

# Computer Vision System Design

## (Deep Learning and 3D vision)

**Mandar Gujrathi**

**Applications Engineer**

**[Mandar.Gujrathi@mathworks.com.au](mailto:Mandar.Gujrathi@mathworks.com.au)**

# Why are we talking about Computer Vision today?

## **INTEL ACQUIRES COMPUTER VISION FOR IOT, AUTOMOTIVE**

By Doug Davis

## **Computer Vision Hardware and Software Market to Reach \$48.6 Billion by 2022, According to Tractica**

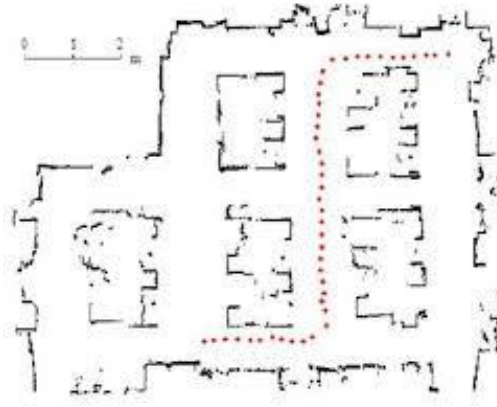
**Automotive and Consumer Markets to Surge in the Next Few Years, with Strong Growth Also Expected for Robotics and Security Applications**

## **Human-in-the-loop deep learning will help drive autonomous cars**

NAVEEN RAO, NERVANA SYSTEMS    JUNE 25, 2016 11:30 AM

# Agenda

- Stereo Vision



- Deep Learning

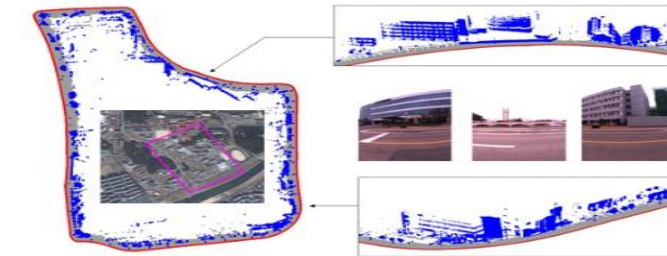
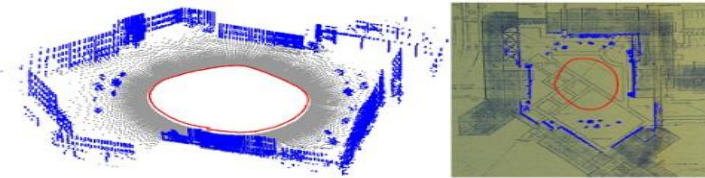
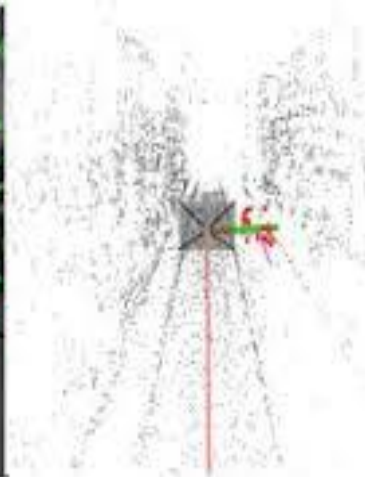


# Stereo Vision

- *Stereo vision is the process of extracting 3-D information from multiple 2-D views of a scene.*
- *Where can I find this capability?*

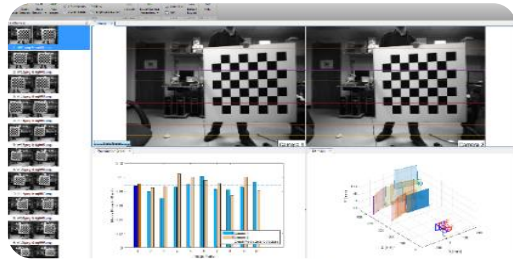
[Computer Vision System Toolbox](#)

# Point Cloud Application : ADAS

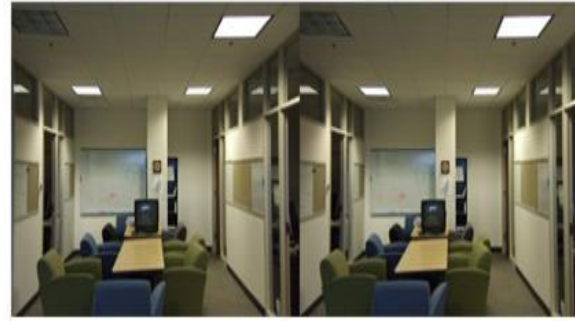




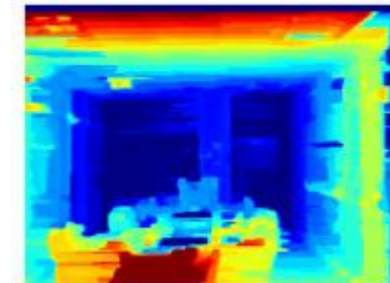
# Stereo vision workflow



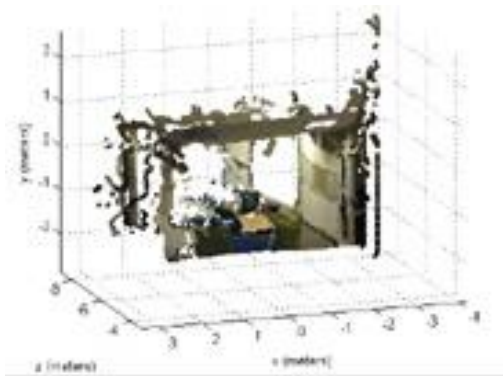
**Perform Stereo Camera Calibration**  
Get parameters



**Image Rectification**  
`RectifyStereoImages()`



**Compute Pixel Differences:**  
`disparity()`



**3D point cloud**  
`pcshow()`

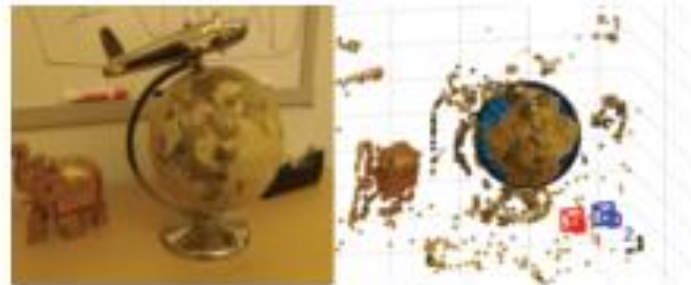
Stereo Vision

# Camera Calibration

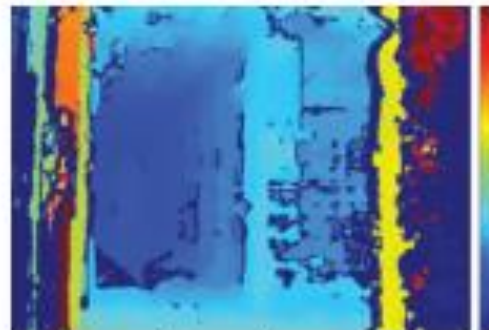
- Estimates the parameters of video camera.



Remove Lens Distortion



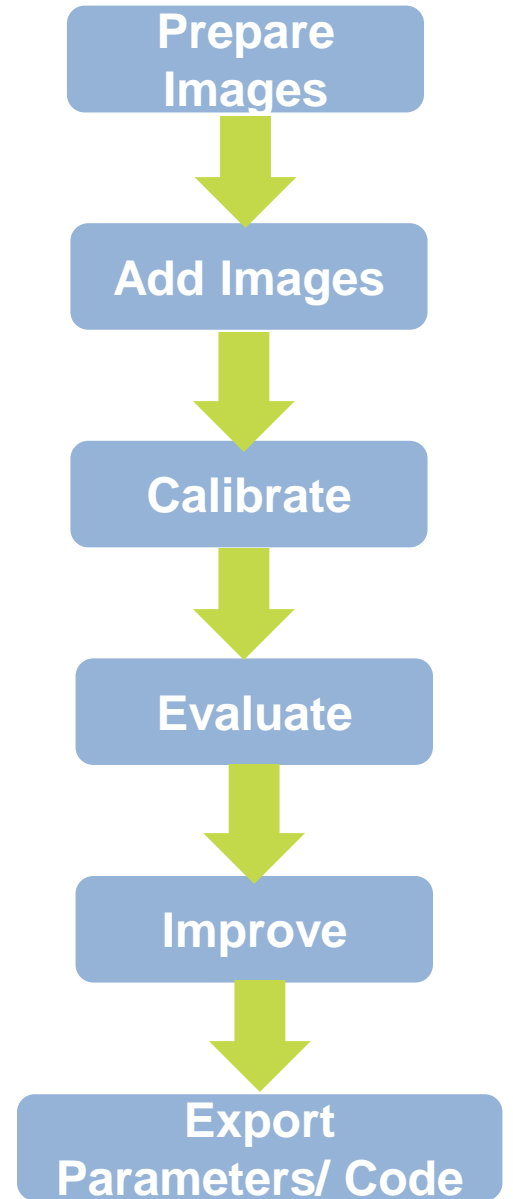
Estimate 3-D Structure from Camera Motion



Estimate Depth  
Using a Stereo Camera



Measure Planar Objects



# Camera Calibration Apps

**Stereo Camera Calibrator - Extrinsic**

**CALIBRATION**

Radial Distortion: Compute:  2 Coefficients  Skew  Tangential Distortion  3 Coefficients  Tangential Distortion

Optimization Options:  Optimize

Calibrate  Zoom In  Zoom Out  Pan  Default Layout  Export Camera Parameters  Help

FILE OPTIONS OPTIMIZATION CALIBRATE ZOOM LAYOUT EXPORT RESOURCES

Data Browser

Image

1: image1.png & image1.png  
 2: image10.png & image10.png  
 3: image11.png & image11.png  
 4: image12.png & image12.png  
 5: image13.png & image13.png  
 6: image14.png & image14.png  
 7: image15.png & image15.png  
 8: image16.png & image16.png  
 9: image17.png & image17.png  
 13: calib15.jpg

Camera 1

Camera 2

Legend:   
 ● Detected points   
 + Reprojected points   
 □ Checkerboard origin

Reprojection Errors

Mean Error in Pixels

Drag to select outliers

Overall Mean Error: 0.18 pixels

Extrinsic

Y (mm)

Z (mm)

X (mm)

Show pattern-centric view

Extrinsic

Z (mm)

X (mm)

Show pattern-centric view

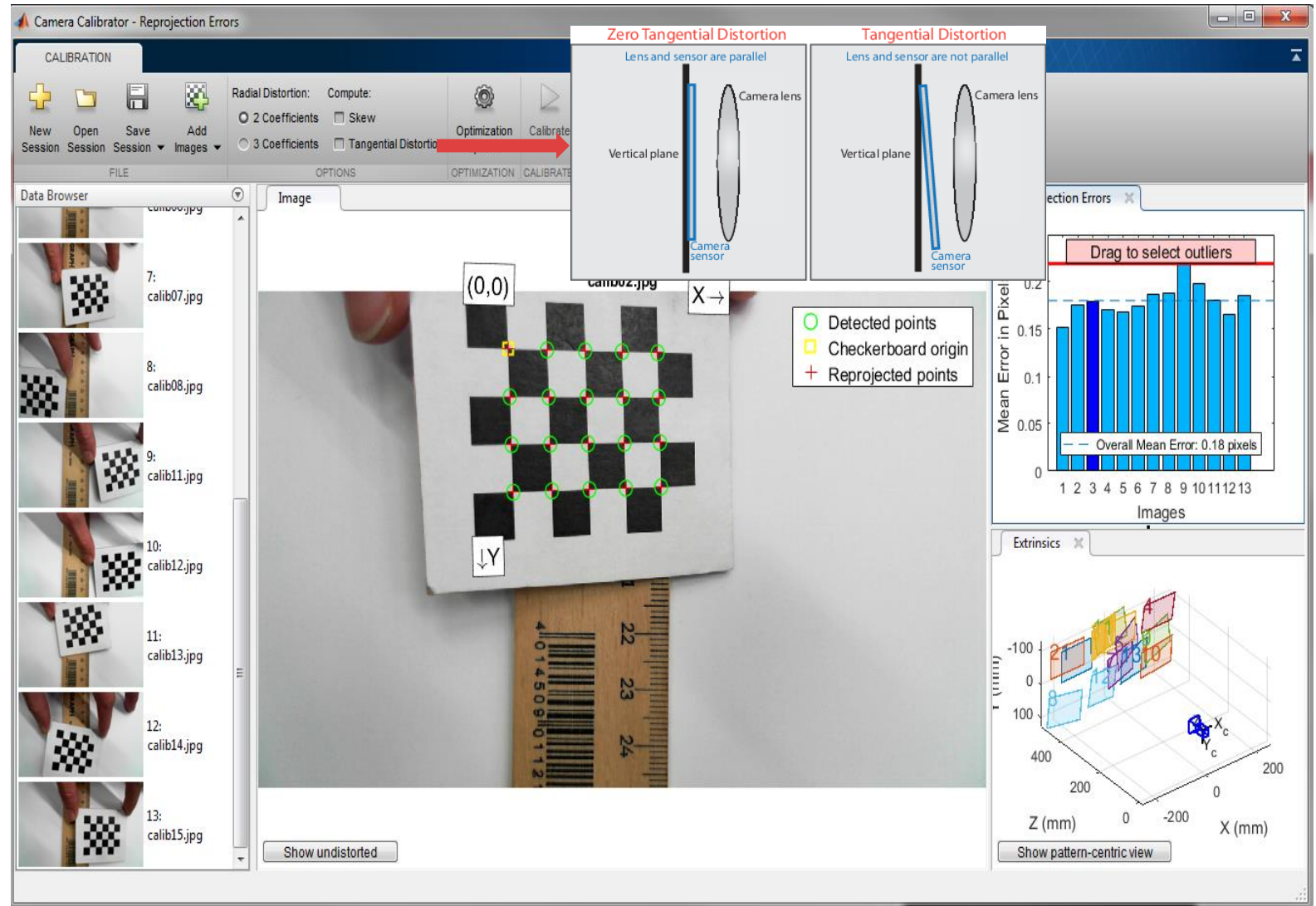
Show undistorted

Points and origin points



# Camera Calibration App Advantages

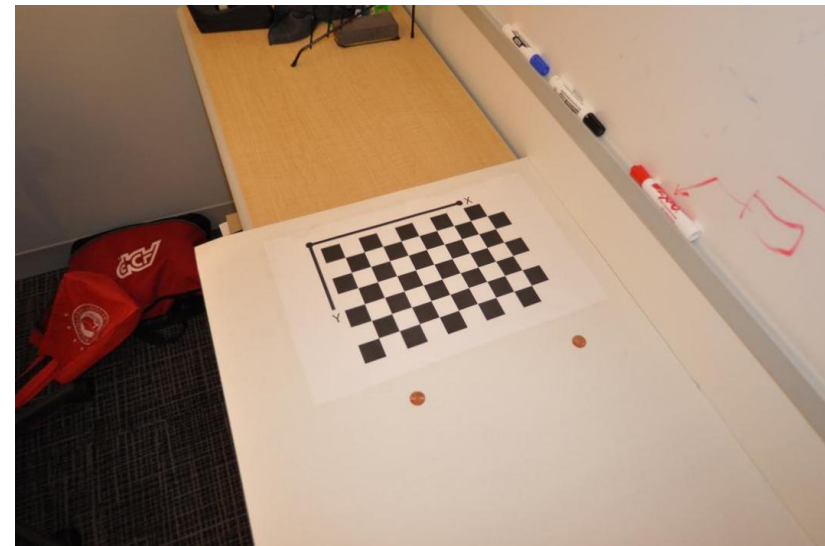
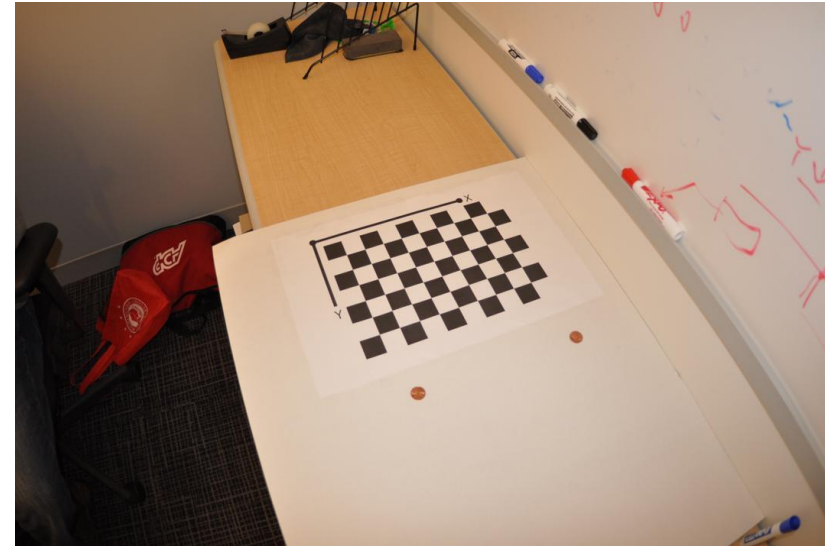
- Simplified workflow for estimating camera intrinsic and extrinsic parameters
- Removes the effects of lens distortion from an image
- Automatically detects checkerboard patterns
- Helps to evaluate accuracy
- Generates code !



# Remove Lens Distortion From an Image

**Removes radial and tangential distortion.**

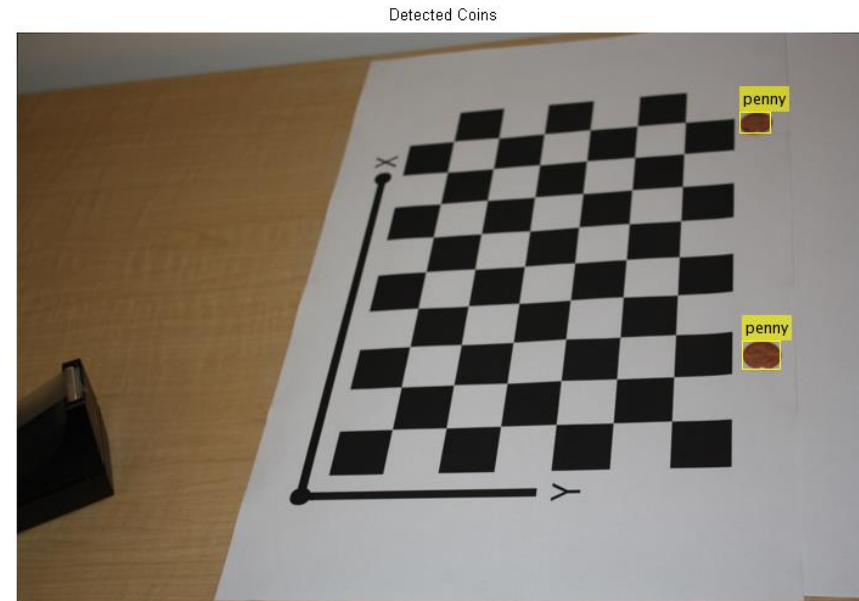
- Radial distortion (“barrel” or “pincushion”) is caused by the curvature of the lens
- Tangential distortion is caused by misalignment between the lens and the sensor



# Measuring Planar Objects With a Calibrated Camera

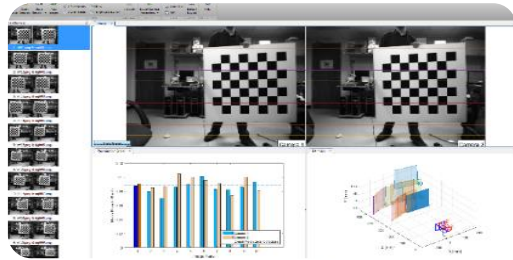
**Featured example: measure the diameter of a penny in millimeters.**

- Undistort the image
- Detect the penny
- Project points from the image into the world
- Measure the diameter in millimeters

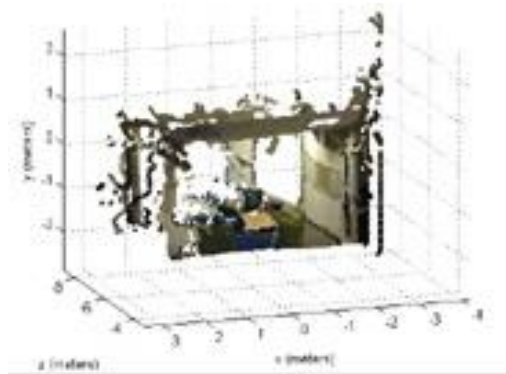


```
>> showdemo ( 'MeasuringPlanarObjectsExample' )
```

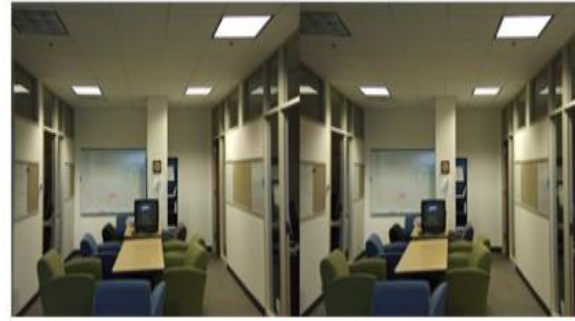
# Stereo vision workflow



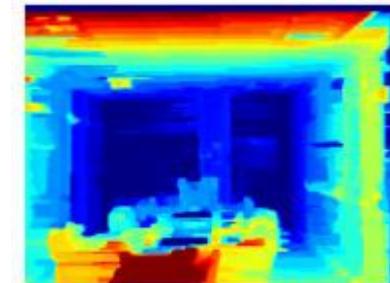
**Perform Stereo Camera Calibration**  
Get parameters



**3D point cloud**  
pcshow()



**Image Rectification**  
RectifyStereoImages()



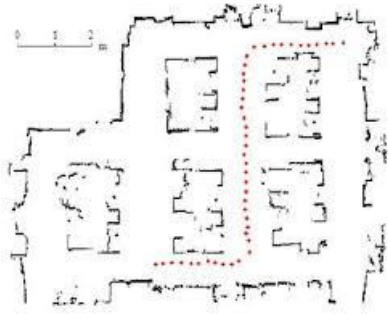
**Compute Pixel Differences:**  
Disparity map  
disparity()



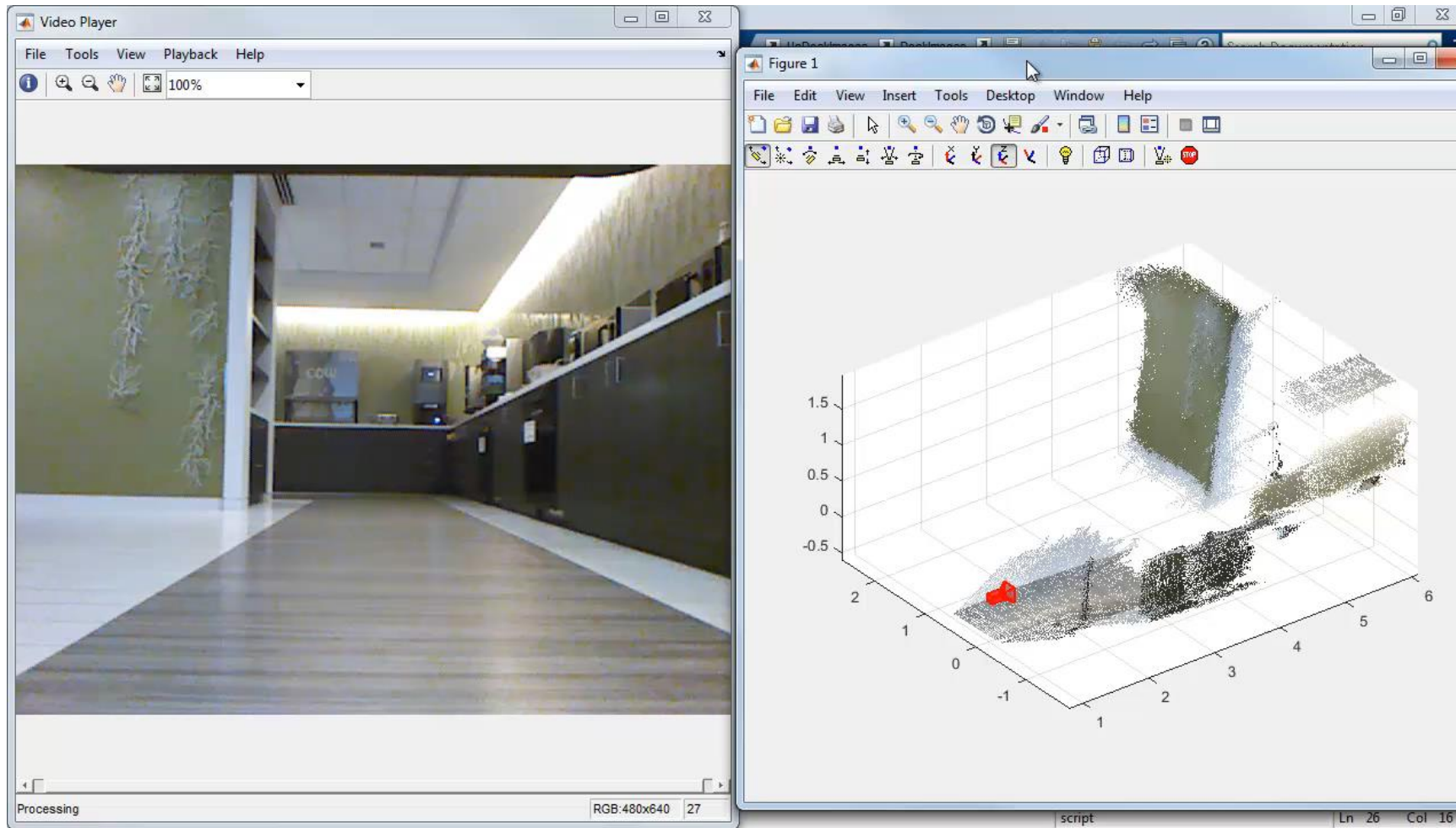
Stereo Vision



# Point Cloud Application – Robot Vision.

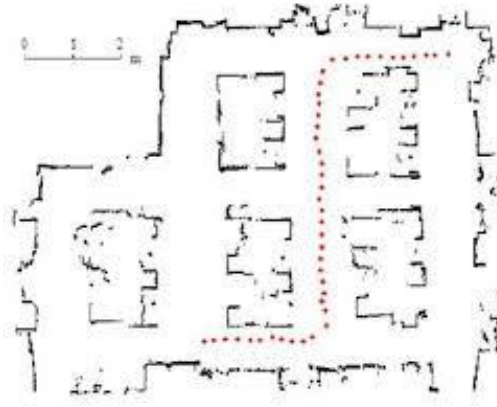


# 3-D Vision for Robotics



# Agenda

- Stereo Vision



- Deep Learning

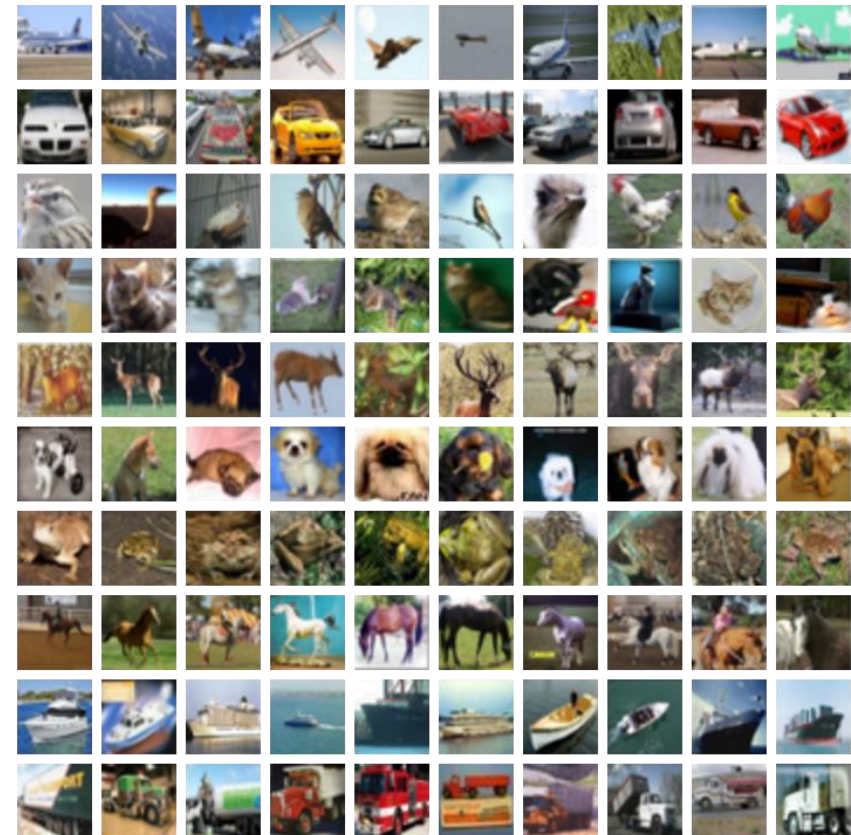


# Deep Learning

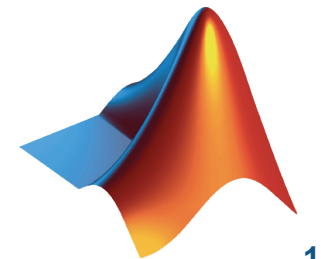
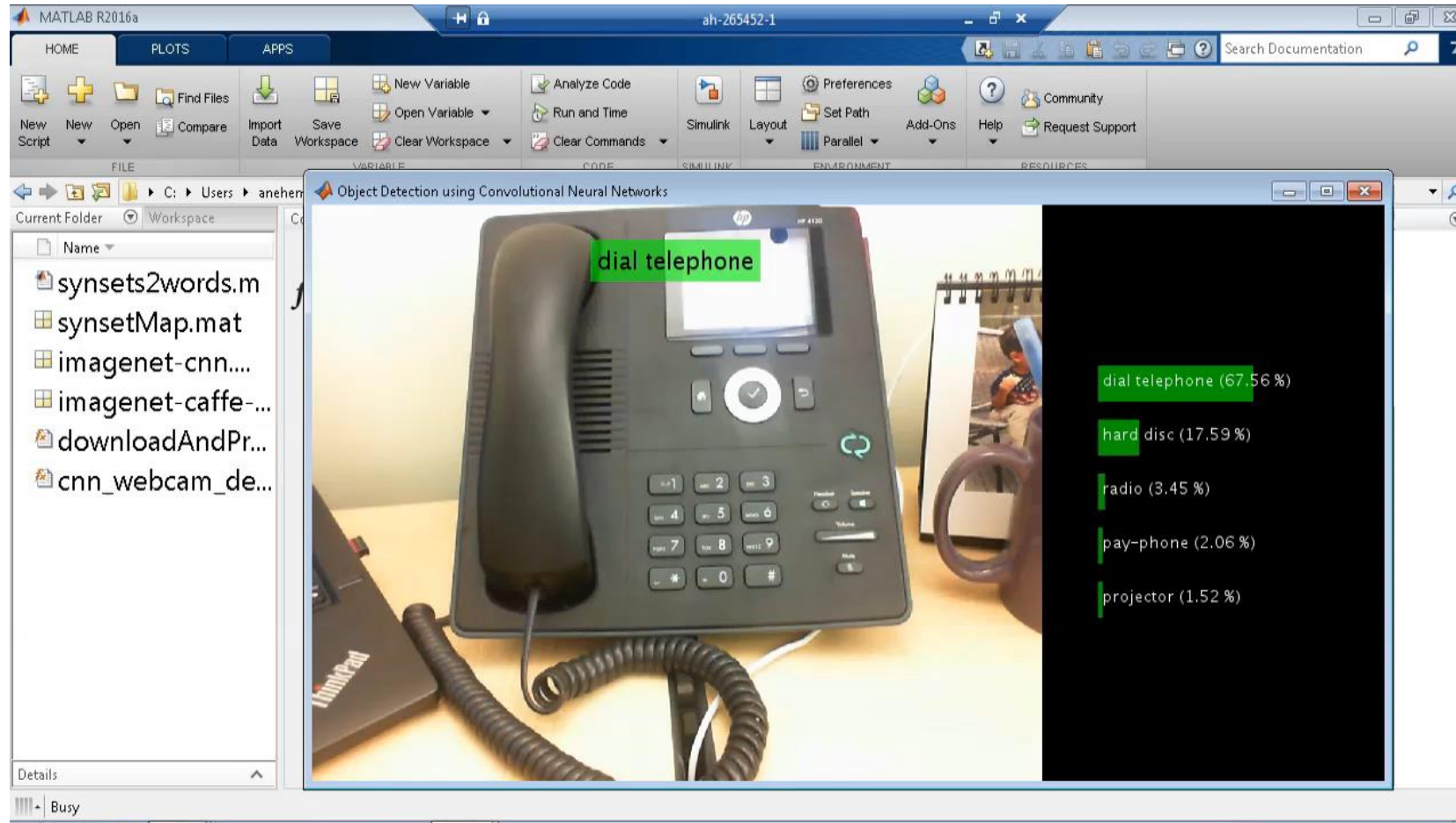


## How long it took us to...

- Question – At what age does a person recognise:
  - Car or Plane
  - Car or SUV
  - Toyota or Mazda



# Demo : Live Object Recognition with Webcam



# Demo

The image shows the MATLAB R2016a environment. The main window displays a script named 'TwoClassTransferLearning.m' with the following code:

```

163
164 %% Detect Cars using Background Subtraction then classify
165 se = strel('square', 3); % morphological filter for noise removal
166 while ~isDone(videoReader)
167
168     frame
169     frame
170     frame
    
```

The workspace on the left shows the following variables:

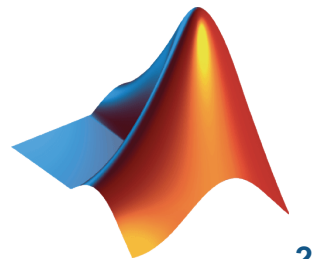
Name	Value
ans	0.9565
blobAna...	1x1 Blo...
carFiles	155x1 c...
cnnMat...	'C:\User...
confMat	[0.9348
foregro...	101x24...
foregro...	1x1 For...
frame	401x98...
frameS...	101x24...
i	150
imagen...	1x1 stru...
imagesI...	10x2 ce...
imds	1x1 lmc...
labels	92x1 ca...

The Command Window shows the output:

```

ans =
    0.9565
fx >>
    
```

A 'Deployable Video Player' window is overlaid on the script, displaying a video frame of a building with a red roof and a sign that reads 'MathWorks APPLE HILL CAMPUS'. The video player also shows a 'Dover' sign in the background.





# Deep Learning is Ubiquitous

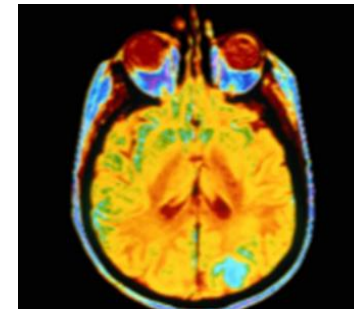
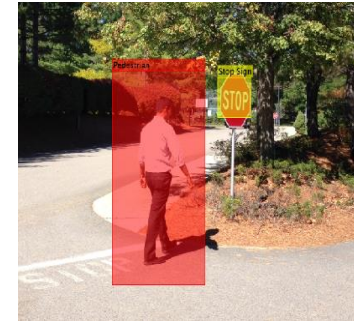
## Computer Vision

- Pedestrian and traffic sign detection
- Landmark identification
- Medical diagnosis and drug discovery
- Surveillance

## Text and Signal Processing

- Speech Recognition
- Speech & Text Translation

## Robotics & Controls



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 ✗

# 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 %



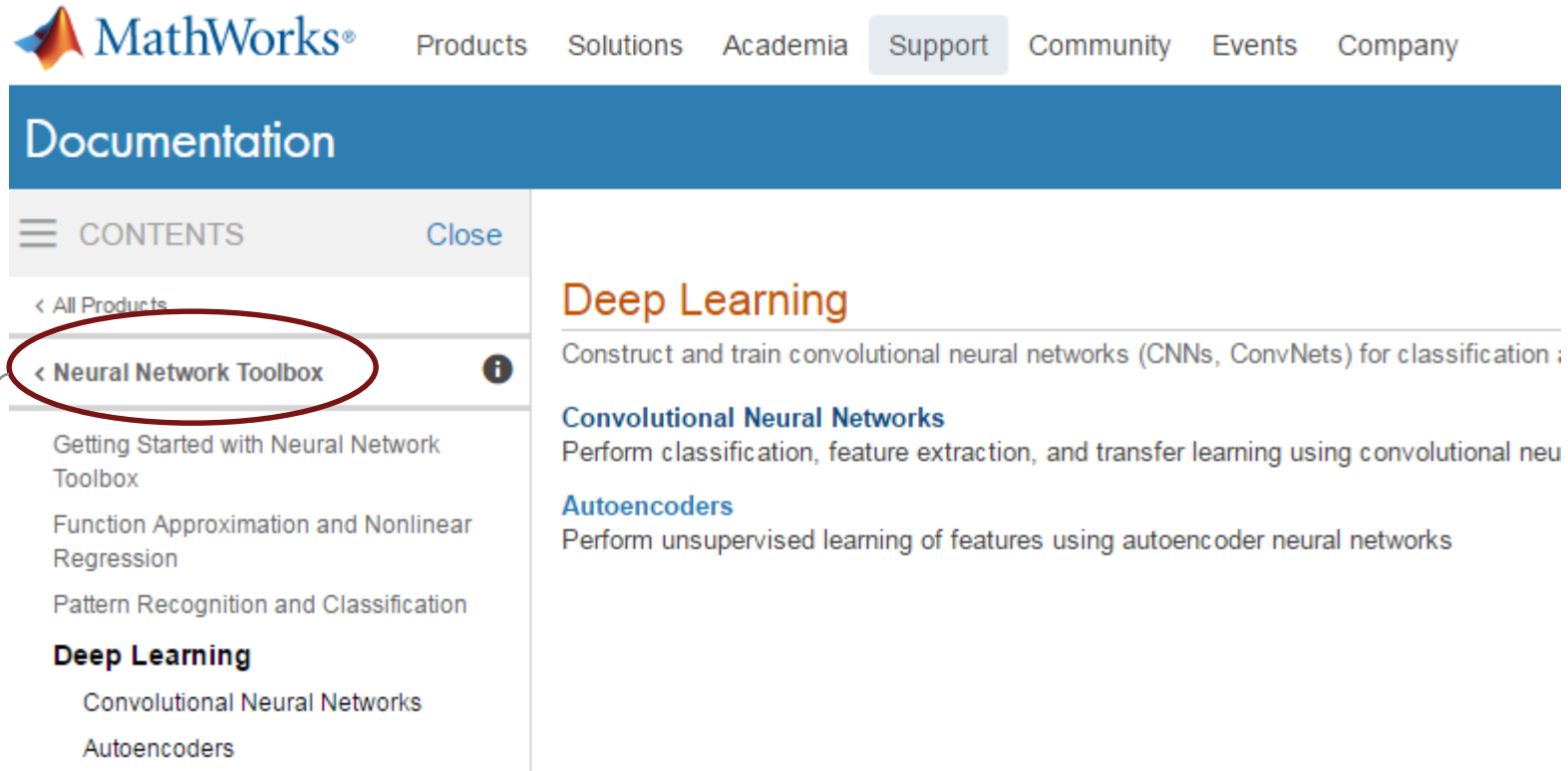
# How to build a Deep learning network?

# Challenges using Deep Learning for Computer Vision

<b>Steps</b>	<b>Challenge</b>
Importing Data	Managing large sets of labeled images
Preprocessing	Resizing, Data augmentation
Choosing an architecture	Background in neural networks (deep learning)
Training and Classification	Computation intensive task (requires GPU)
Iterative design	



# MATLAB does Deep Learning ?



The screenshot shows the MathWorks documentation website. At the top, there is a navigation bar with links for Products, Solutions, Academia, Support, Community, Events, and Company. Below this is a blue header with the word 'Documentation'. A sidebar menu is open, showing a tree structure of products. The 'Neural Network Toolbox' is highlighted with a red oval, and an arrow points from it to the text 'Neural Network Toolbox' below. Underneath 'Neural Network Toolbox', there are several sub-items, including 'Deep Learning', which is further expanded to show 'Convolutional Neural Networks' and 'Autoencoders'. The main content area on the right shows the 'Deep Learning' page, with a title 'Deep Learning' and a description: 'Construct and train convolutional neural networks (CNNs, ConvNets) for classification :'. Below this, there are two sub-sections: 'Convolutional Neural Networks' and 'Autoencoders', each with a brief description.

[Neural Network Toolbox](#)

[Deep Learning](#)

# Convolutional Neural Network

- Train “deep” neural networks on s
- Implements Feature Learning: Elin
- Trained using GPUs for performar

```

Construct Network Architecture
imageInputLayer
convolution2dLayer
reluLayer
crossChannelNormalizationLayer
averagePooling2dLayer
maxPooling2dLayer
fullyConnectedLayer
dropoutLayer
softmaxLayer
classificationLayer
    
```

Images, signals, text)  
“crafted” features

```

Train Network
trainingOptions
trainNetwork
    
```

```

Extract Features and Predict Outcomes
activations
predict
classify
    
```

```

Classes
Construct Network Architecture
Layer
    
```

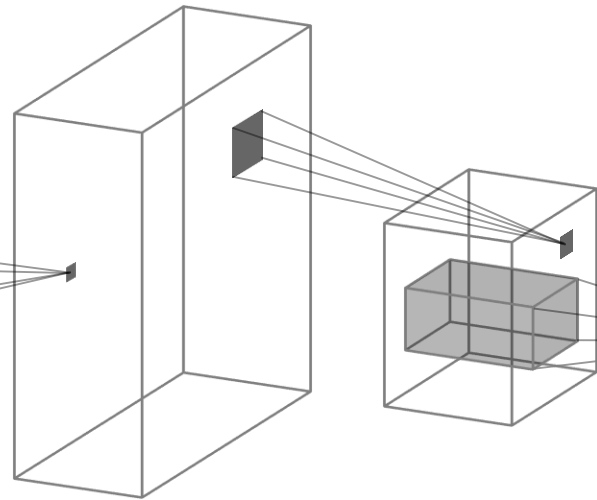
```

Train Network
SeriesNetwork
TrainingOptionsSGDM
    
```

more ↓



Input

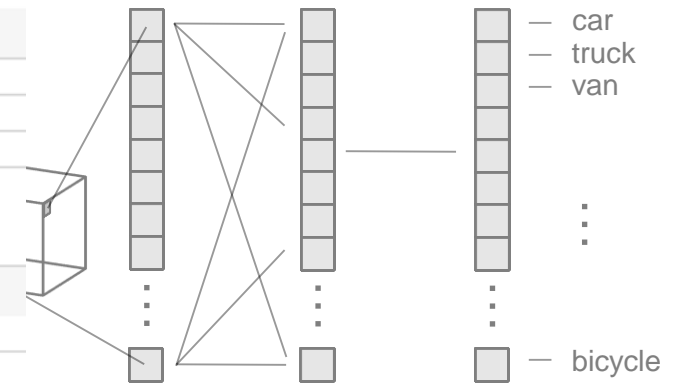


Convolution +  
ReLu

Pooling

Cor

Feature Learning



Flatten

Fully  
Connected

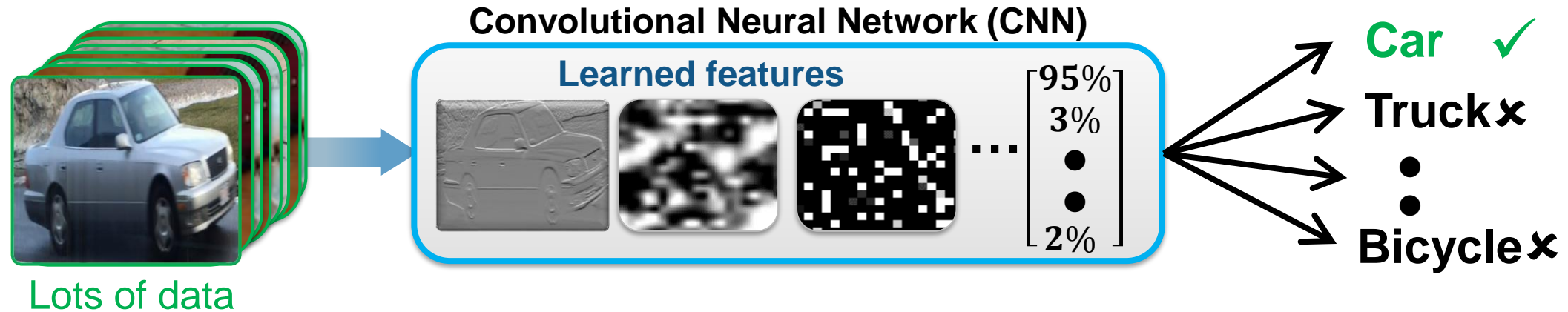
Softmax

Classification

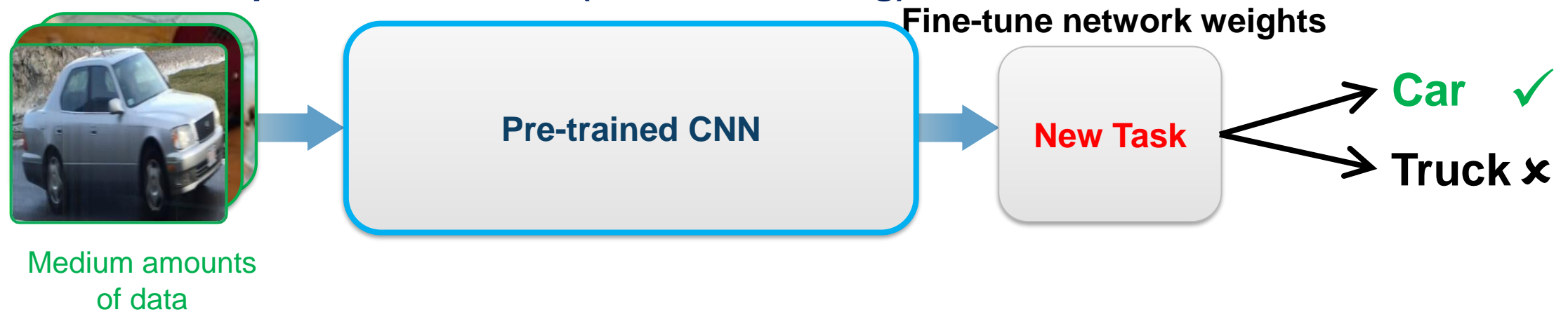
— car  
— truck  
— van  
⋮  
— bicycle

# Two Approaches for Deep Learning

## 1. Train a Deep Neural Network from Scratch

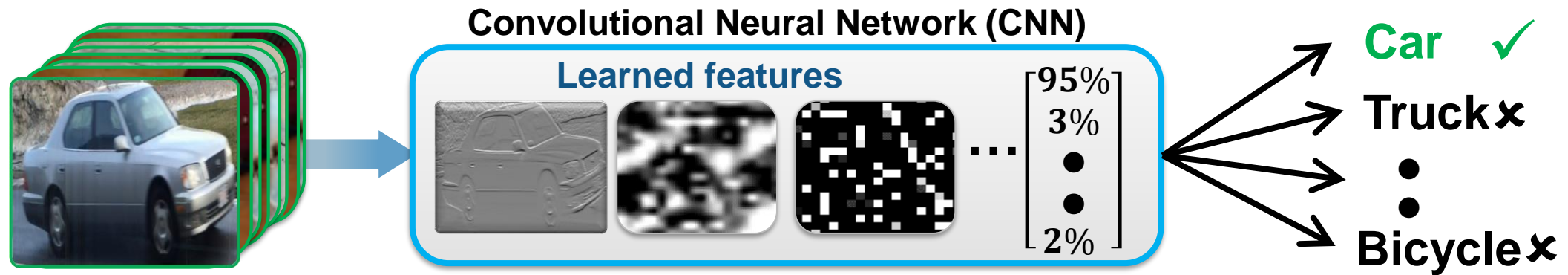


## 2. Fine-tune a pre-trained model ( transfer learning)



# Two Deep Learning Approaches

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



### Recommended only when:

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



# Demo: Classifying the CIFAR-10 dataset

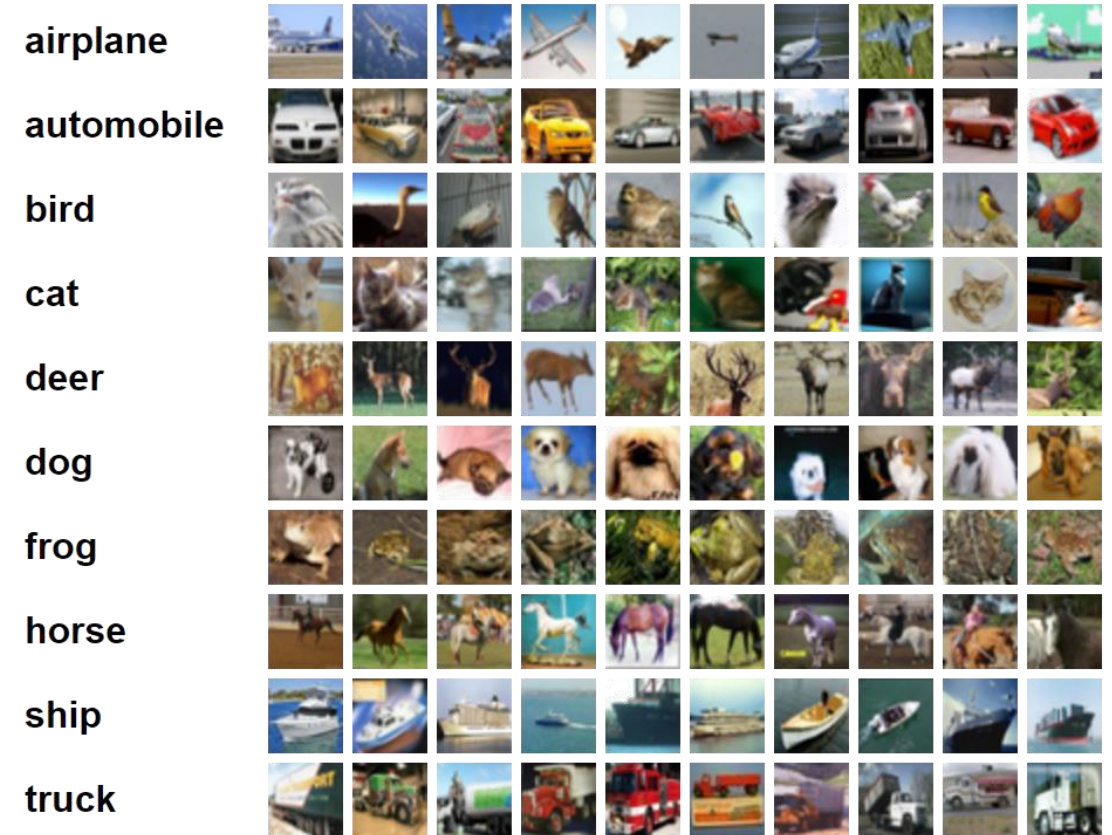
**Objective:** Train a Convolutional Neural Network to classify the CIFAR-10 dataset

## Data:

Input Data	Thousands of images of 10 different Classes
Response	AIRPLANE, AUTOMOBILE, BIRD, CAT, DEER, DOG, FROG, HORSE, SHIP, TRUCK

## Approach:

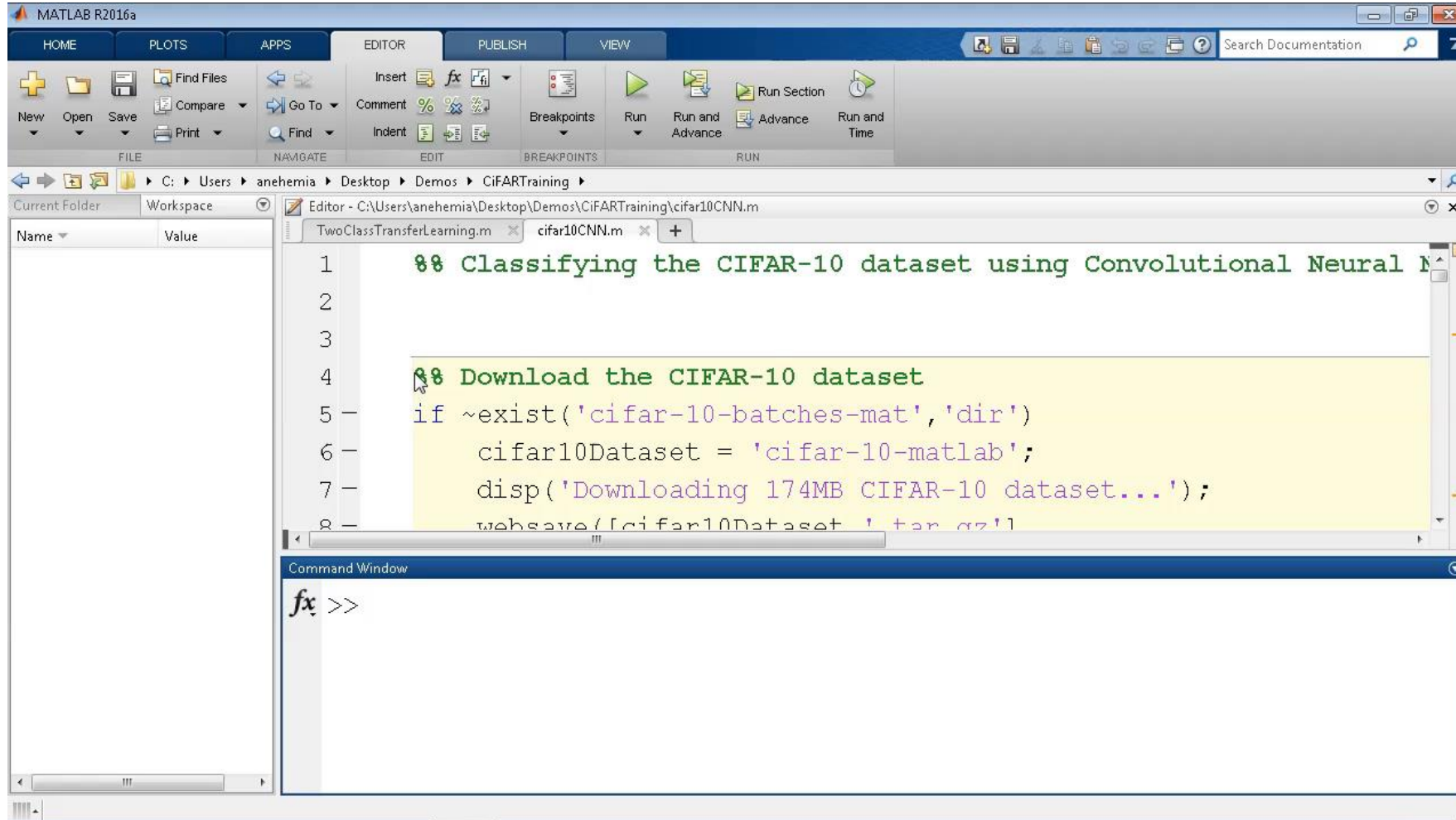
- Import the data
- Define an architecture
- Train and test the CNN



**Data Credit:** [Learning Multiple Layers of Features from Tiny Images](https://www.cs.toronto.edu/~kriz/tiny_images/), Alex Krizhevsky, 2009.

<https://www.cs.toronto.edu/~kriz/cifar.html>

# Demo: Classifying the CIFAR-10 dataset



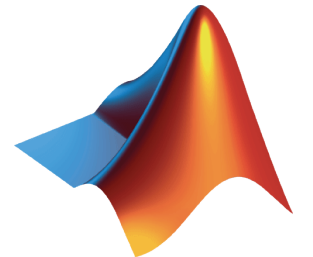
The image shows the MATLAB R2016a environment. The main window displays a script titled 'cifar10CNN.m' with the following code:

```

1  %% Classifying the CIFAR-10 dataset using Convolutional Neural N
2
3
4  %% Download the CIFAR-10 dataset
5  if ~exist('cifar-10-batches-mat','dir')
6      cifar10Dataset = 'cifar-10-matlab';
7      disp('Downloading 174MB CIFAR-10 dataset...');
8      websave([cifar10Dataset 'tar.gz']

```

The Command Window at the bottom shows the prompt `fx >>`.



# 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

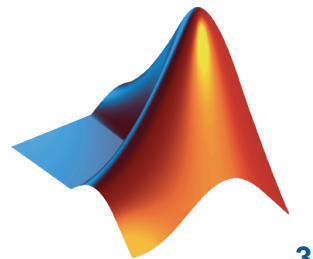


### Recommended when:

<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

# Demo

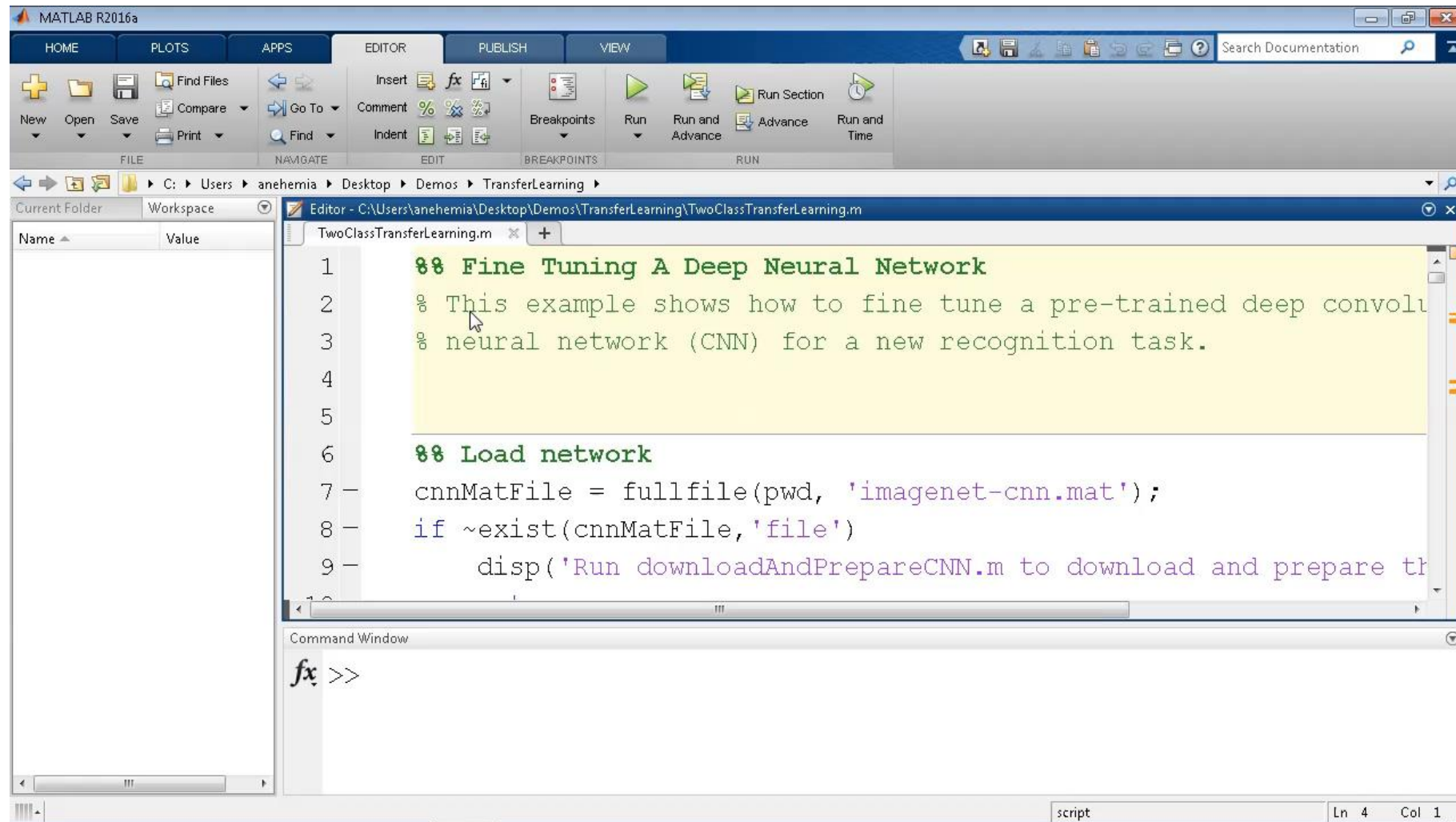
## *Fine-tune a pre-trained model ( transfer learning)*





# Demo

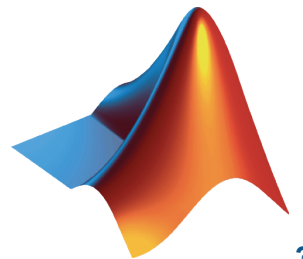
## *Fine-tune a pre-trained model ( transfer learning)*




The image shows the MATLAB R2016a software interface. The main window displays a script titled 'TwoClassTransferLearning.m' in the Editor. The script content is as follows:

```
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.

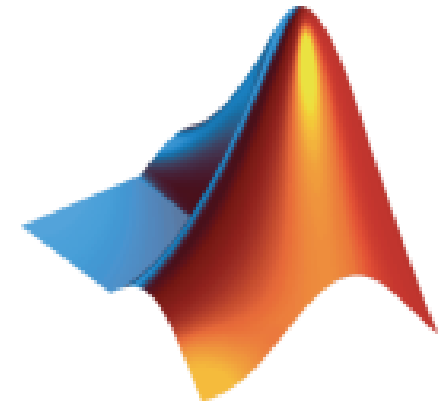


# Addressing Challenges in Deep Learning for Computer Vision

<b>Challenge</b>	<b>Solution</b>
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

## Key Takeaways

- Range of functionalities to support feature detection, 3D vision and camera calibration.
- Consider Deep Learning when:
  - Accuracy of traditional classifiers is not sufficient
    - *ImageNet classification problem*
  - You have a pre-trained network that can be fine-tuned
  - Too many image categories (100s – 1000s or more)
    - *Face recognition*
- Explore deep learning by building your own architecture in MATLAB



# Further Resources on our File Exchange

- <http://www.mathworks.com/matlabcentral/fileexchange/38310-deep-learning-toolbox>

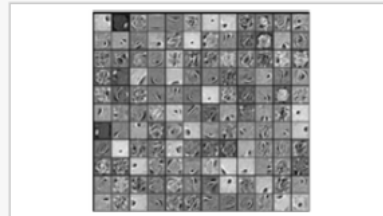


Search:

[Create Account](#) | [Log In](#)

[Products](#) [Solutions](#) [Academia](#) [Support](#) **[Community](#)** [Events](#) [Company](#)

## File Exchange



### Deep Learning Toolbox

by [Rasmus Berg Palm](#)  
24 Sep 2012 (Updated 02 Dec 2015)

Deep Belief Nets, Stacked Autoencoders, Convolutional Neural Nets and more. With examples.

 [Watch this File](#)



4.4 | 38 ratings  
[Rate this file](#)

807 Downloads (last 30 days)  
File Size: 16 MB  
File ID: #38310  
Version: 1.2

### Get from GitHub

 [Download Zip](#)

View license information and collaborate at [rasmusbergpalm/DeepLearnToolbox](#)

**Note:** Information from GitHub is updated daily.

**Thank You!**

Questions?