

HAILO DOCUMENTATION (VER 1)

MỤC LỤC

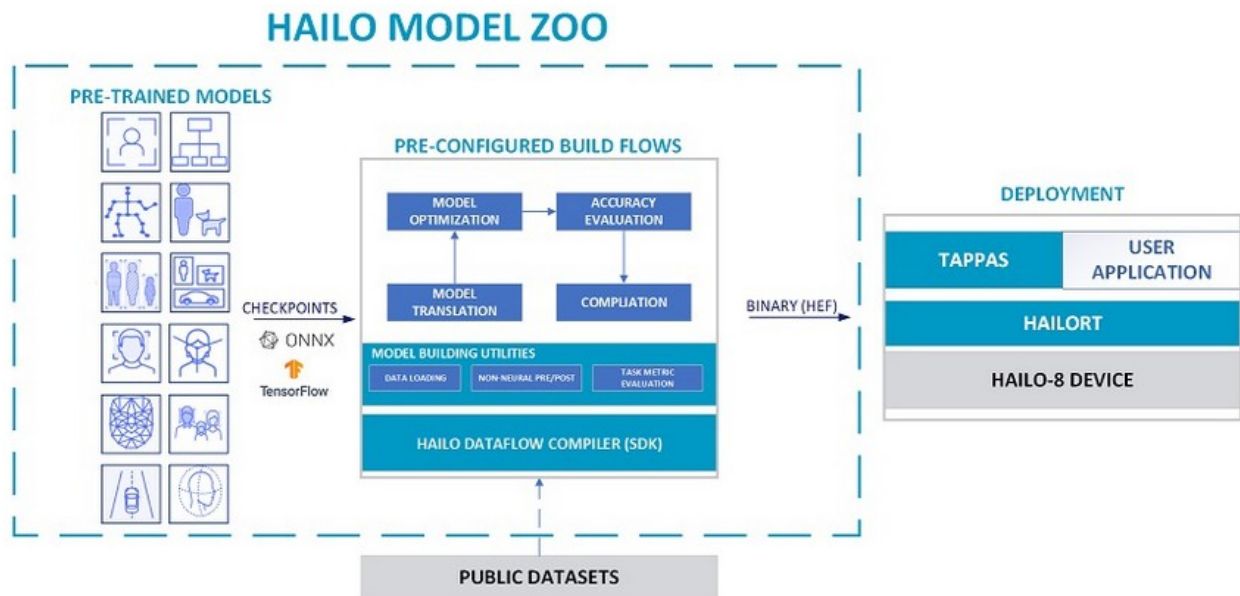
I. HAILO SDK

II . HAILO TAPPAS

I. HAILO SDK

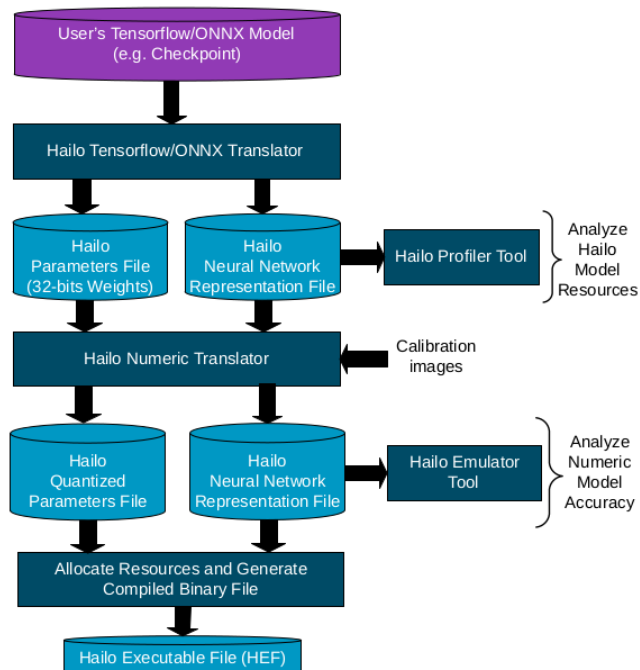
- SDK Hailo được viết trên nền tảng Python
- Bao gồm thư viện HailoRT để chạy inference từ C/C++ hoặc Oython

I.1 Sơ đồ khối



- Model gốc dưới dạng tf hoặc ONNX
- Convert qua dạng binary(HEF) sử dụng Hailo Dataflow Compiler
- Deploy model qua HailoRT (tầng thấp) và TAPPAS (tầng cao) hoặc qua ứng dụng người dùng tự phát triển.

I.2 Quá trình biên dịch model sang binary



1. Dịch model từ dạng tf/onnx sang dạng .hn của hailo (có thể biểu diễn model dưới dạng json). Weight được lưu dưới dạng numpy npz.
2. Profiler tính toán và phân tài nguyên tùy thuộc vào FPS mong muốn (đưa model vào và đưa FPS mong muốn → hailo phân tài nguyên).
3. Emulator : Mô phỏng phục vụ mục đích test
4. Quantize từ float32 sang int 8.
5. Compile model xuống dạng binary.
6. Deploy dùng thư viện HailoRT qua API C/C++ hoặc Python.

I.3 Một số lệnh trong SDK

I.3.1 Parsing model

- Dịch model từ TF sang HN:

```
[ ]: %matplotlib inline
from IPython.display import SVG
from hailo_sdk_client import ClientRunner, NNFramework
from hailo_sdk_common.preprocessing import Normalization
```

Choose the checkpoint files to be used throughout the example:

```
[ ]: model_name = 'resnet_v1_18'
      ckpt_path = '../ckpt/resnet_v1_18.ckpt'
```

Page 24 Release 3.11.0 Confidential and Proprietary | Copyright © 2021 – Hailo Technologies Ltd.

HAILO Hailo SDK User Guide

Open Tensorboard visualization to verify the start and end nodes of the graph (first and last nodes that are deployed to hardware):

This command is blocking so interrupt the kernel when you are done.

```
[ ]: # !hailo tb {ckpt_path}
      # press Kernel -> Interrupt to proceed
      start_node = 'resnet_v1_18/conv1/Pad'
      end_node = 'resnet_v1_18/predictions/Softmax'
```

The main API of the SDK that the user interacts with is the ClientRunner class. To parse a new checkpoint, use the translate_tf_model method:

```
[ ]: runner = ClientRunner(hw_arch='hailo8')
      hn, npz = runner.translate_tf_model(ckpt_path, model_name, start_node_name=start_node, end_node_
      ↪names=[end_node])
```

- Load checkpoint, gọi start và end node và convert.
- Ngoài ra còn có archive, visualizer, normalization và profiler (xem trong docs).
- Ví dụ. Có mạng Resnet 18. Ta có thể đưa ra một FPS mong muốn cho profiler (VD: 300 FPS). Profiler sẽ tự cung cấp tài nguyên để đạt mức FPS. Nếu quá mức FPS maximum, hailo sẽ thông báo là tối đa sẽ chỉ chạy được ... FPS .

I.3.2 Quantization

- Chọn file hailo .har đã được archive.
- Chuẩn bị data quantize
- Có thể quantize xuống 4 bit dùng Quantize aware finetuning
- Có thể dùng analyzer để phân tích độ mất mát

I.3.3 Compilation

- Sau đó, chỉ cần compile sang dạng binary

I.3.4 Inference

- Có hướng dẫn chạy inference dùng streaming, sử dụng nhiều input, output.

II . HAILO TAPPAS

- Demo các App chạy trên Hailo. Bao gồm source code và full luồng từ pre-post processing kèm theo model zoo.

- Tổng cộng số model support : 57 (đang hỏi hailo cụ thể những model nào).

- Một số mạng demo :

*) Detection:

- Centernet-Resnet50:

Size input: 512x512

AP: 31.7 mAP COCO

FPS : 30 (giới hạn bởi FPS luồng stream, app demo centernet res 50 ko có option debug max FPS)

- Centernet-Resnet18:

Size input: 512x512

AP: 26.3 mAP COCO

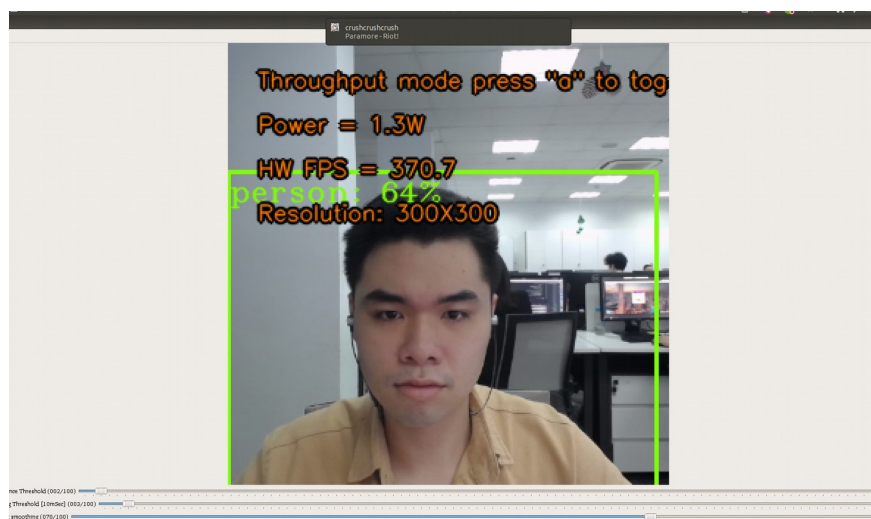
FPS : 113

- Mobilenetv1 SSD:

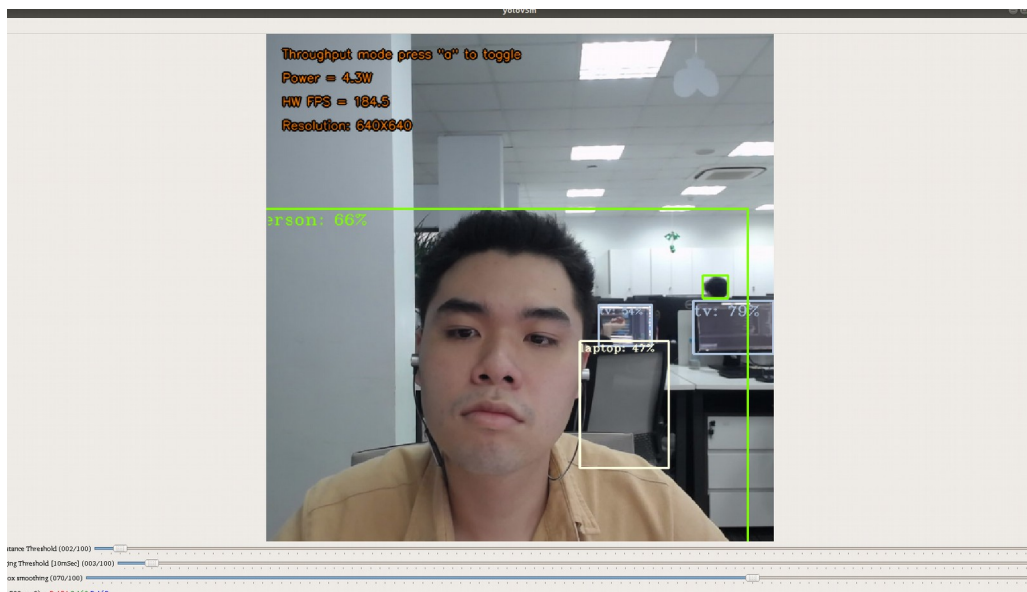
Size input: 300x300x3

AP: 23.1 mAP

FPS : 365



- **YOLOv5m:**
- Size input: 640x640x3
- mAP: 41.7
- FPS: 184



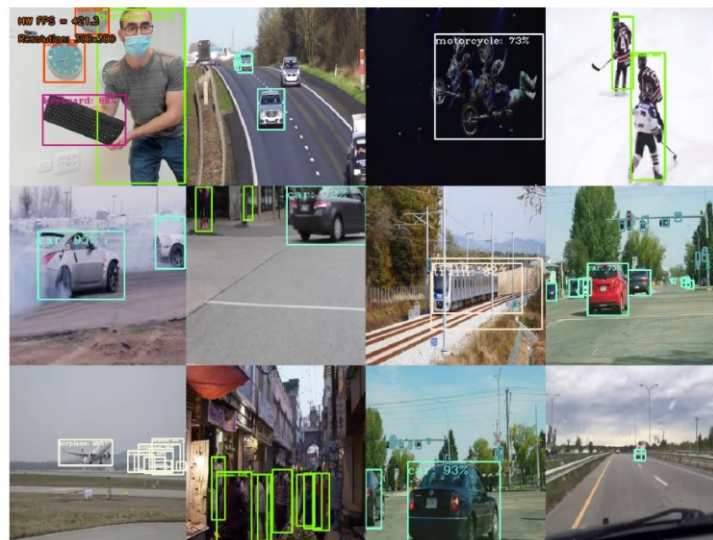
- Note: Yolov5 chạy tốn power hơn các mạng khác
- **YOLOv5s:**

- Size input: 640x640x3
- mAP: 34.35
- FPS: 199

***) Multi stream:**

- Có thể config để chạy nhiều luồng stream một lúc

4.2 Multi Stream



This app shows Hailo's ability to run on multiple streams, leveraging the device's high FPS. It supports multiple networks and varied amount of stream and input sources. You can select number of streams using the `--streams` flag. Inputs are selected as usual using the `--input` flag. However, in this app the `--input` flag can accept multiple inputs. You can select specific devices and files. If the number of streams is larger than the inputs you specify the app will add its default stream.

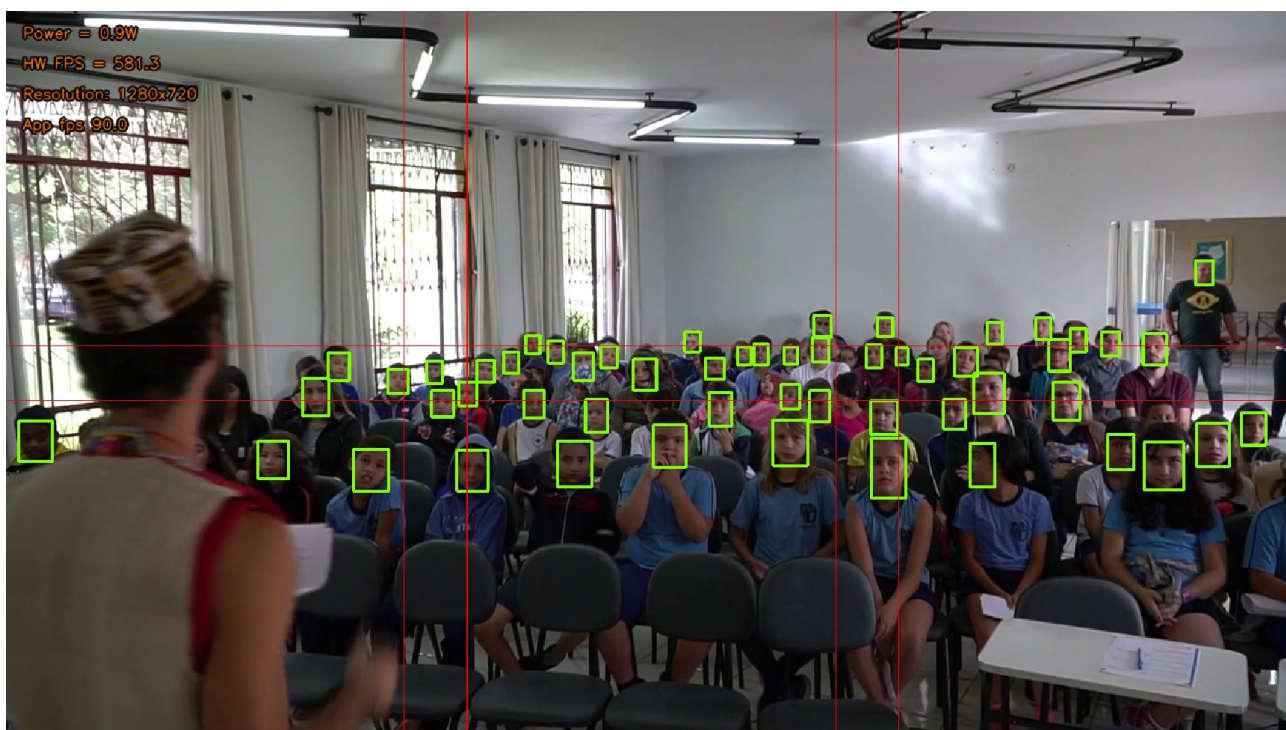
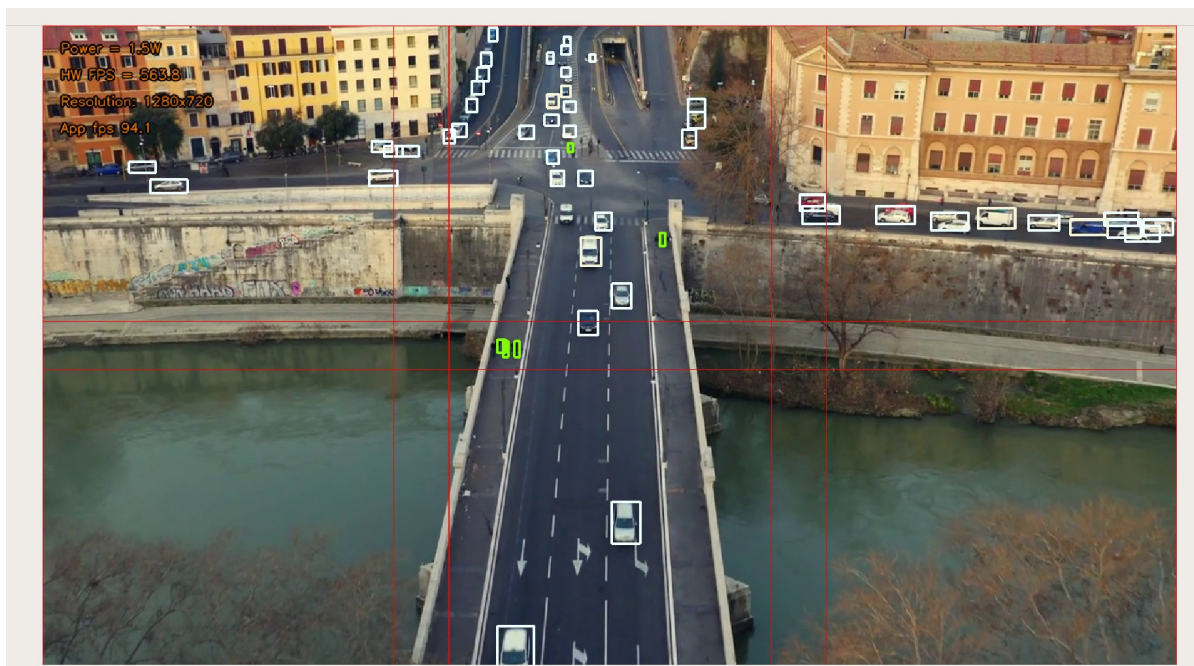
Supported networks mobilenetSSD300, yolov4, yolov3, yolov3_416.

Example commands:

```
hailo run-app multi_stream --network-selection mobilenetSSD300 -i /dev/video0 /dev/video2 some_video.mp4
hailo run-app multi_stream --network-selection yolov3_416 -i /dev/video0 --streams 8
```


***) Tiling:**

- Khi xử lí ảnh lớn và muốn detect object nhỏ → chia thành các tile:



- Có support nhiều chế độ tiling và các model như Retinaface, SSD Mobilenetv1

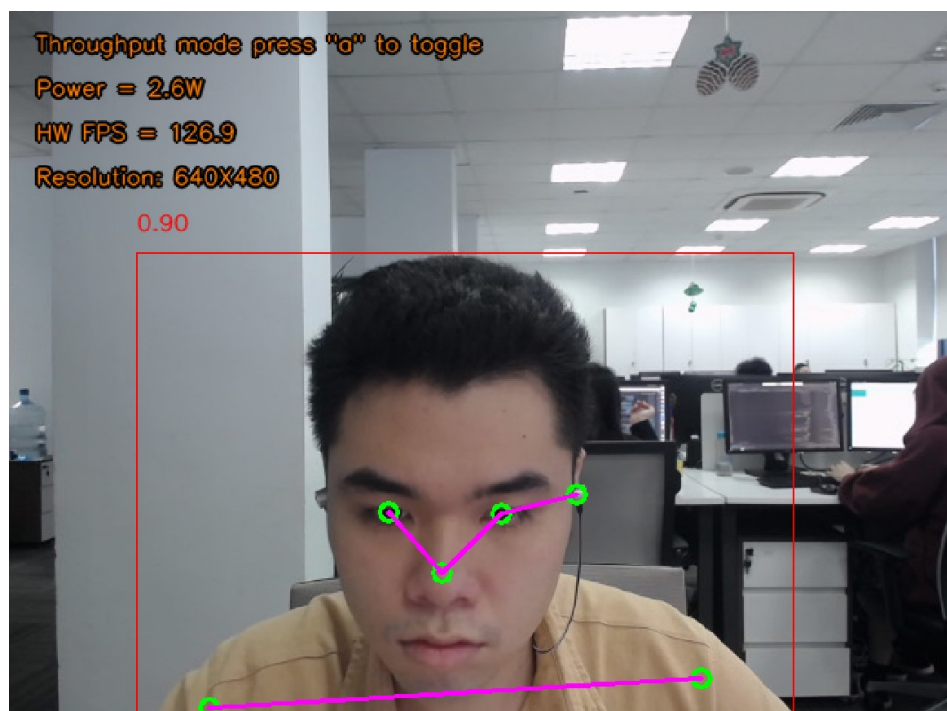
***) Semantic segmentation :**

- Resnet18-FCN8
- Input size: 1920x1024
- Cityscapes 65.18 mIOU



***) Pose estimation :**

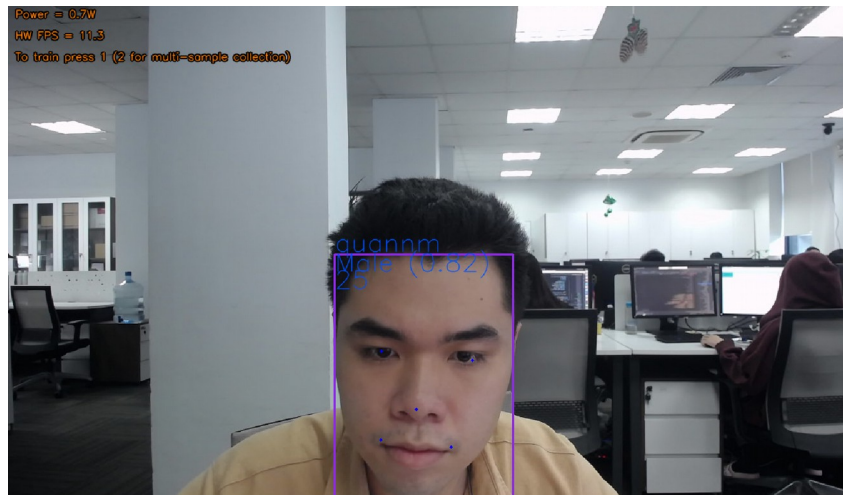
- regnetx_1.6
- 640x640
- 59.4 OKS coco pose



***) Face recognition :**

Input size: 112x112x3:

- Accuracy : 98.5 arcface-MNv1 0.25 , 99.6% arcface rn50 trên LFW



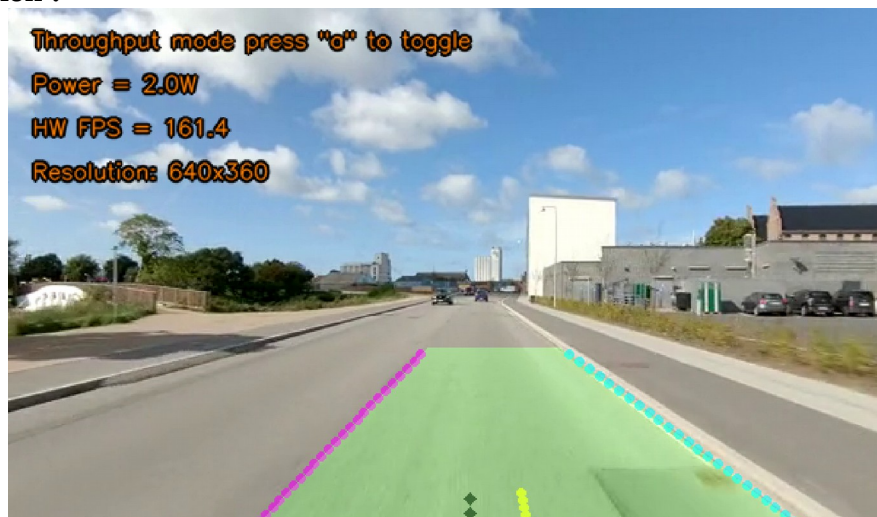
***) Face detection :**

- Retinaface Mobilenet:
- Face 5 landmark detection.
- Input size 1200x736
- 81.2% widerface

- Ngoài ra còn có Ultra-Light-Fast-Generic-Face-Detector-1MB

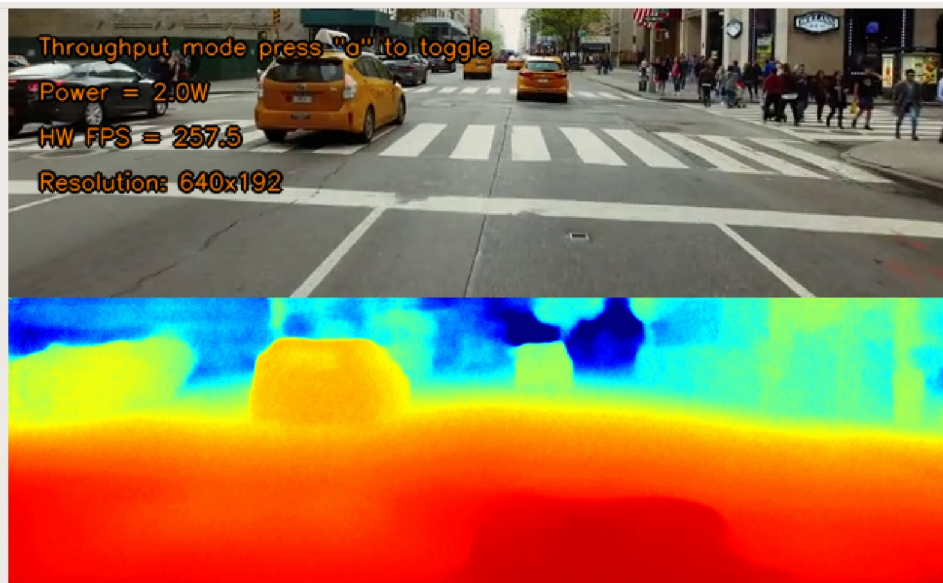
***) Có hỗ trợ instance segmentation**

***) Lane detection :**



- input 640*360*3
- 91.31 TuSimple

***) Depth Estimation**



-Input 640*192*3

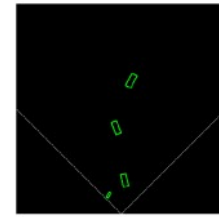
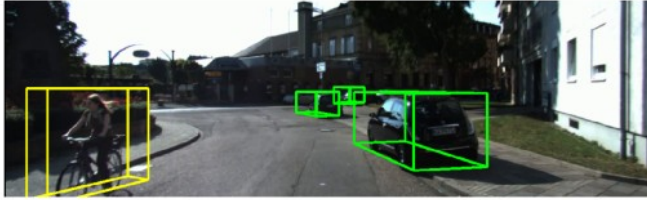
***) Multiple Object Tracking**







- Input: 460x864x3
- 70.4 MOT-16
- Dựa theo FairMOT với backbone RegnetX-800MF

***) 3D Object detection .**

Resolution: 2048x1536



```
$> hailo run-app smoke_regnet800
```

	Resolution	1280x384x3
	Native accuracy	car BEV moderate mAP: 14.14%
	Dataset	kitti 3d object detection
	Model	Based on SMOKE: Single-Stage Monocular 3D Object Detection via Key-point Estimation . Downloaded from SMOKE github

***) Multitask**



- Chạy song song detection và depth estimation

***) Multitask**

- Có hỗ trợ Gstream nhưng em k hiểu lắm

***) Tự viết app**

- Có thể tự viết app dựa theo SDK