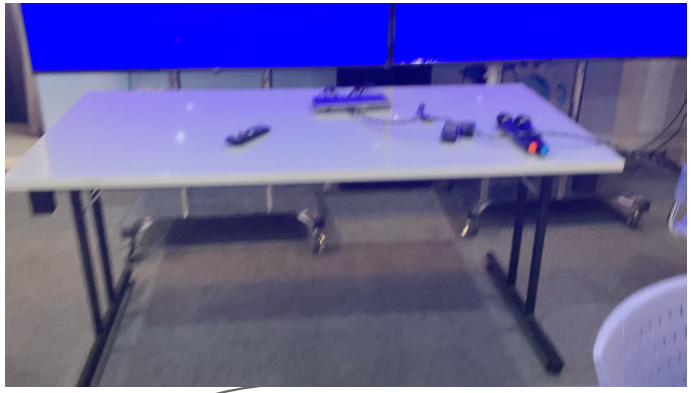


Multichannel Machine Learning

Single Stage



Cascaded



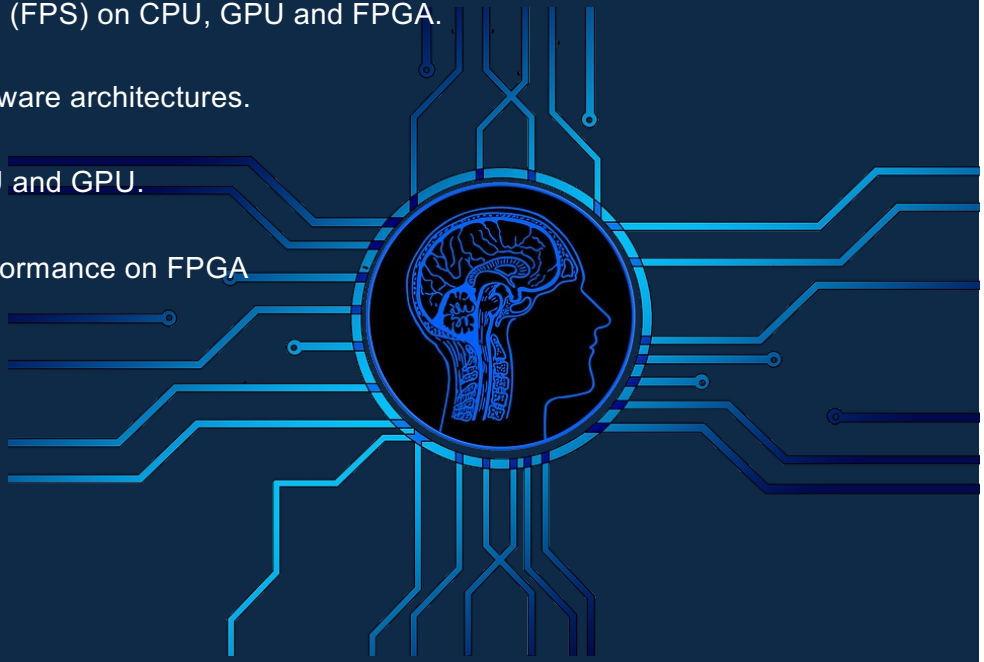
3. Train and test performance with AI models

- We present a comparative analysis of the performance of Frames Per Second (FPS) on CPU, GPU and FPGA.

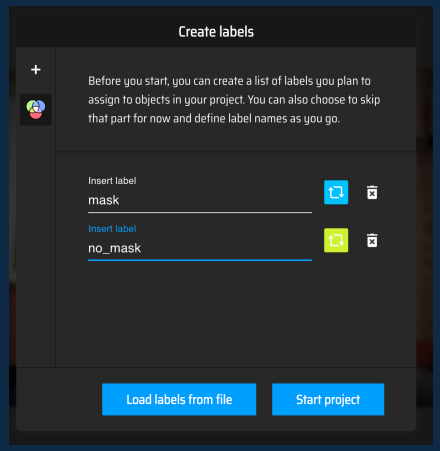
- The FPS performance was measured and compared across the different hardware architectures.

- Train Yolov5 model using PyTorch framework for testing performance on CPU and GPU.

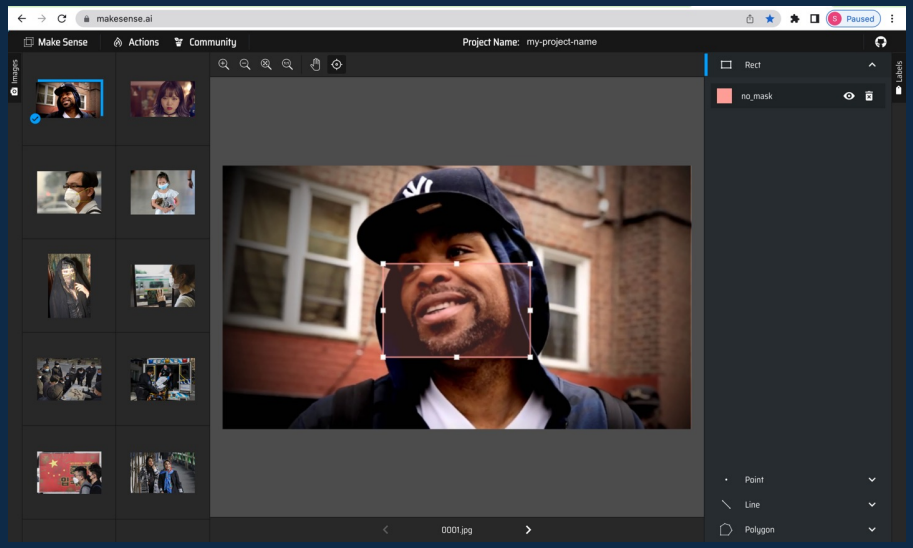
- Train Object classification model using Tensorflow2 framework for testing performance on FPGA



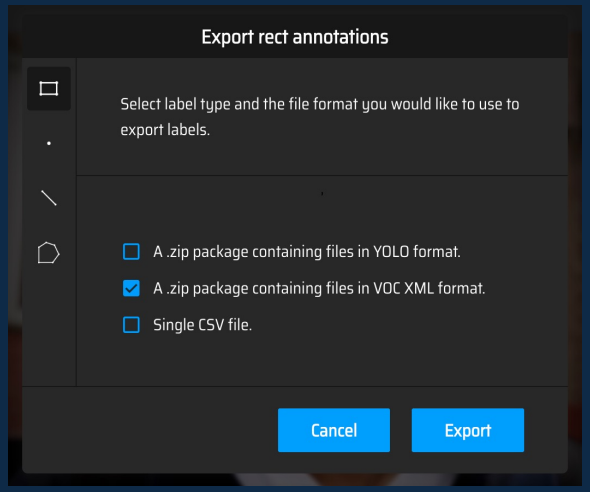
Prepare Dataset (Yolov5)



1



2



3

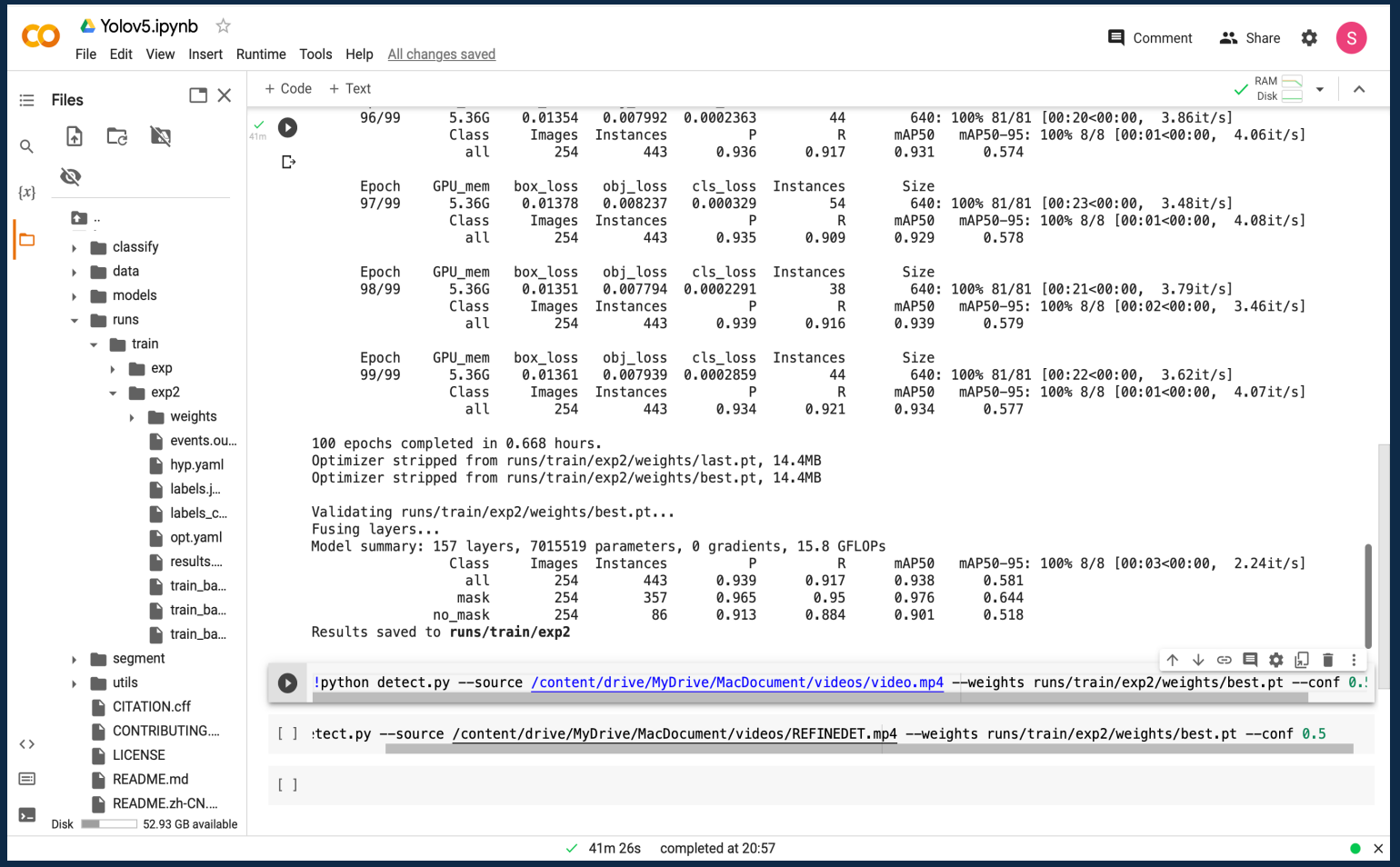
Prepare Dataset

```
(base) Socares-MacBook-Pro:train_data_voc socaresabol$ tree -d
.
├── images
│   ├── train
│   └── val
└── labels
    ├── train
    └── val
```

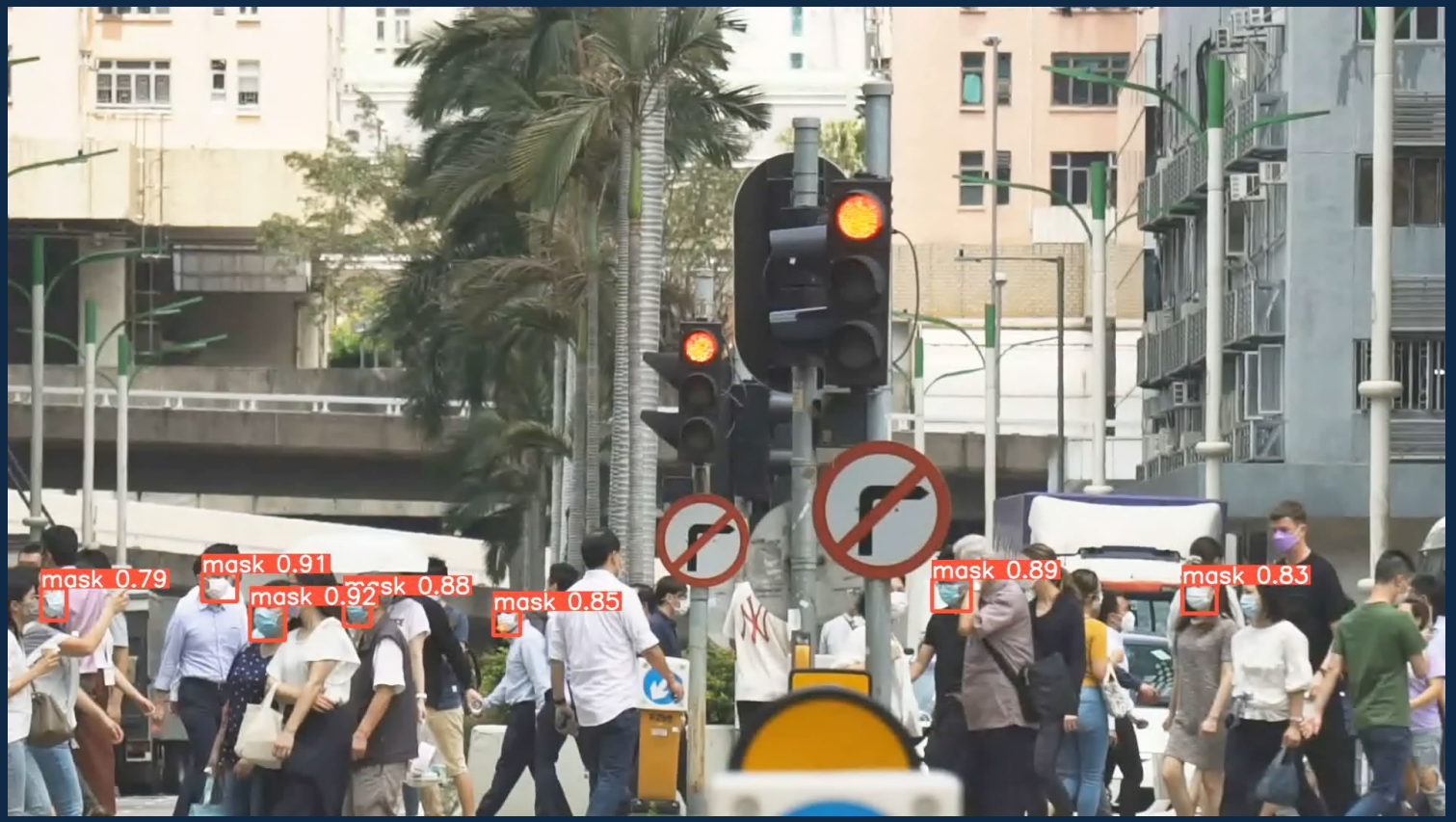
Note:

- Train images = 1295 images
- Val images = 254 images

Training with Yolov5 model



Detect Video



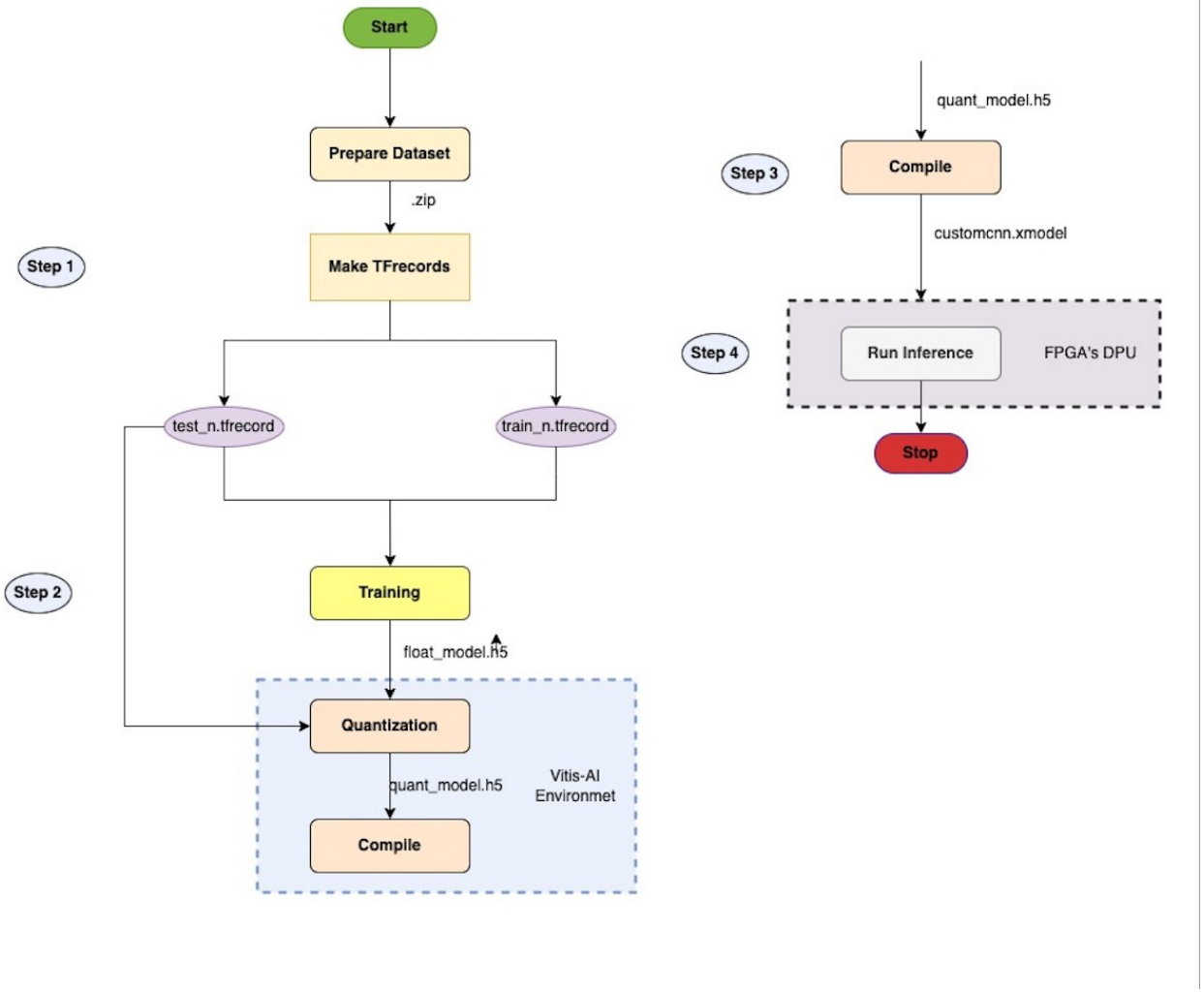
Test performance on CPU and GPU

- ผลลัพธ์จากการนำโมเดล Yolov5 (detection) มาทดสอบประสิทธิภาพบน CPU กับ GPU

ลำดับที่	สถาปัตยกรรมฮาร์ดแวร์	FPS(fps)
1	Intel(R) Xeon(R) (Colab)	3.6193
2	NVIDIA V100 (Colab)	64.1026

Note: FPS = 1/(Inference time)

Training Object Classification

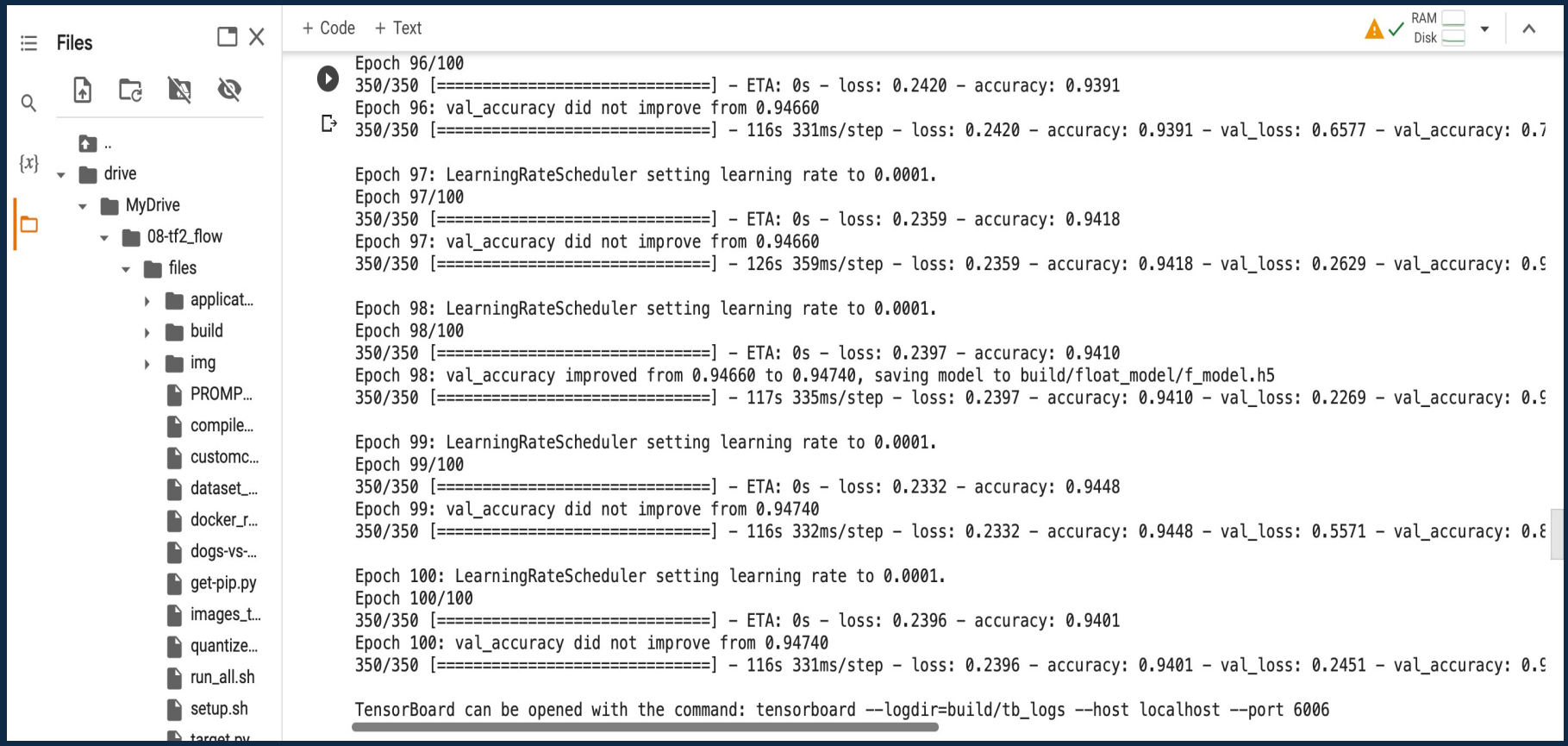


Prepare Dataset (Object Classification)

1. Download dogs-vs-cats dataset from Kaggle.
2. dogs-vs-cat dataset consists of 25000 images of varying dimensions, divided into the two classes of cat and dog
3. Each image is intrinsically labelled or classified by its filename

Example: the image with filename cat.12.jpg is obviously of class cat

Training Object Classification model



Train Classification model

Command line options:

--input_height : 200

--input_width : 250

--input_chan : 3

--tfrec_dir : build/tfrecords

--batchsize : 50

--learnrate : 0.001

--epochs : 100

--chkpt_dir : build/float_model

Output :

- f_model.h5
- accuracy: 0.9401
- val_accuracy: 0.9390

Quantization Classification model

```
------------------------------------
Command line options:
 --float_model  :  build/float_model/f_model.h5
 --quant_model  :  build/quant_model/q_model.h5
 --batchsize    :  50
 --tfrec_dir    :  build/tfrecords
 --evaluate     :  True
------------------------------------
```

** Quantized model was saved as: q_model.h5

Compile Classification model

```
------------------------------------
Command line options:
 --target_dir    :  ./build/target_zcu104
 --image_dir     :  build/dataset/test
 --input_height  :  200
 --input_width   :  250
 --num_images    :  1000
 --app_dir       :  application
 --model         :  ./build/compiled_zcu104/customcnn.xmodel
------------------------------------
```

** Compiled model was saved as: customcnn.xmodel

Inference on FPGA board

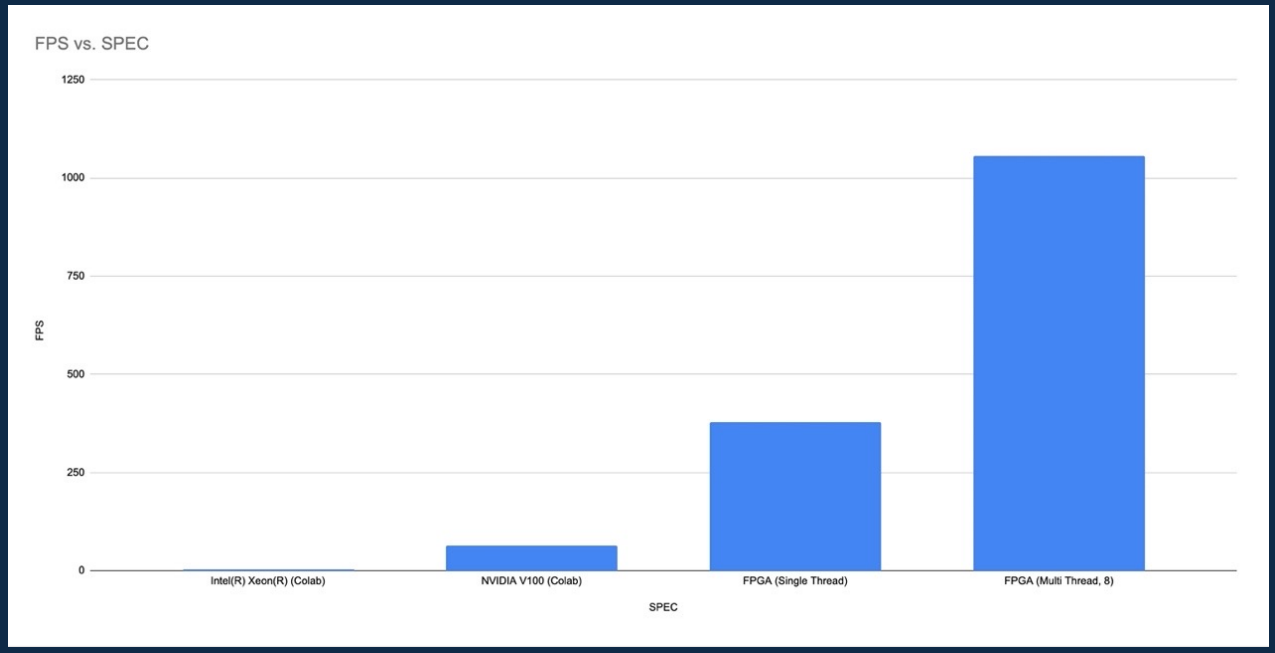
```
Command line options:
 --image_dir :  images
 --threads   :  1
 --model     :  customcnn.xmodel
------------------------------------
Pre-processing 1000 images...
Starting 1 threads...
------------------------------------
```

** Output

- ผลลัพธ์จากการนำโมเดล Classification มาทดสอบประสิทธิภาพบน FPGA

ลำดับที่	สถาปัตยกรรมฮาร์ดแวร์	FPS(fps)
1	FPGA (Single Thread)	377.79
2	FPGA (Multi Thread, 8)	1055.84

Performance Comparative graph



** Comparing each models on a graph

THANKS!

Do you have any questions?

sabolsocare1028@gmail.com
+66 805535324
62050104@go.buu.ac.th

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.