



PyNetsPresso and LaunchX

: An Integrated Toolchain for HW-Aware AI Model Optimization and Benchmarking

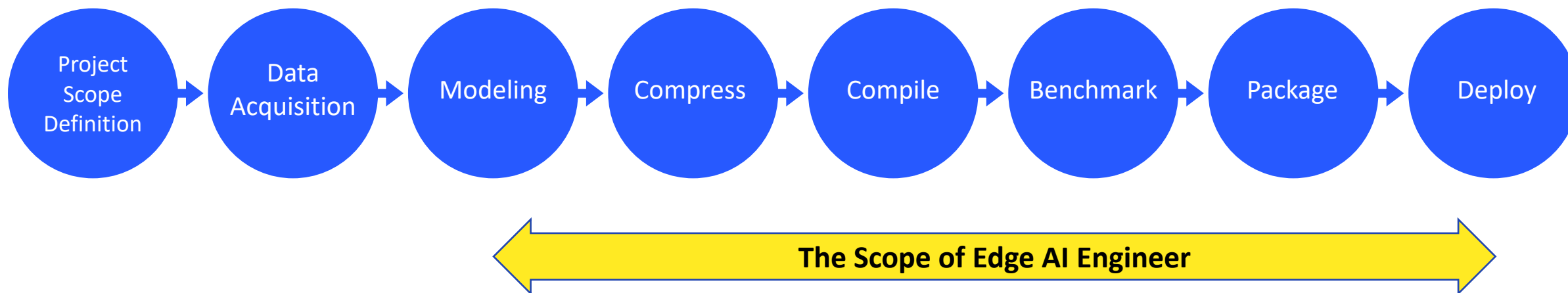


Table of Contents

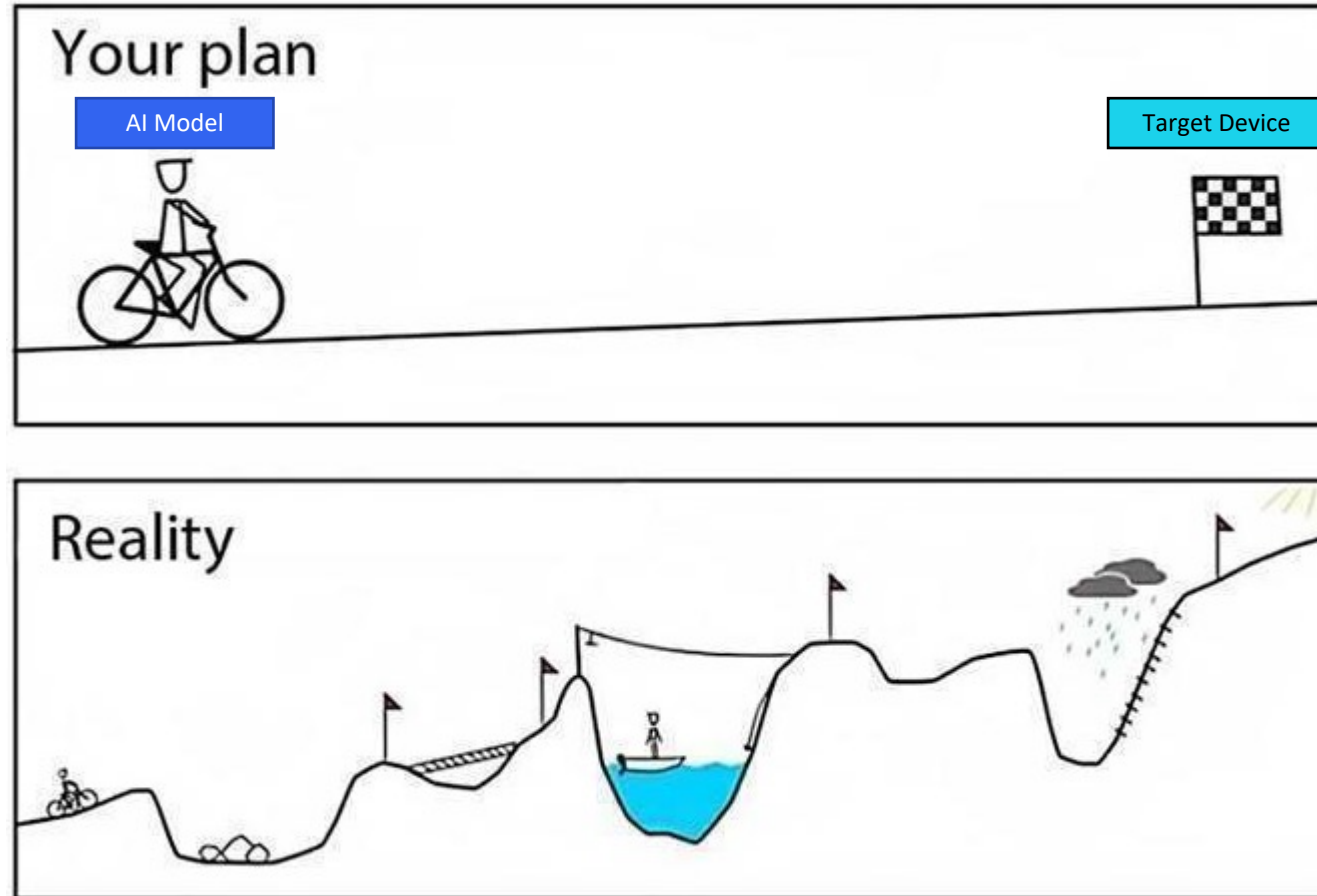
-
1. Introduction
 2. Problems
 3. Solutions
 4. Success Cases
 5. Demo

Introduction: The Journey of Edge AI Engineer

It starts from AI model, to deployment

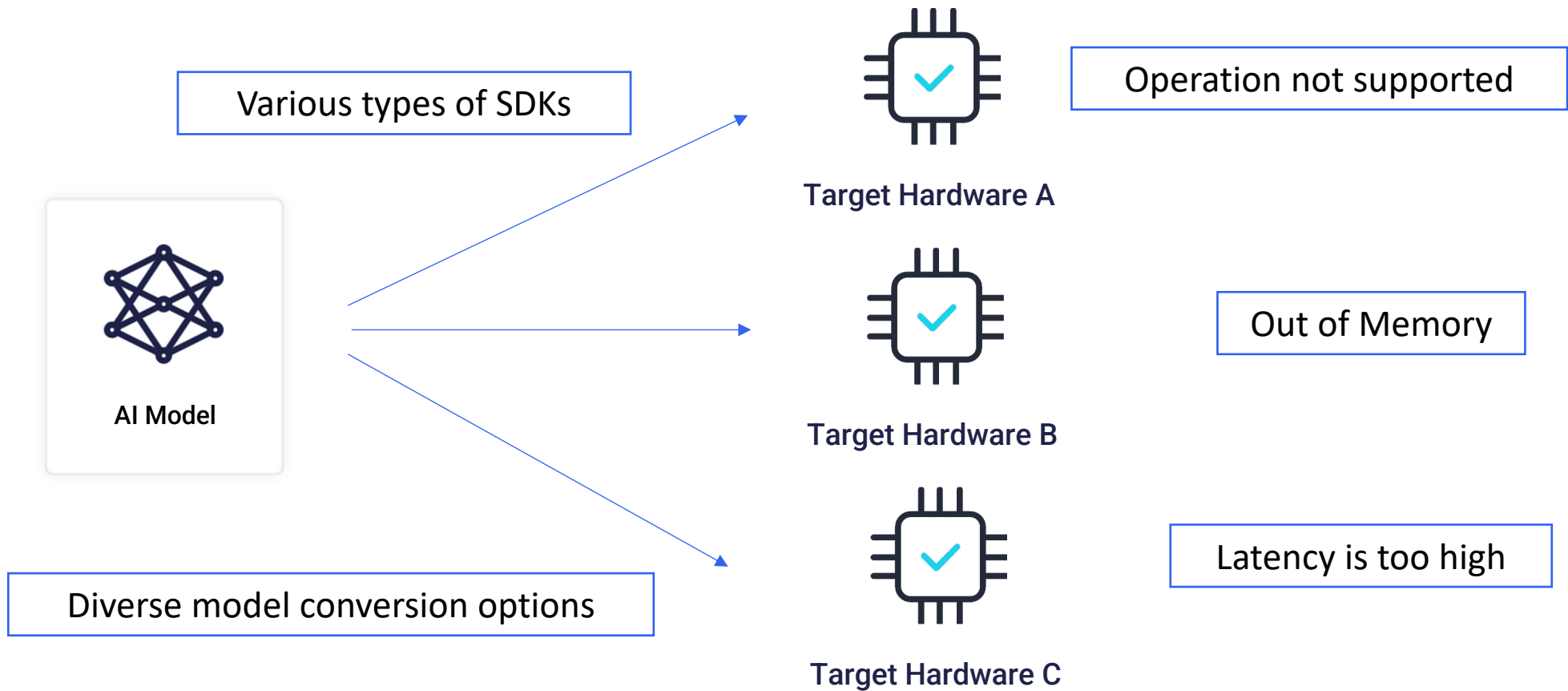


Things Don't Always Go As Planned

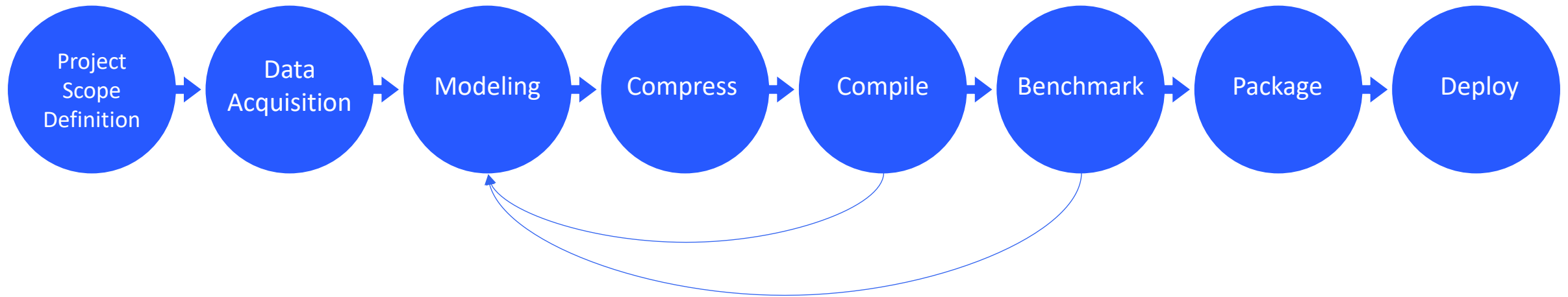


Source: lesswrong.com

Problems: What's happening on the journey?



What's happening on the journey?



If the model does not work, you get back to square one

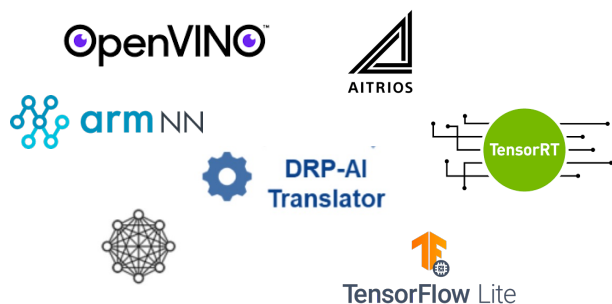
You need to go through the whole thing again to make your model work on the device.

Or, choose better solution

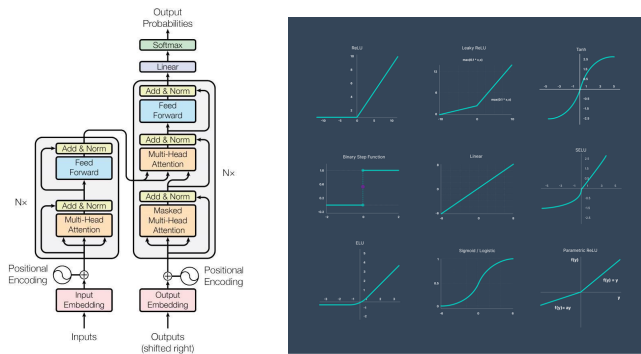
Solution: NetsPresso

The easiest tool for Edge AI engineers

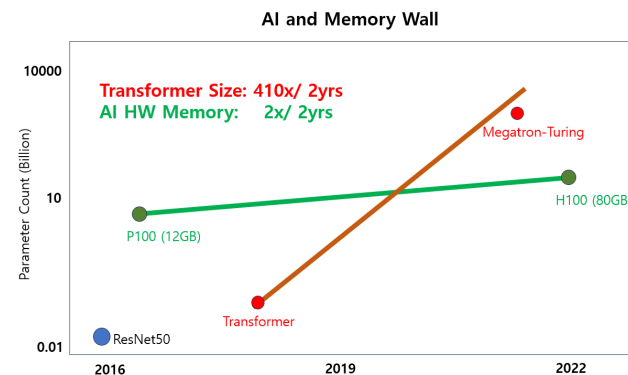
Challenges



Diverse Edge AI SDK



Unsupported layers and operations



Limited memory size
Latency is too high

Solution

NetsPresso

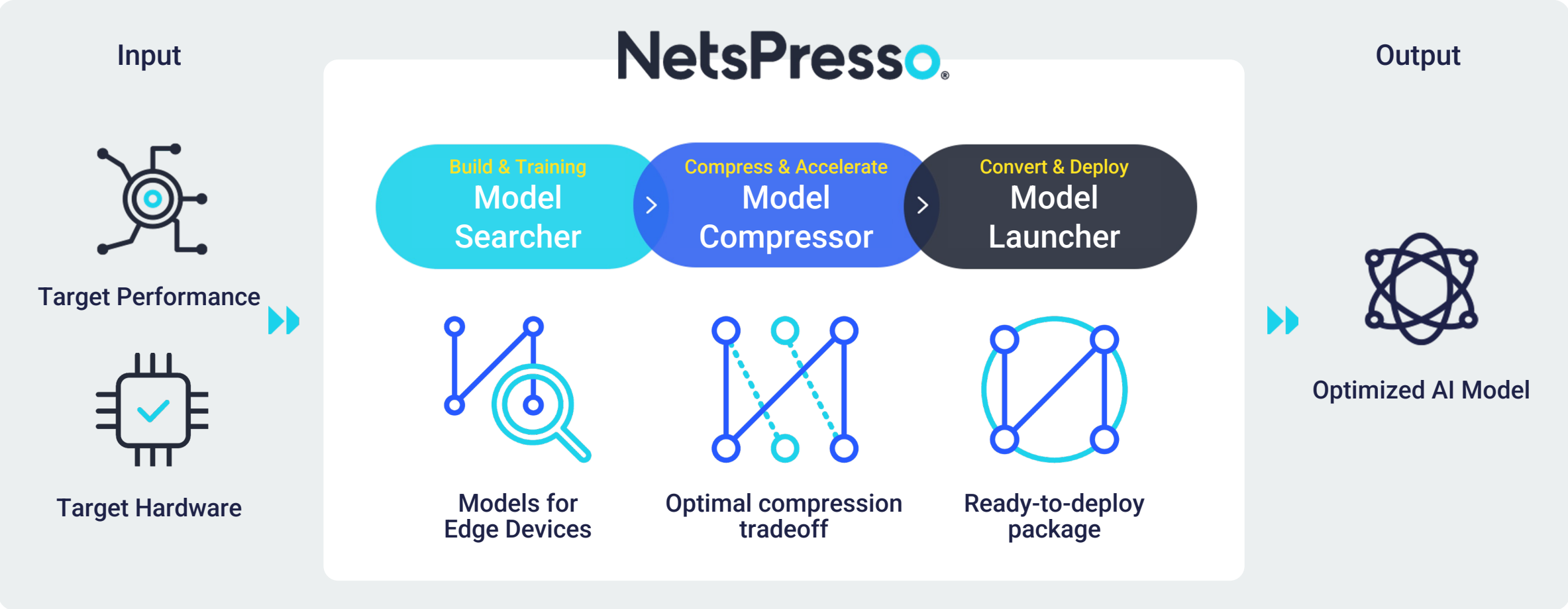
Unified Interface

Operator Conversion

AI Model Compression

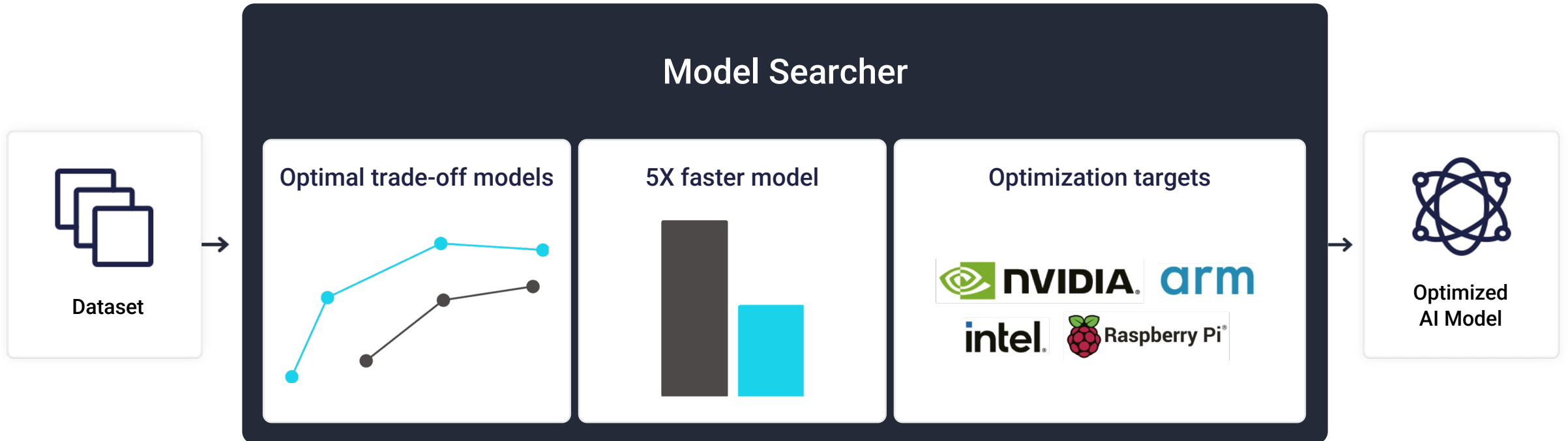
Platform Overview: AI Model Optimization

NetsPresso® simplifies AI model optimization for target devices with automated processes.



Model Searcher Module: Automated Model Search for Your Device

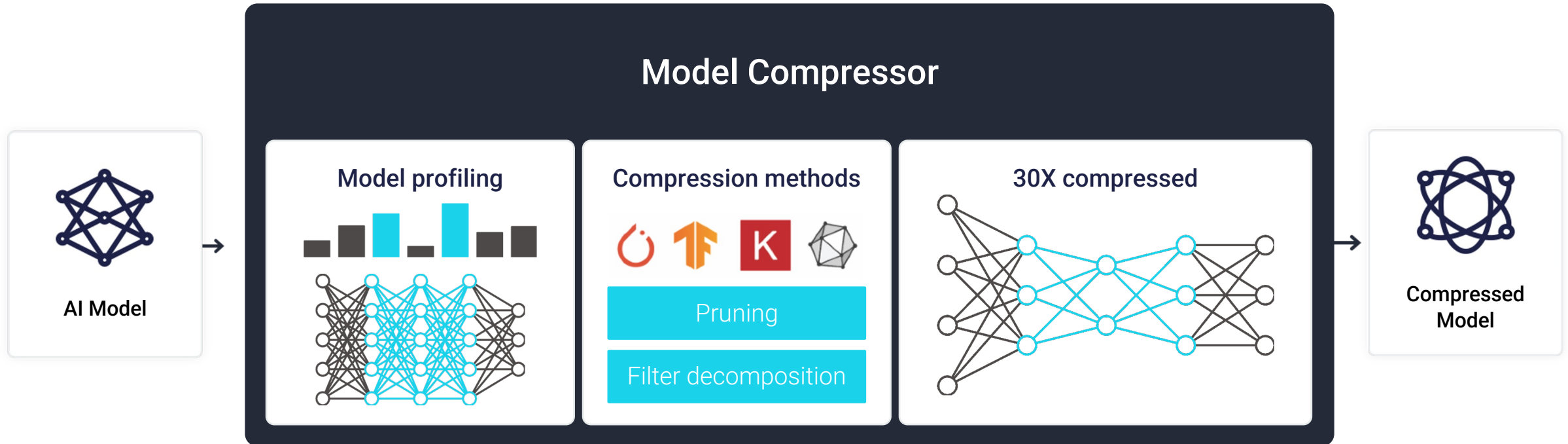
The Model Searcher module provides models optimized for your target device.



- Optimized model design for target hardware
- Multiple models with various options
- Produce models close to production level based on actual hardware testing
- Create models with lower latency

Model Compressor Module: Balancing Performance and Efficiency

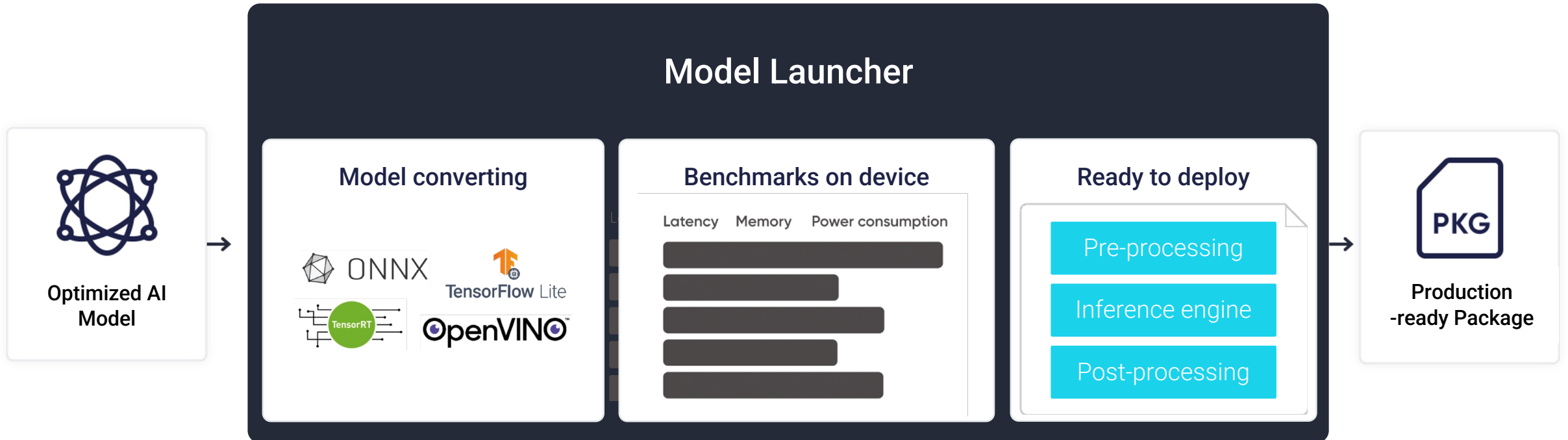
Achieve optimal performance-efficiency tradeoffs with the Model Compressor module.



- Supports all CNN architectures (limitedly transformer models)
- Eliminates months of paper implementation period
- Recommends optimal compression ratios
- Minimal loss of information

Model Launcher Module: Swift Launch for Accelerated Models

Accelerate model deployment with the Model Launcher module.



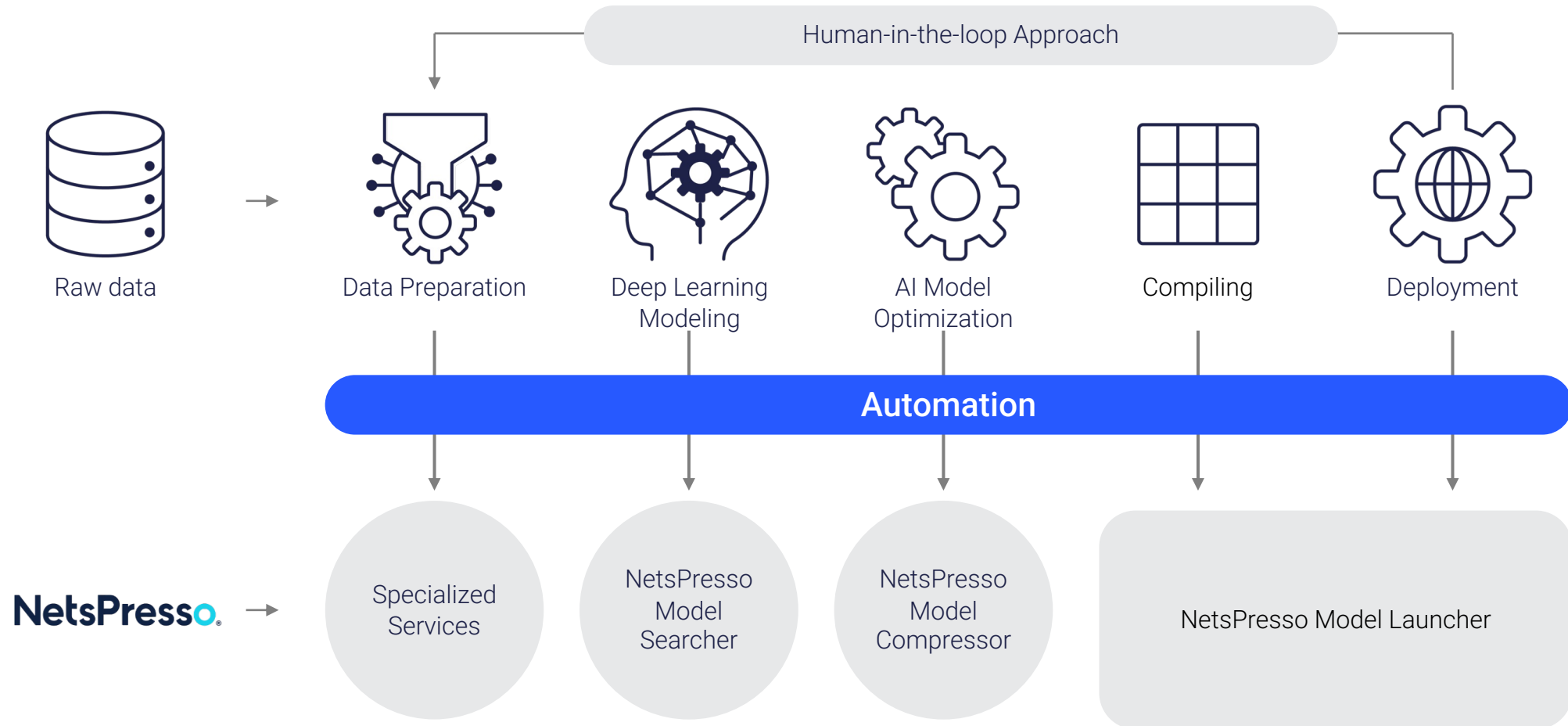
- Unified interface for various SDKs
- Performance benchmarks and recommendations on actual devices
- Conversion to operations supported by target hardware
- Production-ready package

Challenges vs NetsPresso

Challenges	Build & Search Model Searcher	Compress & Accelerate Model Compressor	Convert & Deploy Model Launcher
Various types of SDKs			✓
Diverse model conversion options			✓
Out of Memory	✓	✓	✓
Latency is too high	✓	✓	✓
Operation not supported	✓		✓

Platform Pipeline: Streamlined AI Model Development

NetsPresso facilitates seamless transitions from optimization to deployment.



Success Cases

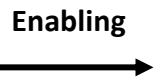
With Renesas

- Target: RA6M3 (ARM Cortex-M4)
- Model: YOLOv2
- Replace unsupported operation to be supported and HW-aware compression.
- Offering an extensive array of options to cater to diverse customer preferences.



Without Nota

Cannot operate
DET model



With Nota

Can operate
with
higher accuracy



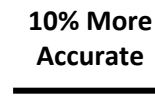
With Sony

- Target: IMX500
- Model: YOLO based model
- IMX500 is commonly used for smart city use cases.
- While preserving latency, Nota maximizes the capacity of the model so we could train higher performing model.



Without Nota

DET Model with
31.4% mAP



With Nota

DET Model with
41.1% mAP



Success Cases

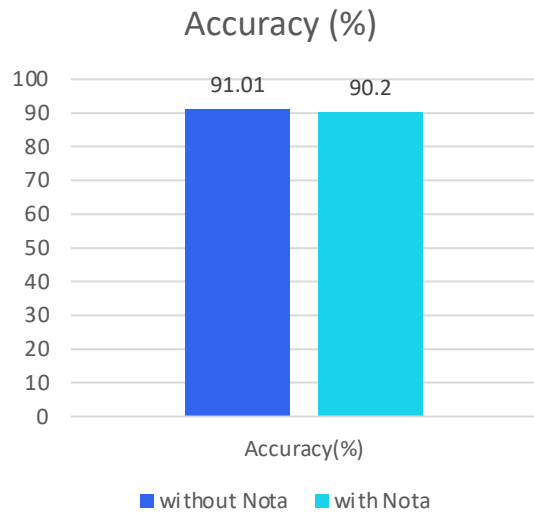
With STMicroelectronics

- Target: STM32H747I-DISCO
- Model: Mobilenet v2 (Tensorflow Flower Dataset)
- Compressed a model from the [STMicroelectronics model zoo](#) using NetsPresso Model Compressor and deployed it to a device using [STM32 Cube AI](#)
- Significant benefits in terms of memory and latency optimization with minimal performance loss
- This experiment was conducted by engineers of STMicroelectronics (sample code will be released)

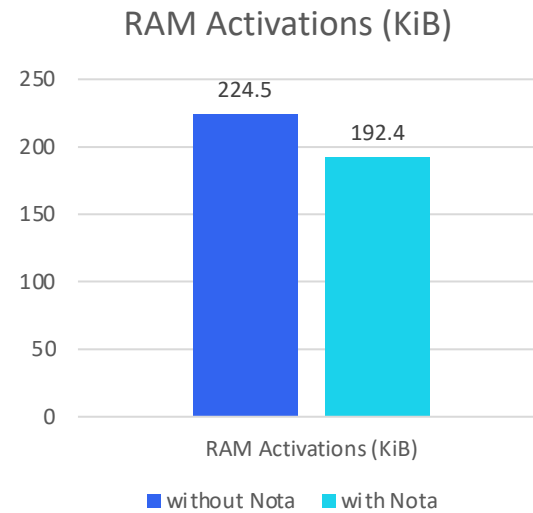


Success Cases

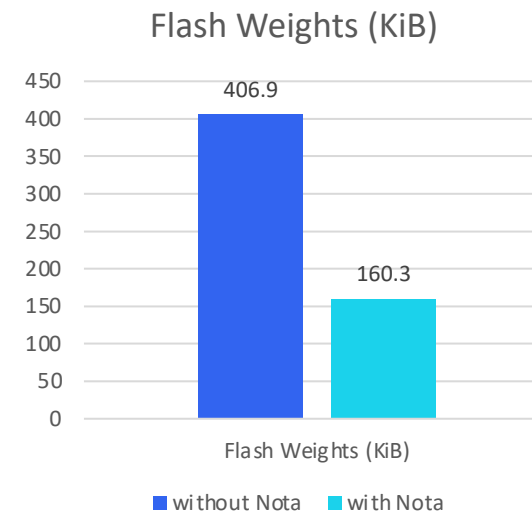
With STMicroelectronics



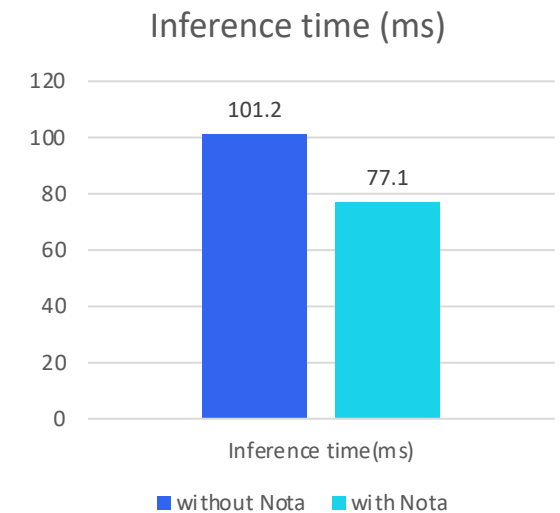
0.8%p drop



14% lighter



61% lighter



24% faster

Basic pruning options were used
(Global pruning ratio 0.5, using recommended layer-wise pruning ratio)

Ecosystem for Edge AI Development

With integration to model/device ecosystem, NetsPresso can bring benefit to broader area.



NetsPresso

NP Model Searcher

NP Model Compressor

NP Model Launcher

Build & Search

Compress & Accelerate

Convert & Deploy

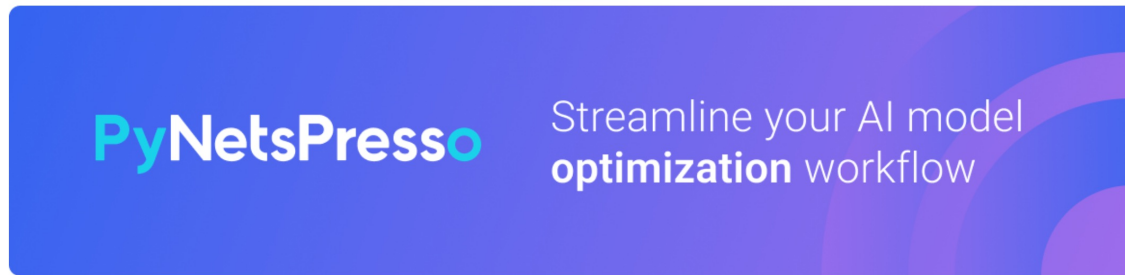


Nota's Device Farm

Nvidia, ARM, Renesas, Intel, Arduino, ...

Demo: PyNetsPresso and LaunchX

PyNetsPresso



[YOLOX](#) | [YOLOv8](#) | [YOLOv7](#) | [YOLOv5](#) | [PIDNet](#) | [PyTorch-CIFAR-Models](#)

python 3.8 | 3.9 | 3.10

TensorFlow 2.3.x ~ 2.8.x

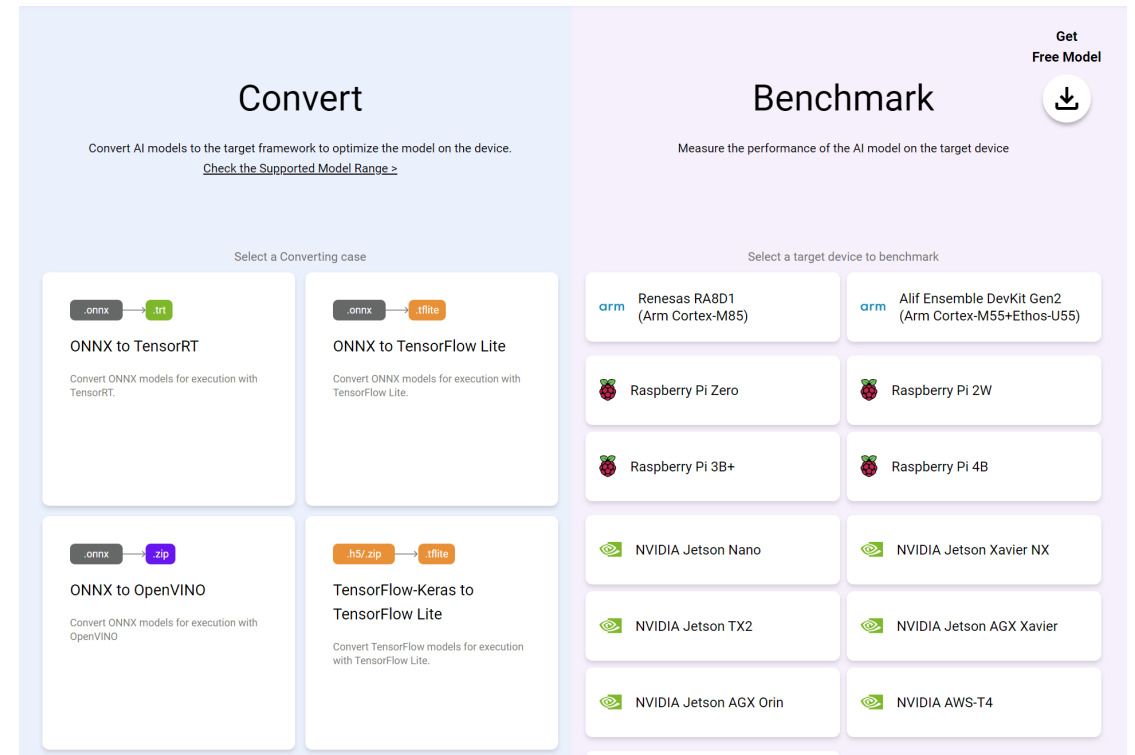
PyTorch 1.11.x ~ 1.13.x

NetsPresso [Open in Website](#)

[ModelZoo](#) [Open in Github](#)

[Best Practice](#) [Open in Colab](#)

LaunchX



* You can see the real demo at Nota AI booth

PyNetsPresso

Login

```
import getpass
from netspresso.client import SessionClient
from netspresso.compressor import ModelCompressor
```

```
email = 'xxx.yyy@st.com'
print('Enter you password')
password = getpass.getpass()
```

```
session = SessionClient(email=email, password=password)
compressor = ModelCompressor(user_session=session)
```

Enter you password

```
2023-10-27 09:42:35.715 | INFO      | netspresso.client:__login:50 - Login successfully
2023-10-27 09:42:37.787 | INFO      | netspresso.client:__get_user_info:67 - successfully got user information
```

PyNetsPresso

Uploading model

```
from netspresso.compressor import Task, Framework

model = compressor.upload_model(
    model_name='model',
    task=Task.IMAGE_CLASSIFICATION,
    framework=Framework.TENSORFLOW_KERAS,
    file_path='experiments_outputs/training/saved_models/best_model.h5',
    input_shapes=[{'batch': 1, 'channel': 3, 'dimension': [128, 128]}])
```

```
2023-10-27 09:52:50.120 | INFO      | netspresso.compressor:upload_model:72 - Uploading Model...
2023-10-27 09:53:06.984 | INFO      | netspresso.compressor:upload_model:83 - Upload model successfully. Model ID: 94d350dc-0297-4964-aa4b-d033b93ffc9f
```

PyNetsPresso

Compression

```
import os

from netspresso.compressor import CompressionMethod
from netspresso.compressor import RecommendationMethod

if not os.path.exists('experiments_outputs/compressed_models'):
    os.makedirs('experiments_outputs/compressed_models')

compressed_model = compressor.recommendation_compression(
    model_id='94d350dc-0297-4964-aa4b-d033b93ffc9f',
    model_name='compressed_model.h5',
    compression_method=CompressionMethod.PR_L2,
    recommendation_method=RecommendationMethod.SLAMP,
    recommendation_ratio=0.5,
    output_path='experiments_outputs/compressed_models/compressed_model.h5',)
```

```
2023-10-27 09:53:59.084 | INFO | netspresso.compressor:recommendation_compression:448 - Compressing recommendation-based model...
2023-10-27 09:54:19.376 | INFO | netspresso.compressor:download_model:223 - Downloading model...
2023-10-27 09:54:31.058 | INFO | netspresso.compressor:download_model:226 - Download model successfully. Local Path: experiments_outputs/compressed_m
odels/compressed_model.h5
2023-10-27 09:54:31.063 | INFO | netspresso.compressor:get_model:196 - Getting model...
2023-10-27 09:54:33.535 | INFO | netspresso.compressor:get_model:202 - Get model successfully.
2023-10-27 09:54:33.538 | INFO | netspresso.compressor:recommendation_compression:505 - Recommendation compression successfully. Compressed Model ID:
49ac11c4-2c56-4575-b215-b7f184ca4693
2023-10-27 09:54:33.542 | INFO | netspresso.compressor:recommendation_compression:506 - 50 credits have been consumed.
```

LaunchX

Convert

Convert AI models to the target framework to optimize the model on the device.
[Check the Supported Model Range >](#)

Select a Converting case

 → 

ONNX to TensorRT

Convert ONNX models for execution with TensorRT.

 → 

ONNX to TensorFlow Lite

Convert ONNX models for execution with TensorFlow Lite.

 → 

ONNX to OpenVINO

Convert ONNX models for execution with OpenVINO

 → 

TensorFlow-Keras to TensorFlow Lite

Convert TensorFlow models for execution with TensorFlow Lite.


Benchmark


Get
Free Model





Measure the performance of the AI model on the target device


Select a target device to benchmark


 Renesas RA8D1
(Arm Cortex-M85)


 Alif Ensemble DevKit Gen2
(Arm Cortex-M55+Ethos-U55)


 Raspberry Pi Zero

 Raspberry Pi 2W

 Raspberry Pi 3B+


 Raspberry Pi 4B


 NVIDIA Jetson Nano

 NVIDIA Jetson Xavier NX

 NVIDIA Jetson TX2

 NVIDIA Jetson AGX Xavier

 NVIDIA Jetson AGX Orin

 NVIDIA AWS-T4

LaunchX

AI Model Converter

File name : lightweight-yolox-nano.onnx

Input Shape

Batch size *

1

Channel *

3

Input size *

416, 416

* Only single input models are supported
For single input, the value is set automatically.
In case of dynamic input, you need to set the value.

- Batch size: The number of combined input datasets that the model processes simultaneously.
- Channel: enter 3 for RGB channel and 1 for gray channel.
- Input size: In computer vision tasks, input size refers to the size of the input images. For example, width=1024, height=768, enter 768,1024.

Target Framework *

TensorFlow Lite

TensorRT

OpenVINO

You can only set the batch size to 1 for TensorFlow Lite conversion.

Target device *

Ensemble-E7-DevKit-Gen2

Output datatype *

FP16 INT8

Cancel

Convert

LaunchX

AI Model Benchmarker

File name: model_test.tflite

Target device

arm Arm

Renesas RA8D1 (Arm Cortex-M85)
with Helium


Renesas RA8D1 (Arm Cortex-M85)

Alif Ensemble DevKit Gen2 (Arm
Cortex-M55 + Ethos-U55) with
Helium

Alif Ensemble DevKit Gen2 (Arm
Cortex-M55 + Ethos-U55)

 NVIDIA

No devices for this model

 Raspberry Pi

Raspberry Pi 4B

Raspberry Pi
3B+

Raspberry Pi
Zero 2 W

Raspberry Pi
Zero W

intel Intel

No devices for this model

Cancel

Benchmark

LaunchX

AI Model Benchmark

! Result

File name: model_test.tflite

Output format	Data type	Batchsize	Input size	Channel
Tensorflow Lite	None	1	416, 416	3

Inference latency
(avg)

64.391 ms

File size: 0.96 MB

🔧 Device Info

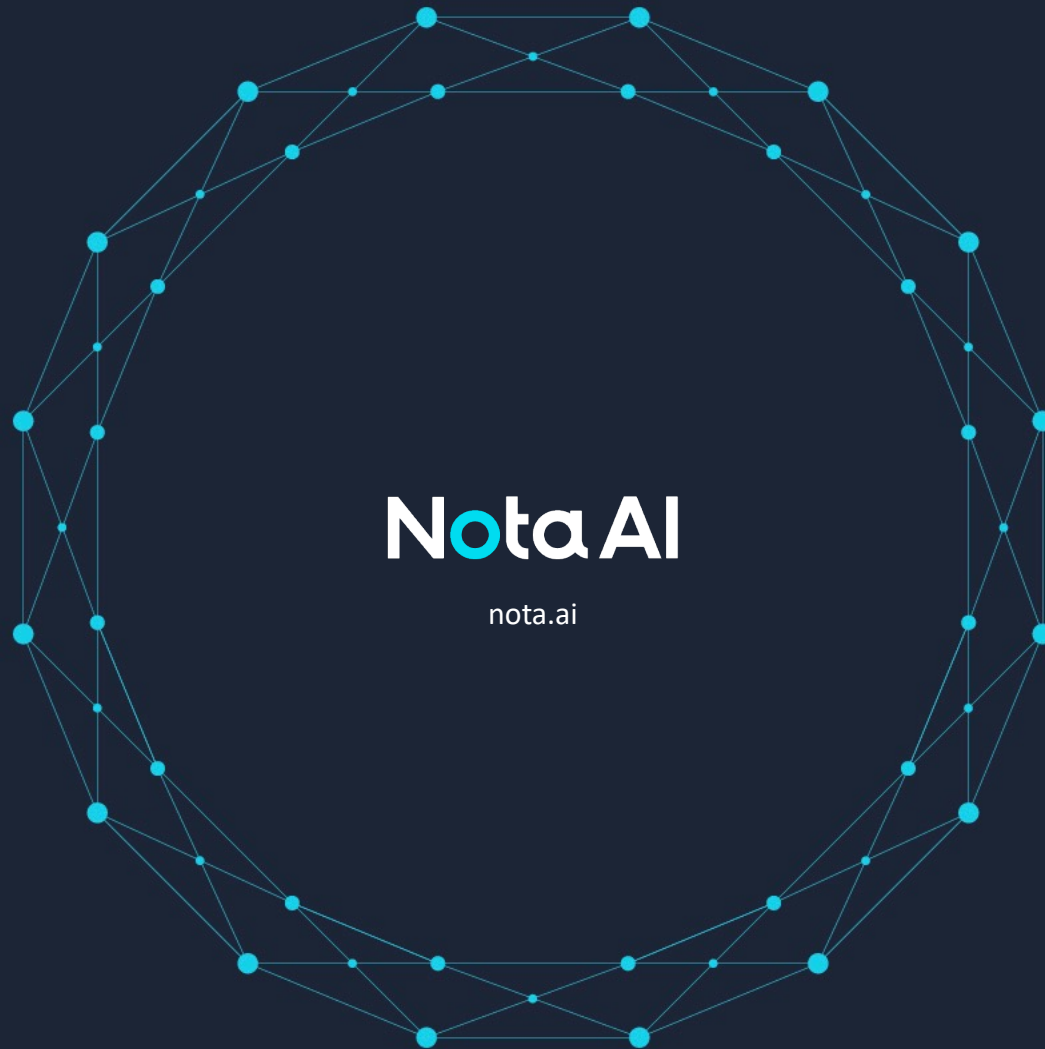
Ensemble-E7-DevKit-Gen2

- Library: Python 3.10.12 NumPy 1.26.1 TFLite 2.9.1
- CPU: Arm Dual-core CORTEX-M55 up to 400 MHz, with Helium, Dual-core ETHOS-U55 NPU
- GPU: N/A
- RAM: 15.827 GB

* You can do the same task with PyNetsPresso



Let's do happy edge AI engineering with NetsPresso!



Copyright Notice

This presentation in this publication was presented as a tinyML® Asia Technical Forum. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org