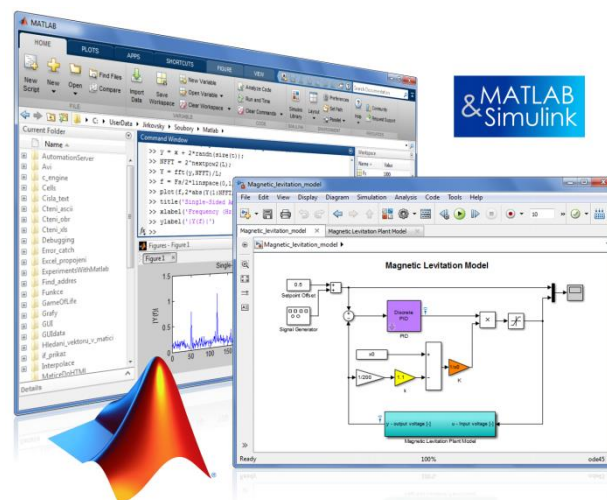


8.9.2016 Brno

# TCC 2016

## Deep Learning with MATLAB



Jan Studnička  
studnicka@humusoft.cz

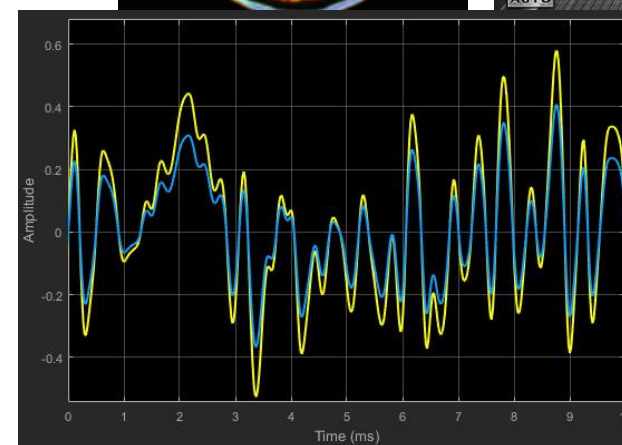
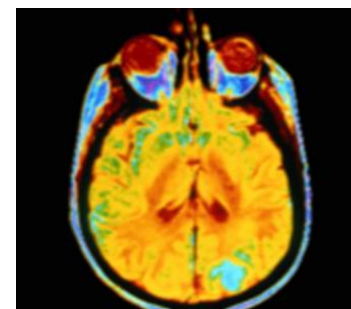
[www.humusoft.cz](http://www.humusoft.cz)  
[info@humusoft.cz](mailto:info@humusoft.cz)

[www.mathworks.com](http://www.mathworks.com)

# Computer Vision Applications

## Computer Vision

- Pedestrian and traffic sign detection
- Landmark identification
- Scene recognition
- Medical diagnosis and drug discovery
- Public Safety / Surveillance
- Automotive
- Robotics



and many more...

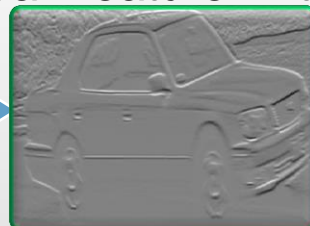
# What is Deep Learning ?

Deep learning performs **end-end learning** by learning **features, representations and tasks** directly from images, text and sound

## Traditional Machine Learning



Manual Feature Extraction



Classification

Machine Learning

Car ✓

Truck ✗

⋮

Bicycle ✗

## Deep Learning approach



Convolutional Neural Network (CNN)

End-to-end learning

Feature learning + Classification

Car ✓

Truck ✗

⋮

Bicycle ✗

# Deep Learning with MATLAB for Computer Vision

- Autoencoders

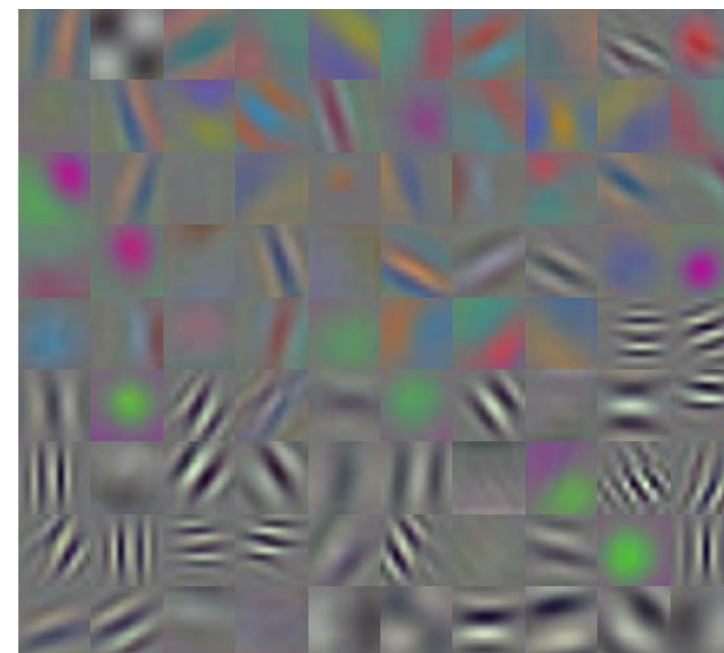
- Example: Classify digits in images

- Convolutional Neural Networks (CNN)

- Trained on massive sets of data
- High accuracy

**Confusion Matrix**

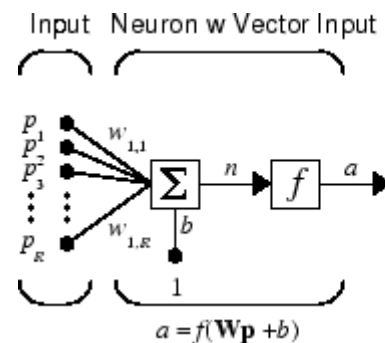
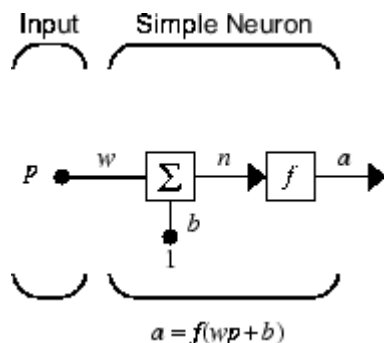
Output Class	1	2	3	4	5	6	7	8	9	10	Accuracy
1	418 8.4%	3 0.1%	3 0.1%	1 0.0%	3 0.1%	16 0.3%	15 0.3%	9 0.2%	7 0.1%	7 0.1%	36.7%
2	10 0.2%	407 8.1%	21 0.4%	5 0.1%	2 0.0%	3 0.1%	8 0.2%	8 0.2%	8 0.2%	10 0.2%	34.4%
3	14 0.3%	10 0.2%	401 8.0%	0 0.0%	36 0.7%	0 0.0%	11 0.2%	31 0.6%	2 0.0%	4 0.1%	78.8%
4	14 0.3%	4 0.1%	1 0.0%	446 8.9%	0 0.0%	5 0.1%	0 0.0%	14 0.3%	12 0.2%	0 0.0%	39.9%
5	1 0.0%	2 0.0%	42 0.8%	0 0.0%	407 8.1%	29 0.6%	0 0.0%	46 0.9%	3 0.1%	6 0.1%	75.9%
6	20 0.4%	1 0.0%	7 0.1%	11 0.2%	11 0.2%	384 7.7%	0 0.0%	32 0.6%	13 0.3%	31 0.6%	75.3%
7	23 0.5%	21 0.4%	6 0.1%	8 0.2%	0 0.0%	0 0.0%	435 8.7%	24 0.5%	6 0.1%	0 0.0%	33.2%
8	0 0.0%	8 0.2%	12 0.2%	12 0.2%	32 0.6%	28 0.6%	10 0.2%	275 5.5%	12 0.2%	27 0.5%	66.1%
9	0 0.0%	21 0.4%	0 0.0%	16 0.3%	1 0.0%	4 0.1%	19 0.4%	22 0.4%	429 8.6%	10 0.2%	32.2%
10	0 0.0%	23 0.5%	7 0.1%	1 0.0%	8 0.2%	31 0.6%	2 0.0%	39 0.8%	8 0.2%	405 8.1%	77.3%
	83.6%	81.4%	80.2%	89.2%	81.4%	76.8%	87.0%	55.0%	85.8%	81.0%	80.1%
	16.4%	18.6%	19.8%	10.8%	18.6%	23.2%	13.0%	45.0%	14.2%	19.0%	19.9%
	1	2	3	4	5	6	7	8	9	10	
	<b>Target Class</b>										



A visualization of learned weights of the first layer of a CNN.

# Neural Network

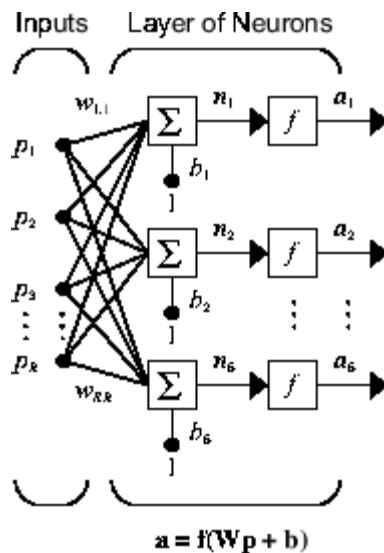
- Single neuron



Where

$R$  = number of elements in input vector

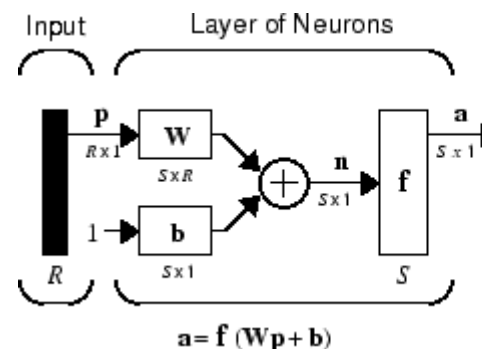
- Layer of Neurons



Where

$R$  = number of elements in input vector

$S$  = number of neurons in layer



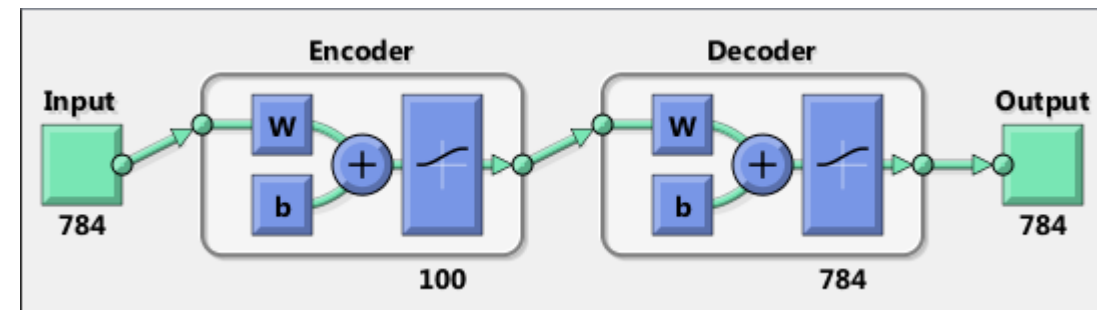
Where...

$R$  = number of elements in input vector

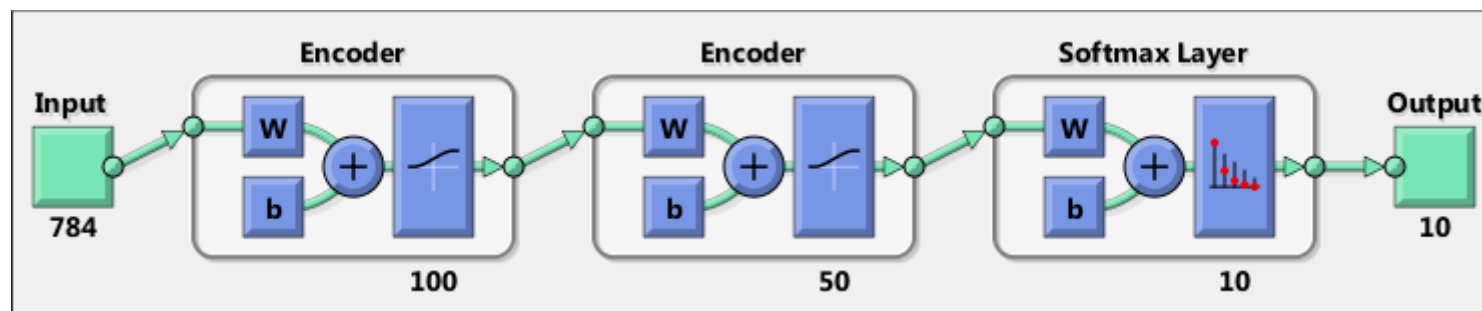
$S$  = number of neurons in layer 1

# Autoencoders

- **Unsupervised Learning**
  - Hidden layer – Encoder



- **Pretrain Deep Neural Network**
  - Hidden layers – Encoders of pretrained Autoencoders

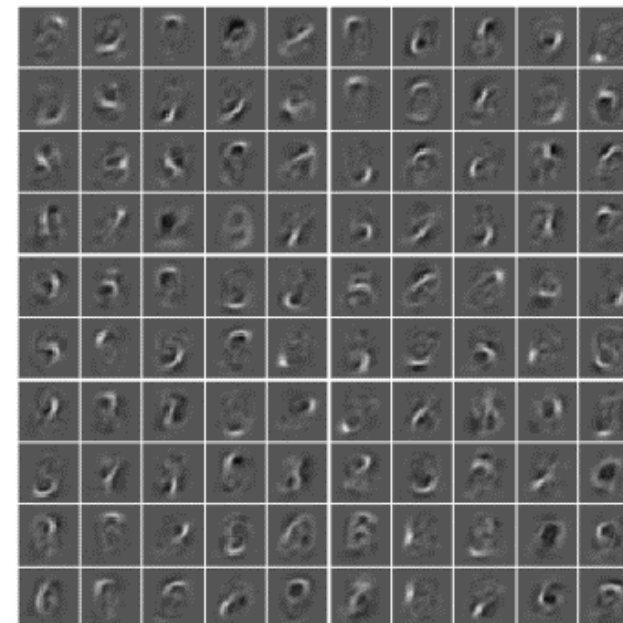


# Digit Classification

- **Classify digits in images**

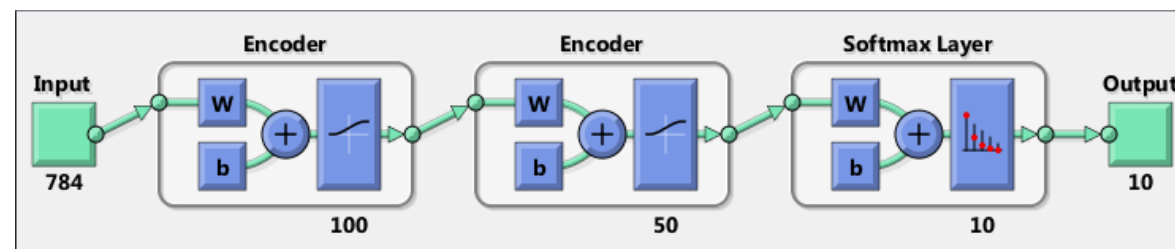
- **Data:**

- 28 x 28 pixels
- 10 digit classes
- 5000 samples

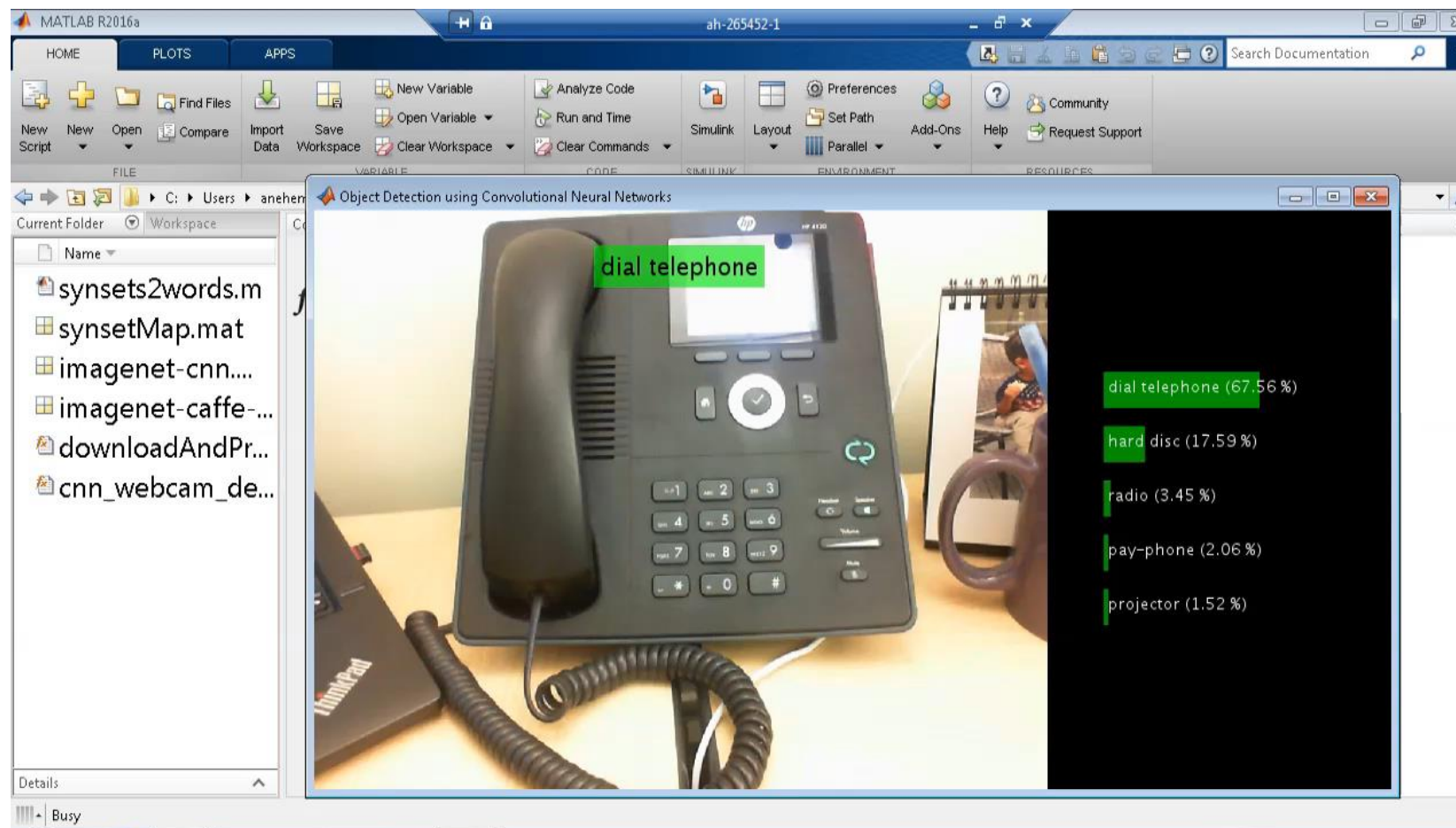


- **Solution:**

- 2 hidden layers – autoencoders
- Classification – Softmax layer
- Stack the Encoders with the Softmax layer to form a Deep Network
- Fine-tune the entire Deep Network – Classification



# Convolutional Neural Networks: Live Object Recognition with Webcam

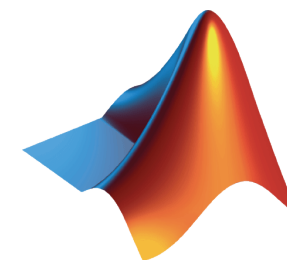


The image shows a MATLAB R2016a window titled "Object Detection using Convolutional Neural Networks". The main display area shows a webcam feed of a desk with a black HP 4120 dial telephone. A green bounding box is drawn around the telephone, with the text "dial telephone" overlaid. To the right of the image, a list of detected objects is shown with their corresponding confidence percentages:

- dial telephone (67.56 %)
- hard disc (17.59 %)
- radio (3.45 %)
- pay-phone (2.06 %)
- projector (1.52 %)

The MATLAB interface includes a toolbar with various icons for file operations, workspace management, and code execution. The left sidebar shows a file explorer with a list of files in the current folder:

- synsets2words.m
- synsetMap.mat
- imagenet-cnn...
- imagenet-caffe...
- downloadAndPr...
- cnn\_webcam\_de...





# Why is Deep Learning so Popular ?

- **Results: Achieved substantially better results on ImageNet large scale recognition challenge**

- 95% + accuracy on ImageNet 1000 class challenge

- **Computing Power: GPU's and advances to processor technologies have enabled us to train networks on massive sets of data.**

- **Data: Availability of storage and access to large sets of labeled data**

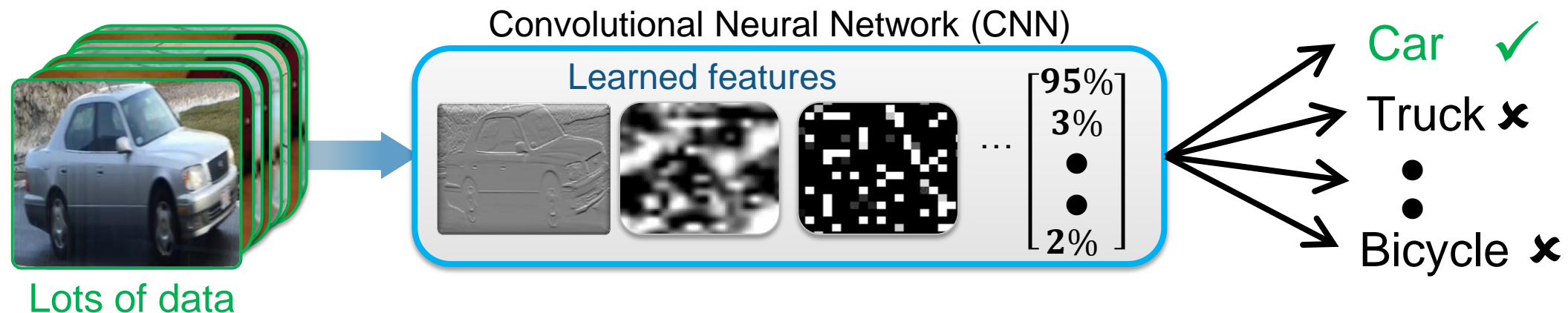
- E.g. ImageNet , PASCAL VoC , Kaggle

Year	Error Rate
Pre-2012 (traditional computer vision and machine learning techniques)	> 25%
2012 (Deep Learning)	~ 15%
2015 (Deep Learning)	<5 %

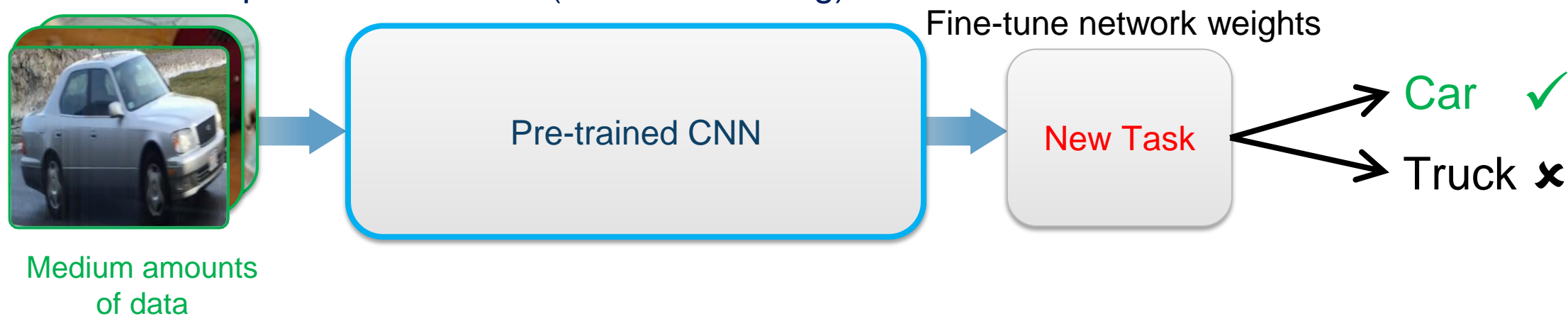


# Two Approaches for Deep Learning

## 1. Train a Deep Neural Network from Scratch

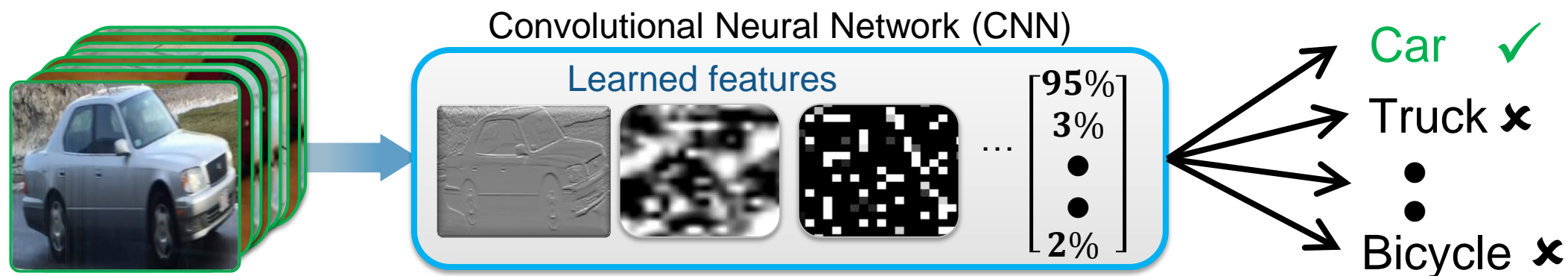


## 2. Fine-tune a pre-trained model (transfer learning)



# Two Deep Learning Approaches

## Approach 1: Train a Deep Neural Network from Scratch



Recommended only when:

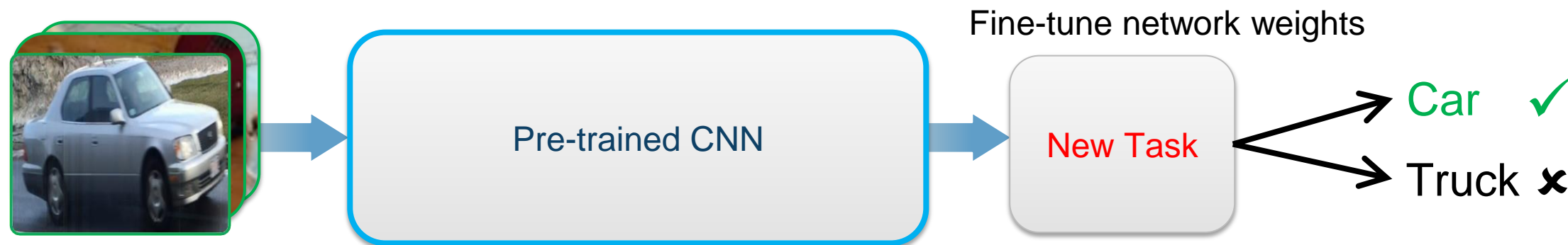
<b>Training data</b>	1000s to millions of labeled images
<b>Computation</b>	Compute intensive (requires GPU)
<b>Training Time</b>	Days to Weeks for real problems
<b>Model accuracy</b>	High (can overfit to small datasets)

# Two Deep Learning Approaches

## Approach 2: Fine-tune a pre-trained model (transfer learning)

### CNN trained on massive sets of data

- Learned robust representations of images from larger data set
- Can be fine-tuned for use with *new data or task* with small – medium size datasets



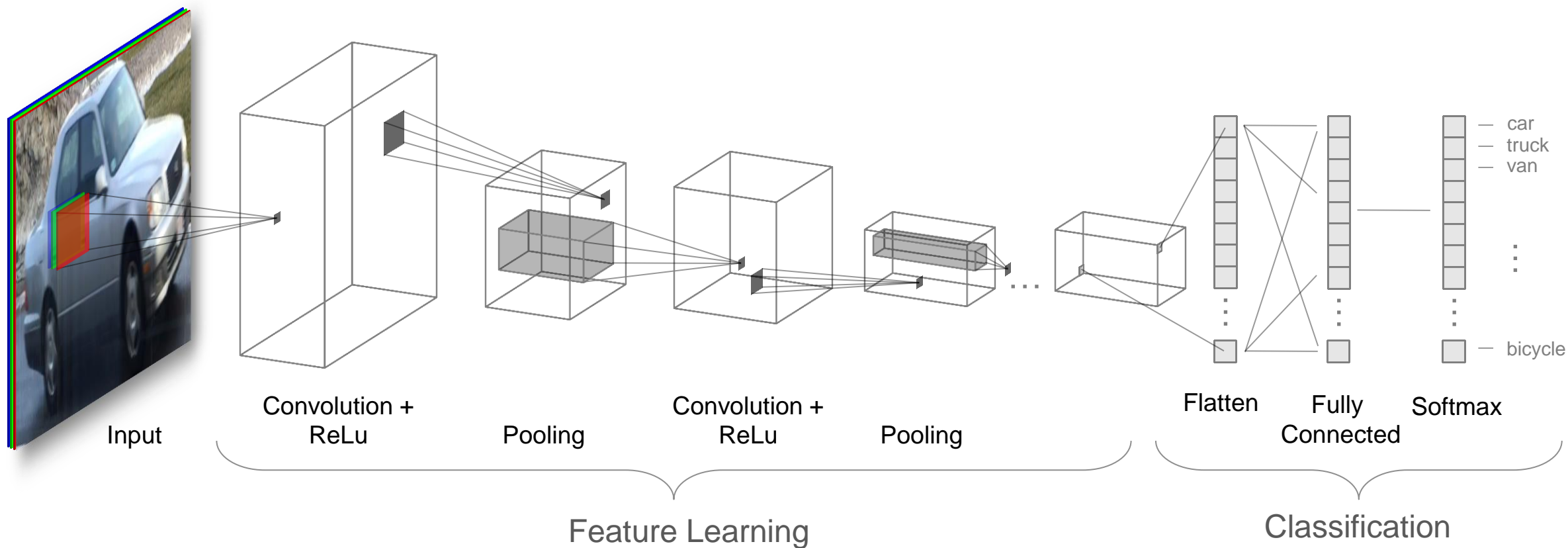
New Data

Recommended when:

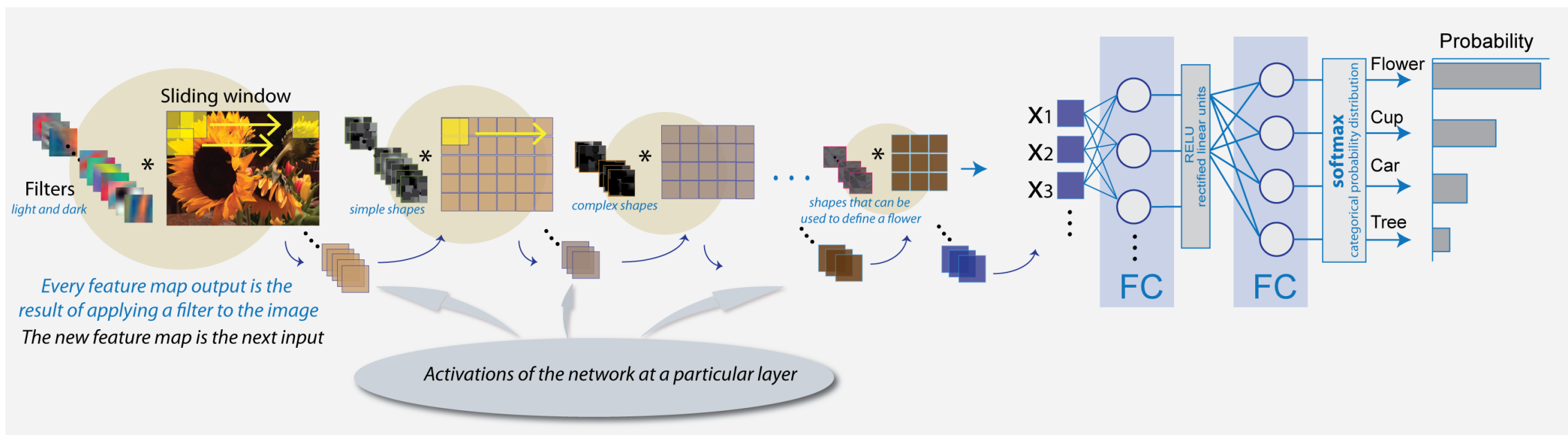
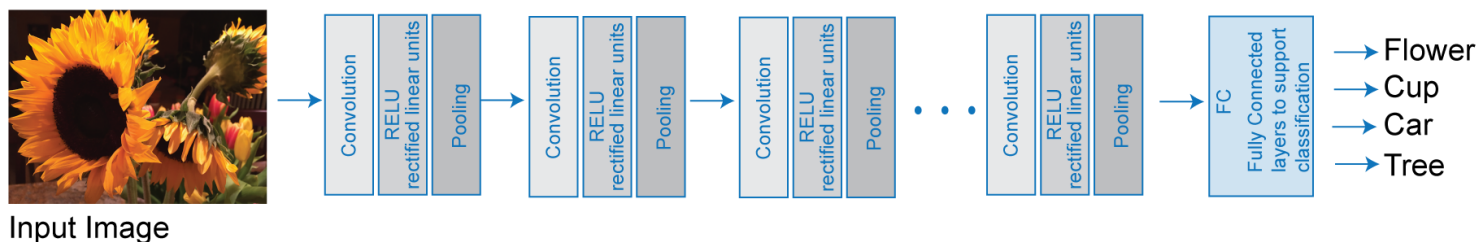
<b>Training data</b>	100s to 1000s of labeled images (small)
<b>Computation</b>	Moderate computation (GPU optional)
<b>Training Time</b>	Seconds to minutes
<b>Model accuracy</b>	Good, depends on the pre-trained CNN model

# Convolutional Neural Networks

- Train “deep” neural networks on structured data (e.g. images, signals, text)
- Implements Feature Learning: Eliminates need for “hand crafted” features
- Trained using GPUs for performance

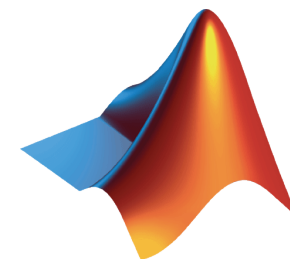


# Convolutional Neural Networks



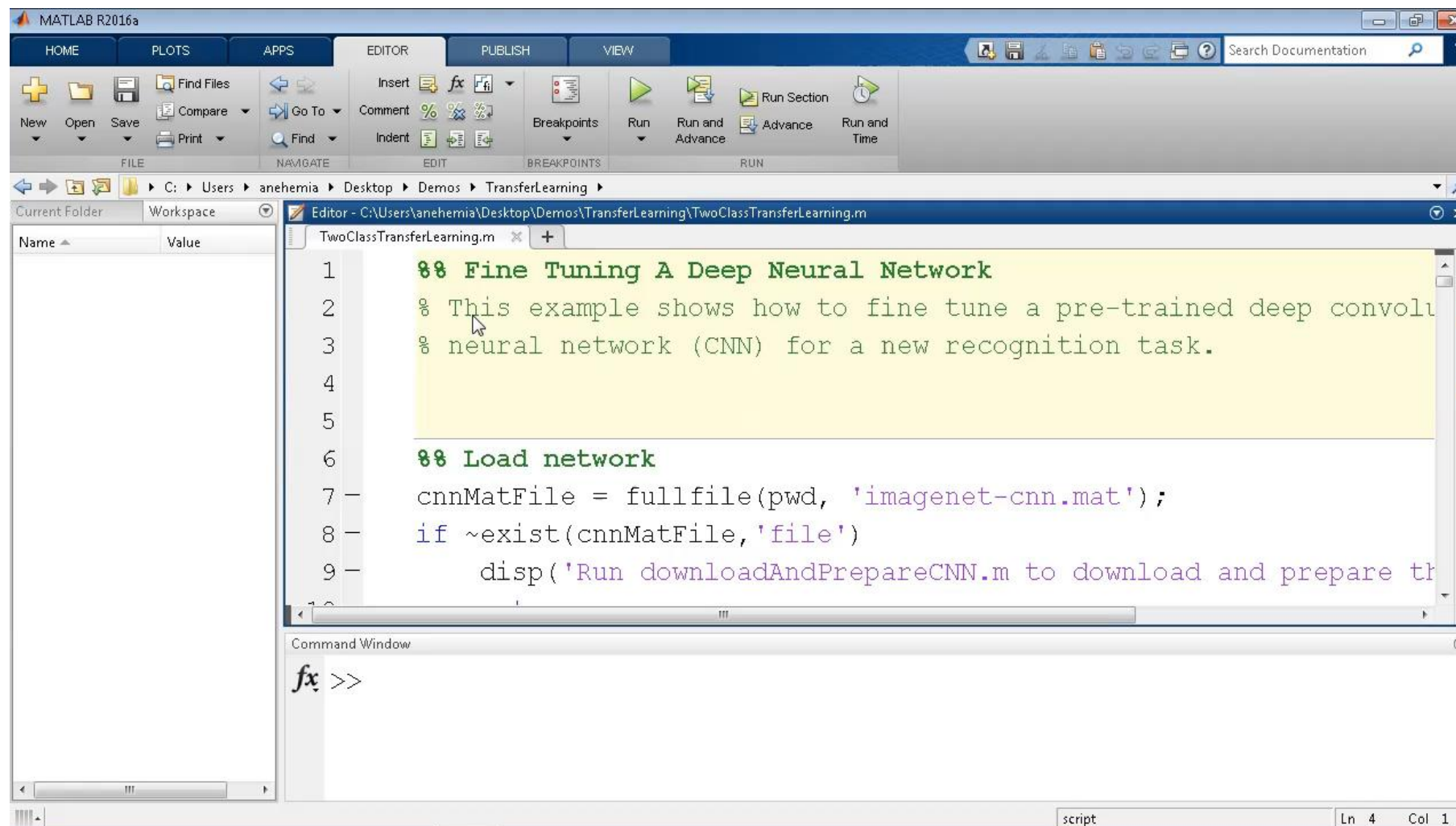
# Demo

*Fine-tune a pre-trained model (transfer learning)*



# Demo

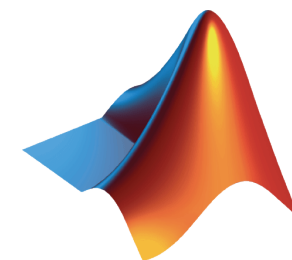
## *Fine-tune a pre-trained model (transfer learning)*



The image shows the MATLAB R2016a software interface. The main window displays a script titled 'TwoClassTransferLearning.m'. The script contains the following code:


```
1  %% Fine Tuning A Deep Neural Network
2  % This example shows how to fine tune a pre-trained deep convolu
3  % neural network (CNN) for a new recognition task.
4
5
6  %% Load network
7  cnnMatFile = fullfile(pwd, 'imagenet-cnn.mat');
8  if ~exist(cnnMatFile, 'file')
9      disp('Run downloadAndPrepareCNN.m to download and prepare th
```

The Command Window at the bottom shows the MATLAB prompt `fx >>`. The status bar at the bottom right indicates the current position is at line 4, column 1.





# Addressing Challenges in Deep Learning for CV

Challenge	Solution
Managing large sets of labeled images	<code>imageSet</code> or <code>imageDataStore</code> to handle large sets of images
Resizing, Data augmentation	<code>imresize</code> , <code>imcrop</code> , <code>imadjust</code> , <code>imageInputLayer</code> , etc.
Background in neural networks (deep learning)	Intuitive interfaces, well-documented architectures and examples
Computation intensive task (requires GPU)	Training supported on GPUs No GPU expertise is required
	Automate. Offload computations to a cluster and test multiple architectures



**International reseller of MathWorks for the Czech Republic and Slovakia**

Pobřežní 20

186 00 Praha 8

Česká republika

Email: [info@humusoft.cz](mailto:info@humusoft.cz)

[www.humusoft.cz](http://www.humusoft.cz)

Tel.: +420 284 011 720

[www.facebook.com/humusoft](http://www.facebook.com/humusoft)

[www.youtube.com/humusoft](http://www.youtube.com/humusoft)

[www.twitter.com/humusoft](http://www.twitter.com/humusoft)

