GPU ACCELERATED DEEP LEARNING WITH CUDNN

Larry Brown Ph.D.

March 2015

AGENDA

1 Introducing cuDNN and GPUs

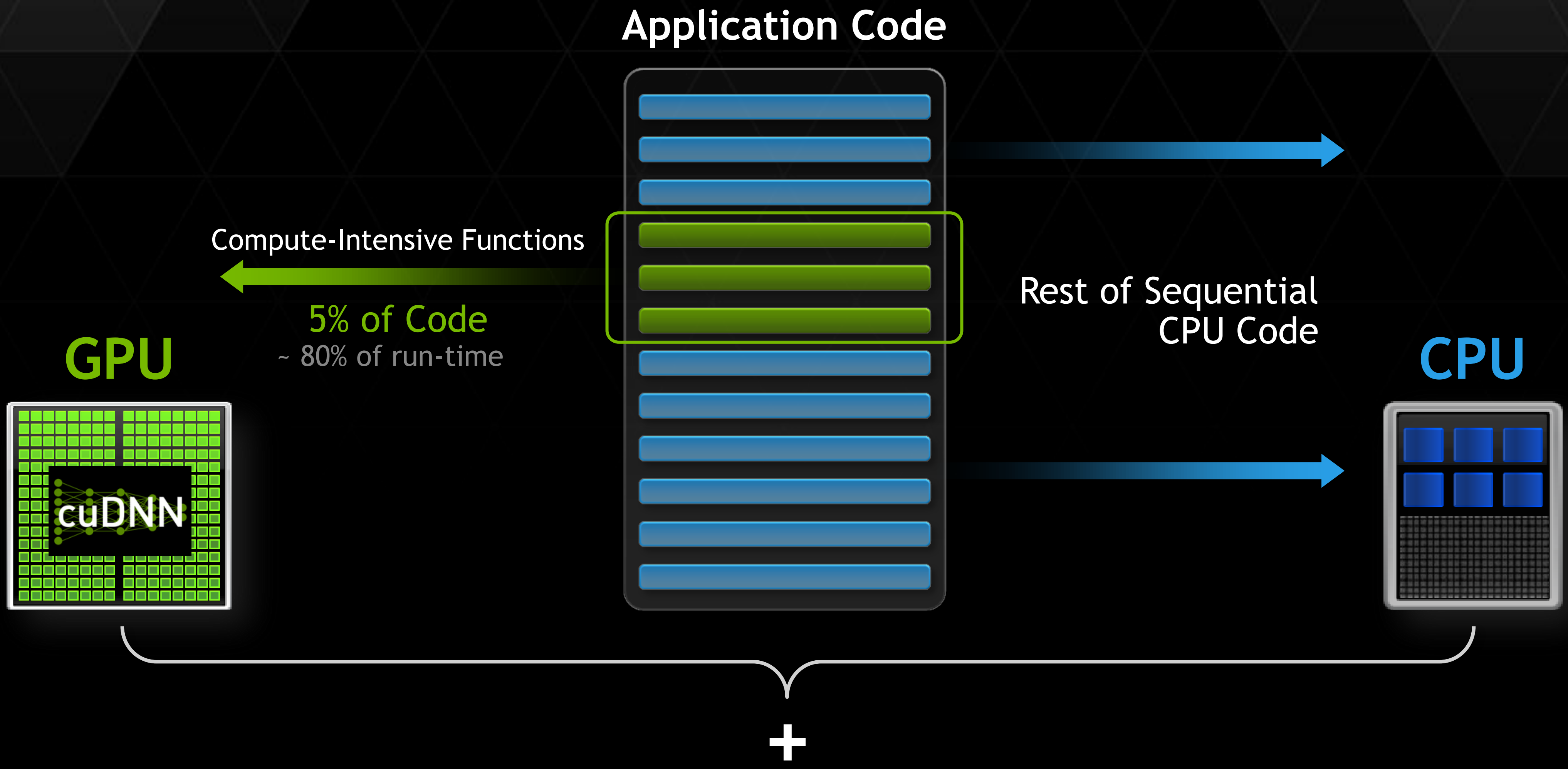
2 Deep Learning Context

3 cuDNN V2

4 Using cuDNN

Introducing cuDNN and GPUs

HOW GPU ACCELERATION WORKS



Application Code

Compute-Intensive Functions

5% of Code

~ 80% of run-time

GPU

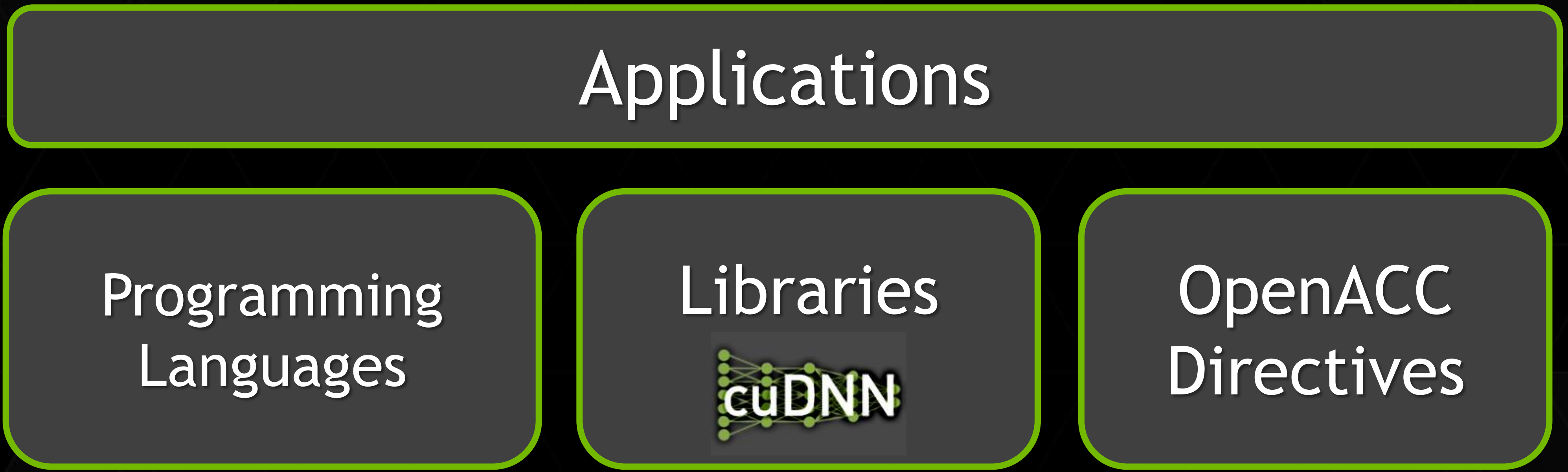
Rest of Sequential
CPU Code

CPU

+

WHAT IS cuDNN?

cuDNN is a library of primitives for deep learning



Applications

Programming Languages

Libraries

cuDNN

OpenACC Directives

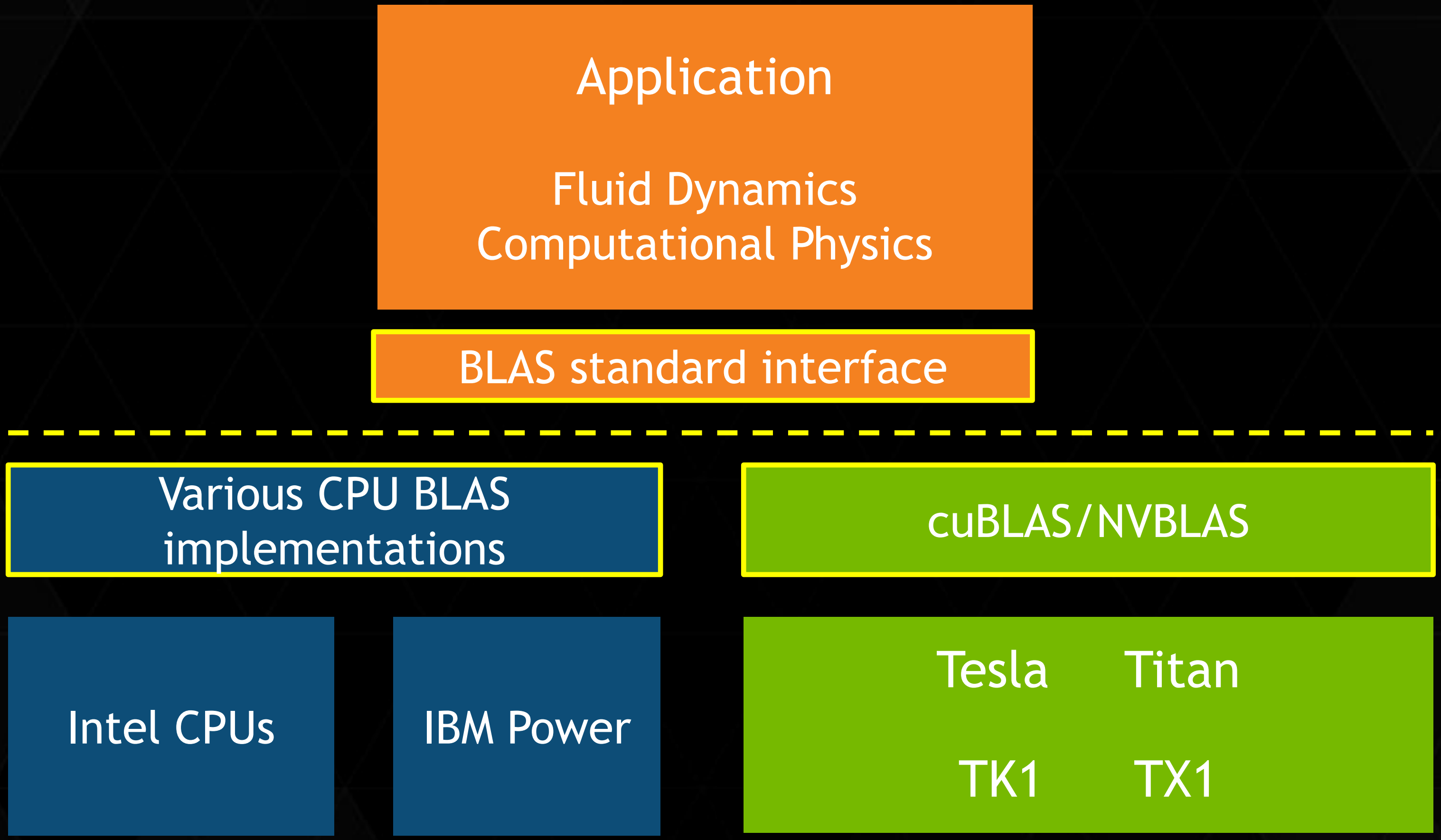
Maximum Flexibility

"Drop-in" Acceleration

Easily Accelerate Applications

ANALOGY TO HPC

cuDNN is a library of primitives for deep learning



Application

Fluid Dynamics
Computational Physics

BLAS standard interface

Various CPU BLAS implementations

cuBLAS/NVBLAS

Intel CPUs

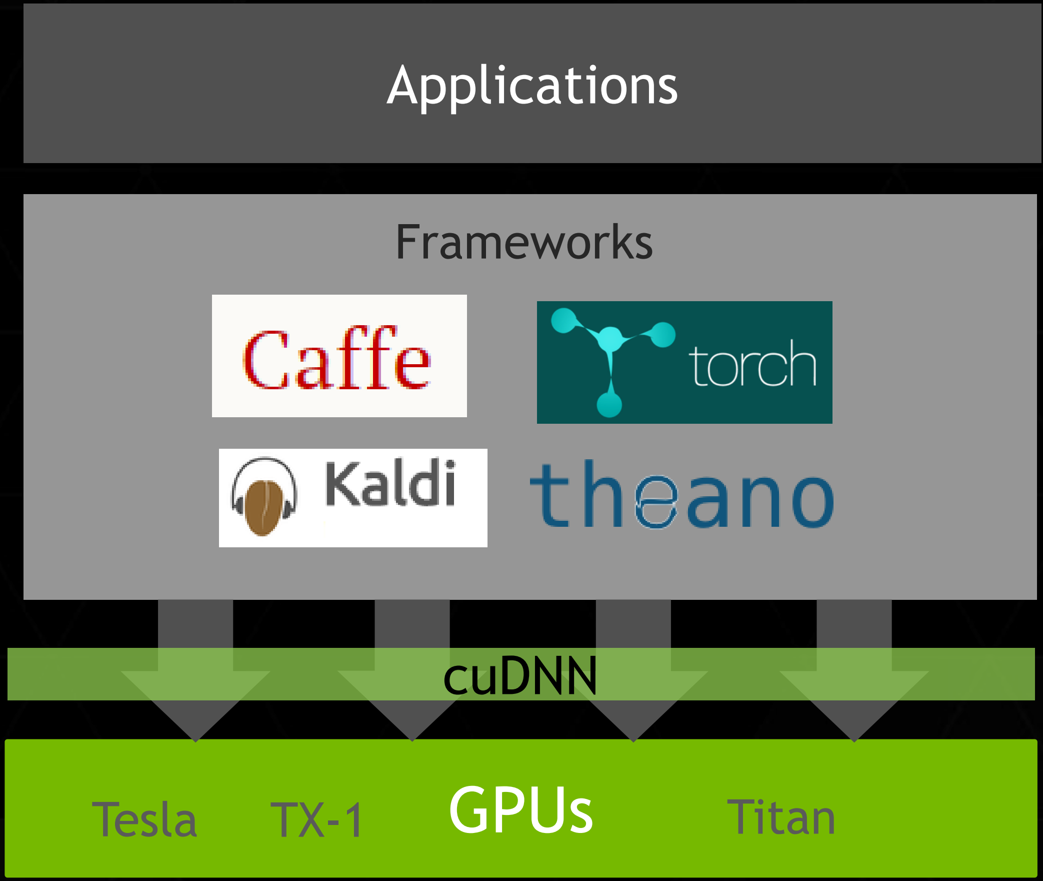
IBM Power

Tesla Titan

TK1 TX1

DEEP LEARNING WITH CUDNN

cuDNN is a library of primitives for deep learning



ANNOUNCING cuDNN V2

cuDNN V2 is focused on …

Performance and,

Features

… for the deep learning practitioner!

Optimized for current and future GPUs

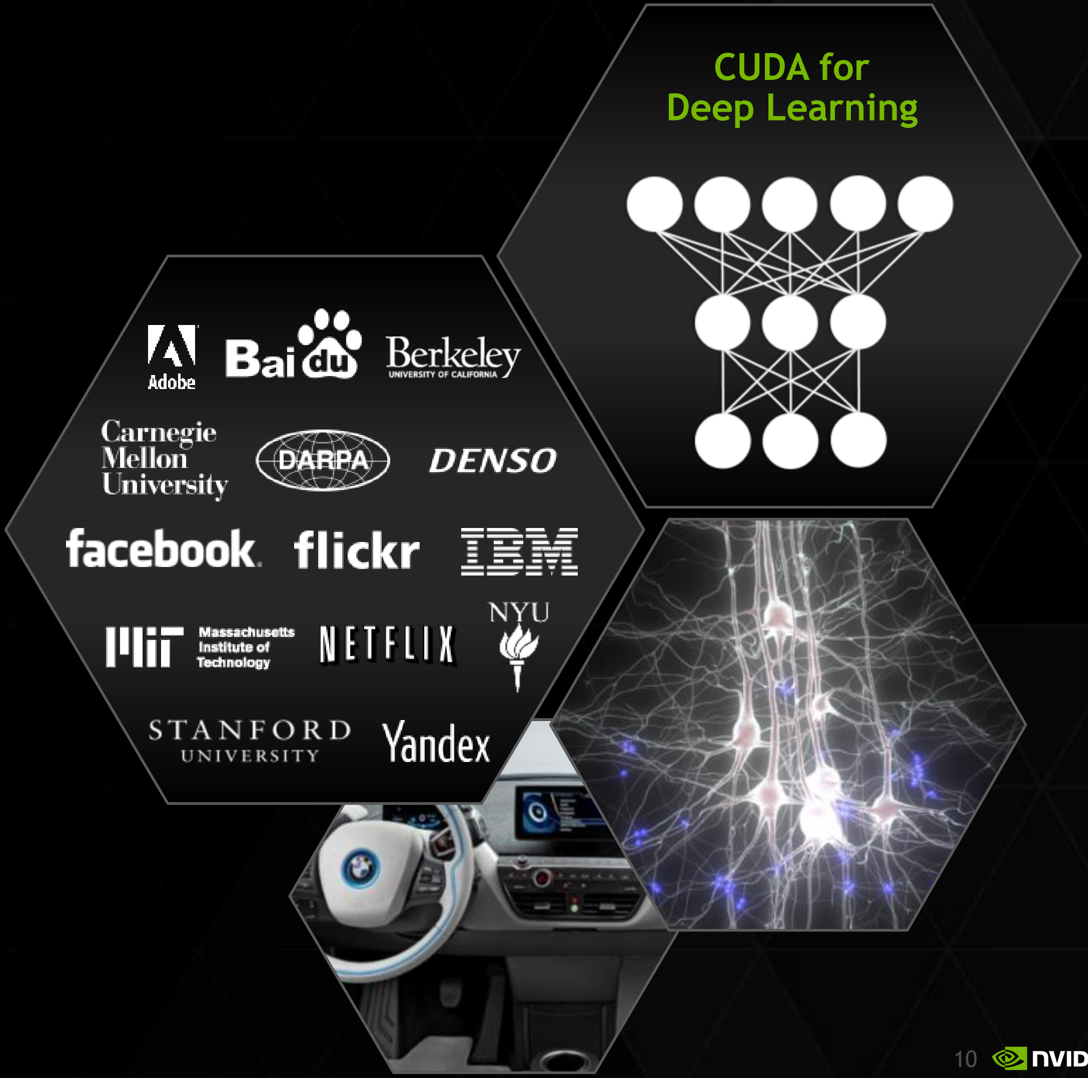
Deep Learning Context

ACCELERATING MACHINE LEARNING

“Machine Learning” is in some sense a rebranding of AI.

The focus is now on more specific, often perceptual tasks, and there are many successes.

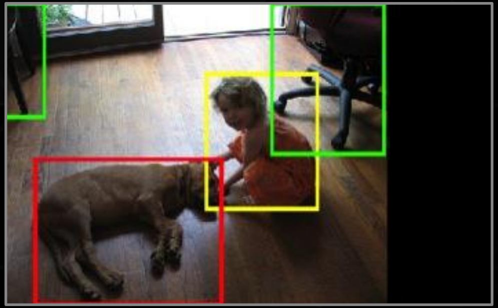
Today, some of the world’s largest internet companies, as well as the foremost research institutions, are using GPUs for machine learning.



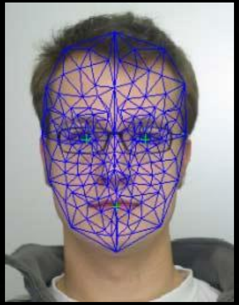
MACHINE LEARNING USE CASES

...machine learning is pervasive

Image Classification, Object Detection, Localization

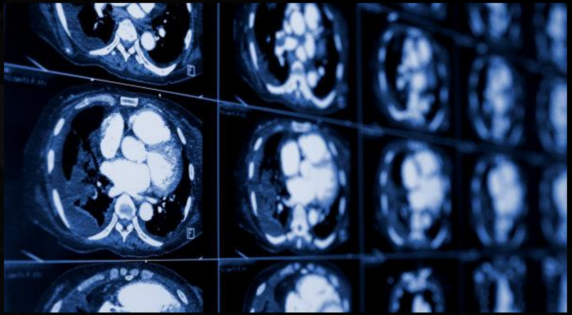


Face Recognition

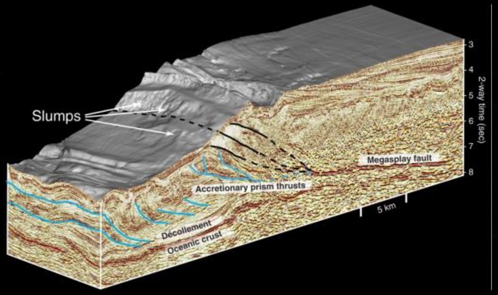


Speech & Natural Language Processing

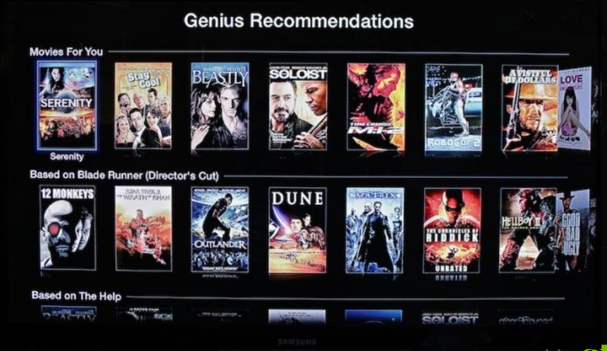
Medical Imaging & Interpretation



Seismic Imaging & Interpretation



Recommendation



Genius Recommendations

Movies For You

Serenity

Based on Blade Runner (Director's Cut)

Based on The Help

WHY IS DEEP LEARNING HOT *NOW*?

THREE DRIVING FACTORS…

1 - Big Data Availability

facebook 350 millions images uploaded per day

Walmart 2.5 Petabytes of customer data hourly

YouTube 100 hours of video uploaded every minute

2 - New ML Techniques

Deep Neural Networks

3 - Compute Density

GPUs

ML systems extract value from Big Data

DIFFERENT MODALITIES…SAME APPROACH

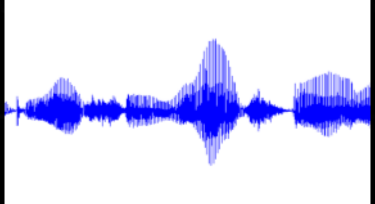
Images/video

Image

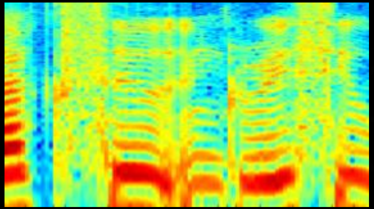
Vision features

Detection

Audio



Audio

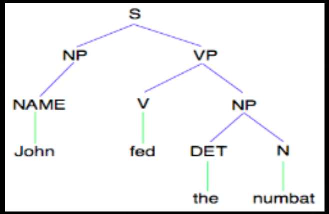


Audio features

Speaker ID

Text

Text



Text features

Text classification, Machine translation, Information retrieval, ...

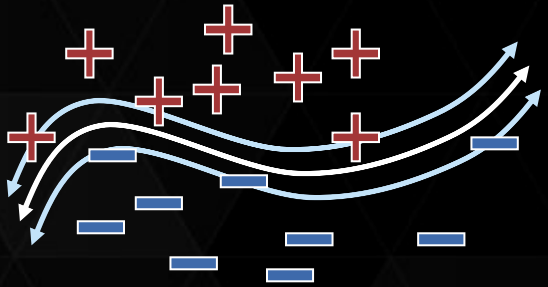
Slide courtesy of Andrew Ng, Stanford University

DEEP LEARNING ADVANTAGES

Deep Learning

- Don't have to figure out the features ahead of time!
- Use same neural net approach for many different problems.
- Fault tolerant.
- Scales well.

Support Vector Machine



Linear classifier

Regression

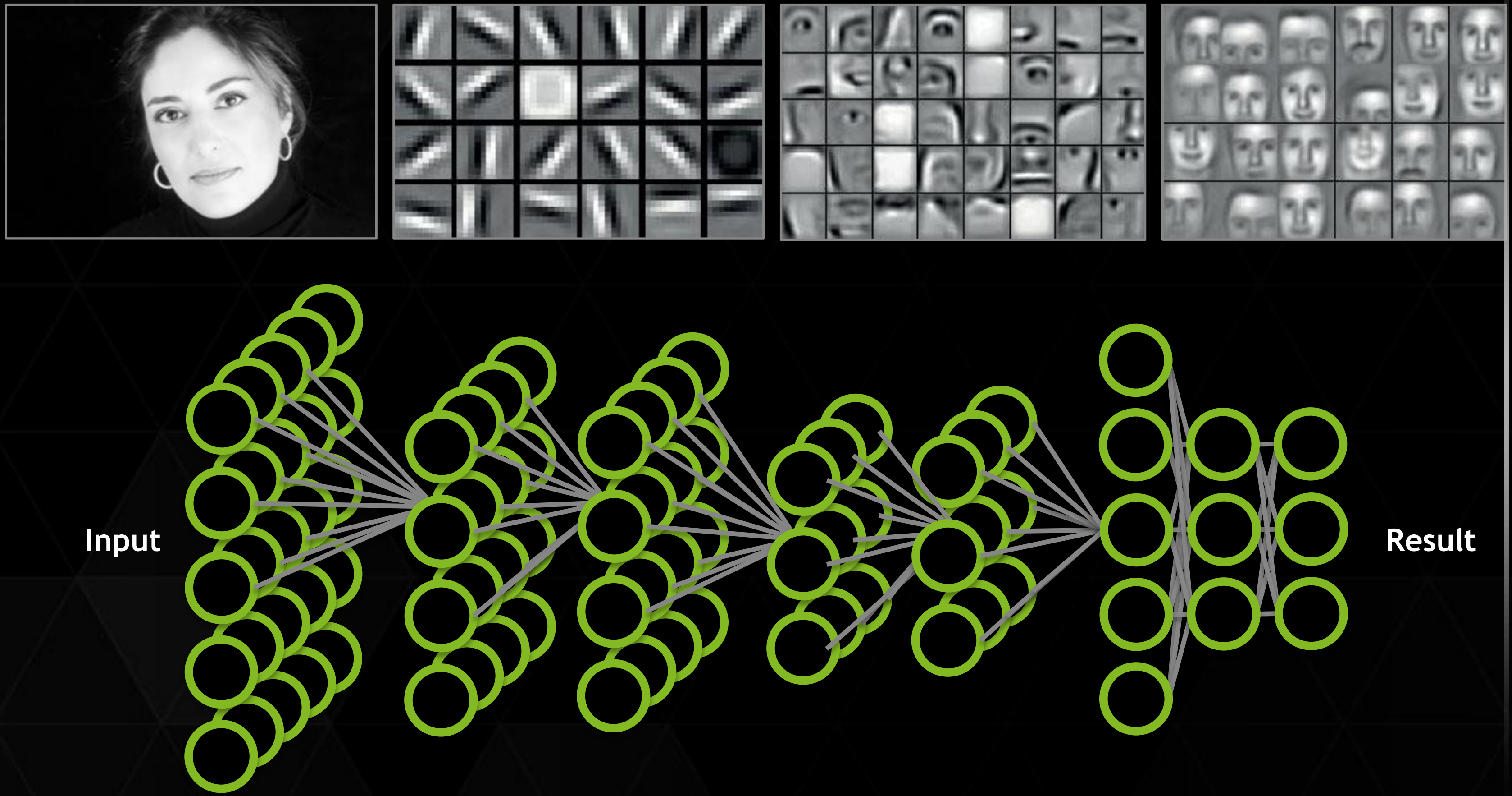
Decision Trees

Bayesian

Clustering

Association Rules

WHAT IS DEEP LEARNING?



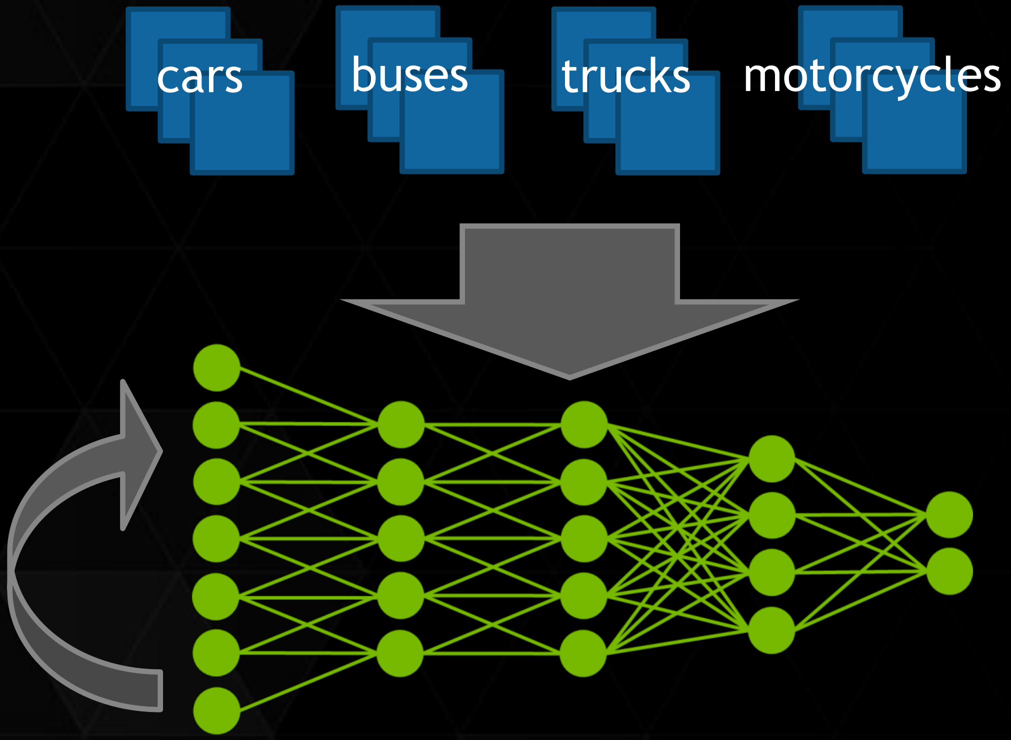
Today's Largest Networks

~10 layers
1B parameters
10M images
~30 Exaflops
~30 GPU days

Human brain has trillions of parameters – only 1,000 more.

CLASSIFICATION WITH DNNs

Training (Development)



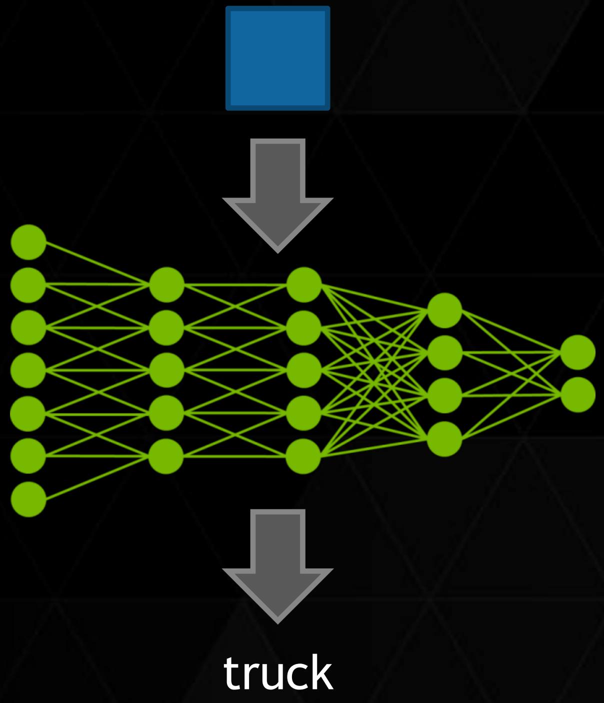
cars

buses

trucks

motorcycles

Inference (Production)

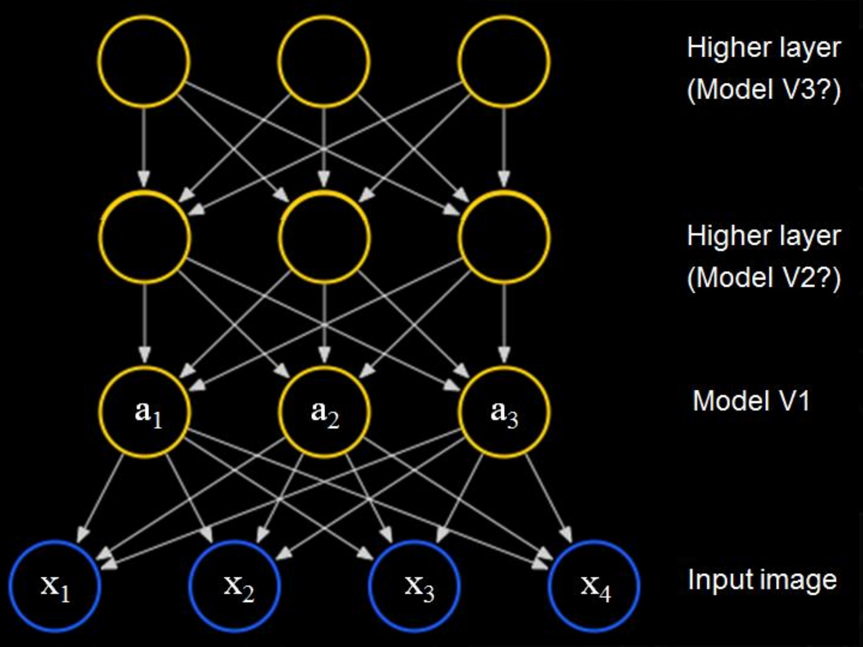


truck

WHY ARE GPUs GREAT FOR DEEP LEARNING?

	Neural Networks	GPUs
Inherently Parallel	✓	✓
Matrix Operations	✓	✓
FLOPS	✓	✓

- GPUs deliver --
  - same *or better* prediction accuracy
  - faster results
  - smaller footprint
  - lower power



[Lee, Ranganath & Ng, 2007]

CONVOLUTIONAL NEURAL NETWORKS

- Biologically inspired.

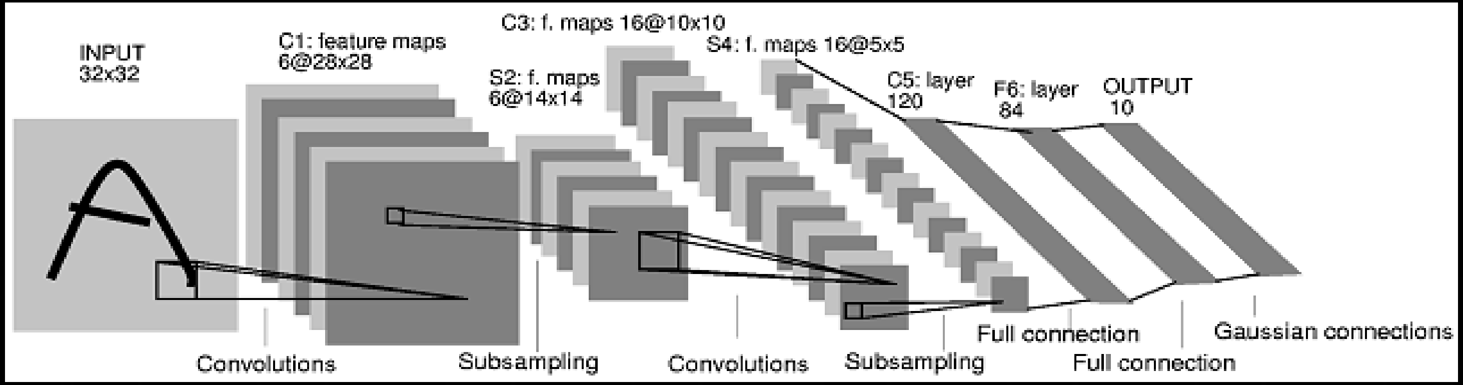
- Neuron only connected to a small region of neurons in layer below it called the *filter* or *receptive field*.

- A given layer can have many convolutional filters/kernels. Each filter has the same weights across the whole layer.

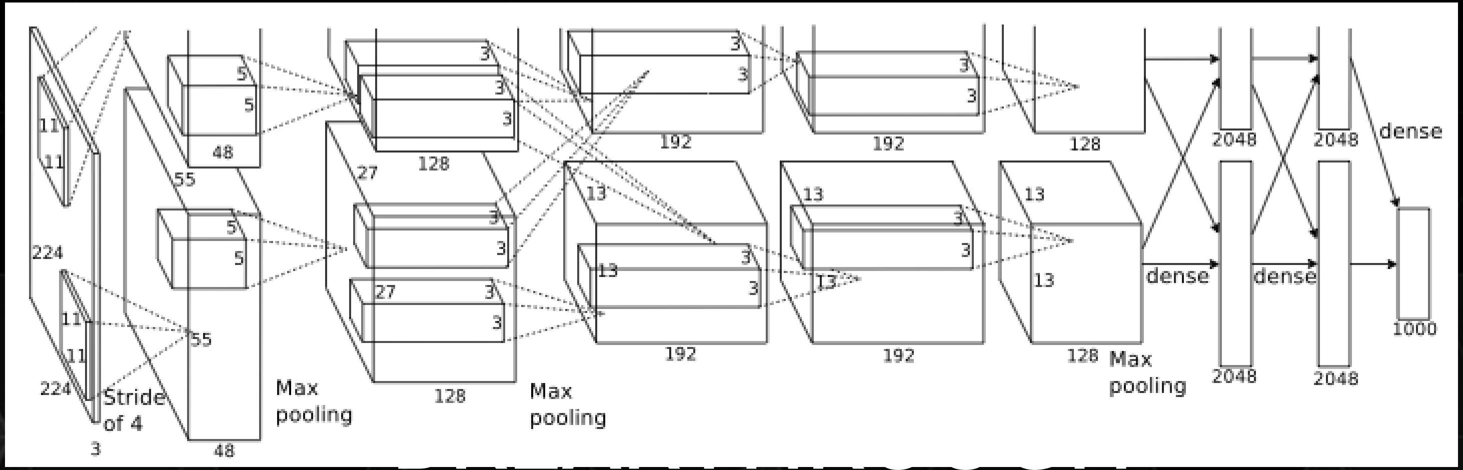
- Bottom layers are convolutional, top layers are fully connected.

- Generally trained via supervised learning.

CONVOLUTIONAL NET EXAMPLES

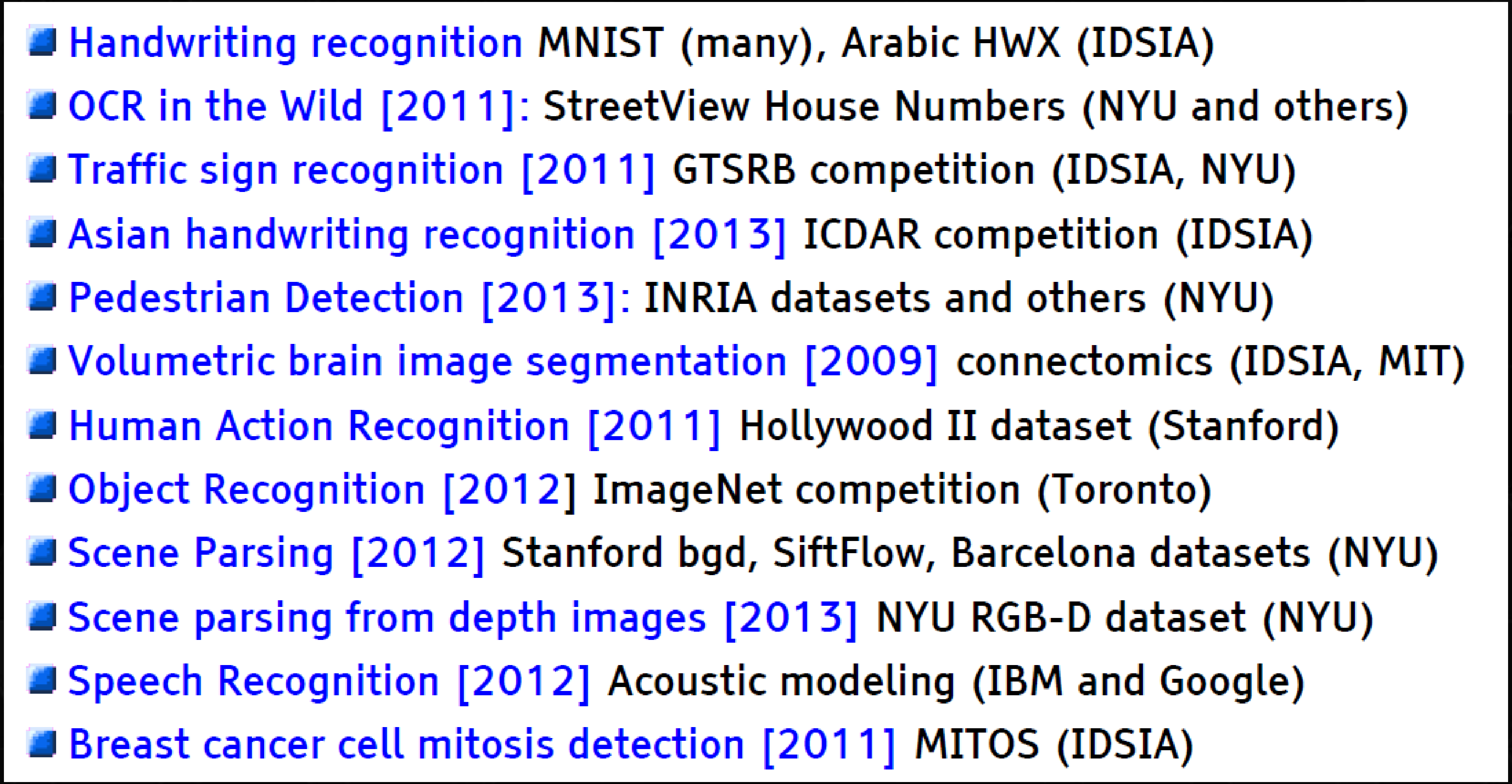


Y. LeCun et al. 1989-1998 : Handwritten digit reading



A. Krizhevsky, G. Hinton et al. 2012 : Imagenet classification winner

CNNS DOMINATE IN PERCEPTUAL TASKS



- Handwriting recognition MNIST (many), Arabic HWX (IDSIA)
- OCR in the Wild [2011]: StreetView House Numbers (NYU and others)
- Traffic sign recognition [2011] GTSRB competition (IDSIA, NYU)
- Asian handwriting recognition [2013] ICDAR competition (IDSIA)
- Pedestrian Detection [2013]: INRIA datasets and others (NYU)
- Volumetric brain image segmentation [2009] connectomics (IDSIA, MIT)
- Human Action Recognition [2011] Hollywood II dataset (Stanford)
- Object Recognition [2012] ImageNet competition (Toronto)
- Scene Parsing [2012] Stanford bgd, SiftFlow, Barcelona datasets (NYU)
- Scene parsing from depth images [2013] NYU RGB-D dataset (NYU)
- Speech Recognition [2012] Acoustic modeling (IBM and Google)
- Breast cancer cell mitosis detection [2011] MITOS (IDSIA)

Slide credit: Yann Lecun, Facebook & NYU

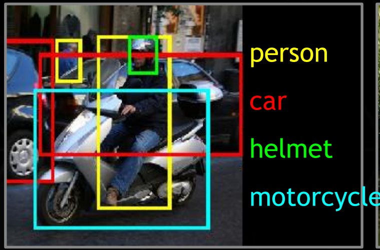
GPUs – *THE* PLATFORM FOR MACHINE LEARNING

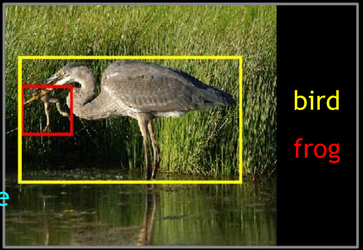
Image Recognition Challenge

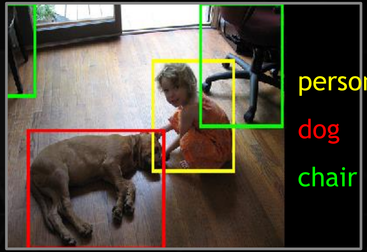
1.2M training images • 1000 object categories

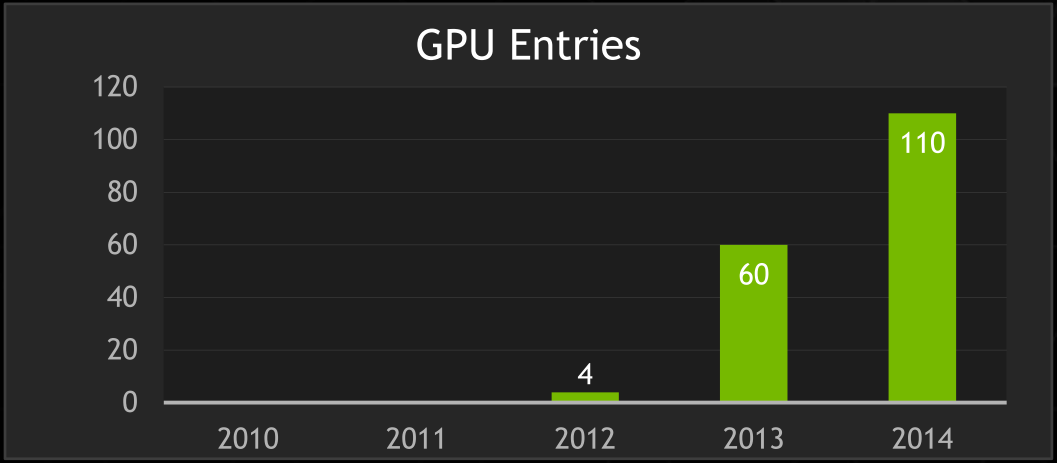
Hosted by

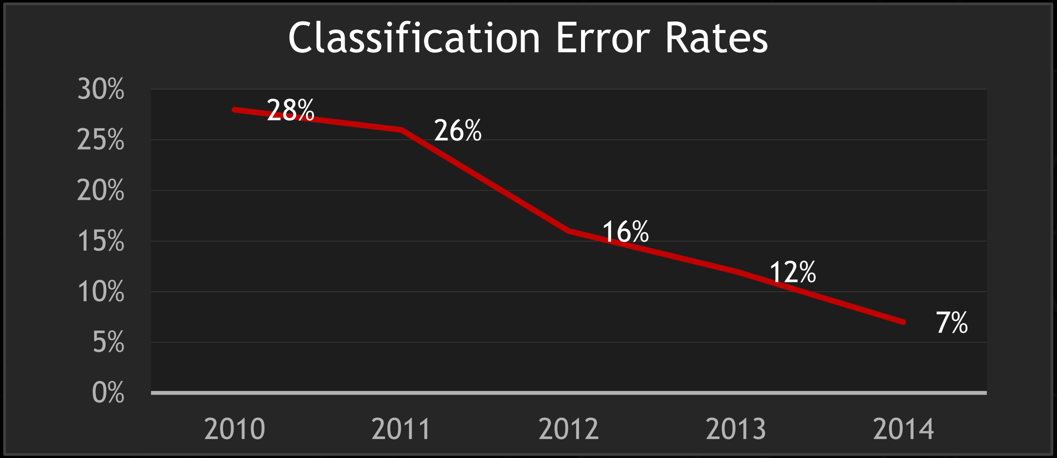
IMAGENET











GPUS MAKE DEEP LEARNING ACCESSIBLE

Deep learning with COTS HPC systems

A. Coates, B. Huval, T. Wang, D. Wu, A. Ng, B. Catanzaro

ICML 2013

> *“Now You Can Build Google’s $1M Artificial Brain on the Cheap”*

WIRED

GOOGLE DATACENTER

1,000 CPU Servers
2,000 CPUs • 16,000 cores

600 kWatts
$5,000,000

STANFORD AI LAB

3 GPU-Accelerated Servers
12 GPUs • 18,432 cores

4 kWatts
$33,000

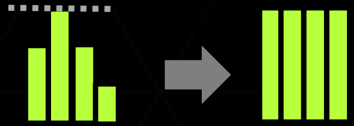
cuDNN version 2

CUDNN DESIGN GOALS

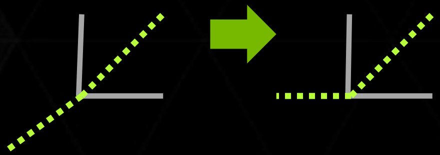
- Basic Deep Learning Subroutines
  - Allow user to write a DNN application without any custom CUDA code
- Flexible Layout
  - Handle any data layout
- Memory – Performance tradeoff
  - Good performance with minimal memory use, great performance with more memory use

cuDNN ROUTINES

- Convolutions – 80-90% of the execution time
- Pooling - Spatial smoothing



- Activation - Pointwise non-linear function



CONVOLUTIONS – THE MAIN WORKLOAD

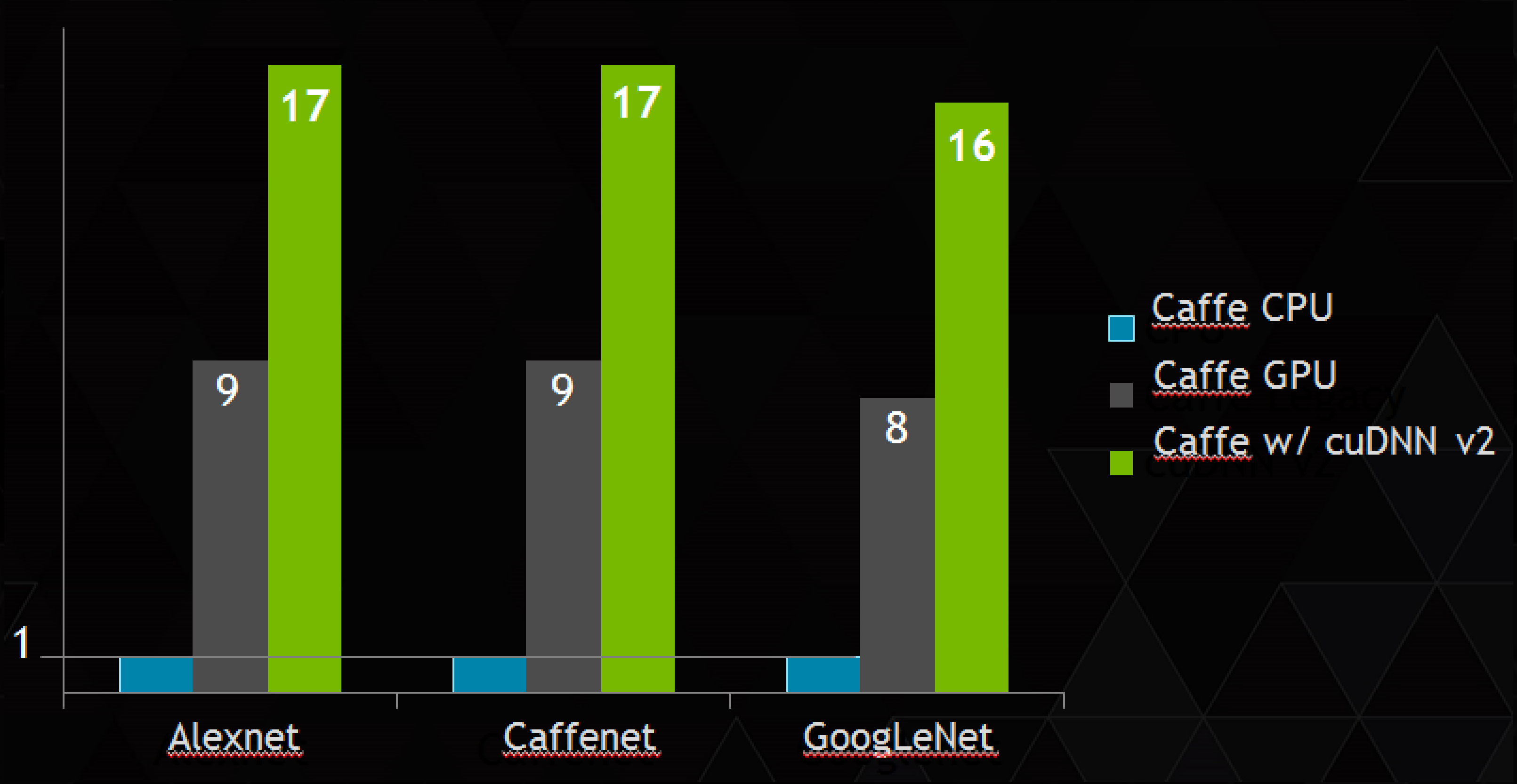
- Very compute intensive, but with a large parameter space

1 Minibatch Size
2 Input feature maps
3 Image Height
4 Image Width
5 Output feature maps
6 Kernel Height
7 Kernel Width
8 Top zero padding
9 Side zero padding
10 Vertical stride
11 Horizontal stride

- Layout and configuration variations

- Other cuDNN routines have straightforward implementations

CUDNN V2 - PERFORMANCE



CPU is 16 core Haswell E5-2698 at 2.3 GHz, with 3.6 GHz Turbo

GPU is NVIDIA Titan X

CUDNN V2 FLEXIBILITY

Can now specify a strategy the library will use to select the best convolution algorithm:

```
PREFER_FASTEST

NO_WORKSPACE

SPECIFY_WORKSPACE_LIMIT
```

...or specify an algorithm directly...

```
GEMM

IMPLICIT_GEMM

IMPLICIT_PRECOMP_GEMM

DIRECT
```

CUDNN V2 NEW FEATURES

Other key new features:

- Support for 3D datasets. Community feedback desired!
- OS X support
- Zero-padding of borders in pooling routines
- Parameter scaling
- Improved support for arbitrary strides
- Support for upcoming Tegra X1 via JIT compilation

See Release Notes for details…

CUDNN V2 API CHANGES

Important – API Has Changed

- Several of the new improvements required changes to the cuDNN API.
- Applications previously using cuDNN V1 are likely to need minor modifications.
- Note Im2Col function is currently exposed public function…but will be removed.

The cuDNN team genuinely appreciates all feedback from the Deep learning community.

The team carefully considers any API change.

cuDNN is still young…API changes expected to become rare in the future.

Using cuDNN

CUDNN EASY TO ENABLE

Caffe

- Install cuDNN on your system
- Download CAFFE
- In CAFFE `Makefile.config`
  - uncomment `USE_CUDNN := 1`
- Install CAFFE as usual
- Use CAFFE as usual.

torch

- Install cuDNN on your system
- Install Torch as usual
- Install `cudnn.torch` module
- Use `cudnn` module in Torch instead of regular `nn` module.
- `cudnn` module is API compatable with standard `nn` module.
  Replace `nn` with `cudnn`

CUDA 6.5 or newer required

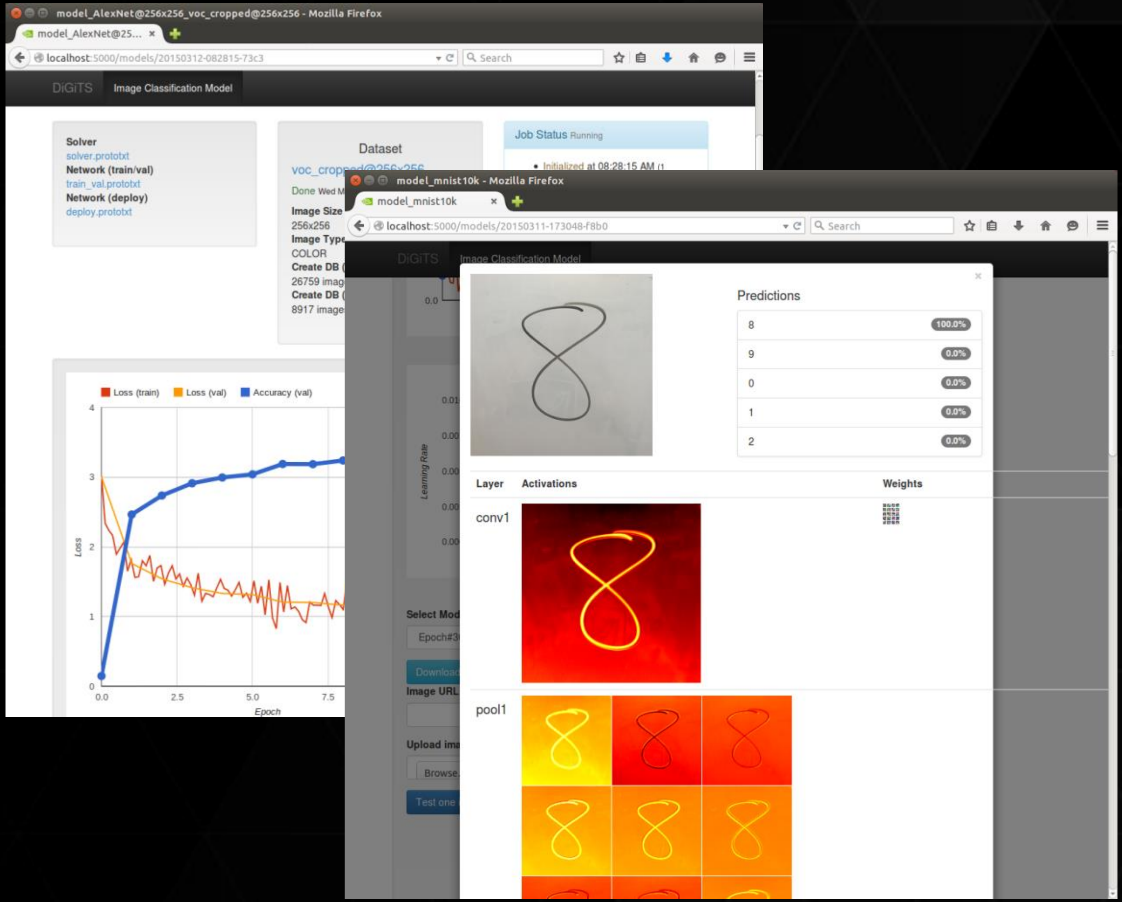
DIGITS

Interactive Deep Learning GPU Training System

Data Scientists & Researchers:

- Quickly design the best deep neural network (DNN) for your data
- Visually monitor DNN training quality in real-time
- Manage training of many DNNs in parallel on multi-GPU systems

developer.nvidia.com/digits



model_AlexNet@256x256_voc_cropped@256x256 - Mozilla Firefox

localhost:5000/models/20150312-082815-73c3

Solver: solver.prototxt
Network (train/val): train_val.prototxt
Network (deploy): deploy.prototxt

Dataset: voc_cropped@256x256
Done Wed M
Image Size: 256x256
Image Type: COLOR
Create DB (: 26759 imag
Create DB (: 8917 image

Job Status Running
• Initialized at 08:28:15 AM (1

model_mnist10k - Mozilla Firefox

localhost:5000/models/20150311-173048-f8b0

Predictions

8	100.0%
9	0.0%
0	0.0%
1	0.0%
2	0.0%

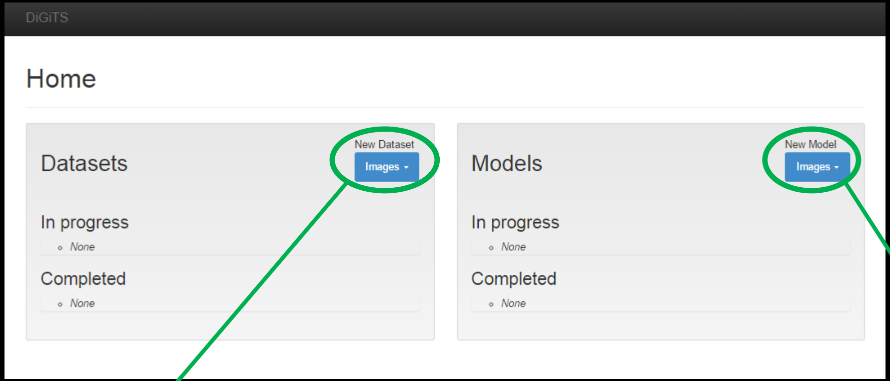
DIGITS Workflow

Create your database

Configure your model

Start training

Main Console



DiGiTS

Home

Datasets

New Dataset Images ▾

In progress
- None

Completed
- None

Models

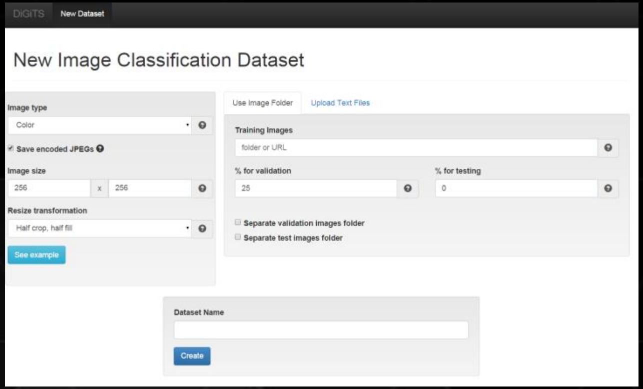
New Model Images ▾

In progress
- None

Completed
- None

Configure your Network

Create your dataset



DiGiTS New Dataset

New Image Classification Dataset

Image type: Color

☑ Save encoded JPEGs

Image size: 256 x 256

Resize transformation: Half crop, half fill

See example

Use Image Folder | Upload Text Files

Training Images: folder or URL

% for validation: 25

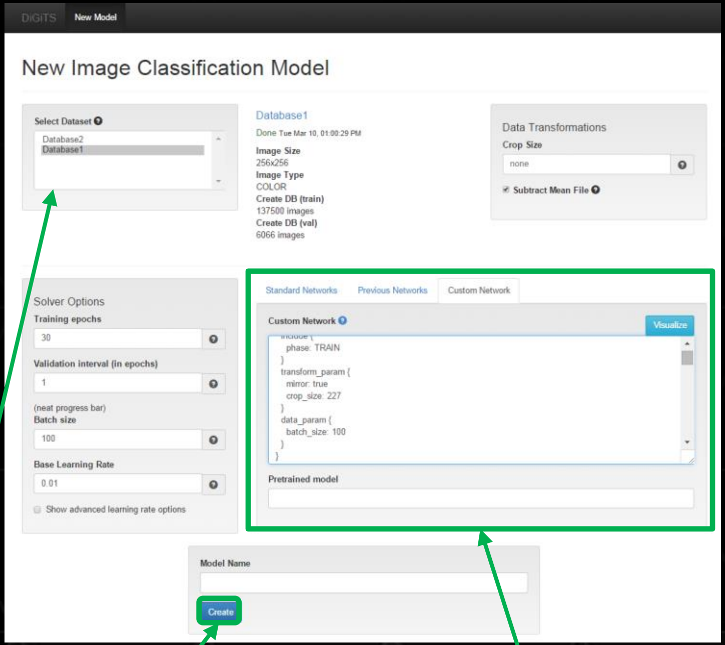
% for testing: 0

□ Separate validation images folder

□ Separate test images folder

Dataset Name

Create



DiGiTS New Model

New Image Classification Model

Select Dataset: Database2, Database1

Database1

Done Tue Mar 10, 01:00:29 PM

Image Size
256x256

Image Type
COLOR

Create DB (train)
137500 images

Create DB (val)
6066 images

Data Transformations

Crop Size: none

☑ Subtract Mean File

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

Standard Networks | Previous Networks | Custom Network

Custom Network

Visualize

```
    include {
      phase: TRAIN
    }
    transform_param {
      mirror: true
      crop_size: 227
    }
    data_param {
      batch_size: 100
    }
}
```

Pretrained model

Model Name

Create

Choose your database

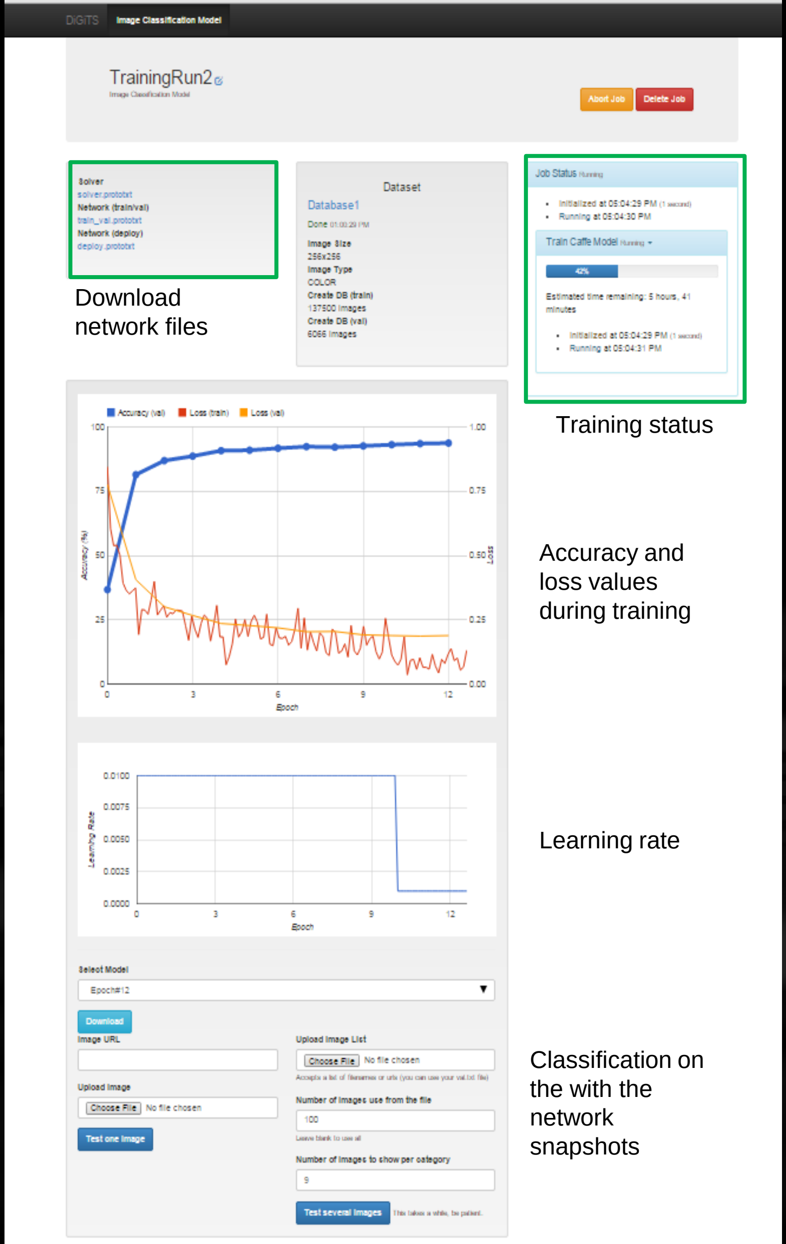
Start Training

Choose a default network, modify one, or create your own

DIGITS

Visualize DNN performance in real time

Compare networks



TrainingRun2
Image Classification Model

Solver
solver.prototxt
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt

Dataset
Database1
Done 01:00:29 PM
Image Size
256x256
Image Type
COLOR
Create DB (train)
137500 images
Create DB (val)
6066 images

Job Status Running

- Initialized at 05:04:29 PM (1 second)
- Running at 05:04:30 PM

Train Caffe Model Running

42%

Estimated time remaining: 5 hours, 41 minutes

- Initialized at 05:04:29 PM (1 second)
- Running at 05:04:31 PM

Select Model
Epoch#12

Image URL

Upload Image
No file chosen

Upload Image List
No file chosen
Accepts a list of filenames or urls (you can use your val.txt file)

Number of images use from the file
100
Leave blank to use all

Number of images to show per category
9

This takes a while, be patient.

Download network files

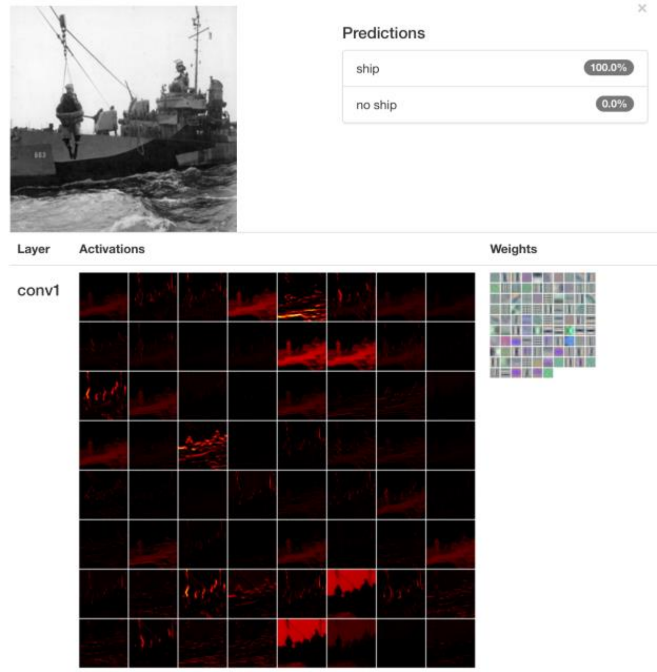
Training status

Accuracy and loss values during training

Learning rate

Classification on the with the network snapshots

Classification



Predictions

ship	100.0%
no ship	0.0%

NVIDIA®

cuDNN

developer.nvidia.com/cuDNN

Try it today!