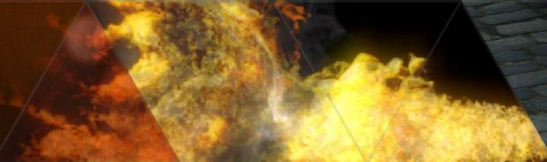
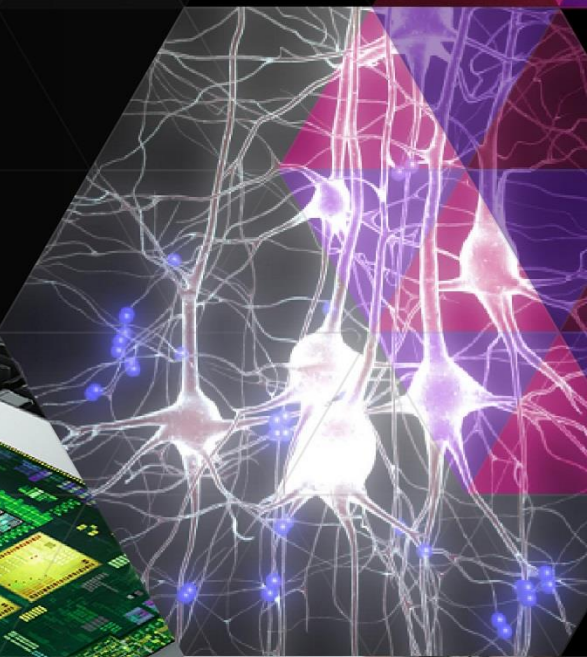
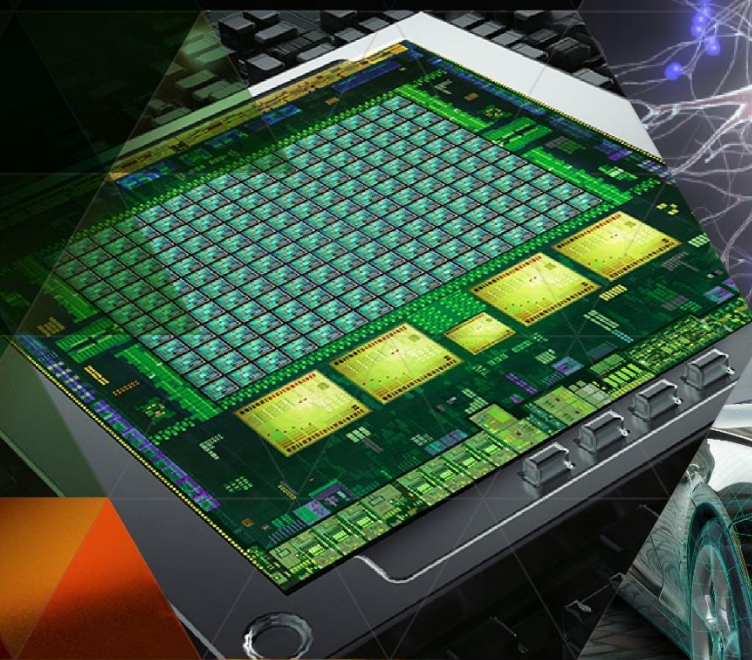




DIGITS DEEP LEARNING GPU TRAINING SYSTEM

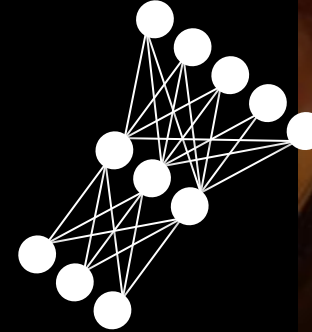
Allison Gray



PRACTICAL DEEP LEARNING EXAMPLES



Image Classification, Object Detection, Localization, Action Recognition, Scene Understanding



Speech Recognition, Speech Translation, Natural Language Processing



Pedestrian Detection, Traffic Sign Recognition



Breast Cancer Cell Mitosis Detection, Volumetric Brain Image Segmentation

WHAT IS DEEP LEARNING?

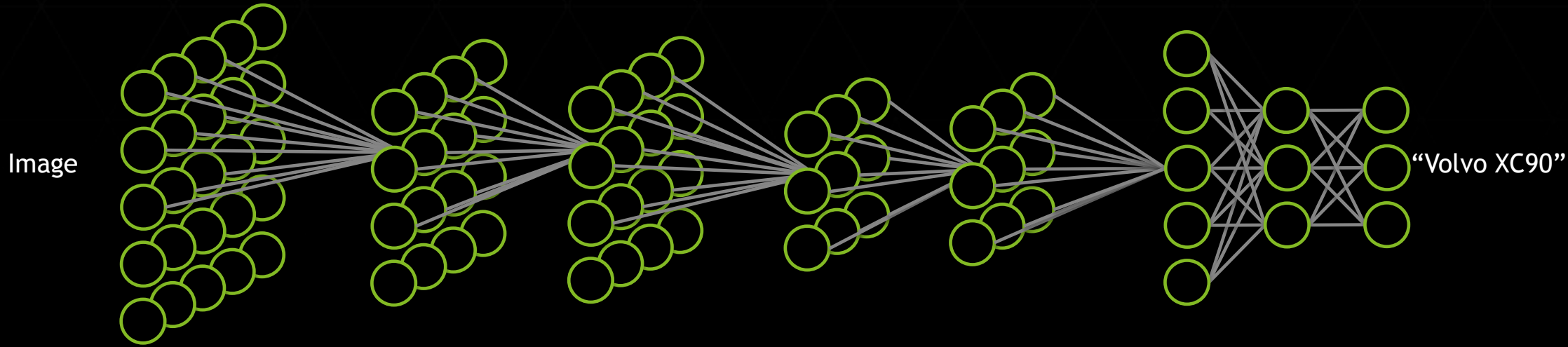
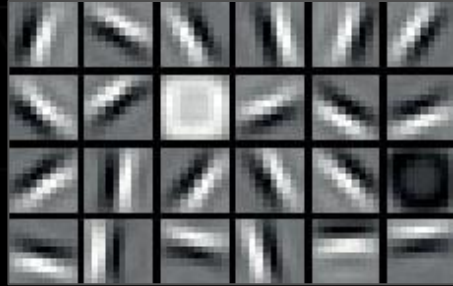


IMAGE CLASSIFICATION WITH DNN

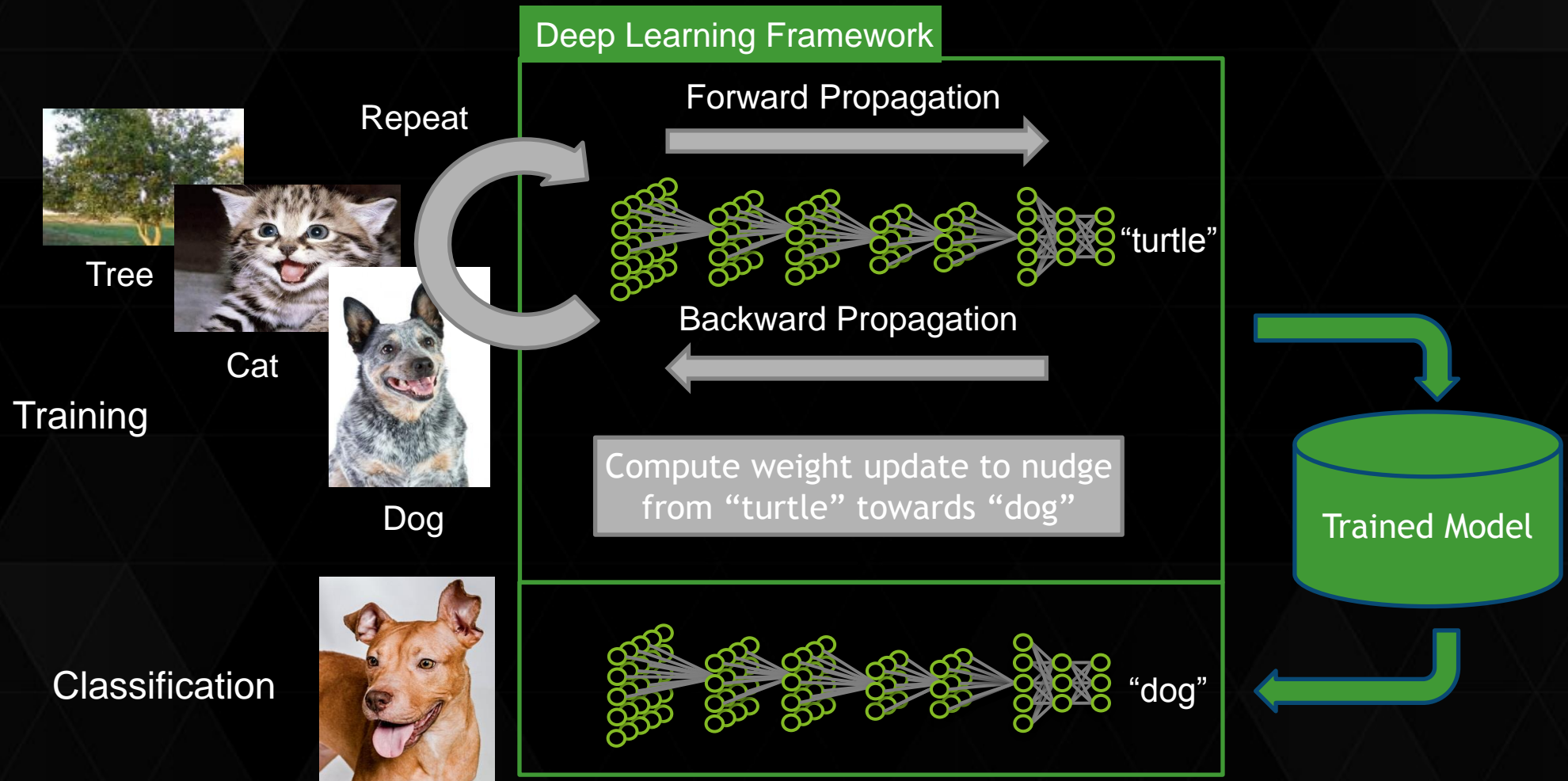


IMAGE CLASSIFICATION WITH DNNs



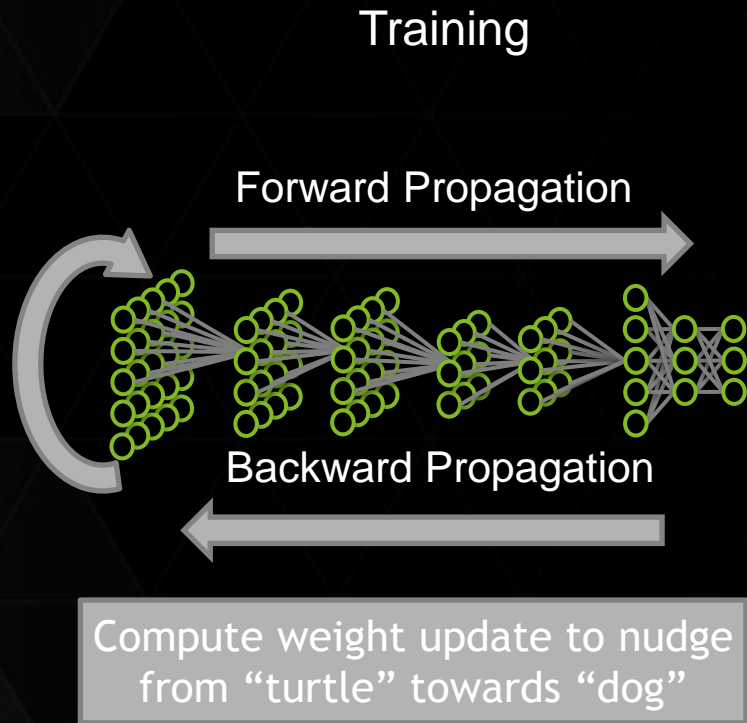
Tree



Cat



Dog



► Typical training run

- Pick a DNN design
- Input thousands to millions training images spanning 1,000 or more categories
- *Day to a week of computation*

► Test accuracy

- If bad: modify DNN, fix training set or update training parameters

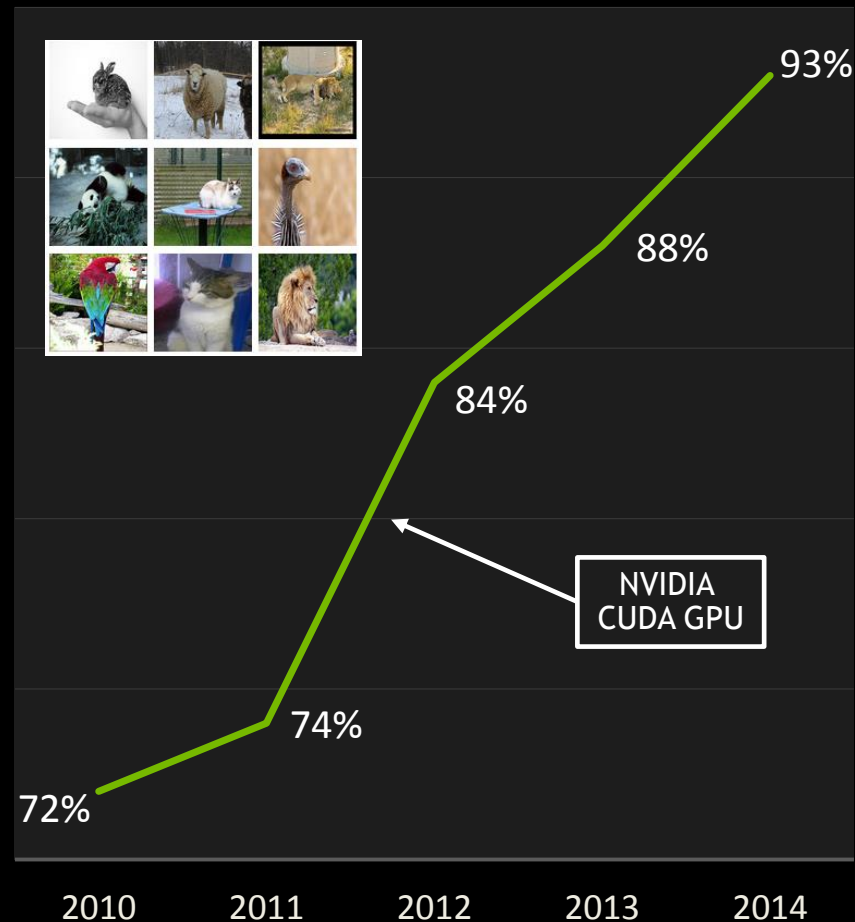
GPUs and Deep Learning

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

GPUs deliver --

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

ImageNet Challenge
Accuracy



NVIDIA DEEP LEARNING PLATFORM

DEVELOPMENT

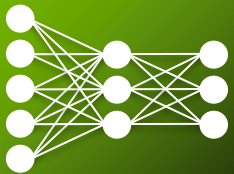
DEPLOYMENT

Applications

DIGITS Tools

Deep Learning Frameworks

Software



cuDNN

Systems



DIGITS DevBox

Hardware



Titan X

Software



System
Management

Systems



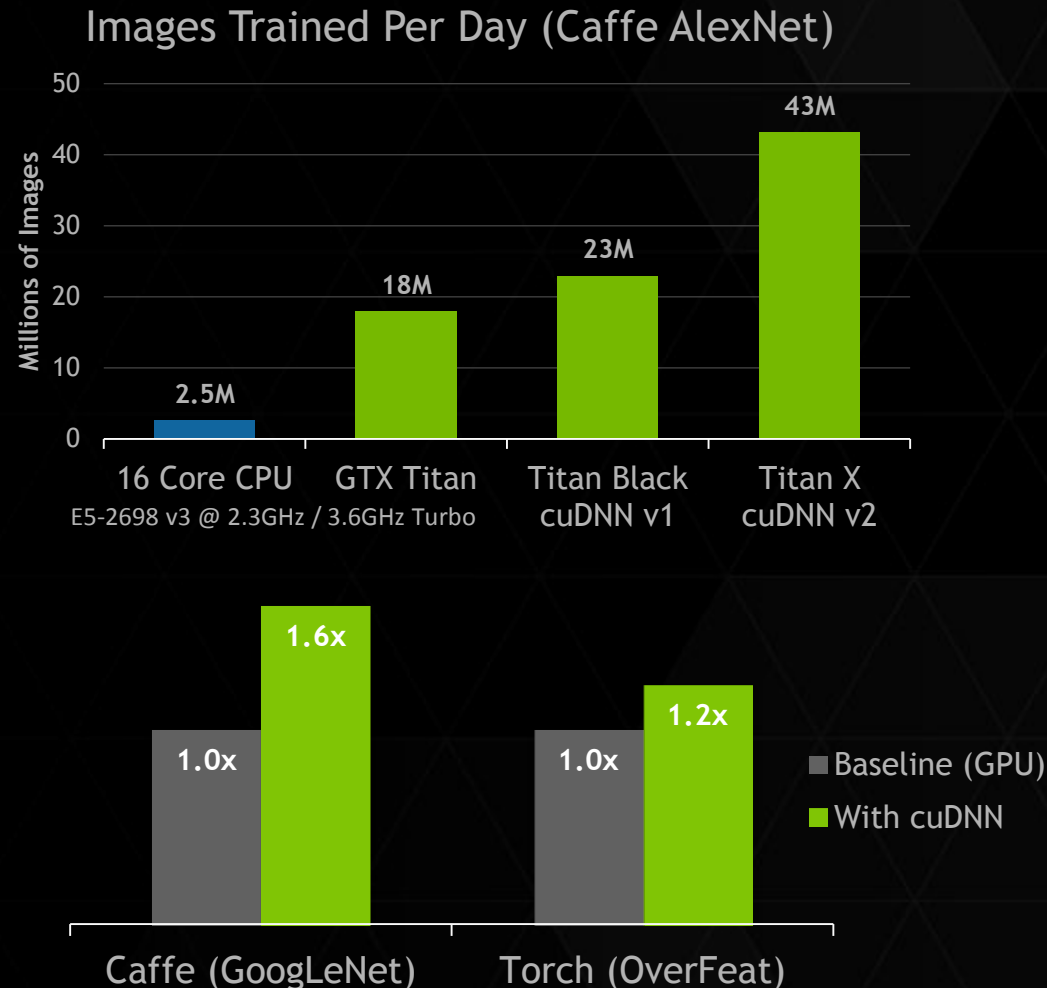
Hardware



Tesla

cuDNN

- ▶ Accelerates key routines to improve performance of neural net training
 - ▶ Routines for convolution and cross-correlation as well as activation functions
 - ▶ Up to 1.8x faster on AlexNet than a baseline GPU implementation
- ▶ Integrated into all major Deep Learning frameworks: Caffe, Theano, Torch

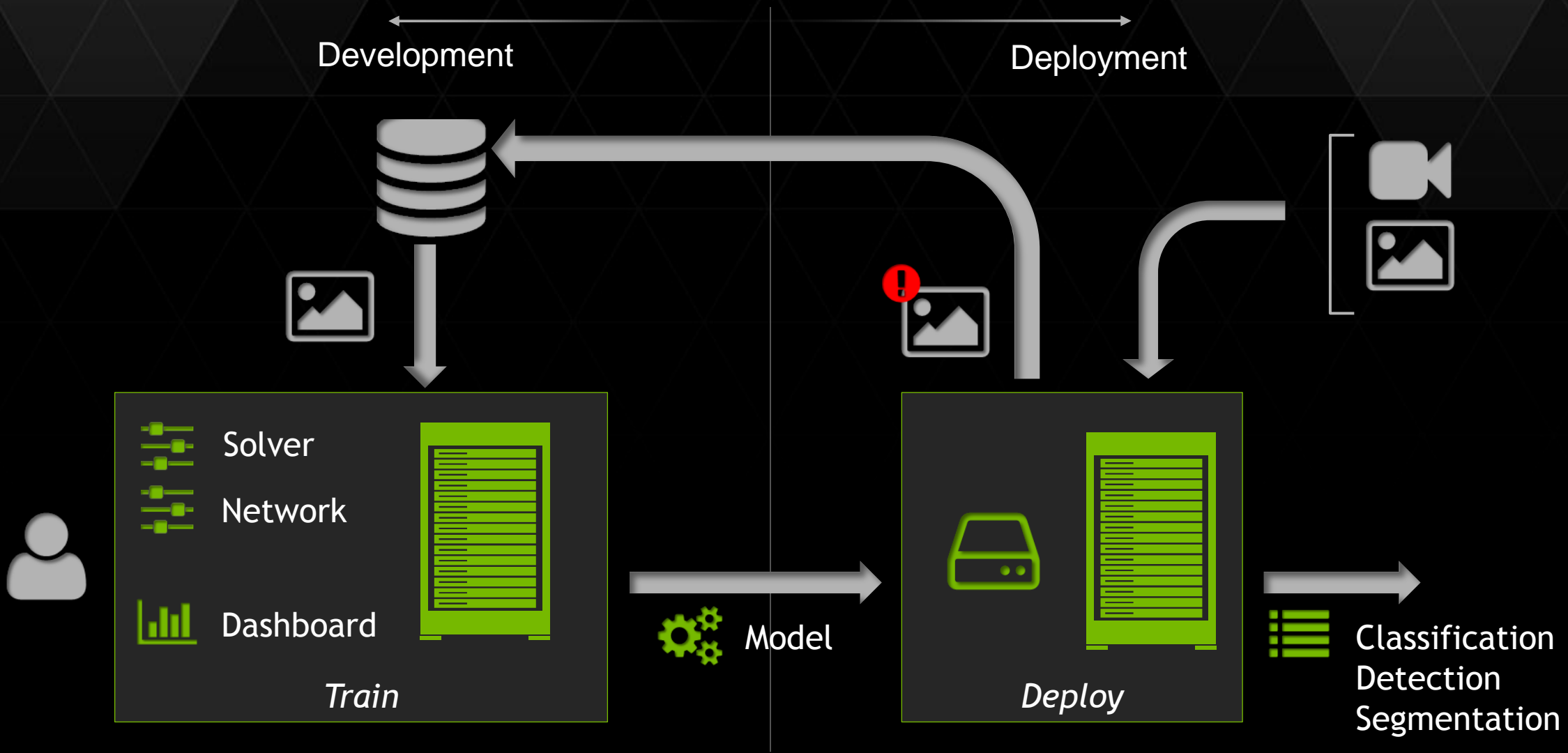


GPU-ACCELERATED DEEP LEARNING FRAMEWORKS

	CAFFE	TORCH	THEANO	CUDA-CONVNET2	KALDI
Domain	Deep Learning Framework	Scientific Computing Framework	Math Expression Compiler	Deep Learning Application	Speech Recognition Toolkit
cuDNN	R2	R2	R2	--	--
Multi-GPU	✓	✓	✓	✓	✓ (nnet2)
Multi-CPU	✗	✗	✗	✗	✓ (nnet2)
License	BSD-2	GPL	BSD	Apache 2.0	Apache 2.0
Interface(s)	Text-based definition files, Python, MATLAB	Python, Lua, MATLAB	Python	C++	C++, Shell scripts
Embedded (TK1)	✓	✓	✗	✗	✗

<http://developer.nvidia.com/deeplearning>

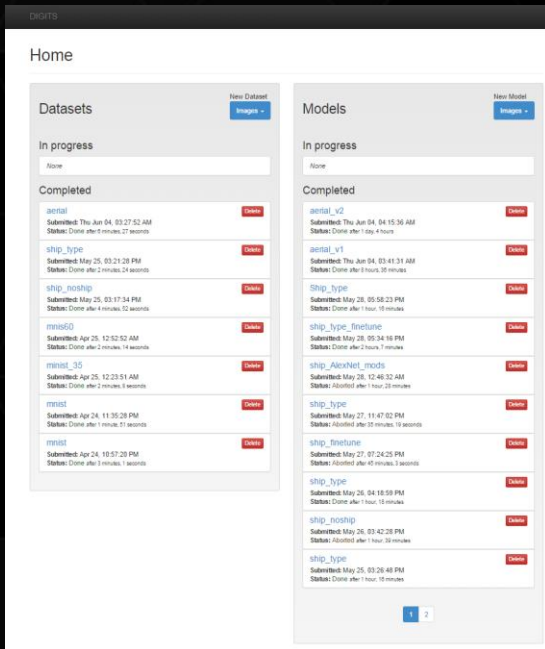
PRODUCTION PIPELINE



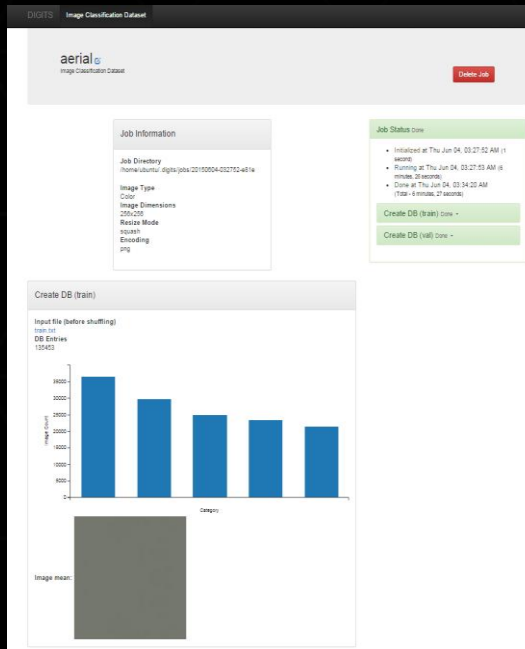
NVIDIA® DIGITS™

Interactive Deep Learning GPU Training System

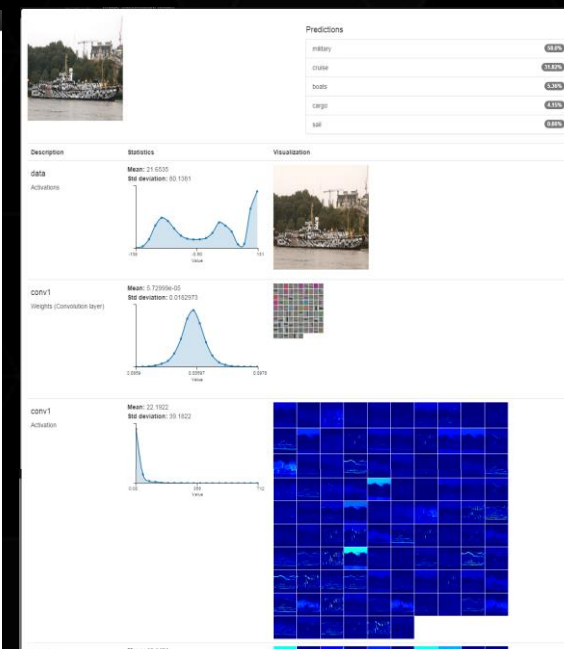
Dashboard



Real time monitoring

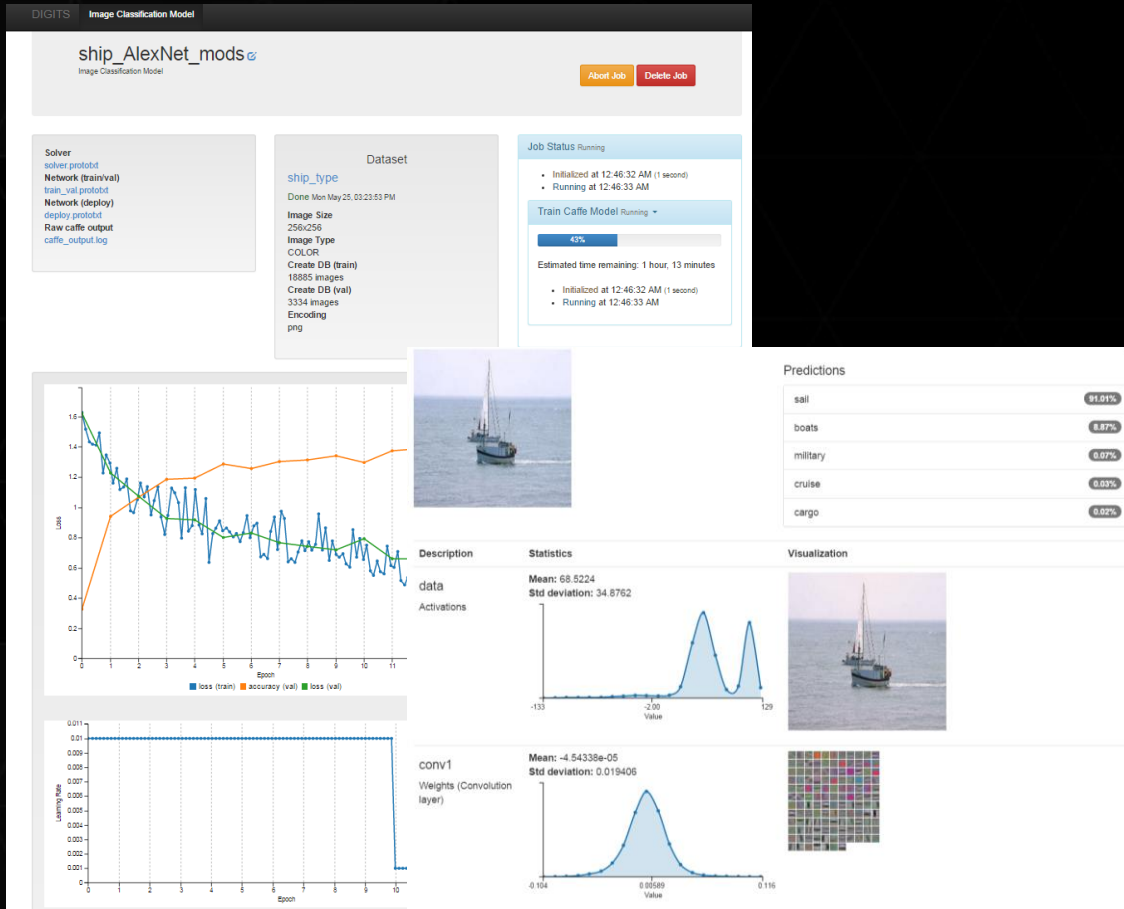


Network Visualization



NVIDIA® DIGITS™

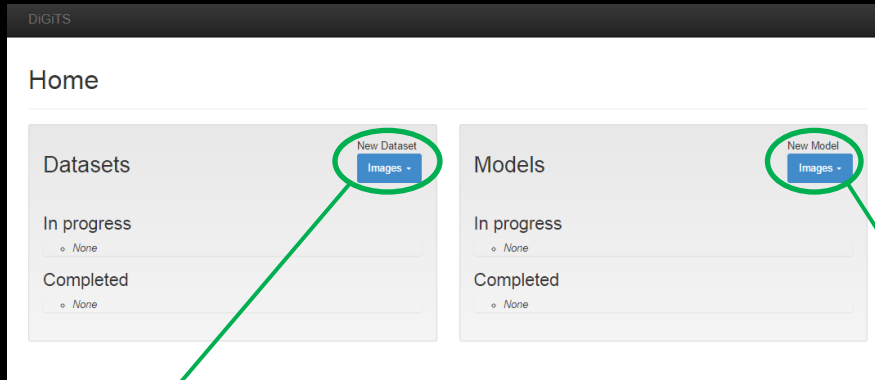
Interactive Deep Learning GPU Training System



Data Scientists & Researchers:

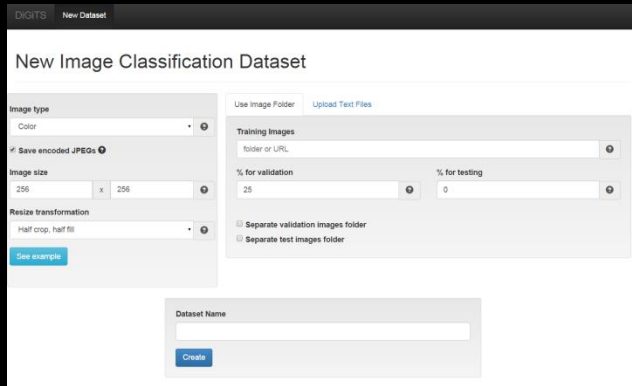
- Quickly design the best deep neural network (DNN) for your data
- Visually monitor DNN training quality in real-time
- Manage training of many DNNs in parallel on multi-GPU systems
- Open source!
- <https://developer.nvidia.com/digits>

Main Console

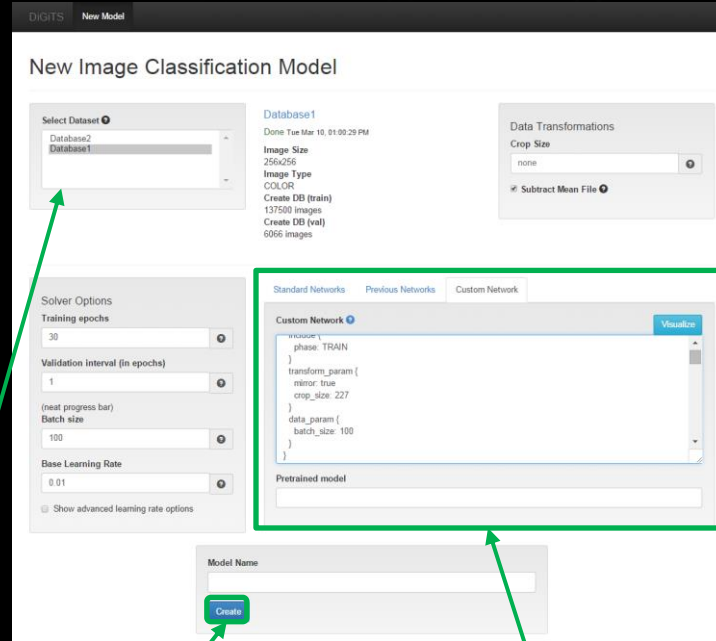


Configure your Network

Create your dataset



Choose your database



Start Training

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

DIGITS WORKFLOW

CREATE YOUR DATABASE
CONFIGURE YOUR MODEL

CREATING THE DATABASE

DIGITS New Dataset

New Image Classification Dataset


Image type
Color

☒ Save encoded JPEGs

Image size
256 x 256

Resize transformation
Squash

[See example](#)



Use Image Folder [Upload Text Files](#)

Training Images
/path/to/images

% for validation 25 % for testing 0

☐ Separate validation images folder
☐ Separate test images folder

DIGITS can automatically create your training and validation set

OR

Insert the path to your train and validation set

Set	Text file	Image folder (optional)
Training	Choose File train.txt	/path/to/train/
<input checked="" type="checkbox"/> Validation	Choose File val.txt	/path/to/val/
<input type="checkbox"/> Test	Choose File No file chosen	

Labels
[Choose File](#) synsets.txt

Image parameter options

Dataset Name
Database

[Create](#)

Create your dataset

OR use a URL list

CREATE THE DATABASE

DIGITS New Dataset

New Image Classification Dataset


Image type
Color

☒ Save encoded JPEGs

Image size
256 x 256

Resize transformation
Squash

[See example](#)



Use Image Folder [Upload Text Files](#)

Training Images
/path/to/images

% for validation: 25 % for testing: 0

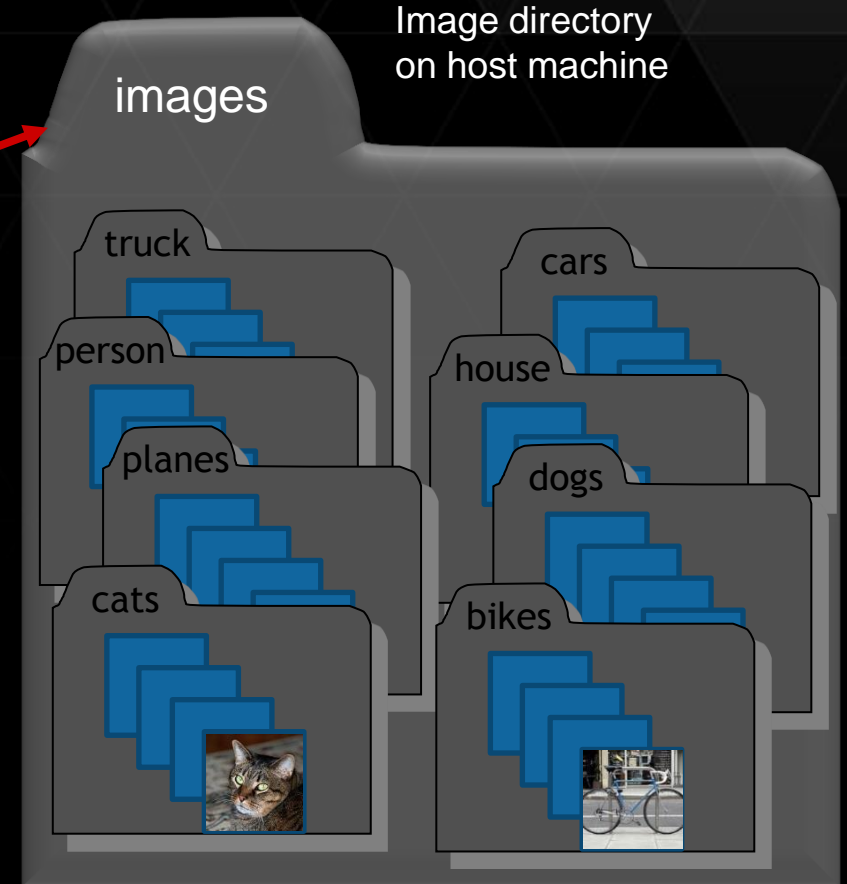
☐ Separate validation images folder ☐ Separate test images folder

Insert the path to your images here

Dataset Name
Database3

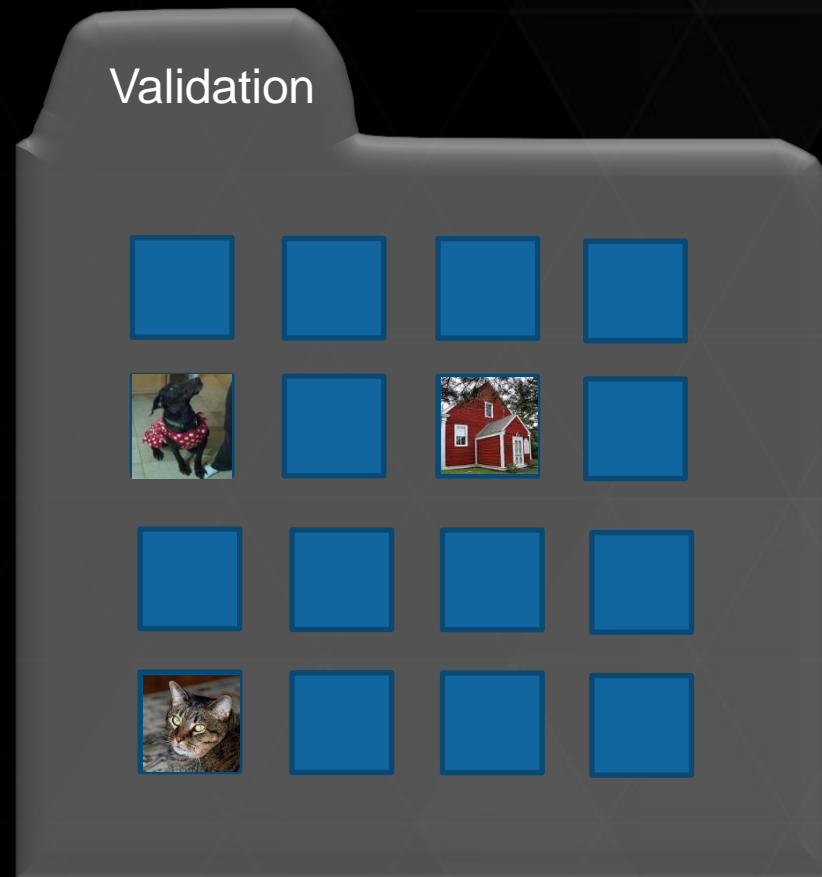
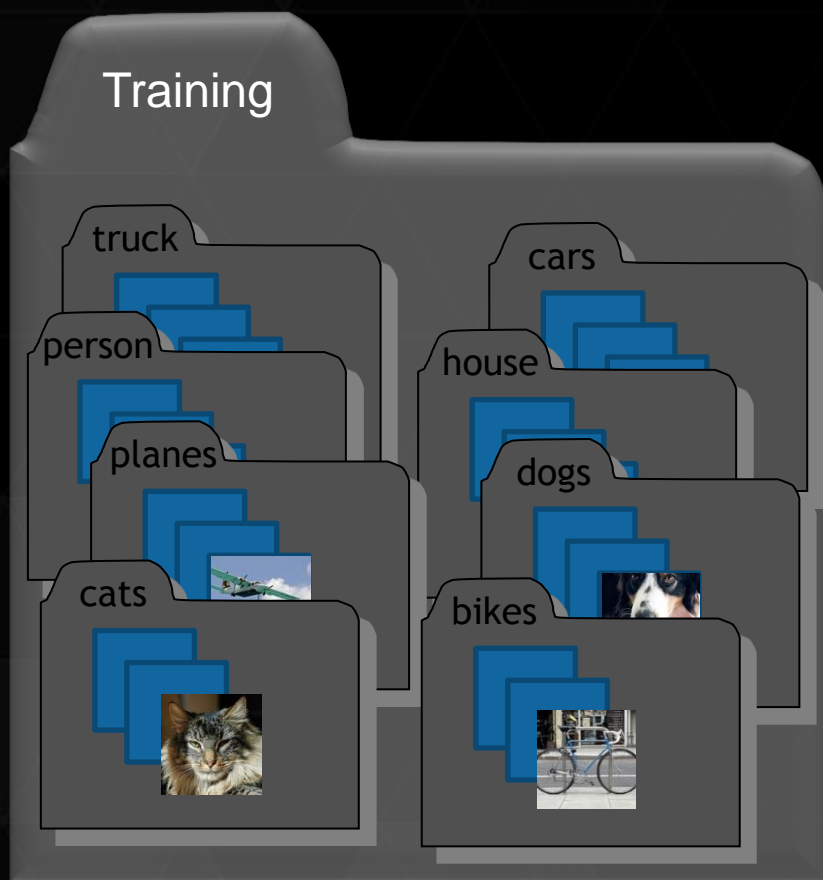
[Create](#)

DIGITS creates your training and validation set for you.



CREATE THE DATABASE

Create Training and Validation Set



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](#)

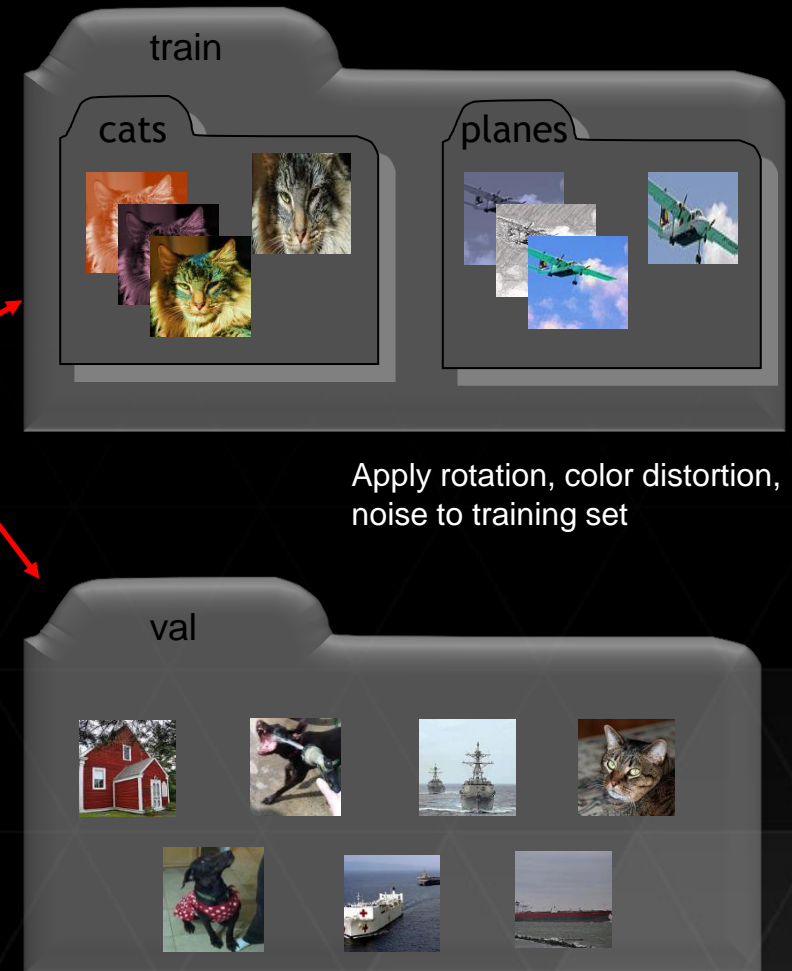
Use Image Folder Upload Text Files

Set	Text file	Image folder (optional)
Training	<input type="text" value="train.txt"/>	<input type="text" value="/path/to/train"/>
<input checked="" type="checkbox"/> Validation	<input type="text" value="val.txt"/>	<input type="text" value="/path/to/val"/>
<input type="checkbox"/> Test	<input type="text" value="No file chosen"/>	

Labels

Dataset Name

[Create](#)



CREATE THE DATABASE

CREATING THE DATABASE

Training and validation
data set information

Category data information is
posted

DIGITS Image Classification Dataset

Job Information

Job Directory
/home/agraj/.digits/jobs/20150408-101548-57c6

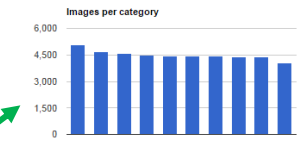
Image Type
Color
Image Dimensions
28x28
Resize Mode
half_crop

Parse Folder (train/val)

Folder
/home/agraj/data/mnist

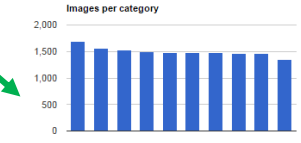
Create DB (train)

Input file
train.txt

Images per category


Create DB (val)

Input file
val.txt

Images per category


Job Status Initialized

- Initialized at 10:15:48 AM

Parse Folder (train/val) Done

Create DB (train) Running

10%

Estimated time remaining: 1 minute, 35 seconds

- Initialized at 10:15:48 AM (2 seconds)
- Running at 10:15:50 AM

Create DB (val) Running

30%

Estimated time remaining: 25 seconds

- Initialized at 10:15:48 AM (2 seconds)
- Running at 10:15:51 AM

Job status
information displays
here – progress,
completion, and
errors

NETWORK CONFIGURATION

DIGITS New Model

New Image Classification Model

Select Dataset

- ship_type
- ship_noship
- mnist60
- mnist_35
- mnist

Data Transformations

Crop Size

none

☒ Subtract Mean File

Select training dataset

Solver Options

Training epochs

30

Snapshot interval (in epochs)

1

Validation interval (in epochs)

1

Random seed

[none]

Batch size

[network defaults]

Solver type

Stochastic gradient descent (SGD)

Base Learning Rate

0.01

☐ Show advanced learning rate options

OR choose a previous configuration

Standard Networks Previous Networks Custom Network

Network	Details	Intended image size
LeNet	Original paper [1998]	28x28 (gray)
AlexNet	Original paper [2012]	256x256
GoogLeNet	Original paper [2014]	256x256

Choose a preconfigured network

Select which GPU you would like to use

- ☒ Next available
- ☐ #0 - GRID K520
- ☐ #1 - GRID K520
- ☐ #2 - GRID K520
- ☐ #3 - GRID K520

Model Name

Create

GPU Resources

Start training

Standard Networks Previous Networks Custom Network

Network	Pretrained Model	
<input type="radio"/> TrainingRun5 View	None	Customize
<input type="radio"/> TrainingRun4 View	None	Customize
<input type="radio"/> TrainingRun3 View	None	Customize

- Load one of your pretrained networks

Standard Networks Previous Networks Custom Network

Custom Network

Visualize

Insert your network here

Pretrained model

- Create a your own network
- Customize a network
- Load a pretrained network and fine tune it

New Image Classification Model

Select Dataset ⓘ

ship_type

ship_noship

mnist50

mnist_35

mnist

mnist

ship_type

Done Mon May 25, 03:23:53 PM

Image Size
256x256

Image Type
COLOR

Create DB (train)
18885 images

Create DB (val)
3334 images

Encoding
png

Data Transformations

Crop Size
none ⓘ

☒ Subtract Mean File ⓘ

Select training dataset

Solver Options

Training epochs
30 ⓘ

Snapshot interval (in epochs)
1 ⓘ

Validation interval (in epochs)
1 ⓘ

Random seed
[none] ⓘ

Batch size
[network defaults] ⓘ

Solver type
Stochastic gradient descent (SGD) ▼ ⓘ

Base Learning Rate
0.01 ⓘ

☐ Show advanced learning rate options

Standard Networks Previous Networks Custom Network

Custom Network ⓘ Visualize

name: "AlexNet"

layer {

name: "data"

type: "Data"

top: "data"

include {

phase: TRAIN

}

transform_param {

Pretrained model
ⓘ

Make network changes here

Select which GPU you would like to use

☒ Next available

☐ #0 - GRID K520

☐ #1 - GRID K520

☐ #2 - GRID K520

☐ #3 - GRID K520

Model Name
 ⓘ

Create

NETWORK CONFIGURATION

Select a standard network and start training

OR

Customize a Standard Network

New Image Class

Select Dataset

- ship_type
- ship_noship
- mnist00
- mnist_35
- mnist
- mnist

Select training datas

Solver Options

Training epochs

30

Snapshot interval (in epochs)

1

Validation interval (in epochs)

1

Random seed

[none]

Batch size

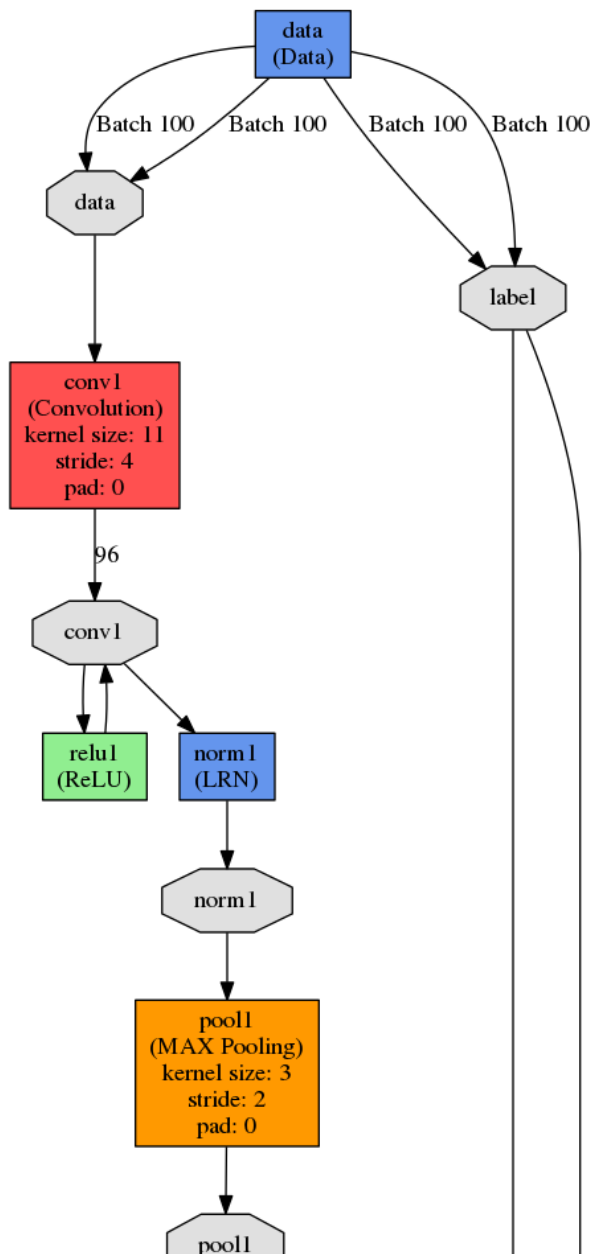
[network defaults]

Solver type

Stochastic gradient descent (SGD)

Base Learning Rate

0.01

☐ Show advanced learning rate options

NETWORK CONFIGURATION

Select a standard network and start training

OR

Customize a Standard Network

Visualize your network

New Image Classification Model

Select Dataset

ship_type

ship_noship

mnist500

mnist_35

mnist

ship_type

Done Mon May 25, 03:23:53 PM

Image Size
256x256

Image Type
COLOR

Create DB (train)
18885 images

Create DB (val)
3334 images

Encoding
png

Data Transformations

Crop Size
none

☒ Subtract Mean File

Select training dataset

Solver Options

Training epochs
30

Snapshot interval (in epochs)
1

Validation interval (in epochs)
1

Random seed
[none]

Batch size
[network defaults]

Solver type
Stochastic gradient descent (SGD)

Base Learning Rate
0.01

☐ Show advanced learning rate options

Standard Networks Previous Networks Custom Network

Custom Network

name: "AlexNet"

layer {

name: "data"

type: "Data"

top: "data"

top: "label"

include {

phase: TRAIN

}

transform_param {

Pretrained model

Make network changes here

Select a standard network and start training

OR

Customize a Standard Network

Visualize your network

Start training

Start training

Select which GPU you would like to use

☒ Next available
☐ #0 - GRID K520
☐ #1 - GRID K520
☐ #2 - GRID K520
☐ #3 - GRID K520

Model Name

Create

ship_type_finetune

Image Classification Model

Abort Job

Delete Job

Solver
solver.prototxt
Network (train/val)
train_val.prototxt
Network (deploy)
deploy.prototxt
Raw caffe output
caffe_output.log
Pretrained Model
home/ubuntu/digits/models/150627-234702-55ac/snapshots_iter_1512_caffemodel

Download network files

Dataset

ship_type

Done Mon May 25, 03:23:03 PM

Image Size

256x256

Image Type

COLOR

Create DB (train)

18885 Images

Create DB (val)

3334 Images

Encoding

png

Job Status (Waiting)

- Initialized at 05:34:16 PM (1 second)
- Running at 05:34:17 PM

Train Caffe Model (Waiting)

20%

Estimated time remaining: 1 hour, 35 minutes

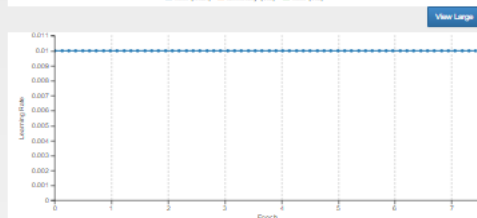
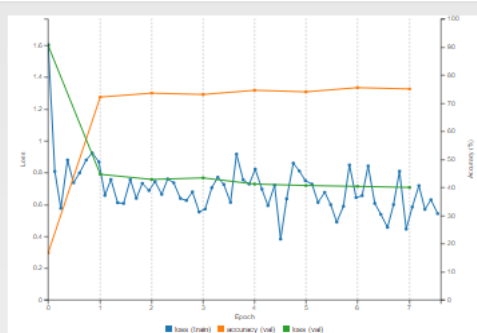
- Initialized at 05:34:16 PM (1 second)
- Running at 05:34:17 PM

Training status

Accuracy and loss values during training

Learning rate

Classification on the with the network snapshots



Select Model

Epoch #7

Download

Image URL

Upload Image

Choose File

No file chosen

Accept a list of filenames or urls (you can use your val.dat file)

Classify Many Images

Number of Images use from the file

100

Leave blank to use all

Number of Images to show per category

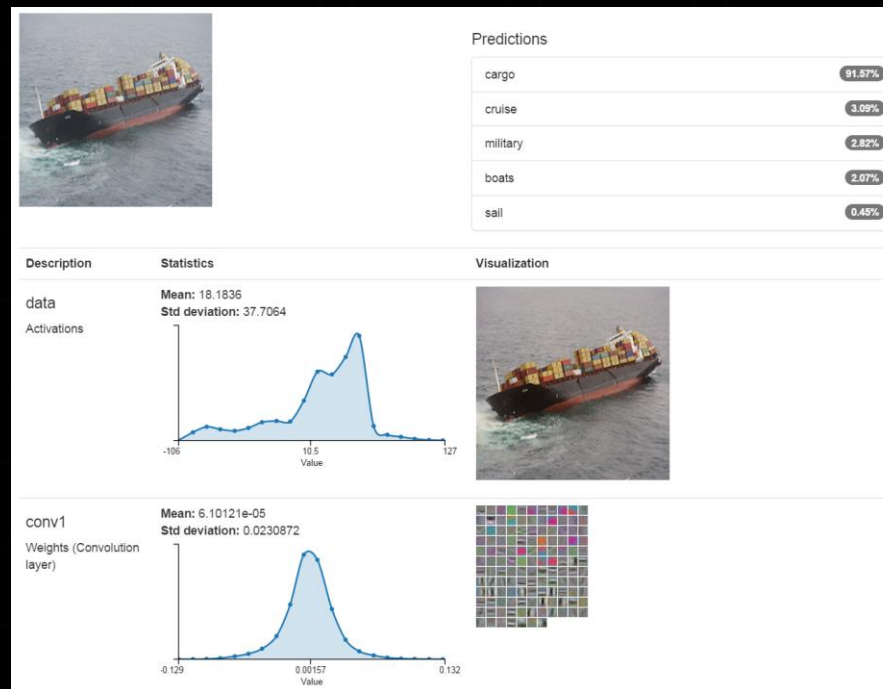
9

Top N Predictions per Category

TRAINING

Visualize DNN performance in real time
Compare networks

Classification



COMPARE RESULTS

DIGITS

Home

Datasets

New Dataset
Images -

In progress
None

Completed

mnist
Submitted: Wed Apr 01, 09:05:45 AM
Status: Done after 3 minutes, 36 seconds

Database3
Submitted: Mar 12, 10:52:38 PM
Status: Aborted after 2 days, 23 hours

Database2
Submitted: Mar 10, 01:42:18 PM
Status: Done after 8 minutes, 17 seconds

Database1
Submitted: Mar 10, 12:46:39 PM
Status: Done after 13 minutes, 50 seconds

Models

New Model
Images -

In progress

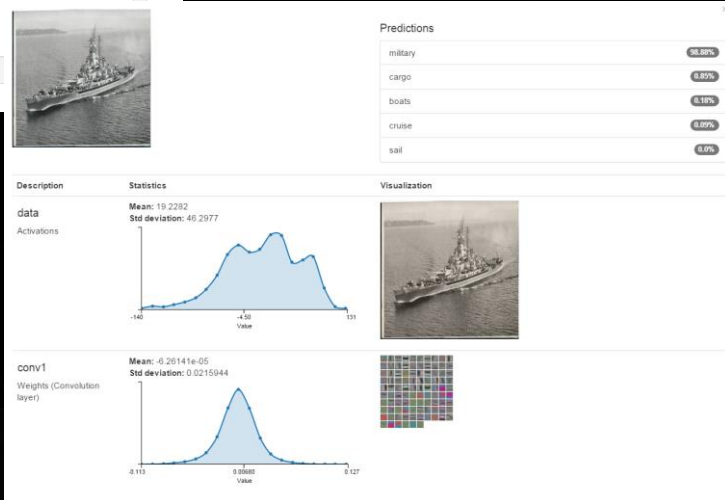
Completed

TrainingRun7
Submitted: 04:44:51 PM (10 minutes, 53 seconds ago)
Status: Running

TrainingRun6
Submitted: Mar 13, 01:26:02 PM
Status: Aborted after 2 days, 8 hours

TrainingRun4
Submitted: Mar 10, 09:30:50 PM
Status: Aborted after 7 hours, 54 minutes

TrainingRun2
Submitted: Mar 10, 05:04:29 PM
Status: Done after 9 hours, 48 minutes



DIGITS

Image Classification Model

ship_type

Image Classification Model

Delete Job

Solver

solver:prototxt
Network (trainval)
train_val.prototxt
Network (deploy)
deploy.prototxt
Raw caffe output
caffe_output.log

Dataset

ship_type

Done Mon May 26, 03:23:50 PM

Image Size
256x256

Image Type
COLOR

Create DB (train)
18886 images

Create DB (val)
3334 images

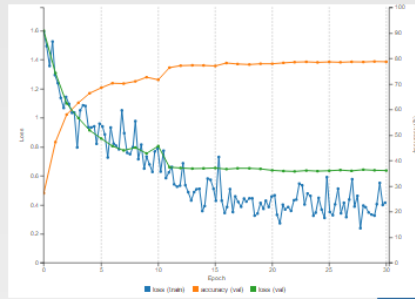
Encoding
png

Job Status

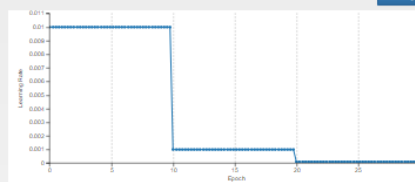
View

- Initialized at Mon May 25, 03:26:48 PM (1 second)
- Running at Mon May 25, 03:26:50 PM (1 hour, 10 minutes)
- Done at Mon May 25, 04:42:50 PM (1 total - 1 hour, 16 minutes)

Train Caffe Model



Loss (train) Accuracy (val) Loss (val)



View Logs

Select Model

Epoch #30

Download

Image URL

Upload Image List

Choose File No file chosen

Accepts a list of filenames or urls (you can use your val list file)

Upload Image

Choose File No file chosen

Classify Many Images

Number of images use from the file

100

Learn labels too (yes/no)

Classify One Image

Number of images to show per category

5

Top N Predictions per Category

PRESENTATION LINEUP

- ▶ **Monday**

- ▶ 4 - Introduction to Graph Analytics

- ▶ **Tuesday**

- ▶ 2 - Performance Testing in Virtual Environments
- ▶ 3 - Leverage GPUs for Image Processing with ENVI

- ▶ **Wednesday**

- ▶ 1 - Accelerating FMV PED Workflows with Real-Time Image Processing
- ▶ 3 - Legion: A CUDA-based Engine for Geospatial Analytics
- ▶ 4 DL Open House

- ▶ **Thursday**

- ▶ 1:30 - Accelerating Visualization and Analytics in Socet GXP with GPUs