



How to Select, Train, Optimize and Deploy Edge Vision AI Models in Three Days

Steven Kim

Chief Executive Officer

Nota America Inc.

NotaAI

Before We Start

- When a new hardware is released in the market, customers want to run the latest AI model on the device
 - Example
 - Customers: Want to run YOLOv5 on a device to maximize performance (latency, accuracy, and power consumption)
 - Device: Only supports YOLOv2 and 3
- In order to close this gap, hardware-aware optimization is a must!

NetsPresso® (Not The Coffee Maker)

Nota AI

Expectation vs Reality

- **Expectation:**

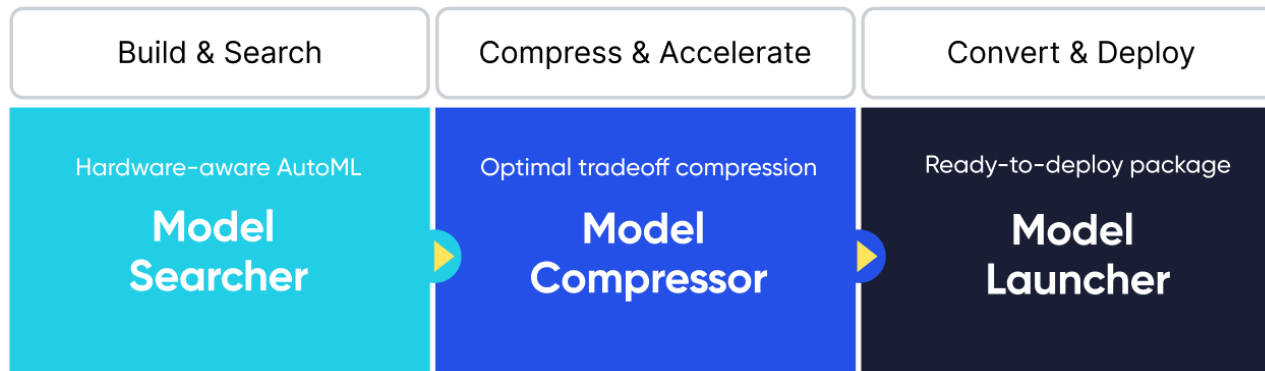
- How do we build the perfect model for the next 6 months?
- Once trained, the model is complete

- **Reality:**

- AI model development does not end after a project
 - AI model deployment is NOT a one-off process
 - AI model deployment is a CONTINUAL process

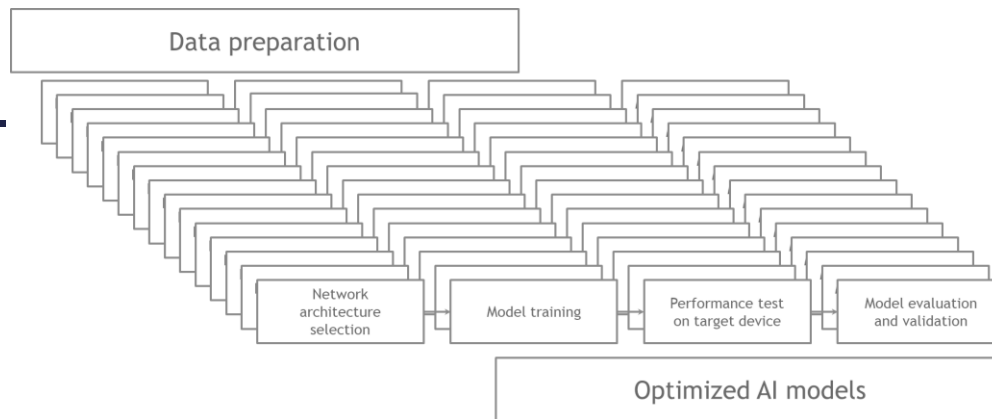
Why We Built Our Pipeline

- **Question from AI engineers:**
 - How do we build a pipeline to update and deploy a model as fast as possible?
- Thus, Nota AI built its own pipeline, **NetsPresso®**



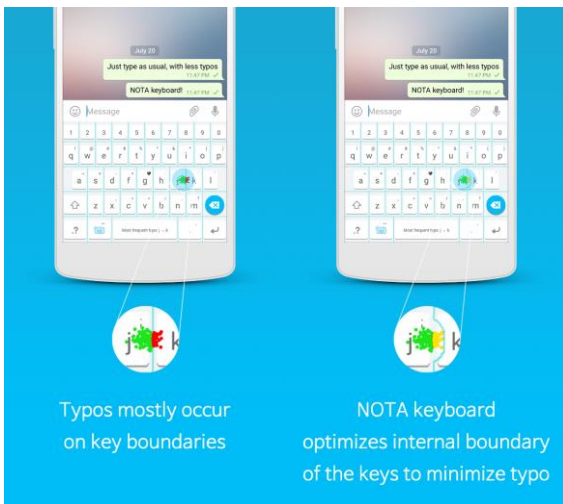
Pain Point: Traditional AI Model Development Process

- Development process is **repetitive and resource-intensive**

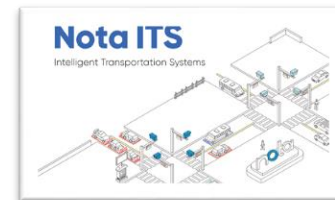


The Beginning: Nota AI

- Once upon a time,
 - Nota AI was started as a smart mobile keyboard app company



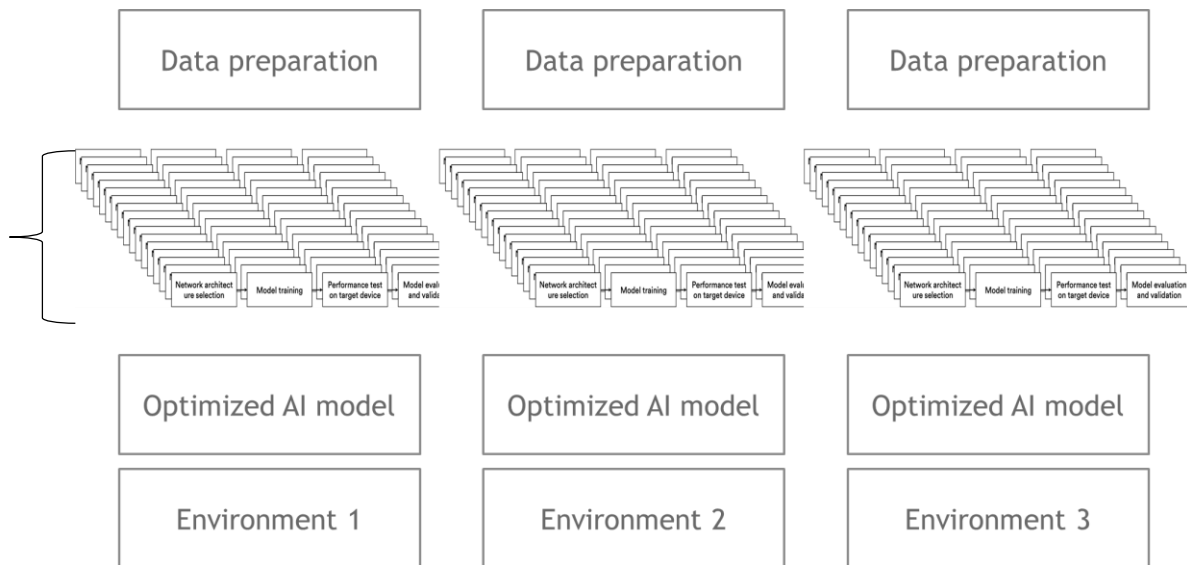
NetsPresso®



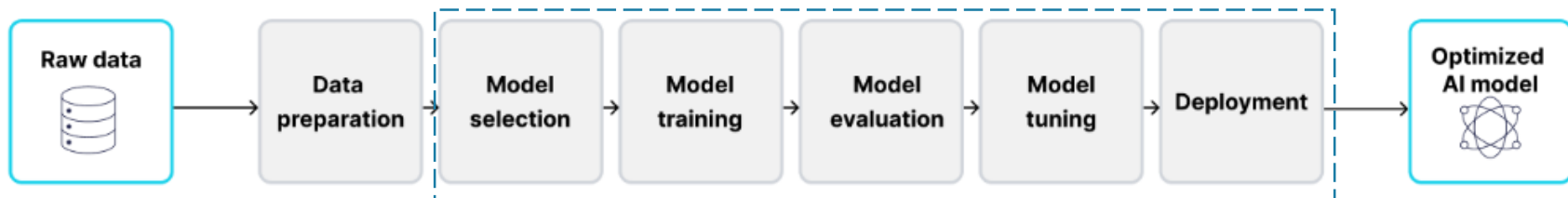
Pain Point: HW-Aware AI Model Development Process

- Development process is **even more repetitive!!!**

Manual
process



Vision AI Model Development Process



Without NetsPresso: 6 to 12 weeks
With NetsPresso : 2 to 3 days

- Better performance & more diverse applications in a short time



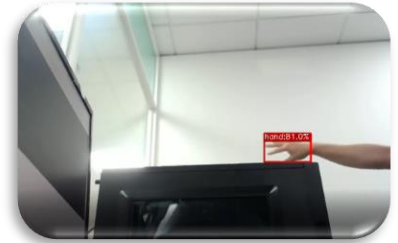
Intrusion detection



Fire detection



Pothole & crack detection



Hand Detection



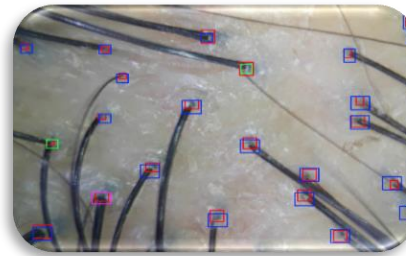
Fall Detection



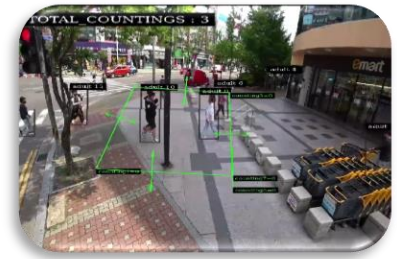
Safety Helmet
Detection



Railroad and
Obstacle Detection



Pore Detection



People counting

Case Study

Nota AI

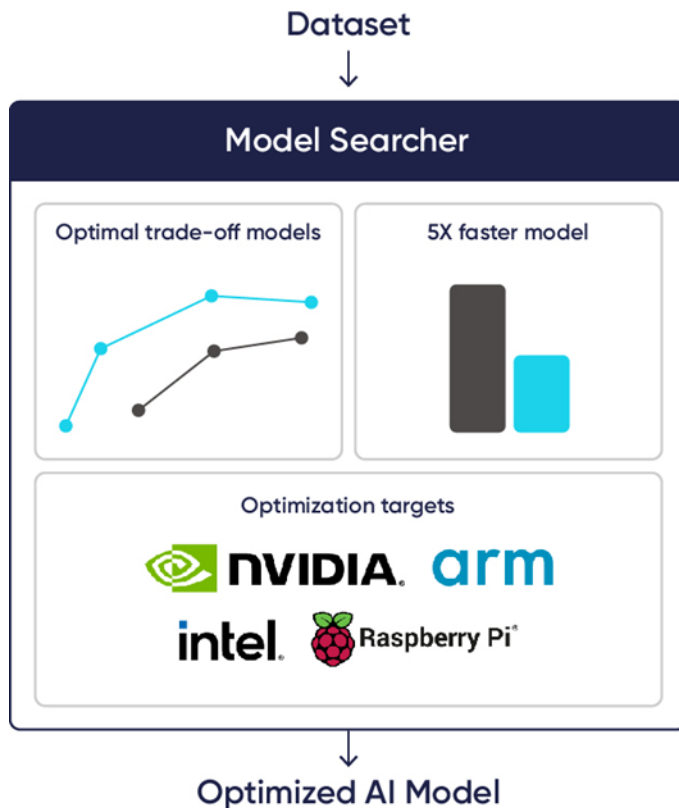
- **Challenge**

- The response time of the existing model was too slow (**589 ms**) to be able to detect potholes from a dashcam
- The requirement was less than **300 ms**



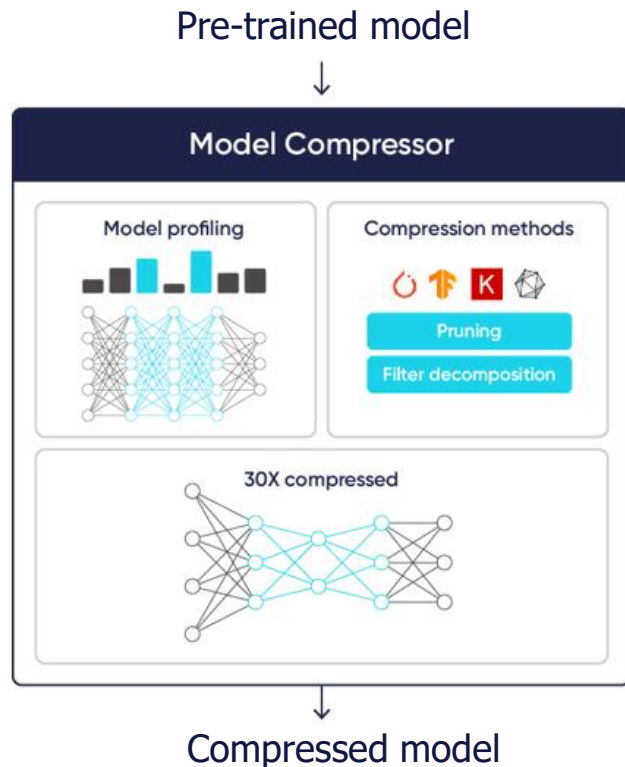
Pothole Detection

- With NetsPresso®
 - Using Model Searcher, found two models with better latency for Jetson Nano
 - 343 ms (-42%)
 - 186 ms (-68%)



Pothole Detection

- With NetsPresso®
 - Using Model Compressor, improved the latency by 20-30%
 - 343 ms -> **239 ms**
 - 186 ms -> **147 ms**



Pothole Detection

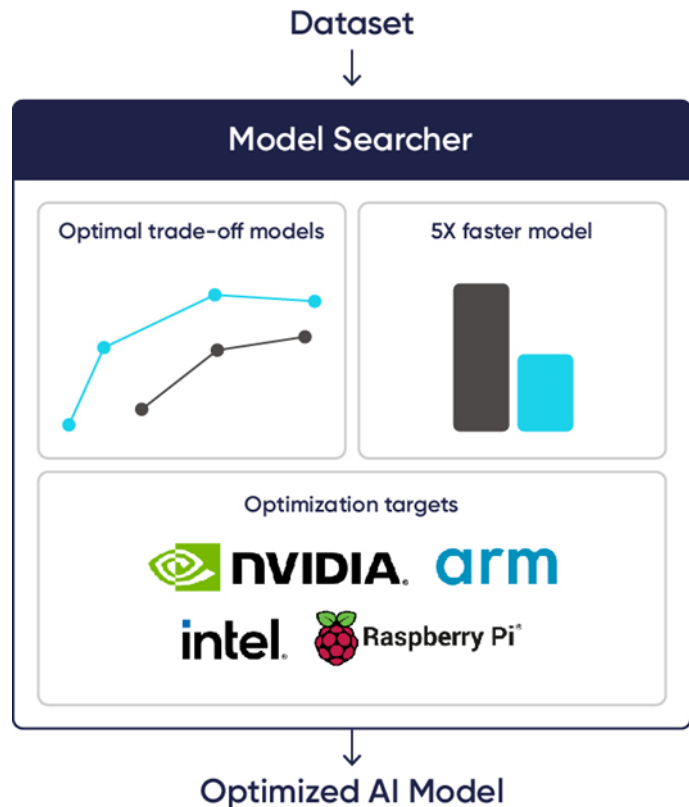


Fire Detection

- Challenge
 - Greater than 10 FPS on Jetson Nano to detect fire
- Result
 - With NetsPresso®: > 30 FPS



- With NetsPresso®
 - Using Model Searcher, found a model with >3x better FPS
 - 31.2 FPS (+170%)



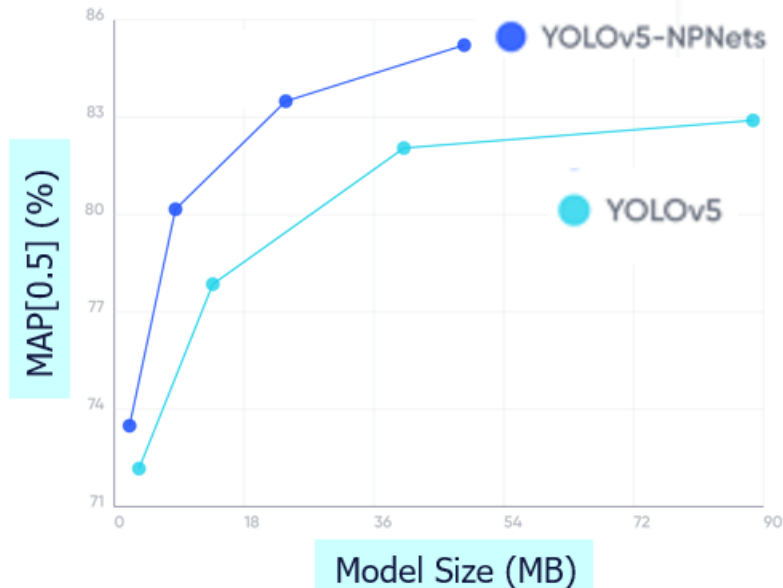
Case Study II: Fire Detection



Model Searcher: Performance Benchmark

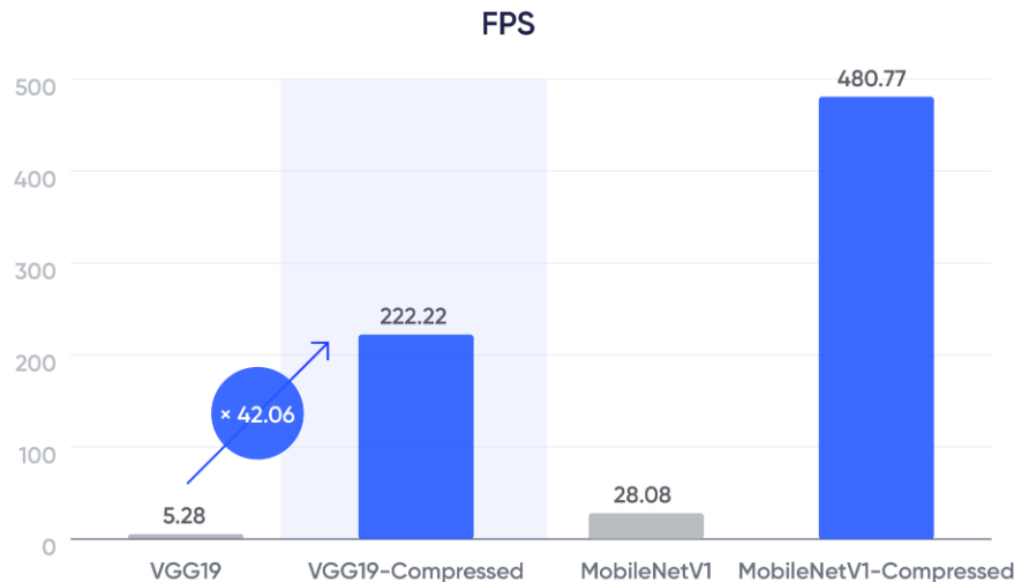
- Both accuracy (mAP) and GFLOPs are improved

Object Detection



Model Compressor: Performance Benchmark






- With a minimal drop in accuracy ($\sim 1\%$), a significant improvement (**> 4,200%**) made for FPS



Benefit: Inference Server Cost Saving

- By using the optimized AI model, up to **85%** inference server cost saving was achieved.

Comparison Table

	Original		NotaAI
	V100	Lower →	T4
	X 8	→	X 16
	22FPS	14% faster →	25FPS
	97.14%	- 0.1% →	97.03%
	\$2,203	85% save →	\$391

How We Can Help

NotaAI

Business Model: NetsPresso® & Solutions

- **Platform**

- NetsPresso
- Professional Service

- **AI Solution**

- Intelligent Transportation System
- Driver Monitoring System

<https://www.nota.ai/contact-us>

Key Takeaways

- Closing the performance gap between what the device companies can offer and what the market wants requires optimization and SW+HW co-design to seize the market opportunity
- Keeping up with newer models, datasets, and frameworks has increased the need for a pipeline to retrain and deploy AI models more efficiently across different edge devices
- If you want better AI models faster, please talk to us!

NetsPresso: HW-aware AI Model Development Pipeline

The screenshot shows the NetsPresso web interface with a dark sidebar on the left containing navigation links: Models, Datasets, Projects (highlighted), Compress, and Package. The main content area is titled 'Target device' and includes several configuration sections:

- Target device ***: A red box highlights this section, which contains radio buttons for 'NVIDIA Jetson', 'Raspberry Pi', 'Intel Xeon W-2223', and 'AVH Corstone-300 (Ethos-U65 High End)'. A dropdown menu next to 'NVIDIA Jetson' is set to 'Nano'.
- Output Format**: Includes 'Framework *' (TensorRT) and 'SW version *' (JetPack 4.6).
- Output datatype ***: Radio buttons for FP32, FP16 (selected), INT8, and INT4.
- Inference batch size ***: A text input field containing '1'. A tooltip below indicates a support range of 1~32 and that TFLite only supports batch size 1.
- Model training**: A red box highlights this section, which includes 'Target latency (ms) *' with a text input field containing '500'. A tooltip below indicates a support range of 10~5000.

Hardware-aware
(More to come)

Sign up for free credit

<https://www.netspresso.ai/>

For more information on Nota AI:

<https://www.nota.ai>

To try NetsPresso®:

<https://www.netspresso.ai>

Visit our booth #217

