AWS re:Invent

AIM 302

Machine Learning at the Edge

Julien Simon
Principal Tech. Evangelist, AI/ML
Amazon Web Services
@julsimon

Brian Kursar
Chief Data Scientist
Toyota Connected Data Services
Brian.Kursar@toyotaconnected.com
@briankursar

Nimish Amlathe
Data Scientist
Toyota Connected Data Services
Nimish.Amlathe@toyotaconnected.com

AWS re:Invent

Agenda

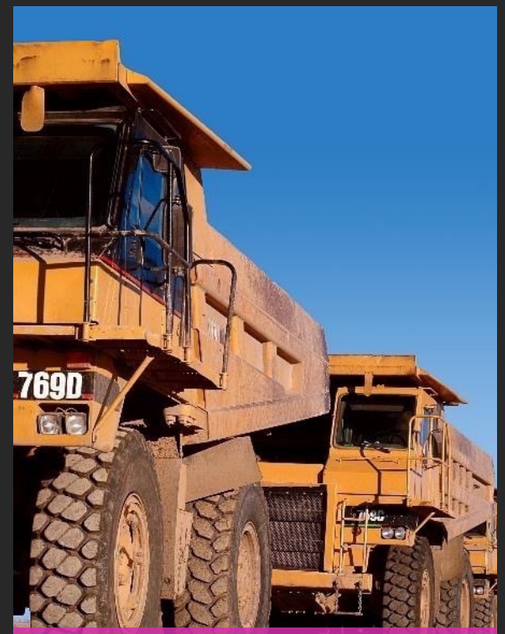
- Machine Learning at the Edge?
- Leveraging AWS Services
- Case study: Toyota Connected Data Services
- Alternative scenarios
- Optimizing for inference at the Edge
- Getting started

Machine Learning at the Edge?

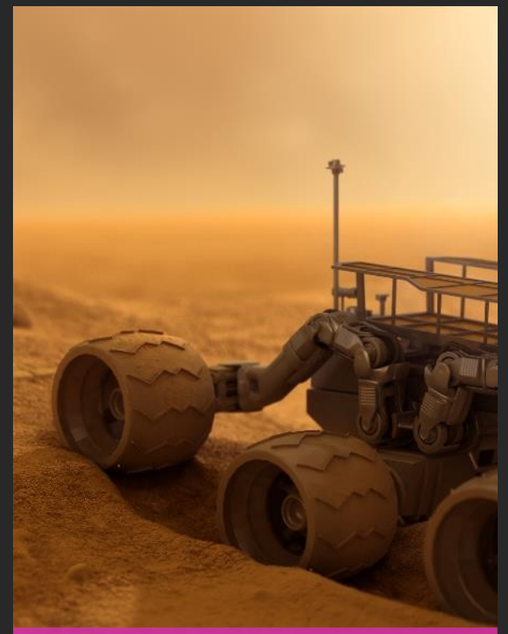
AWS re:Invent

Most machine data never reaches the cloud

Medical equipment

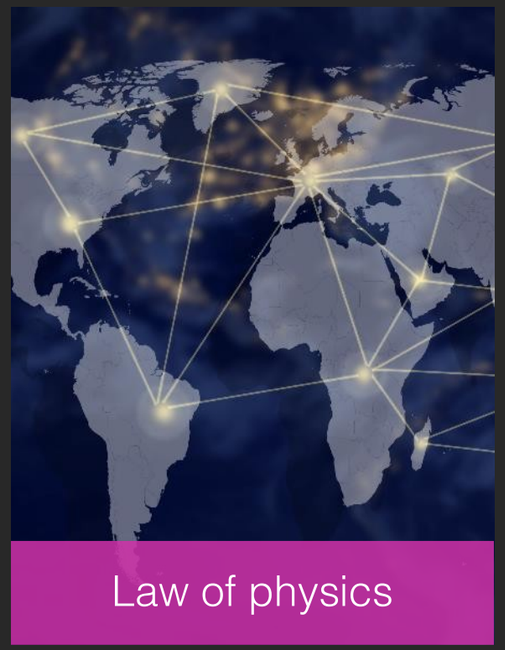


Industrial machinery

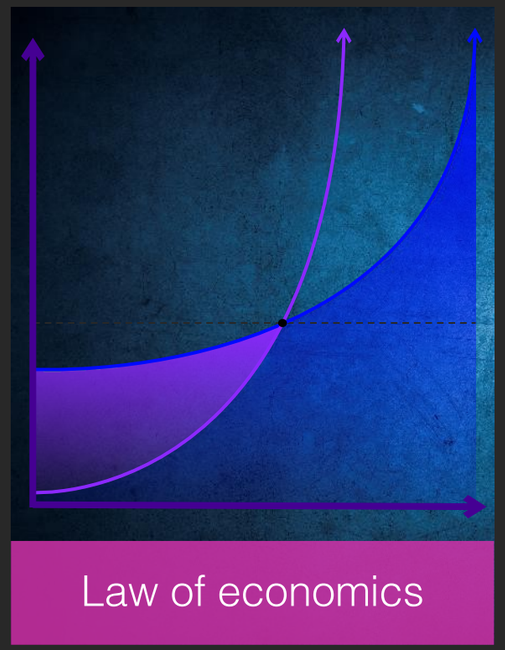


Extreme environments

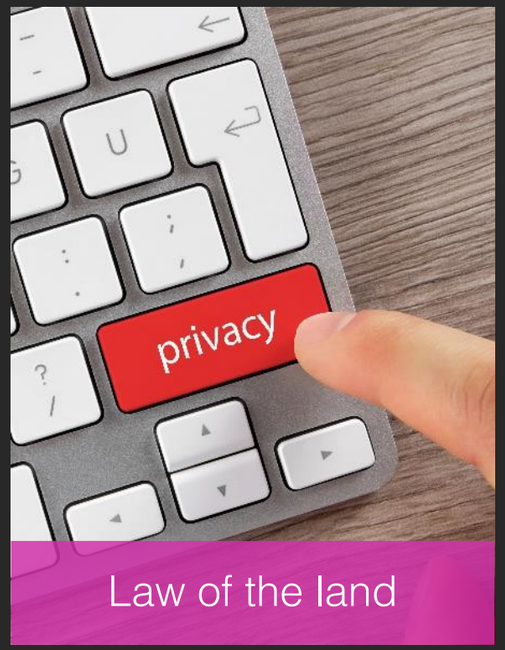
Why this problem isn’t going away



Law of physics



Law of economics



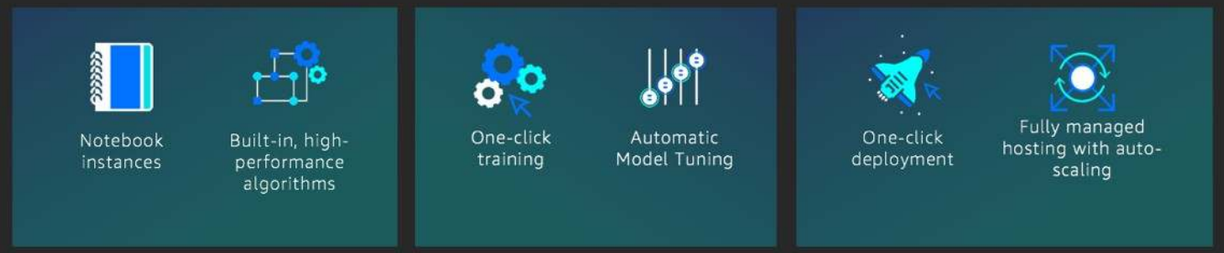
Law of the land

Leveraging AWS Services

AWS re:Invent

Experimenting and training in the Cloud

Amazon SageMaker



AWS Deep Learning AMI

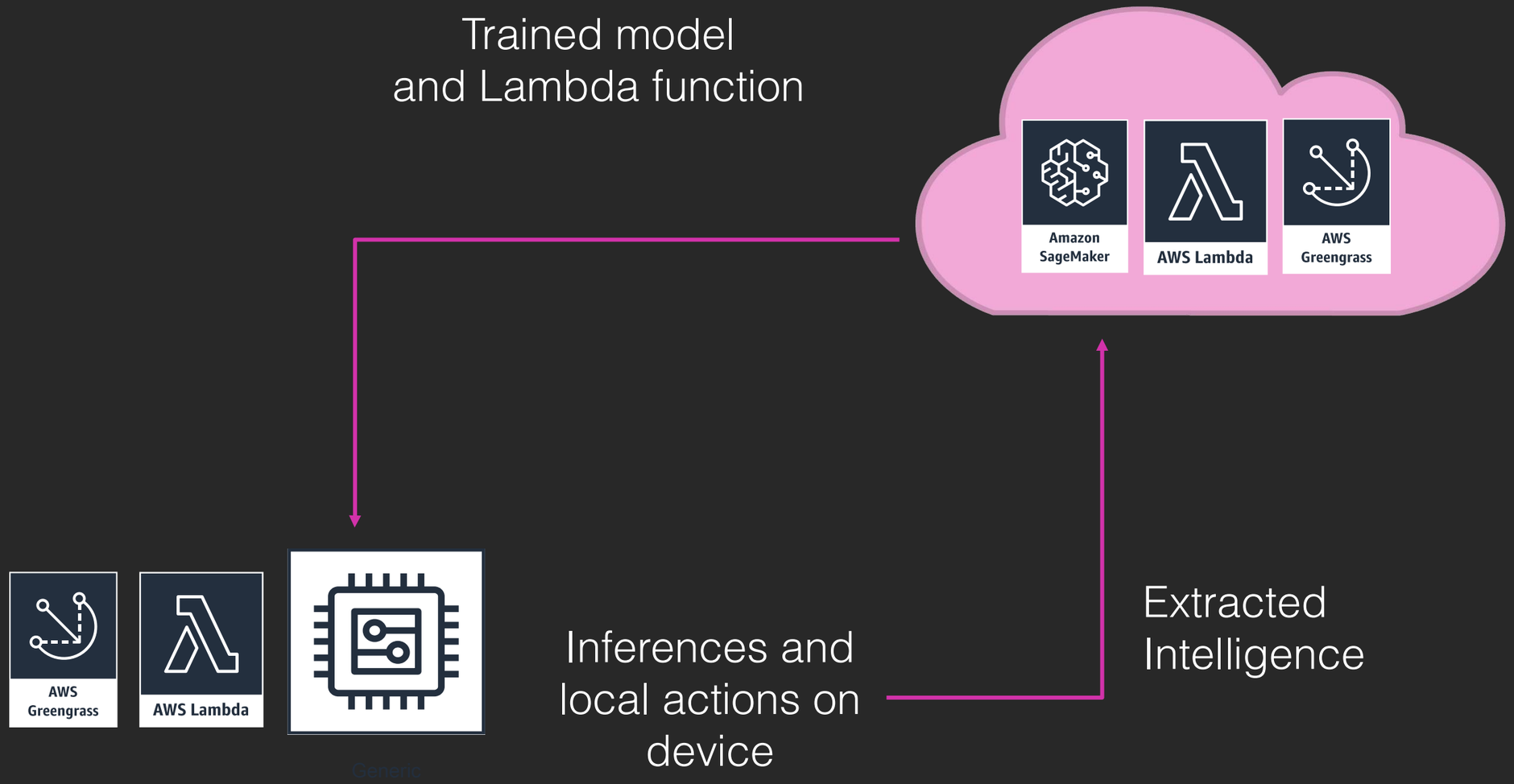
TensorFlow, K, Microsoft CNTK, ONNX, mxnet, GLUON, Chainer, HOROVOD, Caffe2, PYTORCH, torch

Amazon EC2

c5

p3

Deploying and predicting with AWS Greengrass



Deploying with AWS Greengrass

- Train a model in Amazon SageMaker (or import your own)
- Write an AWS Lambda function for prediction
- Add both as resources in your AWS Greengrass group
- Let AWS Greengrass handle deployment and updates

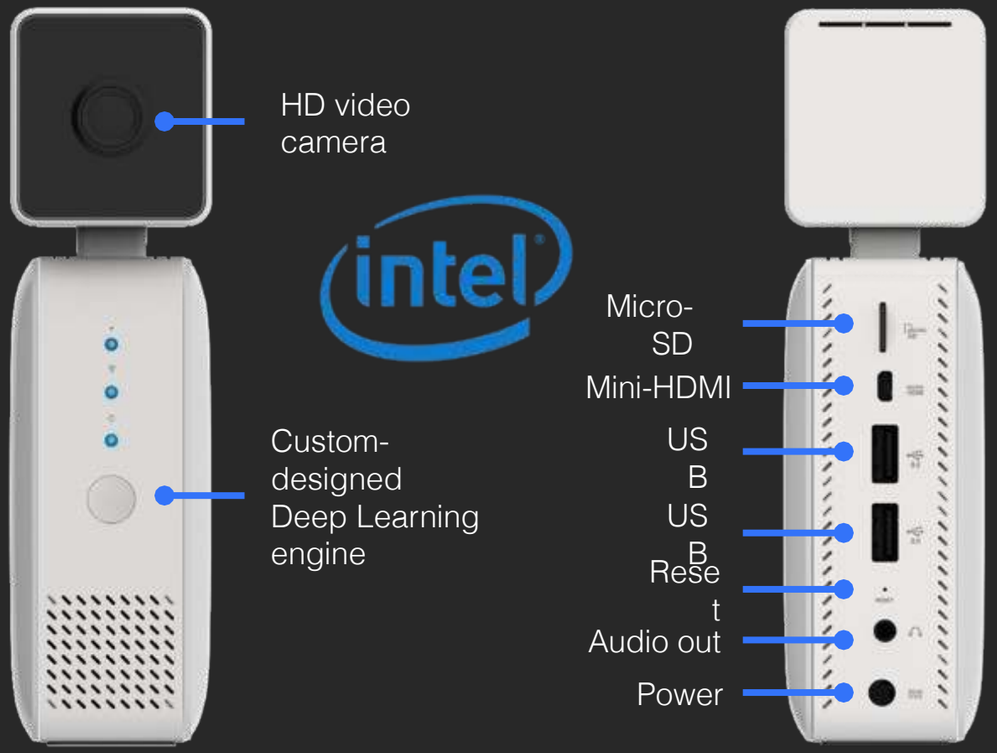
Best when
You want the same programming model in the Cloud and at the Edge.
Code and models need to be updated, even if network connectivity is infrequent or unreliable.
You need one device in the group to perform prediction on behalf on other devices.

Requirements
Devices are powerful enough to run AWS Greengrass
Devices are provisioned in AWS IoT Core (certificate, keys).

AWS DeepLens

AWS re:Invent

AWS DeepLens



A new way to learn

Custom built for Deep Learning

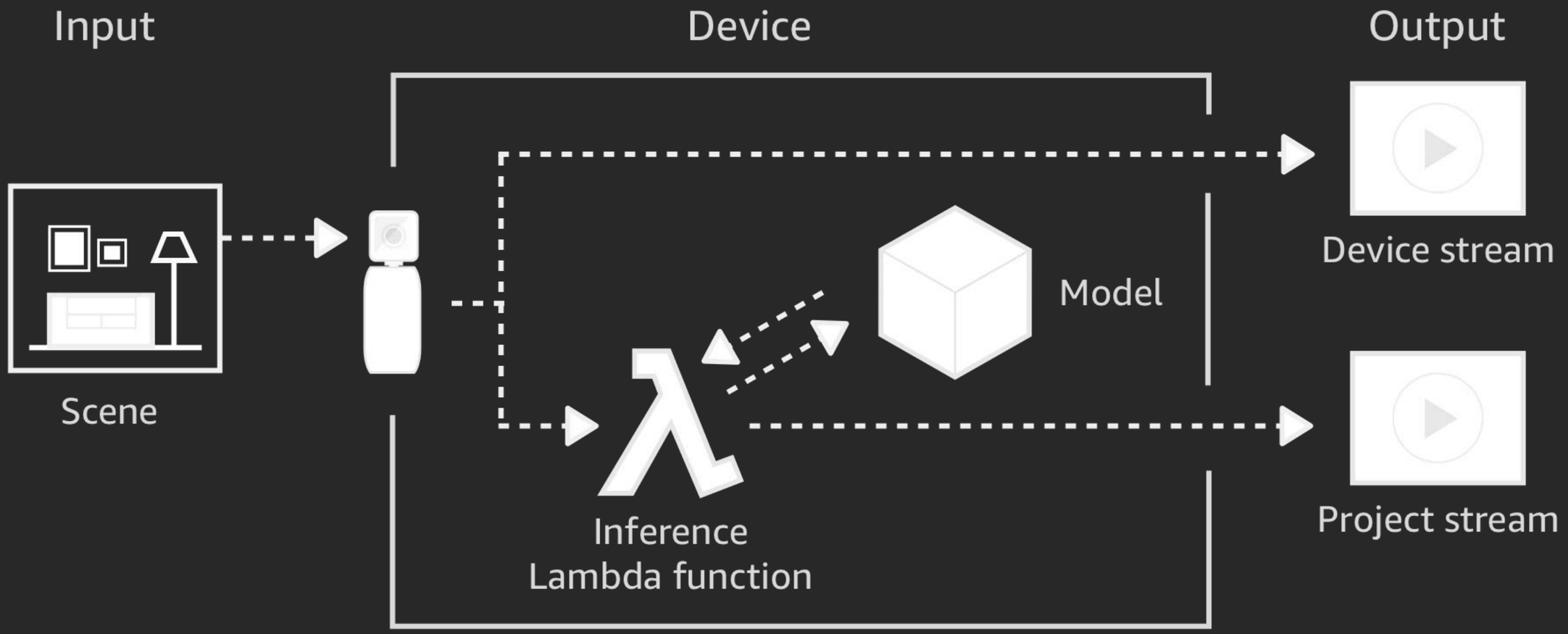
Broad framework support

Deploy models from Amazon SageMaker

Integrated with AWS

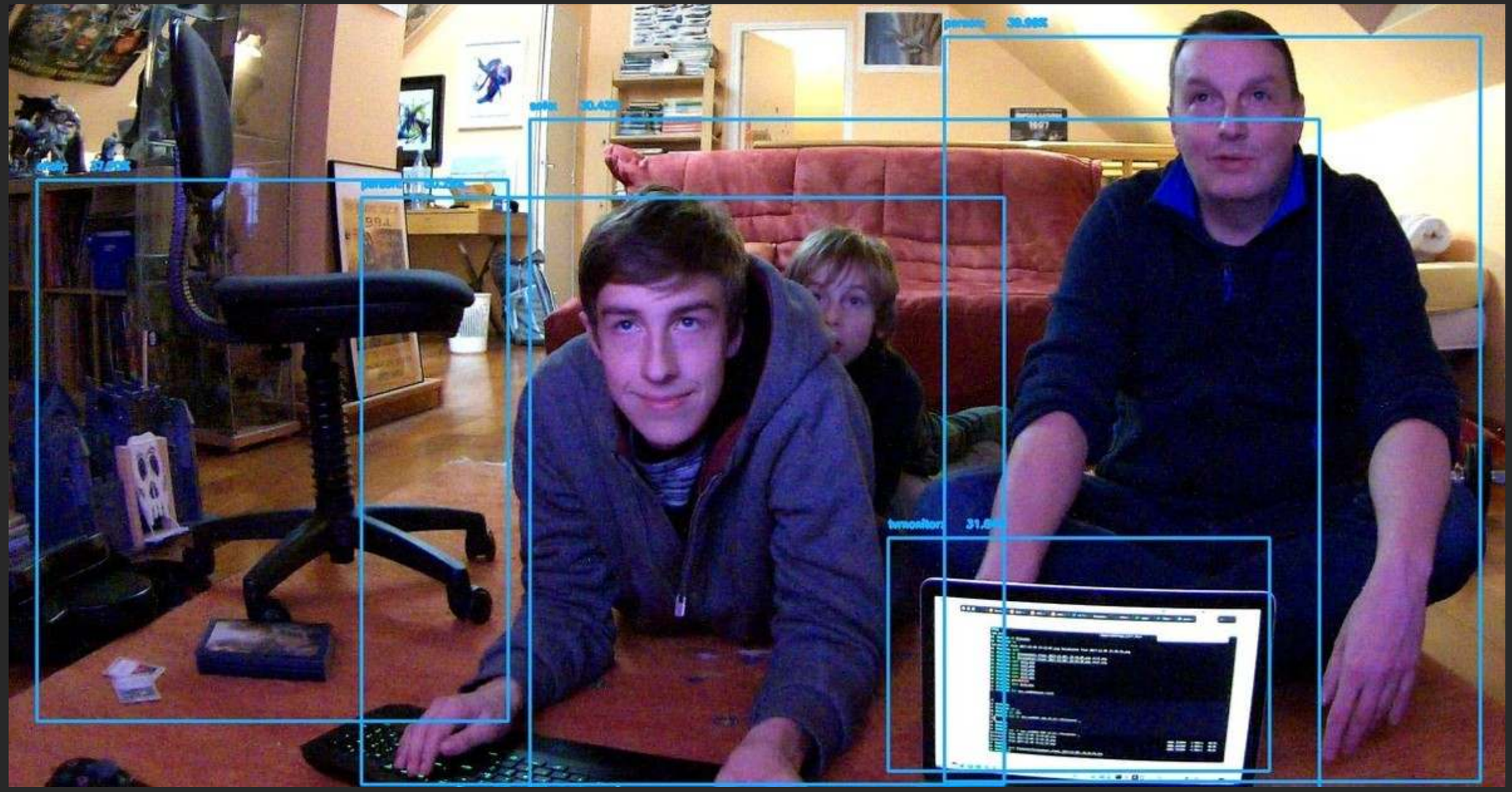
Fully programmable with AWS Lambda

AWS DeepLens



AWS re:Invent

Object detection with AWS DeepLens



Using your own models with AWS DeepLens

- AWS DeepLens can run TensorFlow, Caffe and Apache MXNet models.
  - Inception
  - MobileNet
  - NasNet
  - ResNet
  - Etc.
- Train or fine-tune your model on Amazon SageMaker
- Deploy to DeepLens with AWS Greengrass

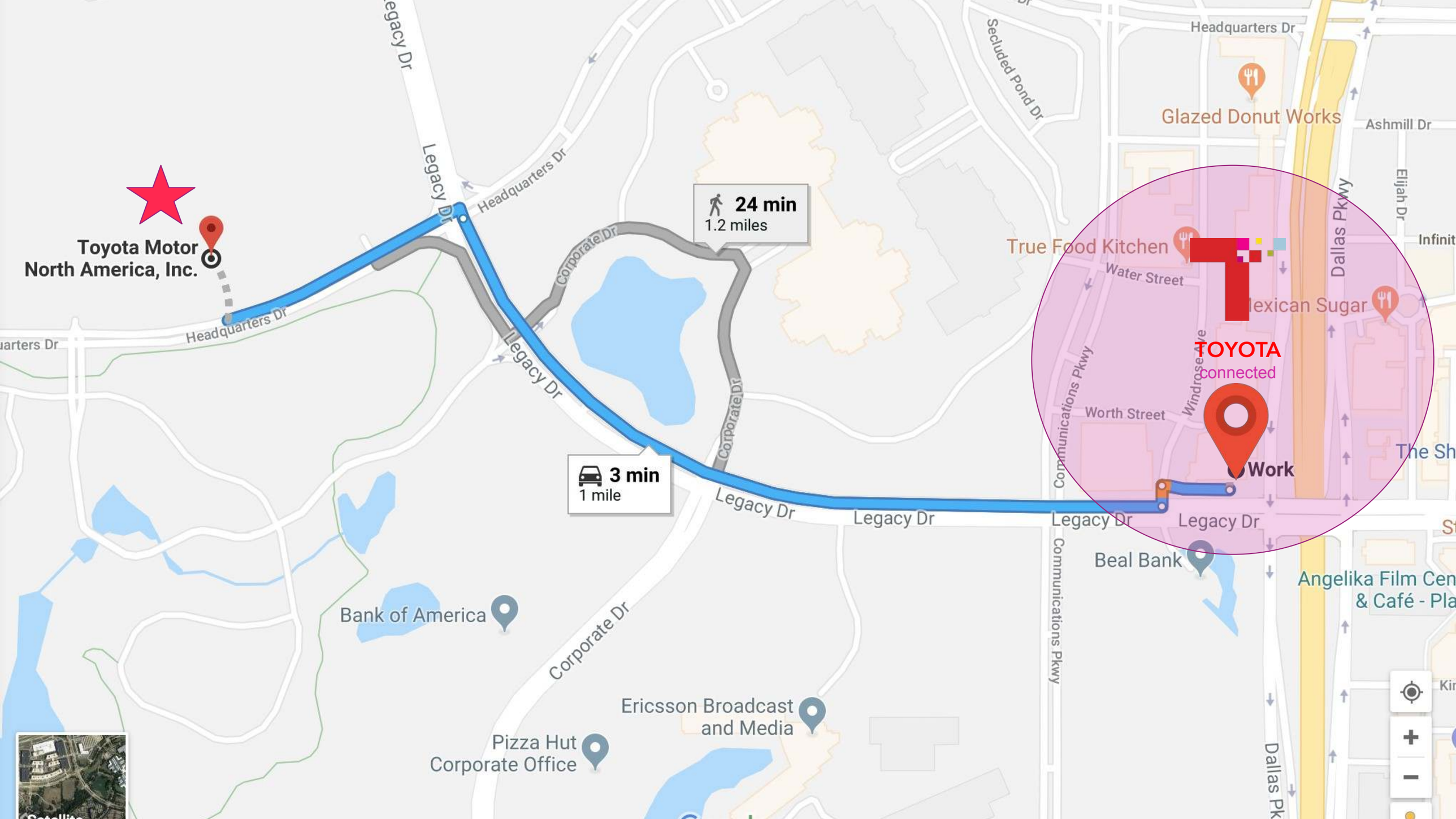
TOYOTA
connected

DEEP LEARNING AT THE EDGE

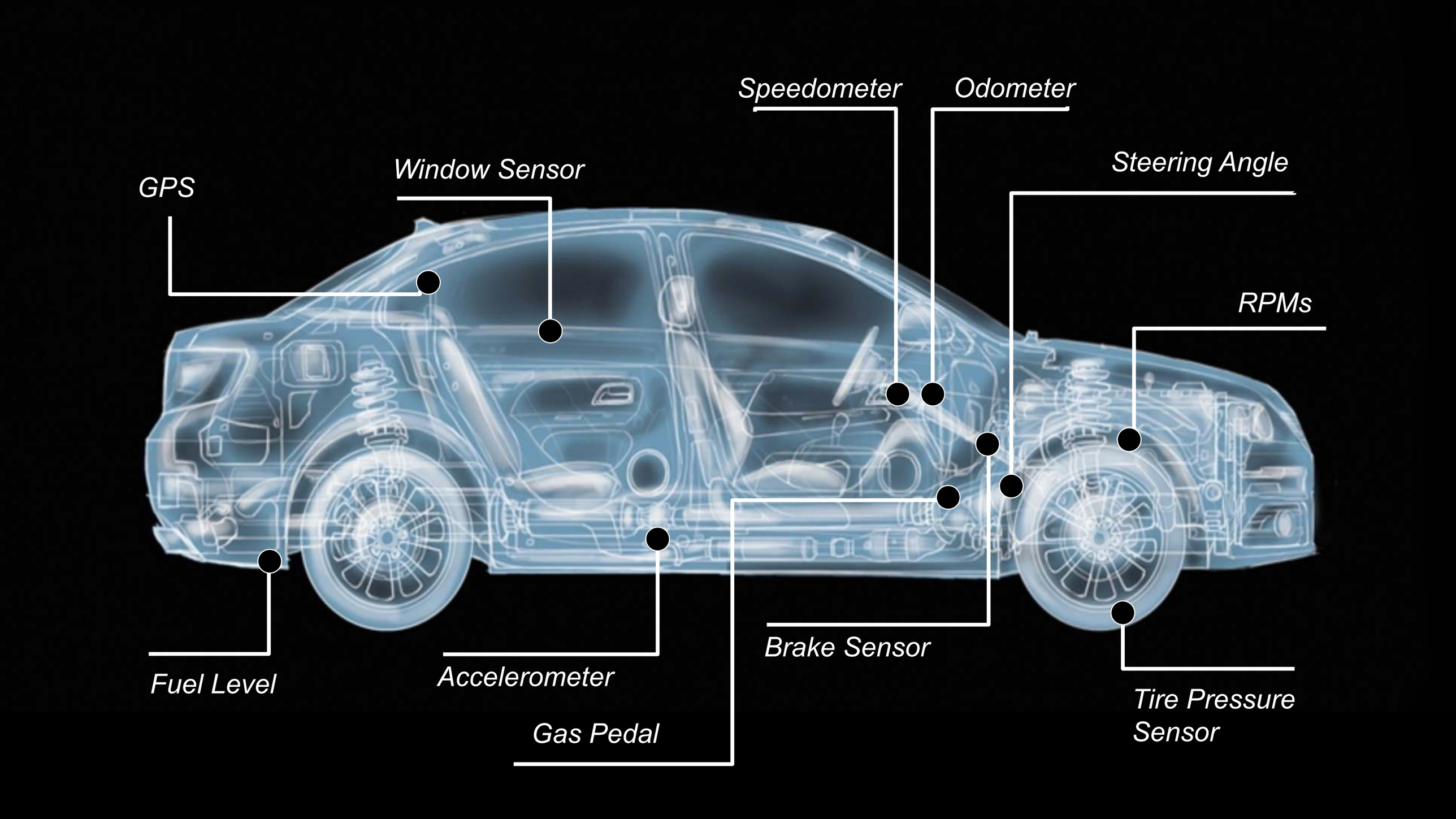
BRIAN KURSAR
VICE PRESIDENT AND CHIEF DATA SCIENTIST

NIMISH AMLATHE
DATA SCIENTIST

TOYOTA CONNECTED DATA SERVICES


Toyota Motor
North America, Inc.
Headquarters Dr
Legacy Dr
Corporate Dr
24 min
1.2 miles
3 min
1 mile
Work
TOYOTA
connected
Glazed Donut Works
True Food Kitchen
Water Street
Worth Street
Windrose Ave
Communications Pkwy
Dallas Pkwy
Ashmill Dr
Elijah Dr
Secluded Pond Dr
Beal Bank
Bank of America
Ericsson Broadcast
and Media
Pizza Hut
Corporate Office

AWS re:Invent


Speedometer
Odometer
Steering Angle
Window Sensor
GPS
RPMs
Fuel Level
Accelerometer
Brake Sensor
Gas Pedal
Tire Pressure Sensor

48 average driving minutes per day

500+ unique data points
generated every 200 milliseconds
captured in real-time

7.2 million data points
per connected vehicle
per day

Data services that drive customer satisfaction

Data insights that improve our products

AWS re:Invent

2021: vehicles will be able to upload video in addition to telemetry

CARS FOR GOOD

Can we use video from vehicle cameras
to do positive things for the World?

CARS FOR GOOD

Detect and alert other drivers on dangerous weather conditions

CARS FOR GOOD

Keep our environment clean

CARS FOR GOOD

Reduce graffiti

CARS FOR GOOD

Automatically improve maps in real-time

CARS FOR GOOD

4 People

Optimize public transportation routes

CARS FOR GOOD

4 Occupants

Optimize Emergency services

CARS FOR GOOD

Improve safety outside the car

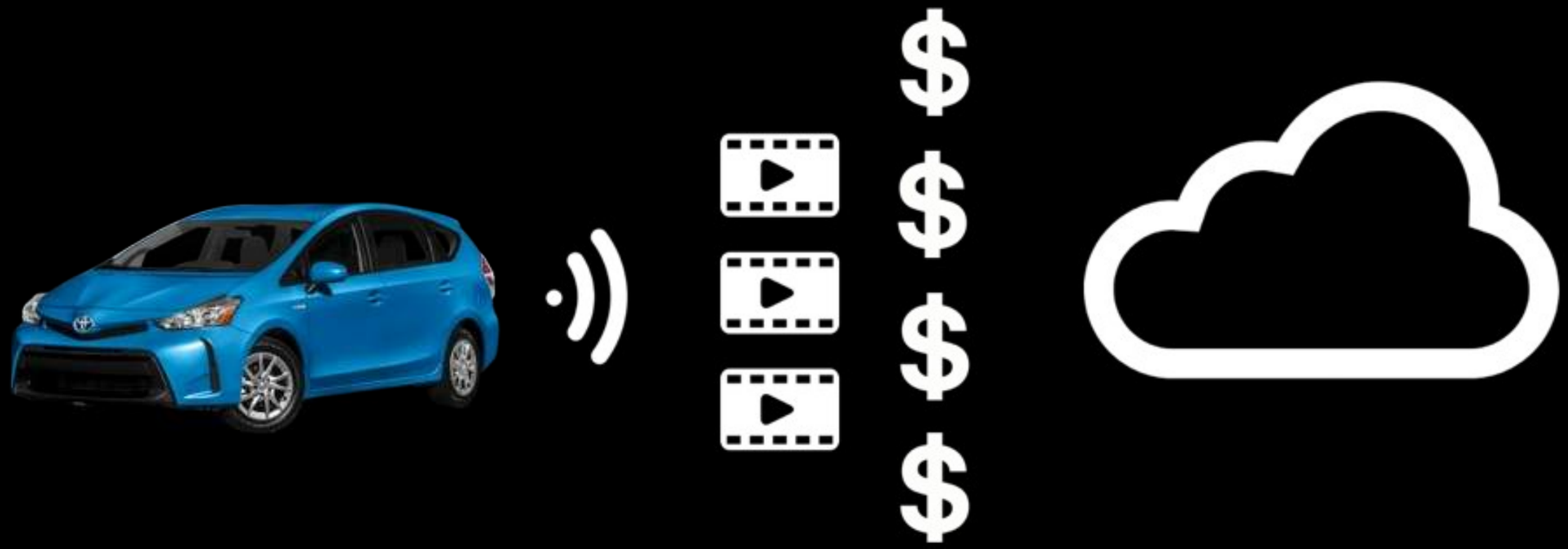
CARS FOR GOOD

Improve safety inside the car

We can,
but…

CARS FOR GOOD

Transferring to the cloud HD video from multiple cameras
is too expensive to support real-time Computer Vision services



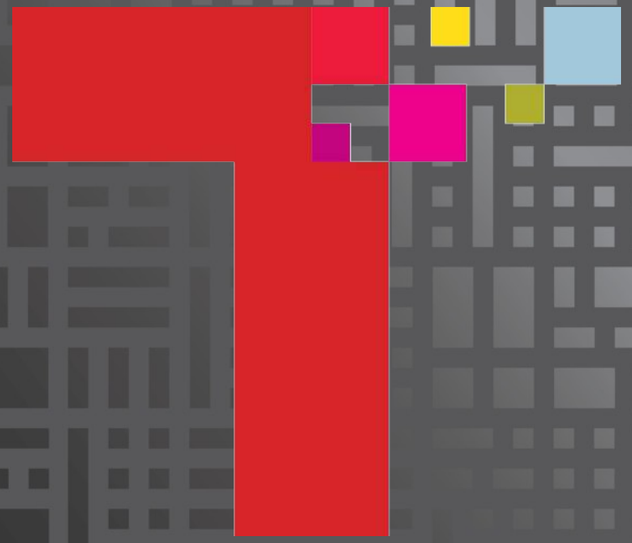
7.5 million vehicles,
1 camera in each vehicle,
1 minute of HD video from each vehicle

The cost of uploading is…

Billions of dollars per year

Innovation

AWS re:Invent



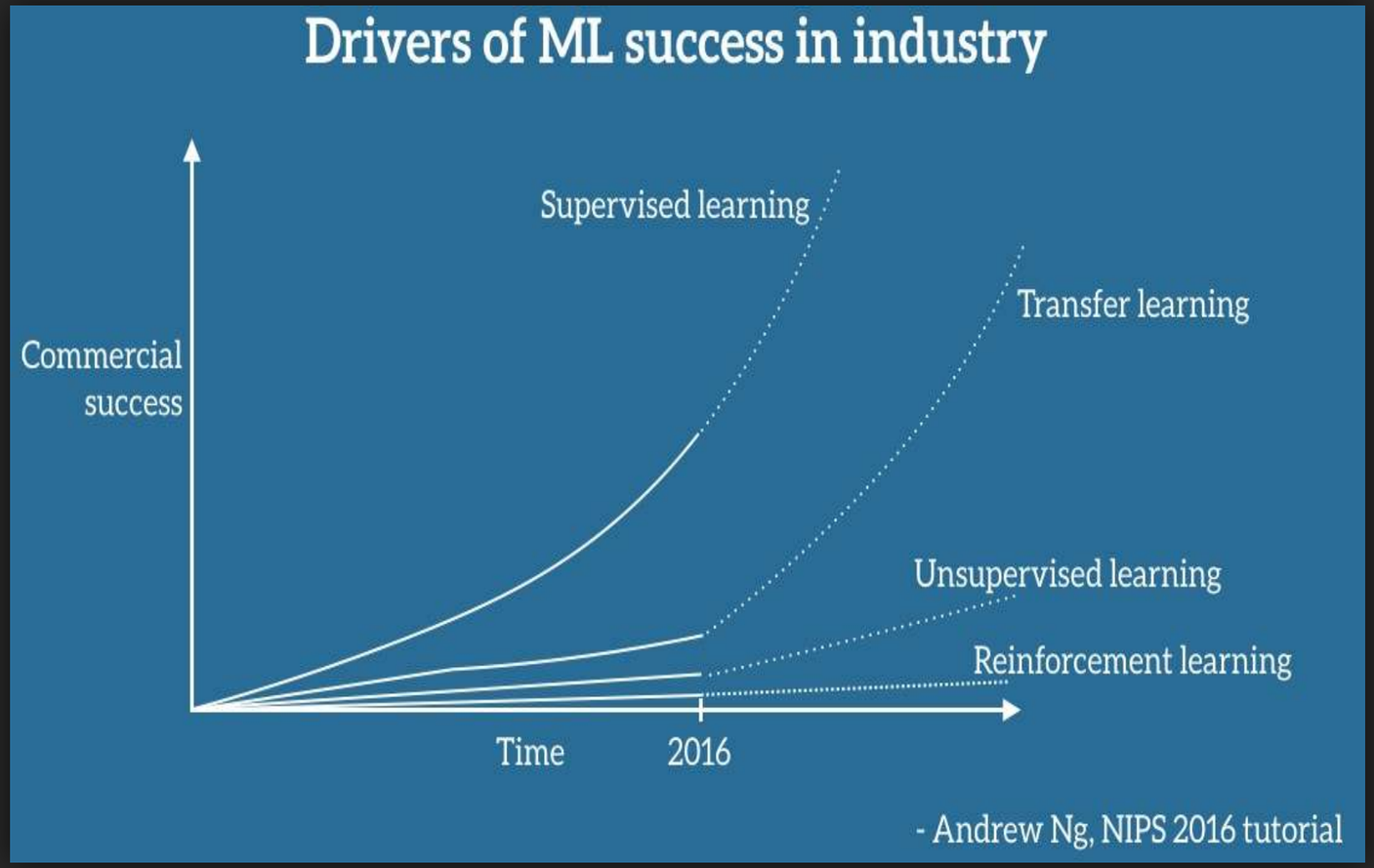
AWS re:Invent

Can we predict driver distraction?

Training dataset

- Multiple drivers performing 4 complex actions
- Labelled as: Normal Driving, Texting, Looking Away, On the Phone

Transfer Learning



"Don't be a Hero"

Instead of rolling your own network architecture, look at whatever currently works best on ImageNet, download a pretrained model and finetune it on your data.

You should rarely ever have to design a model from scratch or even train from scratch or.

ResNet-152

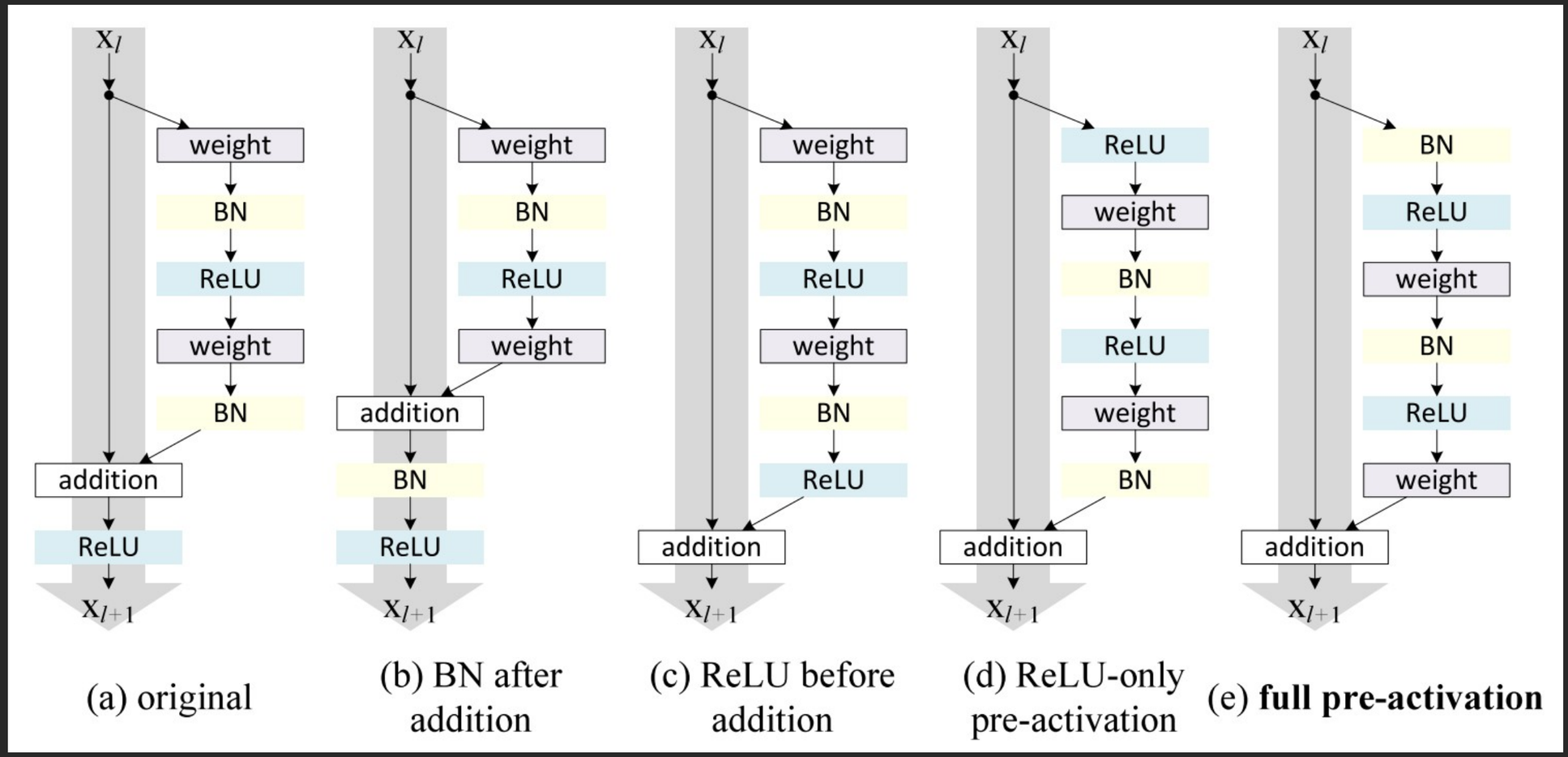


Image Augmentation

We performed random rotations, random crop, RGB mean centering and translations on each image sample.

The augmented dataset makes the resulting model more robust and less prone to overfitting.

Pre-processing and hyper parameter tuning

Optimized Hyper Parameters

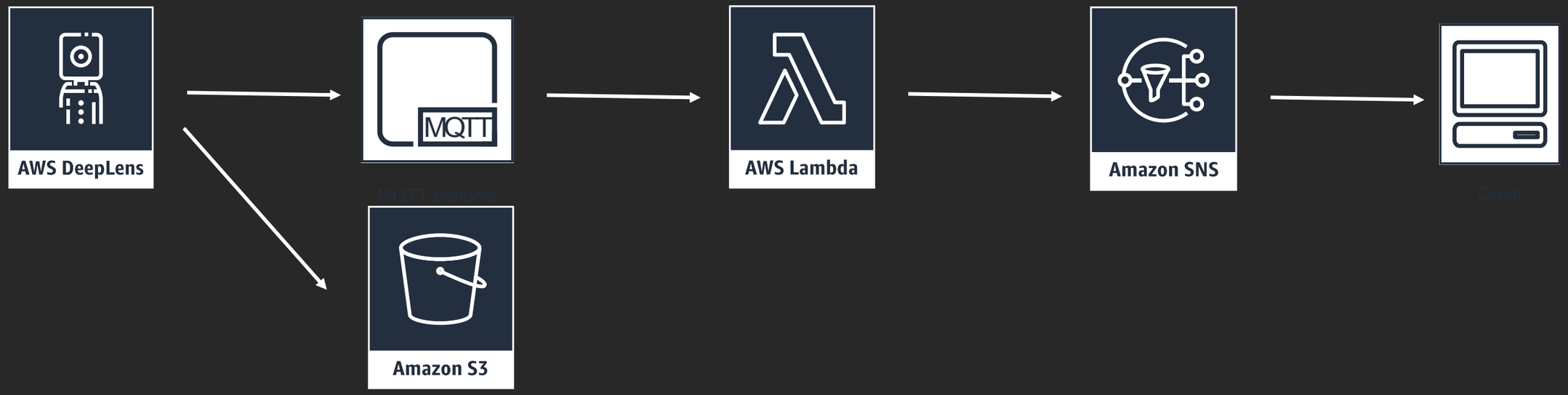
- Epochs : 50
- Image Shape : 3,224,224
- Learning_rate : 0.001
- Lr_scheduler_factor : 0.0001
- Lr_scheduler_step : 5
- Mini_batch_size : 64
- Num_Layers : 152
- Num_training_samples : 16180
- Optimizer : sgd

Data preprocessing : Down sample the images from 640x480 to 224x224 so that their dimension matches the one used by the pre-trained model.

Fine-tune pretrained network : after multiple iterations on training sample data to find hyper parameters, train using the pre-learned weights and our image data.

Deep Learning at the Edge

- Train on Amazon Sagemaker: Amazon EC2 P2 instances, 8 NVIDIA Tesla K80 GPUs
- Deploy model artifacts on AWS DeepLens with AWS Greengrass
- Use the Intel Model Optimizer for faster GPU inferences
- Upload predicted images to Amazon S3 in real-time
- Send inference results to AWS IoT

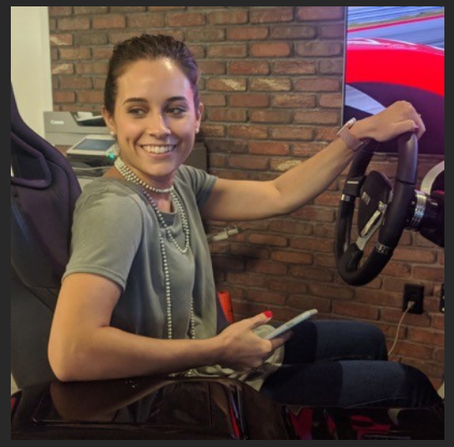


Video

AWS re:Invent

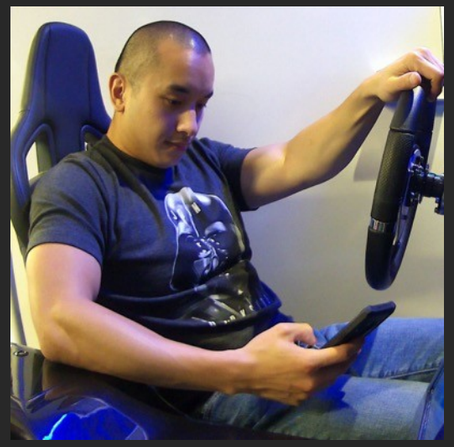
Our volunteers

Dany Benjamin



Tori Salido

Steve King



Blake Wong

Trish Batson

Will Dombrowski

Toshi Tanaka

Michelle Haught

AWS re:Invent

Lessons Learnt

Use of other state-of-the-art models at the Edge?

Ensemble is a very powerful tool but performance on constrained devices is a challenge.

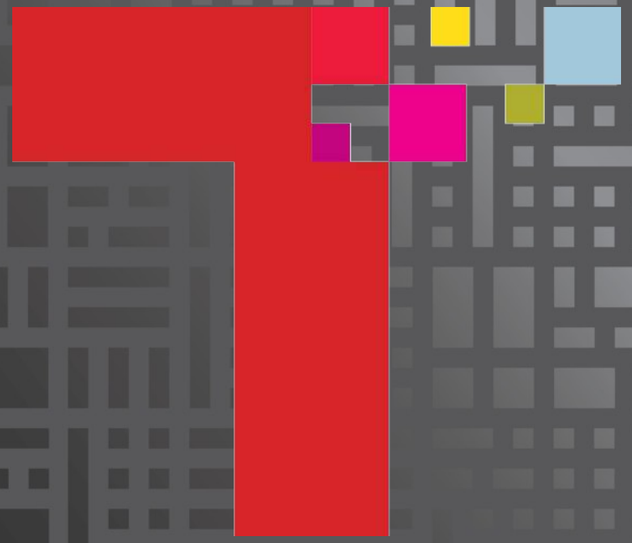
Image Preprocessing is the key.

Focus on the part of the image that contains the driver and cut out the useless part of the image.

Clipping does not work.

Clipping the loss with various threshold does not help. The reason is that the model has different level of confidence on different classes .

How often you run the inference and how you interpret it is more important than model metrics like accuracy.



AWS re:Invent

Optimizing performance for inference at the edge

- Constrained devices require smaller and faster models
  - Reduce memory usage
  - Reduce computational complexity
  - Reduce power consumption

- Code optimization: Intel Inference Engine, Intel MKL, NNPACK
- Quantization / mixed mode training: smaller weights and activations
- Pruning: remove useless connections
- Compression: encode weights

- Some of these techniques are available in Apache MXNet or TensorFlow

What if local inference is not possible?

AWS re:Invent

Device can support HTTPS

Invoke an Amazon SageMaker endpoint

- Train a model in SageMaker (or import your own).
- Deploy it to a prediction endpoint.
- Device invokes the HTTPS endpoint and receives prediction results.

Best when
Devices are not powerful enough for local inference.
Models can’t be easily deployed to devices.
Additional cloud-based data is required for prediction.
Prediction activity must be centralized.

Requirements
Network is available.
Devices support HTTP.

Device cannot support HTTPS

Invoke an AWS Lambda function through AWS IoT Core

- Train a model in Amazon SageMaker (or import your own).
- Deploy it to a prediction endpoint.
- Write an AWS Lambda function performing prediction.
- Device sends an MQTT message to AWS IoT Core, triggers the function and receives prediction results.

Best when
Devices can support neither HTTP nor local inference (e.g. Arduino).
Costs must be kept as low as possible.

Requirements
Network is available (MQTT is less demanding than HTTP).
Devices are provisioned in AWS IoT Core (certificate, keys).

Getting started

AWS re:Invent

Resources

https://ml.aws/

https://aws.amazon.com/sagemaker
https://aws.amazon.com/machine-learning/amis/
https://aws.amazon.com/greengrass
https://aws.amazon.com/greengrass/ml
https://aws.amazon.com/deeplens

https://medium.com/@julsimon

Please complete the session survey in the mobile app.

Thank you!

Julien Simon
Principal Tech. Evangelist, AI/ML
Amazon Web Services
@julsimon

Brian Kursar
Chief Data Scientist
Toyota Connected Data Services
Brian.Kursar@toyotaconnected.com
@briankursar

Nimish Amlathe
Data Scientist
Toyota Connected Data Services
Nimish.Amlathe@toyotaconnected.com

AWS re:Invent