

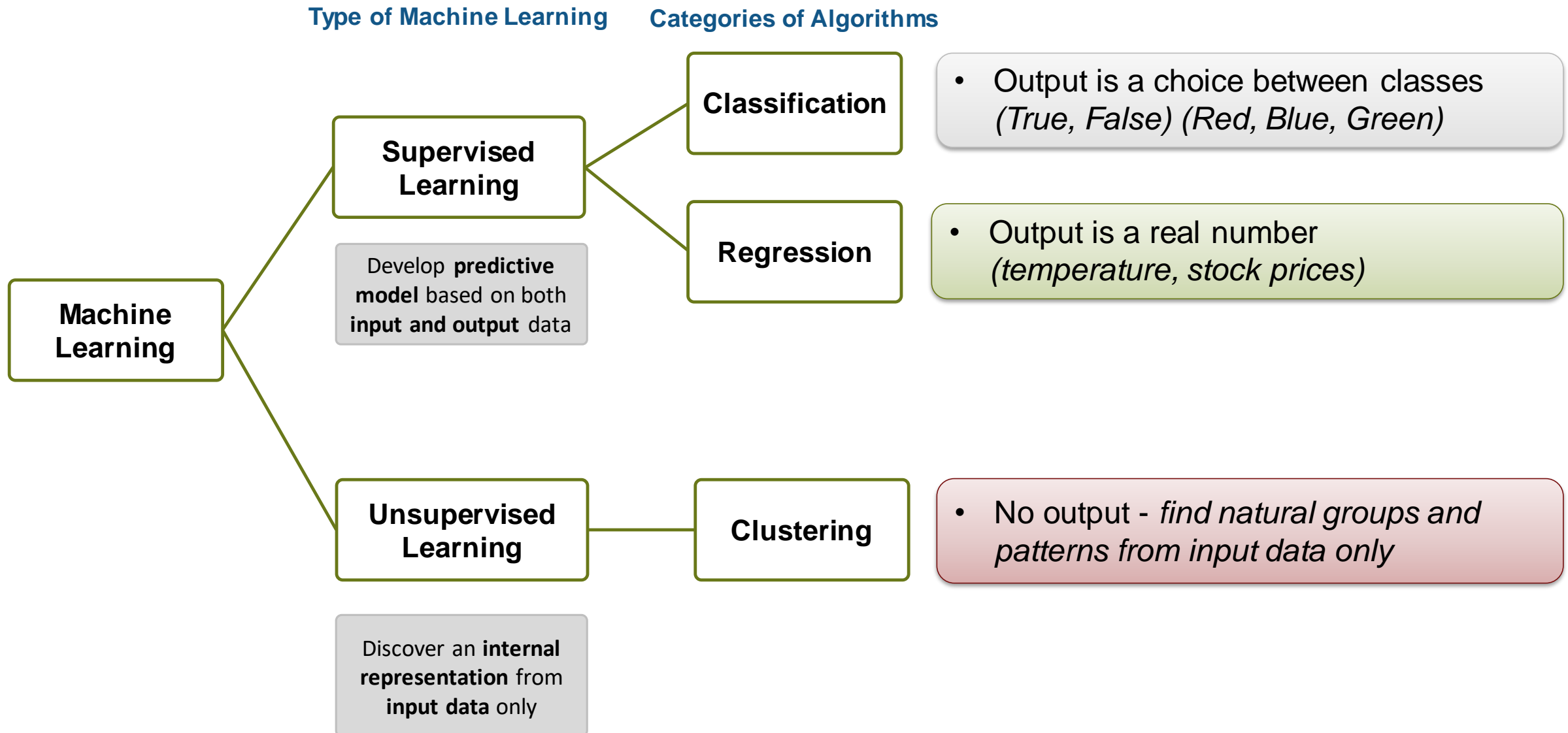
# Introduction to Deep Learning in Signal Processing & Communications with MATLAB

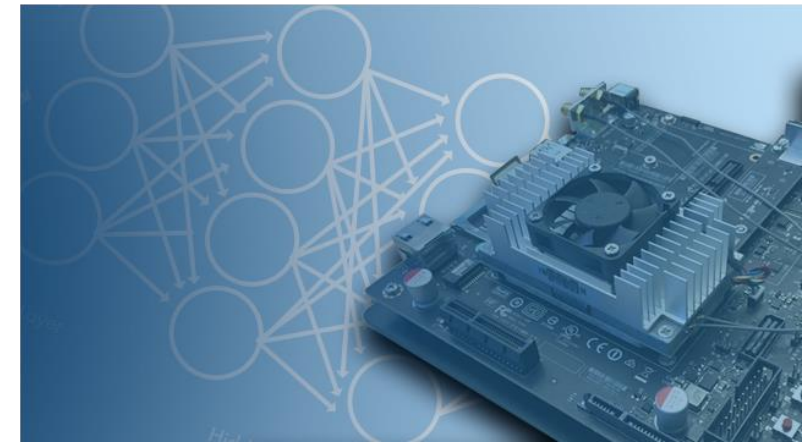
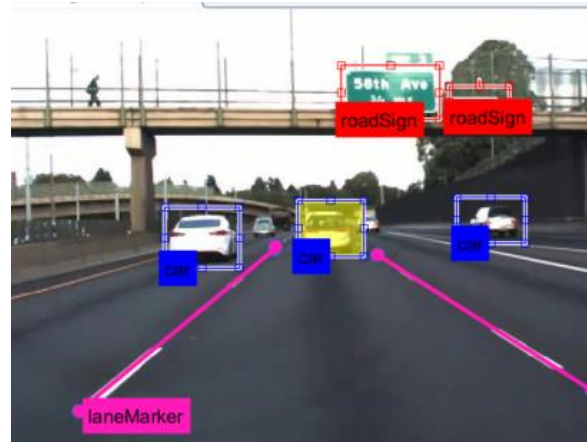
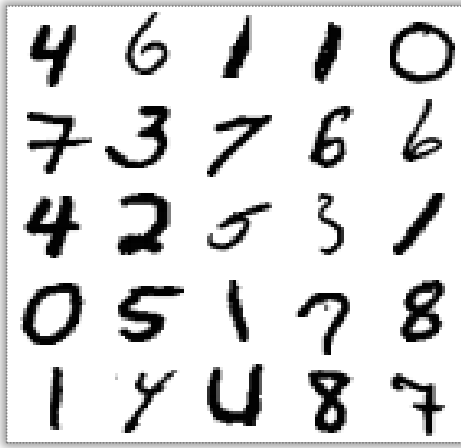
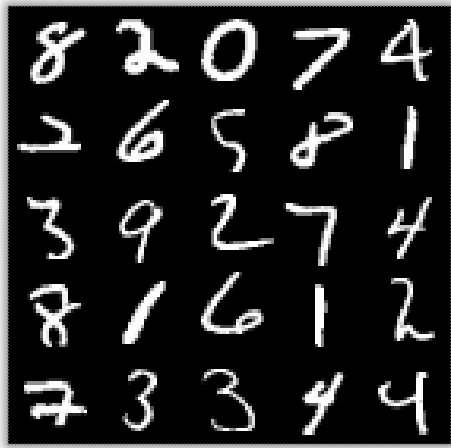
Dr. Amod Anandkumar

Pallavi Kar

*Application Engineering Group, Mathworks India*

# Different Types of Machine Learning





# What is Deep Learning?



12 40.0%	0 0.0%	100% 0.0%
0 0.0%	18 60.0%	100% 0.0%
100% 0.0%	100% 0.0%	100% 0.0%

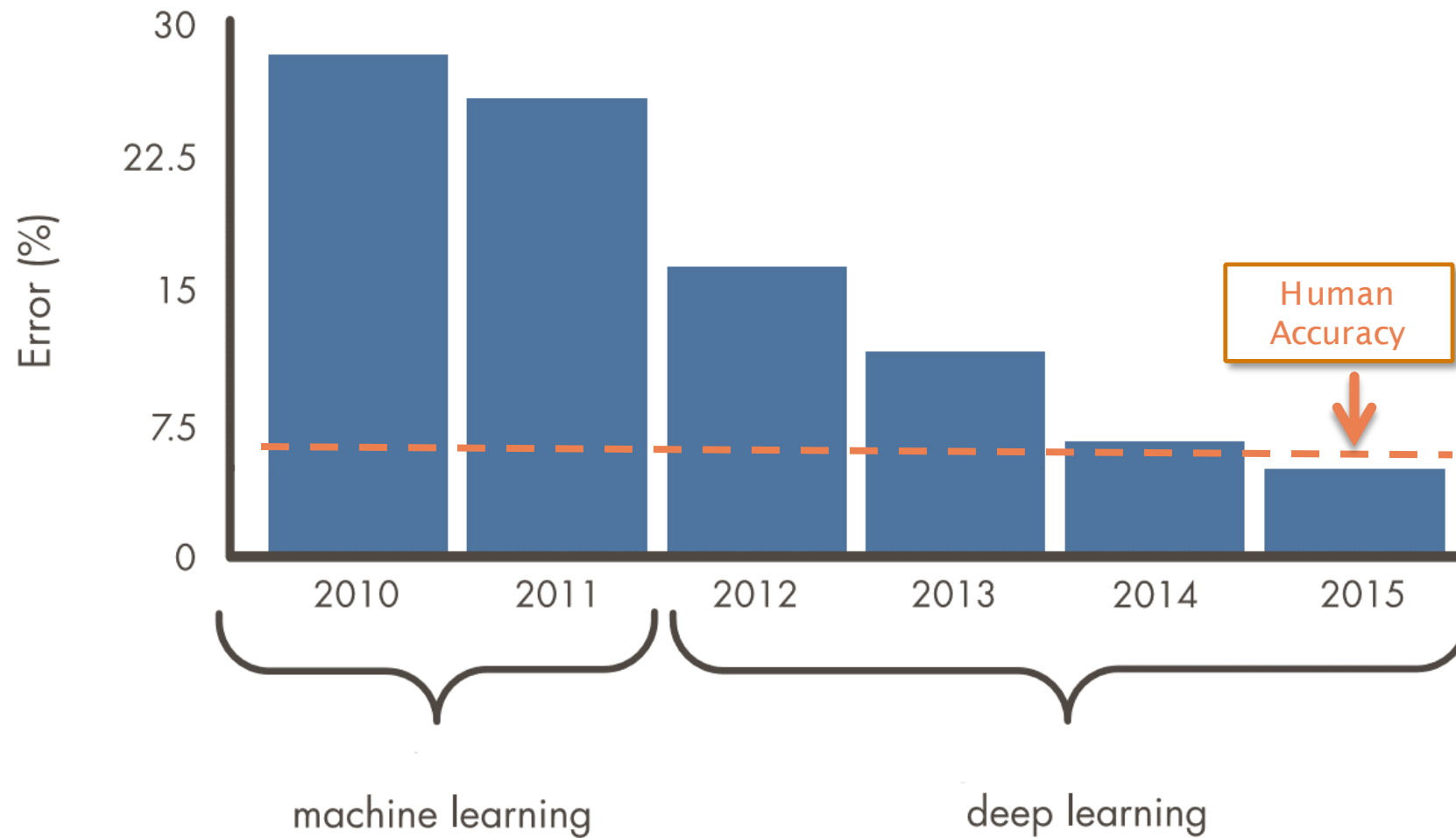
**Deep learning** is a type of supervised machine learning in which a model learns to perform classification tasks directly from images, text, or sound.

Deep learning is usually implemented using a **neural network**.

The term “deep” refers to the **number of layers** in the network—the more layers, the deeper the network.

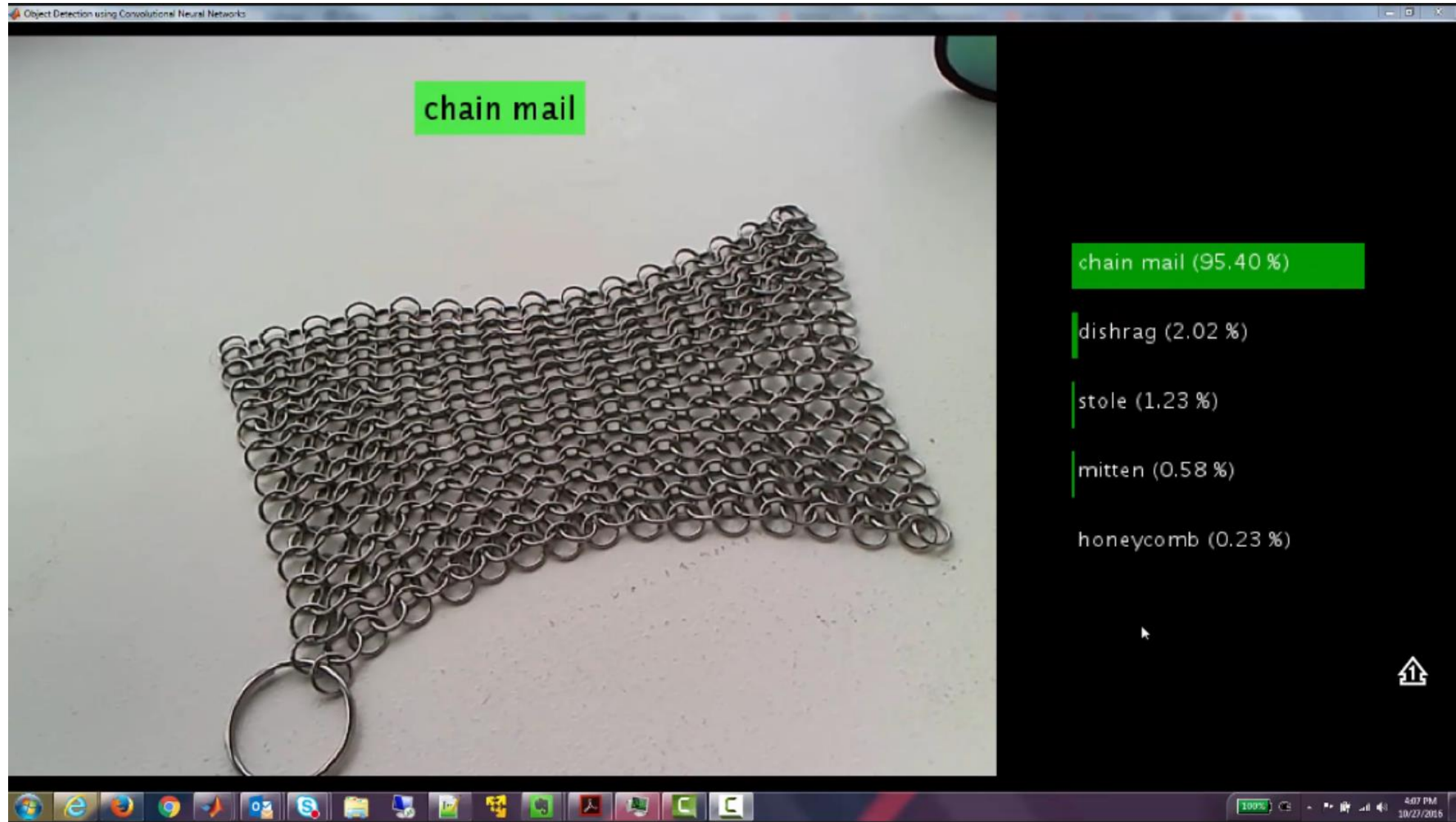


# Why is Deep Learning So Popular Now?



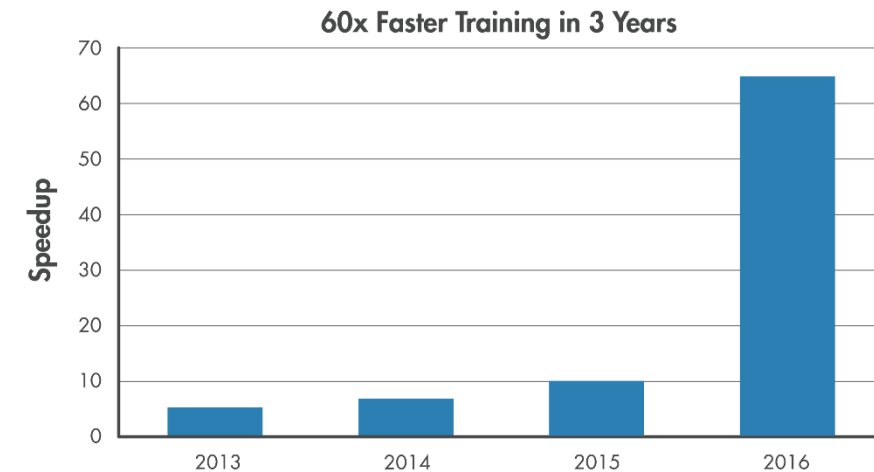
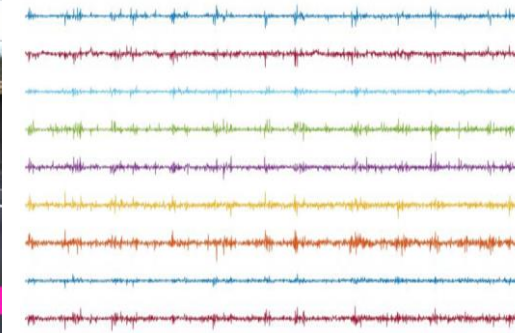
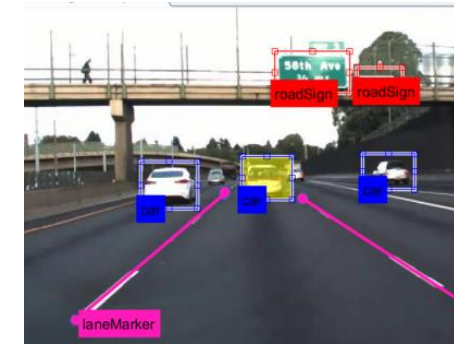
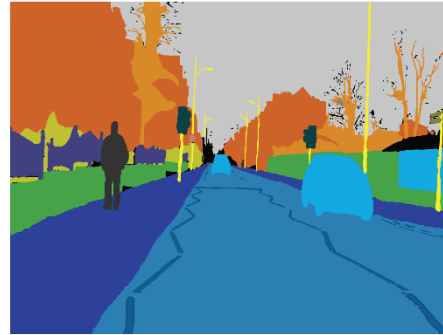
Source: ILSVRC Top-5 Error on ImageNet

Vision applications have been driving the progress in deep learning producing surprisingly accurate systems



# Deep Learning success enabled by:

- Labeled public datasets
- Progress in GPU for acceleration
- World-class models and connected community



**AlexNet**  
PRETRAINED  
MODEL

**VGG-16**  
PRETRAINED  
MODEL

**ResNet-50**  
PRETRAINED MODEL

**ONNX Converter**  
MODEL CONVERTER

**Caffe**  
IMPORTER

**GoogLeNet**  
PRETRAINED  
MODEL

**TensorFlow-  
Keras**  
IMPORTER

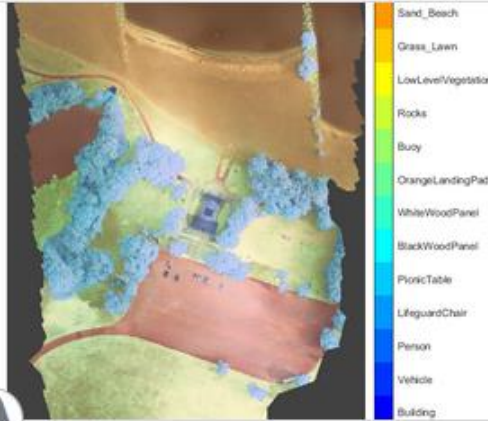
**Inception-v3**  
MODELS

# Deep Learning is Versatile

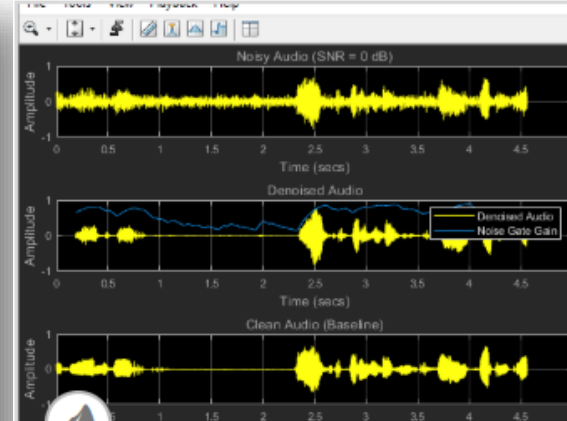
[MATLAB Examples Available Here](#)



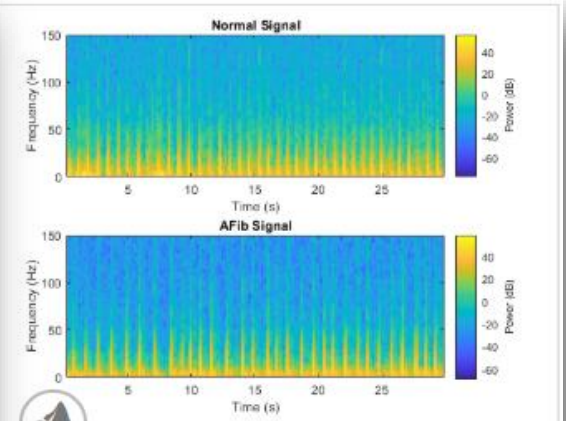
**Object Detection Using  
Faster R-CNN Deep  
Learning**



**Semantic Segmentation of  
Multispectral Images Using  
Deep Learning**



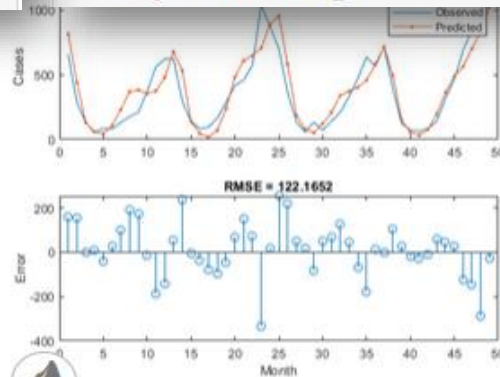
**Denoise Speech Using  
Deep Learning Networks**



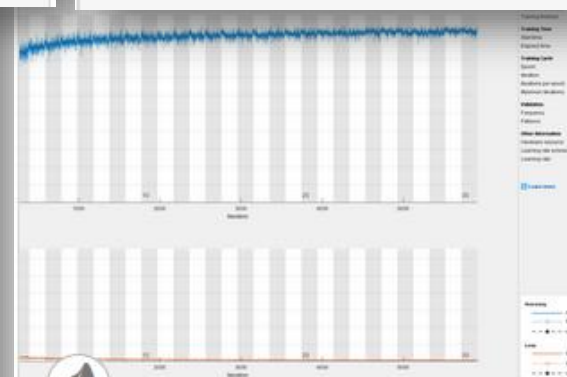
**Classify ECG Signals  
Using Long Short-Term  
Memory Networks**



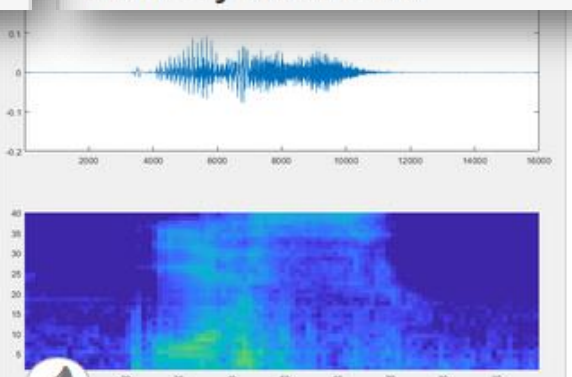
**Classify Image Using  
GoogLeNet**



**Time Series Forecasting  
Using Deep Learning**



**Classify Text Data Using  
Deep Learning**

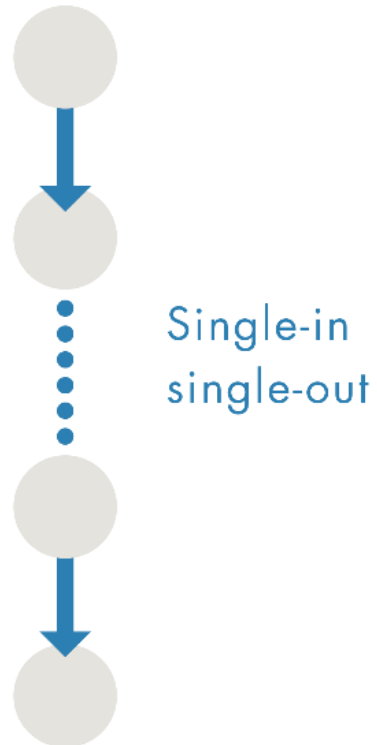


**Deep Learning Speech  
Recognition**



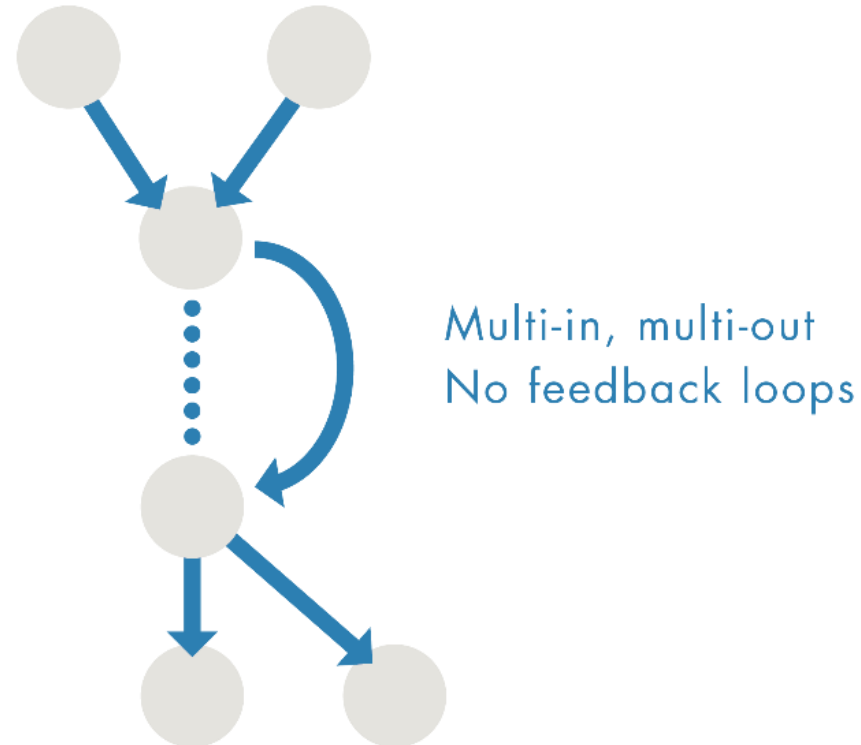
# Many Network Architectures for Deep Learning

Series Network



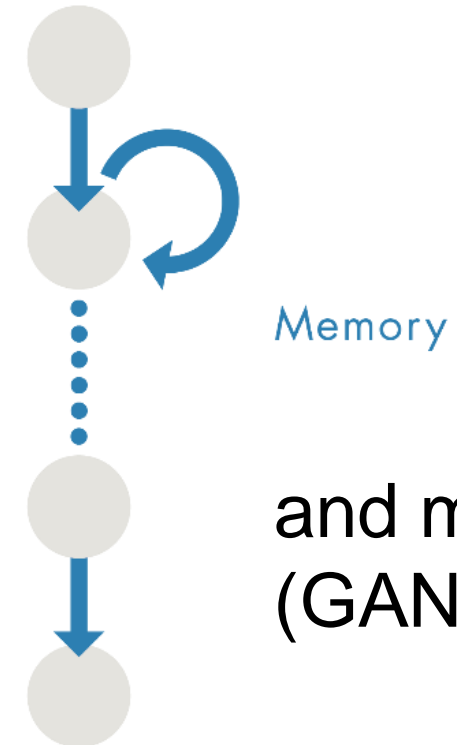
AlexNet  
YOLO

Directed Acyclic  
Graph Network



ResNet  
R-CNN

Recurrent Network

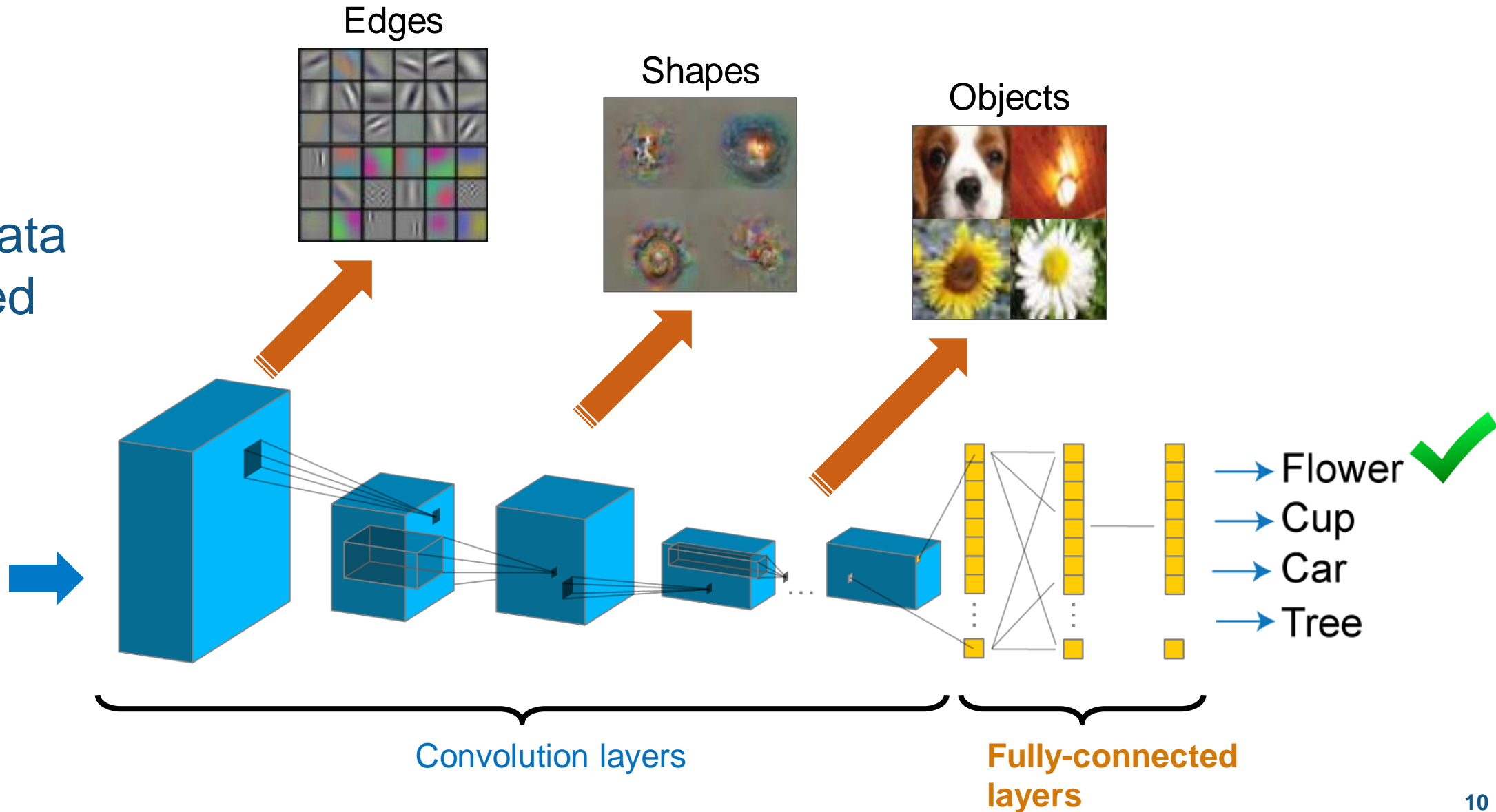


LSTM

and more  
(GAN, DQN,...)

# Convolutional Neural Networks

A lot of data  
is required



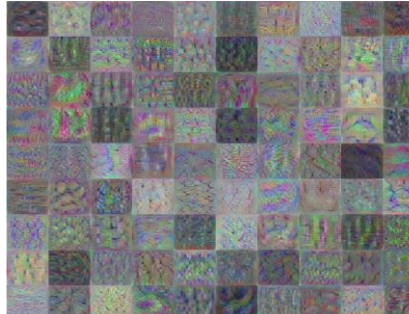
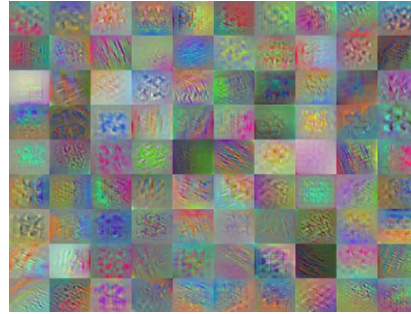
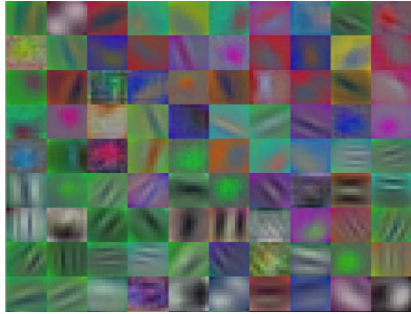
# Deep Learning Inference in 4 Lines of Code

```
>> net = alexnet;  
>> I = imread('peacock.jpg')  
>> I1 = imresize(I,[227 227]);  
>> classify(net,I1)  
  
ans =  
  
    categorical  
  
    peacock
```



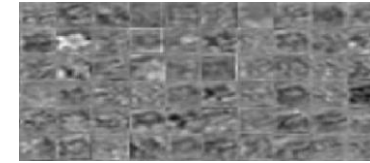
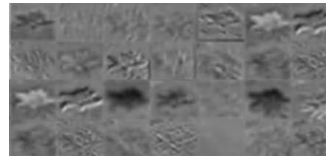
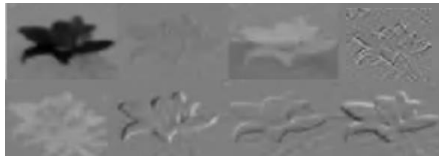
# Understanding network behavior using visualizations

Filters



Deep Dream

Activations



- Custom visualizations
  - Example: Class Activation Maps  
(See [blog post](#))



# Visualization Technique – Deep Dream

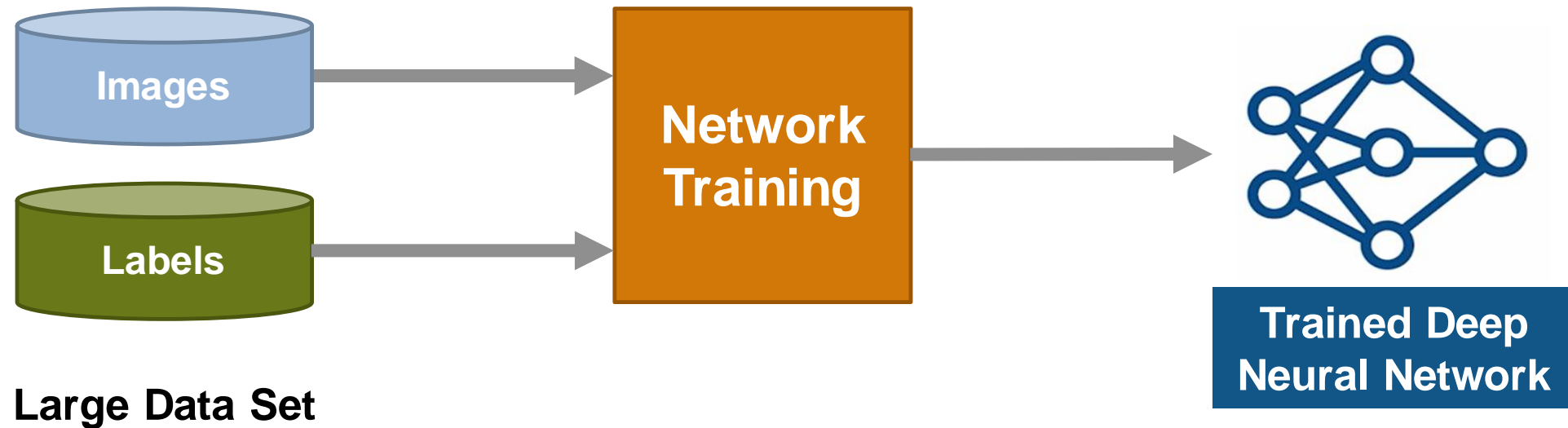
```
deepDreamImage(...  
    net, 'fc8', channel,  
    'NumIterations', 50, ...  
    'PyramidLevels', 4, ...  
    'PyramidScale', 1.25);
```

Synthesizes images that strongly activate a channel in a particular layer



[Example Available Here](#)

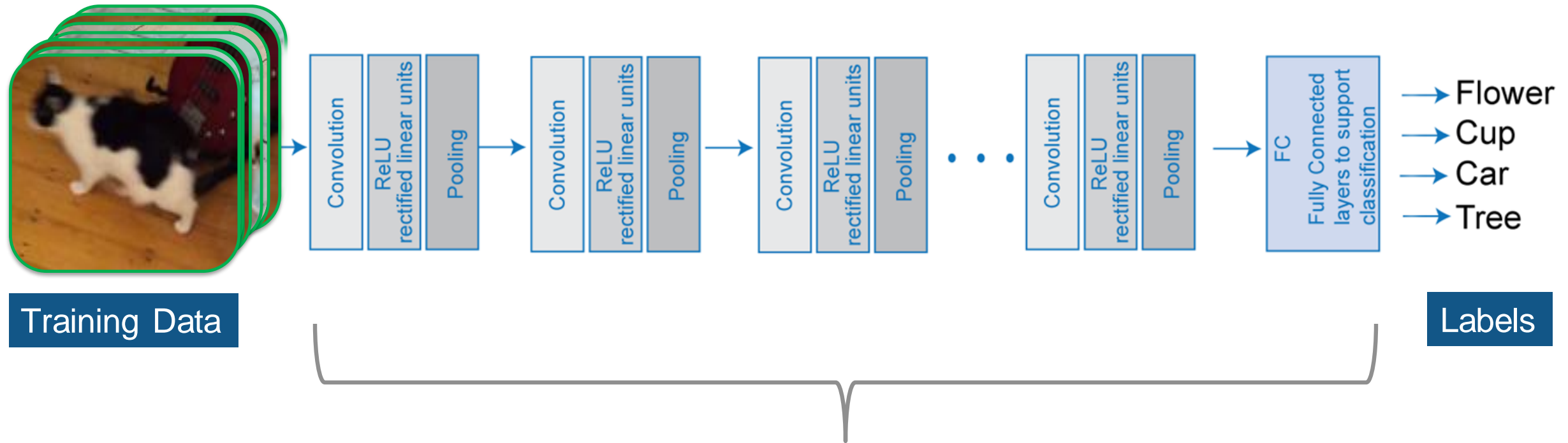
# What is Training?



During training, neural network architectures learn features directly from the data without the need for manual feature extraction

# What Happens During Training?

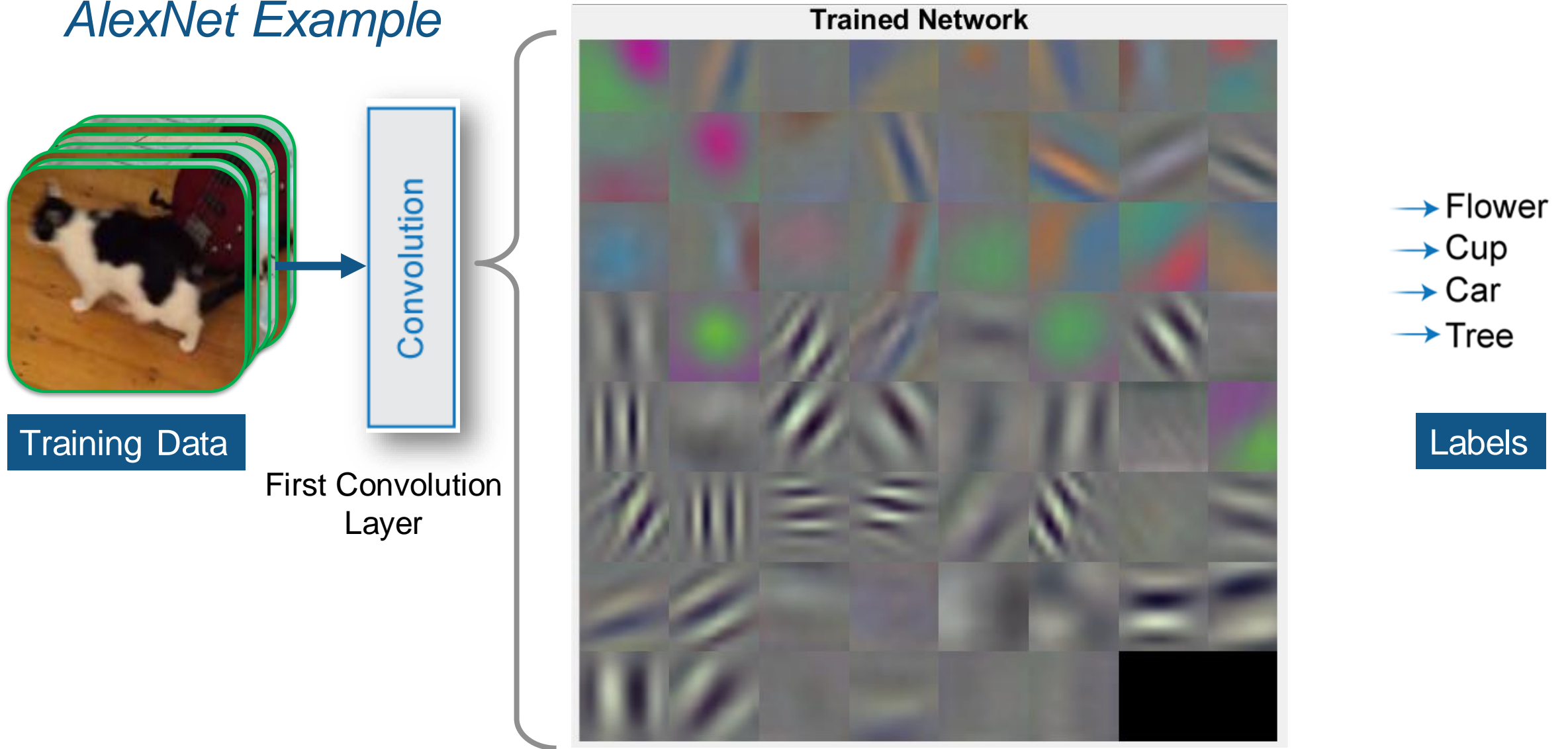
## *AlexNet Example*



**Layer weights are learned  
during training**

# Visualize Network Weights During Training

## *AlexNet Example*





# Visualize Features Learned During Training

## *AlexNet Example*



Sample Training Data

Category: Arctic Fox Epoch 17



Features Learned by Network

# Visualize Features Learned During Training

## *AlexNet Example*

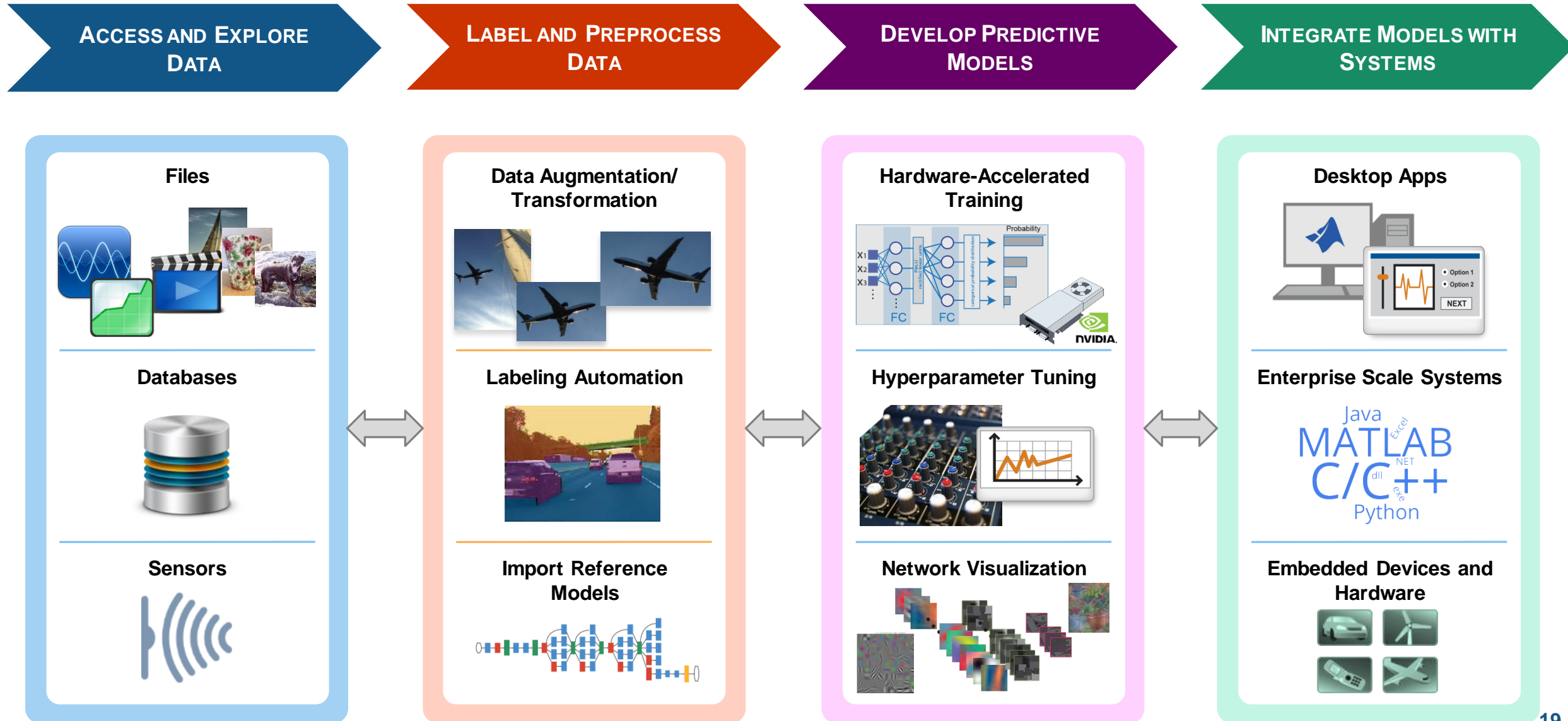


Sample Training Data



Features Learned by Network

# Deep Learning Workflow



# Deep Learning Challenges

## **Data**

- Handling large amounts of data
- Labeling thousands of signals, images & videos
- Transforming, generating, and augmenting data (for different domains)

## **Training and Testing Deep Neural Networks**

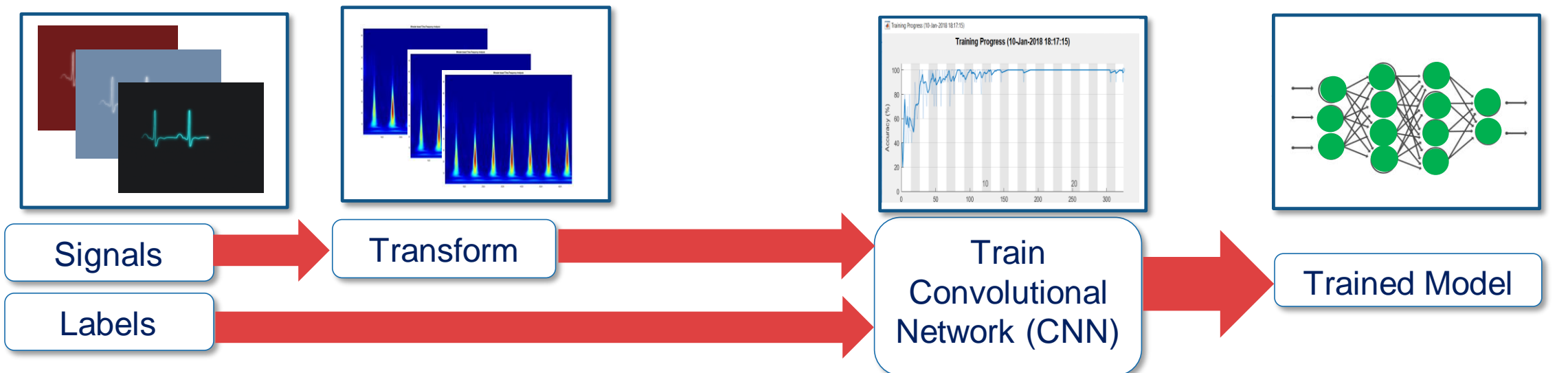
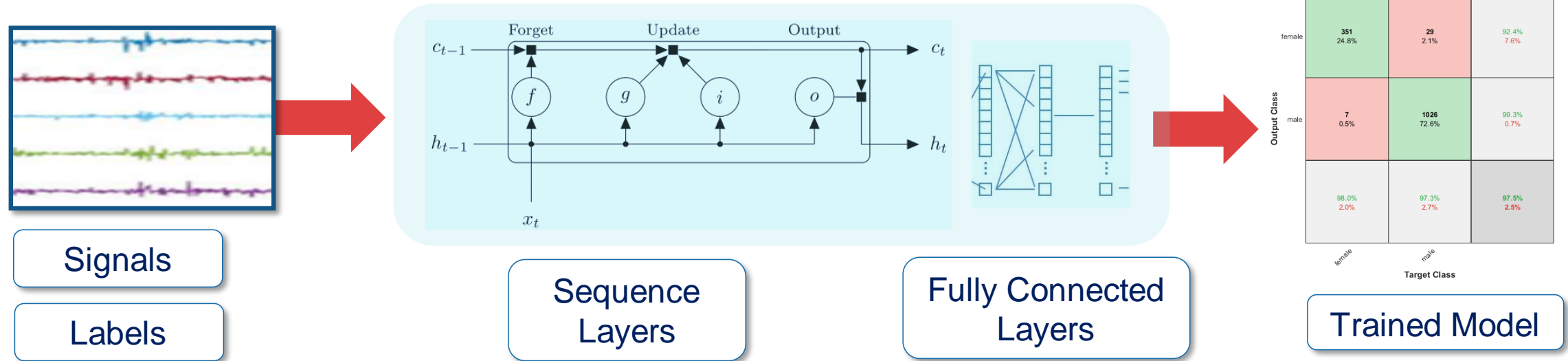
- Accessing reference models from research
- Understanding network behaviour
- Optimizing hyperparameters
- Training takes hours-days

## **Rapid and Optimized Deployment**

- Desktop, web, cloud, and embedded hardware

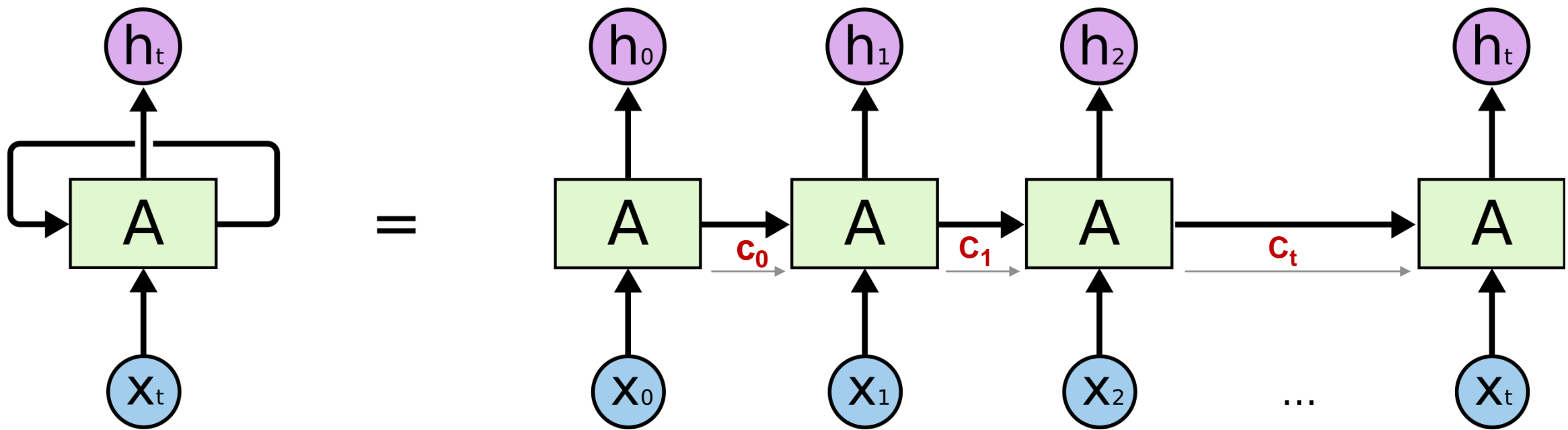


# Working with Signal Data

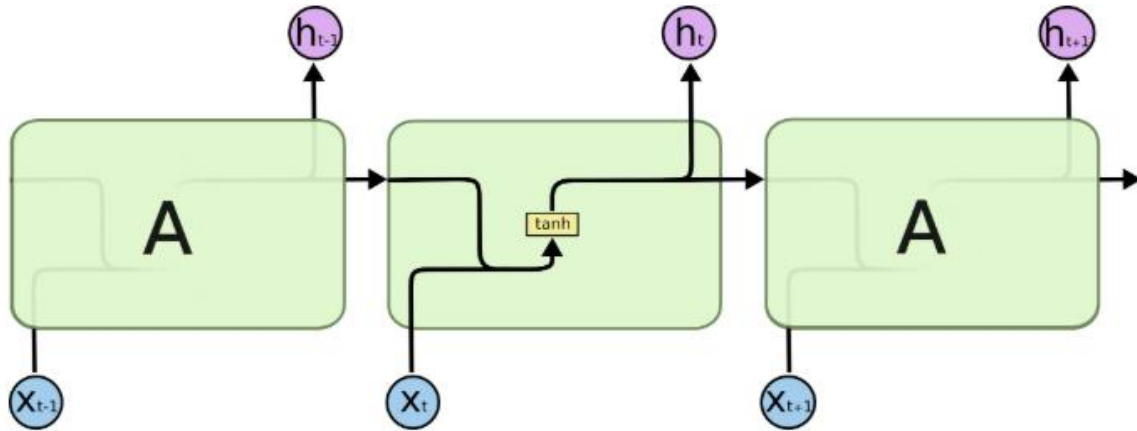


# Long Short Term Memory Networks from RNNs

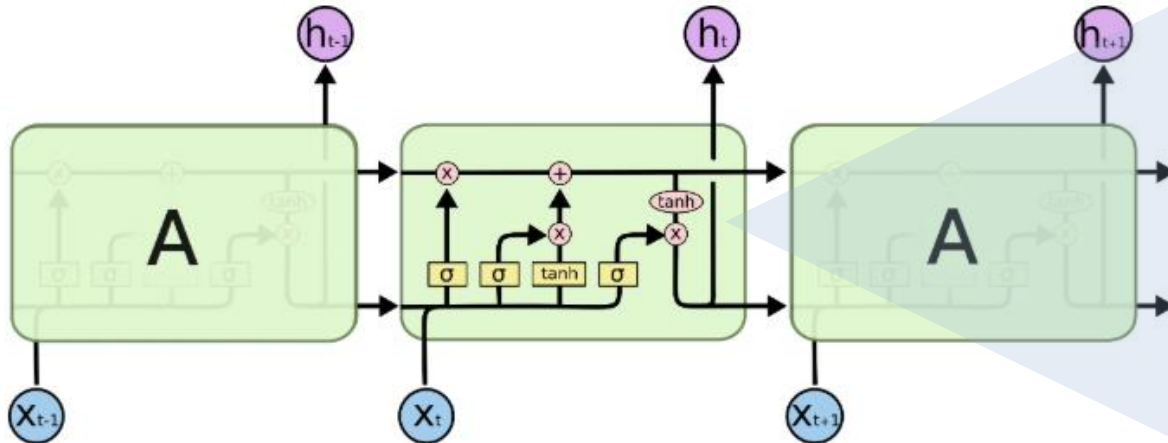
- Recurrent Neural Network that carries a memory cell throughout the process
- Sequence Problems – Long term dependency does not work well with RNNs



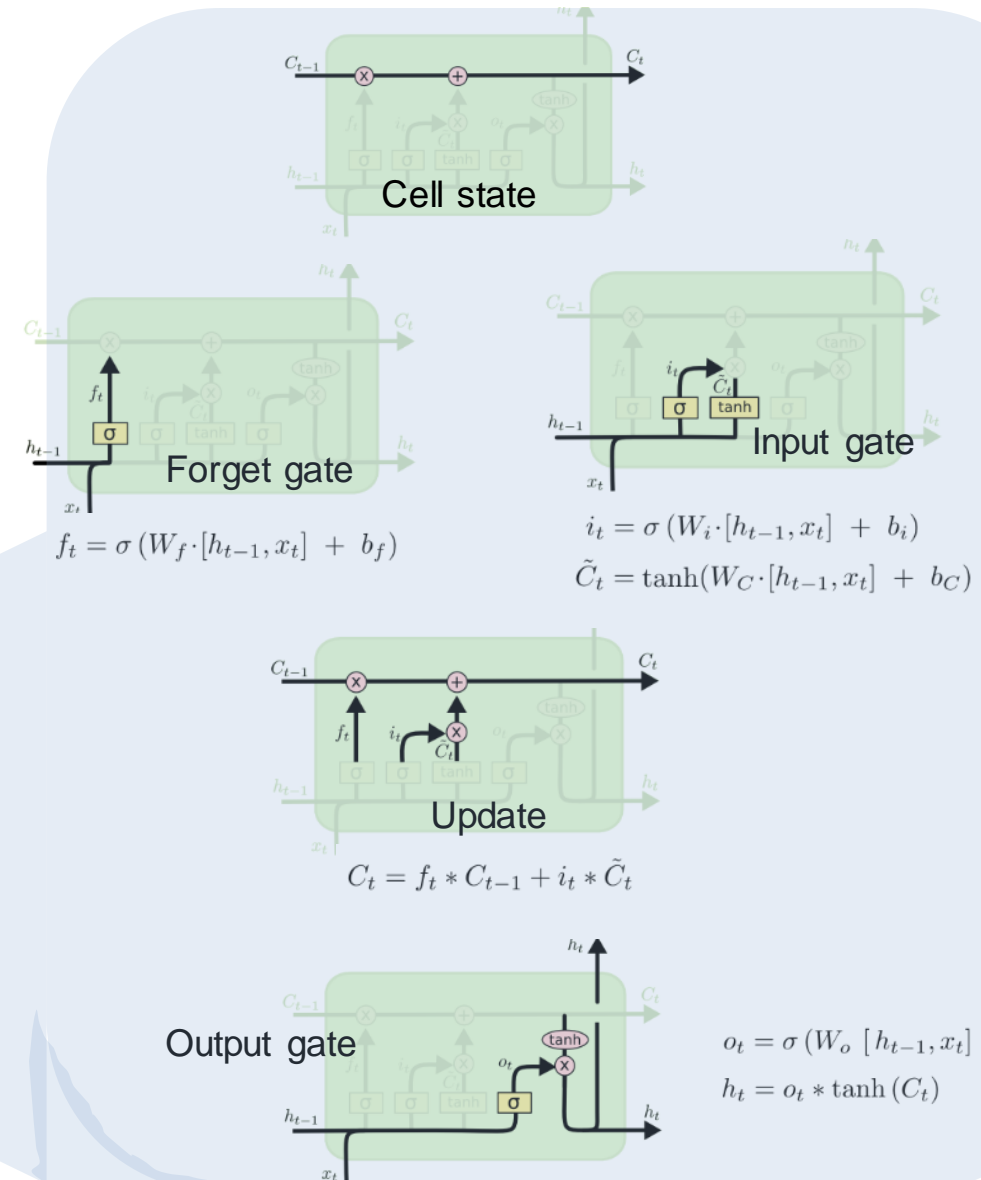
# RNN to LSTM



The repeating module in a standard RNN contains a single layer.

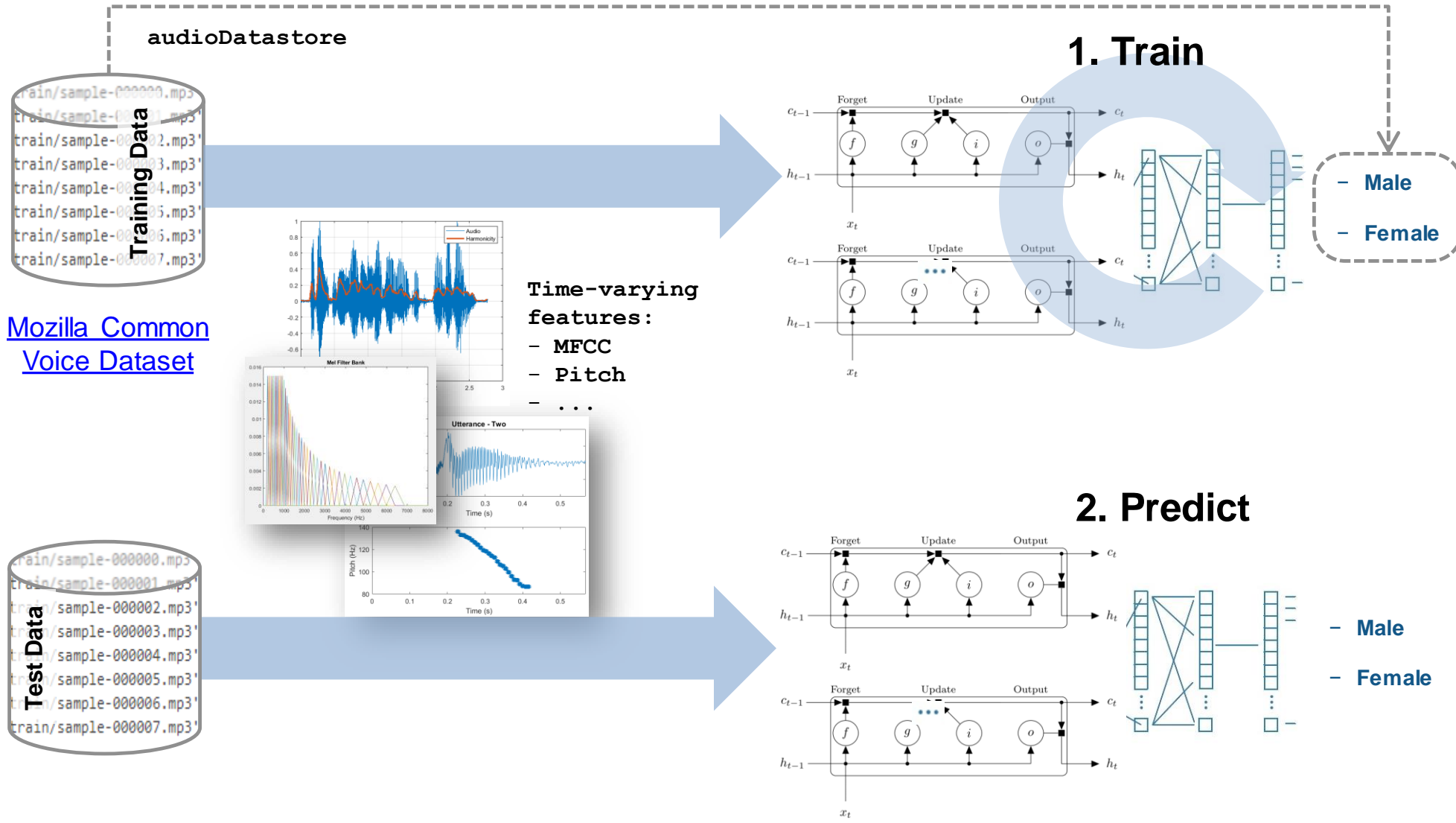


The repeating module in an LSTM contains four interacting layers.



# Example: Speaker Gender Recognition Using Deep Learning LSTM Network for Audio based Speaker Classification

**R2018b**



Testing Accuracy - Majority Rule

Confusion Matrix

	female	male	
female	351 24.8%	29 2.1%	92.4% 7.6%
male	7 0.5%	1026 72.6%	99.3% 0.7%
	98.0% 2.0%	97.3% 2.7%	
female		male	

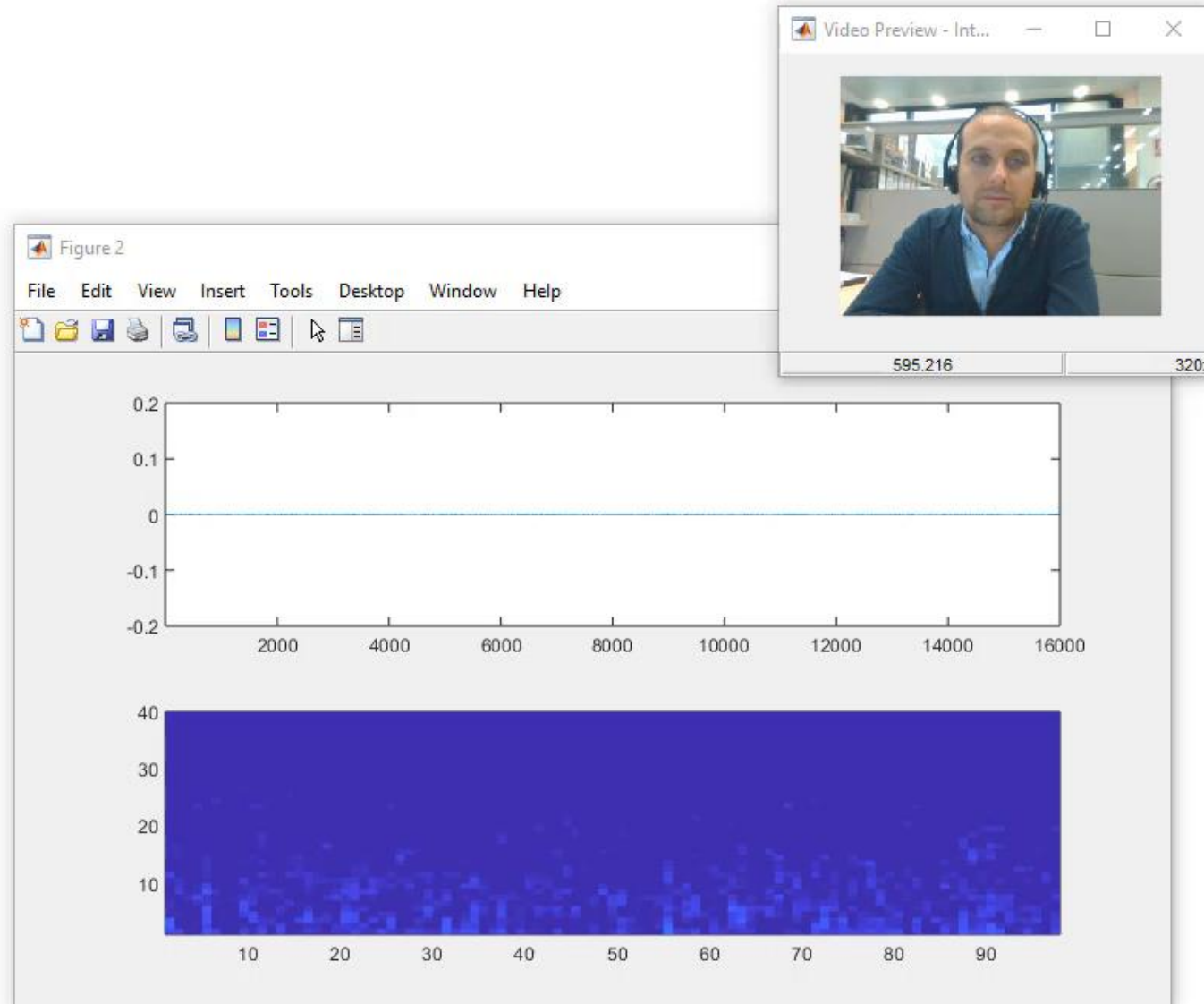
Output Class

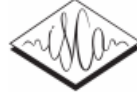
Target Class

**97.5%**  
**2.5%**



# Some audio and speech applications following CNN workflows





# Convolutional Neural Networks for Small-footprint Keyword Spotting

Tara N. Sainath, Carolina Parada

Google, Inc. New York, NY, U.S.A

{tsainath,

## Abstract

We explore using Convolutional Neural Networks (CNNs) for a small-footprint keyword spotting (KWS) task. CNNs are attractive for KWS since they have been shown to outperform DNNs with far fewer parameters. We consider two different applications in our work, one where we limit the number of multiplications of the KWS system, and another where we limit the number of parameters. We present new CNN architectures to address the constraints of each application. We find that CNN architectures offer between a 27-44% relative improvement in false reject rate compared to a DNN, while fitting the constraints of each application.

## 1. Introduction

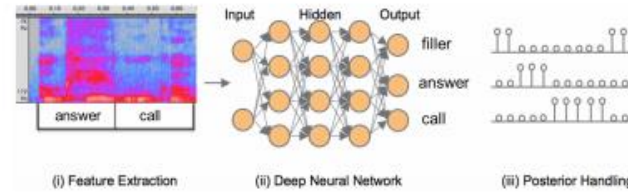


Figure 1: Framework of Deep KWS system, components from left to right: (i) Feature Extraction (ii) Deep Neural Network (iii) Posterior Handling

## 3. CNN Architectures

In this section, we describe CNN architectures as an alternative to the DNN described in Section 2. The feature extraction and posterior handling stages remain the same as Section 2.

### 3.1. CNN Description

A typical CNN architecture is shown in Figure 2. First, we are given an input signal  $V \in \mathbb{R}^{t \times f}$ , where  $t$  and  $f$  are the input feature dimension in time and frequency respectively. A weight

The second convolutional filter has a filter frequency, and no max-pooling is performed.

For example, in our task if we want a number of parameters below 250K, a typical architecture is shown in Table 1. We will refer to this architecture as `cnn-trad-fpool3` in this paper. In Section 5, we will show the benefit of this architecture, particularly the pooling in frequency, compared to other architectures.

However, a main issue with this architecture is the number of multiplies in the convolutional layer, which is exacerbated in the second layer because of the large filter size. This type of architecture is infeasible for power-of-two footprint KWS tasks where multiplies are limited by power-of-two multiplies, other architectures which pool in time are more suited for KWS. Below we present alternative architectures to address the tasks of limiting parameters.

type	m	r	n	p	q
conv	20	8	64	1	3
conv	10	4	64	1	1

model	layer	m	r	n	s	q	Params
cnn-tstride2	conv	16	8	78	2	3	10.0K
	conv	9	4	78	1	1	219.0K
	lin	-	-	32	-	-	20.0K
cnn-tstride4	conv	16	8	100	4	3	12.8K
	conv	5	4	78	1	1	200.0K
	lin	-	-	32	-	-	25.6K
cnn-tstride8	conv	16	8	126	8	3	16.1K
	conv	5	4	78	1	1	190.5K
	lin	-	-	32	-	-	32.2K

Table 4: CNNs for Striding in Time

### 3.4.2. Pooling in Time

An alternative to striding the filter in time is to pool in time, by a non-overlapping amount. Table 5 shows configurations as we vary the pooling in time  $p$ . We will refer to these architectures as `cnn-tpool2` and `cnn-tpool4`. For simplicity, we have omitted certain variables held constant for all experiments, namely time and frequency stride  $s = 1$  and  $v = 1$ . Notice that by pooling in time, we can increase the number of feature maps  $n$  to keep the total number of parameters constant.

model	layer	m	r	n	p	q	Params
cnn-tpool2	conv	21	8	94	2	3	5.6M
	conv	6	4	94	1	1	1.8M
	lin	-	-	32	-	-	65.5K
cnn-tpool3	conv	15	8	94	3	3	7.1M
	conv	6	4	94	1	1	1.6M
	lin	-	-	32	-	-	65.5K

Table 5: CNNs for Pooling in Time

# Solution2: Speech Command Recognition with Deep Learning(MATLAB)

Raw Data

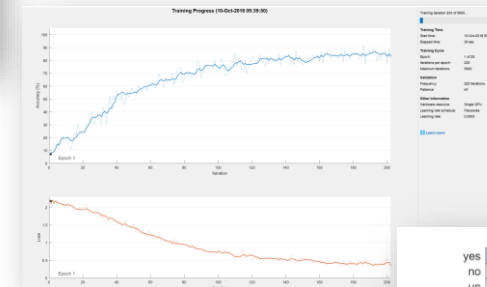
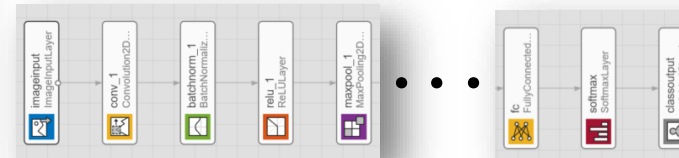
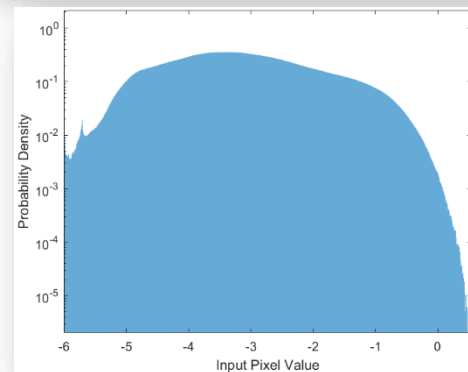
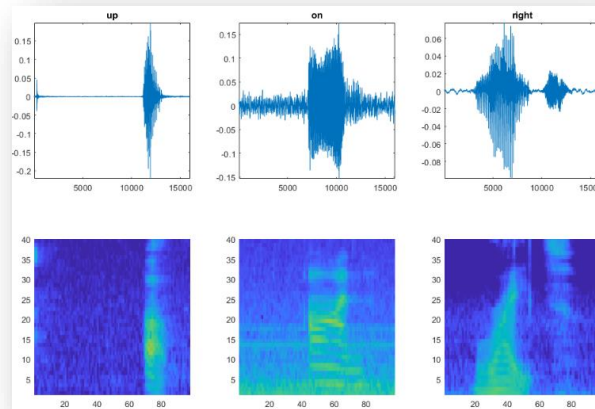
Preprocessed Data

Models and Algorithms

Integrated Analytic Systems

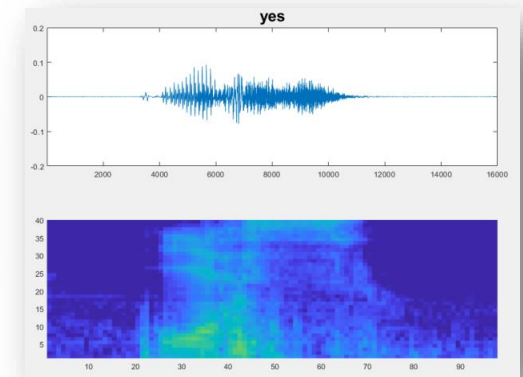
no  
off  
on  
one  
right

0a7c2a8d\_nohash\_0.wav  
0a7c2a8d\_nohash\_1.wav  
0a9f9af7\_nohash\_0.wav  
0ab3b47d\_nohash\_0.wav  
0ac15fe9\_nohash\_0.wav  
0ac15fe9\_nohash\_1.wav  
0b09edd3\_nohash\_0.wav  
0b40aa8e\_nohash\_0.wav  
00b01445\_nohash\_0.wav  
0bd689d7\_nohash\_0.wav  
0bde966a\_nohash\_0.wav

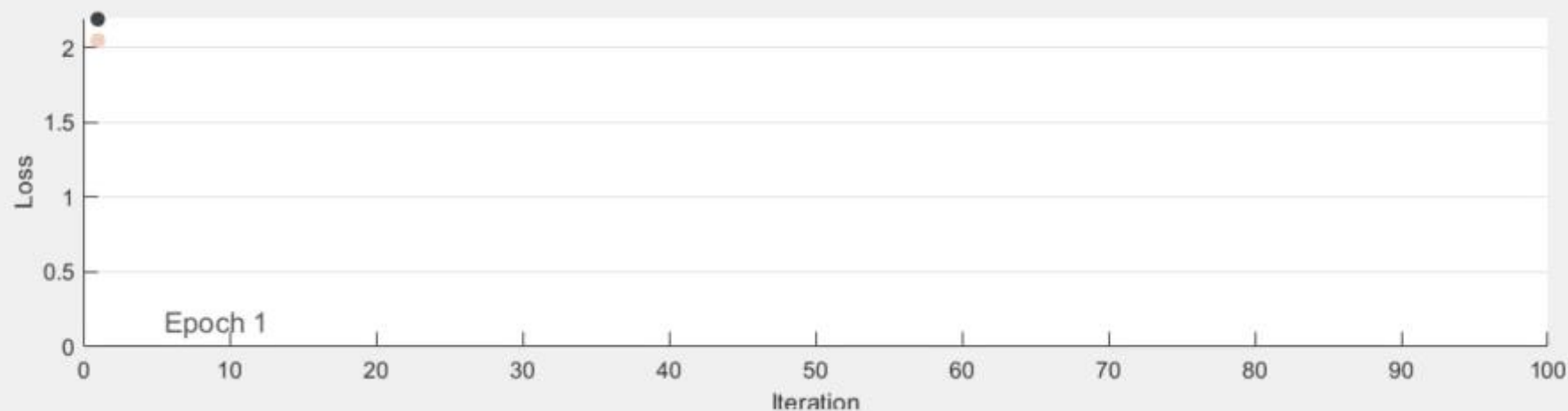
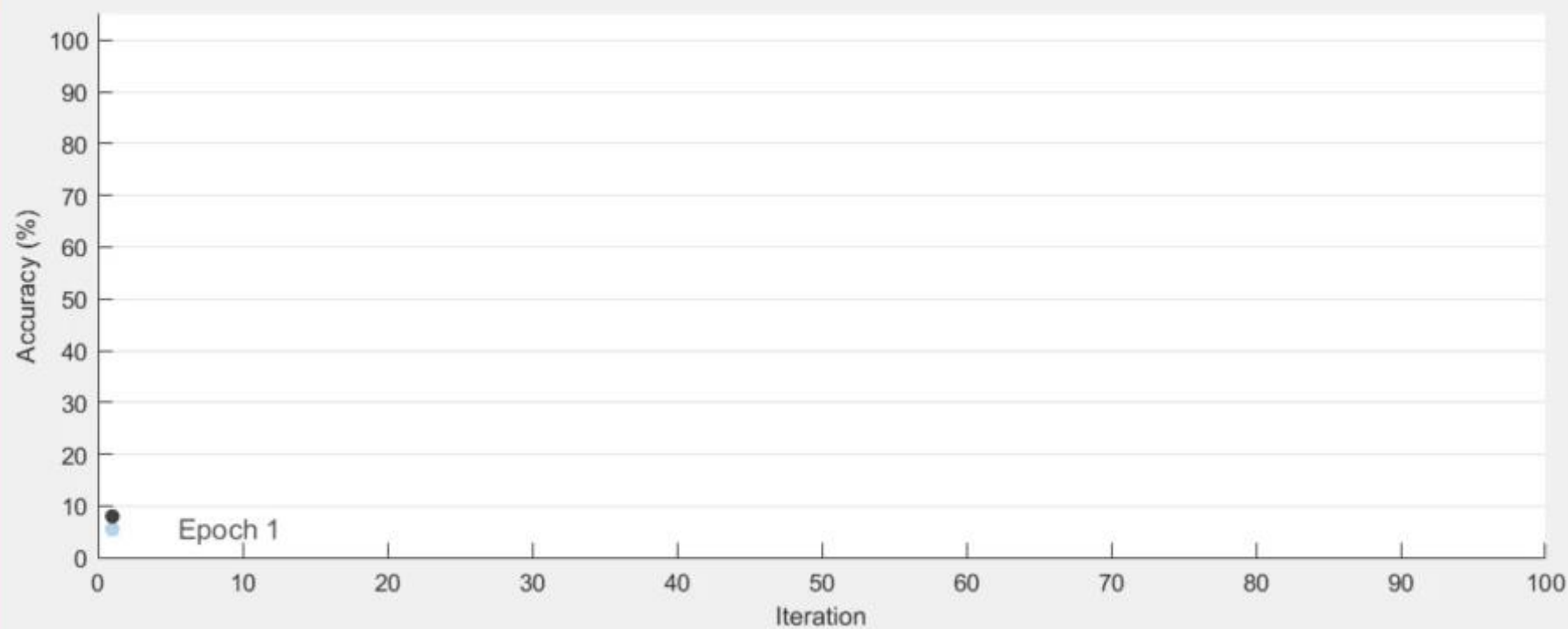


Confusion Matrix for Validation Data

	yes	no	up	down	left	right	on	off	stop	go	unknown	background
yes	251	2	9	4	1	1	7	1	6			
no	1	232	9	4	1	1	7	1	6			
up	1	5	1	255	1	2	1	4	7			
down	1	5	1	255	1	2	1	4	7			
left	1	2	1	240	3	2	10	1	1			
right	1	2	1	240	3	2	10	1	1			
on	1	2	1	240	3	2	10	1	1			
off	2	1	2	1	242	3	1	235	3			
stop	3	1	1	3	2	252	3	247	4			
go	1	1	1	2	2	252	3	2				
unknown	14	16	7	11	4	7	16	10	8	9	804	1
background												400
	90.0%	88.2%	94.9%	89.2%	93.8%	94.8%	93.3%	95.1%	89.8%	94.7%	96.8%	97.3%
	7.0%	10.8%	5.1%	10.8%	6.3%	5.1%	6.7%	4.9%	11.2%	5.3%	3.4%	2.7%



## Training Progress (08-Oct-2018 12:53:57)



Training iteration 1 of 5500...

## Training Time

Start time: 08-Oct-2018 12:53:57

Elapsed time: 0 sec

## Training Cycle

Epoch: 0 of 25

Iterations per epoch: 220

Maximum iterations: 5500

## Validation

Frequency: 220 iterations

Patience: Inf

## Other Information

Hardware resource: Single GPU

Learning rate schedule: Piecewise

Learning rate: 0.0003

## Accuracy

Training (smoothed)

Training

Validation

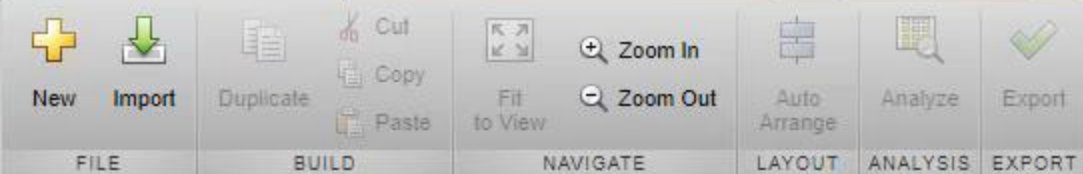
## Loss

Training (smoothed)

Training



Validation










## LAYERS

## INPUT

-  ImageInputLayer
-  SequenceInputLayer


## LEARNABLE

-  Convolution2DLayer
-  TransposedConvolution2DLayer
-  FullyConnectedLayer
-  LSTMLayer
-  BiLSTMLayer

## ACTIVATION

-  ReLULayer
-  LeakyReLULayer
-  ClippedReLULayer

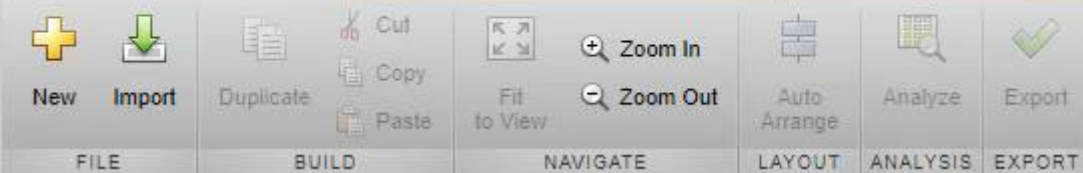
## NORMALIZATION AND DROPOUT

-  BatchNormalizationLayer

## PROPERTIES

Number of layers	0
Number of connections	0
Input type	None
Output type	None

## DEEP NETWORK DESIGNER



## LAYERS

## INPUT



ImageInputLayer



SequenceInputLayer

## LEARNABLE



Convolution2DLayer



TransposedConvolution2DLayer



FullyConnectedLayer



LSTMLayer



BiLSTMLayer

## ACTIVATION



ReLULayer



LeakyReLULayer



ClippedReLULayer

## NORMALIZATION AND DROPOUT

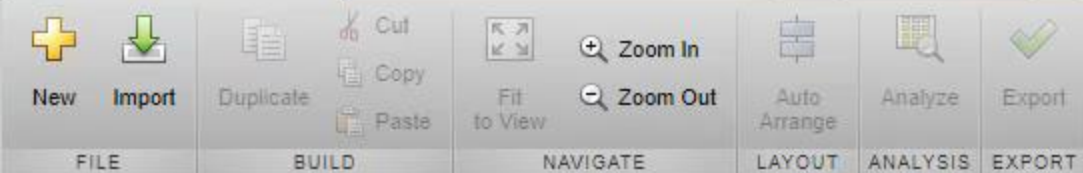


BatchNormalizationLayer





## PROPERTIES

Number of layers	0
Number of connections	0
Input type	None
Output type	None








## LAYERS

## INPUT

-  ImageInputLayer
-  SequenceInputLayer


## LEARNABLE

-  Convolution2DLayer
-  TransposedConvolution2DLayer
-  FullyConnectedLayer
-  LSTMLayer
-  BiLSTMLayer

## ACTIVATION

-  ReLULayer
-  LeakyReLULayer
-  ClippedReLULayer

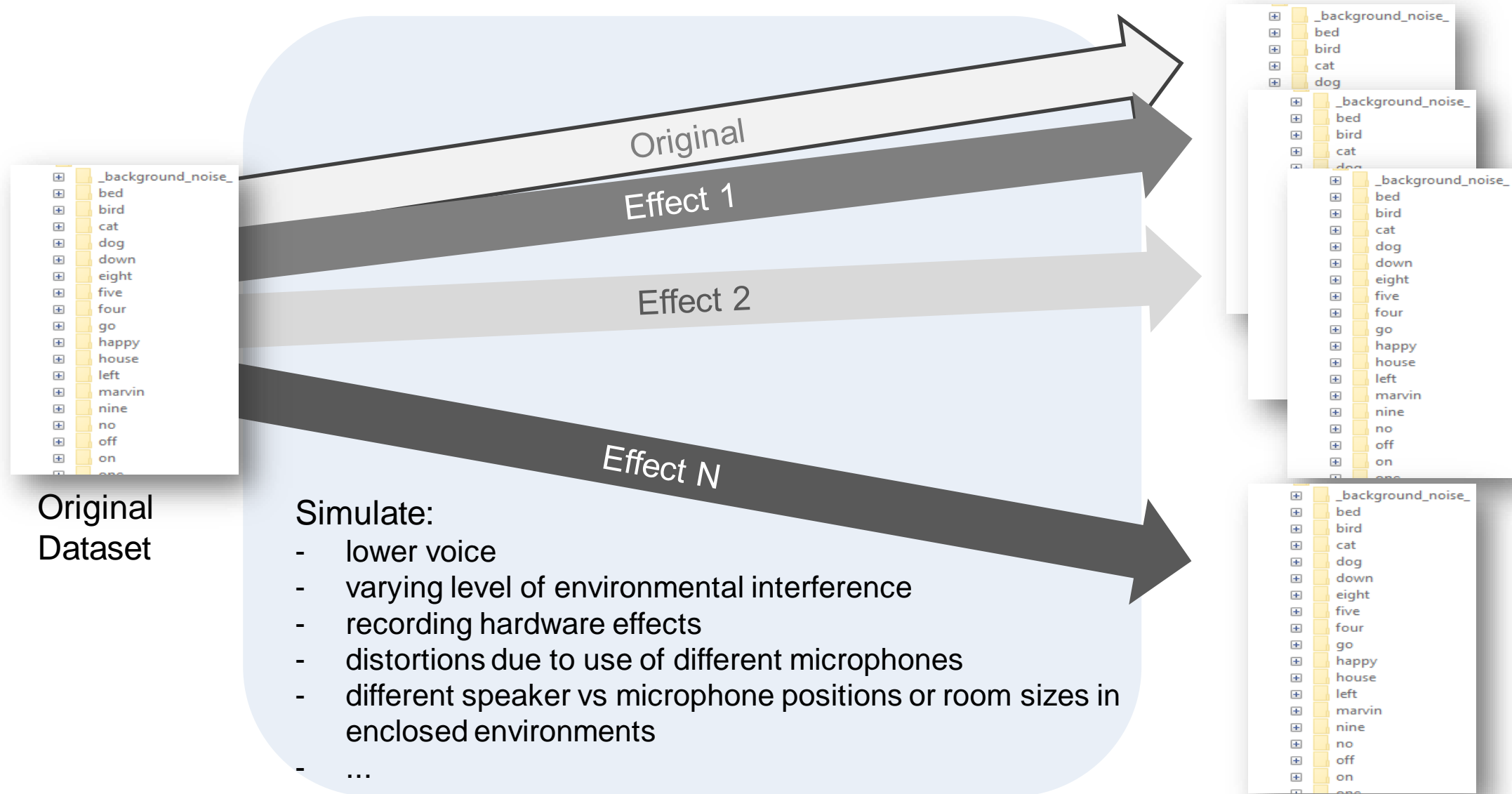
## NORMALIZATION AND DROPOUT

-  BatchNormalizationLayer

## PROPERTIES

Number of layers	0
Number of connections	0
Input type	None
Output type	None

# Data augmentation allows training more advanced networks and generating more robust models







Serial

```
% Cycle continuously and automatically through files in datastore
mfccFile = zeros(numel(ads.Files),numMfcc)
while hasdata(ads)
    [data,info] = read(ads);
    % do something with data
end
```

Parallel

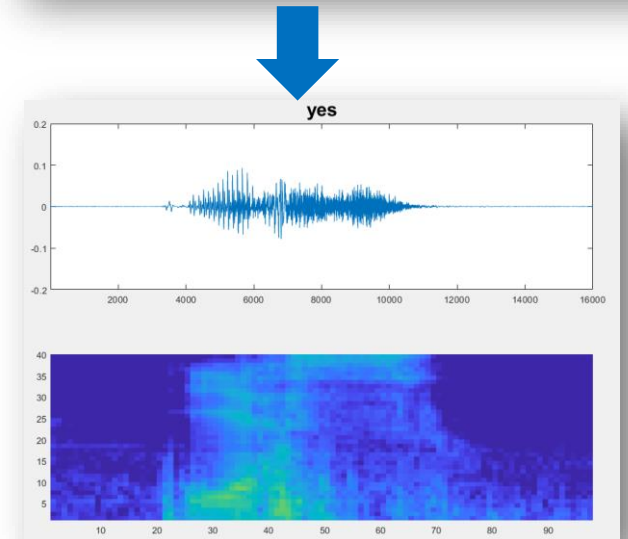
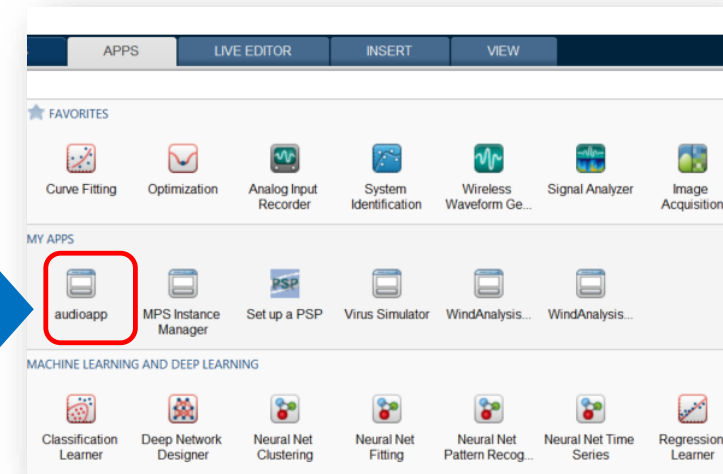
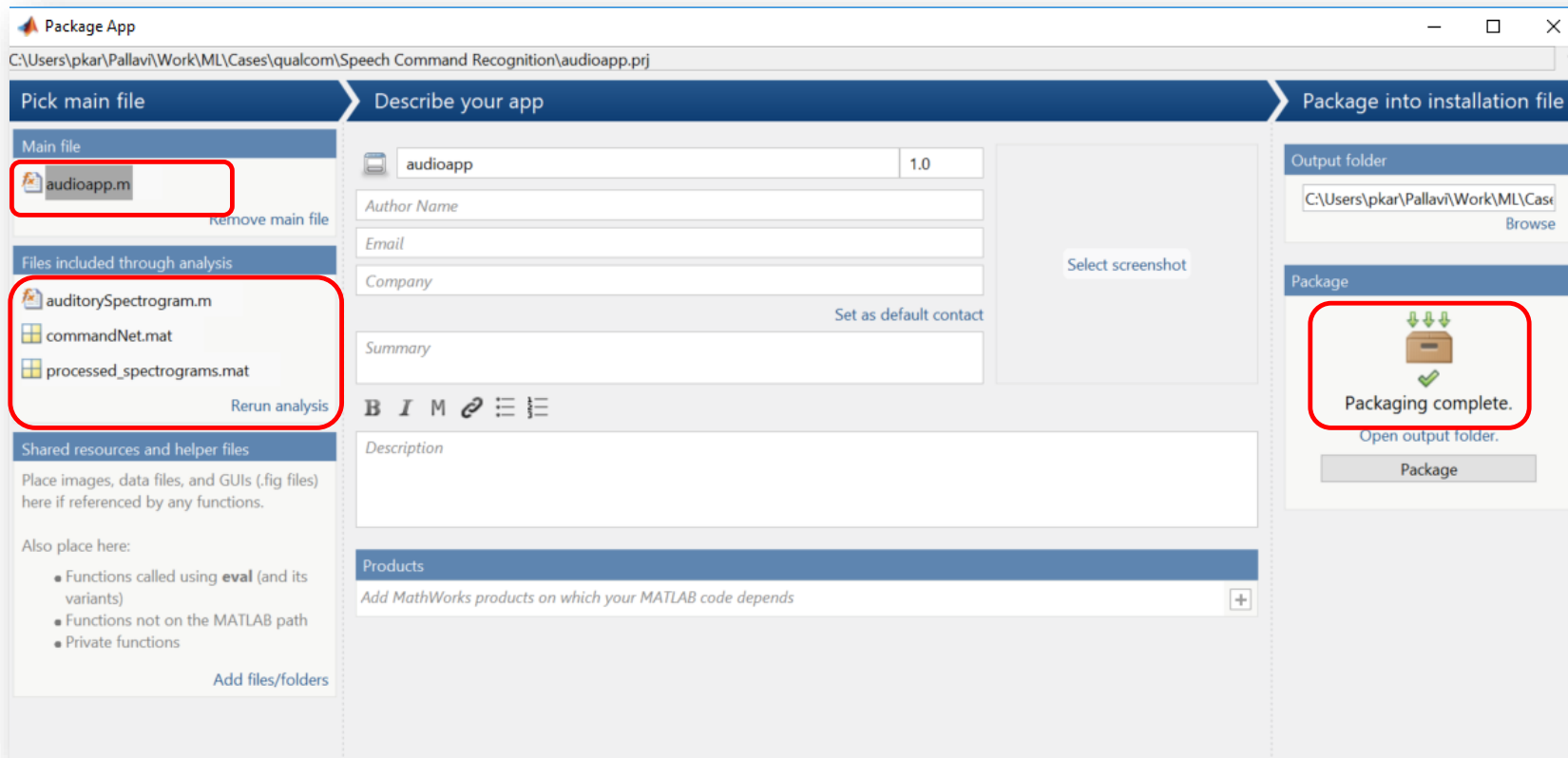
```
% Partition and explicit parfor parallel processing pattern
N = numpartitions(ads);

parfor index=1:N
    subads = partition(ads,N,index);
    while hasdata(subads)
        data = read(subads);
        % do something with data
    end
end
```

```
% Tall array and "implicit" parallel processing pattern
T = tall(ads);

dcOffsets = cellfun(@(x) mean(x), T, 'UniformOutput', false);
gather(dcOffsets);
```

# Package Speech Recognition App



## Commands

- Yes
- No
- Up
- Down
- Left
- Right
- On
- Off
- Stop
- Go

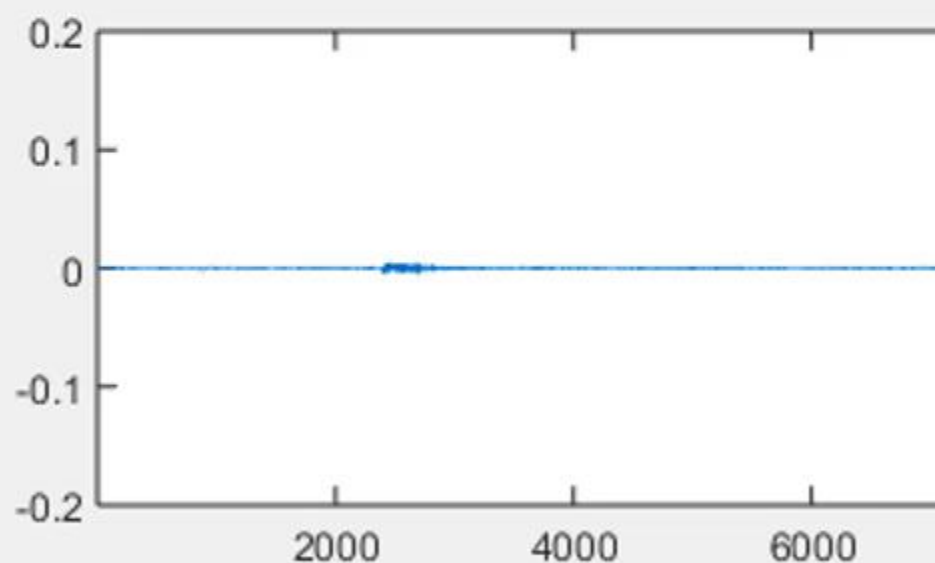
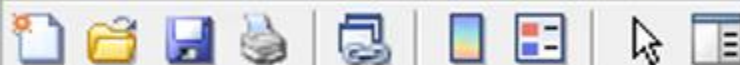
## Non-Commands (= Unknown)

- Bed
- Bird
- Cat
- Dog
- Happy
- House
- Marvin
- Sheila
- Tree
- Wow
- Zero
- One
- Two
- Three
- Four



Figure 2

File Edit View Insert Tools Desktop Window Help



40  
30

BREAK



# Deep Learning Challenges

## Data

Not a deep learning expert

- ✓ Handling large amounts of data
  - Labeling thousands of signals, images & videos
  - Transforming, generating, and augmenting data (for different domains)

## Training and Testing Deep Neural Networks

- ✓ Understanding network behavior
  - Accessing reference models from research
  - Optimizing hyperparameters
  - Training takes hours-days

## Rapid and Optimized Deployment

- Desktop, web, cloud, and embedded hardware

# BREAK

# Segment and label audio signals automatically

## Read speech recording

Load speech recording from (.wav) file

```
1 fileName = 'Counting-16-44p1-mono-15secs.wav';  
2 pathName = fullfile(matlabroot, 'toolbox', 'audio', 'samples', fileName);  
3 [x,fs] = audioread(pathName);
```

Plot samples over time

```
4 t = (1/60)*(0:1/60:(15-1/60))-1);  
5 hpl = plot(t, x);  
6 xlabel('time (s)')
```

## Playback content

```
7 soundsc(x,fs)
```

## Segment automatically

Use a custom function based on combined thresholding of signal energy and spectral centroid

```
8 [segm, ~] = findSpeechSegments(x,fs);
```

Plot segmented time intervals

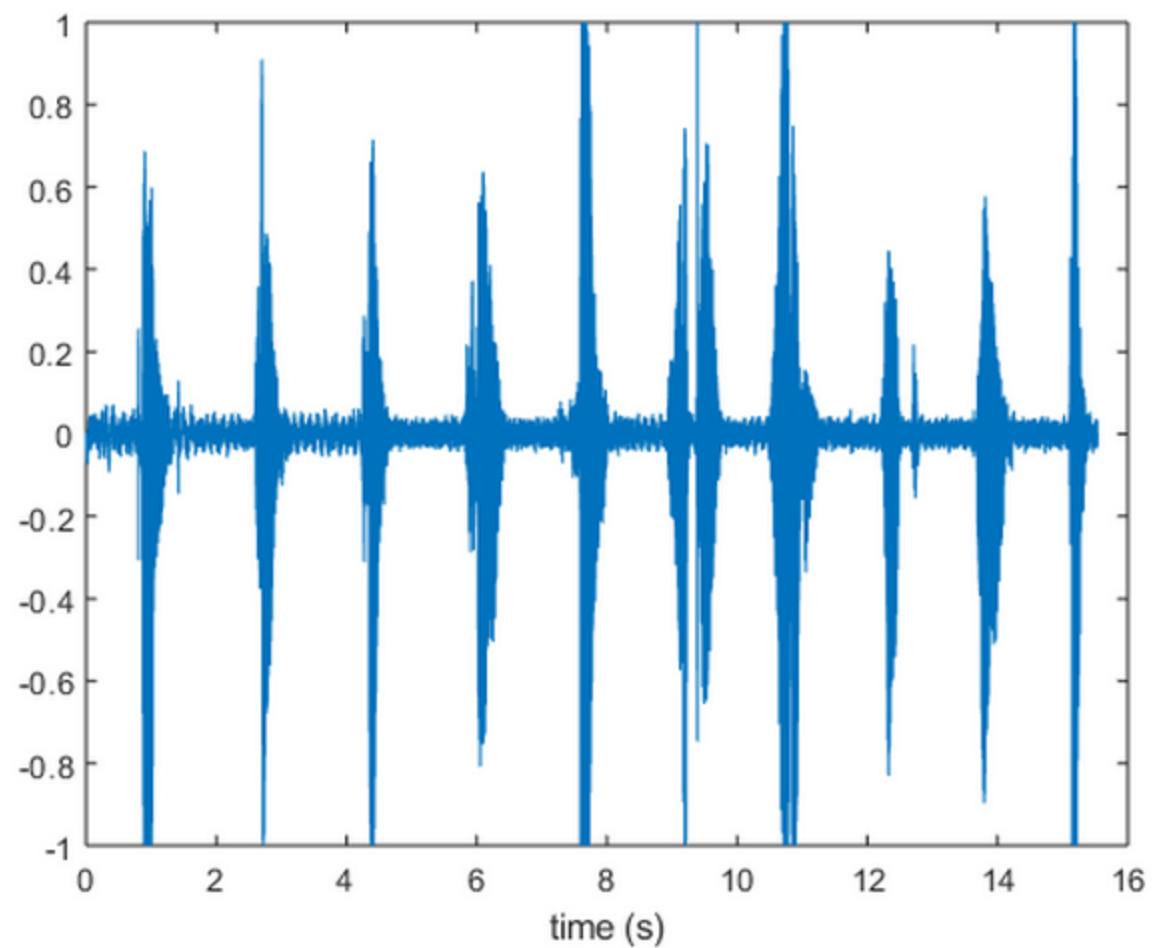
```
9 hold on  
10 hax = hpl.Parent;  
11 xlr = segm(:,:);
```

5

```
hpl = plot(t, x);
```

6

```
xlabel('time (s)')
```



## Playback content

7

```
soundsc(x, fs)
```

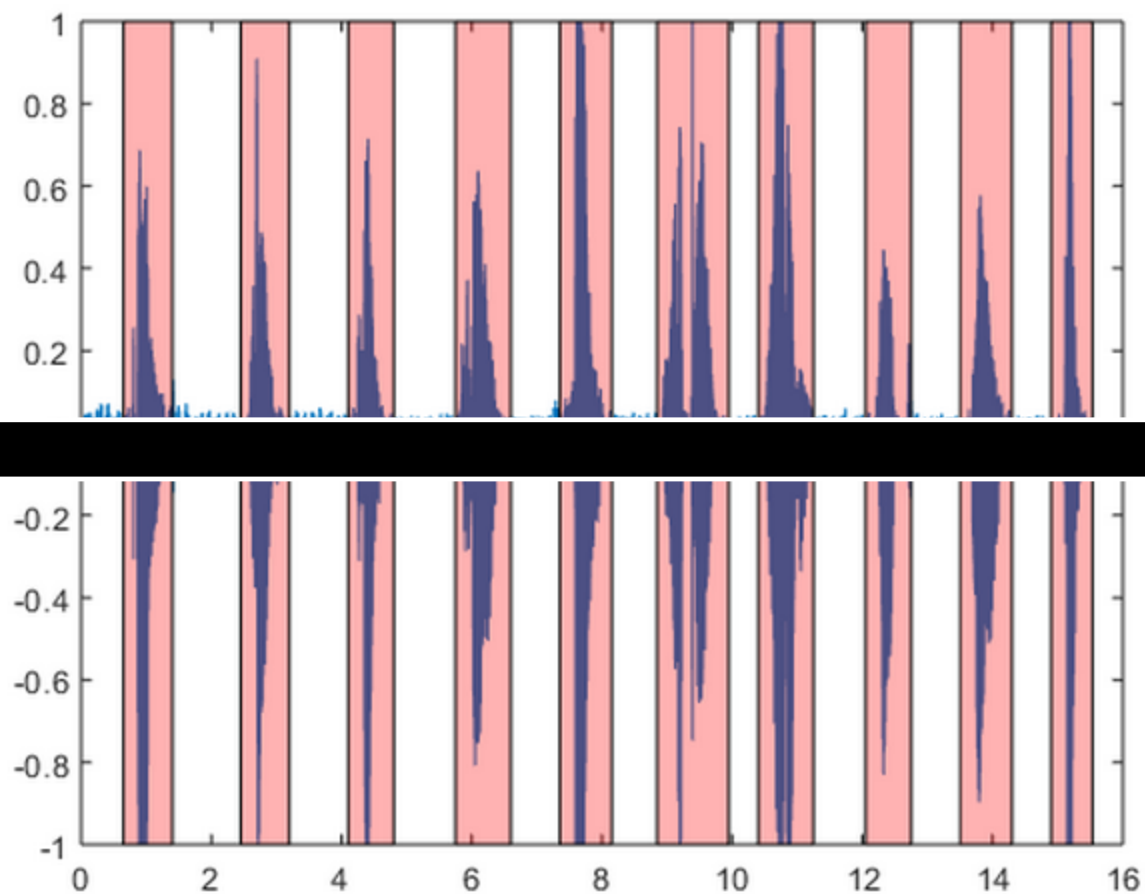
## Segment automatically

Use a custom function based on combined thresholding of signal energy and spectral centroid

```
[segm, ~] = findSpeechSegments(x,fs);
```

Plot segmented time intervals

```
plotSegments(hp1, segm/fs)
```





## Segment automatically

Use a custom function based on combined thresholding of signal energy and spectral centroid

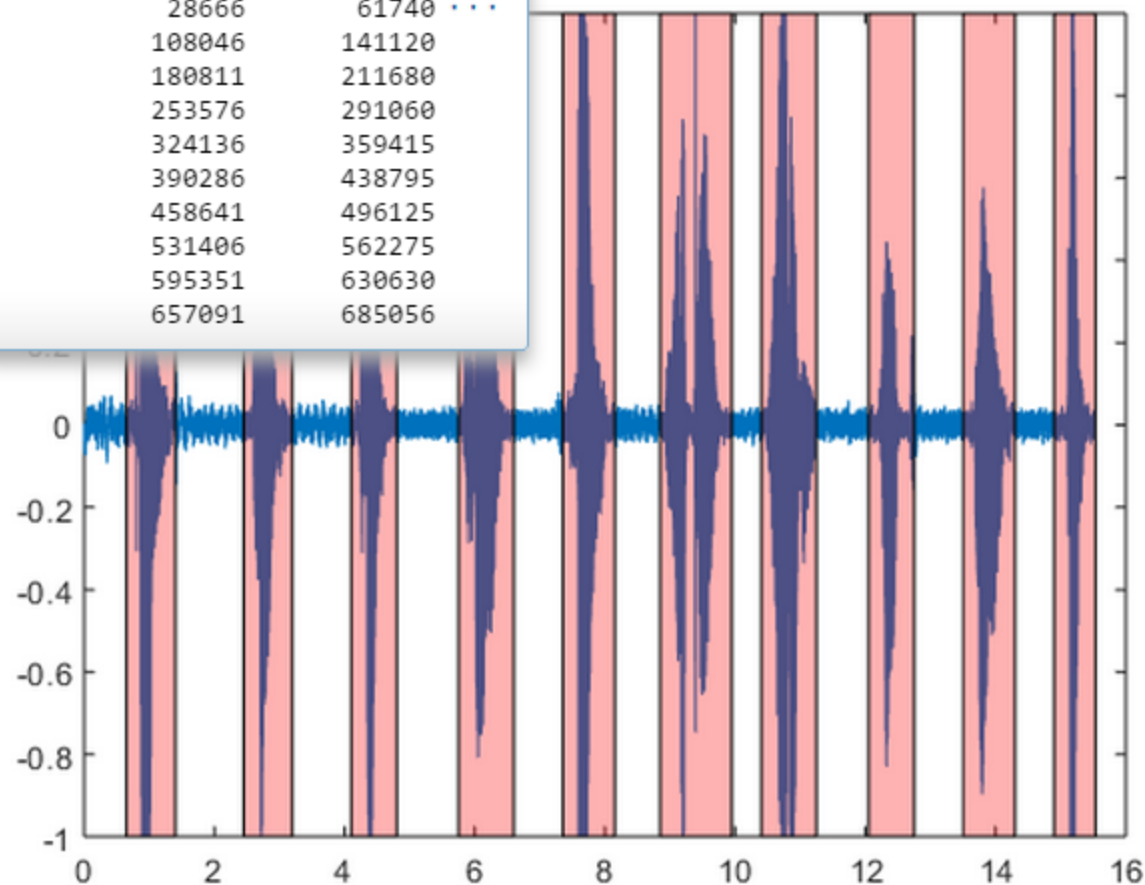
```
[segm, ~] = findSpeechSegments(x,fs);
```

Plot segmented time intervals

```
plotSegments(hp1, segm/fs)
```

segm = 10x2

28666	61740
108046	141120
180811	211680
253576	291060
324136	359415
390286	438795
458641	496125
531406	562275
595351	630630
657091	685056



# Automate labeling with speech content using speech-to-text services

Initialize speech transcription wrapper

```
speechObject = speechClient('Google','languageCode','en-GB');
```

Loop over segments

```
autoLabels = strings(numSegments,1);
```

```
for idx = 1:numSegments
```

Get segment boundary

```
start = segm(idx,1);
```

```
stop = segm(idx,2);
```

```
fprintf('Querying transcript for segment %02d [%5.02f,
```

Get speech transcription using the Google Speech API

```
tableOut = speech2text(speechObject(x(start:stop,1) fs);
```

Store output

```
if(ismember('TRANSCRIPT',tableOut.Properties.VariableNames))
```

```
    autoLabels(idx) = tableOut.TRANSCRIPT(1);
```

```
end
```

```
end
```



Products Solutions Academia Support Community Events

File Exchange

Search File Exchange

MATLAB Central ▾ Files Authors Tags Comments My File Exchange Submit  
About



speech2text

version 1.0 (109 KB) by MathWorks Audio System Toolbox Team

Automatic speech-to-text conversion

## Visualize results in Audio Labeler

Create label definition

## Automate labeling with speech content using speech-to-text services

Initialize speech transcription wrapper

```
10 speechObject = speechClient('Google','languageCode','en-GB');
```

Loop over segments

```
11 autoLabels = strings(numSegments,1);  
12 for idx = 1:numSegments
```

Get segment boundary

```
13     start = segm(idx,1);  
14     stop = segm(idx,2);  
15     fprintf('Querying transcript for segment %02d [%5.02f, %5.02f]s of file "%s"\n', idx, start/fs, stop/fs, fileName)
```

Get speech transcription using the Google Speech API

```
16     tableOut = speech2text(speechObject, x(start:stop,1), fs);
```

Store output

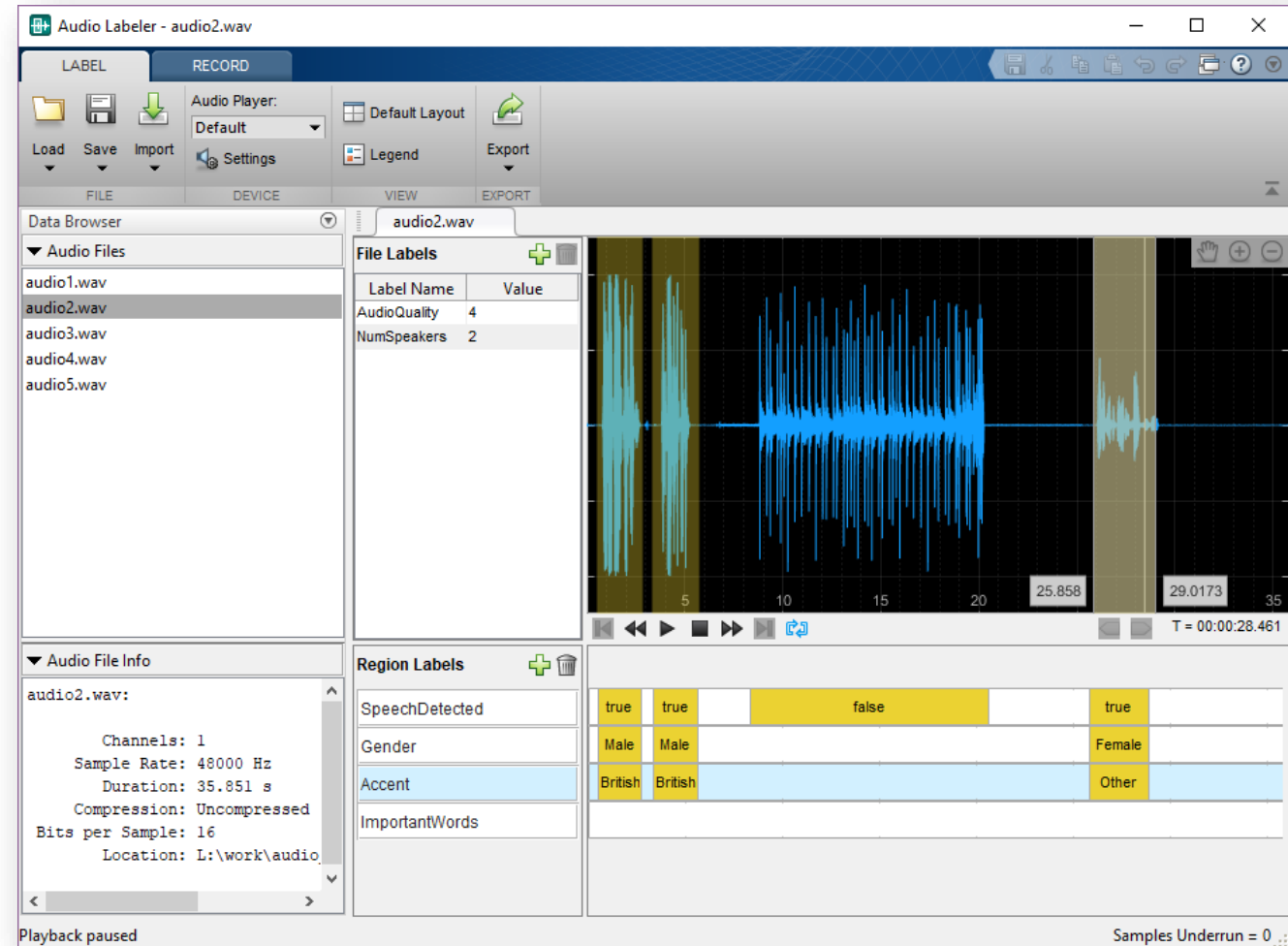
```
17     if(ismember('TRANSCRIPT',tableOut.Properties.VariableNames))  
18         autoLabels(idx) = tableOut.TRANSCRIPT(1);  
19     end  
20  
21 end
```

## Visualize results in Audio Labeler

Create label definition

# Audio Labeler

- Work on collections of recordings or record new audio directly within the app
- Navigate dataset and playback interactively
- Define and apply labels to
  - Entire files
  - Regions within files
- Import and export audio folders, label definitions and datastores



# Apps for Data Labeling



Image Labeler



Video Labeler



Audio Labeler

Image Labeler

06\_highway\_cutin\_20s14

Video Labeler - videoLabelingSessionAtrium

atrim.mp4

tree

Audio Labeler - audio2.wav

audio2.wav

File Labels

Label Name	Value
AudioQuality	4
NumSpeakers	2

Region Labels

SpeechDetected	Gender	Accent	ImportantWords
true	Male	British	
true	Male	British	
false			
true	Female	Other	

Audio File Info

audio2.wav:

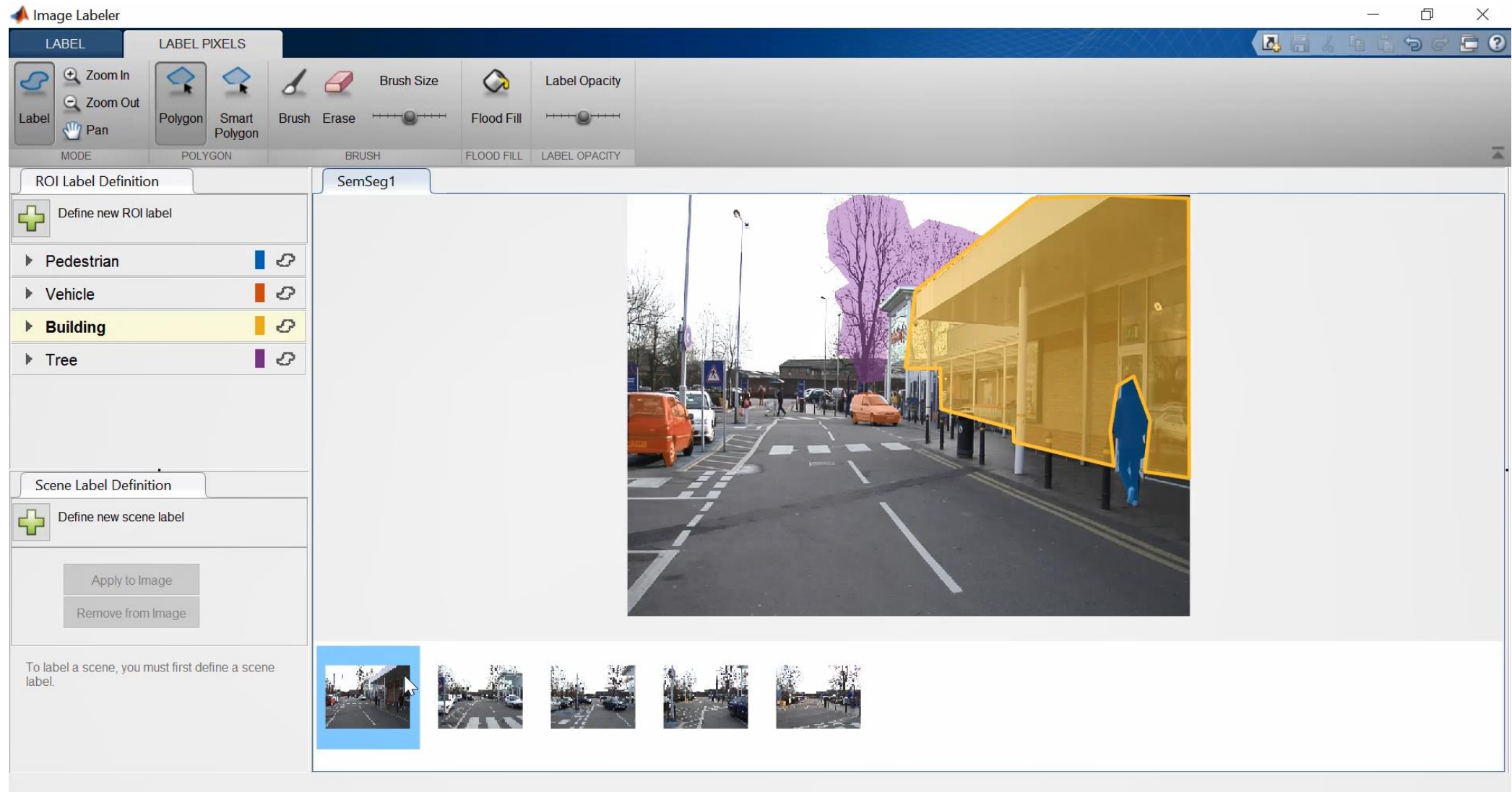
Channels: 1  
Sample Rate: 48000 Hz  
Duration: 35.851 s  
Compression: Uncompressed  
Bits per Sample: 16  
Location: L:\work\audio

Playback paused

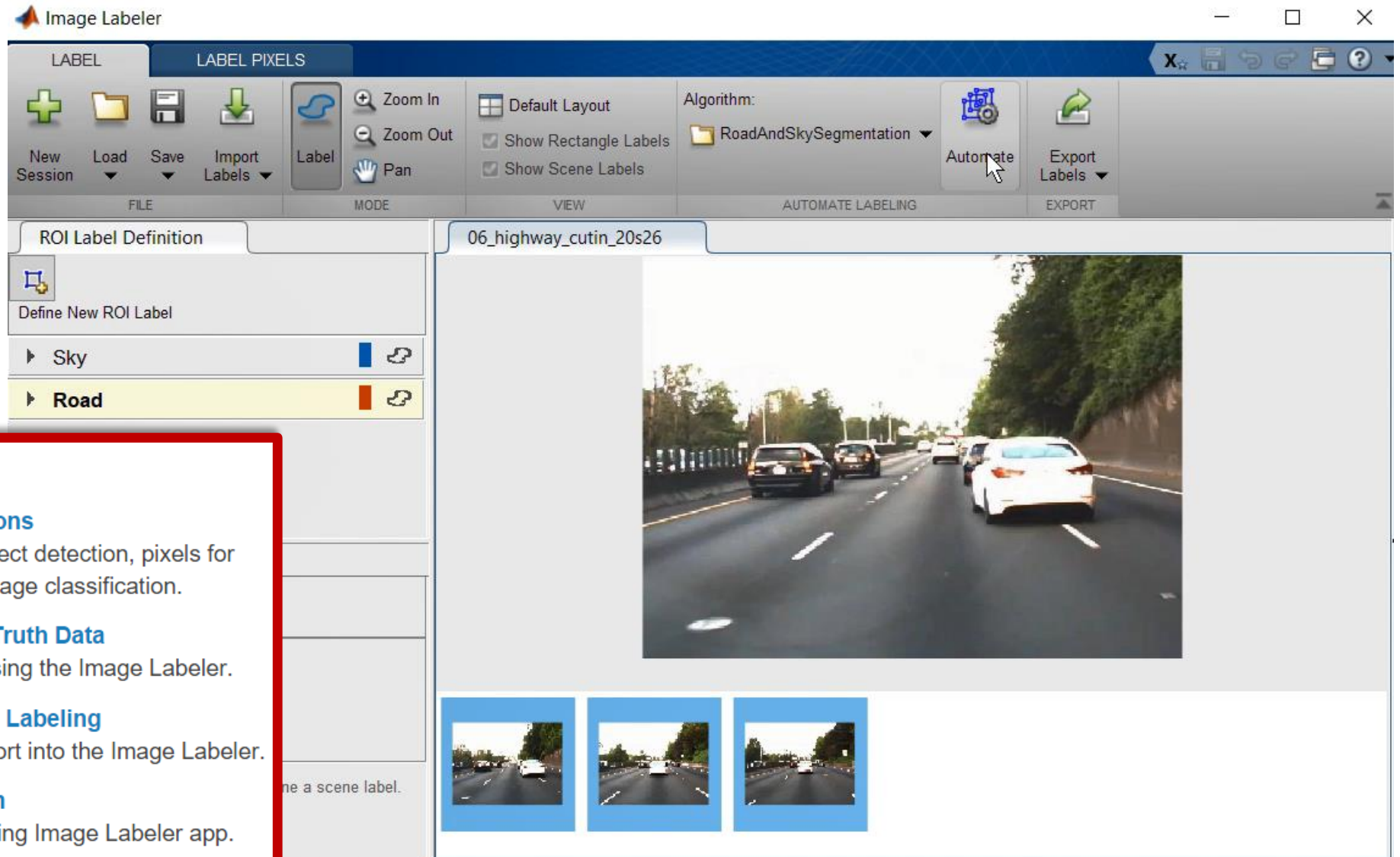
Samples Underrun = 0



# Label Images Using Image Labeler App



# Accelerate Labeling With Automation Algorithms



## [Learn More](#)

### [Define Ground Truth for Image Collections](#)

Interactively label rectangular ROIs for object detection, pixels for semantic segmentation, and scenes for image classification.

### [Train an Object Detector from Ground Truth Data](#)

Create training data for object detection using the Image Labeler.

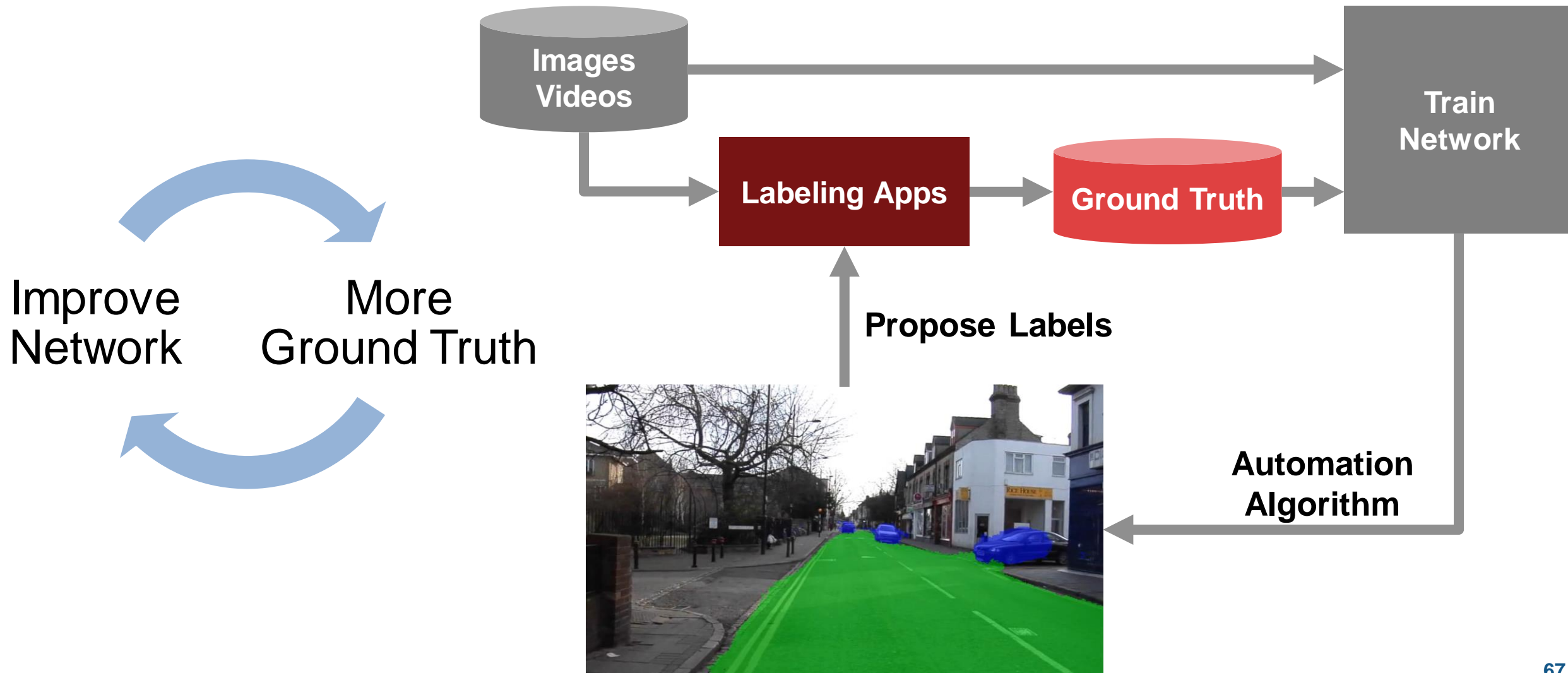
### [Create Automation Algorithm for Image Labeling](#)

Create a custom tracking algorithm to import into the Image Labeler.

### [Label Pixels for Semantic Segmentation](#)

Label pixels for semantic segmentation using Image Labeler app.

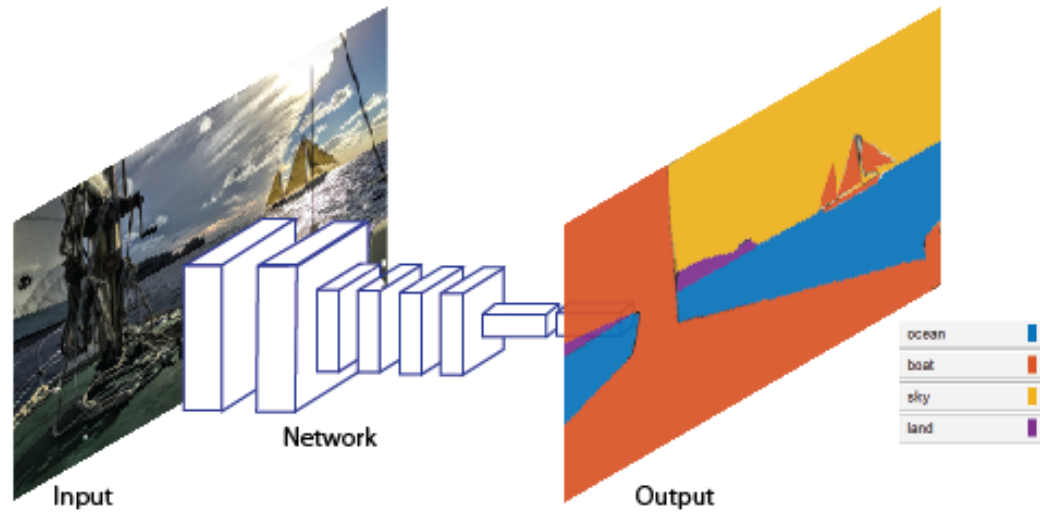
# Perform Bootstrapping to Label Large Datasets





# Example – Semantic Segmentation

[Available Here](#)



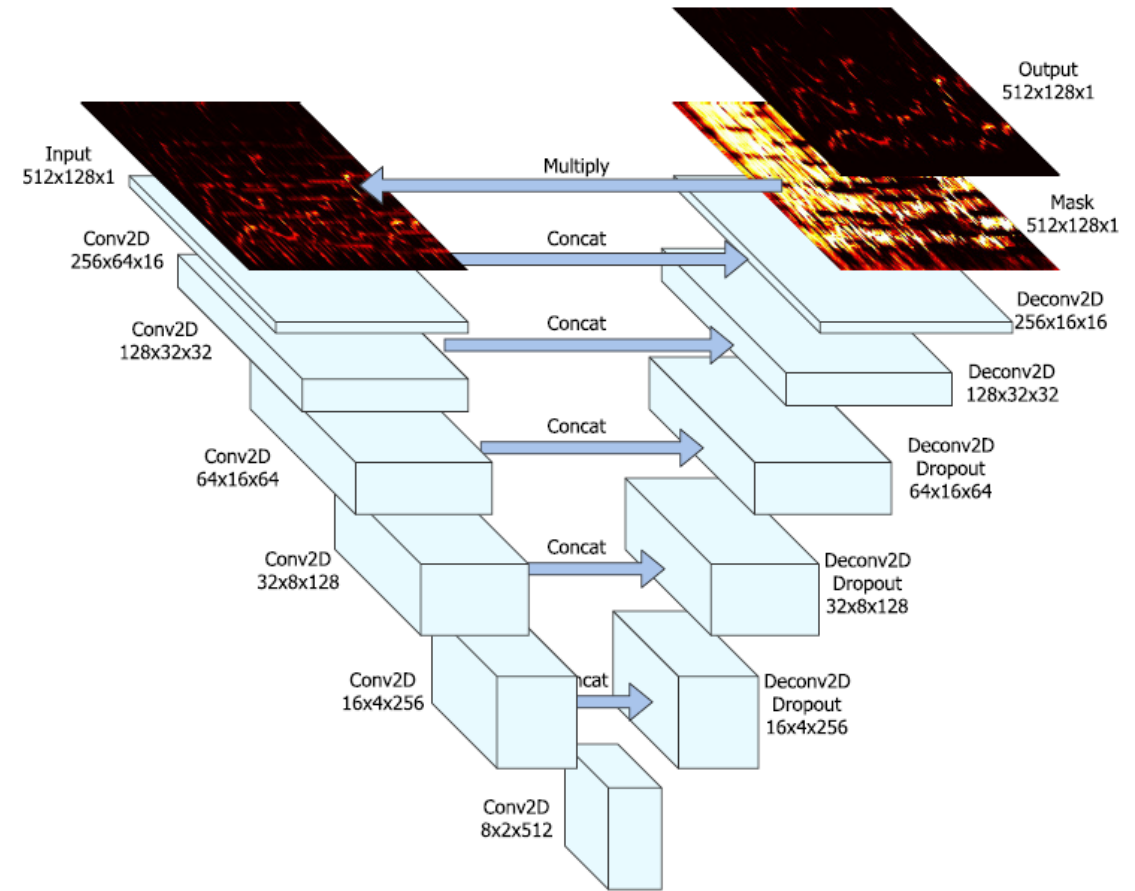
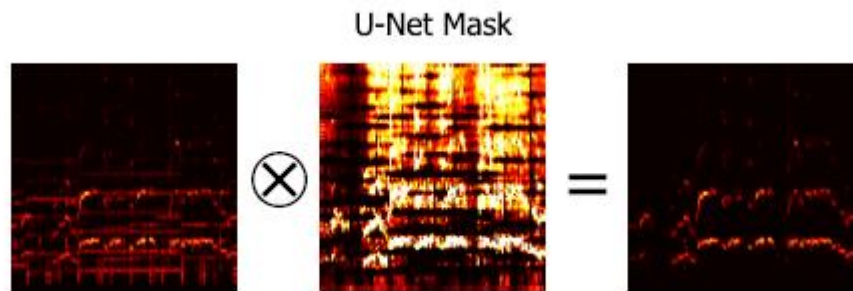
- Classify pixels into 11 classes
  - Sky, Building, Pole, Road, Pavement, Tree, SignSymbol, Fence, Car, Pedestrian, Bicyclist
- CamVid dataset



Brostow, Gabriel J., Julien Fauqueur, and Roberto Cipolla. "Semantic object classes in video: A high-definition ground truth database." Pattern Recognition Letters Vol 30, Issue 2, 2009, pp 88-97.

# Example: Singing Voice Separation

- Source separation
- Based on U-Net architecture

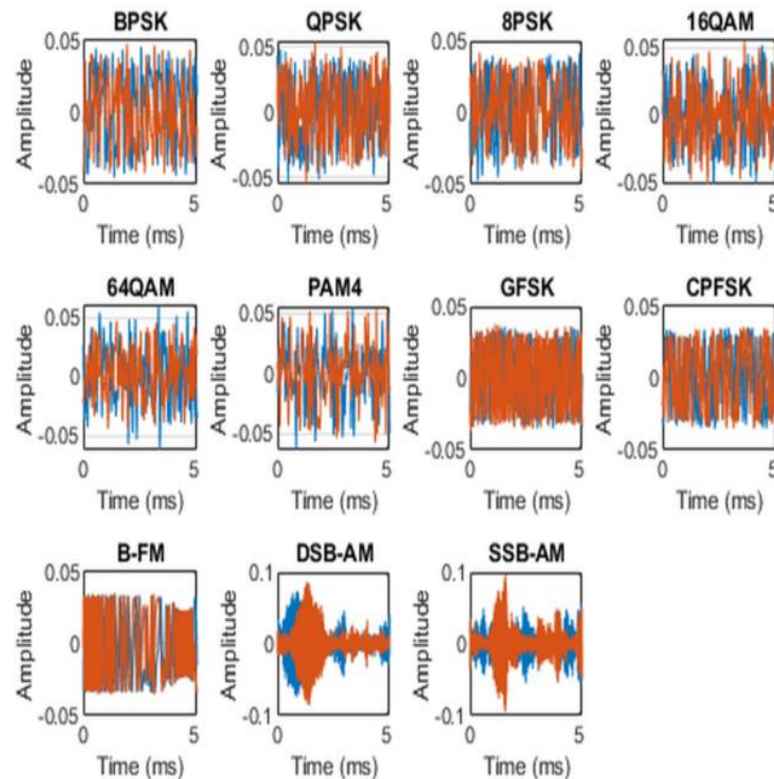




# Synthetically generating labeled data

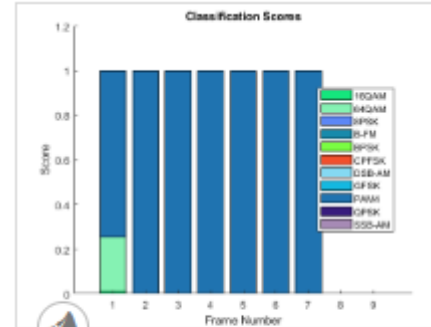
# Modulation Classification with Deep Learning

- Generate synthetic modulated signals
- Apply channel impairments
- Train a CNN to classify modulation types



Confusion Matrix for Test Data

	16QAM	64QAM	8PSK	B-FM	BPSK	CPFSK	DSB-AM	GFSK	PAM4	QPSK	SSB-AM		
16QAM	737	206	26						6	25		73.7%	26.3%
64QAM	367	611	9						2	11		61.1%	38.9%
8PSK	5	1	875		1	1			1	116		87.5%	12.5%
B-FM				999					1			99.9%	0.1%
BPSK					997	1			1	1		99.7%	0.3%
CPFSK					1	999						99.9%	0.1%
DSB-AM							941				59	94.1%	5.9%
GFSK								1000				100.0%	
PAM4	3	3			2				991	1		99.1%	0.9%
QPSK	8		193			1				798		79.8%	20.2%
SSB-AM							61				939	93.9%	6.1%
	16QAM	64QAM	8PSK	B-FM	BPSK	CPFSK	DSB-AM	GFSK	PAM4	QPSK	SSB-AM		



## Modulation Classification with Deep Learning

In this example, you generate synthetic, channel-impaired waveforms. Using the generated waveforms as training data, you

R2019a

# Deep Learning Challenges

## **Data**

- ✓ Handling large amounts of data
- ✓ Labeling thousands of signals, images & videos
- ✓ Transforming, generating, and augmenting data (for different domains)

## **Training and Testing Deep Neural Networks**

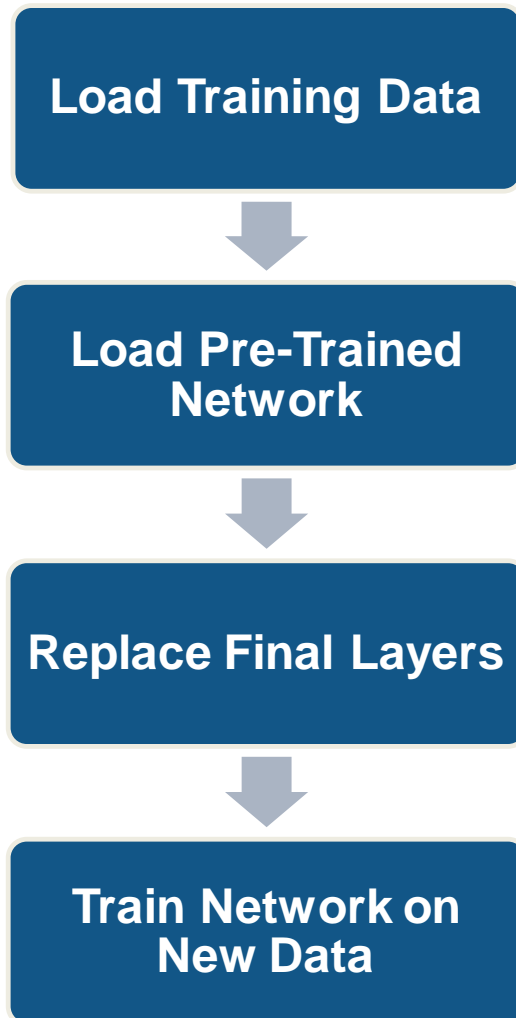
- ✓ Understanding network behavior
  - Accessing reference models from research
  - Optimizing hyperparameters
  - Training takes hours-days

## **Rapid and Optimized Deployment**

- Desktop, web, cloud, and embedded hardware

# Transfer Learning

## 8 lines of MATLAB Code



```
%% Create a datastore
1 imds = imageDatastore('Data',...
    'IncludeSubfolders',true,'LabelSource','foldernames');
2 num_classes = numel(unique(imds.Labels));
%% Load Referece Network
3 net = alexnet;
4 layers = net.Layers
%% Replace Final Layers
5 layers(end-2) =
    fullyConnectedLayer(num_classes,'Name',['fc5']);
6 layers(end) = classificationLayer('Name','classOut');
%% Set Training Options & Train Network
7 trainOpts = trainingOptions('sgdm',...
    'MiniBatchSize', 64);
8 net = trainNetwork(imds, layers, trainOpts);
```

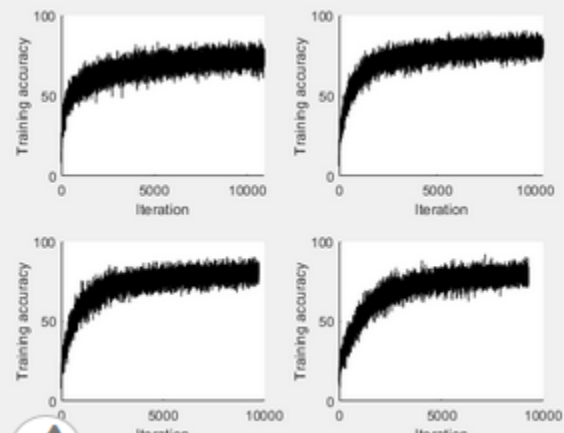
# Tune Hyperparameters to Improve Training

## Many hyperparameters


- depth, layers, solver options, learning rates, regularization, ...

## Techniques

- Parameter sweep
- Bayesian optimization




The figure displays four subplots arranged in a 2x2 grid. Each subplot shows 'Training accuracy' on the y-axis (ranging from 0 to 100) against 'Iteration' on the x-axis (ranging from 0 to 10,000). The plots show a rapid increase in accuracy from approximately 50% to 80% within the first 5,000 iterations, followed by a plateau. The top-left plot shows a noisy plateau around 80%. The top-right plot shows a smoother plateau around 85%. The bottom-left plot shows a noisy plateau around 80%. The bottom-right plot shows a smoother plateau around 85%.


 **Use parfeval to Train Multiple Deep Learning Networks**

Use parfeval for a parameter sweep on the depth of the network architecture. Deep Learning training often takes hours or days, and

[Open Script](#)



The figure shows a 3x3 grid of small images, each with a label and an accuracy percentage above it. The labels and accuracies are: cat, 86.4%; ship, 99.9%; airplane, 83.1%; frog, 99.7%; cat, 89.5%; ship, 100%; airplane, 49.6%; ship, 100%; frog, 99.8%.

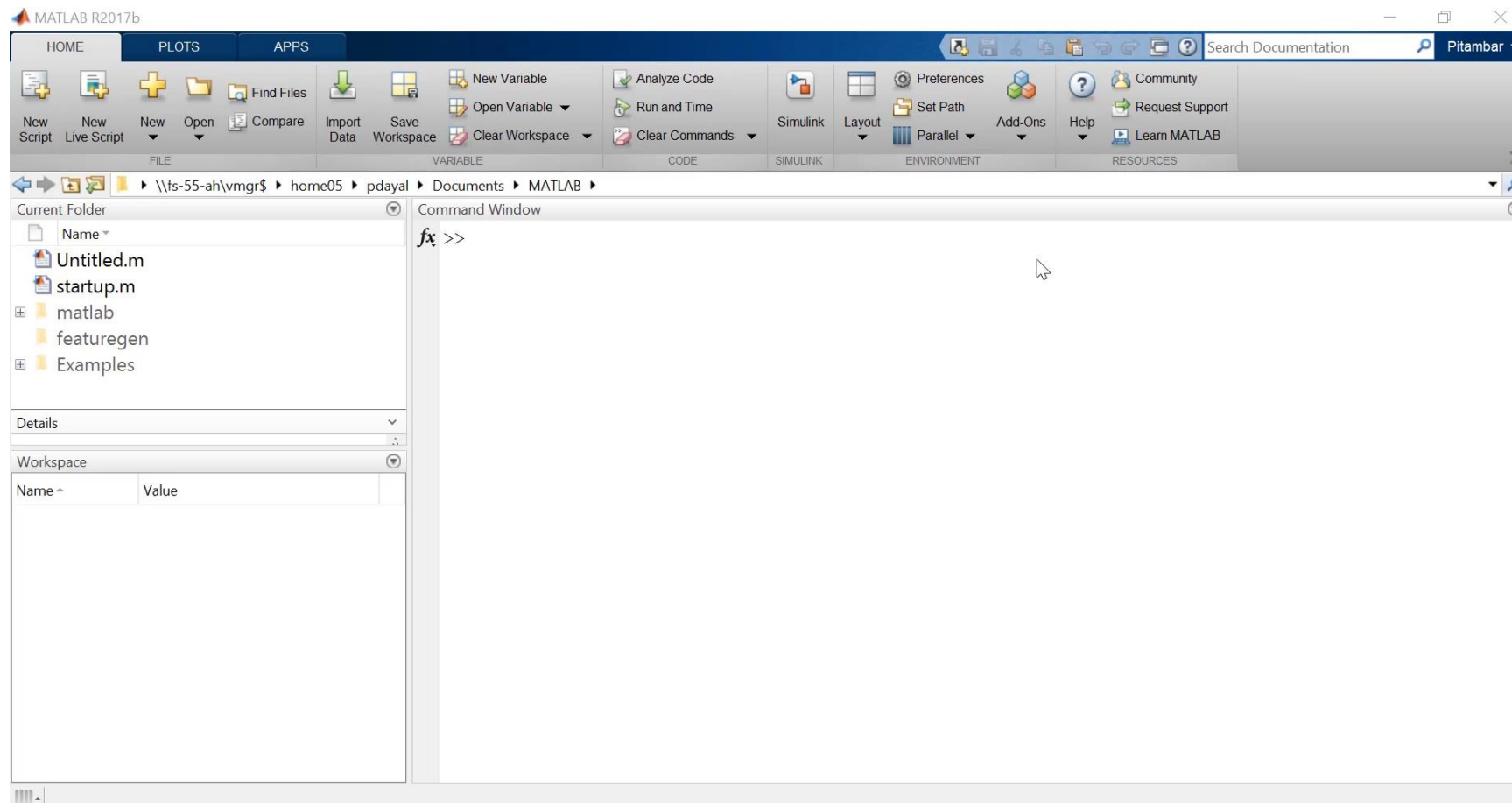
 **Deep Learning Using Bayesian Optimization**

Apply Bayesian optimization to deep learning and find optimal network parameters and training options for convolutional neural networks.

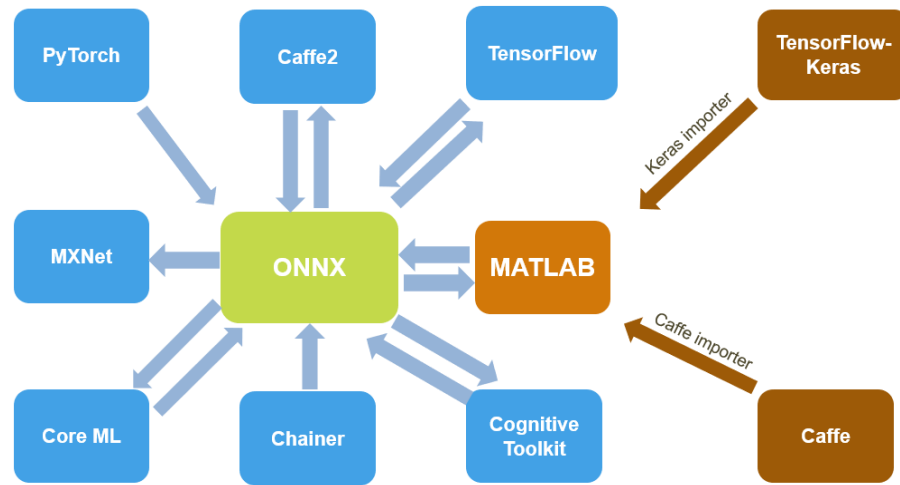
[Open Live Script](#)



# Keras-Tensorflow Importer

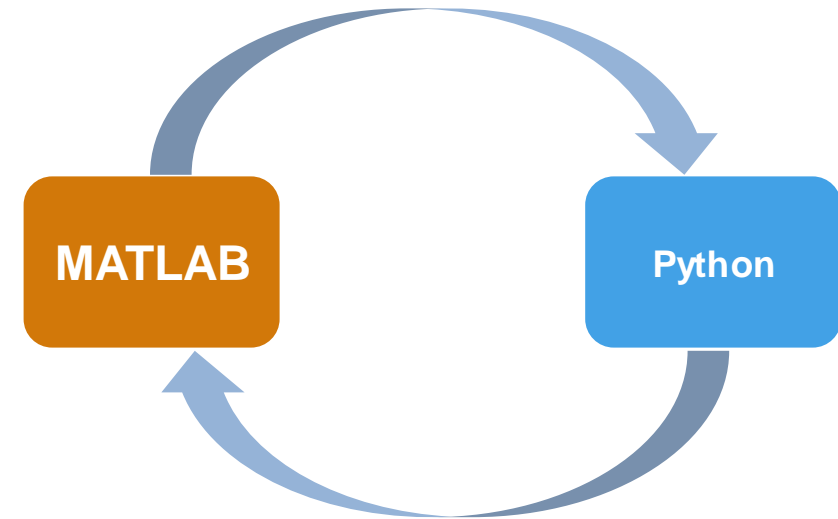


# Model Exchange and Co-execution



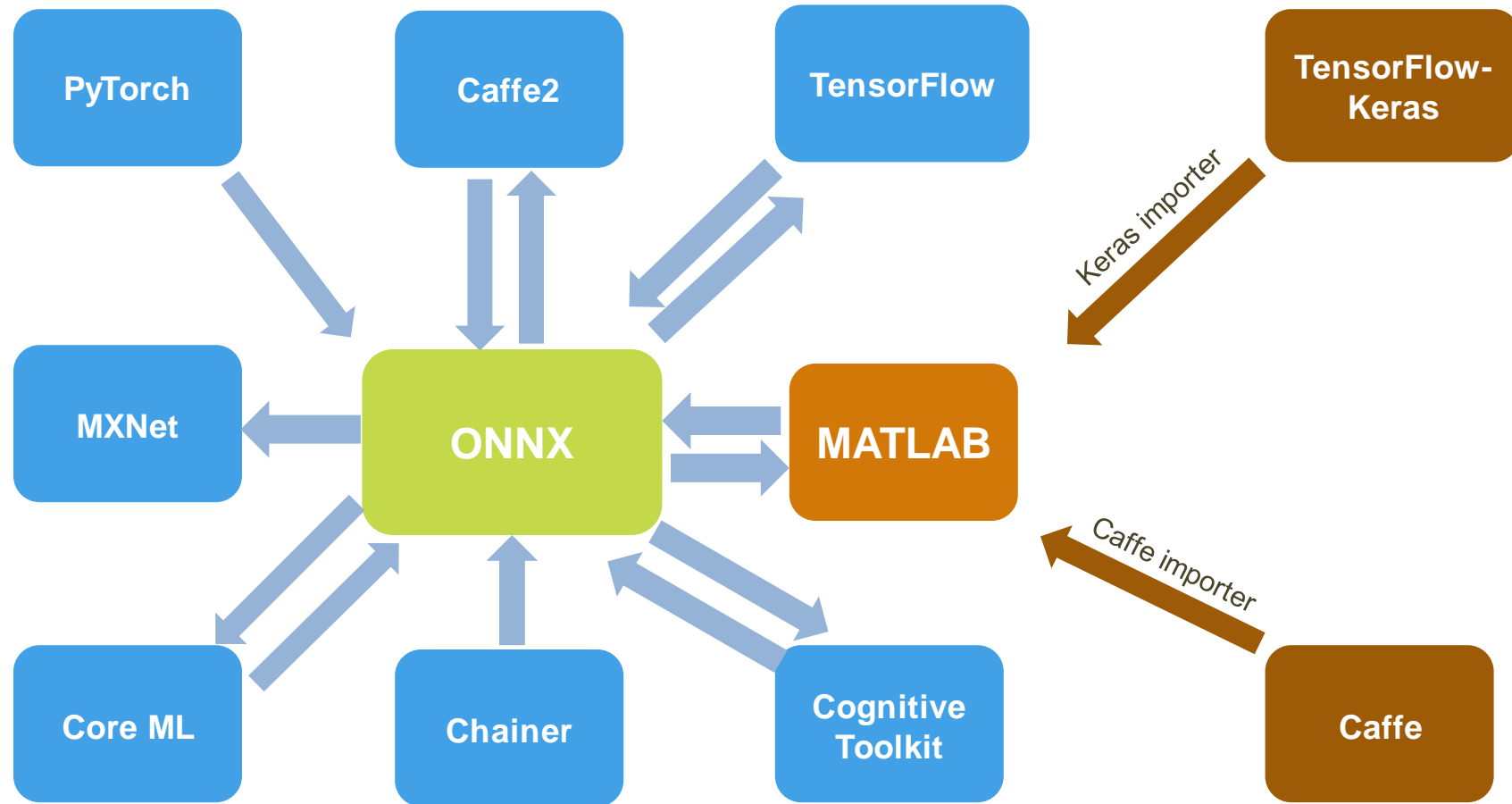
ONNX = Open Neural Network Exchange Format

Model Exchange



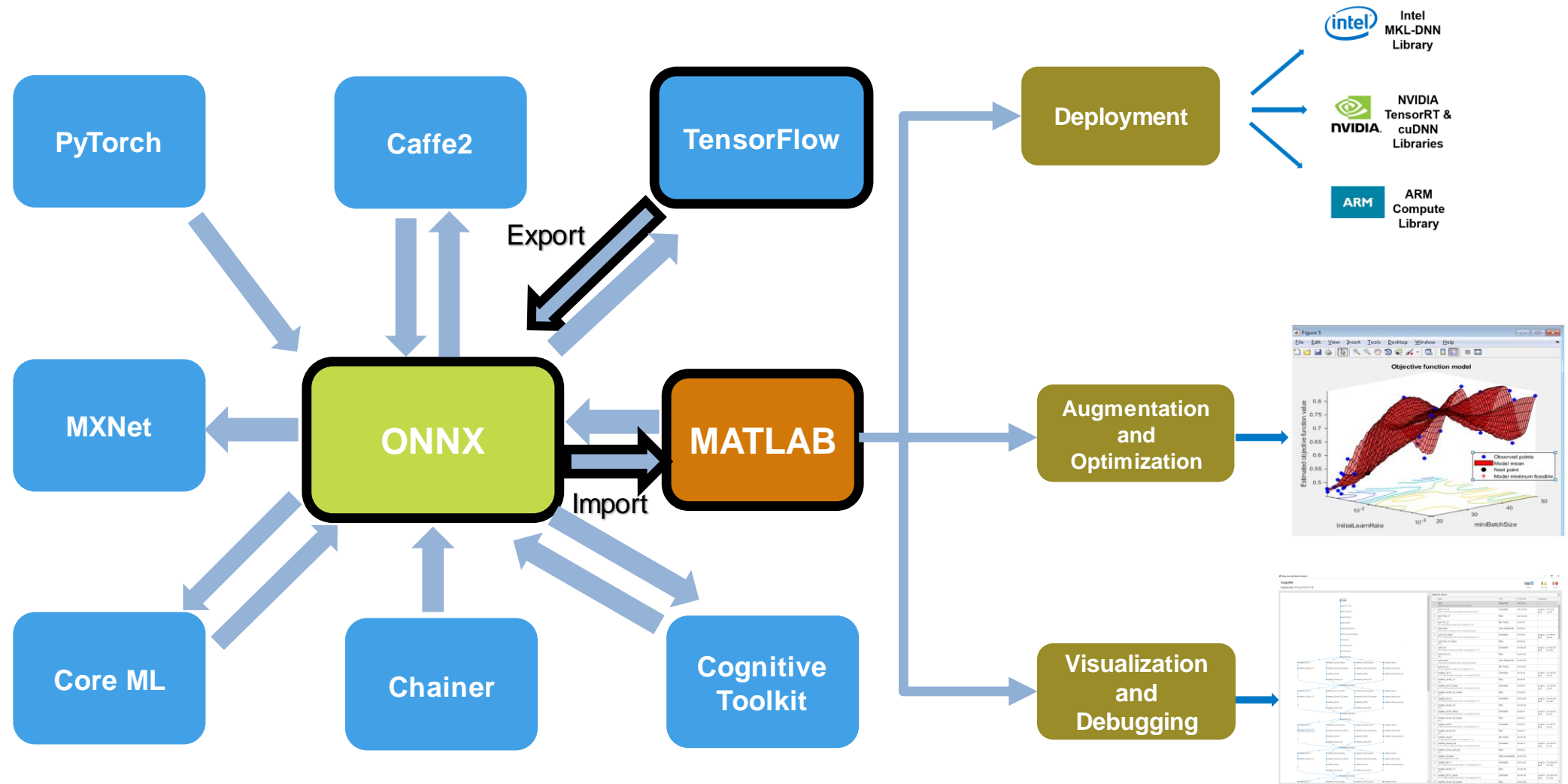
Co-execution

# Model Exchange With Deep Learning Frameworks



ONNX = Open Neural Network Exchange Format

# Interoperate With Deep Learning Frameworks – Use Cases



**ONNX = Open Neural Network Exchange Format**

# Model Exchange With Deep Learning Frameworks

## Caffe Model Importer

- `importCaffeLayers`
- `importCaffeNetwork`

## TensorFlow-Keras Model Importer

- `importKerasLayers`
- `importKerasNetwork`

## ONNX Converter

- `importONNXNetwork`
- `exportONNXNetwork`

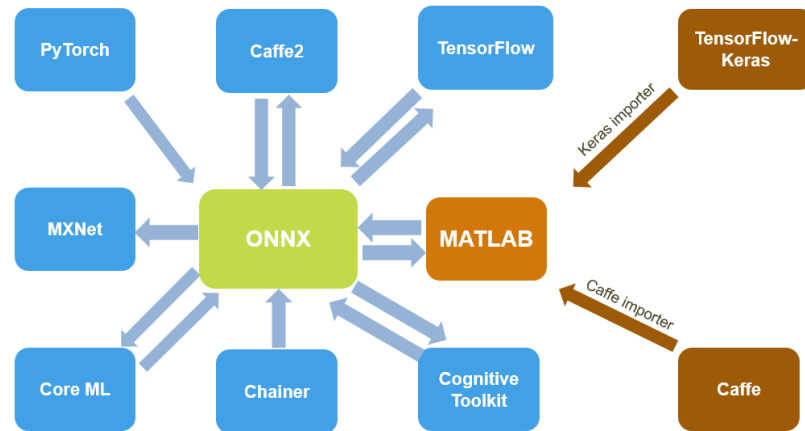
Caffe  
MODELS

KERAS IMPORTER

Importer for TensorFlow-Keras Models

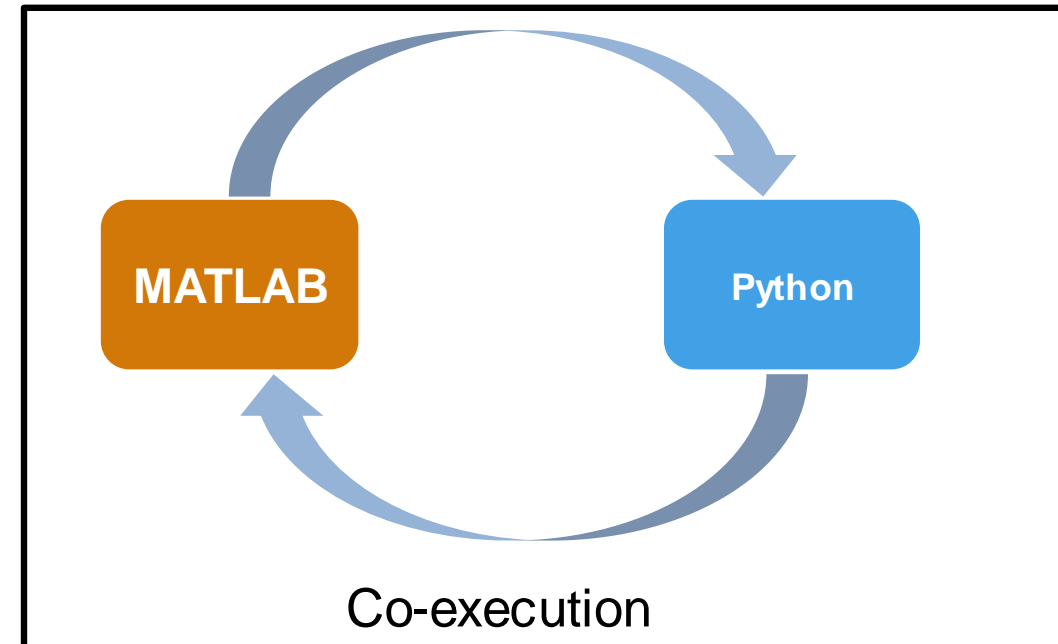
ONNX Converter

# Model Exchange and Co-execution



ONNX = Open Neural Network Exchange Format

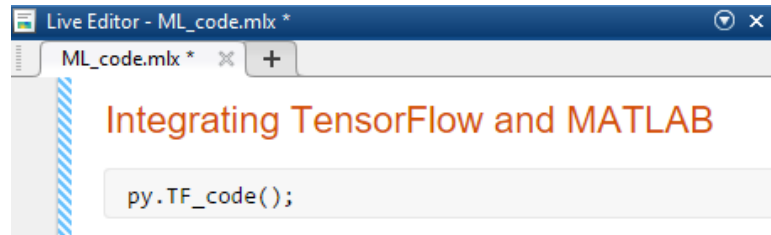
Model Exchange





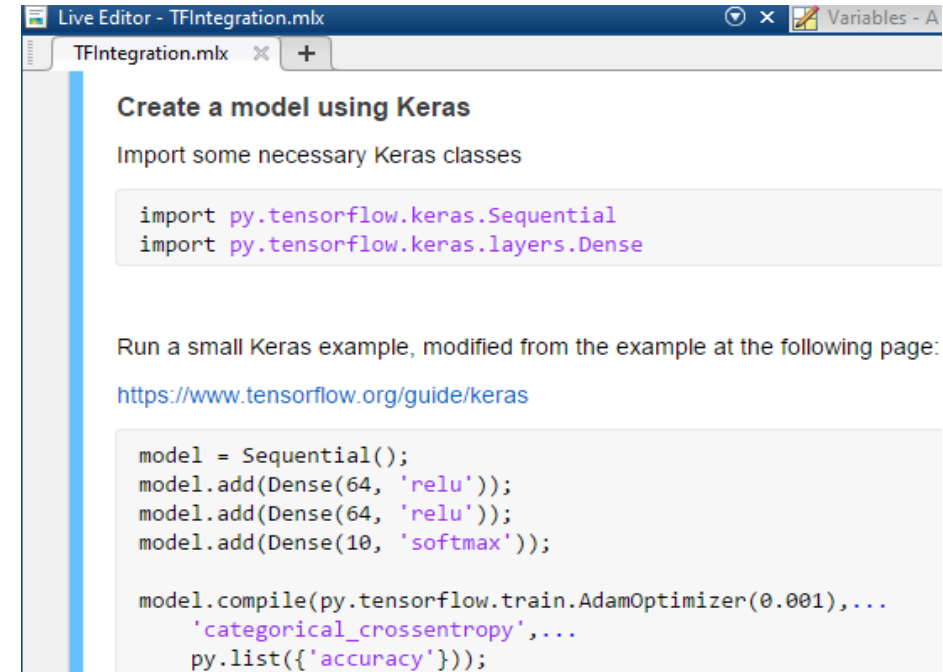
# MATLAB-Python Co-Execution – The ‘How’

## Call Python file from MATLAB

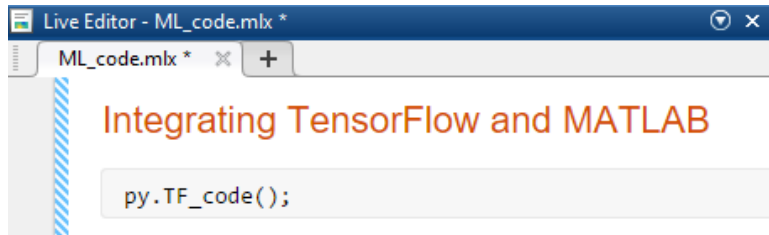


TF\_code.py

## Call TensorFlow commands from MATLAB



# MATLAB-Python Co-Execution – Method A



TF\_code.py

1. Copy the code into a .PY file
2. Wrap entry point in a function

```
import tensorflow as tf
from tf import keras
```

```
def myTFCode():
    for x in y:
        a.foo()
```

```
def foo(x):
    return x + 1
```

3. Add module to Python path and then call from MATLAB via:

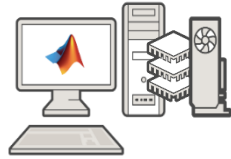
```
py.myModule.myTFCode();
```

# Deep Learning on CPU, GPU, Multi-GPU and Clusters

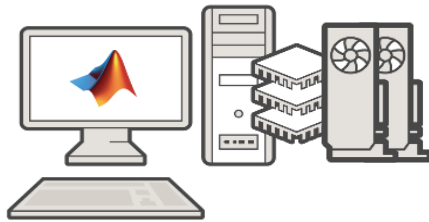
## HOW TO TARGET?



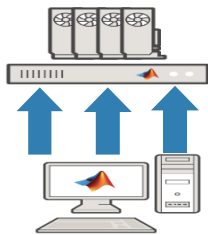
Single  
CPU



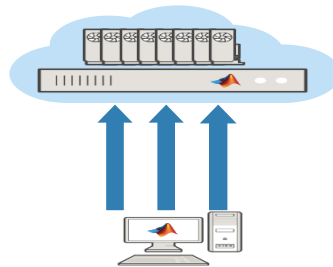
Single CPU  
Single GPU



Single CPU, Multiple GPUs



On-prem server with  
GPUs



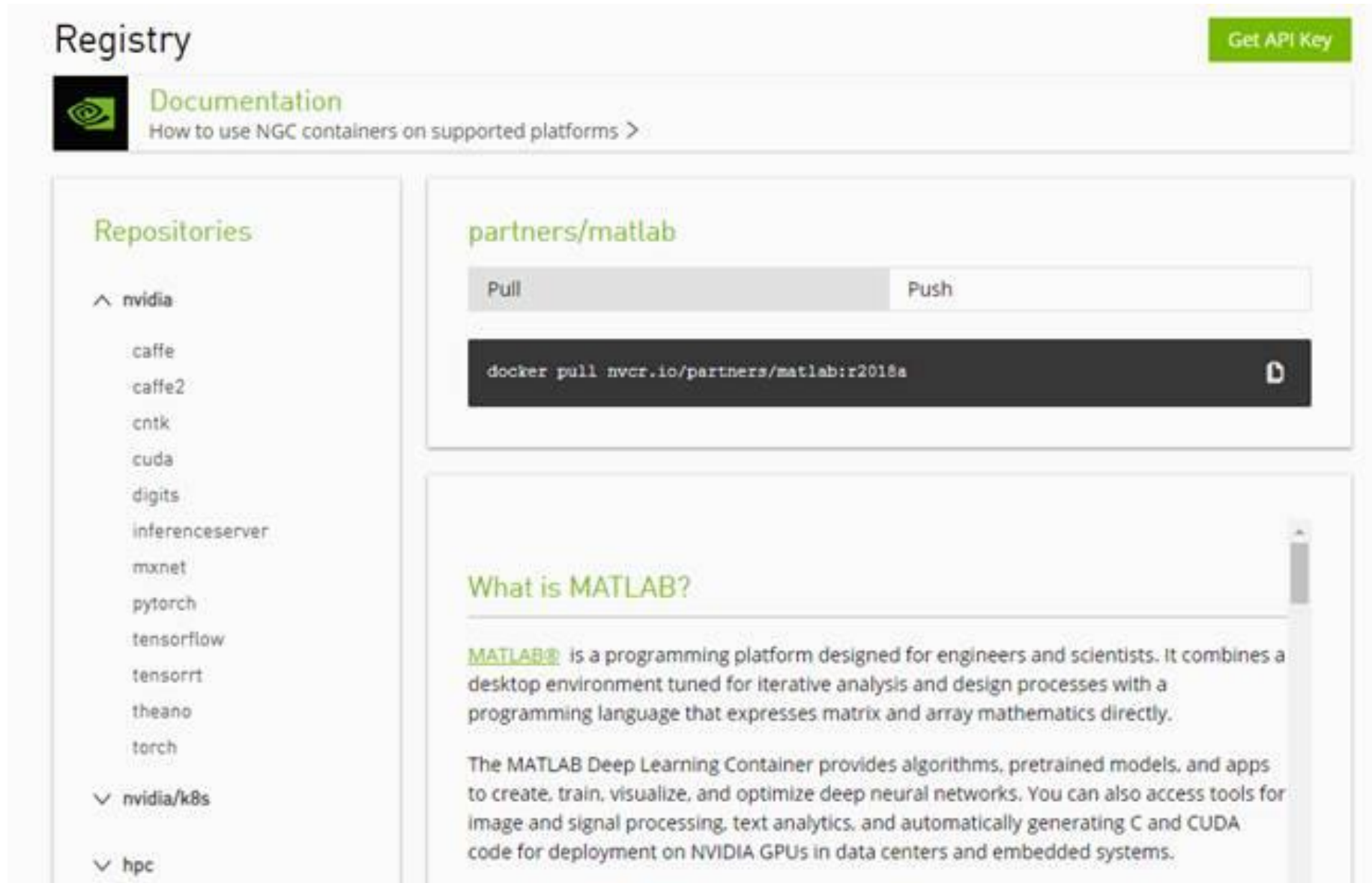
Cloud GPUs  
(AWS)

```
opts = trainingOptions('sgdm', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 250, ...
    'InitialLearnRate', 0.00005, ...
    'ExecutionEnvironment', 'auto' );
```

```
opts = trainingOptions('sgdm', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 250, ...
    'InitialLearnRate', 0.00005, ...
    'ExecutionEnvironment', 'multi-gpu' );
```


```
opts = trainingOptions('sgdm', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 250, ...
    'InitialLearnRate', 0.00005, ...
    'ExecutionEnvironment', 'parallel' );
```

# MATLAB Containers for NVIDIA GPU Cloud & DGX



The screenshot shows the NVIDIA NGC Registry interface. At the top, there's a 'Registry' header with a 'Get API Key' button. Below it is a 'Documentation' section with a link to 'How to use NGC containers on supported platforms'. The main content area is divided into two columns. The left column, titled 'Repositories', lists various NVIDIA containers under the 'nvidia' namespace, including 'caffe', 'caffe2', 'cntk', 'cuda', 'digits', 'inferenceserver', 'mxnet', 'pytorch', 'tensorflow', 'tensorrt', 'theano', and 'torch'. The right column displays the 'partners/matlab' repository. It features a 'Pull' button and a 'Push' button. Below these buttons is a terminal window showing the command: `docker pull nvcr.io/partners/matlab:r2018a`. At the bottom of the right column, there is a section titled 'What is MATLAB?' which describes MATLAB as a programming platform for engineers and scientists, combining a desktop environment with a programming language for matrix and array mathematics. It also mentions the MATLAB Deep Learning Container and its capabilities for creating, training, visualizing, and optimizing deep neural networks.

Registry [Get API Key](#)

 **Documentation**  
How to use NGC containers on supported platforms >

**Repositories**

- ^ nvidia
  - caffe
  - caffe2
  - cntk
  - cuda
  - digits
  - inferenceserver
  - mxnet
  - pytorch
  - tensorflow
  - tensorrt
  - theano
  - torch
- ▼ nvidia/k8s
- ▼ hpc

**partners/matlab**

Pull Push

```
docker pull nvcr.io/partners/matlab:r2018a
```

**What is MATLAB?**

**MATLAB®** is a programming platform designed for engineers and scientists. It combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly.

The MATLAB Deep Learning Container provides algorithms, pretrained models, and apps to create, train, visualize, and optimize deep neural networks. You can also access tools for image and signal processing, text analytics, and automatically generating C and CUDA code for deployment on NVIDIA GPUs in data centers and embedded systems.

# Deep Learning Challenges

## **Data**

- ✓ Handling large amounts of data
- ✓ Labeling thousands of signals, images & videos
- ✓ Transforming, generating, and augmenting data (for different domains)

## **Training and Testing Deep Neural Networks**

- ✓ Accessing reference models from research
- ✓ Understanding network behaviour
- ✓ Optimizing hyperparameters
- ✓ Training takes hours-days

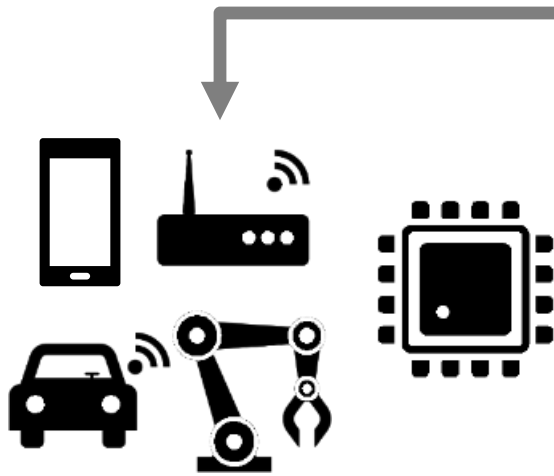
## **Rapid and Optimized Deployment**

- Desktop, web, cloud, and embedded hardware



# Deploying Deep Learning Application

Embedded Hardware

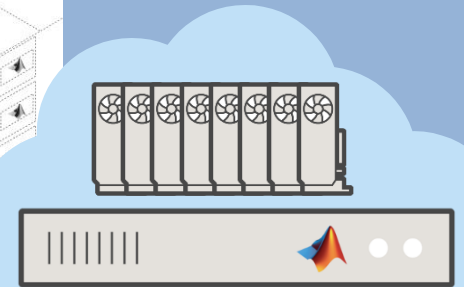
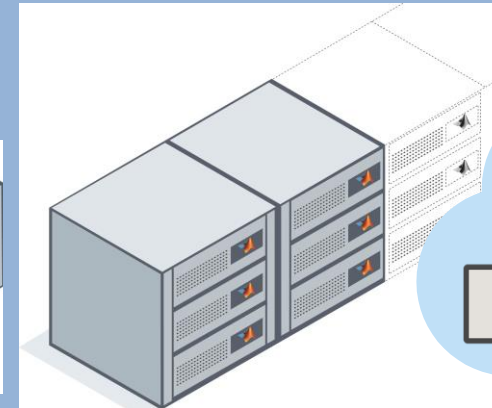
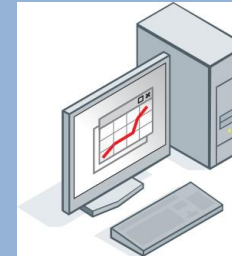


Application  
logic



Code  
Generation

Desktop, Web, Cloud



Application  
Deployment

Standalone  
Application

Excel  
Add-in

C/C++

Java

++

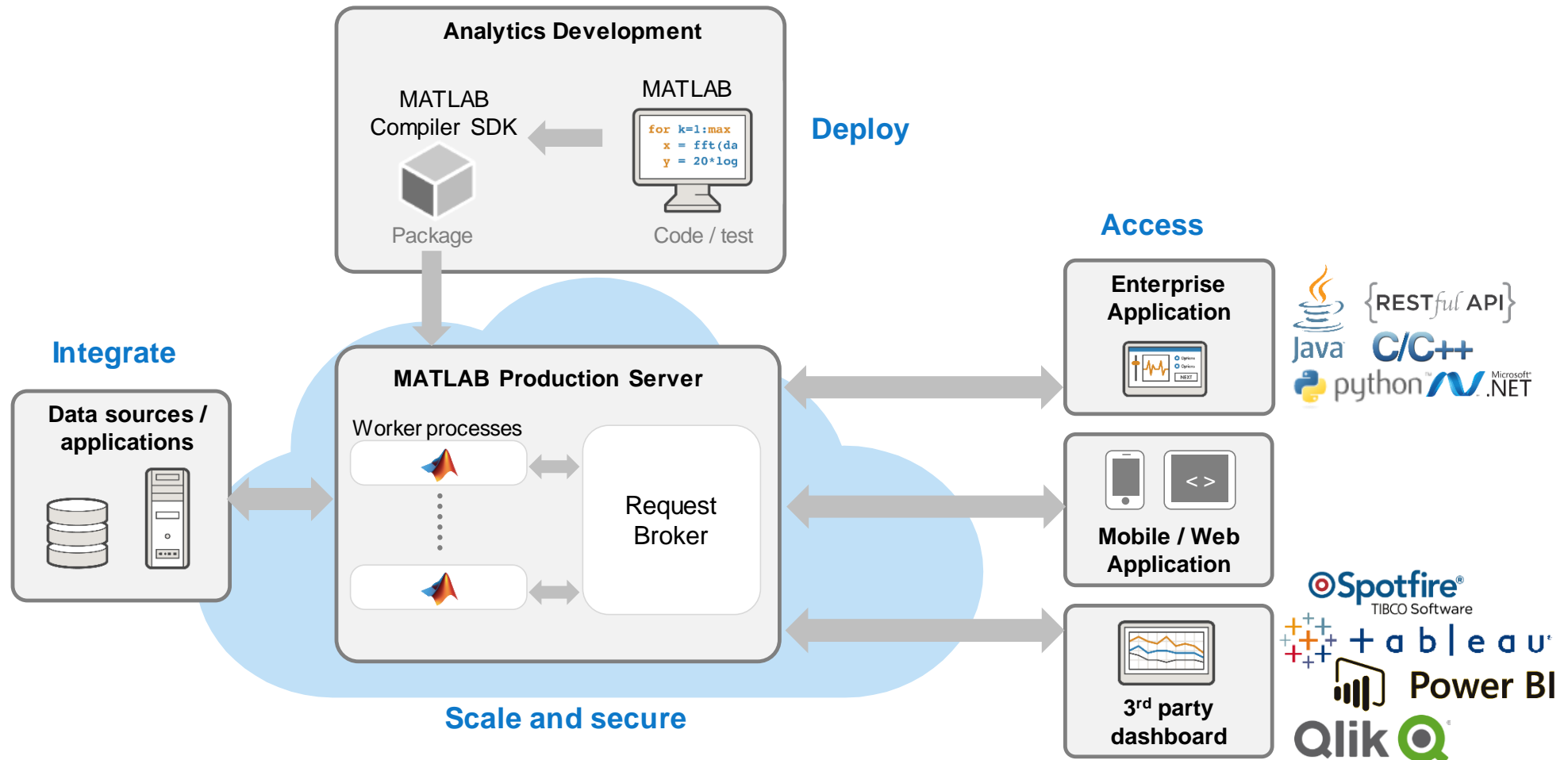
MATLAB  
Production  
Server

Hadoop  
Spark

Python

.NET

# MATLAB Production Server is an application server that publishes MATLAB code as APIs that can be called by other applications



# MATLAB support for Cloud

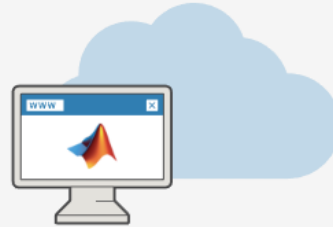
## Use MATLAB in the Cloud

Run in different cloud environments from MathWorks Cloud to public clouds including AWS, Azure, and others

### MathWorks Cloud

MathWorks Cloud provides you with instant access to MATLAB and other products and services you are licensed for hosted on MathWorks managed cloud infrastructure. With [MATLAB Online™](#), you can use MATLAB in a web browser without installing, configuring, or managing any software. MathWorks Cloud also provides [MATLAB Drive™](#), giving you the ability to store, access, and work with your files from anywhere. You can access MathWorks Cloud solutions anywhere across different devices, use them to teach and learn, and to incorporate MATLAB analytics for a variety of applications.

[Learn more](#) about hosted offerings.



### Public Clouds

Use MATLAB on virtual machines in public cloud environments like Amazon Web Services (AWS) and Microsoft Azure. These vendors provide access to on-demand computing resources. They also offer wide-ranging, prebuilt services for data storage, data streaming, elastic scaling, load balancing, security, and more.


If you are not a cloud expert, or if you want a head start, use a MathWorks published [reference architecture](#). Templates in these reference architectures automatically create and configure the cloud infrastructure for running MATLAB. You can also adapt or extend the reference architectures to better meet your specific needs.

Learn more about running MATLAB and other products on:

[AWS](#)
[Azure](#)
[Other Clouds](#)


[Features](#)
[Business](#)
[Explore](#)
[Marketplace](#)
[Pricing](#)

[Sign in](#)
[Sign up](#)




## MathWorks Reference Architectures

Reference Architectures

<http://www.mathworks.com/cloud>

[Repositories 6](#)
[People 1](#)
[Projects 0](#)



### Grow your team on GitHub

GitHub is home to over 28 million developers working together. Join them to grow your own development teams, manage permissions, and collaborate on projects.

[Sign up](#)

Type: All
Language: All

#### mps-on-azure

Stand up a MATLAB Production Server using Azure Deployment

PowerShell
3
Updated 11 days ago

#### mps-on-aws

Stand up a MATLAB Production Server using CloudFormation

6
Updated 13 days ago

#### matlab-on-aws

Stand up a MATLAB desktop with Remote Desktop access using AWS CloudFormation

4
Updated 13 days ago

#### mdcs-on-aws

Stand up a MATLAB Distributed Computing Server cluster using CloudFormation

Shell
4
1
Updated 14 days ago

#### mdcs-on-azure


Stand up a MATLAB Distributed Computing Server cluster using Azure Deployment

PowerShell
2
1
Updated 14 days ago

#### Top languages

- PowerShell
- Shell



#### People


[jfluet](#)
John Fluet

## Deployment Steps

### Step 1. Launch the Template

Click the **Deploy to Azure** button to deploy resources on Azure. This will open the Azure Portal in your web browser.

Windows Server 2016 VM	Ubuntu 16.04 VM
 MATLAB Release: R2018a	 MATLAB Release: R2018a

**Note:** Creating resources on Azure can take at least 30 minutes.

# IAAS to PAAS

Microsoft Azure

Search resources, services

Home > Custom deployment

Custom deployment

Deploy from a custom template

TEMPLATE

Customized template

10 resources

Edit template

Edit parameters

Learn more

BASICS

Subscription

AEG - Pallavi Kar, Prashant Rao, Amit Doshi

Resource group

Create new

Use existing

Create a resource group

Location

West US

SETTINGS

Server VM Instance Size

Standard\_D4s\_v3

Instance Count

2

Admin Username

Admin Password

Allow connections from

TERMS AND CONDITIONS

Azure Marketplace Terms

Azure Marketplace

Purchase

```

graph LR
    Client[MATLAB Production Server Client] -- https --> VNet[Virtual Network]
    subgraph VNet [Virtual Network]
        VM[Server Machine VM  
• Cloud Console  
• Flex License Manager]
        ASG[Worker Auto Scaling Group]
    end
    VM -- https public IP --> IP[https public IP]
    VM --> SA[Storage Account]
    ASG --> SA
  
```

90



# Automatic Cluster Creation

WorkshopMPS  
Resource group

Search (Ctrl+/) <<

+ Add Edit columns Delete resource group Refresh Move Assign tags Delete

Subscription (change)  
AEG - Pallavi Kar, Prashant Rao, Amit Doshi  
Subscription ID  
a2c5822b-afab-4a1d-896d-c5302aba11e2  
Deployments  
2 Succeeded  
Tags (change)  
[Click here to add tags](#)

Filter by name... All types All locations No group...

10 items ☐ Show hidden types ⓘ

<input type="checkbox"/>	NAME ↑↓	TYPE ↑↓	LOCATION ↑↓	SUBSCRIPTION ↑↓	SUBSCRIPTION ID ↑↓
<input type="checkbox"/>	serverlogqscakziq15xci	Storage account	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	servermachine	Virtual machine	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	servermachine_OsDisk_1_c542f186579a438691bed34eca6e0a35	Disk	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	servermachine-nic	Network interface	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	servermachine-public-ip	Public IP address	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	vmss1qsca	Virtual machine scale set	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	vmss1qsca-agw	Application gateway	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	vmss1qsca-pip	Public IP address	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	vmss1qsca-rdp-nsg	Network security group	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***
<input type="checkbox"/>	vmss1qsca-vnet	Virtual network	West US	AEG - Pallavi Kar, Prashant Rao, Ami...	a2c5822b-afab-4a1d-896d-c5302aba... ***

Overview  
Activity log  
Access control (IAM)  
Tags  
Events

SETTINGS  
Quickstart  
Resource costs  
Deployments  
Policies  
Properties  
Locks  
Automation script

MONITORING  
Alerts

# Get Public IP for access

api connection

Home > AzureIoTWorkshop > servermachine-public-ip > AzureIoTWorkshop > servermachine

**servermachine**  
Virtual machine

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problem...

Settings

Networking

Disks

Size

Security

Extensions

Connect Start Restart Stop Capture Delete Refresh

Advisor (1 of 5): Follow Security Center recommendations →

Resource group (change)  
[AzureIoTWorkshop](#)

Status  
Running

Location  
East US

Subscription (change)  
[AppDeploy-PFT / EI-DTST](#)

Subscription ID  
063d5d18-9fa4-4908-ab19-5ea8c33ace74

Computer name  
servermachine

Operating system  
Windows

Size  
Standard D1 (1 vcpu, 3.5 GB memory)

Public IP address  
[137.117.102.181](#) Click to copy

Virtual network/subnet  
[vmss1gpvx-vnet/vmss1gpvxsubnet](#)

DNS name  
[Configure](#)

Tags (change)

Description : Virtual machine running the MATLAB Pr... provider : D36A3EDC-0566-4EE4-86D3-64F20D2D... owner : ae:pkar

# Server Machine VM access

- MPS console endpoint: <https://xxx.xx.xx.xx>

Connect Start Restart Stop Capture Delete Refresh

**Advisor (1 of 1):** Use availability sets for improved fault tolerance →

Resource group [change](#)  
WorkshopMPS

Status  
Running

Location  
West US

Subscription [change](#)  
AEG - Pallavi Kar, Prashant Rao, Amit Doshi

Subscription ID  
a2c5822b-afab-4a1d-896d-c5302aba11e2

Tags [change](#)  
Description : Virtual machine running the MATLAB Productio... provider : D36A3EDC-0566-4EE4-86D3-64F20D2DDA06

Computer name  
servermachine

Operating system  
Windows

Size  
Standard D1 (1 vcpu, 3.5 GB memory)

Public IP address  
[40.118.147.236](#)

Virtual network/subnet  
[vmss1qscasubnet](#)

DNS name  
[Configure](#)

Status:	Running
Number of MATLAB Production Server VMs:	2
Number of MATLAB Production Server Workers per VM:	4
Total Number of Workers:	8
HTTPS Server Endpoint: ⓘ	<a href="https://mpsqscakzlq15xci.westus.cloudapp.azure.com:9910">https://mpsqscakzlq15xci.westus.cloudapp.azure.com:9910</a>

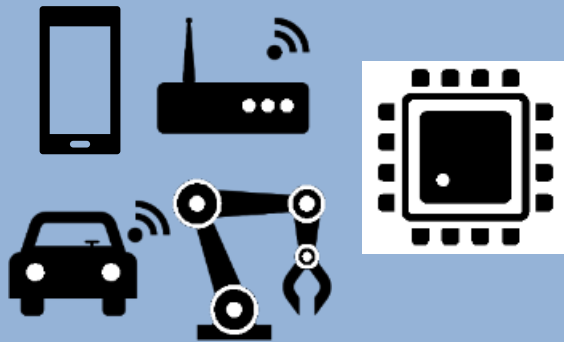
## Additional Information

To start using the server:

- Get a license from the [MathWorks License Center](#) and upload it in the [Manage License](#) section.
- Use the HTTPS server endpoint to make requests to the server.

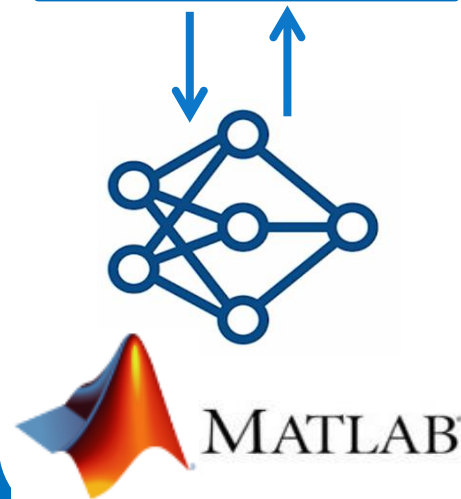
# Deploying Deep Learning Application

Embedded Hardware

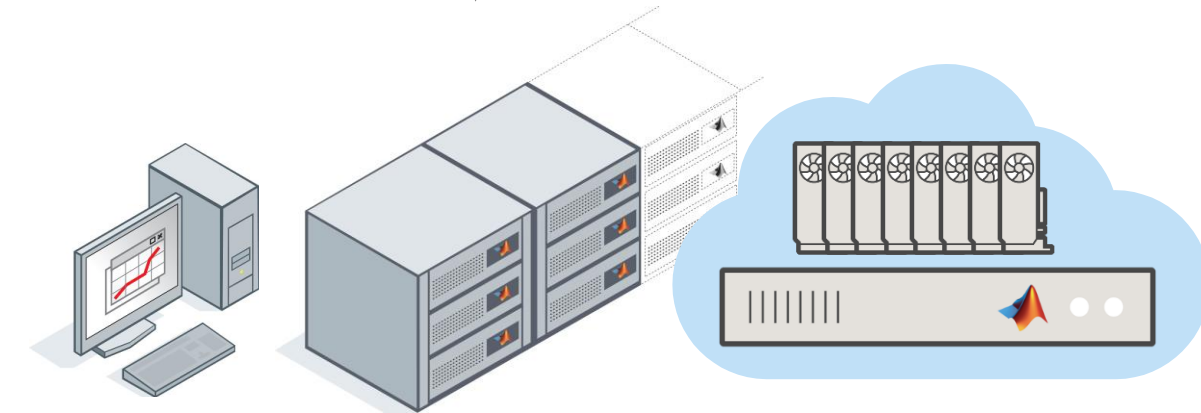


**Code  
Generation**

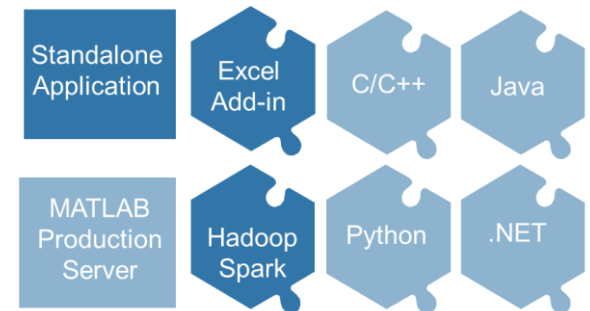
**Application  
logic**



Desktop, Web, Cloud

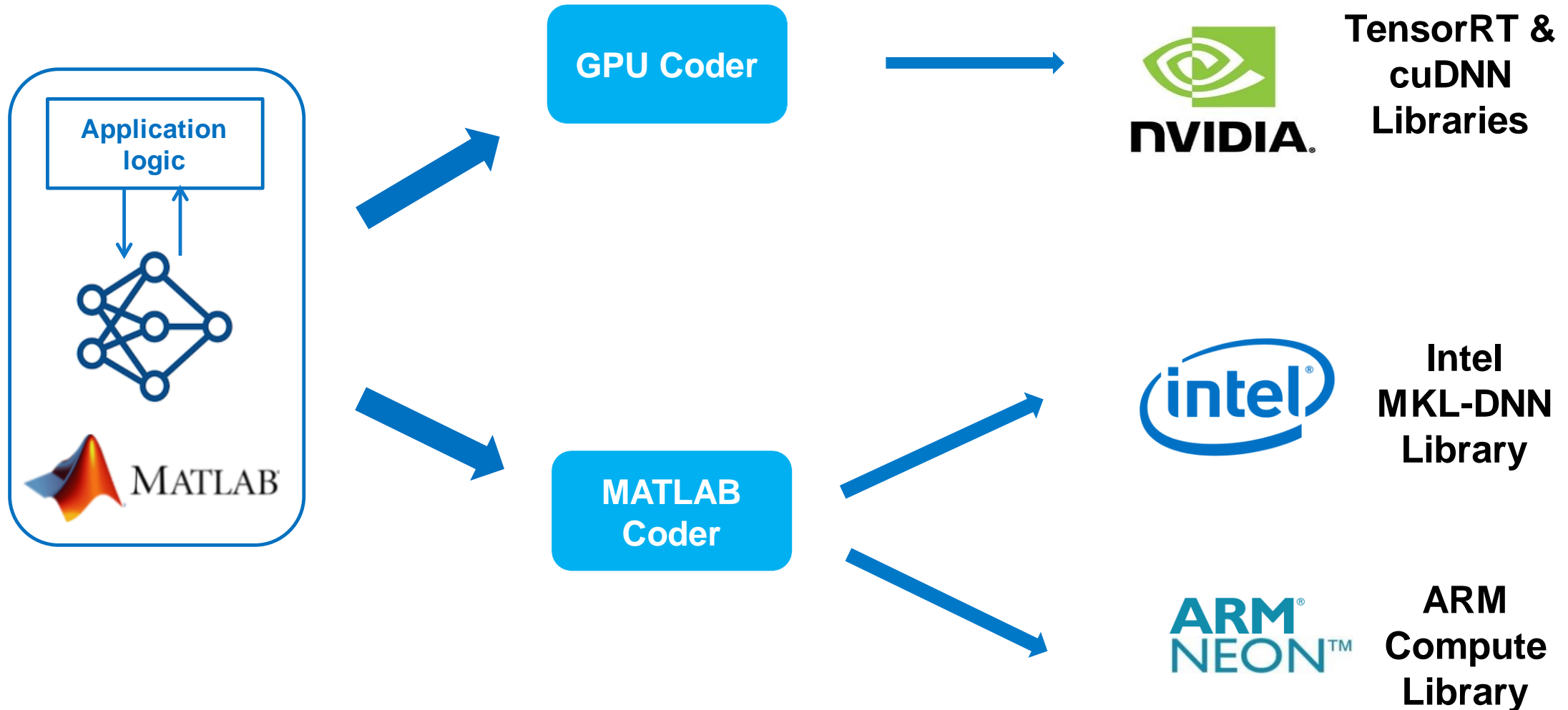


**Application  
Deployment**



# Solution- GPU/MATLAB Coder for Deep Learning Deployment

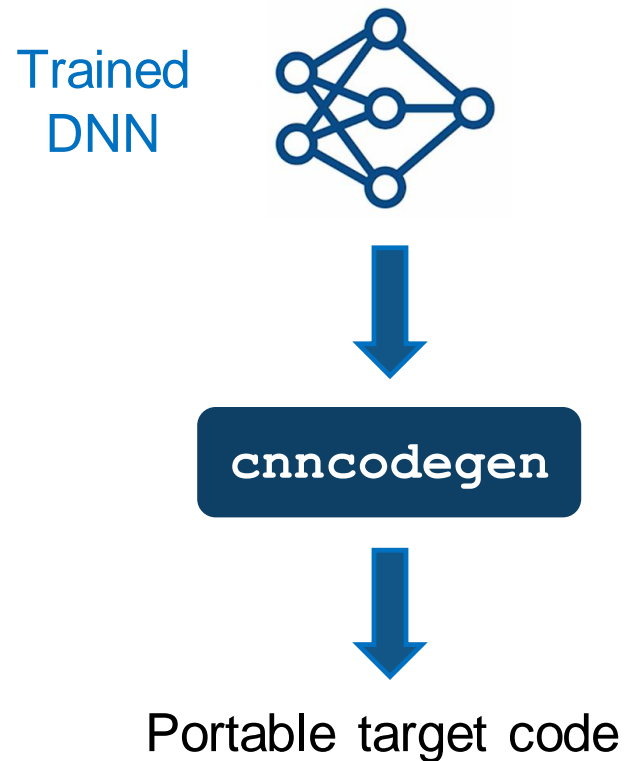
Target Libraries



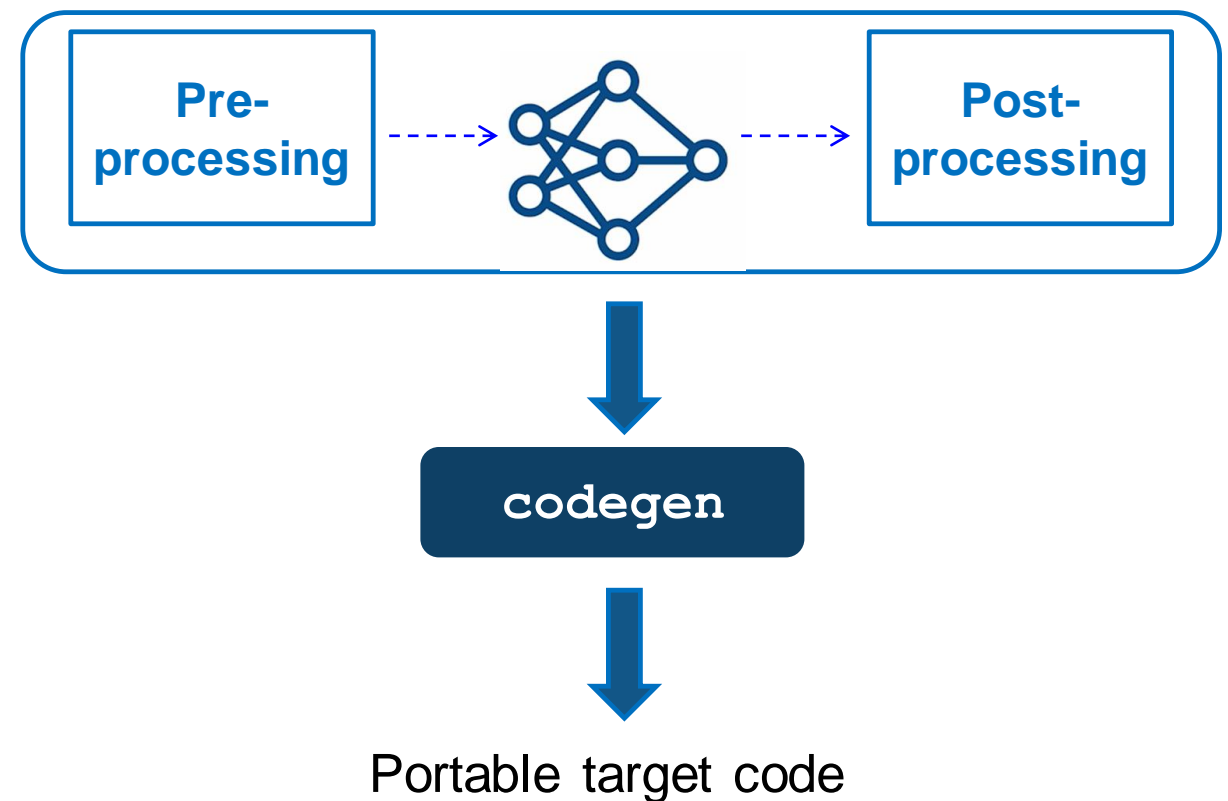


# Deep Learning Deployment Workflows

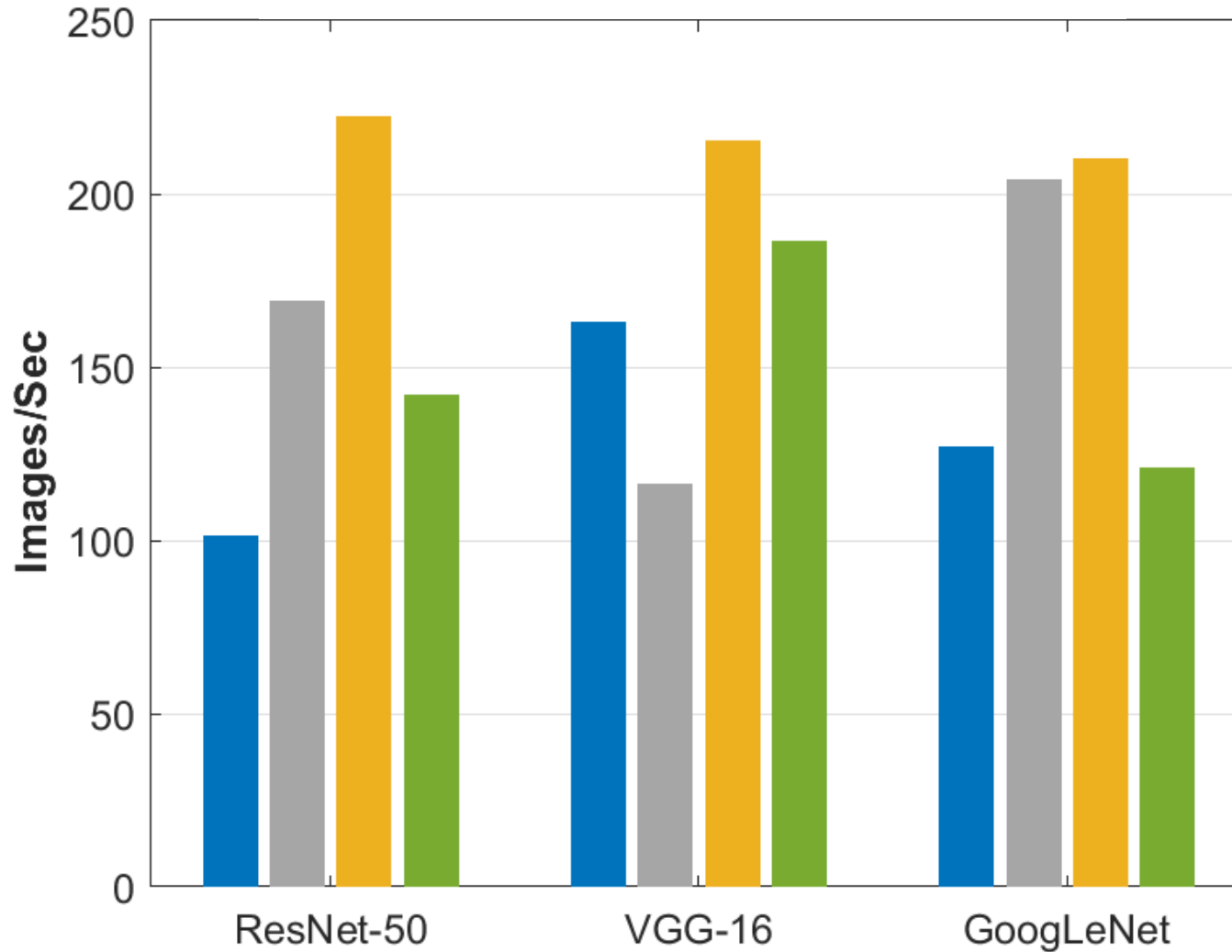
## INFERENCE ENGINE DEPLOYMENT



## INTEGRATED APPLICATION DEPLOYMENT



# Single Image Inference on Titan Xp using cuDNN



**TensorFlow** (1.8.0)

**MXNet** (1.2.1)

**GPU Coder** (R2018b)

**PyTorch** (0.3.1)

# NVIDIA Hardware Support Package (HSP)

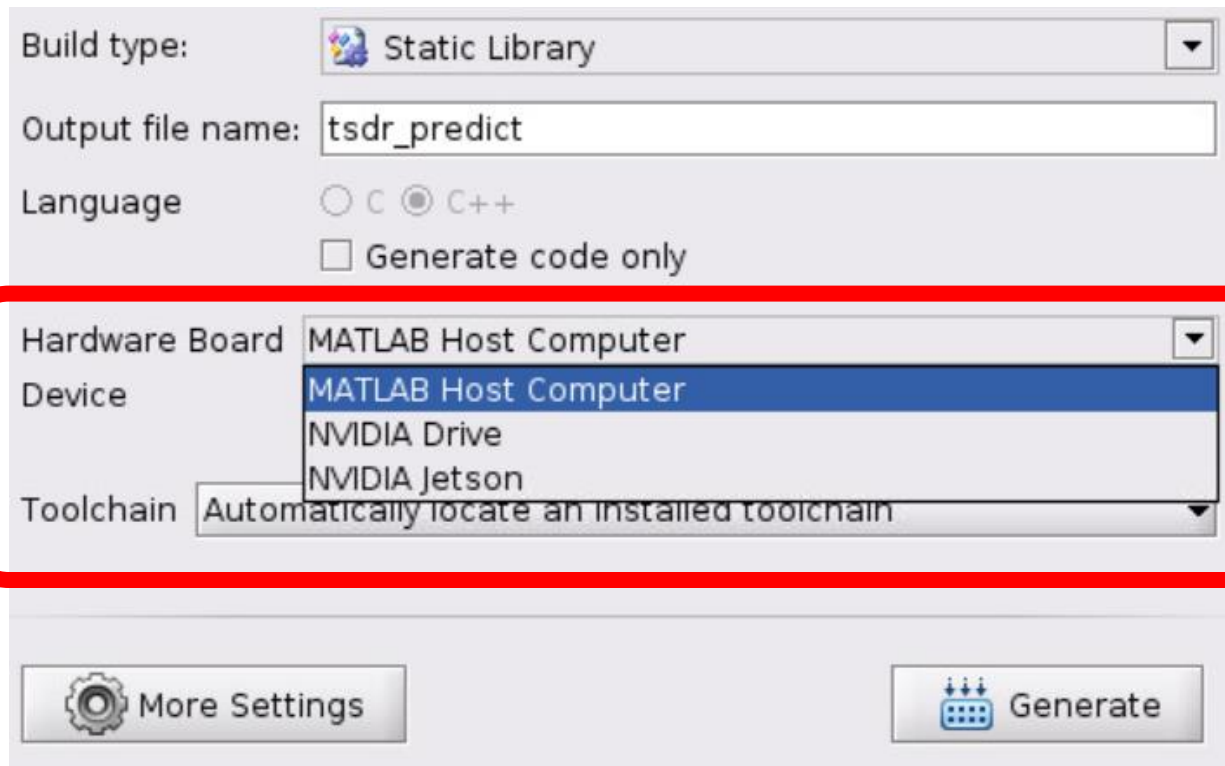
**Simple out-of-box targeting for NVIDIA boards:**



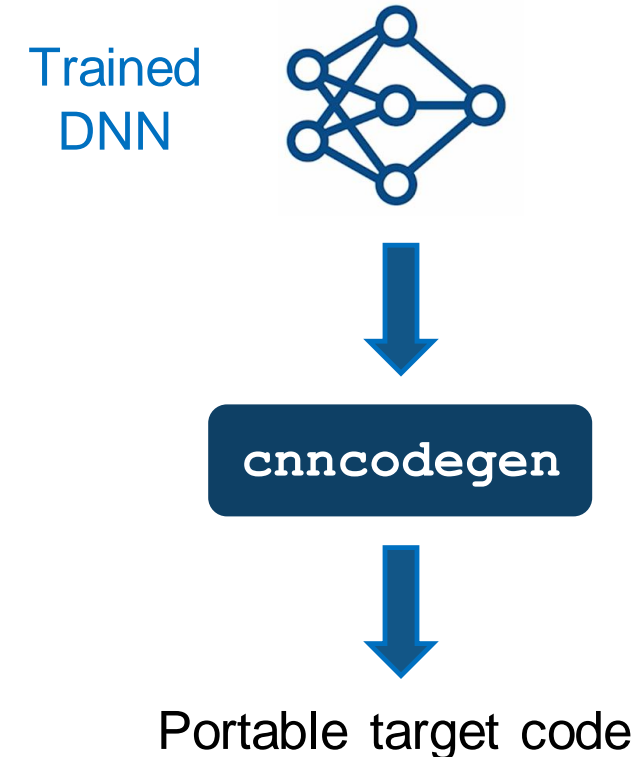
**Jetson**



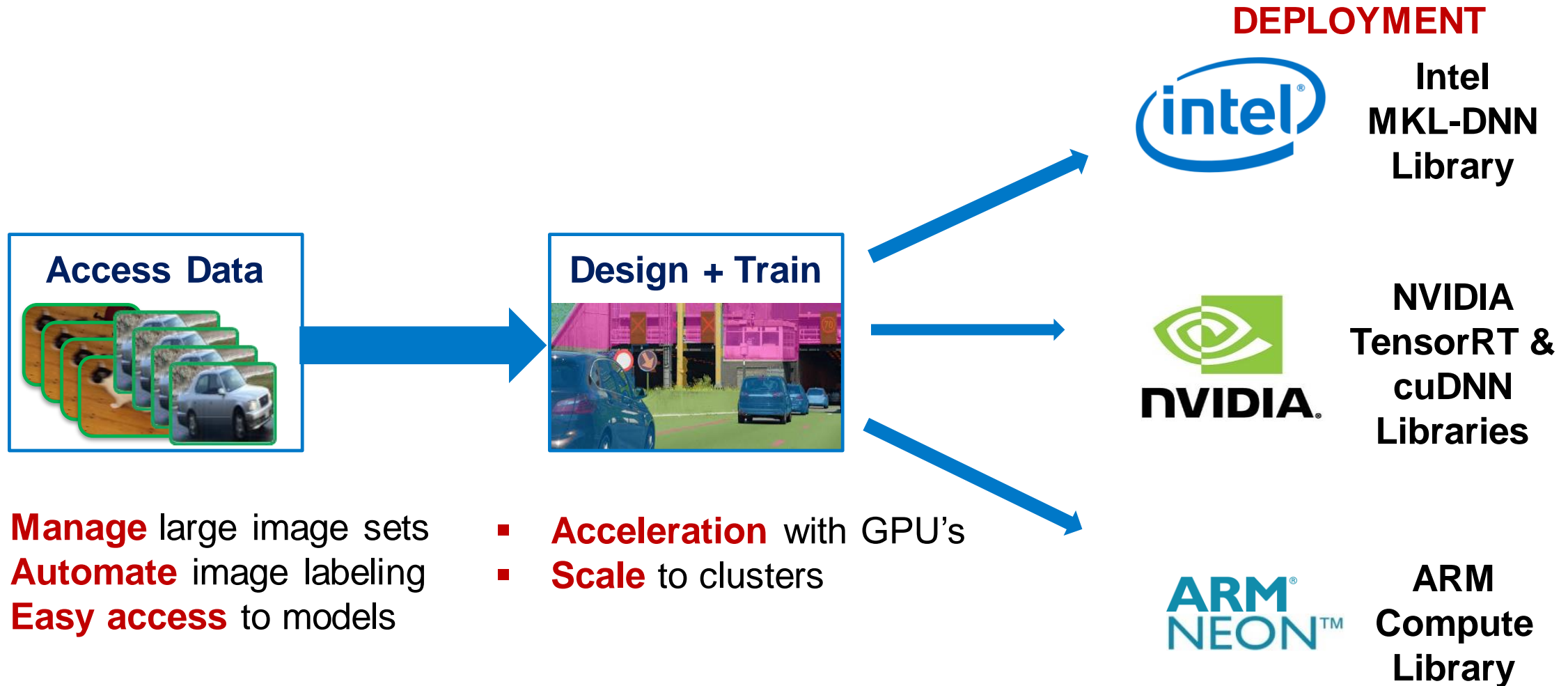
**Drive platform**



## INFERENCE ENGINE DEPLOYMENT

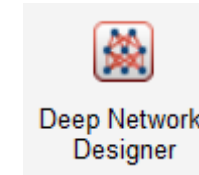


# Summary- MATLAB Deep Learning Framework



# Summary

- ✓ Create and validate Deep learning models
- ✓ Automate ground truth labeling
- ✓ Access large amount of data from cluster/cloud
- ✓ Interoperability with Deep learning frameworks
- ✓ Visualization and hyperparameter tuning
- ✓ Seamlessly scale training to GPUs, clusters and cloud
- ✓ Deployment on embedded targets and web services



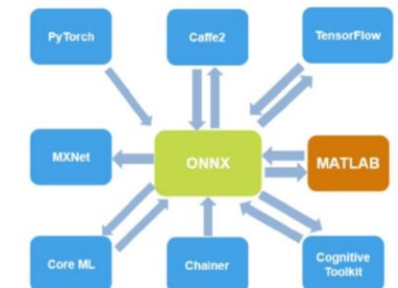
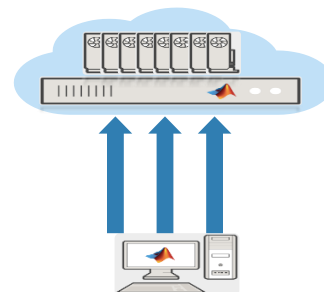
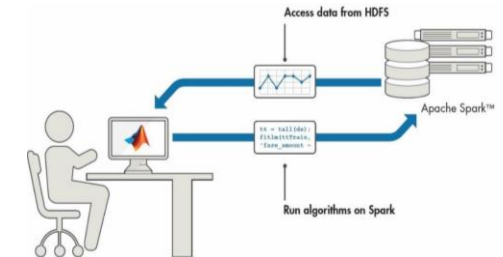
Audio Labeler



Image Labeler



Video Labeler



MATLAB Courses

https://matlabacademy.mathworks.com

MathWorks®

## MATLAB Courses

Search MathWorks.com

### My Self-Paced Courses

**Most Recent Activity**

**Deep Learning Onramp**

[Resume](#)

13% Unlimited access

[View/Share Certificate](#)

### Getting Started

**Launch**

**MATLAB Onramp**

Get started quickly with the basics of MATLAB.

[View/Share Certificate](#)

[Settings](#)

**Resume**

**Deep Learning Onramp**

Get started quickly using deep learning methods to perform image recognition.

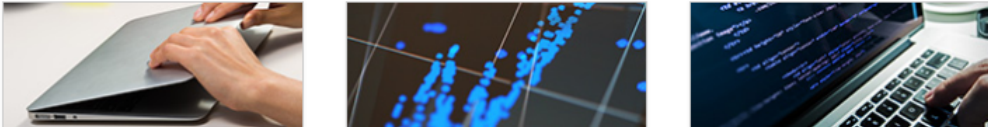
13% Unlimited access

[View/Share Certificate](#)

[Settings](#)

### Additional Self-Paced Courses

#### Core MATLAB



Deep Learning Onramp

https://matlabacademy.mathworks.com/R2018a/portal.html?course=deeplearning

MY COURSES

## Deep Learning Onramp

(0% complete) Gabriele Bunkheila

Deep Learning Onramp

[First time here?](#)

### 1. Introduction

Familiarize yourself with Deep Learning concepts and the course.

- [Deep Learning for Image Recognition Course Overview](#)

### 2. Using Pretrained Networks

Perform classifications using a network already created and trained.

- [Course Example - Identify Objects in Some Images](#)
- [Making Predictions](#)
- [CNN Architecture](#)
- [Investigating Predictions](#)
- [Image Datastores](#)

### 3. Performing Transfer Learning

Modify a pretrained network to classify images into specified classes.

- [What is Transfer Learning](#)
- [Components Needed for Transfer Learning](#)
- [Preparing Training Data](#)
- [Modifying Network Layers](#)
- [Setting Training Options](#)
- [Training the Network](#)
- [Evaluating Performance](#)
- [Transfer Learning Summary](#)

### 4. Preprocessing Images

Adjust raw images to make them usable with a given network.

- [Preparing Images to Use as Input](#)
- [Adding Custom Import Functions to Image Datastores](#)
- [Augmenting Images in a Datastore](#)

### 5. Conclusion

Learn next steps and give feedback on the course.

- [Further Deep Learning Tasks](#)
- [Survey](#)

[Available Here](#)





# MATLAB and Simulink Training

Overview | Course Offerings | Course Schedule | Self-Paced Courses | Training At Your Facility | Certification | More ▾ | Contact Training

< Course Schedule

## Prerequisites

*MATLAB Fundamentals*  
*Deep Learning Onramp*



**This course is also offered in an online, self-paced format.**

Self-paced courses provide active engagement with MATLAB through in-browser, hands-on exercises that you can complete anytime, anywhere, at your own pace.

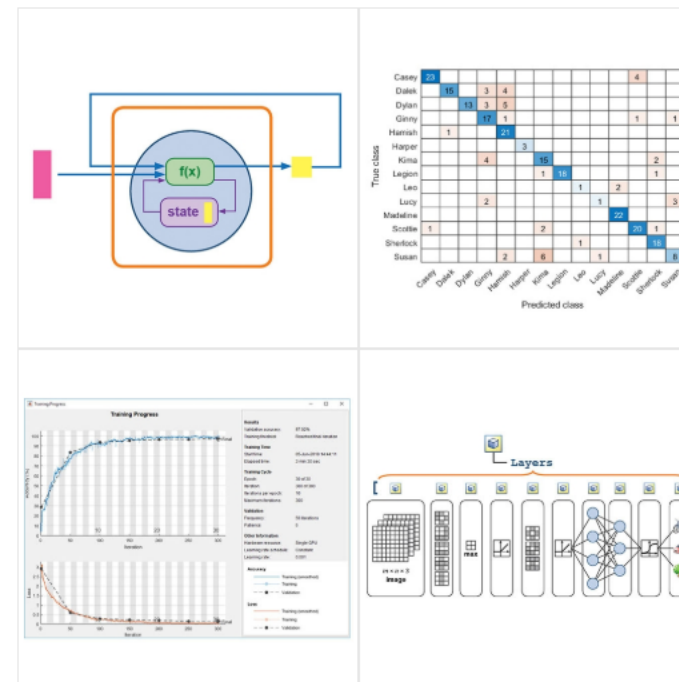
▶ Watch: The Advantages of Self-Paced Training (1:03)

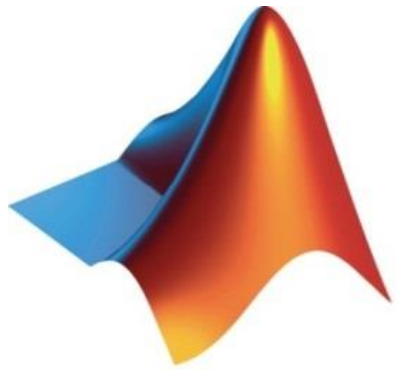
## Deep Learning with MATLAB

This two-day course provides a comprehensive introduction to practical deep learning using MATLAB®. Attendees will learn how to create, train, and evaluate different kinds of deep neural networks. Topics include:

- Importing image and sequence data
- Using convolutional neural networks for image classification, regression, and object detection
- Using long short-term memory networks for sequence classification and forecasting
- Modifying common network architectures to solve custom problems
- Improving the performance of a network by modifying training options

[See detailed course outline](#)





# MathWorks®

*Accelerating the pace of engineering and science*

