



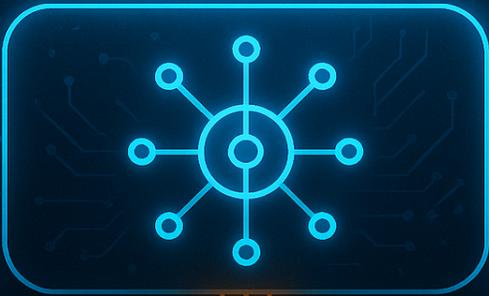
ACCELERATING EDGE AI WITH QUALCOMM AI HUB

— CVPR 2025 Tutorial —

The IEEE/CVF Conference on Computer Vision and Pattern Recognition 2025

Nashville, TN, USA

TUTORIAL AGENDA



- 1 Introduction to Qualcomm AI Hub
- 2 API Token and Python Environment
- 3 Model Compilation
- 4 Performance Profiling
- 5 Model Inference

QUALCOMM AI HUB

Overview

Qualcomm AI Hub is a developer-centric platform that streamlines the deployment of on-device AI for Snapdragon-powered hardware. It enables seamless workflows—from model import and optimization to profiling and deployment.

Model Conversion

Transform trained models (e.g., PyTorch, ONNX) for optimal on-device performance.

Validation

Verify numerical correctness by comparing on-device inference outputs against reference model outputs to ensure fidelity.

Performance Profiling

Get comprehensive on-device metrics including runtime, load time, and compute unit utilization.

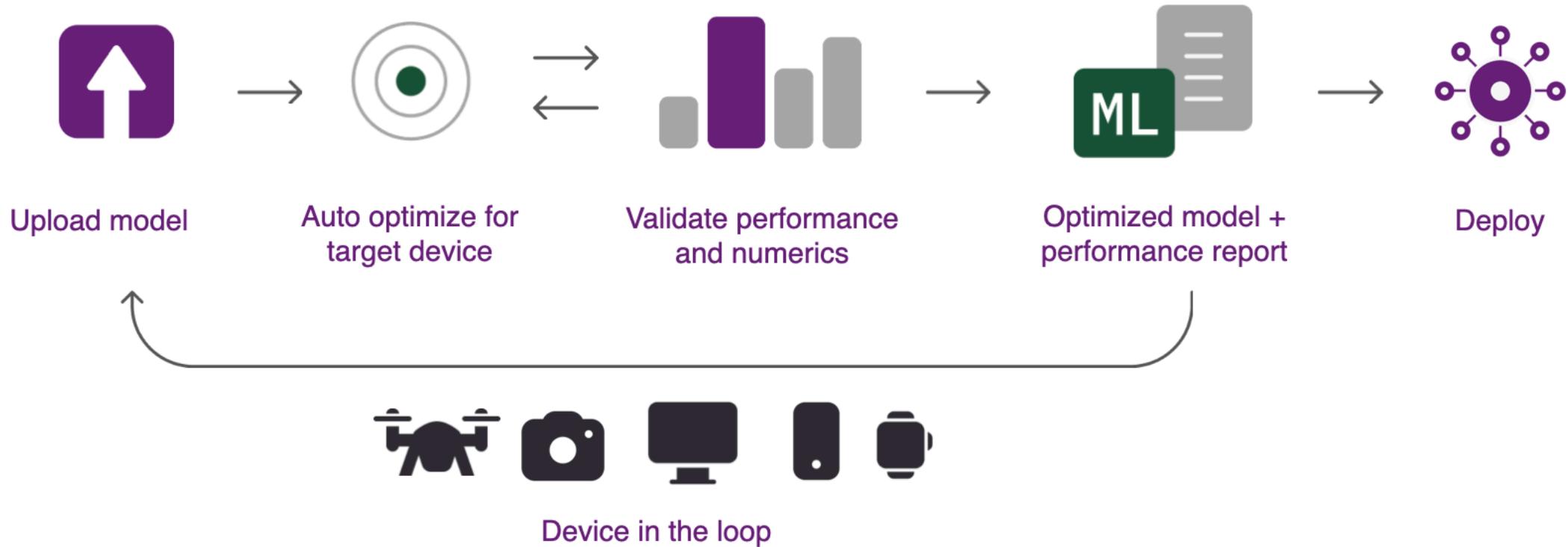
Flexible Deployment

Receive device-ready artifacts (e.g., DLC files and runtime config) and sample apps for easy integration into your Edge AI projects.



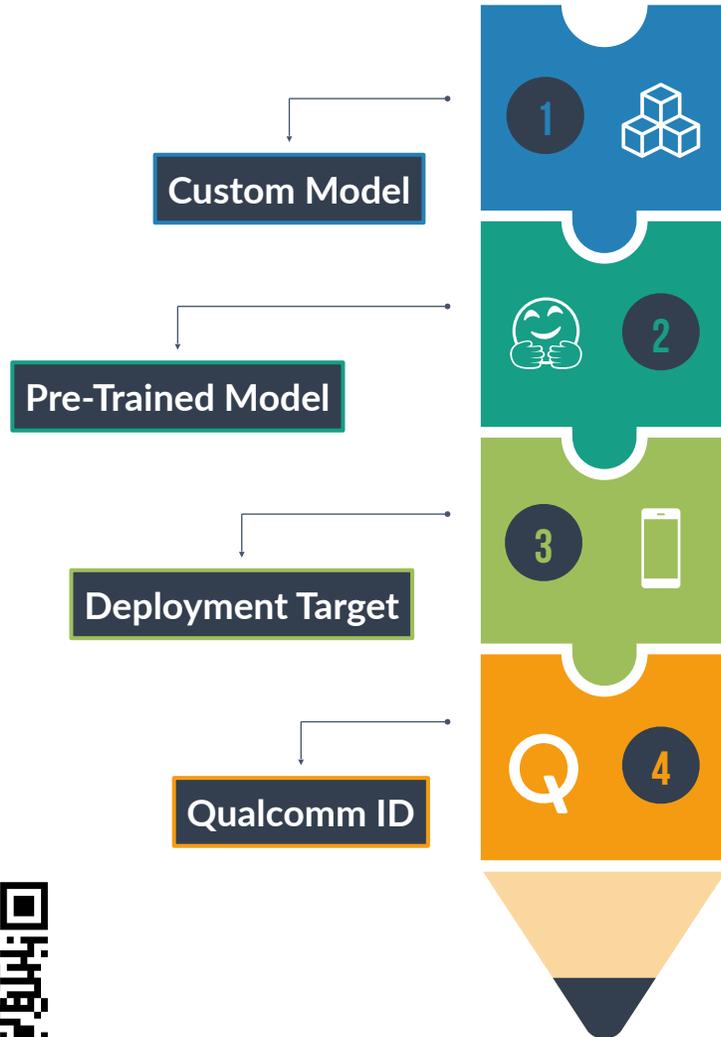
QUALCOMM AI HUB

How does it work?



QUALCOMM AI HUB

What do you need?



01

Custom Model

A trained model that can be in Pytorch, TFLite, or ONNX format.

02

Pre-Trained Model

Qualcomm also has several models available on GitHub and Hugging Face.

03

Deployment Target

This can be a specific device (FairPhone 5, Pixel 6) or a range of devices.

04

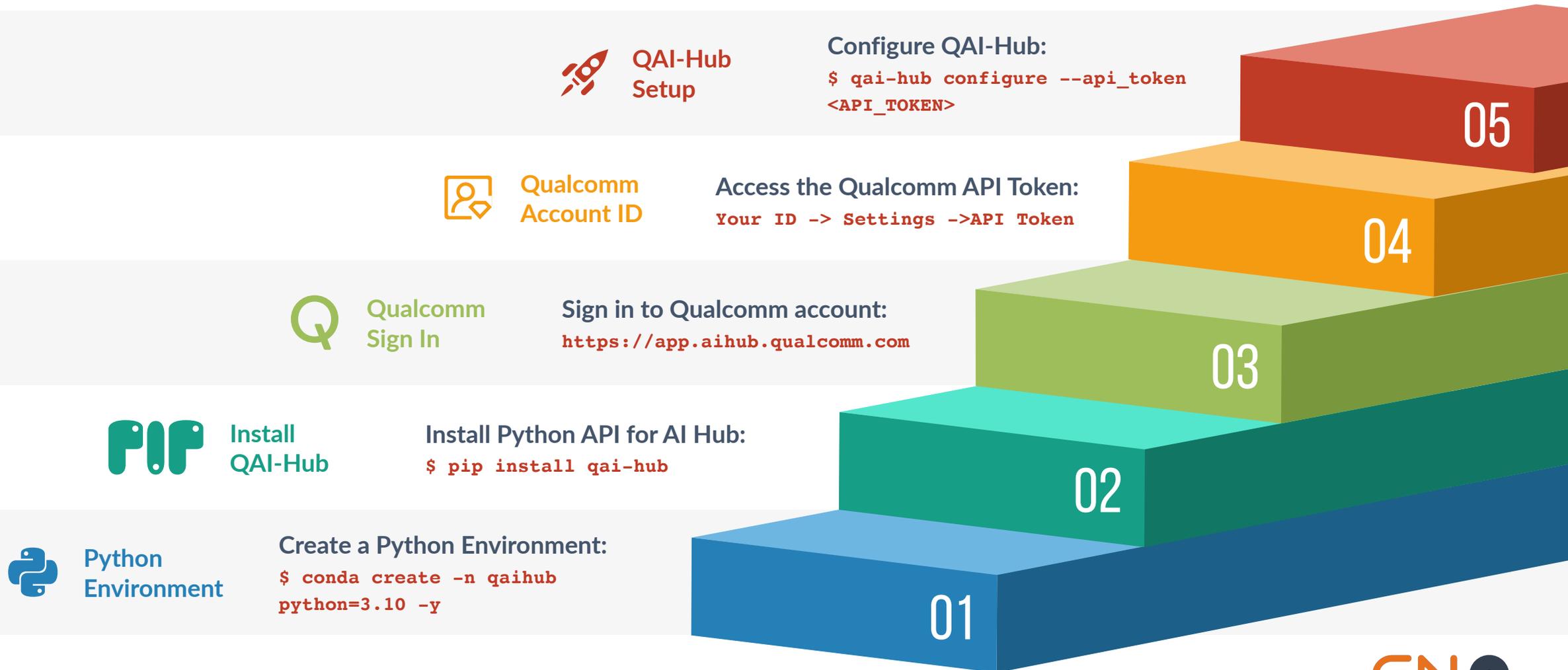
Qualcomm ID

An account on Qualcomm AI HUB.



QUALCOMM AI HUB

Installation



CHECK AVAILABLE DEVICES

List of Devices

“ Choose based on device type: Automotive, IOT, XR, Windows, or Mobile. ”

CLI

Type this command on your Terminal:

```
$ qai-hub list-devices
```



WEB

Access the following website on your browser:

<https://app.aihub.qualcomm.com/devices>



CHECK AVAILABLE DEVICES

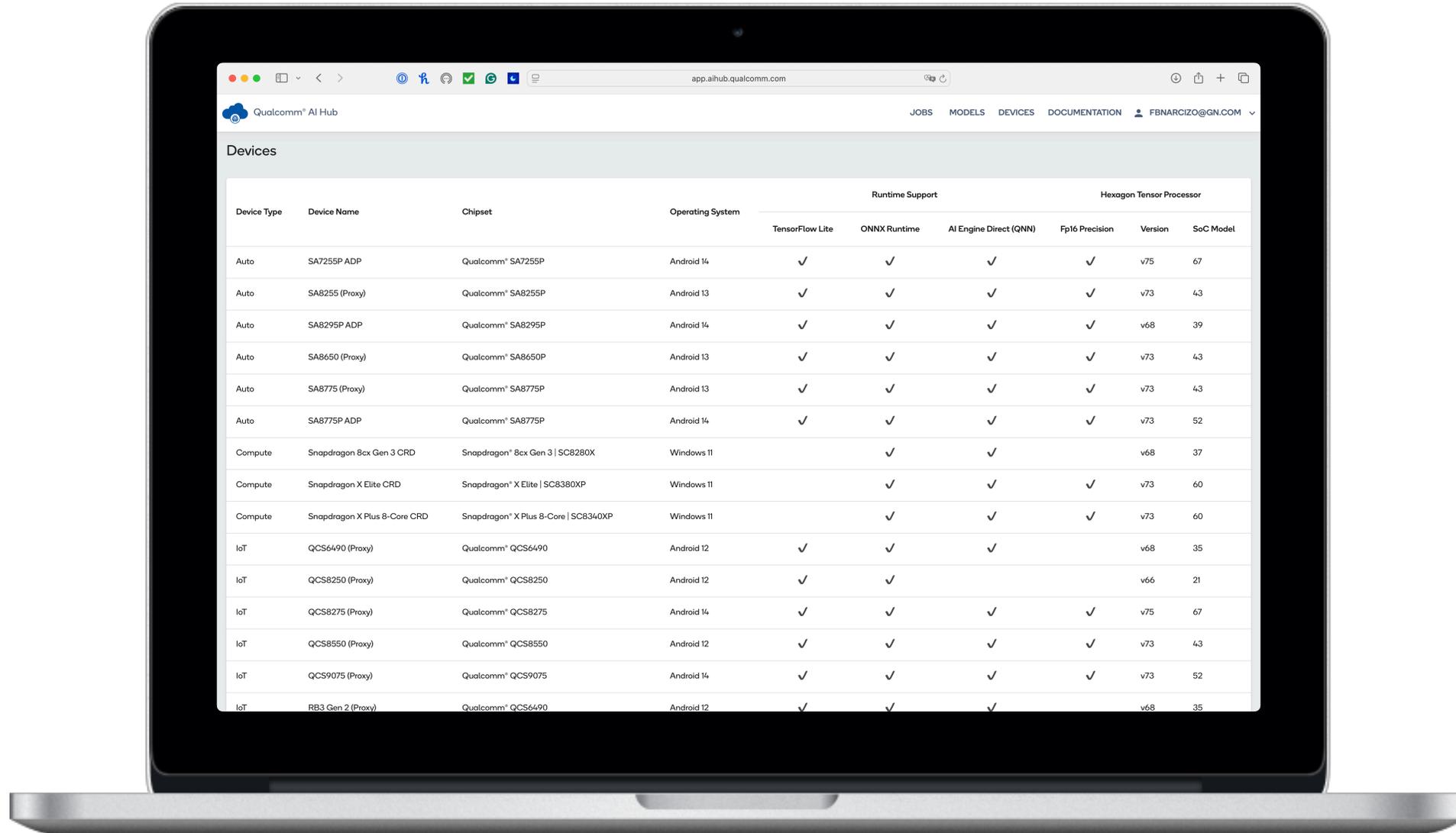
Terminal

```
fabricio@Narcizos-MacBook-Pro:~/Qualcomm
~/Qualcomm 3.10.18 (qaihub)
qai-hub list-devices
```

Device	OS	Vendor	Type	Chipset
Google Pixel 3 (Family)	Android 10	Google	Phone	qualcomm-snapdragon-845, sdm845
Google Pixel 3	Android 10	Google	Phone	qualcomm-snapdragon-845, sdm845
Google Pixel 3a	Android 10	Google	Phone	qualcomm-snapdragon-670, sdm670
Google Pixel 3 XL	Android 10	Google	Phone	qualcomm-snapdragon-845, sdm845
Google Pixel 4	Android 10	Google	Phone	qualcomm-snapdragon-855, sm8150
Google Pixel 4a	Android 11	Google	Phone	qualcomm-snapdragon-730g, sm7150-ab
Google Pixel 5	Android 11	Google	Phone	qualcomm-snapdragon-765g, sm7250
Samsung Galaxy Tab S7	Android 11	Samsung	Tablet	qualcomm-snapdragon-865+, sm8250-ab
Samsung Galaxy Tab A8 (2021)	Android 11	Samsung	Tablet	qualcomm-snapdragon-429, sdm429
Samsung Galaxy Note 20 (Intl)	Android 11	Samsung	Phone	samsung-exynos-990
Samsung Galaxy S21 (Family)	Android 11	Samsung	Phone	qualcomm-snapdragon-888, sm8350
Samsung Galaxy S21	Android 11	Samsung	Phone	qualcomm-snapdragon-888, sm8350

CHECK AVAILABLE DEVICES

Qualcomm AI Hub



WHY IS IT CALLED QCS6490 (PROXY)?

Important Information



Information

This job targets a proxy device, which is intended to mimic the characteristics of a real device. Profiling results may differ from real devices due to differences in operating system, firmware, clock speed, thermal packaging, and other factors.

CHANGES SPECIFIC FOR OUR APP

InferSNPE Android Application



Data Layer Format

When using SNPE, the model input layers changed from NCHW to NHWC.



Input Format

InferSNPE App requires the input format as NHWC (1, 320, 320, 3).



Output Names

output_bboxes: Bounding box coordinates.
output_classes: Class prediction scores.



Layer Names

The InferSNPE App looks for these specific names, not generic **output_0** or **output_1**.

HOW TO CHANGE THE INPUT FORMAT

https://github.com/fabricionarcizo/snpe_optimizer/blob/main/notebooks/qai_hub.ipynb

```
!pip install onnx-graphsurgeon
!pip install scc4onnx
!scc4onnx -if ./assets/models/yolo_nas_s.onnx \
  -of ./assets/models/yolo_nas_s_nhwc.onnx \
  --input_op_names_and_order_dims input "[0,2,3,1]"
```



Qualcomm

PROFILE JOB

INFERENCE
LATENCY



MEMORY USAGE

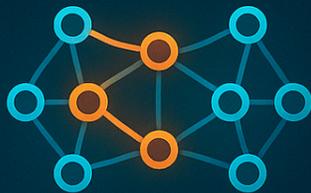


LOAD TIME



CPU
GPU AI DSP

RUNNING MODEL



PROFILE JOB

Measure Performance

The Profile Job feature in Qualcomm AI Hub enables detailed performance benchmarking of AI models on actual Snapdragon hardware. It provides insights into how efficiently a model runs, revealing critical deployment metrics.



Inference Latency

Measures how long it takes for the model to produce outputs once inputs are received—critical for real-time applications.



Load Time

Reports how long it takes to initialize and load the model into memory, including compilation overhead if applicable.



Memory Footprint

Indicates the total memory consumed during inference, helping identify models too large for constrained devices.



Compute Breakdown

Displays the distribution of processing workload across CPU, GPU, and DSP, enabling better runtime allocation and performance tuning.

PROFILE ORIGINAL ONNX MODEL

https://github.com/fabricionarcizo/snpe_optimizer/blob/main/notebooks/qai_hub.ipynb

```
def profile_compiled_model(compile_job):  
    """Profile the compiled quantized model performance."""  
    print(f"🇮🇹 Starting model profiling on {TARGET_DEVICE}...")  
  
    target_model = compile_job.get_target_model()  
    profile_job = hub.submit_profile_job(  
        model=target_model,  
        device=hub.Device(TARGET_DEVICE)  
    )  
  
    print(f"⌚ Profile job submitted: {profile_job.job_id}")  
    profile_job.wait()  
  
    status = profile_job.get_status()  
    success = status.code == "SUCCESS" \  
        if hasattr(status, 'code') else str(status).upper() == "SUCCESS"  
  
    if success:  
        print("✅ Profiling completed successfully!")  
    else:  
        print(f"❌ Profiling failed: {status}")
```

IMPORTANT



PROFILE ORIGINAL ONNX MODEL

Original YOLO-NAS S Model Inference

The screenshot shows the Qualcomm AI Hub interface for a job titled 'Profile Job Results'. The job ID is 'jpv082zr5' and the status is 'Results Ready'. The page is divided into several sections: Information, Inference Metrics, and Detailed Metrics.

Information

- Name:** yolo_nas_s_nhwc.onnx
- Target Device:** QCS6490 (Proxy), Android 12, Qualcomm® QCS6490
- Creator:** shaahmed@gnhearing.com
- Target Model:** yolo_nas_s_nhwc.onnx (mmyd08wq)
- Input Specs:** input : float32[1, 320, 320, 3]
- Submission / Completion Time:** 6/7/2025, 4:49:35 PM / 6/7/2025, 4:52:02 PM
- Versions:** ONNX Runtime : 1.21.1, Android : 12 (SPIA.210812.016), AI Hub : aihub-2025.05.30.0
- Information:** This job targets a proxy device, which is intended to mimic the characteristics of a real device. Profiling results may differ from real devices due to differences in operating system, firmware, clock speed, thermal packaging, and other factors.

Inference Metrics

- Minimum Inference Time:** 83.8 ms
- Estimated Peak Memory Usage:** 18 - 32 MB
- Compute Units:** CPU 187

Detailed Metrics

Stage	Time	Memory
Compilation	0.0 ms	0.0 MB

CALIBRATION DATA

Overview

WHAT IS IT?

Representative input samples used during model quantization to preserve accuracy when converting from **float32** → **int8**

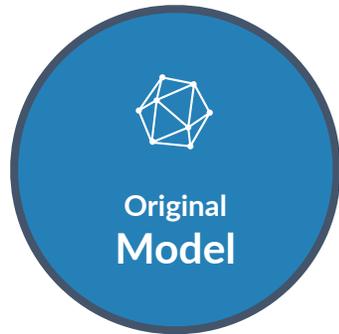
REASONS

Edge devices need **8-bit models** for speed and efficiency. Naive quantization **can destroy model accuracy**. We need to understand the typical value ranges in each model layer.



COMPILE JOB

Transform and Optimize



Input model from PyTorch, ONNX, and TensorFlow in FP32.



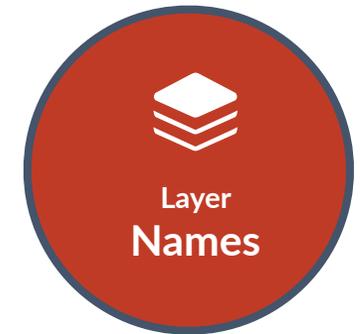
Convert the original model to Qualcomm format.



Quantize the Qualcomm model from FP32 to INT8.



Quantize the input and output layers from the INT8 model.



Directly give names for outputs required for InferSNPE App.

COMPILE JOB CODE

https://github.com/fabricionarcizo/snpe_optimizer/blob/main/notebooks/qai_hub.ipynb

```
# Build compilation options for Android app compatibility.
compile_options = [
    "--target_runtime qnn_dlc",           # Qualcomm runtime.
    "--quantize_full_type int8",        # 8-bit quantization.
    "--quantize_io",                   # Quantize I/O.
    "--output_names output_bboxes,output_classes" # Custom output names.
]
options_str = " ".join(compile_options)

# Input specification (NHWC format as required by app).
input_specs = {"input": (1, 320, 320, 3)}
print(f"\n📄 Compilation Configuration:")
print(f"  Input specs: {input_specs}")
print(f"  Options: {options_str}")
print(f"  Calibration samples: {len(calibration_data['input'])}")

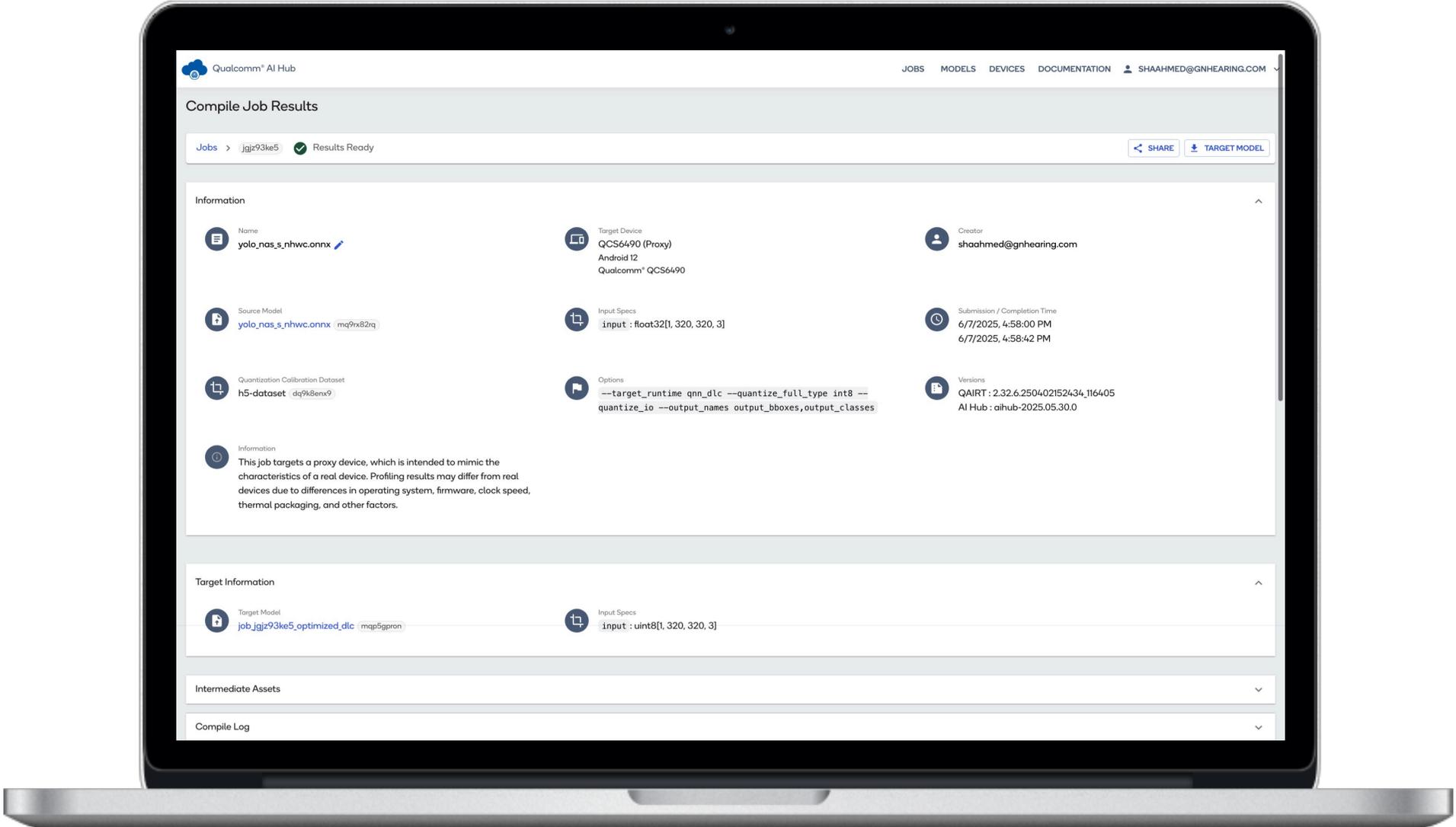
compile_job = hub.submit_compile_job(
    model=model_path,
    device=hub.Device(TARGET_DEVICE),
    input_specs=input_specs,
    options=options_str,
    calibration_data=calibration_data
)
```

IMPORTANT



COMPILE JOB

YOLO-NAS S Model



PROFILE AFTER COMPILATION

YOLO-NAS S Model

The screenshot shows the Qualcomm AI Hub interface for a profile job. The page is titled "Profile Job Results" and shows a job named "jpee64vp" with a status of "Results Ready". The job information includes the name "job_jgiz93ke5_optimized_dlc", target device "QCS6490 (Proxy)", and creator "shaahmed@gnhearing.com". The target model is "job_jgiz93ke5_optimized_dlc" with a map icon. The input specs are "input : uint8[1, 320, 320, 3]". The submission and completion times are "6/7/2025, 5:01:19 PM" and "6/7/2025, 5:03:02 PM" respectively. The versions listed are QAIRT: v2.32.6.250402152434_116405, QNN Backend API: 5.32.0, QNN Core API: 2.24.0, Android: 12 (SPIA.210812.016), and AI Hub: aihub-2025.05.30.0. The inference metrics show a minimum inference time of 3.8 ms, estimated peak memory usage of 0 - 30 MB, and 289 compute units (NPU). The detailed metrics table shows a compilation stage with 0.0 ms time and 0.0 MB memory.

Qualcomm AI Hub

JOB MODELS DEVICES DOCUMENTATION SHAAHMED@GNHEARING.COM

Profile Job Results

Jobs > jpee64vp Results Ready SHARE

Information

Name
job_jgiz93ke5_optimized_dlc

Target Device
QCS6490 (Proxy)
Android 12
Qualcomm QCS6490

Creator
shaahmed@gnhearing.com

Target Model
job_jgiz93ke5_optimized_dlc map

Input Specs
input : uint8[1, 320, 320, 3]

Submission / Completion Time
6/7/2025, 5:01:19 PM
6/7/2025, 5:03:02 PM

Versions
QAIRT : v2.32.6.250402152434_116405
QNN Backend API : 5.32.0
QNN Core API : 2.24.0
Android : 12 (SPIA.210812.016)
AI Hub : aihub-2025.05.30.0

Information
This job targets a proxy device, which is intended to mimic the characteristics of a real device. Profiling results may differ from real devices due to differences in operating system, firmware, clock speed, thermal packaging, and other factors.

Inference Metrics

Minimum Inference Time ?
3.8 ms

Estimated Peak Memory Usage ?
0 - 30 MB

Compute Units ?
NPU 289

Detailed Metrics

Stage	Time	Memory
Compilation ?	0.0 ms	0.0 MB

RUN INFERENCE

https://github.com/fabricionarcizo/snpe_optimizer/blob/main/notebooks/qai_hub.ipynb

```
def simple_inference_pipeline(
    image_path: str, compile_job, show_inline: bool = True):
    print(f"🚀 Simple YOLO inference on: {os.path.basename(image_path)}")
    print(f"    Format: int8, NHWC")
    print(f"    Output: Top 10 detections only")

    # 1. Preprocess image.
    input_data, original_image, scale_x, scale_y = \
        simple_preprocess(image_path)

    # 2. Run inference.
    print("🔄 Running inference...")
    target_model = compile_job.get_target_model()

    inference_job = hub.submit_inference_job(
        model=target_model,
        device=hub.Device(TARGET_DEVICE),
        inputs={"input": [input_data]}
    )

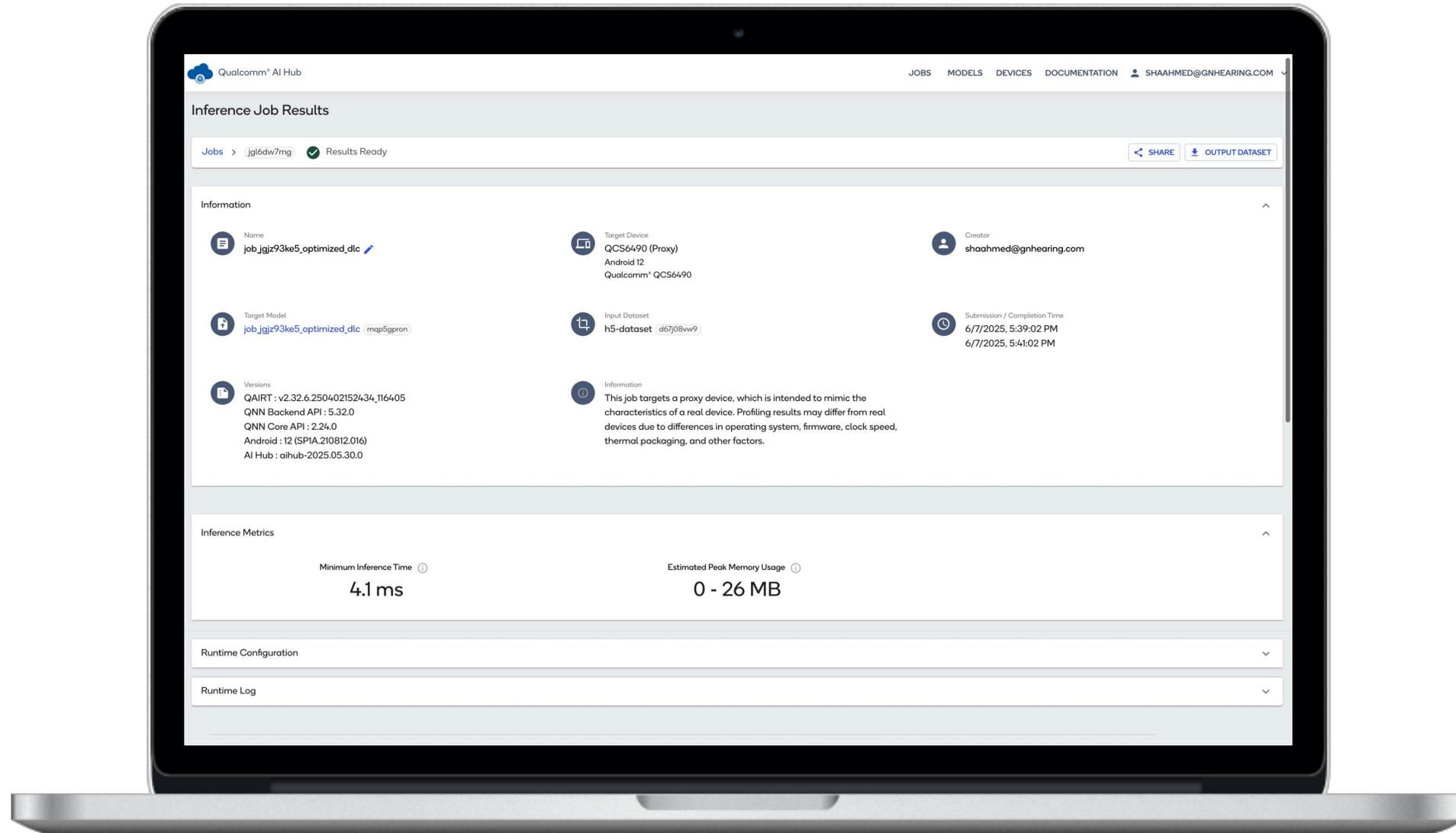
    inference_job.wait()
    results = inference_job.download_output_data()
    ...
```

IMPORTANT



INFERENCE RESULTS

Quantized YOLO-NAS S Model



INFERENCE OUTPUT IMAGE

Example



DOWNLOAD QUANTIZED MODEL

https://github.com/fabricionarcizo/snpe_optimizer/blob/main/notebooks/qai_hub.ipynb

```
# Download and save to your specified path
target_model = compile_job.get_target_model()
model_path = 'assets/models/yolo_nas_s_int8.dlc'
target_model.download(model_path)

print(f"✅ Model downloaded and saved to: {model_path}")
```



PROFILE ONNX HAGRID

Original YOLO-hagRID Model Inference

The screenshot displays the Qualcomm AI Hub interface for a profile job. The page is titled "Profile Job Results" and shows the job ID "jp3vdo8ng" with a status of "Results Ready".

Information

- Name:** yolo_hagRID_nhwc.onnx
- Target Device:** QCS6490 (Proxy), Android 12, Qualcomm® QCS6490
- Creator:** shaahmed@gnhearing.com
- Target Model:** yolo_hagRID_nhwc.onnx (mqkdl7yxm)
- Input Specs:** input : float32[1, 640, 640, 3]
- Submission / Completion Time:** 6/7/2025, 5:48:20 PM to 6/7/2025, 5:50:32 PM
- Versions:** ONNX Runtime : 1.211, Android : 12 (SPIA.210812.016), AI Hub : aihub-2025.05.30.0
- Information:** This job targets a proxy device, which is intended to mimic the characteristics of a real device. Profiling results may differ from real devices due to differences in operating system, firmware, clock speed, thermal packaging, and other factors.

Inference Metrics

- Minimum Inference Time:** 103.3 ms
- Estimated Peak Memory Usage:** 51 - 67 MB
- Compute Units:** CPU 240

Detailed Metrics

Stage	Time	Memory
Compilation	0.0 ms	0.0 MB

PROFILE ONNX HAGRID

Quantized YOLO-hagRID Model Inference

The screenshot displays the 'Profile Job Results' page on the Qualcomm AI Hub. The page is titled 'Profile Job Results' and shows a job with ID 'jpv0824r5' that is 'Results Ready'. The job name is 'job_jgokxdkmp_optimized_dlc'. The target device is 'QCS6490 (Proxy)' with Android 12 and Qualcomm QCS6490. The creator is 'shaahmed@gnhearing.com'. The target model is 'job_jgokxdkmp_optimized_dlc' with version 'mq3kv073m'. The input specs are 'input : uint8[1, 640, 640, 3]'. The submission and completion times are '6/7/2025, 6:05:58 PM' and '6/7/2025, 6:08:32 PM' respectively. The versions listed are QAIRT: v2.32.6.250402152434_116405, QNN Backend API: 5.32.0, QNN Core API: 2.24.0, Android: 12 (SPIA.210812.016), and AI Hub: aihub-2025.05.30.0. The inference metrics show a minimum inference time of 5.6 ms, estimated peak memory usage of 1 - 29 MB, and 331 NPU compute units. The detailed metrics table shows a compilation stage with 0.0 ms time and 0.0 MB memory.

Qualcomm AI Hub

JOB MODELS DEVICES DOCUMENTATION SHAAHMED@GNHEARING.COM

Profile Job Results

Jobs > jpv0824r5 Results Ready SHARE

Information

- Name**: job_jgokxdkmp_optimized_dlc
- Target Device**: QCS6490 (Proxy)
Android 12
Qualcomm QCS6490
- Creator**: shaahmed@gnhearing.com
- Target Model**: job_jgokxdkmp_optimized_dlc (mq3kv073m)
- Input Specs**: input : uint8[1, 640, 640, 3]
- Submission / Completion Time**: 6/7/2025, 6:05:58 PM
6/7/2025, 6:08:32 PM
- Versions**: QAIRT : v2.32.6.250402152434_116405
QNN Backend API : 5.32.0
QNN Core API : 2.24.0
Android : 12 (SPIA.210812.016)
AI Hub : aihub-2025.05.30.0
- Information**: This job targets a proxy device, which is intended to mimic the characteristics of a real device. Profiling results may differ from real devices due to differences in operating system, firmware, clock speed, thermal packaging, and other factors.

Inference Metrics

- Minimum Inference Time**: 5.6 ms
- Estimated Peak Memory Usage**: 1 - 29 MB
- Compute Units**: NPU 331

Detailed Metrics

Stage	Time	Memory
Compilation	0.0 ms	0.0 MB

HOW DOES IT WORK?

Overview



Intelligent Translation

Automatically converts models from source frameworks to device-optimized runtime.



Cloud-Based Validation

Provision real devices in the cloud for accurate performance profiling.



Hardware-Aware Optimization

Applies Qualcomm-specific optimizations for maximum performance.



Physical Testing

Validates both performance metrics and numerical correctness on actual hardware.



QUESTIONS & ANSWERS

T H A N K Y O U !