

JETSON ORIENTATION

A.I. LEARNING LAB

Dustin Franklin
Developer Relations



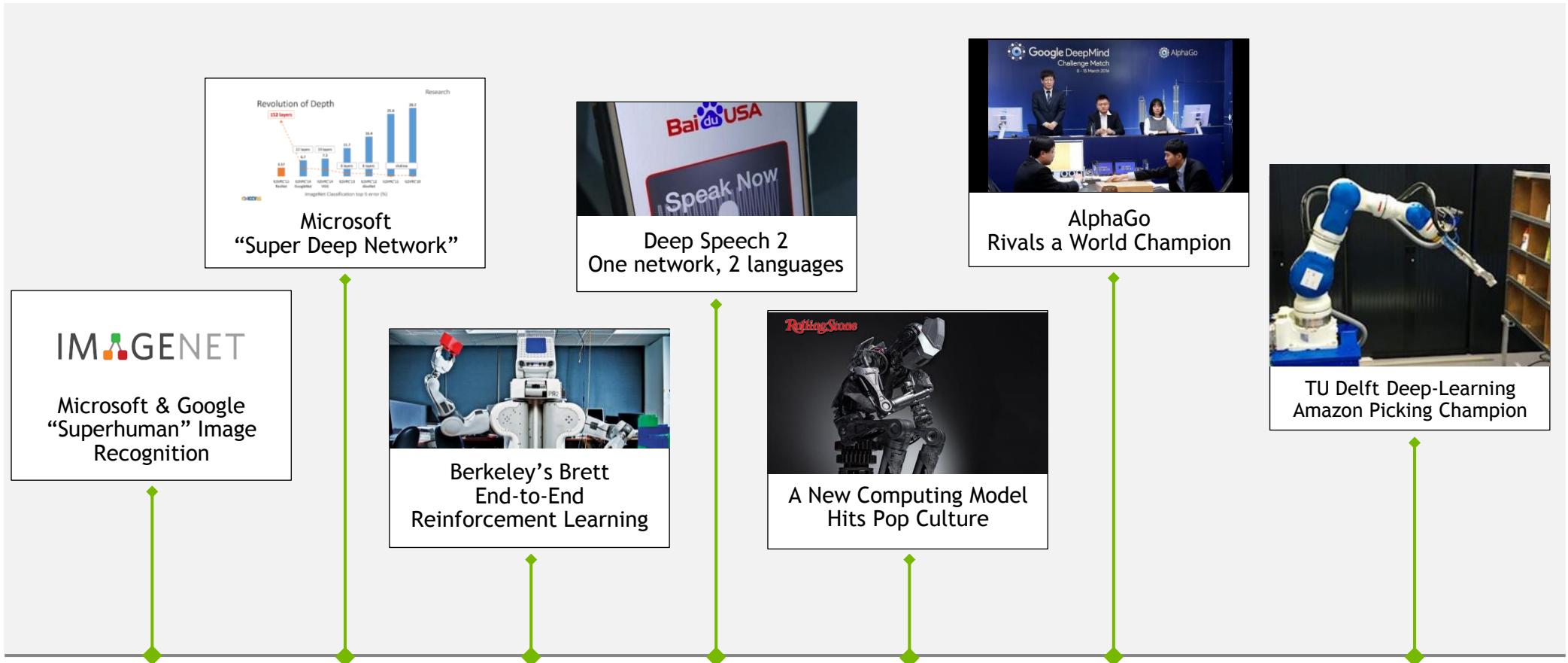
FIRST Robotics Day
Thursday August 4, 2016

Workshop Agenda

Topic:

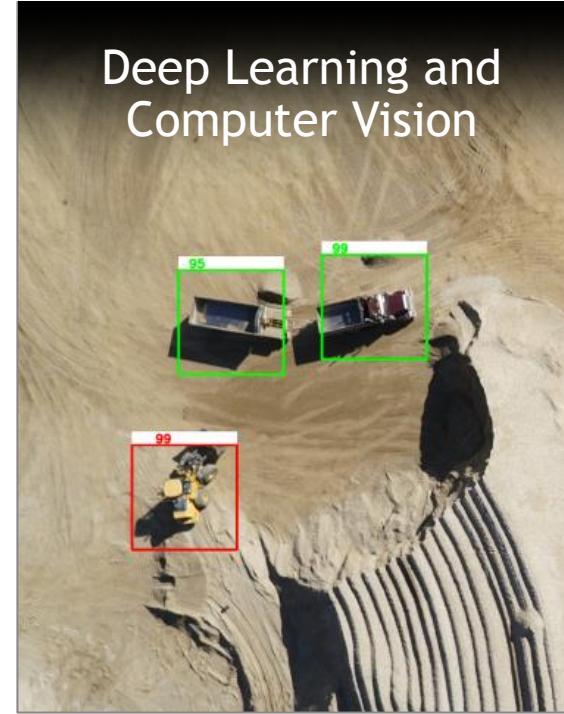
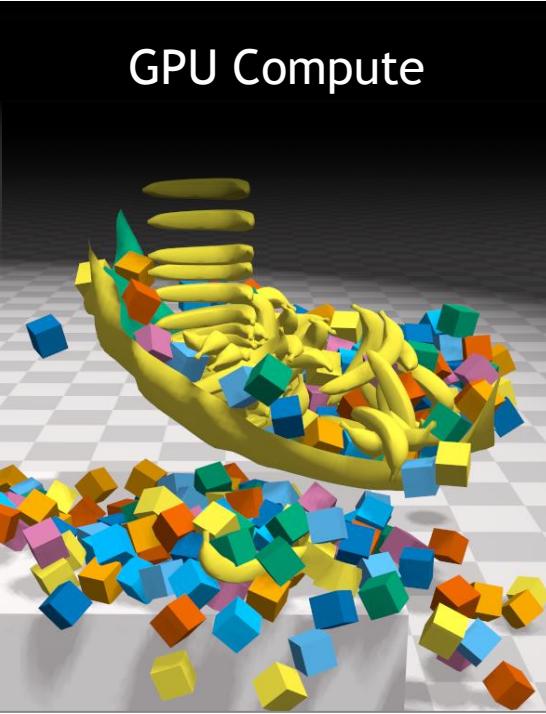
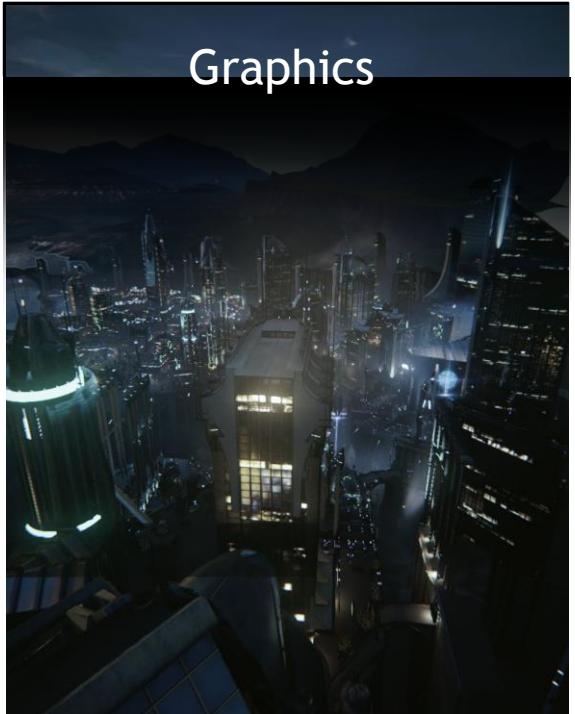
- Background on Jetson TX1
 - Fundamentals of Deep Learning
 - Deploying Realtime Networks with GPU Inference Engine
-
- Reinforcement Learning
 - Why?
 - How?
 - Conclusion / Packing Up
-

MILESTONES IN AI





NVIDIA GPU: MORE THAN GRAPHICS

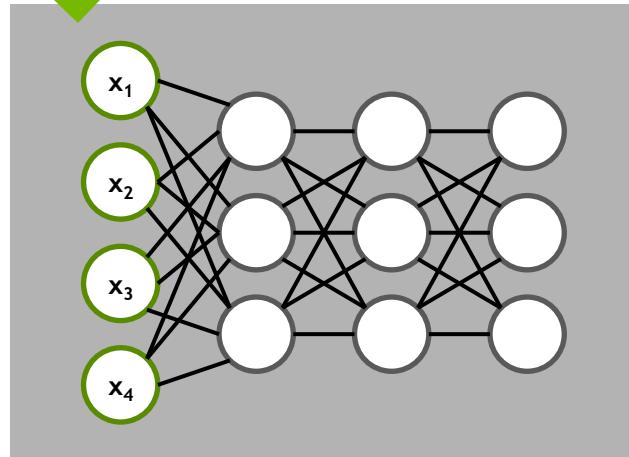


AT THE FRONTIER OF AUTONOMOUS MACHINES

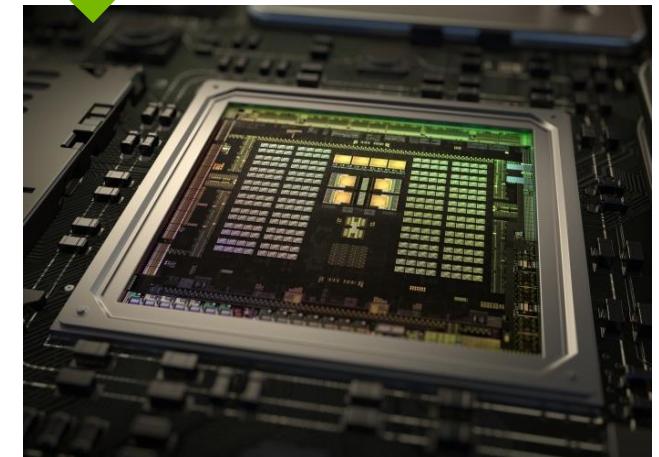
New use cases
demand autonomy

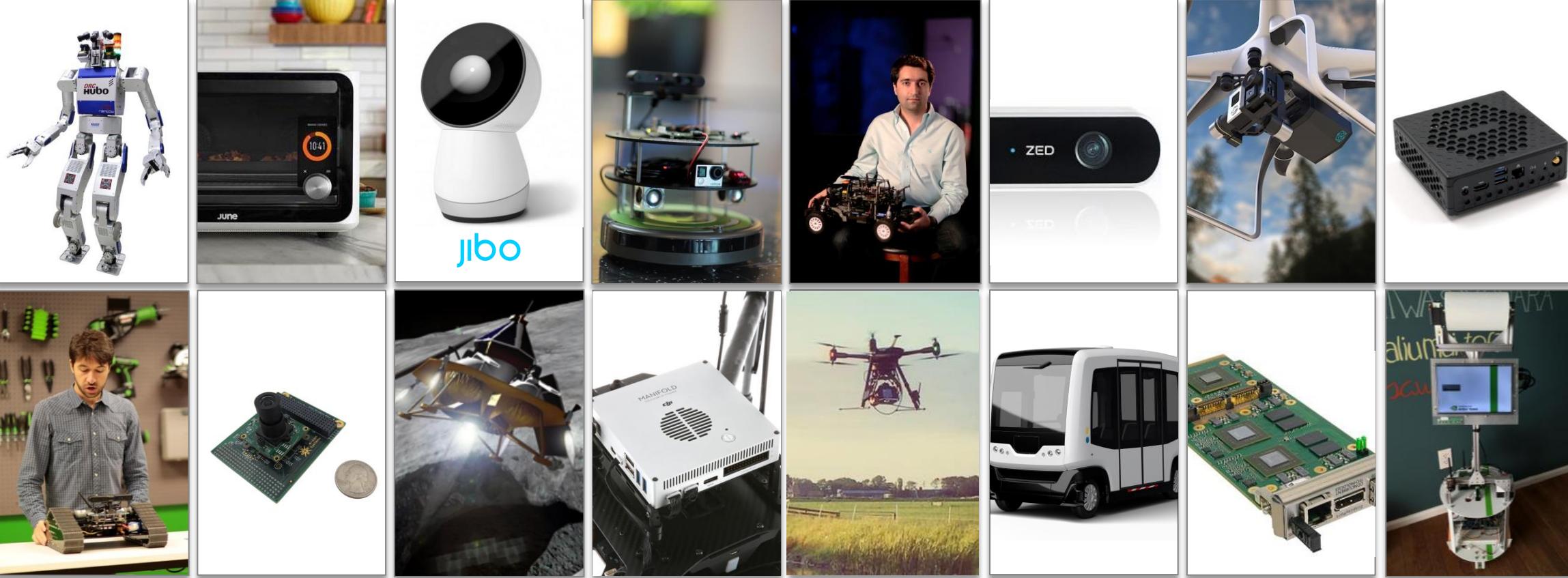


Onboard sensing and
deep learning,
enable autonomy



GPUs deliver superior
performance and
efficiency





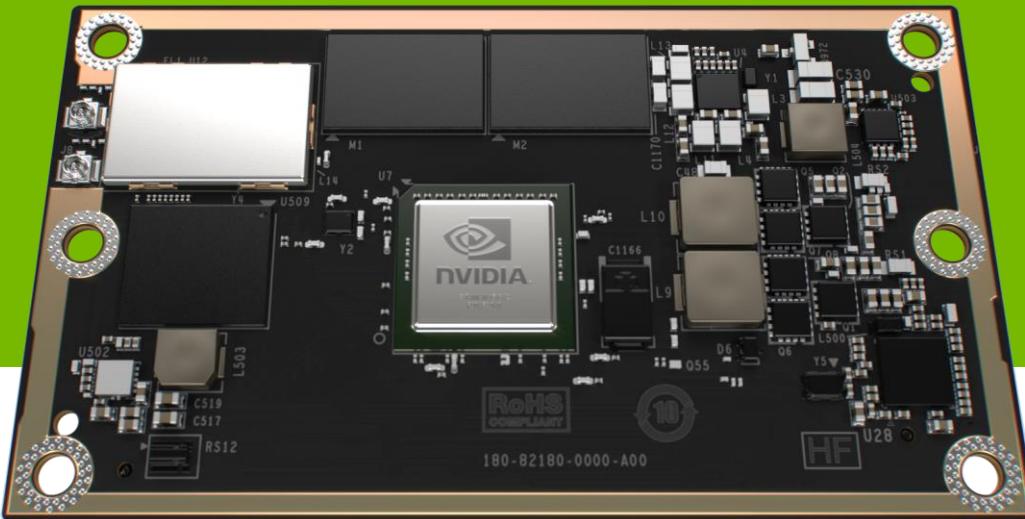
JETSON AT THE LEADING EDGE

Powering the Next Generation of Autonomous Machines

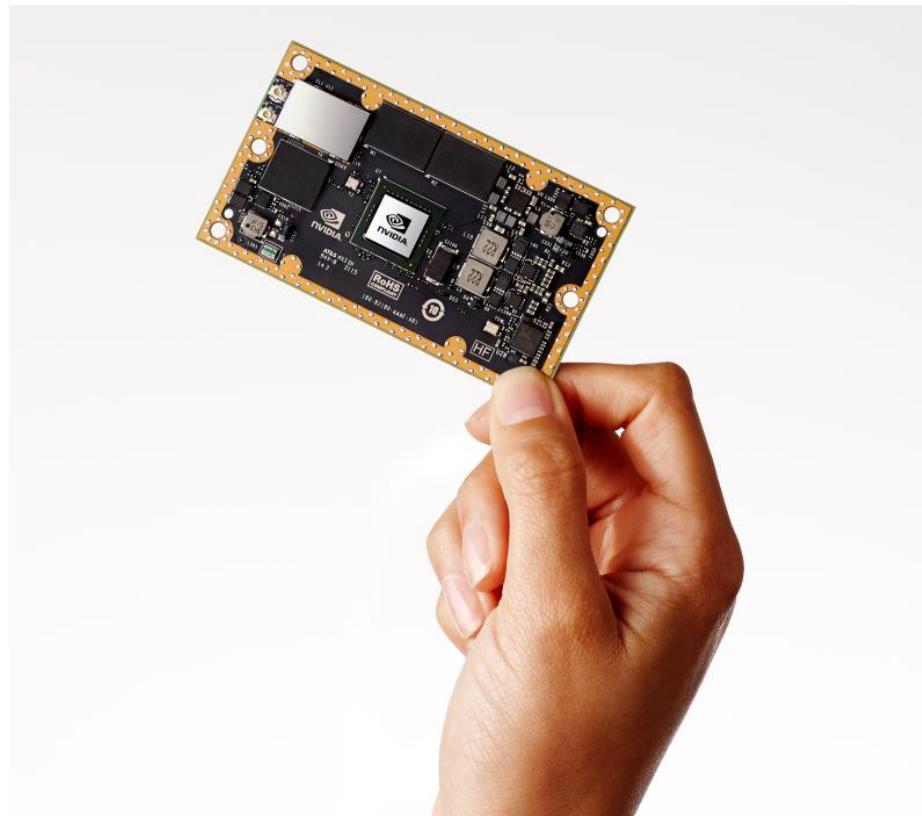
Jetson TX1

Supercomputer on a Module

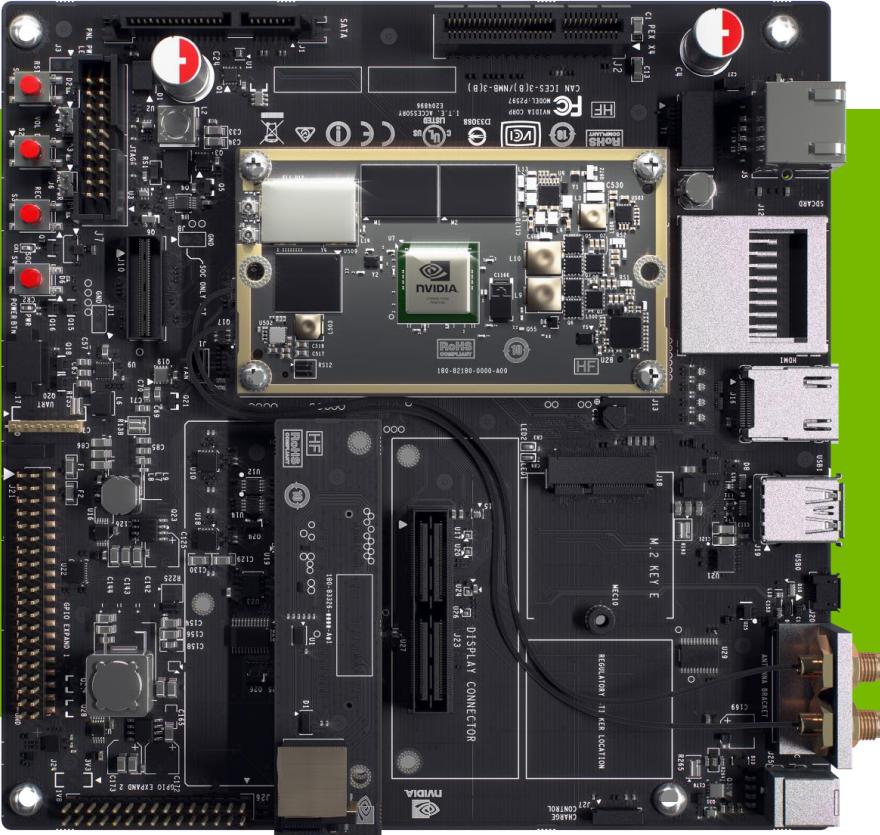
Unmatched performance under 10W
Advanced tech for autonomous machines
Smaller than a credit card



System on Module



JETSON TX1	
GPU	1 TFLOP/s 256-core Maxwell
CPU	4x 64-bit ARM A57 CPUs 1.6 GHz
Memory	4 GB LPDDR4 25.6 GB/s
Video decode	4K 60Hz H.264
Video encode	4K 30Hz H.264
CSI	Up to 6 cameras 1400 Mpix/s
Display	2x DSI, 1x eDP 1.4, 1x DP 1.2/HDMI
Wi-Fi	802.11 2x2 ac
Networking	1 Gigabit Ethernet
PCI-E	Gen 2 1x1 + 1x4
Storage	16 GB eMMC, SDIO, SATA
Other	3x UART, 3x SPI, 4x I2C, 4x I2S, GPIOs
Power	10-15W, 6.6V-19.5VDC
Size	50mm x 87mm



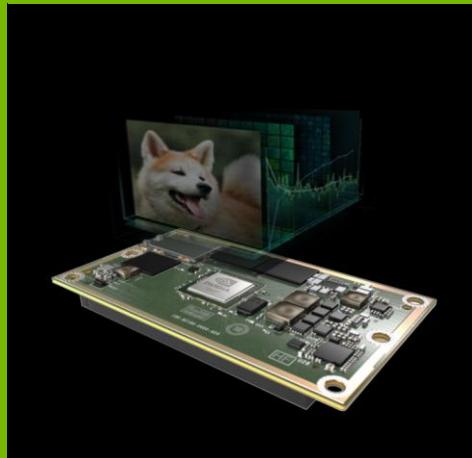
Jetson TX1 Developer Kit

Jetson TX1
Developer Board
5MP Camera

NVIDIA JETPACK



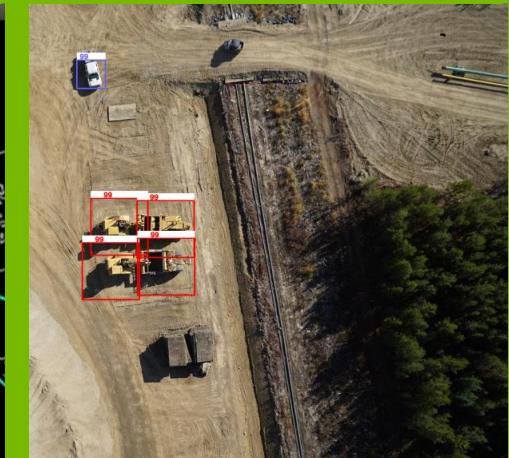
Deep Learning SDK



DIGITS Workflow



VisionWorks



Jetson Multimedia SDK

and other technologies:

CUDA, Linux4Tegra, NSIGHT EE, OpenCV4Tegra,
OpenGL, Vulkan, System Trace, Visual Profiler, Ubuntu 14.04

JETSON SDK: THE DETAILS

Vertically Integrated Packages

Machine Learning



Caffe
theano



Vision



Compute (CUDA)



cuFFT CUDA Math Library
cuBLAS NPP cuSPARSE
cuRAND Thrust cuSolver

Graphics



Linux for Tegra



PULSEAUDIO



V4L2
libjpeg

Tools



Source code editor

Debugger

Profiler

System Trace

NVTX

NVIDIA Tools eXtension

Jetson TX1

VISIONWORKS™

CUDA-accelerated Computer Vision Toolkit

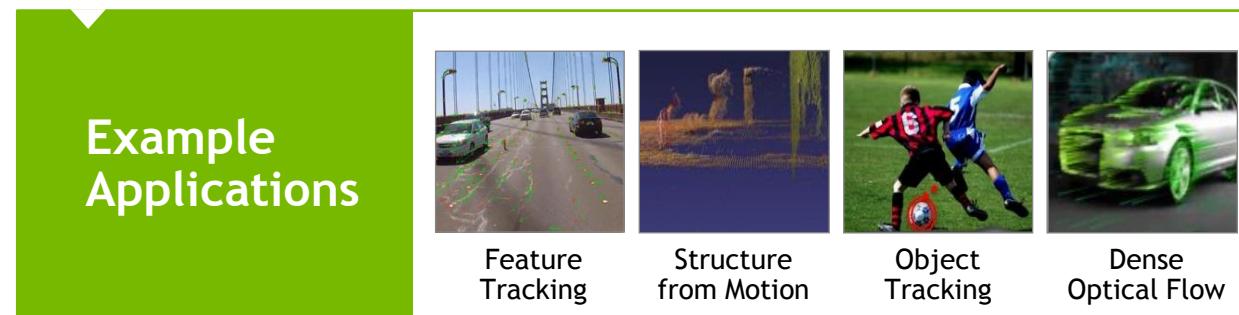
- Full OpenVX 1.1 implementation
- Easy integration with existing CV pipelines
- Custom extensions



Robotics

Augmented Reality

Drones



VisionWorks™ API + FrameWorks			
IMAGE ARITHMETIC Absolute Difference Accumulate Image Accumulate Squared Accumulate Weighted Add / Subtract / Multiply Channel Combine Channel Extract	GEOMETRIC TRANSFORMS Affine Warp + Perspective Warp Flip Image Gaussian Pyramid Remap Scale Image	Features Canny Edge Detector Fast Corners + Fast Track Harris Corners + Harris Track Hough Circles Hough Lines	

Comprehensive Developer Platform

- Jetpack SDK
- Libraries
- Developer tools
- Design collateral
- Developer Forum
- Training and Tutorials
- Ecosystem

The screenshot shows the NVIDIA Embedded Computing website. At the top, there's a navigation bar with links for Register, Develop, Downloads, Community, Learn, Buy, and a search bar. Below the navigation is a banner with the text "Embedded Computing" and a green circular graphic. Underneath the banner, the page title is "Home > Embedded Computing". There are tabs for View, Edit, Outline, Revisions, and Grant, with "View" being the active tab. The main content area features a heading "Meet the Jetson Embedded Platform" and a paragraph describing the NVIDIA Jetson platform. It highlights its use in Computer Vision projects like drones and autonomous systems, and its appeal to independent developers and hobbyists. A link "Learn more about The Jetson Embedded Platform" is provided. To the right of the text is a photograph of the NVIDIA Jetson TX1 module. At the bottom of the page are six green buttons with white icons: REGISTER (with a code tag icon), DEVELOP (with a gear icon), DOWNLOAD (with a download arrow icon), COMMUNITY (with a globe icon), LEARN (with a notepad icon), and BUY (with a shopping cart icon).

<http://developer.nvidia.com/embedded-computing>

GETTING STARTED

JETSON COMMUNITY

Developer Forums devtalk.nvidia.com

NVIDIA ACCELERATED COMPUTING Downloads Training Ecosystem Forums

Home > CUDA ZONE > Forums > Accelerated Computing > Jetson & Embedded Systems > Jetson TX1

+ Create Topic 

Jetson TX1

	Activity	Started By	Last Comment
JetPack 2.2 and L4T R24.1 for Jetson TX1 released	23 Replies 1,978 Views	dusty_nv 4 weeks ago	zain 7 hours ago
Building Autonomous Machines: NVIDIA webinar this Thursday 7/14	0 Replies 61 Views	dusty_nv 2 days ago	dusty_nv 2 days ago
EDU discount for JTX1	144 Replies 15,171 Views	dusty_nv 7 months ago	dusty_nv 2 days ago
MIPI DSI/CSI Design and Develop Guide	7 Replies 727 Views	Trumany 1 month ago	Trumany 1 week ago
Deep Learning Inference: Performance validation on TX1	6 Replies 1,295 Views	jachen 2 months ago	Take!! 4 weeks ago
Jetson TX1	211 Replies 167,565 Views	NVD 8 months ago	Jijikos 2 months ago
Create a GL 3+ Context without X	10 Replies 130 Views	slembcke 2 weeks ago	slembcke 28 minutes ago
Trouble using Jetson TX1 Developer Kit SATA with Samsung 850 EVO SSD	1 Replies 18 Views	DanielTrebien 2 hours ago	linuxdev 1 hour ago
PCI-E USB 3.0 Card for TX1	1 Replies 12 Views	arane 3 hours ago	linuxdev 2 hours ago
Installing Jetpack L4T on Tegra TX1	8 Replies 90 Views	kshitijsrivastava 5 days ago	linuxdev 2 hours ago
Programming in newlisp	1 Replies 0 Views	grefew 3 hours ago	linuxdev 2 hours ago
Jetson TX1 Module only?	1 Replies 46 Views	kirke 3 hours ago	AlexP312 2 hours ago
PCIe DMA doesn't work for L4T 24.1	7 Replies 168 Views	yahoo2016 6 days ago	yahoo2016 2 hours ago
External Consultants sought	1 Replies 0 Views	maykel 7 days ago	AlexP312 2 hours ago

eLinux Wiki eLinux.org/Jetson_TX1

Dusty-nv 1 Talk Preferences Watchlist New messages (none) Contributions Log out

Page Discussion Read Edit View history More Search [dismiss]



Jetson TX1

NVIDIA's Jetson TX1 is an embedded system-on-module (SoM) with quad-core ARM Cortex-A57, 4GB LPDDR4 and integrated 256-core Maxwell GPU. Useful for deploying computer vision and deep learning, Jetson TX1 runs Linux and provides 1TFLOPS of FP16 compute performance in 10 watts of power.

Contents [hide]

- 1 Jetson TX1 Module
 - 1.1 Processing Components
 - 1.2 Ports & Peripherals
 - 1.3 Form-Factor
 - 1.4 Software Support
- 2 Jetson TX1 Developer Kit
 - 2.1 What's Included
 - 2.2 Getting Started
 - 2.3 Availability
- 3 Platform Documentation
 - 4 Guides and Tutorials
 - 4.1 System Concepts
 - 4.2 Computer Vision
 - 4.3 Deep Learning
 - 5 Ecosystem Products
 - 6 Getting Help

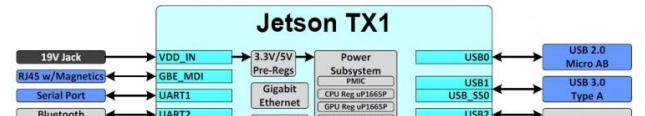
Jetson TX1 Module [edit]

The Jetson TX1 module contains all the active processing components. The ports are broken out through a carrier board. Below is a partial list of the module's features. Please see the [Module Datasheet](#) for the complete specifications.

Processing Components [edit]

- quad-core ARM Cortex-A57
- 256-core Maxwell GPU
- 4GB LPDDR4
- 16GB eMMC

Jetson TX1



19V Jack → VDD_IN → 3.3V/5V Pre-Reg → Power Subsystem PMIC → CPU Reg up1665P → GPU Reg up1665P → GBE MDI → RJ45 w/Magnetics → Gigabit Ethernet → Serial Port → UART1 → UART2 → Bluetooth → USB0 → USB1 → USB SSO → USB2 → USB 2.0 Micro AB → USB 3.0 Type A

THE PERIPHERALS JETSON CONNECTS WITH

including Community Contributions

- Infrared devices:
 - SICK LIDAR (LMS 200); Hokuyo; rpLIDAR
 - Asus Xtion Pro Live (PrimeSense)
 - Intel RealSense (mult. generations)
- Stereo and color cameras:
 - StereoLabs Zed (consumer-oriented)
 - Point Grey Research USB3 and GigE
 - e-con Systems CSI-MIPI Cameras with external ISP



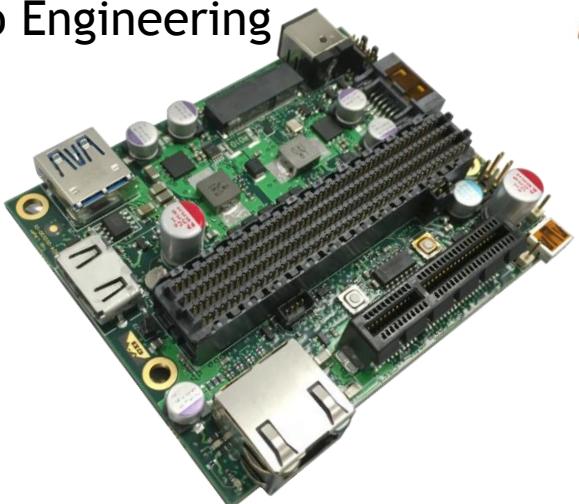
DEPLOYING JETSON TX1

Modular Ecosystem

- Auvidea J120



- Colorado Engineering TX1-SOM

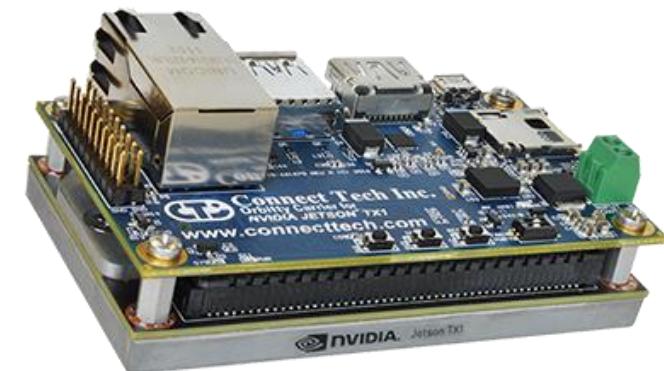


TX1 MODULE

- ConnectTech Orbitty



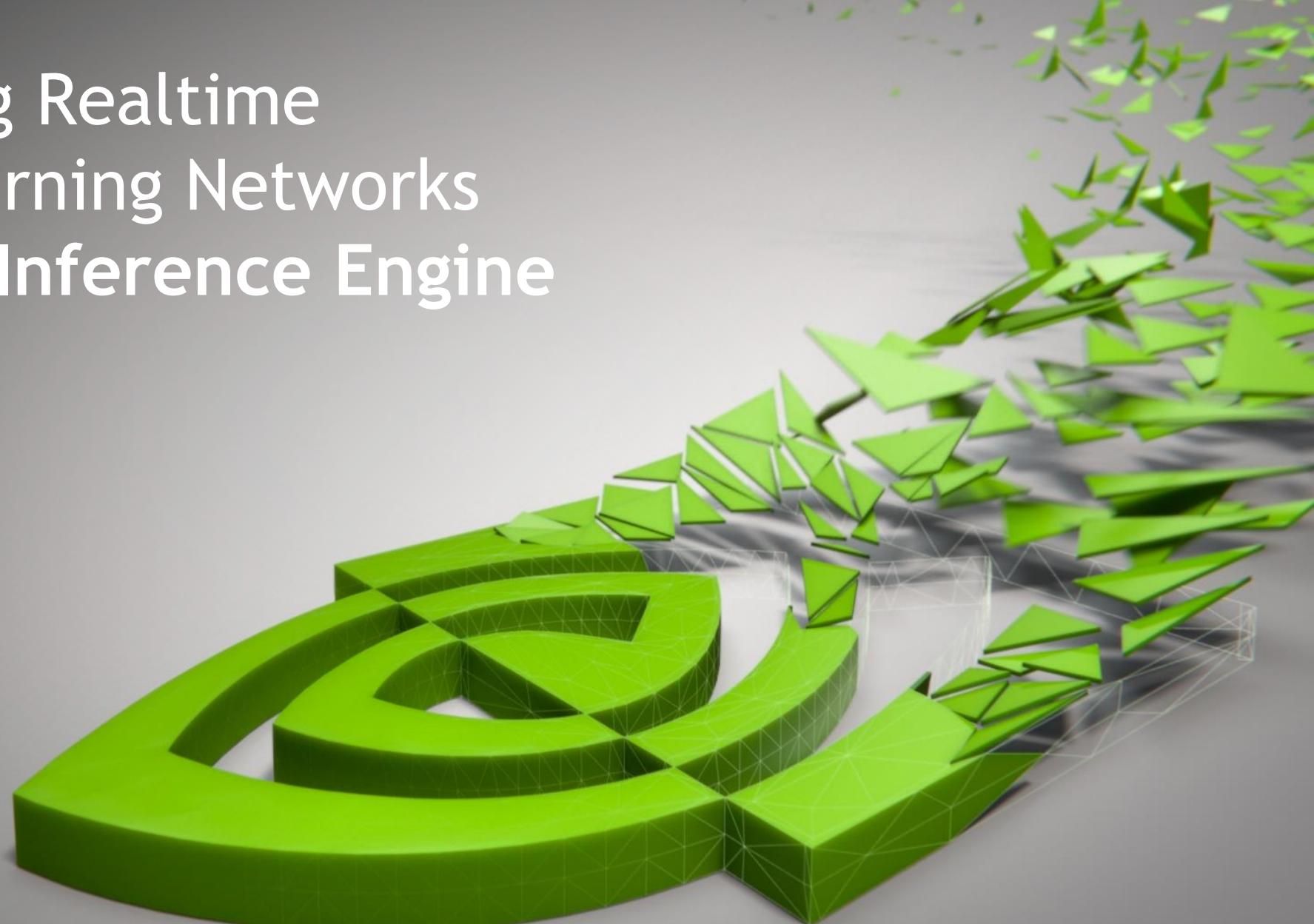
Connect Tech Inc.
Embedded Computing Experts



- ConnectTech Rosie

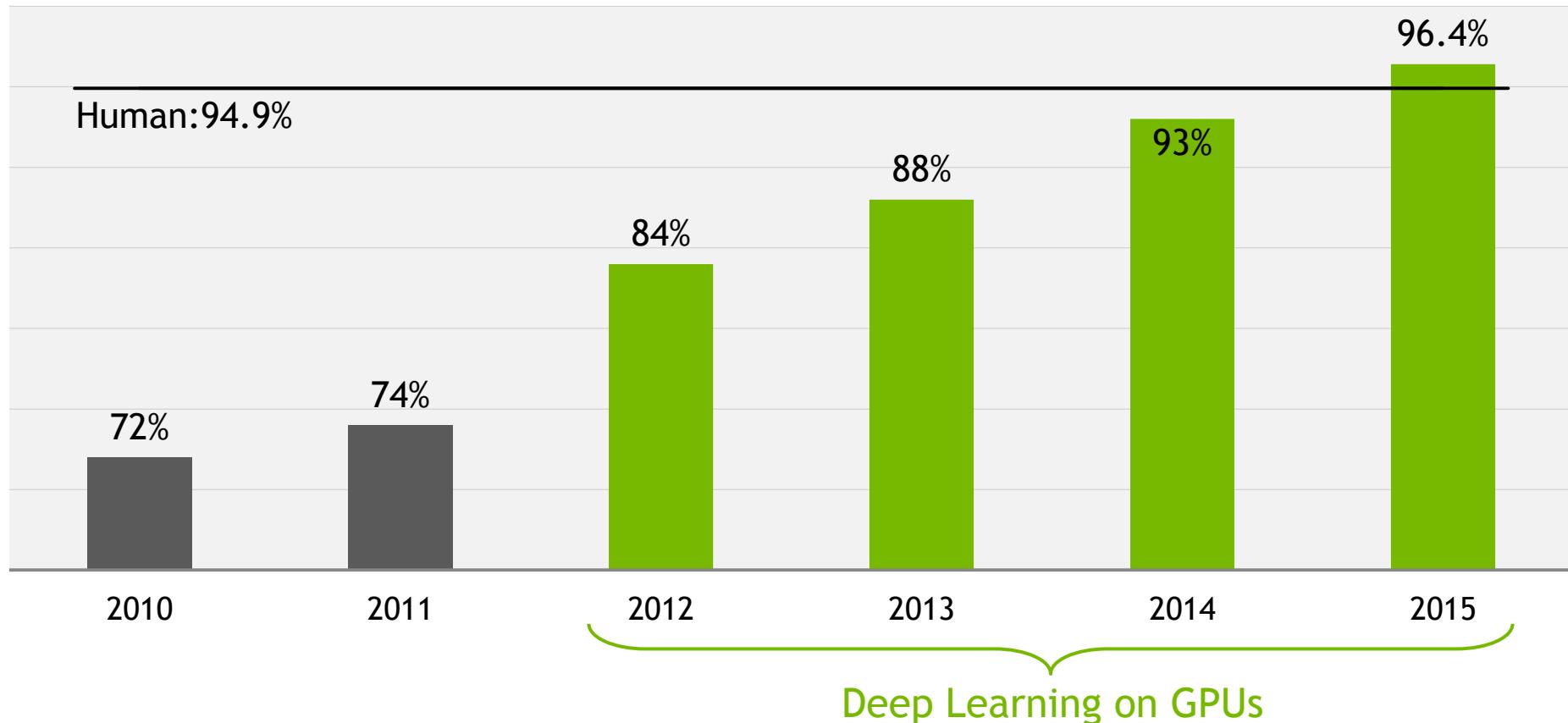


Deploying Realtime Deep-Learning Networks with **GPU Inference Engine**



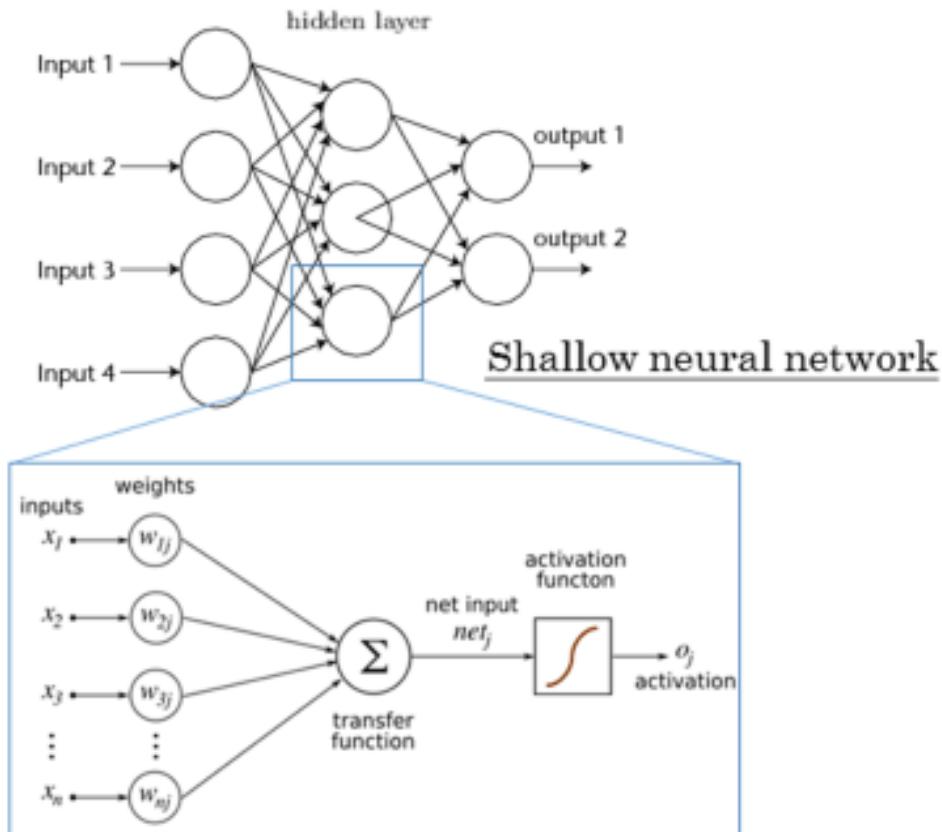
HOW FAR HAS AUTONOMY COME?

ImageNet classification accuracy

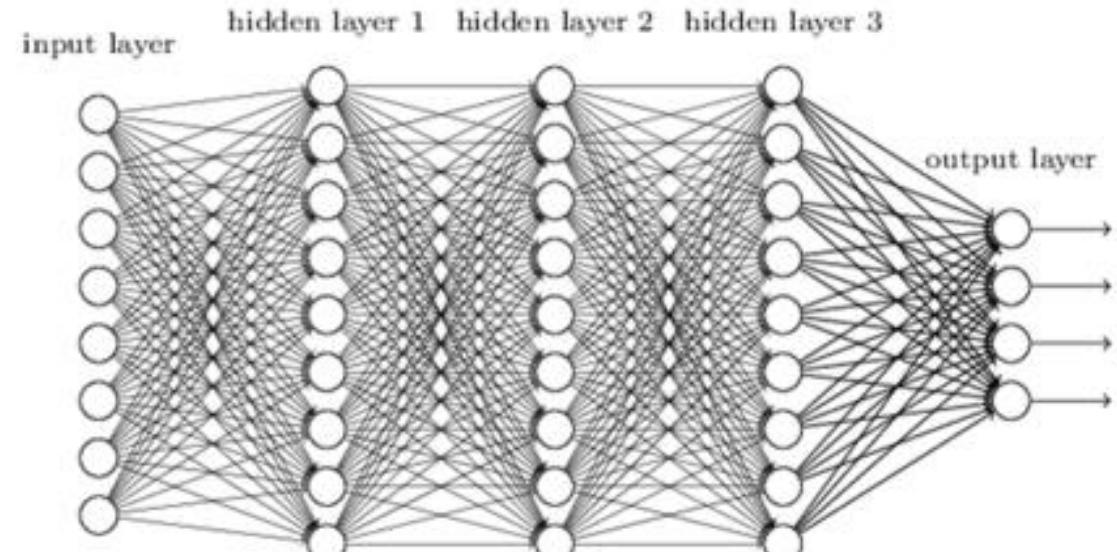


DEEP LEARNING

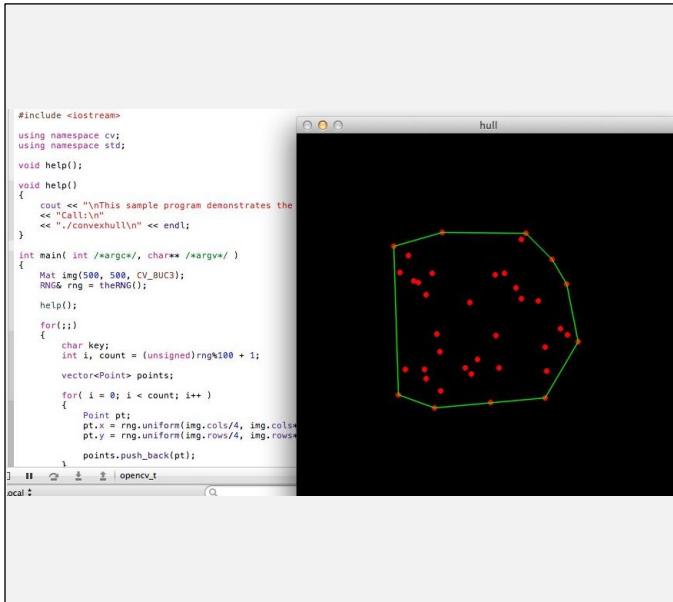
What's Different?



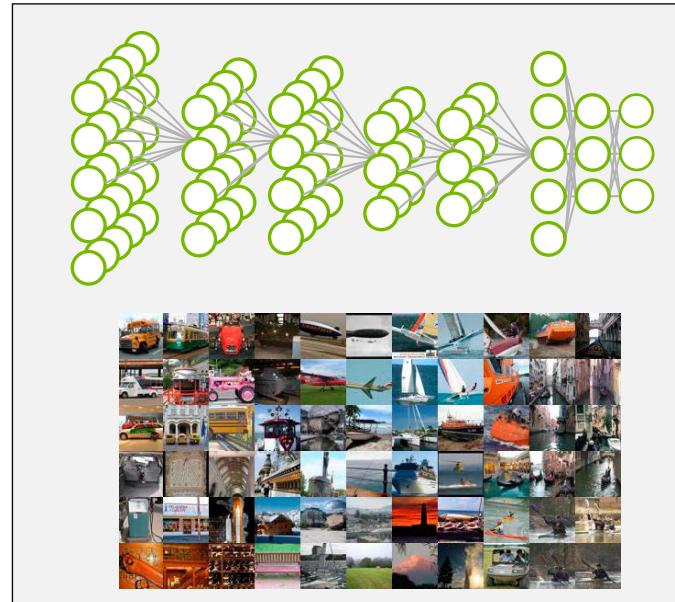
Deep neural network



A NEW COMPUTING MODEL



Traditional Computer Vision
Experts + Time



Deep Learning
DNN + Data + HPC

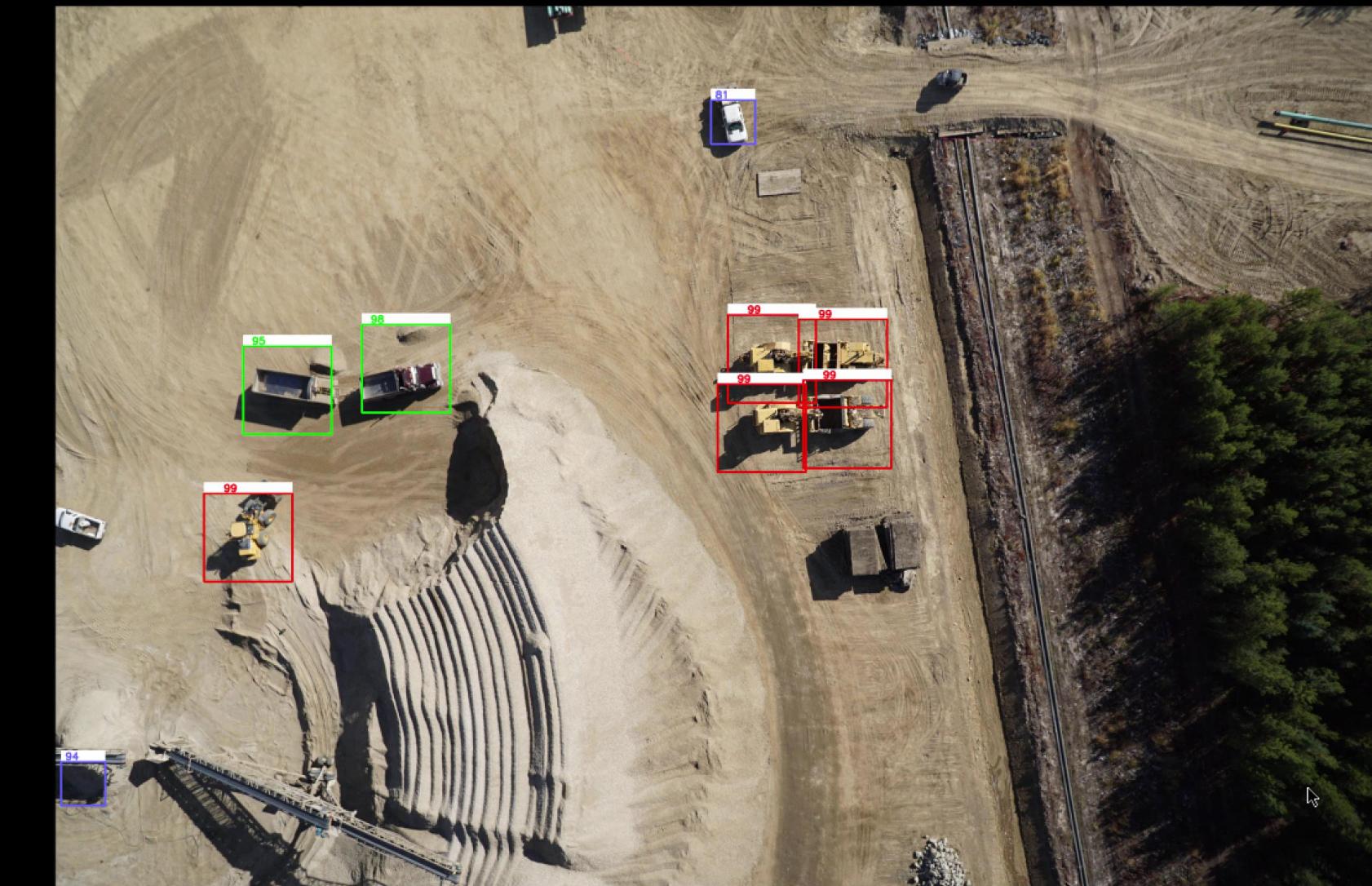


Autonomous Machines
Onboard Intelligence



Light vehicles detected: 2
Delivery vehicles detected: 3
Construction vehicles detected: 5

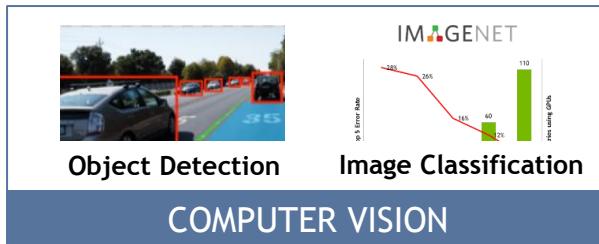
Avg inference time per frame: 400ms



GPU: ON
Deep Learning: ON

POWERING THE DEEP LEARNING ECOSYSTEM

NVIDIA SDK Accelerates Every Major Framework



Object Detection

IMAGENET
Image Classification

COMPUTER VISION



Voice Recognition

Language Translation



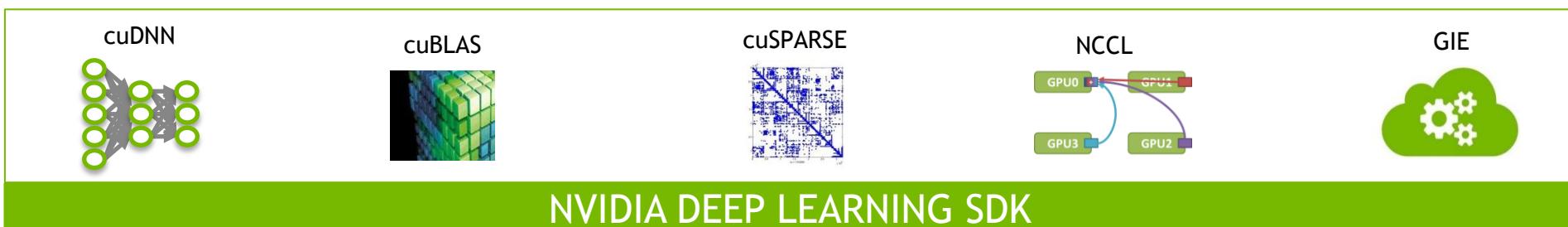
Recommendation
Engines

Sentiment Analysis

NATURAL LANGUAGE PROCESSING

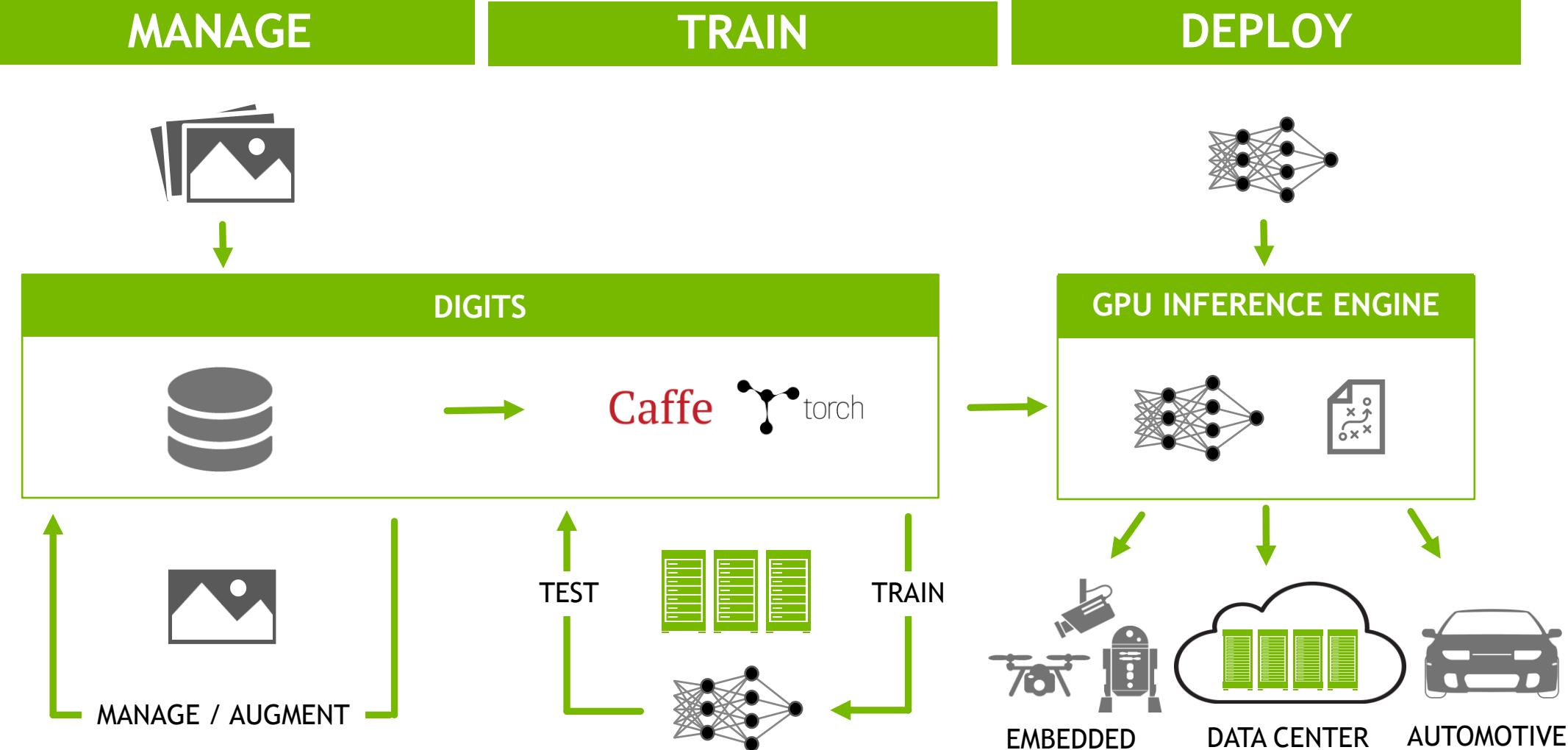


DEEP LEARNING FRAMEWORKS



NVIDIA DEEP LEARNING SDK

A COMPLETE COMPUTE PLATFORM



NVIDIA DIGITS

Interactive Deep Learning GPU Training System

Process Data

DIGITS Image Classification Dataset
voc_cropped@256x256

Image Classification Dataset

Job Information

Job Directory /home/michaelo/.digits
/jobs/20150311-171431-e0d8

Image Type Color

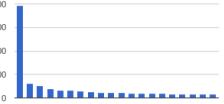
Image Dimensions 256x256

Resize Mode half_crop

Create DB (train)

Input file train.txt

DB Entries 26759



Parse Folder (train/val)

Folder http://sqlr/data/images/voc_cropped/

Number of categories 20

Training images 26759

Validation images 8917 (25.0%)

Configure DNN

DIGITS New Model

Select Dataset PASCAL VOC
ILSVRC 2012
MNIST Dataset

Solver Options

Training epochs 30

Validation interval (in epochs) 1

(neat progress bar)

Batch size 100

Base Learning Rate 0.01

Show advanced learning rate options

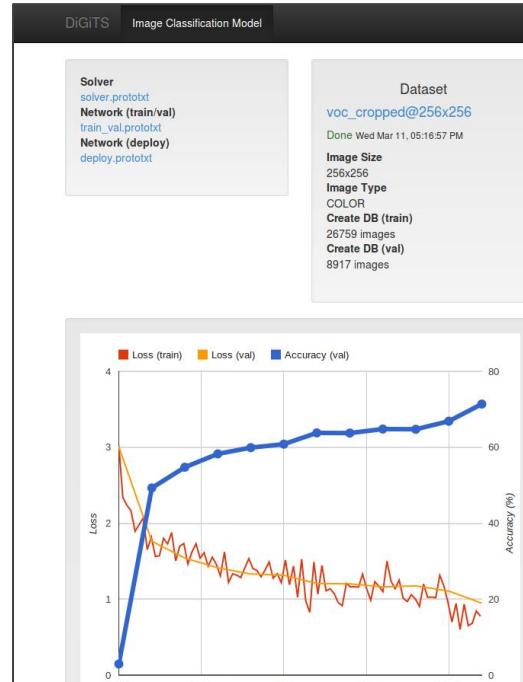
Custom Network

```
layer {  
    name: "conv1"  
    type: "Convolution"  
    bottom: "data"  
    top: "conv1"  
    param {  
        lr_mult: 1  
        decay_mult: 1  
    }  
}
```

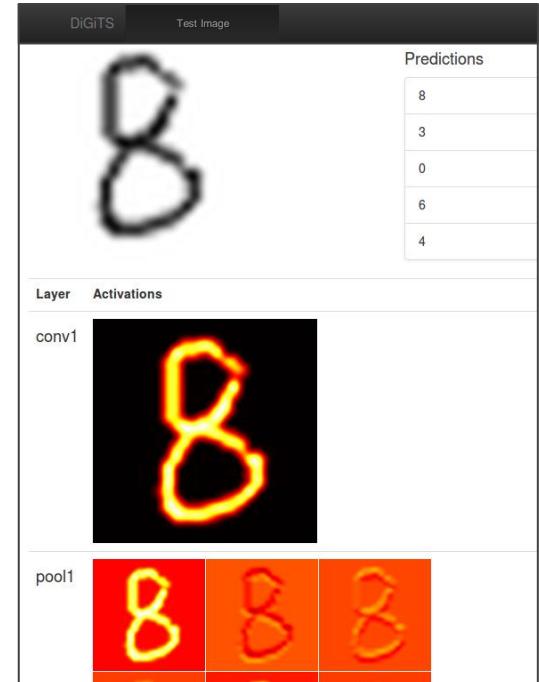
Model Name ImageNet

Create

Monitor Progress

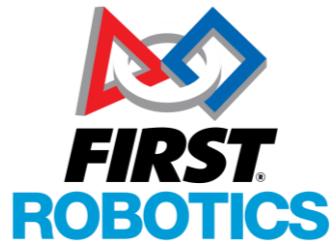


Visualize Layers



developer.nvidia.com/digits

GitHub

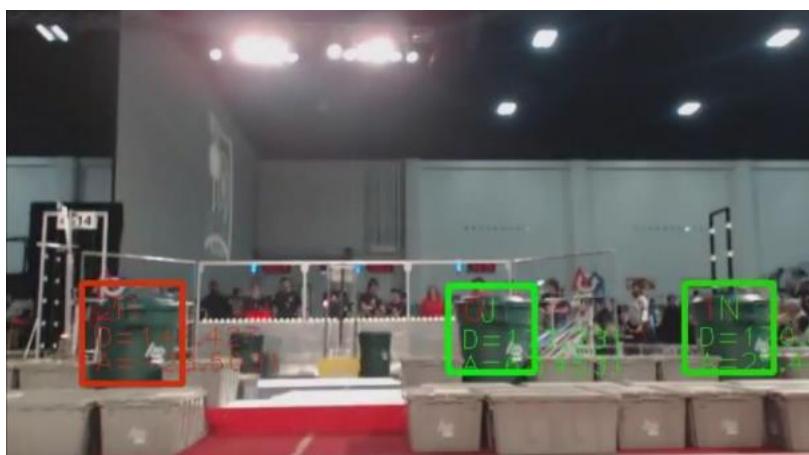
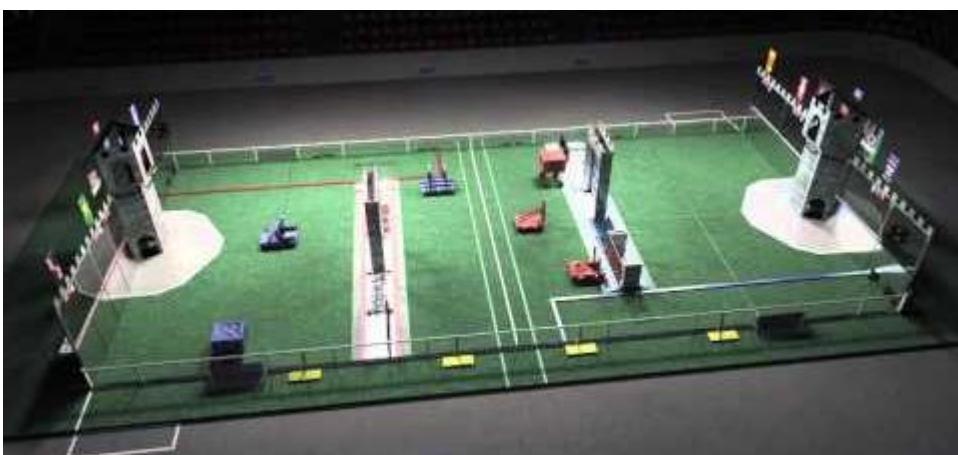
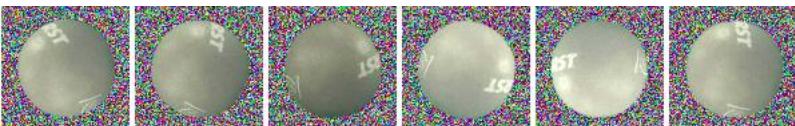


FIRST Team 900

ROBUST DATA COLLECTION

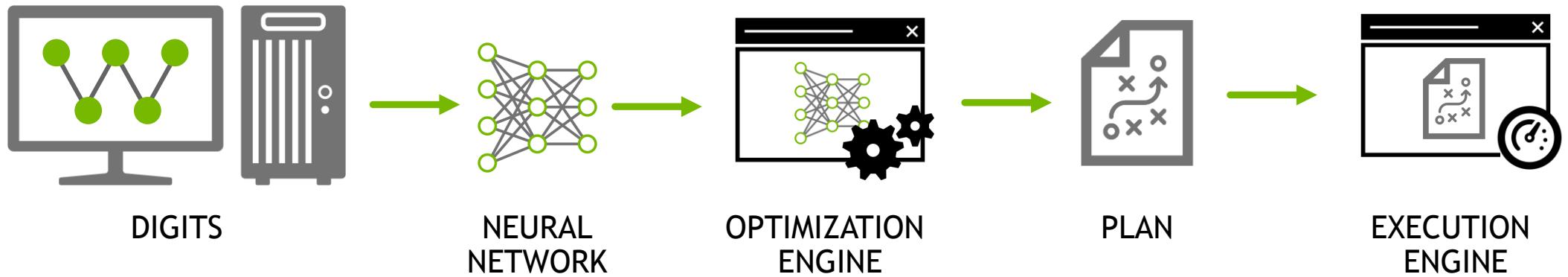


ZEBRACORNS
team900.org



GPU INFERENCE ENGINE

Workflow



Caffe features

Deep Learning model definition

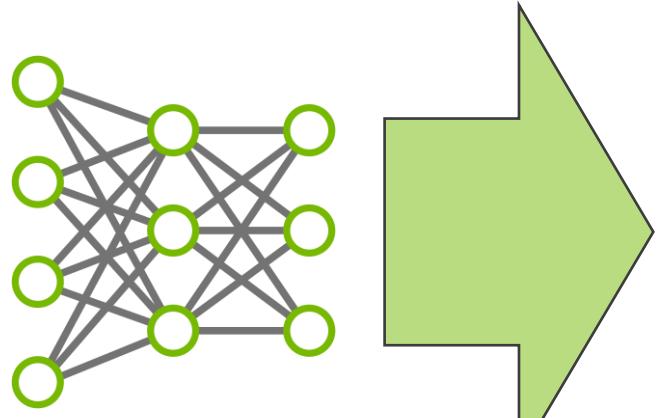
- ▶ Protobuf model format

- ▶ Strongly typed format
- ▶ Human readable
- ▶ Auto-generates and checks Caffe code
- ▶ Developed by Google
- ▶ Used to define network architecture and training parameters
- ▶ No coding required!

```
name: "conv1"
type: "Convolution"
bottom: "data"
top: "conv1"
convolution_param {
    num_output: 20
    kernel_size: 5
    stride: 1
    weight_filler {
        type: "xavier"
    }
}
```

GPU INFERENCE ENGINE

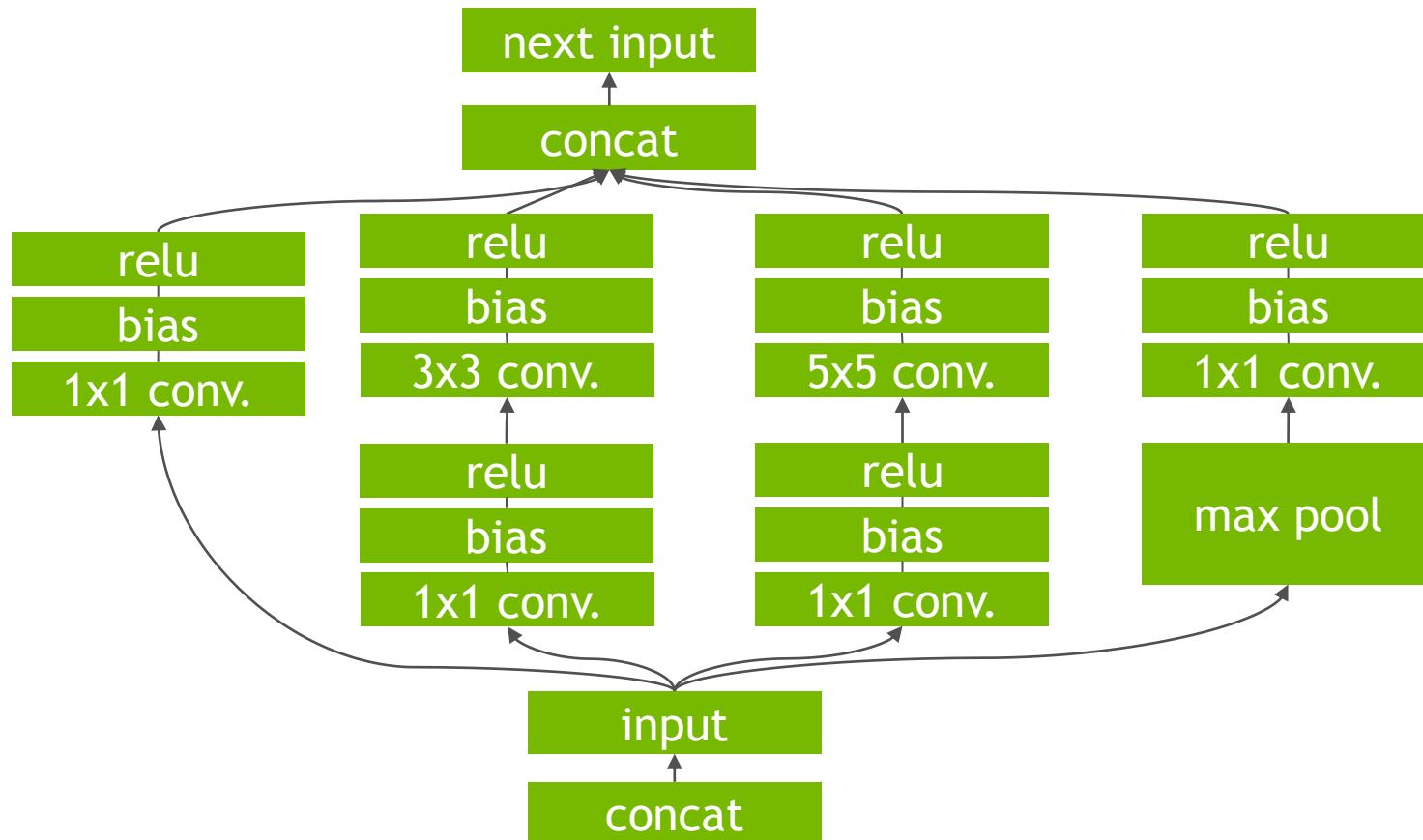
Optimizations



- Fuse network layers
- Eliminate concatenation layers
- Kernel specialization
- Auto-tuning for target platform
- Select optimal tensor layout
- Batch size tuning

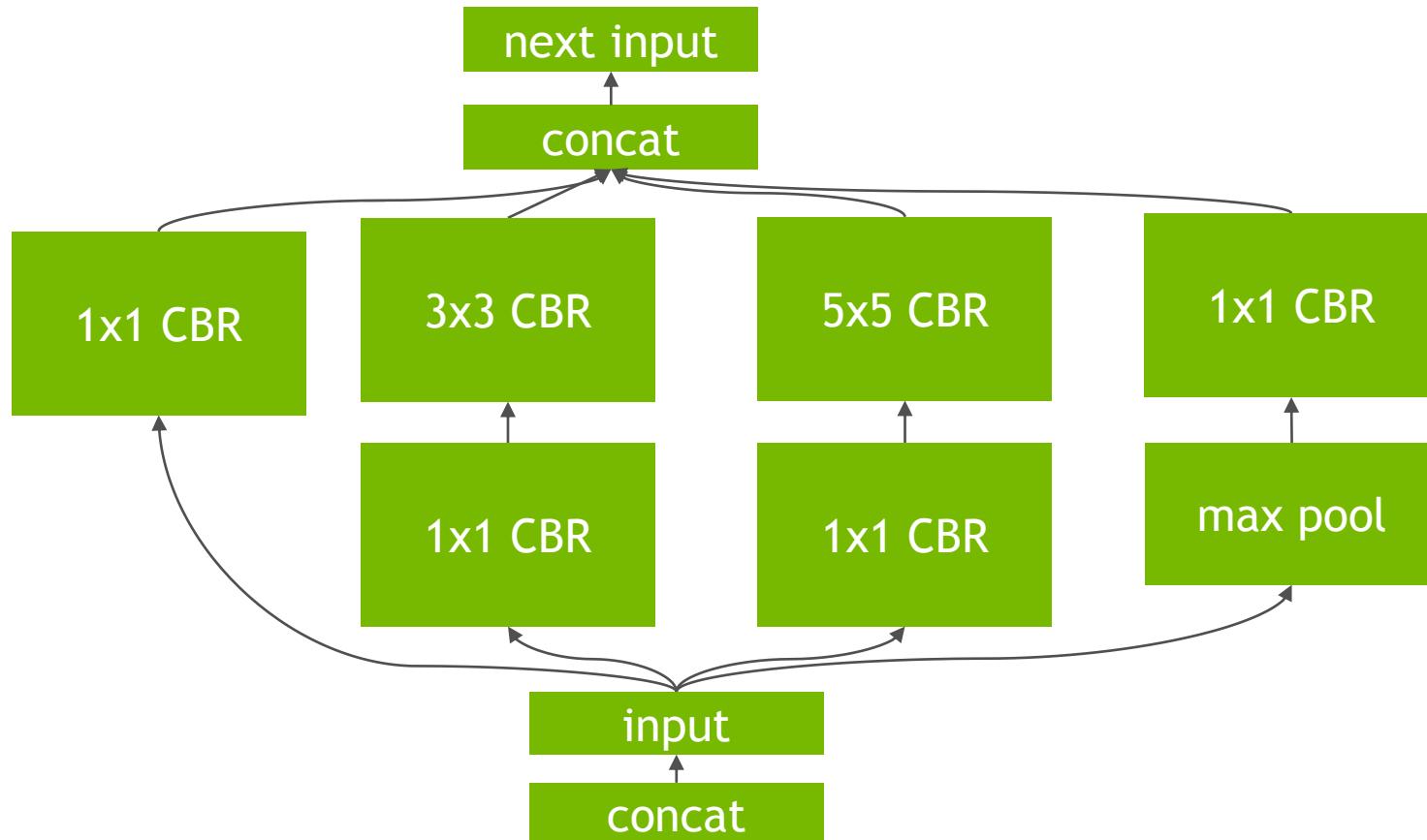


Graph Optimization



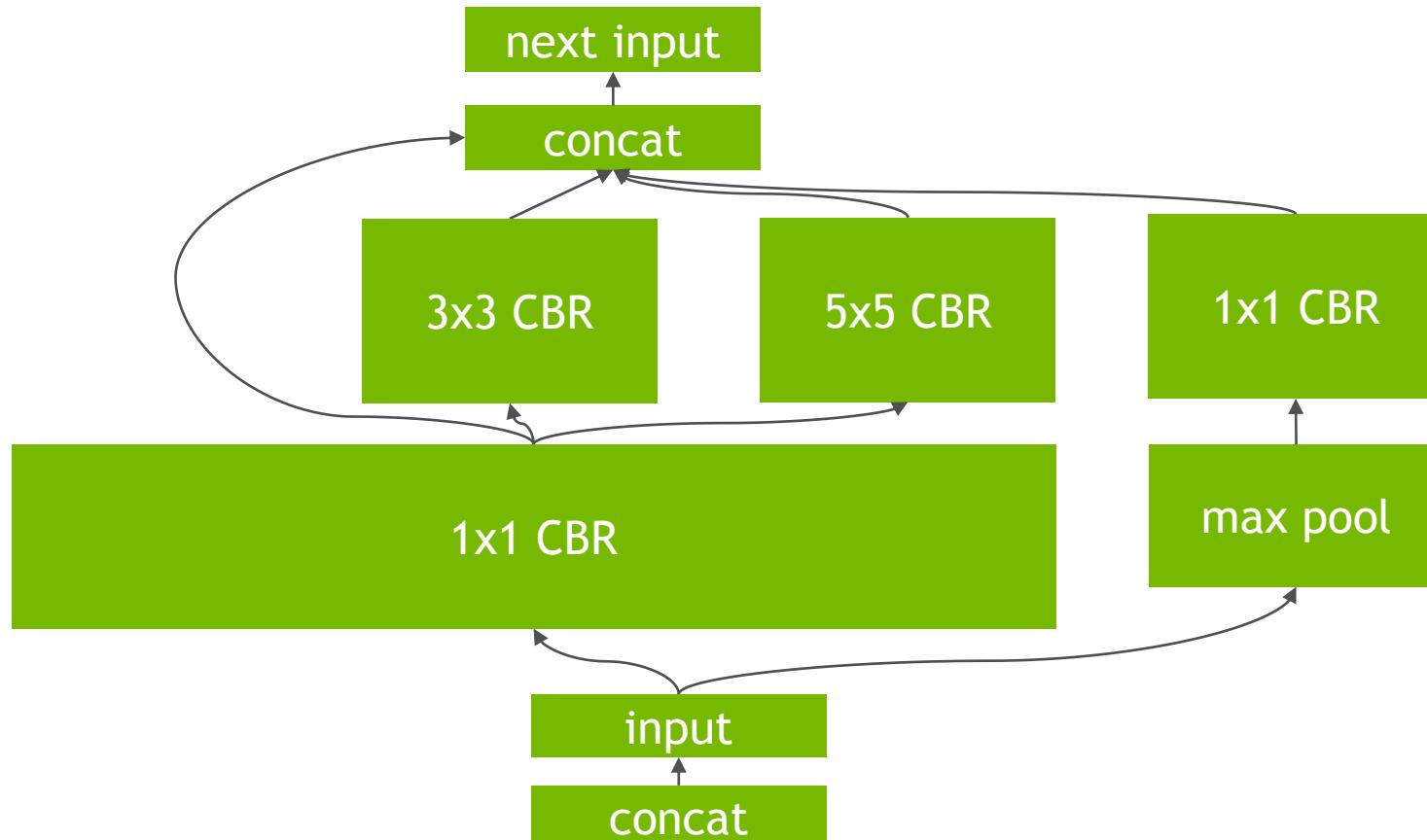
Graph Optimization

Vertical fusion



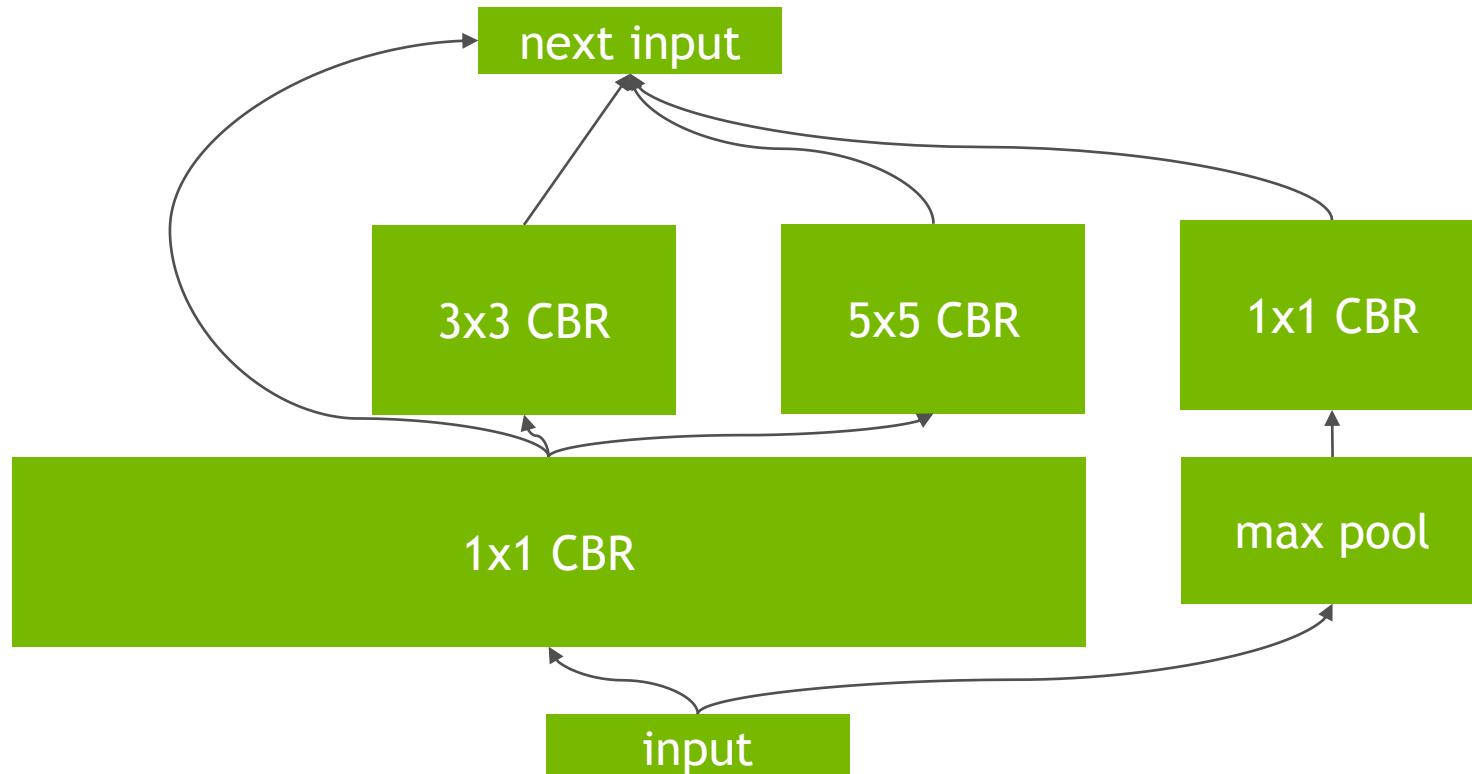
Graph Optimization

Horizontal fusion



Graph Optimization

Concat elision



Autotuning

Choose the fastest kernel for each layer

- Baseline is cuDNN / cuBLAS
- Direct convolution kernels for small batch
- Custom Winograd & Implicit GEMM for Half2
- Custom Deconvolution for filter size == stride case
- Weight pre-transform for Winograd
- Optimal T/N choice for BLAS
- Run `cudnnFindForwardConvolutionEx()` with multiple iterations

Build

Importing a Caffe Model

```
// create the network definition
INetworkDefinition* network = infer->createNetwork();

// create a map from caffe blob names to GIE tensors
std::unordered_map<std::string, infer1::Tensor> blobNameToTensor;

// populate the network definition and map
CaffeParser* parser = new CaffeParser;
parser->parse(deployFile, modelFile, *network, blobNameToTensor);

// tell GIE which tensors are required outputs
for (auto& s : outputs)
    network->setOutput(blobNameToTensor[s]);
```

Build

Engine Creation

```
// Specify the maximum batch size and scratch size
CudaEngineBuildContext buildContext;
buildContext.maxBatchSize = maxBatchSize;
buildContext.maxWorkspaceSize = 1 << 20;

// create the engine
ICudaEngine* engine =
    infer->createCudaEngine(buildContext, *network);

// serialize to a C++ stream
engine->serialize(gieModelStream);
```

Runtime

Binding Buffers

```
// get array bindings for input and output
int inputIndex = engine->getBindingIndex(INPUT_BLOB_NAME),
    outputIndex = engine->getBindingIndex(OUTPUT_BLOB_NAME);

// set array of input and output buffers
void* buffers[2];
buffers[inputIndex] = gpuInputBuffer;
buffers[outputIndex] = gpuOutputBuffer;
```

Runtime

Running the Engine

```
// Specify the batch size
CudaEngineContext context;
context.batchSize = batchSize;

// add GIE kernels to the given stream
engine->enqueue(context, buffers, stream, NULL);

<...>

// wait on the stream
cudaStreamSynchronize(stream);
```

NVIDIA Deep Learning Institute

Hands-on Training for Data Scientists and Software Engineers



Training organizations and individuals to solve challenging problems using Deep Learning

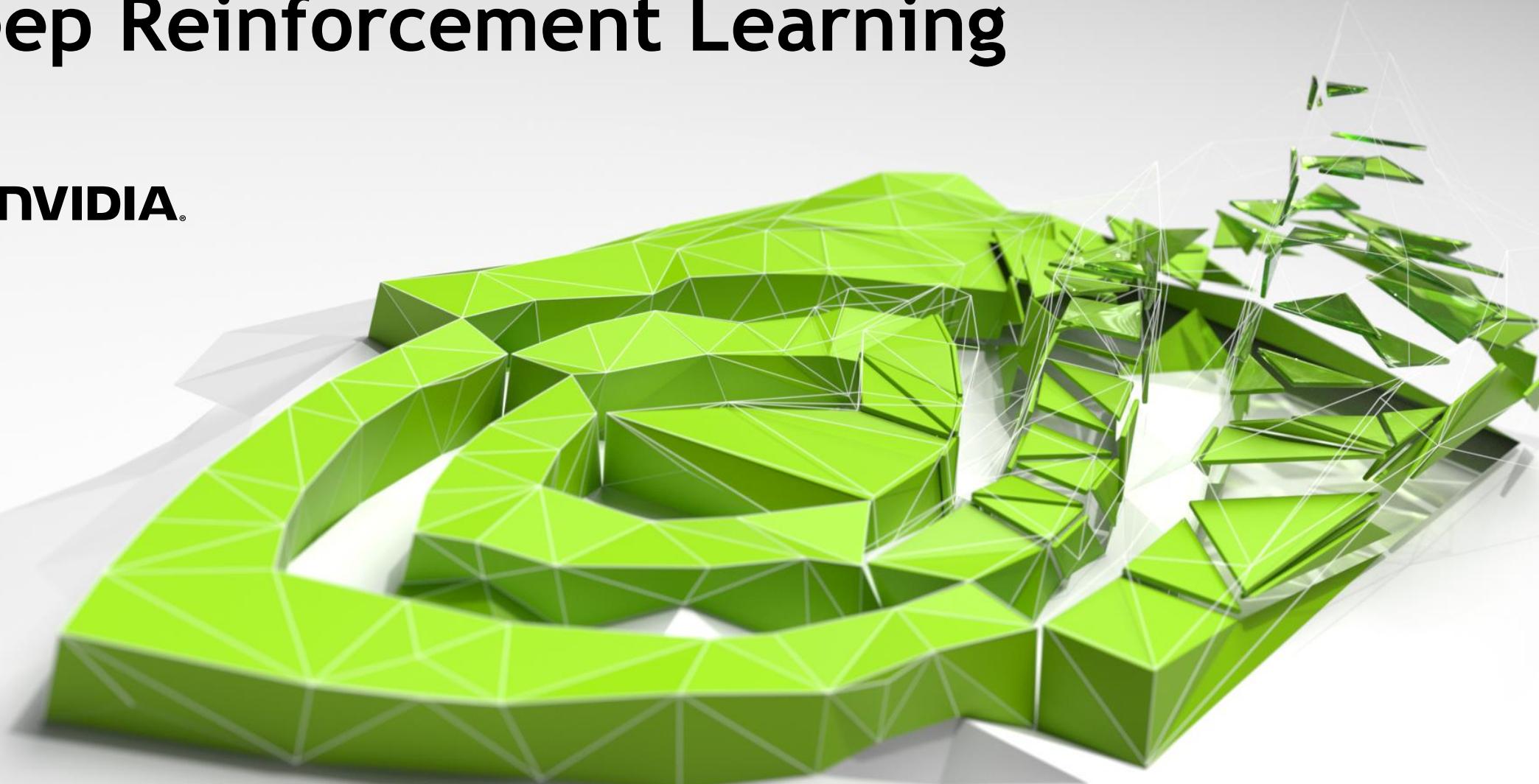
On-site workshops and online courses presented by certified experts

Covering complete workflows for proven application use cases

Image classification, object detection, natural language processing, recommendation systems, and more

www.nvidia.com/dli

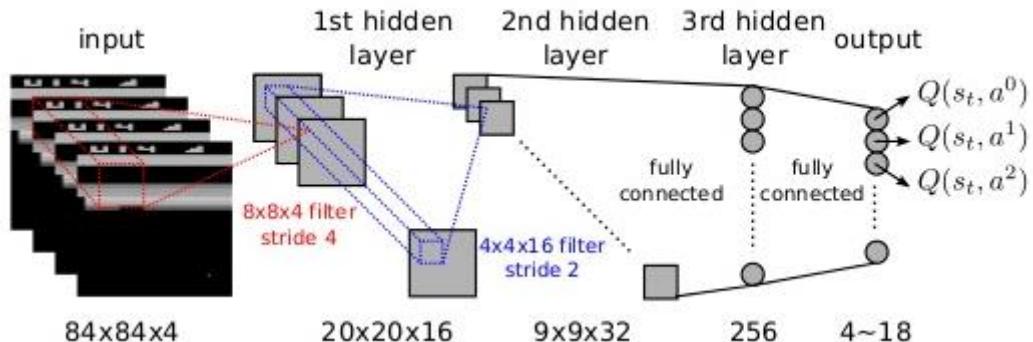
Deep Reinforcement Learning



PLAYING ATARI WITH DEEPMIND



From Pixels to Actions: Human-level control through Deep Reinforcement Learning





AlphaGo wins match against Go champ Lee Se-dol 4-1

Historic Go match shows technological advances in AI research



AlphaGo beats world champion Lee Sedol in the second match of the Go tournament

Go-playing artificial intelligence program AlphaGo won a historic match against Korean Go grandmaster Lee Se-dol with a final score of 4-1.

Although Lee put up a good fight, he resigned after 280 moves in the final showdown, which lasted around five hours -- the longest in the five-game match.

"It is regrettable that I could not wrap up the match by winning the final round," said Lee, attending a press conference after the final game at Four Seasons Hotel in Gwanghwamun on Tuesday.

"I thought I got an upper hand in the earlier stage of the game today, but I eventually lost because I was not good enough," Lee said.

The Go-playing software boasted its almost impeccable capability to read and predict, and was able to maintain a slight lead throughout the final round.

Asynchronous Methods for Deep Reinforcement Learning

Volodymyr Mnih¹

VMNIH@GOOGLE.COM

Adrià Puigdomènech Badia¹

ADRIAP@GOOGLE.COM

Mehdi Mirza^{1,2}

MIRZAMOM@IRO.UMONTREAL.CA

Alex Graves¹

GRAVESEA@GOOGLE.COM

Tim Harley¹

THARLEY@GOOGLE.COM

Timothy P. Lillicrap¹

COUNTZERO@GOOGLE.COM

David Silver¹

DAVIDSILVER@GOOGLE.COM

Koray Kavukcuoglu¹

KORAYK@GOOGLE.COM

¹ Google DeepMind

² Montreal Institute for Learning Algorithms (MILA), University of Montreal

Abstract

We propose a conceptually simple and lightweight framework for deep reinforcement learning that uses asynchronous gradient descent for optimization of deep neural network controllers. We present asynchronous variants of four standard reinforcement learning algorithms and show that parallel actor-learners have a stabilizing effect on training allowing all four methods to successfully train neural network controllers. The best performing method, an asynchronous variant of actor-critic, surpasses the current state-of-the-art on the Atari domain while training for half the time on a single multi-core CPU instead of a GPU. Furthermore, we show that asynchronous actor-critic succeeds on a wide variety of continuous motor control problems as well as on a new task involving finding rewards in random 3D mazes using a visual input.

<http://arxiv.org/abs/1602.01783>



AlphaGo wins match against Go champ Lee Se-dol 4-1

H



Inside Google's DeepMind AlphaGo GPU cluster

January 28, 2016

For the first time, a computer has beaten a human professional at the game of Go – an ancient board game that has long been viewed as one of the greatest challenges for Artificial Intelligence.

Google DeepMind's GPU-accelerated AlphaGo program beat Fan Hui, the European Go champion, five times out of five in tournament conditions.

Demis Hassabis, who oversees DeepMind, mentioned in a [recent article](#) that DeepMind's [deep learning](#) system works pretty well on a single computer equipped with a decent number of GPU accelerators, but for the match against Fan Hui, the researchers used a larger network of computers that spanned about 170 GPUs. This larger computer network both trained the system and played the actual game, drawing on the results of the training.

All
G
A
a
a
w
T
n



ronous Methods for Deep Reinforcement Learning

or Mnih¹

VMNIH@GOOGLE.COM

igdomènech Badia¹

ADRIAP@GOOGLE.COM

irza^{1,2}

MIRZAMOM@IRO.UMONTREAL.CA

ves¹

GRAVESA@GOOGLE.COM

ey¹

THARLEY@GOOGLE.COM

P. Lillicrap¹

COUNTZERO@GOOGLE.COM

ver¹

DAVIDSILVER@GOOGLE.COM

vukcuoglu¹

KORAYK@GOOGLE.COM

DeepMind

¹ Institute for Learning Algorithms (MILA), University of Montreal

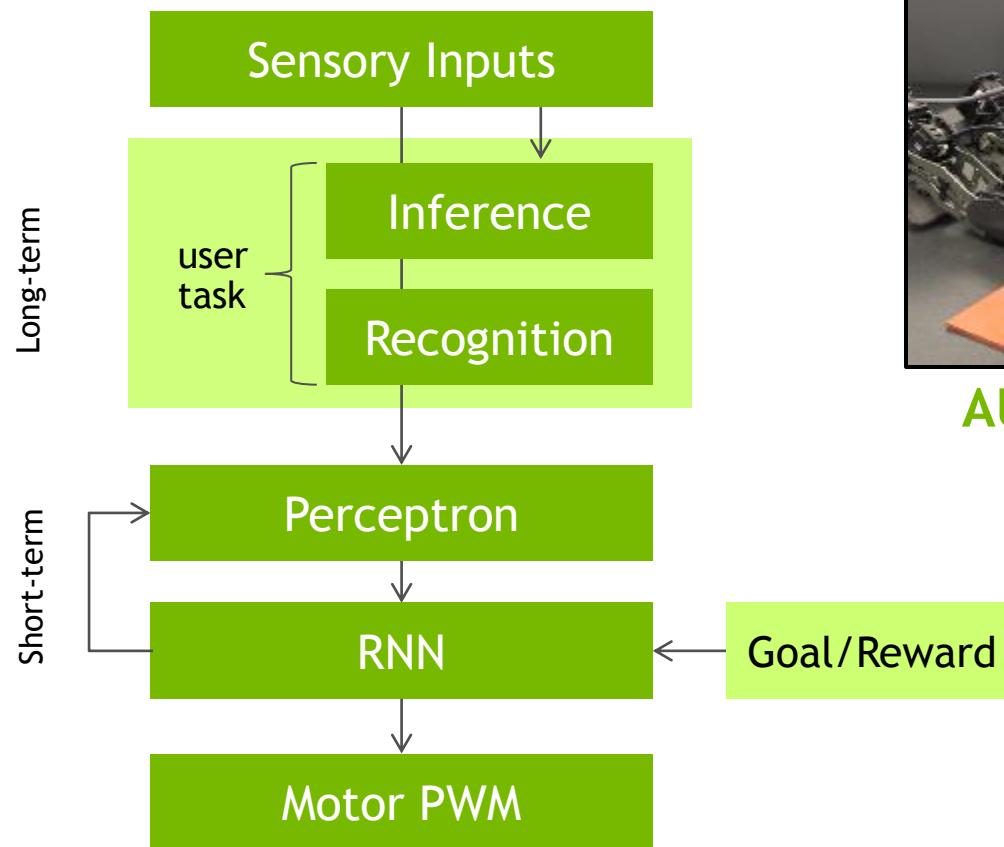
Abstract

pose a conceptually simple and lightweight framework for deep reinforcement learning that uses asynchronous gradient descent for optimization of deep neural controllers. We present asynchronous variants of four standard reinforcement learning algorithms and show that parallel actor-learners have a stabilizing effect on allowing all four methods to successfully train neural network controllers. The learning method, an asynchronous variant of actor-critic, surpasses the current state-of-the-art on the Atari domain while training for half the time on a single CPU instead of a GPU. Furthermore, we show that asynchronous actor-critic on a wide variety of continuous motor control problems as well as on a new task involving finding rewards in random 3D mazes using a visual input.

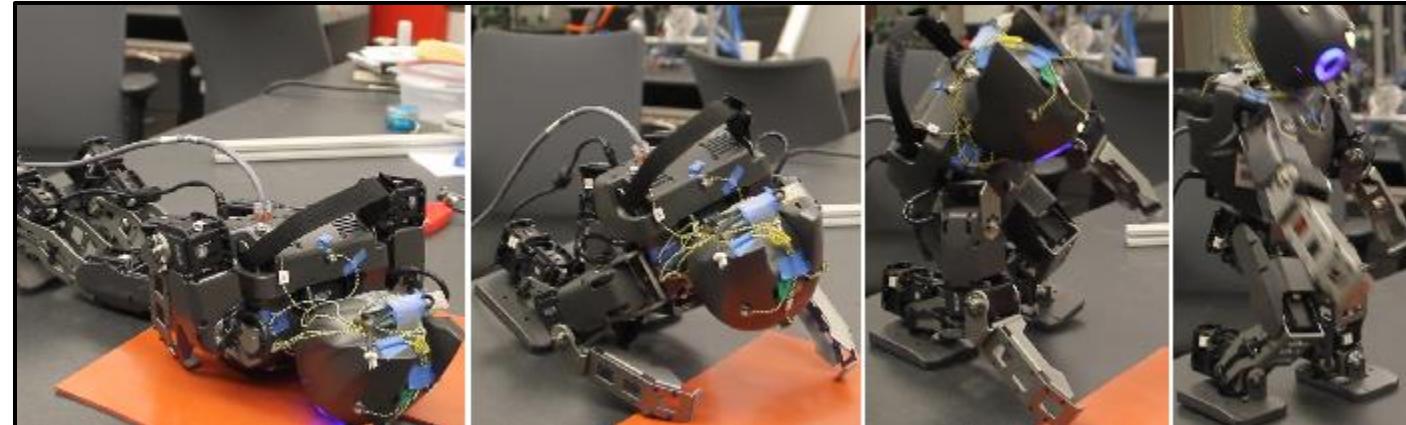
<http://arxiv.org/abs/1602.01783>



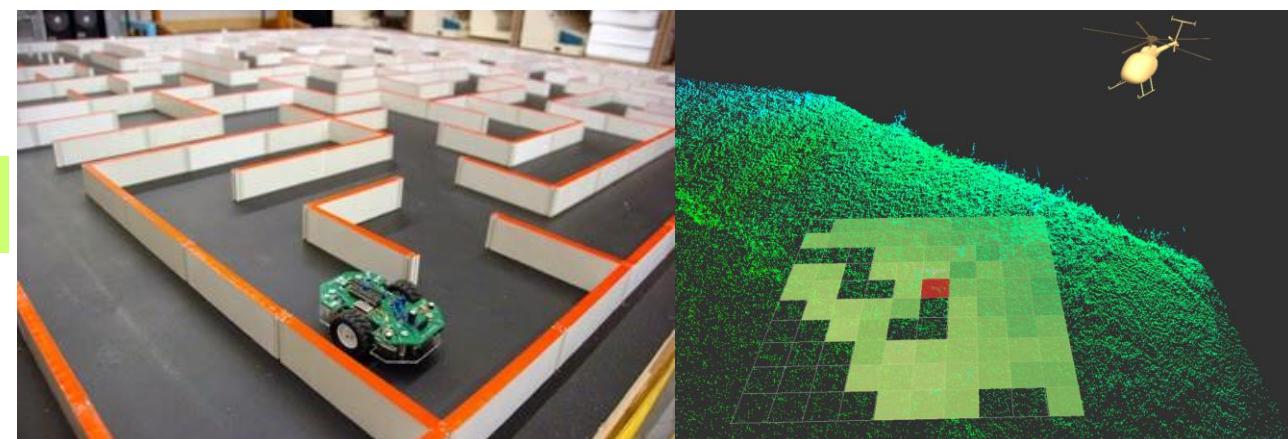
END-TO-END LEARNING



MOTION CONTROL



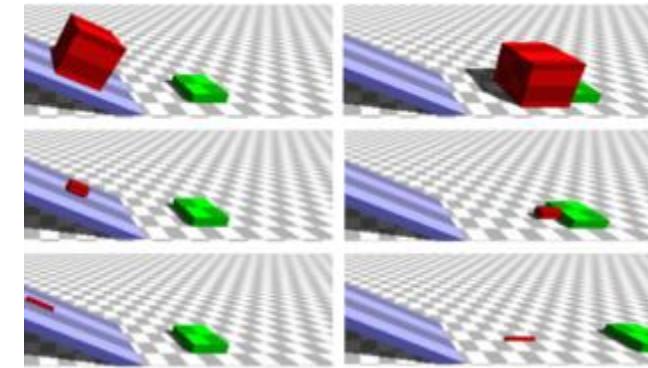
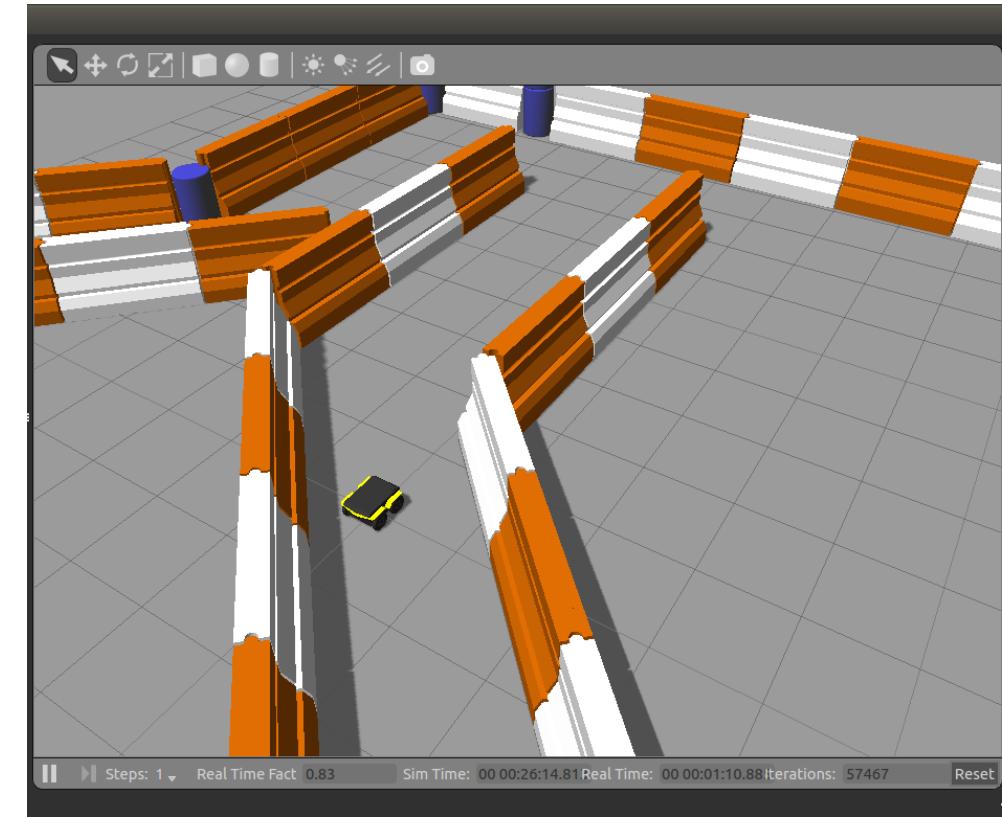
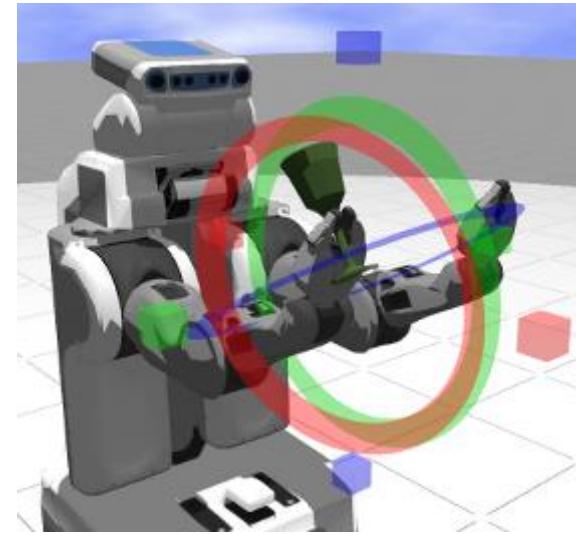
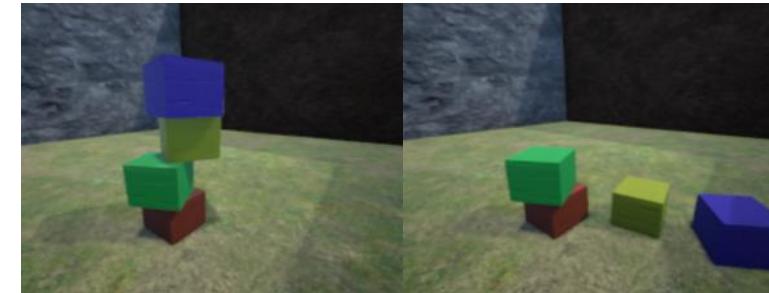
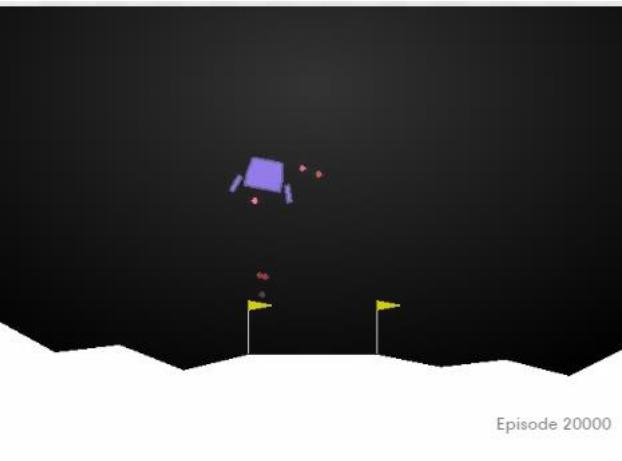
AUTONOMOUS NAVIGATION



SIMULATION

Physical Intuition

- ▶ OpenAI Gym
- ▶ Gazebo
- ▶ Unreal4Torch
- ▶ PhysX
- ▶ Others



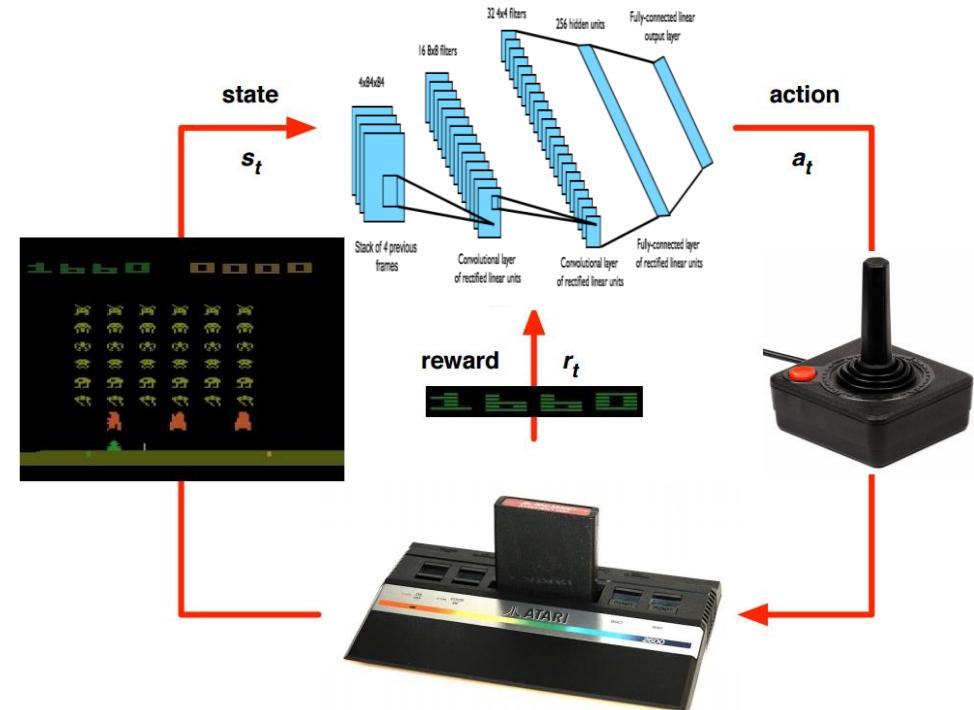
Episode 20000

Steps: 1 ▶ Real Time Factor: 0.83 Sim Time: 00:00:26:14.81 Real Time: 00:00:10.88 Iterations: 57467 Reset

Q-LEARNING

How's it work?

- ▶ A reinforcement learning agent includes:
 - ▶ **state** (environment)
 - ▶ **actions** (controls)
 - ▶ **reward** (feedback)
- ▶ A **value function** predicts the future reward of performing actions in the current state
 - ▶ Given the recent state, action with the maximum estimated future reward is chosen for execution
- ▶ For agents with complex state spaces, deep networks are used as **Q-value** approximator
 - ▶ Numerical solver (**gradient descent**) optimizes the network on-the-fly based on reward inputs

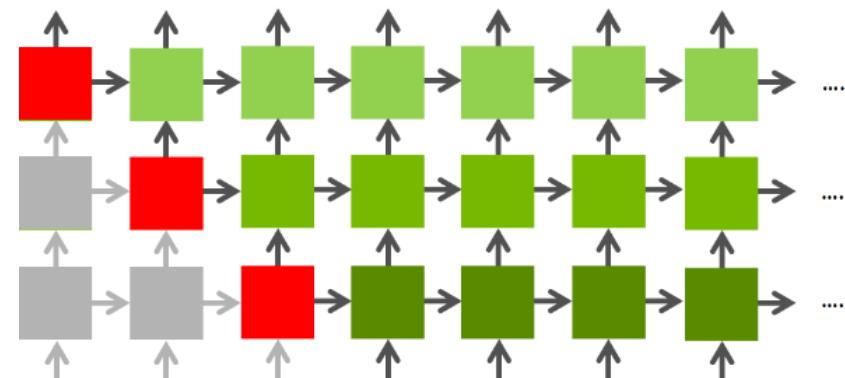
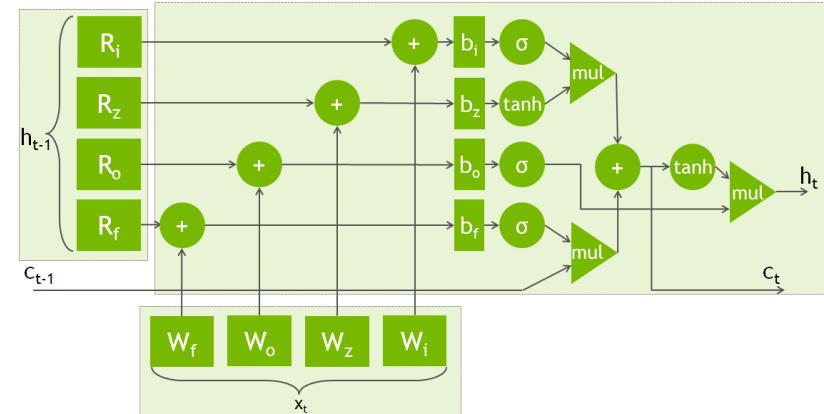


Example NN in Lua

```
1: -- parameters
2: nstates = {16,256,128}
3: fanin = {1,4}
4: filtsize = 5
5: poolsize = 2
6: normkernel = image.gaussian1D(7)
7:
8: -- Container:
9: model = nn.Sequential()
10:
11: -- stage 1 : filter bank -> squashing -> L2 pooling -> normalization
12: model:add(nn.SpatialConvolutionMap(nn.tables.random(nfeats, nstates[1], fanin[1]), filtsize, filtsize))
13: model:add(nn.Tanh())
14: model:add(nn.SpatialLPPooling(nstates[1],2,poolsize,poolsize,poolsize))
15: model:add(nn.SpatialSubtractiveNormalization(16, normkernel))
16:
17: -- stage 2 : filter bank -> squashing -> L2 pooling -> normalization
18: model:add(nn.SpatialConvolutionMap(nn.tables.random(nstates[1], nstates[2], fanin[2]), filtsize, filtsize))
19: model:add(nn.Tanh())
20: model:add(nn.SpatialLPPooling(nstates[2],2,poolsize,poolsize,poolsize))
21: model:add(nn.SpatialSubtractiveNormalization(nstates[2], normkernel))
22:
23: -- stage 3 : standard 2-layer neural network
24: model:add(nn.Reshape(nstates[2]*filtsize*filtsize))
25: model:add(nn.Linear(nstates[2]*filtsize*filtsize, nstates[3]))
26: model:add(nn.Tanh())
27: model:add(nn.Linear(nstates[3], noutputs))
```

LSTM ACCELERATION

- ▶ Launch a 2D grid of RNN cells
- ▶ Multiple layers in a single call are faster
- ▶ Doesn't suffer from vanishing gradient
 - ▶ Able to adopt long-term strategy
- ▶ Supports:
 - ▶ Partially-observable environments
 - ▶ Uni/Bidirectional RNNs
 - ▶ Non-uniform length minibatches
 - ▶ Dropout between layers



DEEP-LEARNING RESEARCH ROVER

TURBO 2.0



[github.org/dusty-nv](https://github.com/dusty-nv)



GitHub

THANK YOU!

Q&A: WHAT CAN I HELP YOU BUILD?

dustinf@nvidia.com
412-419-8418



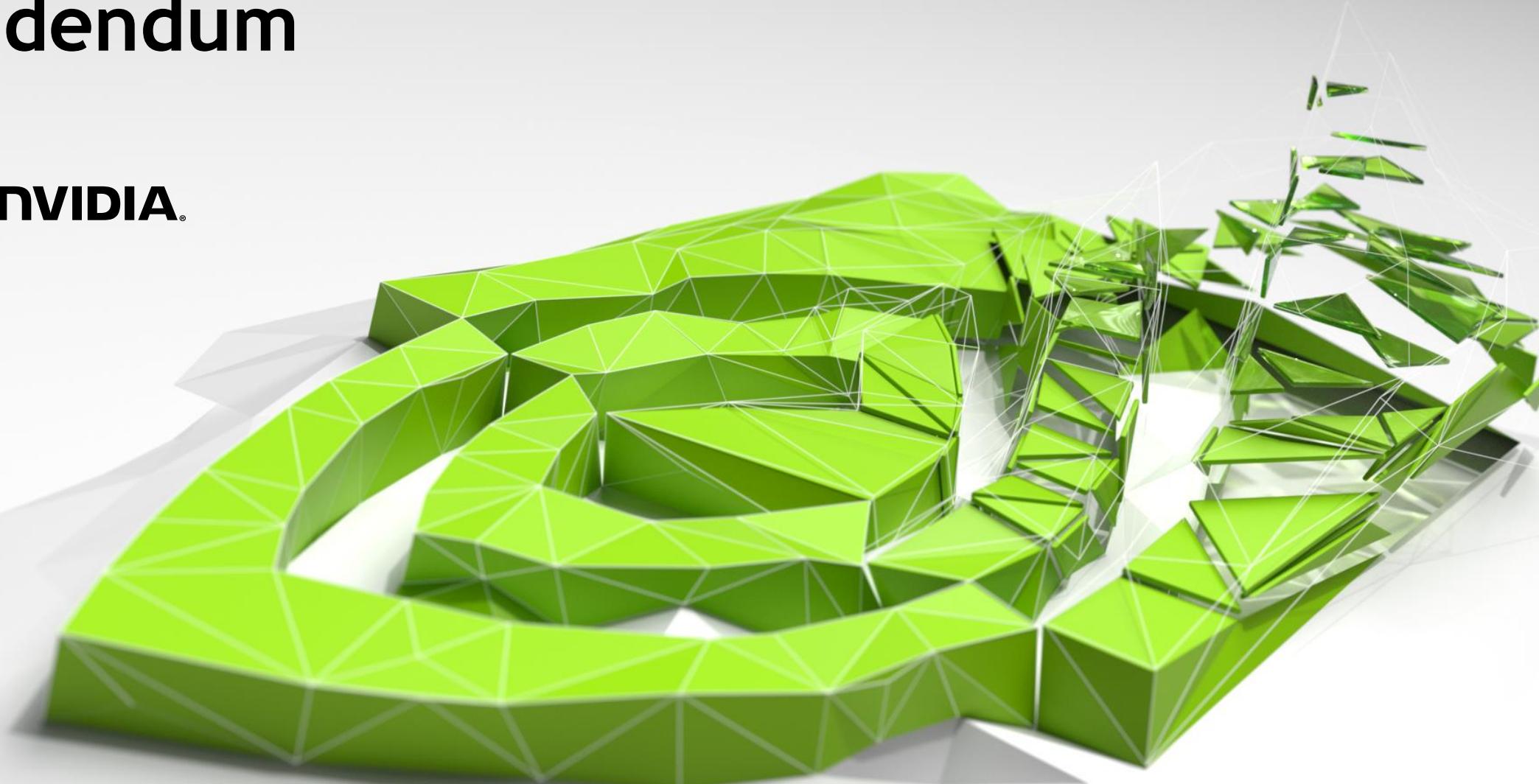
Become a registered developer today!

<https://developer.nvidia.com/embedded-developer-program>

Jetson Academic Discount: <http://www.nvidia.com/edu-jetson>

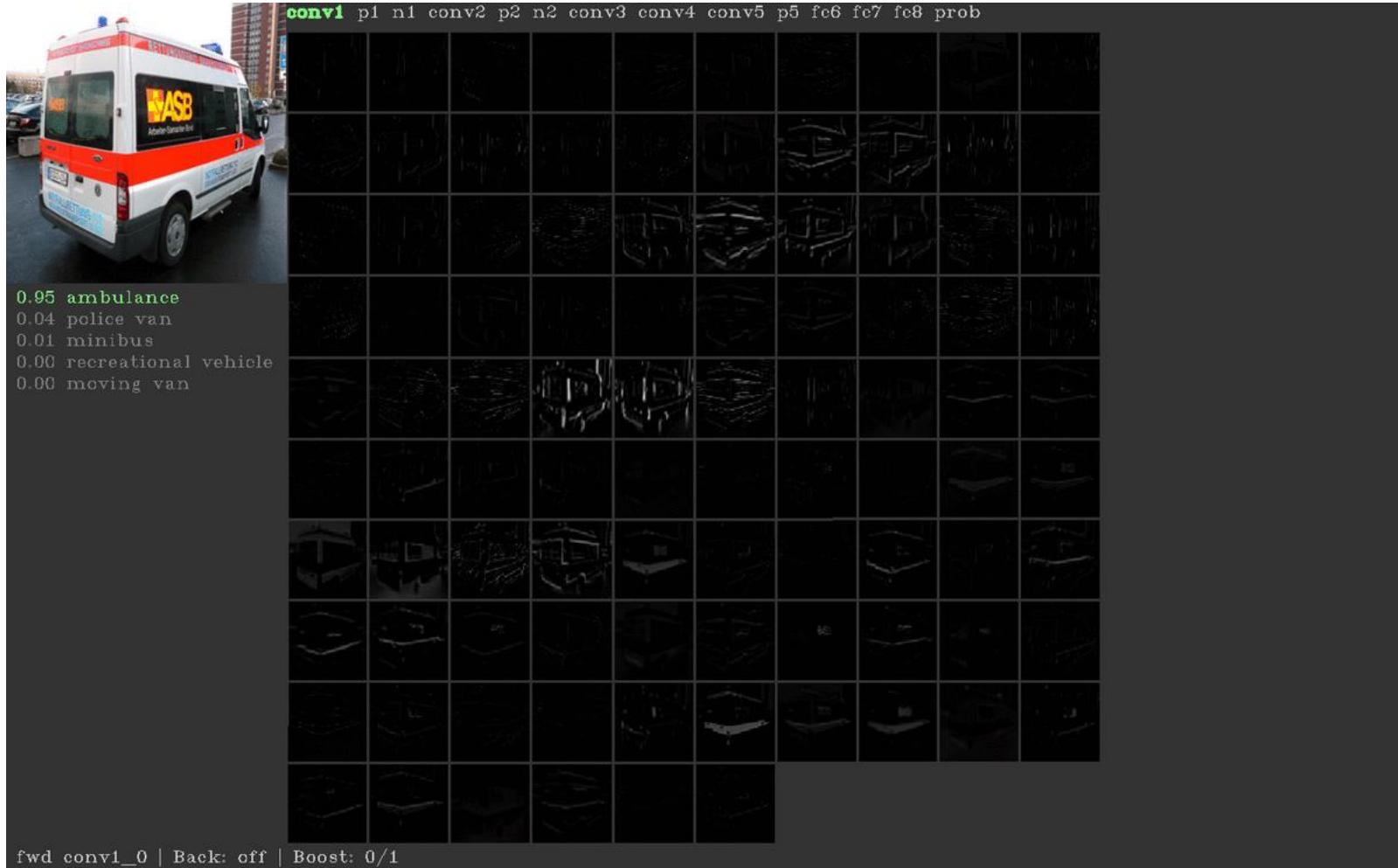
Jetson SDK and Documentation: <https://developer.nvidia.com/embedded-computing>

Addendum



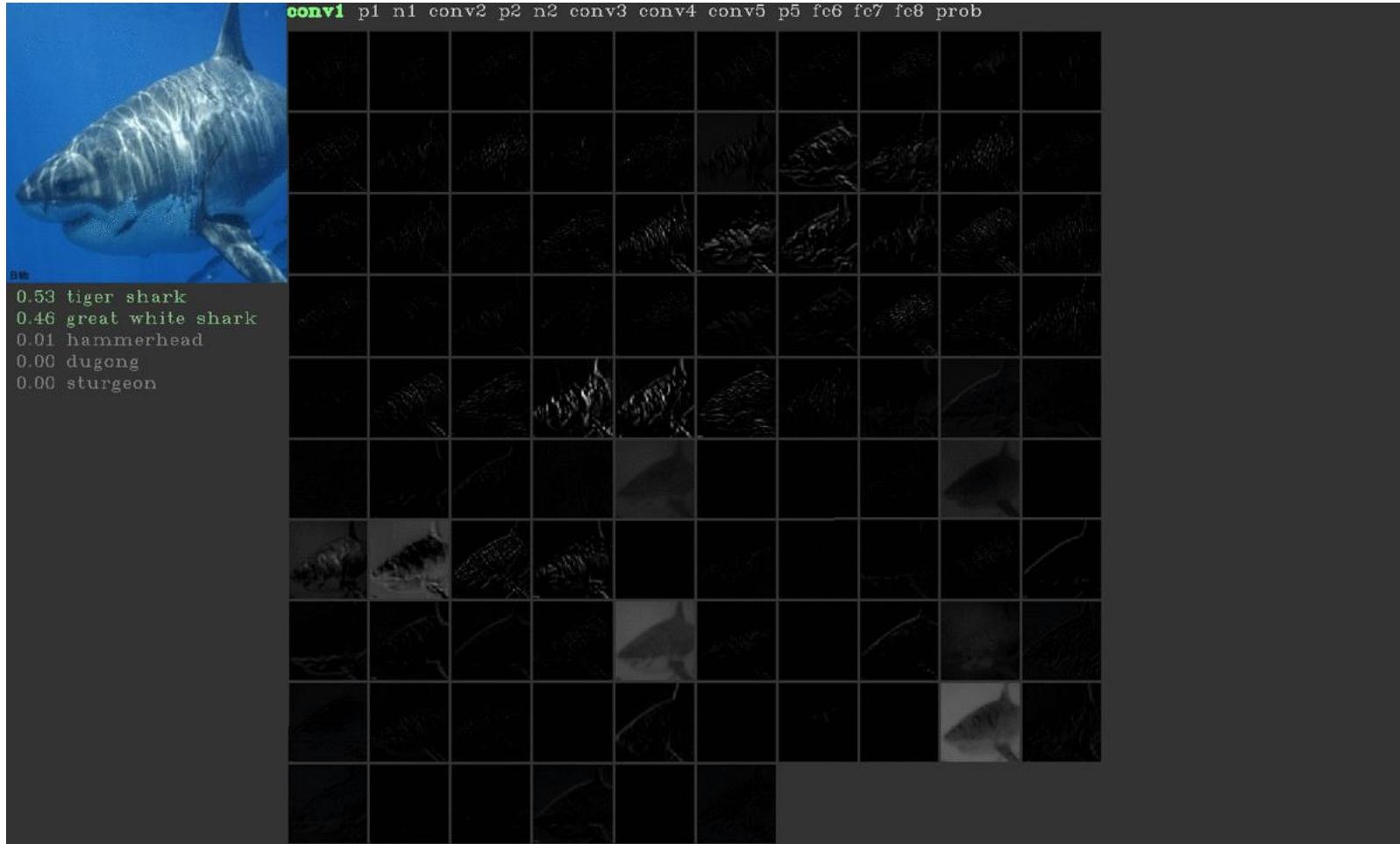
DEEP VISUALIZATION TOOLBOX

IMAGE RECOGNITION



DEEP VISUALIZATION TOOLBOX

IMAGE RECOGNITION





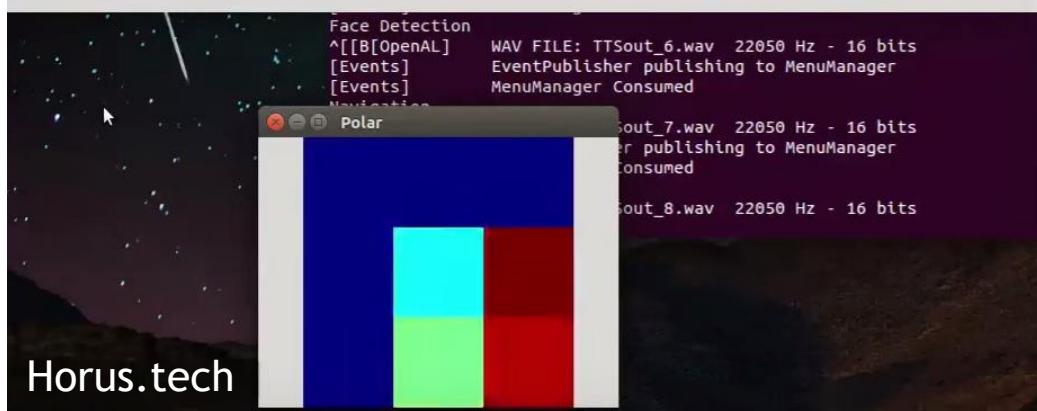
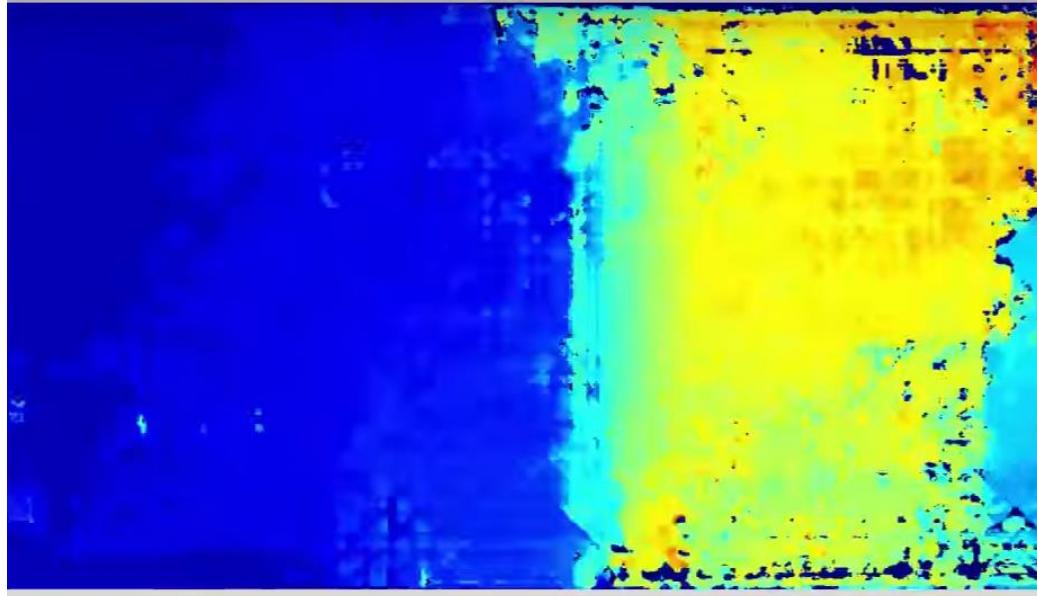
Horus
Technology

HORUS

BLIND ASSIST DEVICE

EMERGING COMPANIES
SUMMIT

SOCIAL INNOVATION
AWARD WINNER



IDIA



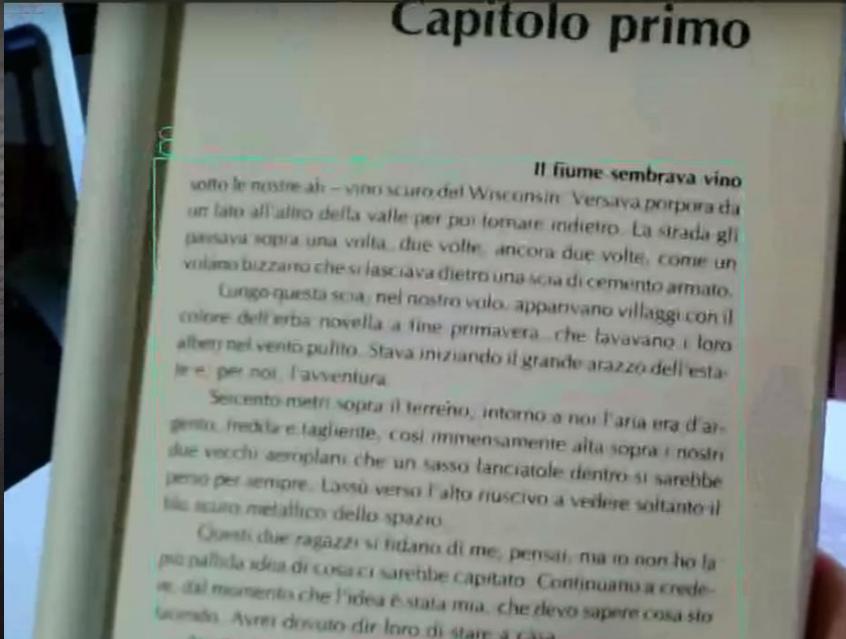
Horus
Technology

HORUS

BLIND ASSIST DEVICE

EMERGING COMPANIES
SUMMIT

SOCIAL INNOVATION
AWARD WINNER



```
cv::Mat rec = regToTrack->readBlock(muramachi, rec);
auto c = std::string(crc.readBlock(rec).get());
cout << c << endl;
wavPlayer->playWAVItts tts2wav(c);
ptSrc.stop();
regToTrack = nullptr;
imshow("Rectified", rec);
```

Il fiume sembrava vino sotto le nostre ali – vino scuro del Wisconsin. Versava porpora da un lato all’altro della valle per poi tornare indietro. La strada gli passava sopra una volta, due volte, ancora due volte, come un volano bizzarro che si lasciava dietro una scia di cemento armato.

Lungo questa scia, nel nostro volo, apparivano villaggi con il colore dell’erba novella a fine primavera, che lavavano i loro alberi nel vento pulito. Stava iniziando il grande arazzo dell'estate e, per noi, l'avventura.

Seicento metri sopra il terreno, intorno a noi l’aria era d’argento, fredda e tagliente, così immensamente alta sopra i nostri due vecchi aeroplani che un sasso lanciatole dentro si sarebbe perso per sempre. Lassù verso l’alto riuscivo a vedere soltanto il blu scuro metallico dello spazio.

Questi due ragazzi si fidano di me, pensai, ma io non ho la più pallida idea di cosa ci sarebbe capitato. Continuano a credere, dal momento che l’idea è stata mia, che devo sapere cosa sto facendo. Avrei dovuto dir loro di stare a casa.

Horus.tech

IMAGE CAPTIONING WITH NEURALTALK2



a man riding a bike on a beach with a dog in the water



a man sitting at a table with a laptop



a street sign on a pole in front of a building



a plate with a sandwich and a salad



a black and white cat sitting in a bathroom sink



a woman is playing tennis on a tennis court



a little boy standing in a field with a kite



a man is sitting on a bench in front of a building



a man and a woman riding on the back of an elephant



a laptop computer sitting on top of a wooden desk



a laptop computer sitting on top of a wooden desk



a bathroom with a sink and a toilet

OBJECT DETECTION

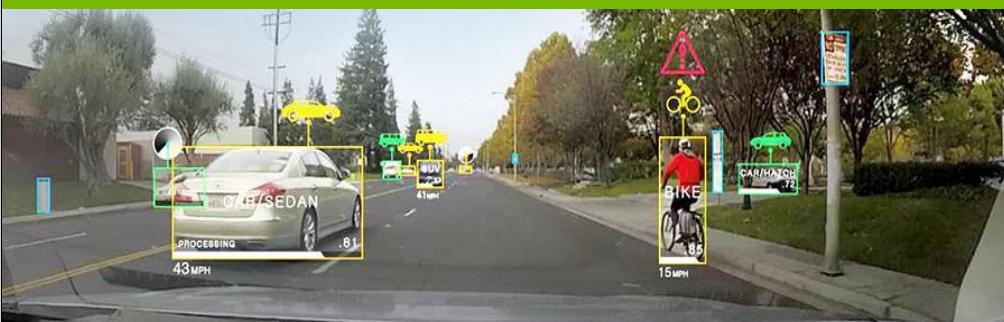
REMOTE SENSING



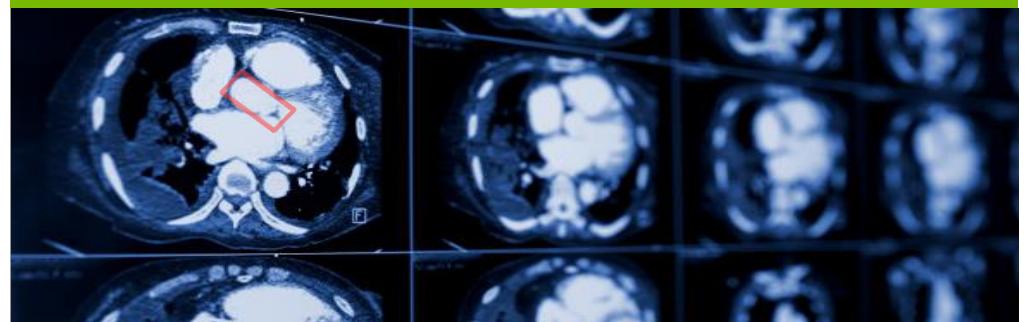
PICKING AND GRASPING



ADVANCED DRIVER ASSISTANCE SYSTEMS (ADAS)



MEDICAL DIAGNOSTICS



ROBERT BOND

DIY INTELLIGENCE



Installing the Particle Photon

The Photon was easy to setup. See the software section below for the code. It sits in the irrigation control box like this:

The black box on the left with the blue LED is a 24VAC to 5VDC converter from eBay. You can see the white relay on the relay board and the blue connector on the converter's 5V output is connected to the Photon's VDD pin. The relay board is basically analog; it has an open-collector NPN transistor with a nominal 3.3V input. Here's the way it is wired:

24VAC converter 24VAC	----- Control box 24VAC OUT
24VAC converter GND	----- Photon VDD (relays to relay board 43.3V)
24VAC converter GND	----- Photon GND, Relay GND
Photon DO COM	----- Relay board signal input
Relay NO	----- Control box 24VAC OUT
Relay NO	----- From relay normally closed

Installing the Jetson

The only hardware bits added to the Jetson are a SATA SSD drive and a small Belkin USB hub. The hub has two wireless dongles that talk to the keyboard and mouse. The Wireless Access Point setup was pretty easy (readily) if you follow [this guide](#). Just use the Ubuntu menus as directed and make sure you add this config setting: I installed Caffe as the [FTP server](#). The configuration is pretty much stock. I did not enable anonymous FTP. I gave the camera a login name and password that is not in my .bashrc file. I installed Caffe using the [InceptionV3 recipe](#). I believe current releases no longer have the LMDNN_MAP_SIZE items so try building it before you make the change. \\$ PYTHONPATH.

myplace.frontier.com/~r.bond

