Object Detection using NVIDIA DIGITS
Customization and Modification

Deep Learning Institute
NVIDIA Corporation

AGENDA

Introduction to Object Detection

Detection by Combining Deep Learning with Traditional Computer Vision

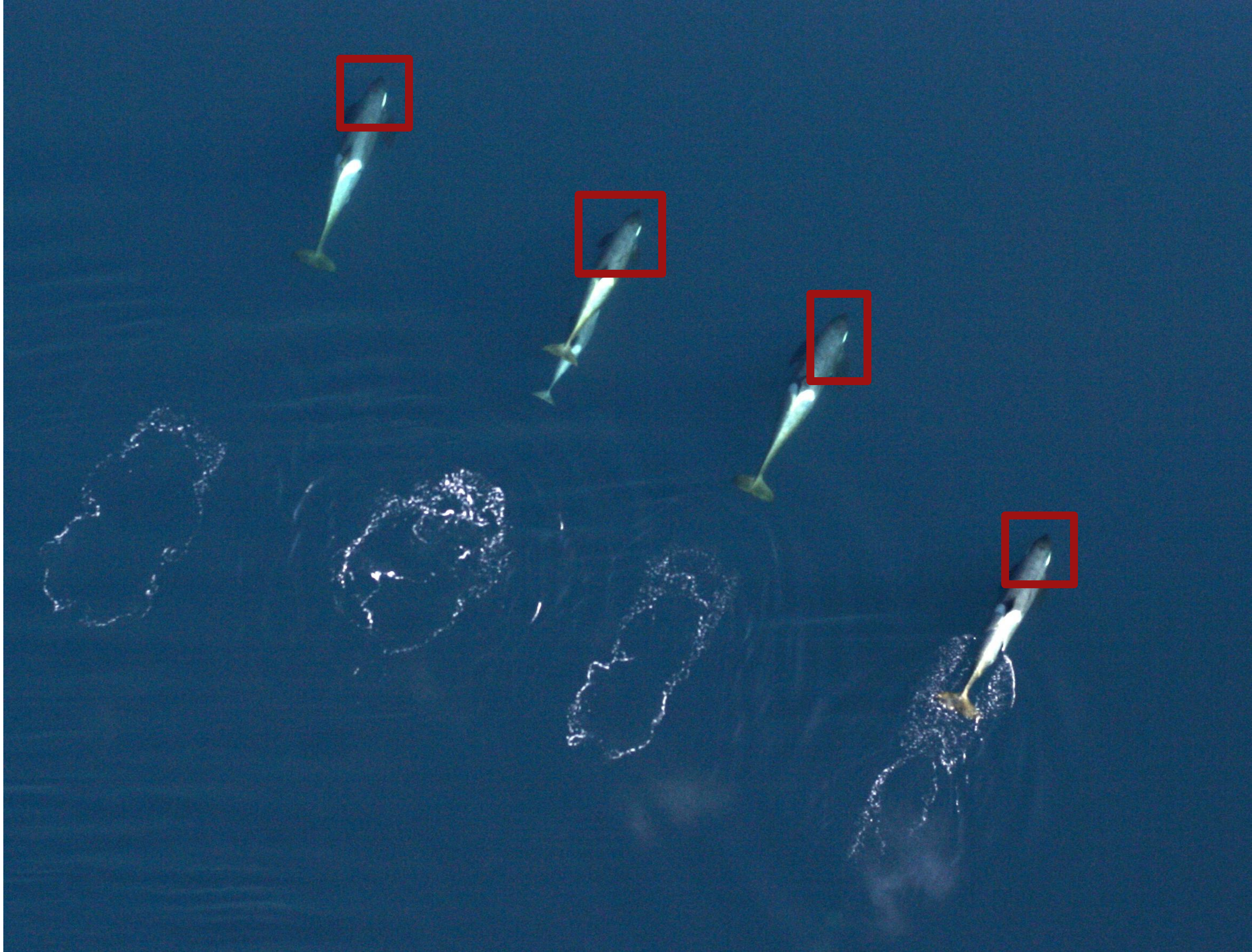
Detection by Modifying Network Architecture

State of the Art Detection

Object Detection

Finding a whale face in the ocean.

We want to know IF there are whale faces in aerial images, and if so, where.



Brainstorm:

How can we use what we know about Image Classification to detect whale faces from aerial images?

Take 2 minutes to think through and write down (paper or computer) ideas.

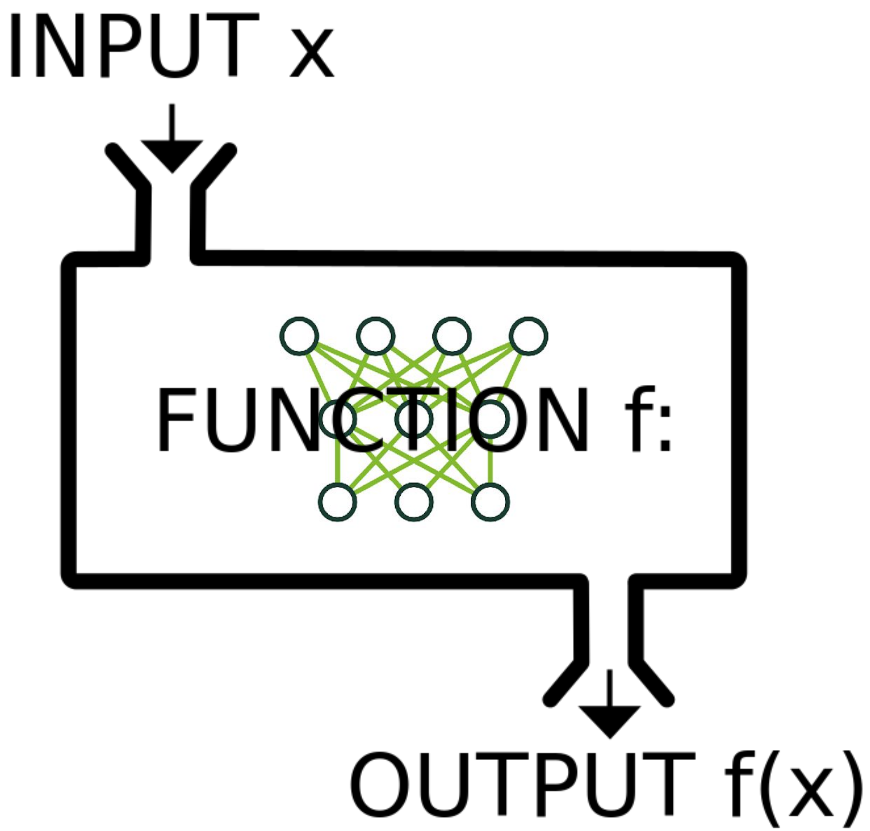
AI at scale

Solving novel problems with code

Applications that combine trained networks with code can create new capabilities

Trained networks play the role of functions

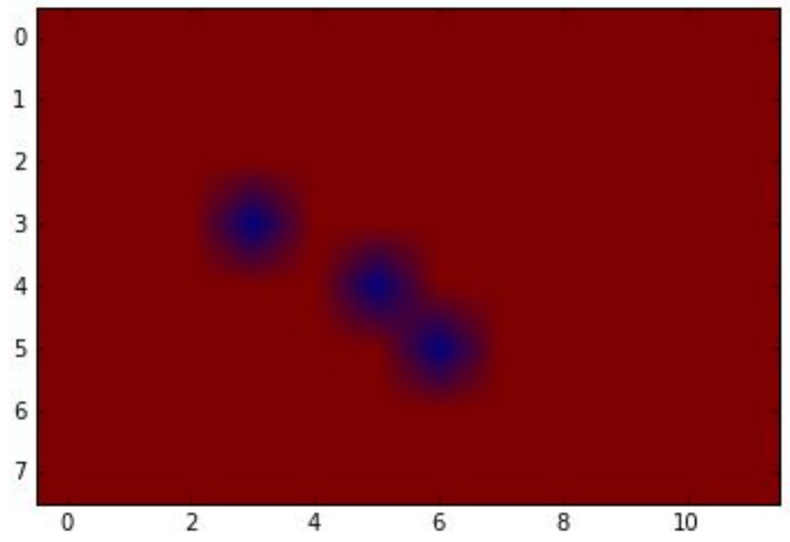
Building applications requires writing code to generate expected inputs and useful outputs



Approach 1: Sliding Window

- Technique:
  - Build a whale face/not whale face classifier
  - Sliding window python application runs classifier on each 256X256 segment
  - Yes = blue, no = red

Total inference time: 10.5373151302 seconds

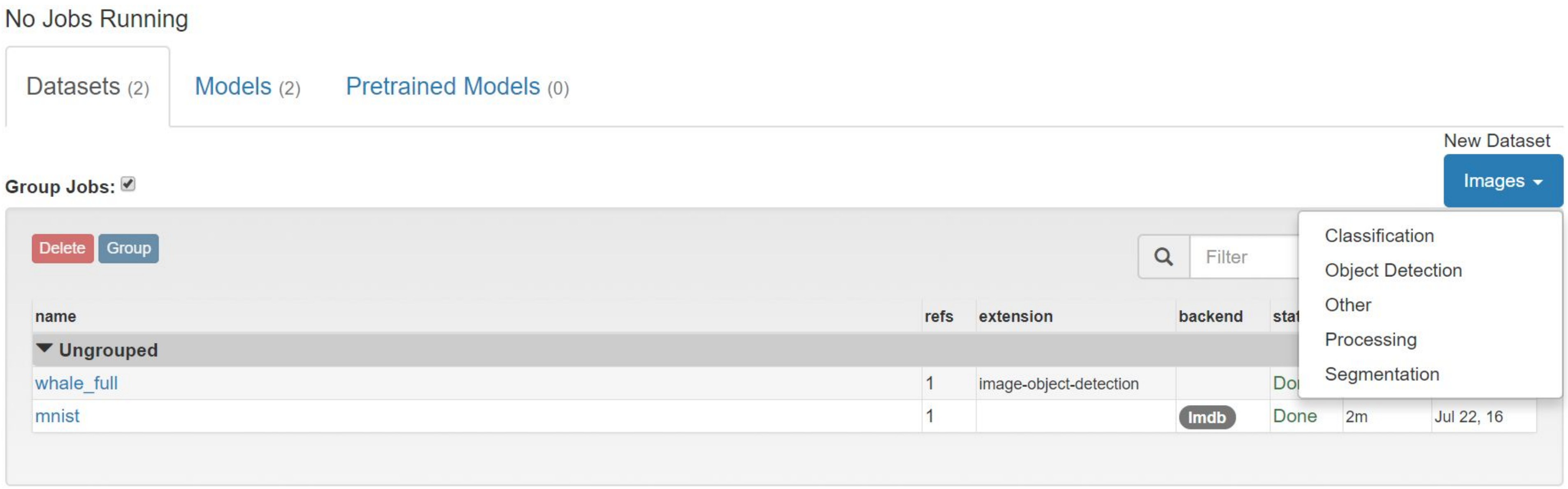


Total inference time: 10.5373151302 seconds

Your turn – Launching lab

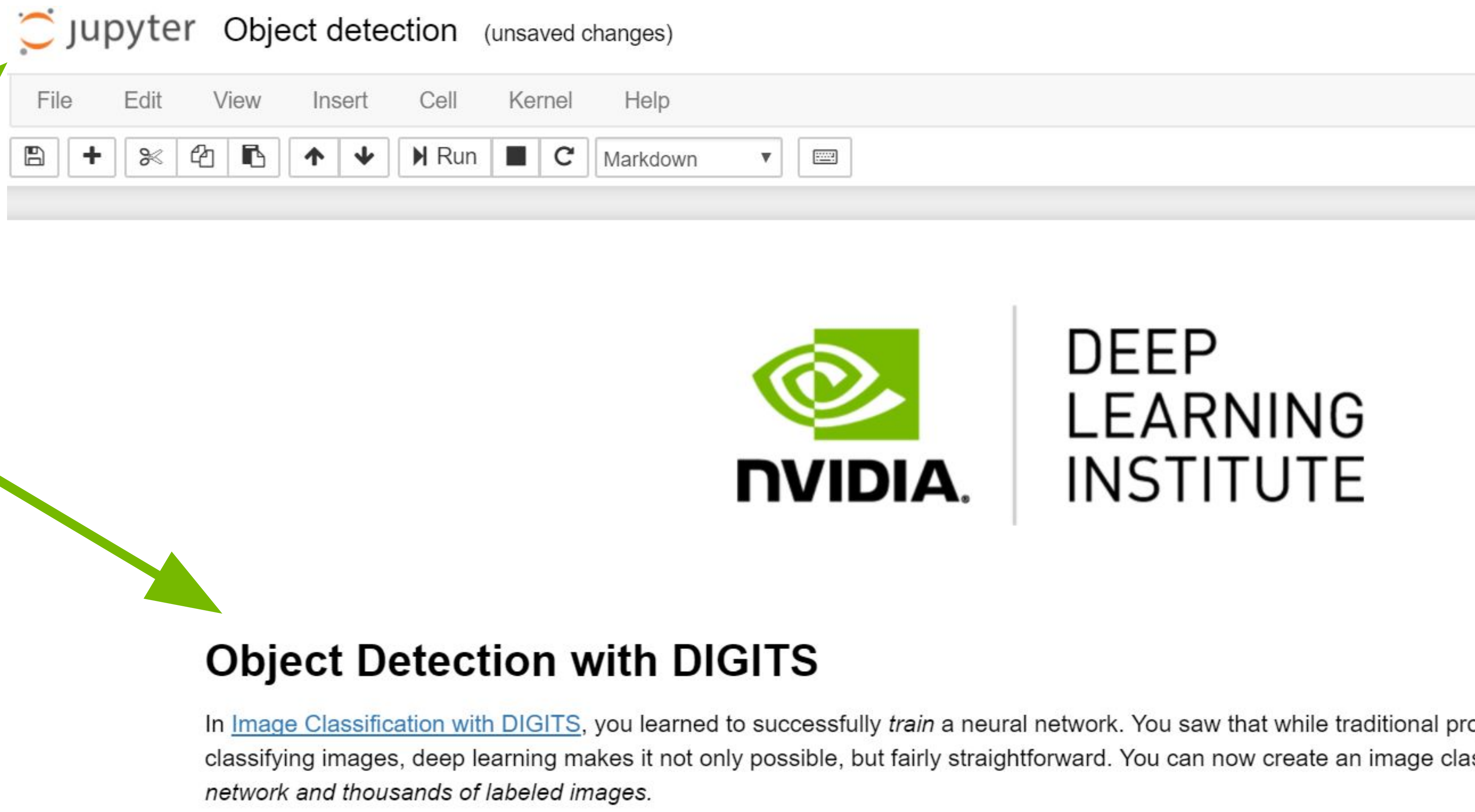
Potential Confusion

Despite existing datasets and models, you will begin the lab by loading a new dataset and training a new *classification* model.



CONNECTING TO THE LAB ENVIRONMENT

Lab will take place in a Jupyter notebook



JUPYTER NOTEBOOK

1. Make changes in code blocks

Copy the job directory (highlighted above) and replace ##FIXME## in the code block below.
Once you've copied the directory, execute the cell (Shift+Enter) to store it to the variable `MODEL_JOB_DIR`

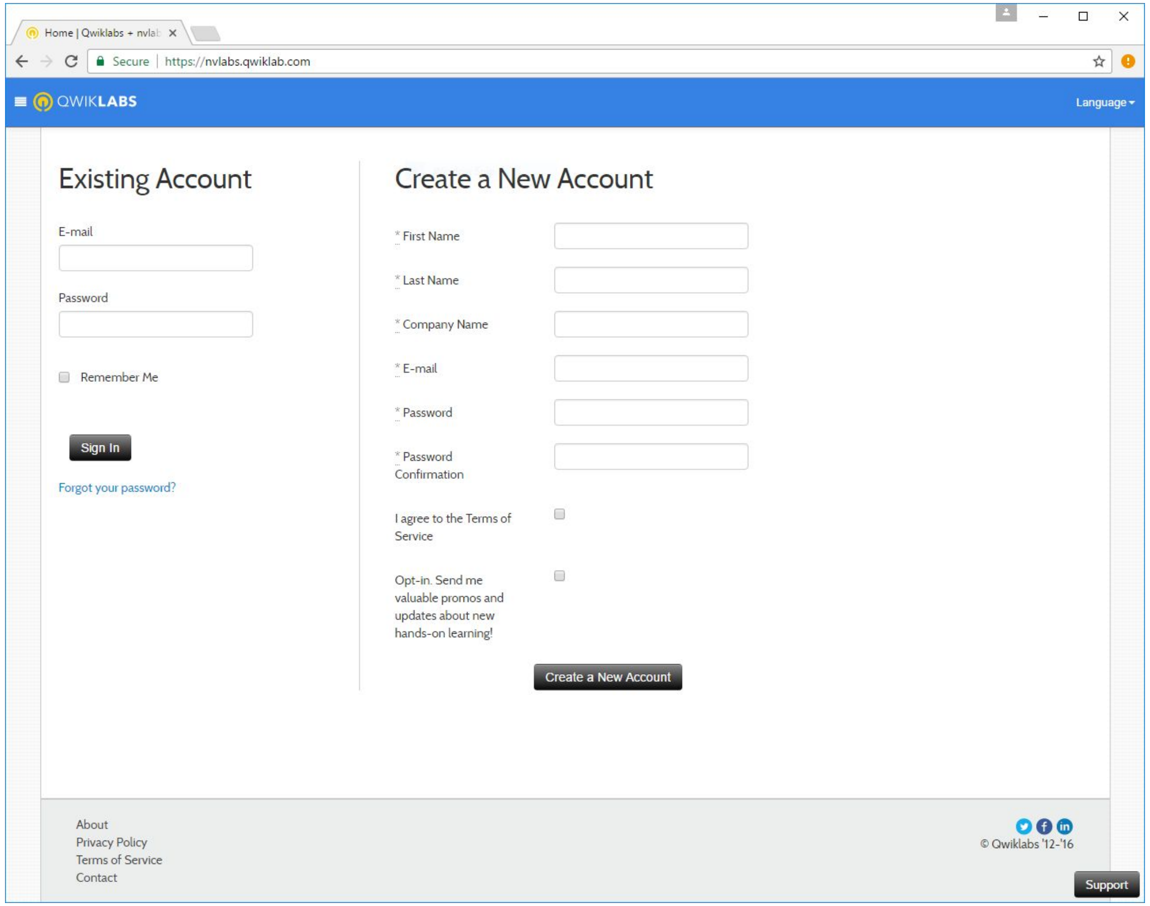
```
In [ ]: MODEL_JOB_DIR = '##FIXME##'  ## Remember to set this to be the job directory for
        print('Got it.')
```

2. Simultanious "Shift" + "Enter" while mouse is in code-block

NAVIGATING TO QWIKLABS

1. Navigate to: https://nvlabs.qwiklab.com

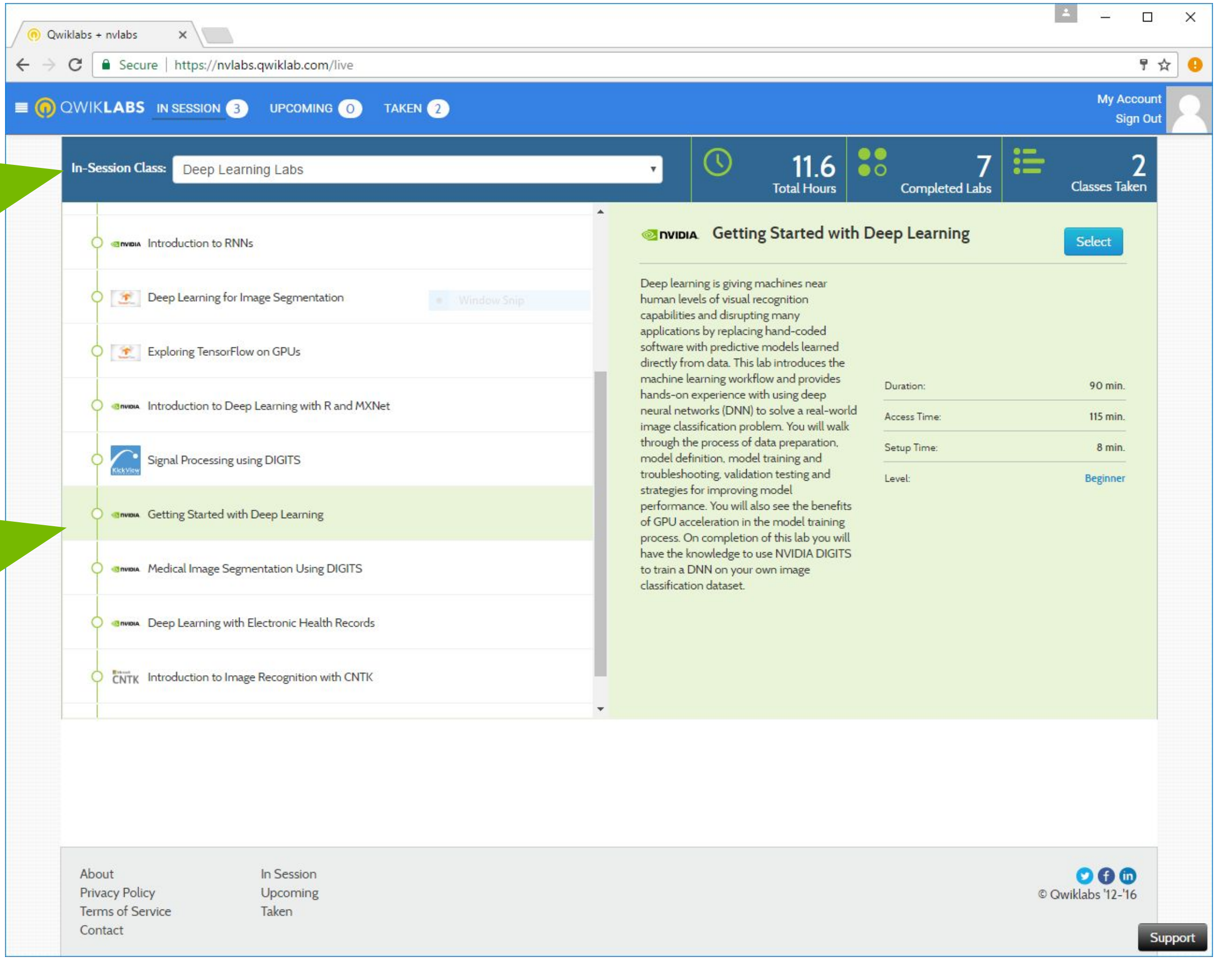
2. Login or create a new account



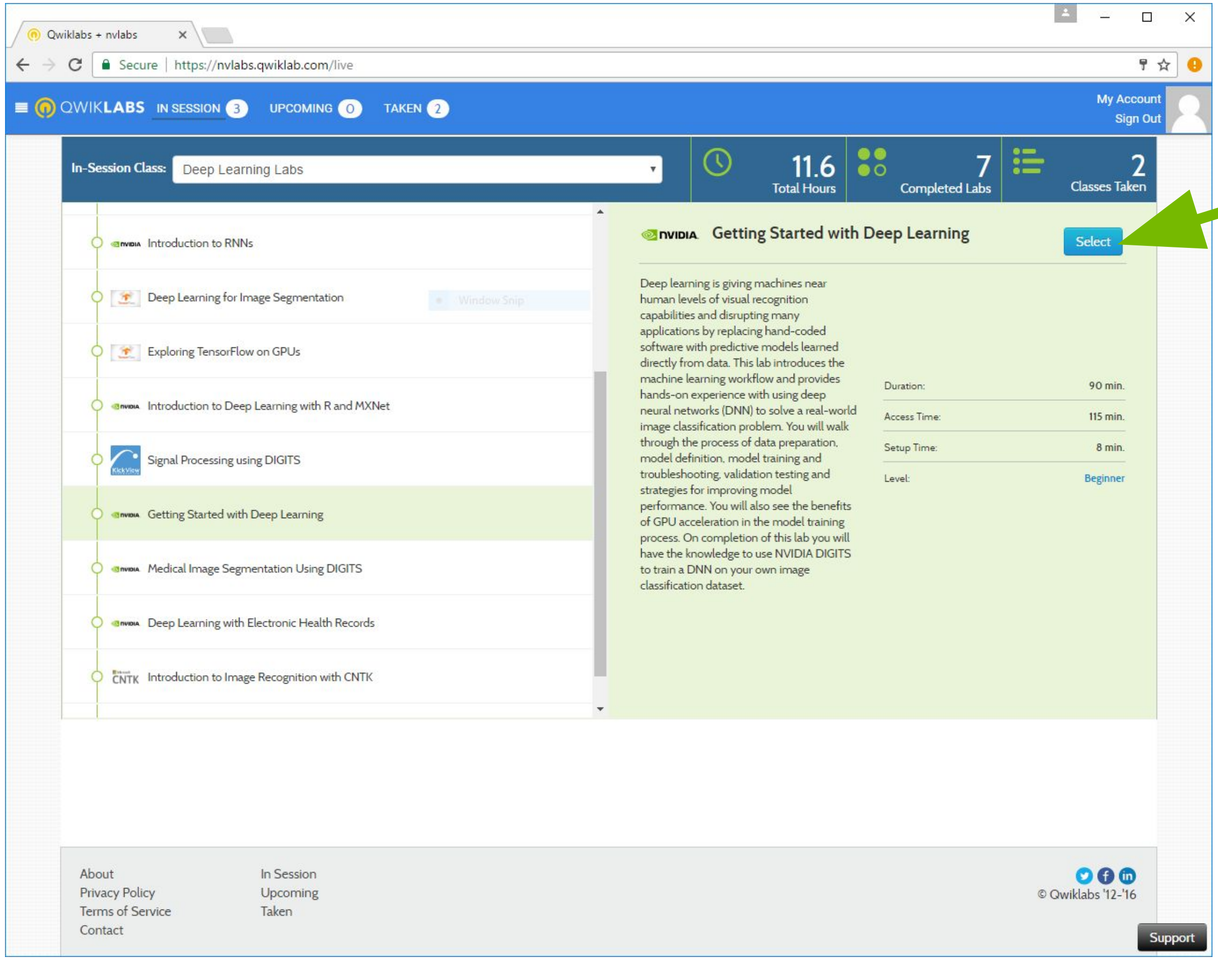
ACCESSING LAB ENVIRONMENT

3. Select the event “Fundamentals of Deep Learning” in the upper left

4. Click the “Object Detection with DIGITS” Class from the list



LAUNCHING THE LAB ENVIRONMENT

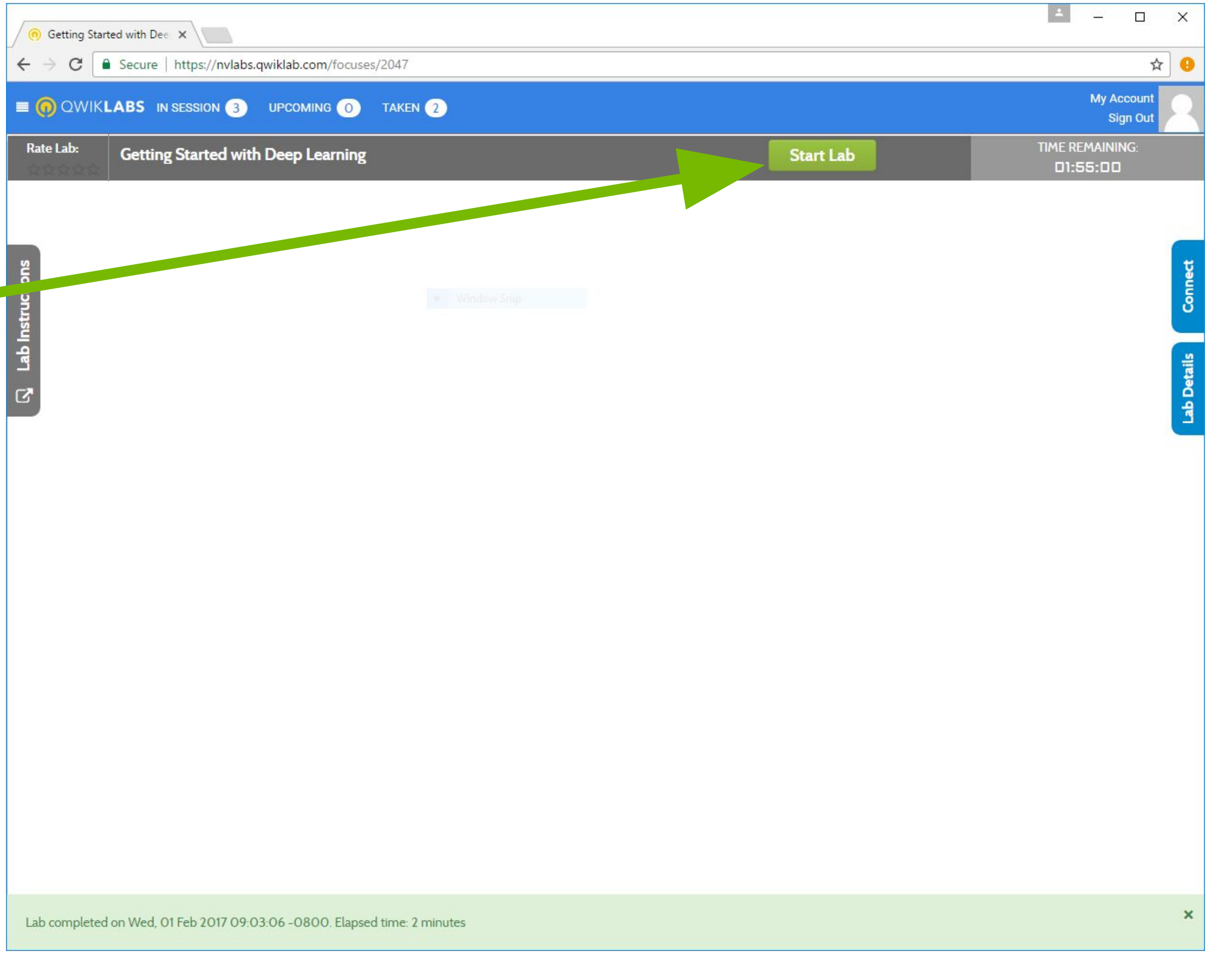


5. Click on the Select button to launch the lab environment

- After a short wait, lab Connection information will be shown
- Please ask Lab Assistants for help!

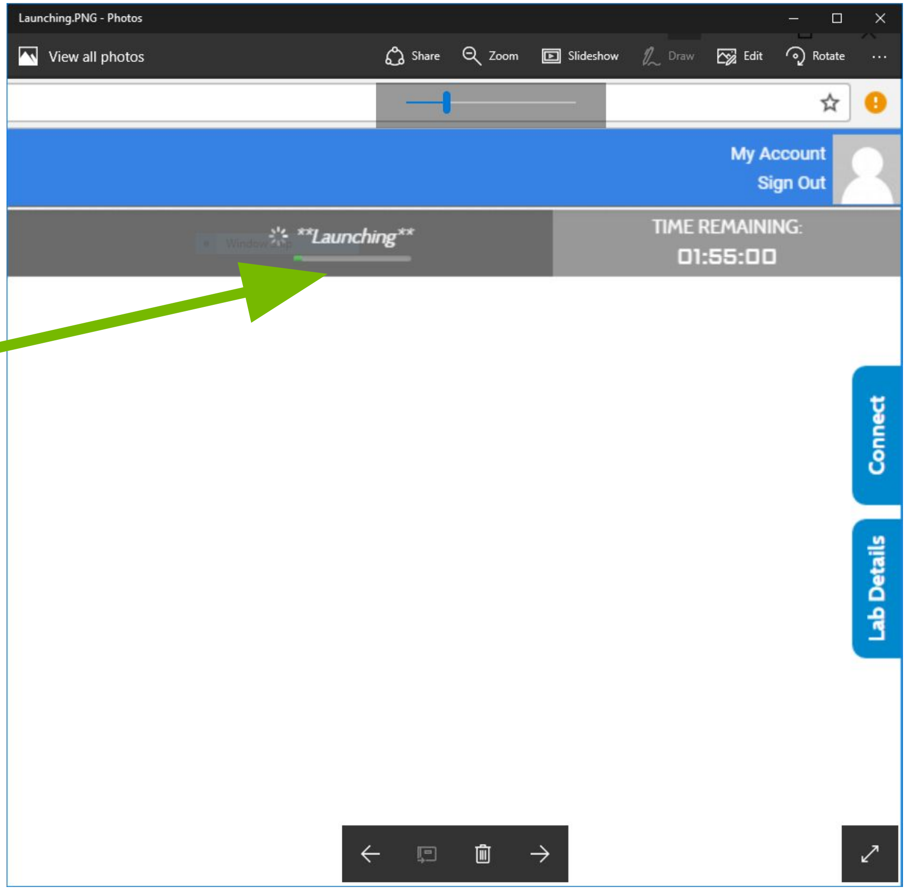
LAUNCHING THE LAB ENVIRONMENT

6. Click on the Start Lab button



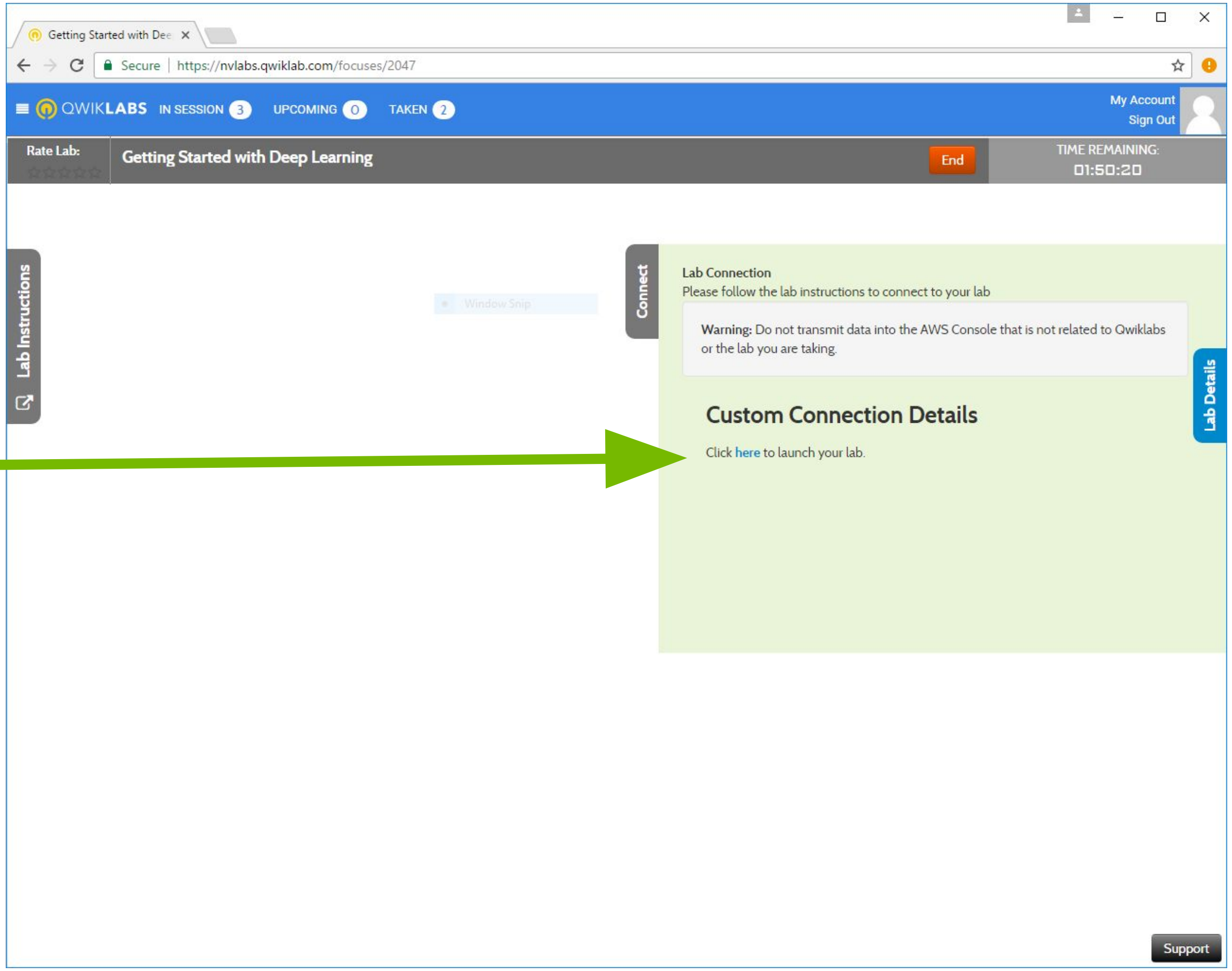
LAUNCHING THE LAB ENVIRONMENT

You should see that the lab environment is "launching" towards the upper-right corner



CONNECTING TO THE LAB ENVIRONMENT

7. Click on “here” to access your lab environment / Jupyter notebook



 NVIDIA DEEP LEARNING INSTITUTE

Follow lab instructions through end of Approach 1

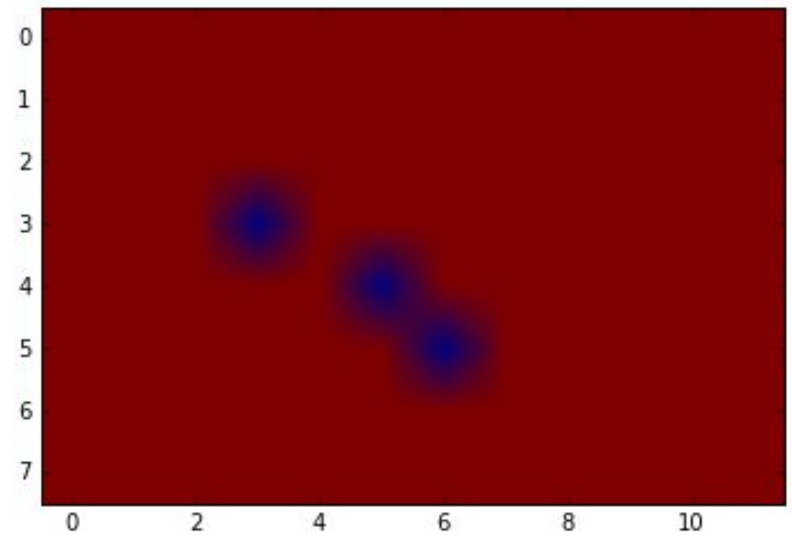
Discuss: Intro to Network Architecture

Approach 1: Sliding Window

- Works but:
    - Needs human supervision
    - Slow – constrained by image size

Total inference time: 10.5373151302 seconds

Total inference time: 10.5373151302 seconds



Approach 2 - Modifying Network Architecture

Layers are mathematical operations on tensors (Matrices, vectors, etc.)

Layers are combined to describe the **architecture** of a neural network

Modifications to network architecture impact **capability** and **performance**

Each **framework** has a different syntax for describing architectures

Regardless of framework: The **output** of each layer *must fit* the **input** of the next layer.

Our current architecture

FRAMEWORK

We've been working in a framework called Caffe.

Each framework requires a different way (syntax) of describing architectures and hyperparameters.

Other frameworks include TensorFlow, MXNet, etc.

NETWORK

We've been working with a network called AlexNet.

Each network can be described and trained using ANY framework.

Different networks learn differently: different training rates, methods, etc. Think different learners.

TOOL - UI

We've been working with a UI called DIGITS

The community works to make model building and deployment easier.

Other tools include Keras, Tensorboard, or APIs with common programming languages.

CAFFE FEATURES

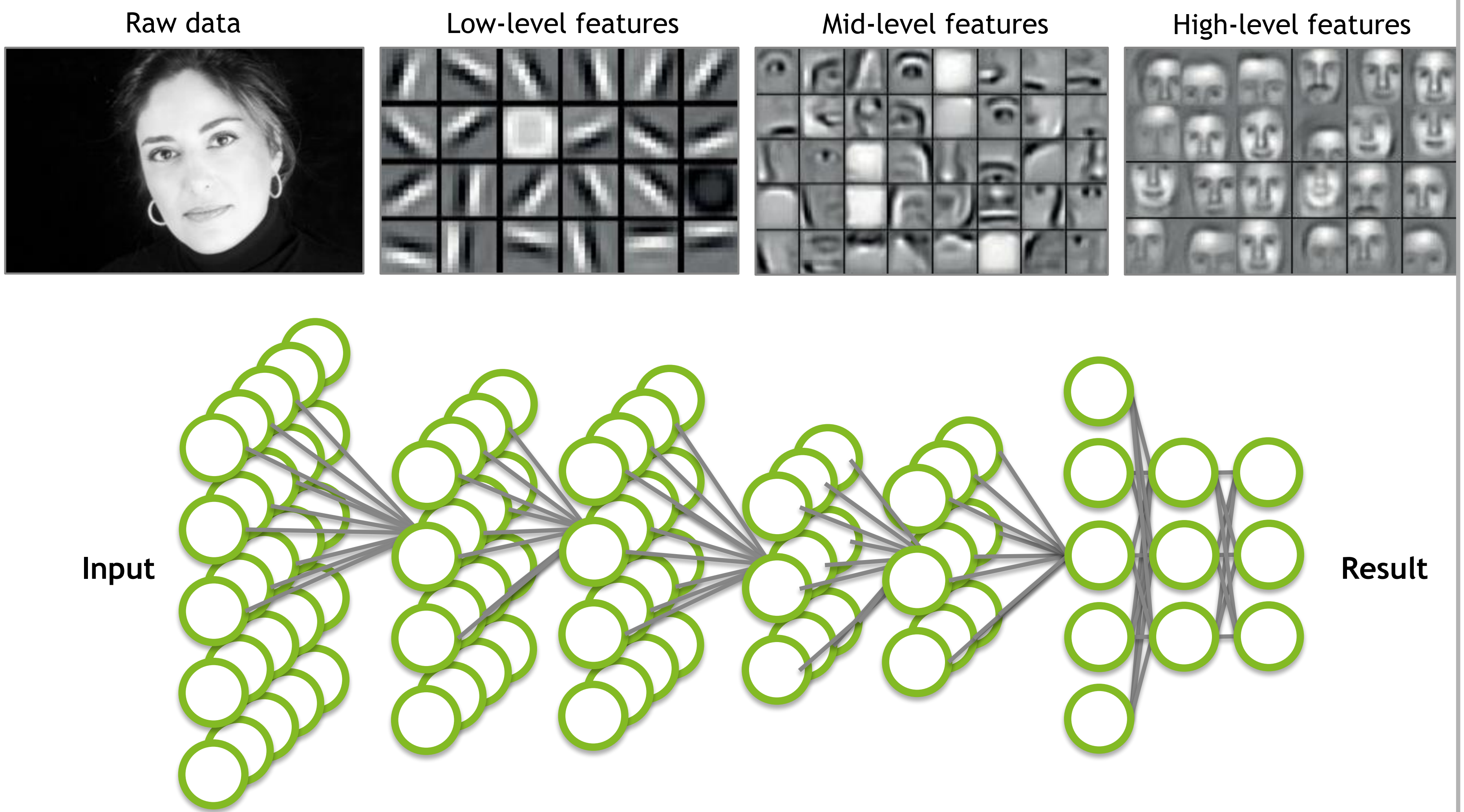
Deep Learning model definition

Protobuf model format

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

```
name: “conv1”
type: “Convolution”
bottom: “data”
top: “conv1”
convolution_param {
    num_output: 16
    kernel_size: 3
    stride: 1
    weight_filler {
        type: “xavier”
    }
}
```

Image Classification Network (CNN)



Application components:

Task objective
e.g. Identify face

Training data
10-100M images

Network architecture
~10s-100s of layers
1B parameters

Learning algorithm
~30 Exaflops
1-30 GPU days

APPROACH 2 - Network Modification

- Modify **AlexNet** by using **Caffe** in **DIGITS**
- Replace **layers** by **reading carefully**

```
}
layer {
  name: "pool5"
  type: "Pooling"
  bottom: "conv5"
  top: "pool5"
  pooling_param {
    pool: MAX
    kernel_size: 3
    stride: 2
  }
}
layer {
  name: "fc6"
  type: "InnerProduct"
  bottom: "pool5"
  top: "fc6"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 2
    decay_mult: 0
  }
  inner_product_param {
    num_output: 4096
    weight_filler {
      type: "gaussian"
      std: 0.005
    }
    bias_filler {
      type: "constant"
      value: 0.1
    }
  }
}
layer {
  name: "relu6"
  type: "ReLU"
  bottom: "fc6"
  top: "fc6"
}
```

```
layer {
  name: "conv6"
  type: "Convolution"
  bottom: "pool5"
  top: "conv6"
  param {
    lr_mult: 1.0
    decay_mult: 1.0
  }
  param {
    lr_mult: 2.0
    decay_mult: 0.0
  }
  convolution_param {
    num_output: 4096
    pad: 0
    kernel_size: 6
    weight_filler {
      type: "gaussian"
      std: 0.01
    }
    bias_filler {
      type: "constant"
      value: 0.1
    }
  }
}
layer {
  name: "relu6"
  type: "ReLU"
  bottom: "conv6"
  top: "conv6"
}
```

RETURN TO THE LAB

Work through the end

We will debrief “Approach 3” post-lab

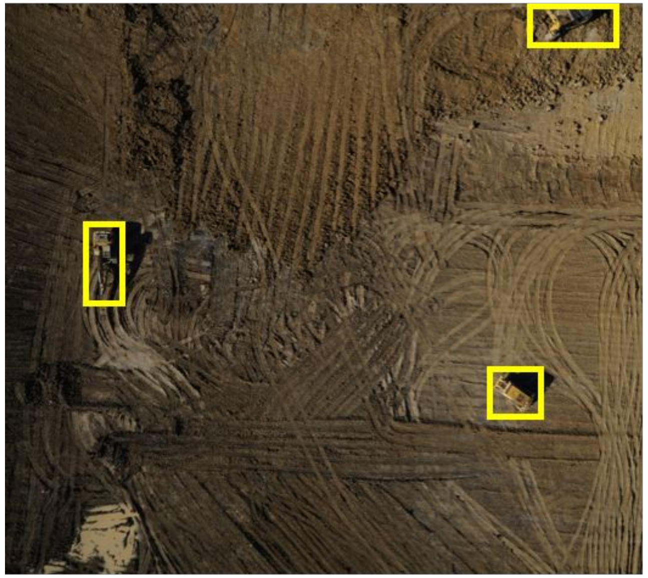
Ask for help if needed

If at any point you get stuck, seek out solutions

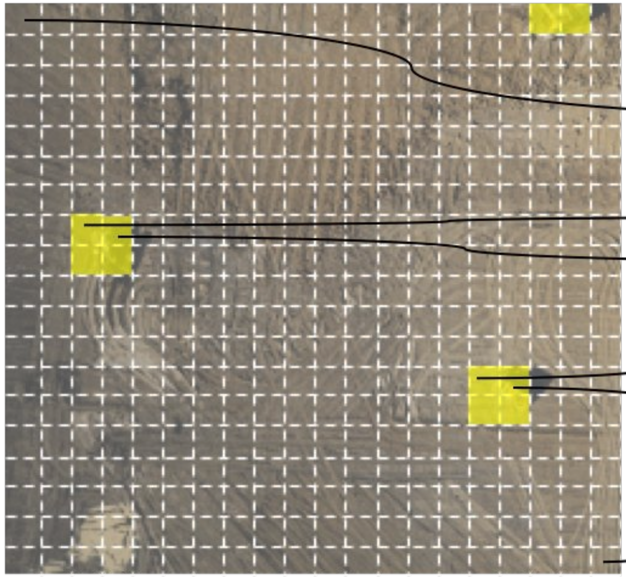
Work through end of lab

Approach 3: End-to-End Solution

Need dataset with inputs and corresponding (often complex) output



Training image with bounding box annotations

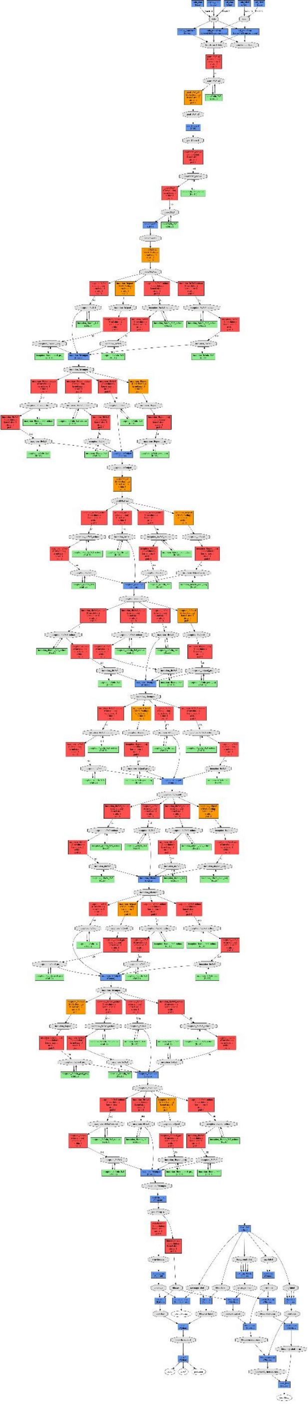


Bounding boxes mapped to grid squares

Bounding box coordinates in pixels relative to center of grid square

class	x_1	y_1	x_2	y_2	coverage
dontcare	0	0	0	0	0
…	…	…	…	…	…
digger	-2	-8	18	24	1
digger	-18	-8	2	24	1
…	…	…	…	…	…
digger	-6	-8	22	24	1
digger	-24	-8	8	24	1
…	…	…	…	…	…
dontcare	0	0	0	0	0

DetectNet input data representation



Approach 3 - End to end solution

High-performing neural network architectures requires **experimentation**

You can benefit from the work of the **community** through the **modelzoo** of each framework

Implementing a new network requires an understanding of data and training **expectations.**

Find projects **similar to your project** as starting points.

Approach 3: End-to-End Solution

- DetectNet:
  - Architecture designed for detecting **anything**
  - Dataset is **whale-face specific**
  - DetectNet is **efficient** and **accurate**

ADDITIONAL APPROACHES TO OBJECT DETECTION ARCHITECTURE

- R-CNN = Region CNN
- Fast R-CNN
- Faster R-CNN Region Proposal Network
- RoI-Pooling = Region of Interest Pooling

Closing thoughts – Creating new functionality

- Approach 1: Combining DL with programming
  - Scaling models programmatically to create new functionality
- Approach 2: Experiment with network architecture
  - Study the math of neural networks to create new functionality
- Approach 3: Identify similar solutions
  - Study existing solutions to implement new functionality

GPU TECHNOLOGY CONFERENCE

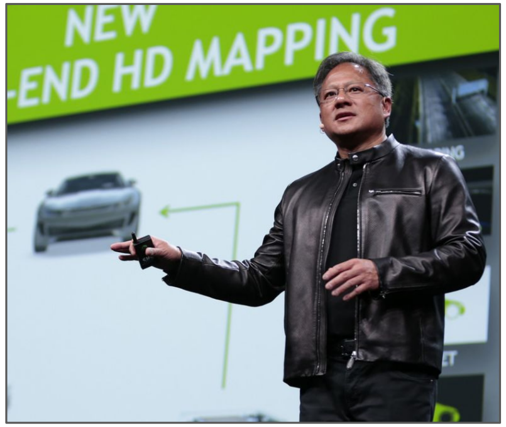
March 26-29, 2018| Silicon Valley | #GTC18
www.gputechconf.com

CONNECT

Connect with technology experts from NVIDIA and other leading organizations

LEARN

Gain insight and valuable hands-on training through hundreds of sessions and research posters



DISCOVER

See how GPUs are creating amazing breakthroughs in important fields such as deep learning and AI

INNOVATE

Hear about disruptive innovations from startups

Enjoy the world’s most important event for GPU developers
March 26-29, 2018 in Silicon Valley

NVIDIA

DEEP
LEARNING
INSTITUTE

www.nvidia.com/dli