

MATLAB EXPO 2017

KOREA

4월 27일, 서울

등록 하기 matlabexpo.co.kr

컴퓨터비전의 최신기술

: Deep Learning, 3D Vision and Embedded Vision

김 종 남
Application Engineer

Three Main Topics

New capabilities for computer vision system design:

- Deep Learning
- 3-D Vision and Image Processing
- Embedded Vision



AlexNet
PRETRAINED MODEL
VGG-16
PRETRAINED MODEL
Caffe
MODELS

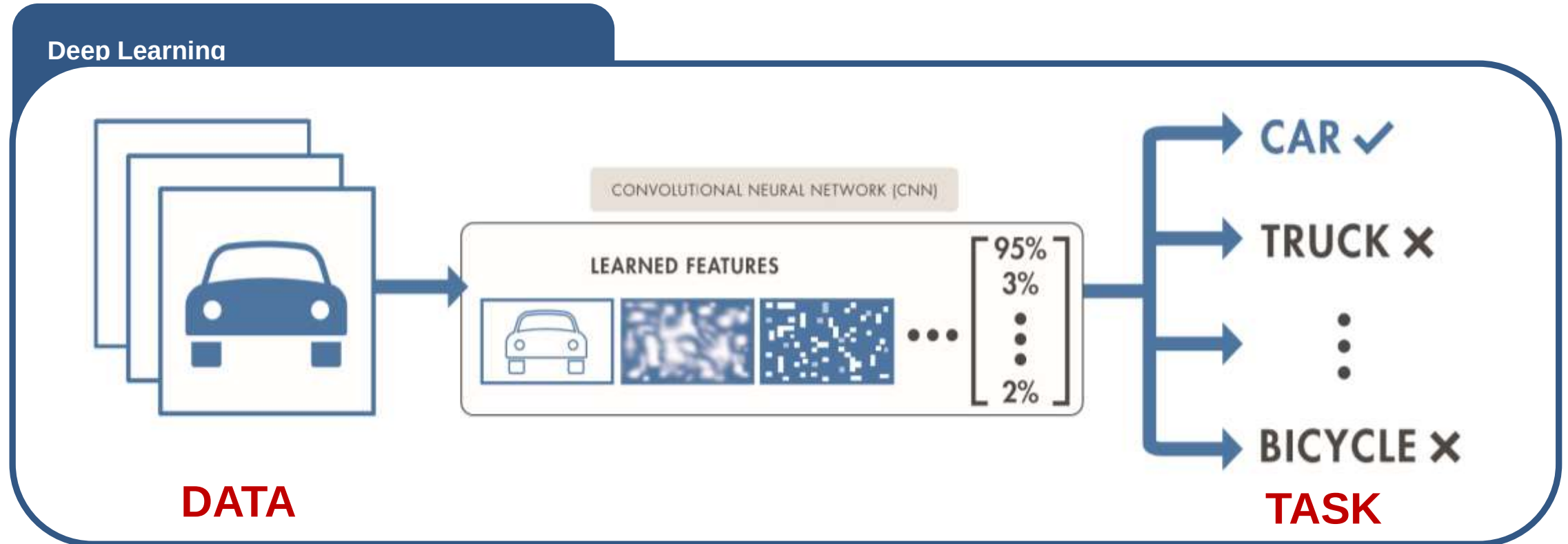
**New MATLAB framework makes deep learning easy
and accessible**

and

**MATLAB can be used by experts for real deep
learning(computer vision) problems**

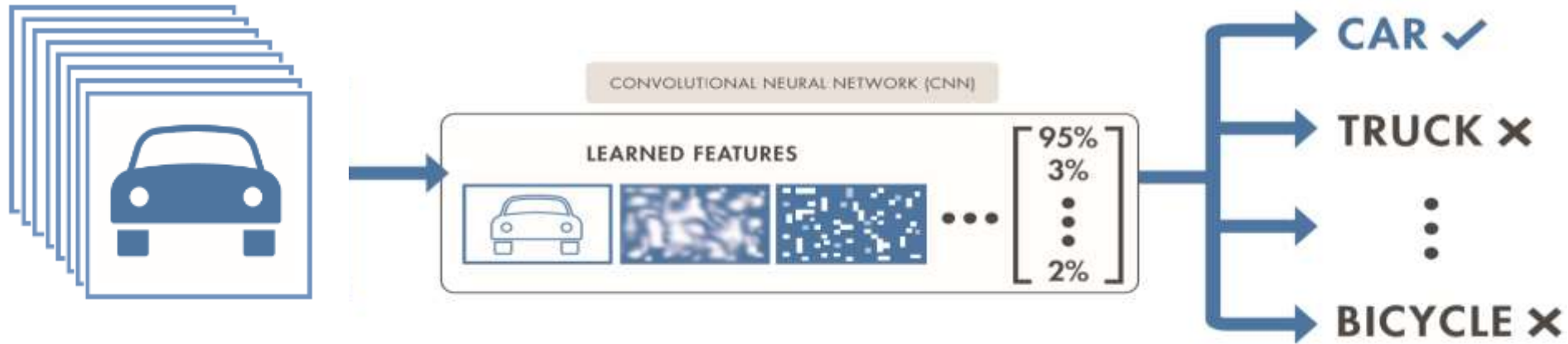
What is Deep Learning ?

Deep learning is a **type of machine learning** that performs **end-to-end learning** by learning **tasks** directly from **images, text, and sound**.

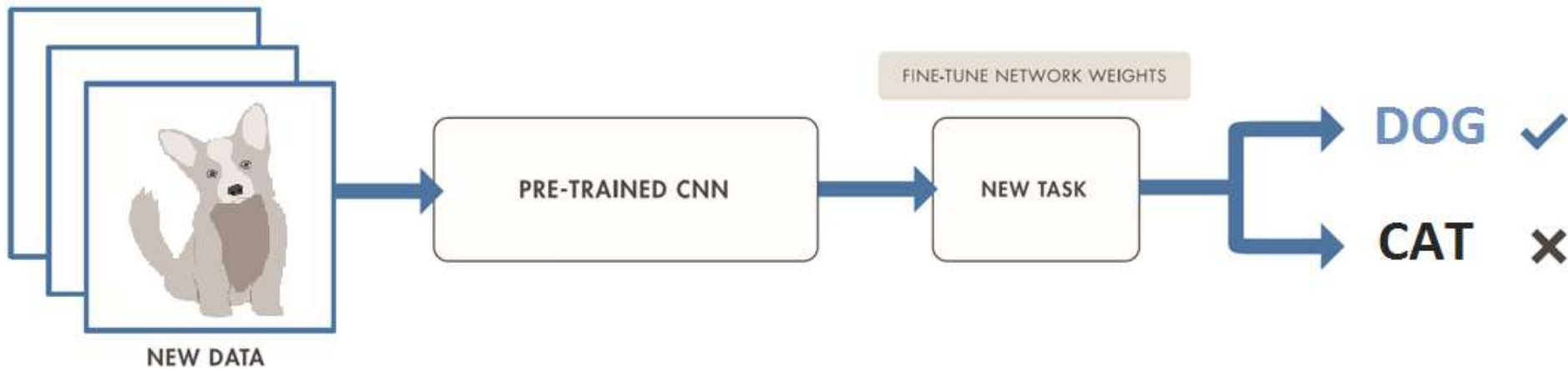


Two Approaches for Deep Learning

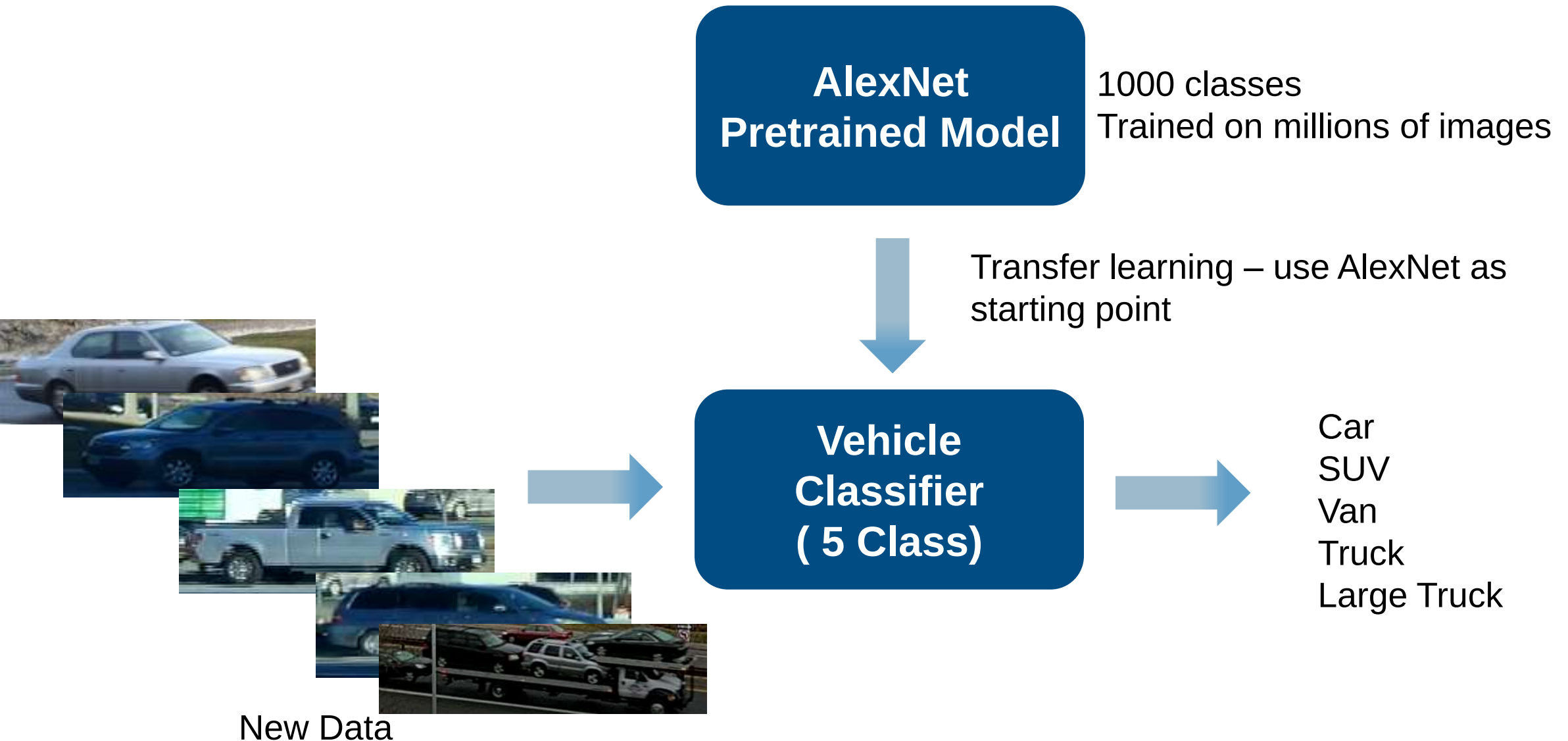
1. Train a Deep Neural Network from Scratch



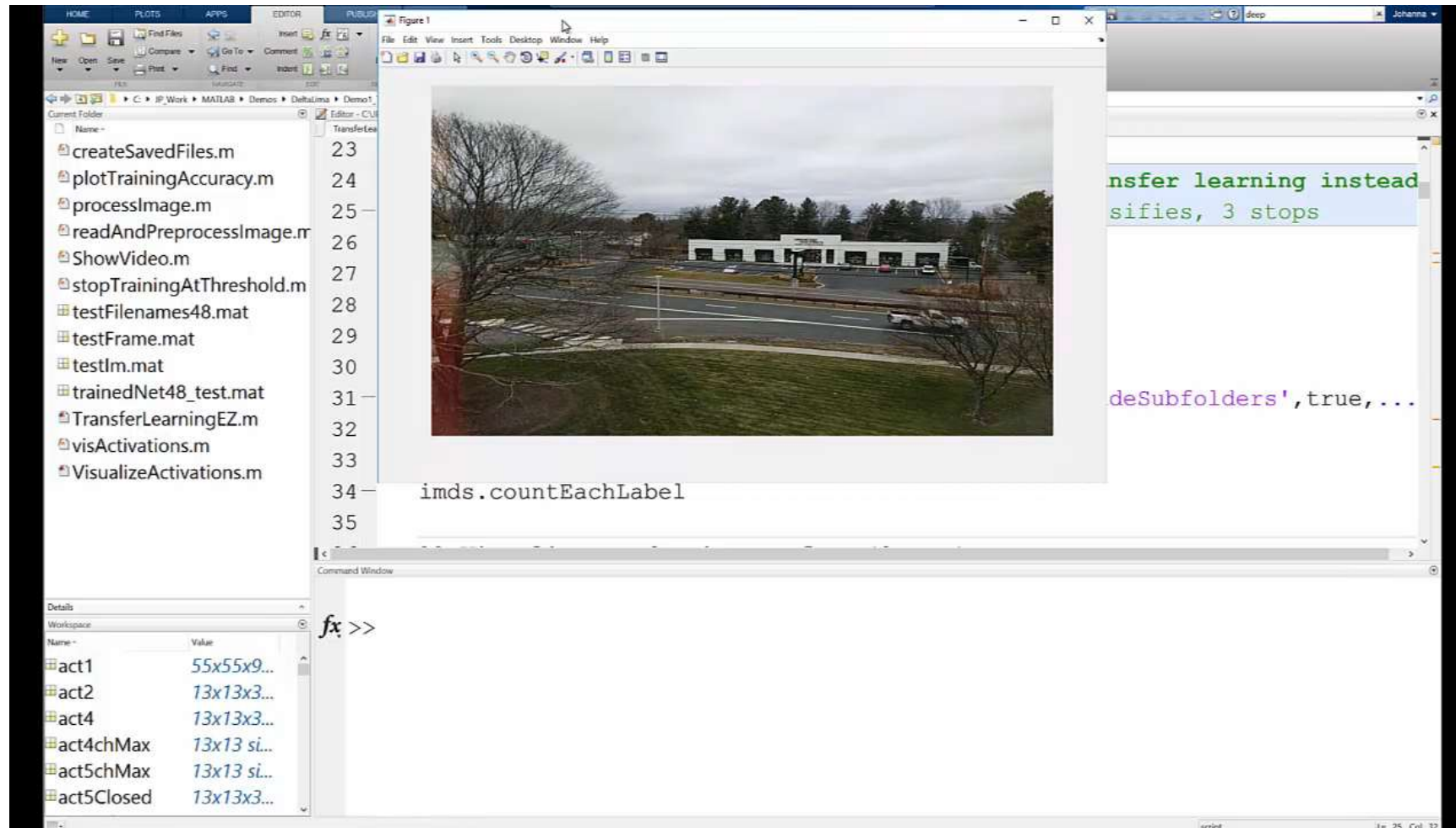
2. Fine-tune a pre-trained model (transfer learning)



Example: Classify Vehicles With Transfer Learning



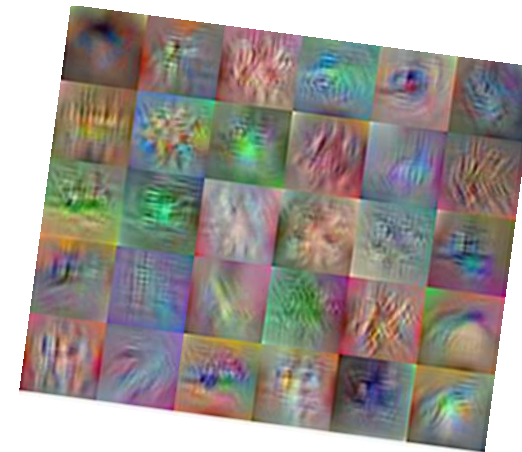
Transfer Learning to Classify New Objects



MATLAB makes Deep Learning Easy and Accessible

Learn about new MATLAB capabilities to

- Handle and label large sets of images
- Accelerate deep learning with GPU's
- Visualize and debug deep neural networks
- Access and use models from experts



AlexNet
PRETRAINED MODEL
Caffe MODELS
VGG-16
PRETRAINED MODEL

3D Image Processing

3-D Image Processing

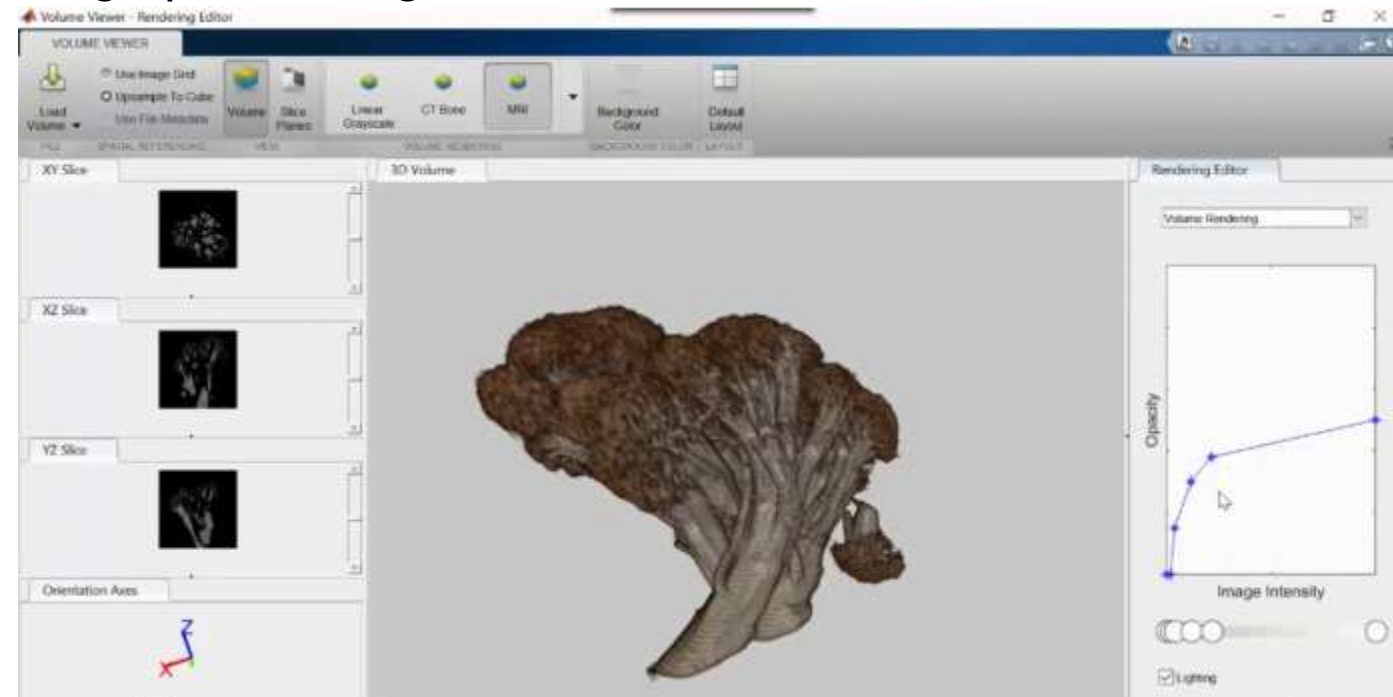
R2016b R2017a

Over 40 functions support 3-D volumetric image processing

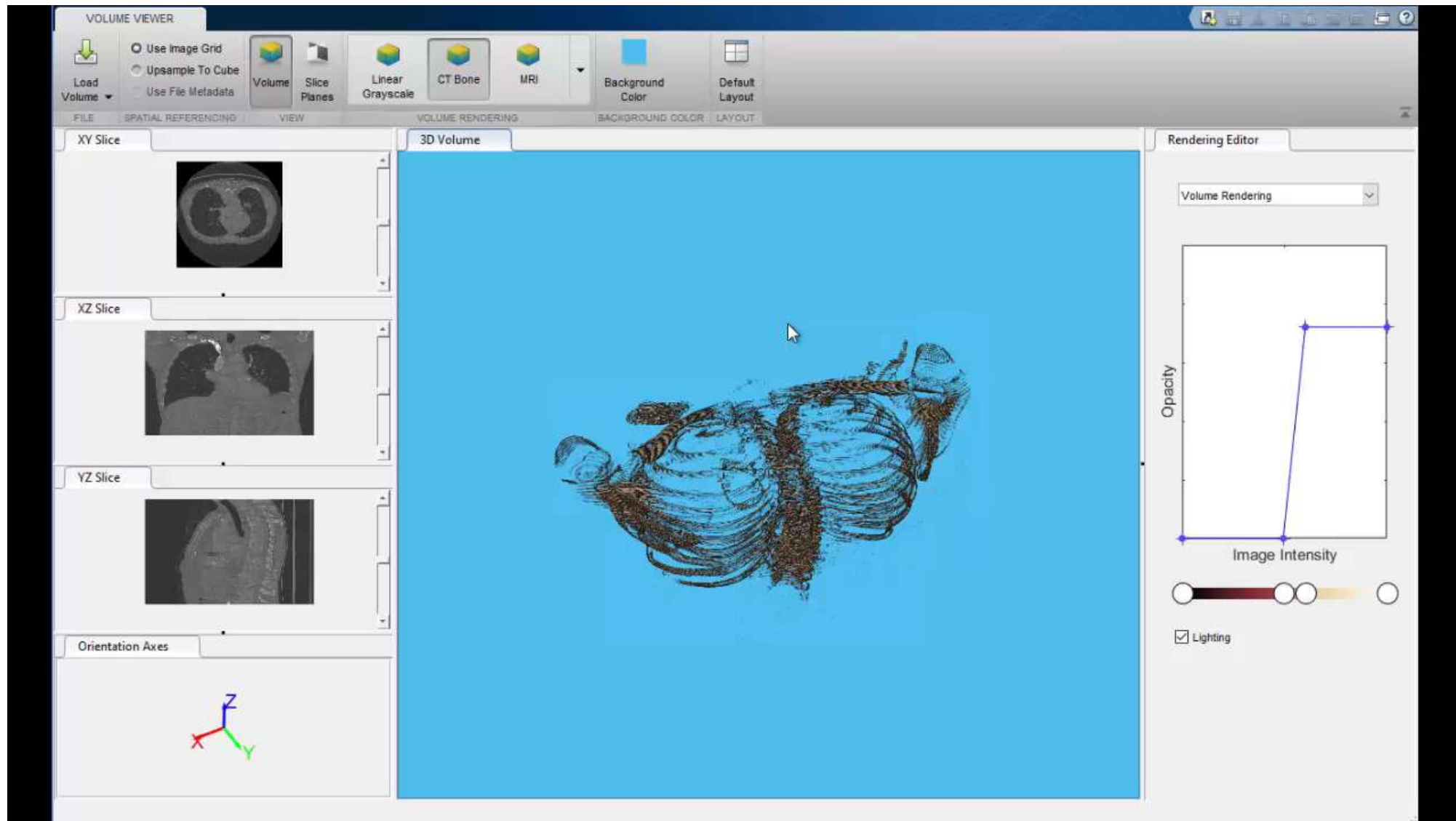
Capabilities Includes:

- Image arithmetic
- Morphology
- Segmentation
- Geometric transforms
- Enhancement

Volume Viewer App for exploration



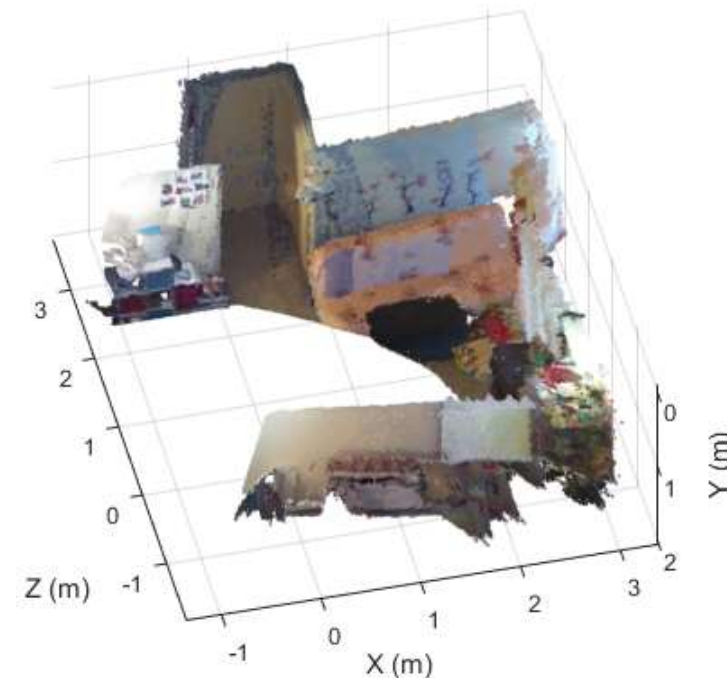
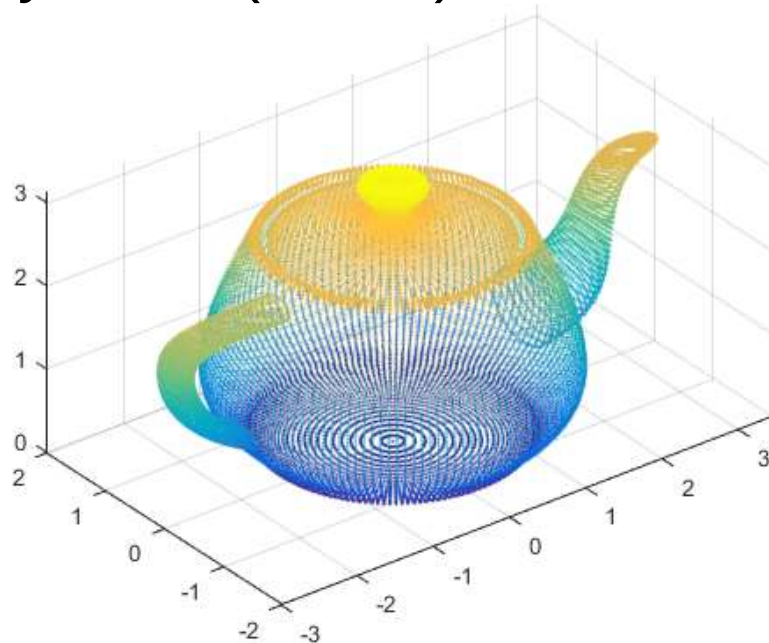
3-D Image Processing



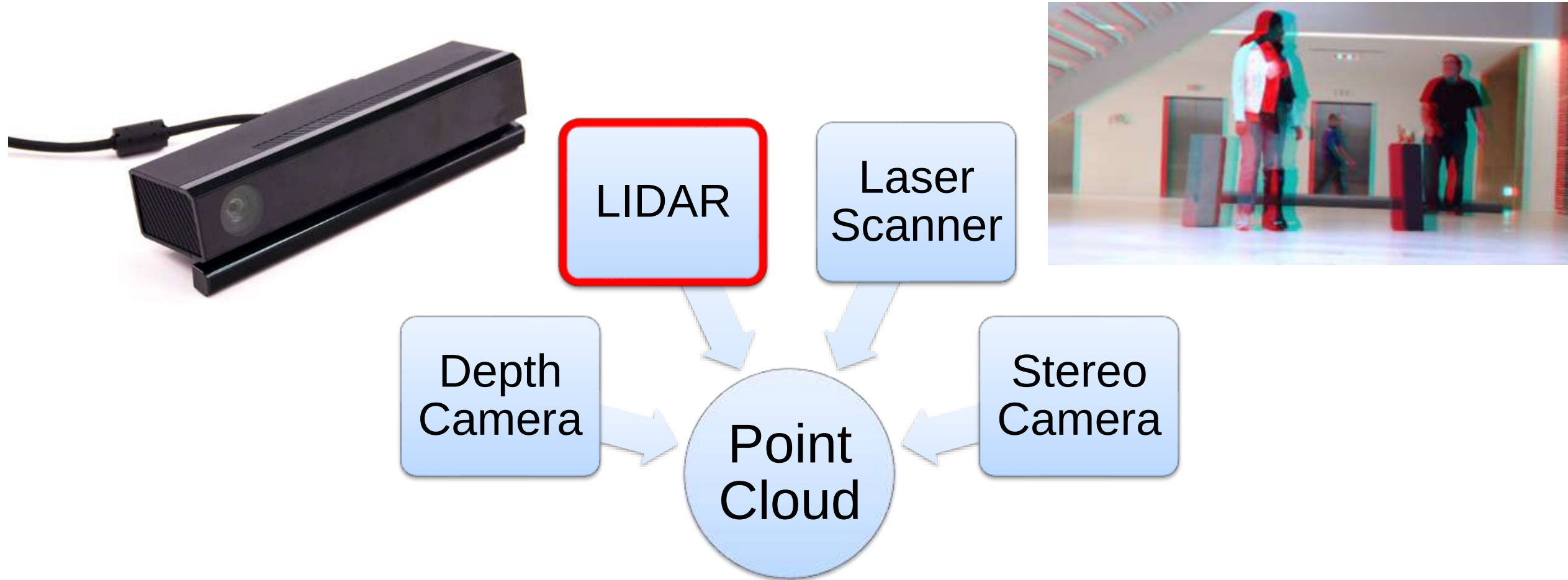
3D Vision – LiDAR Processing

What are Point Clouds?

- Point clouds represent a set of data points in a 3-D coordinate system
- Typically used to measure physical world surfaces
- Used for navigation and perception in robotics and Advanced Driver Assistance Systems (ADAS)

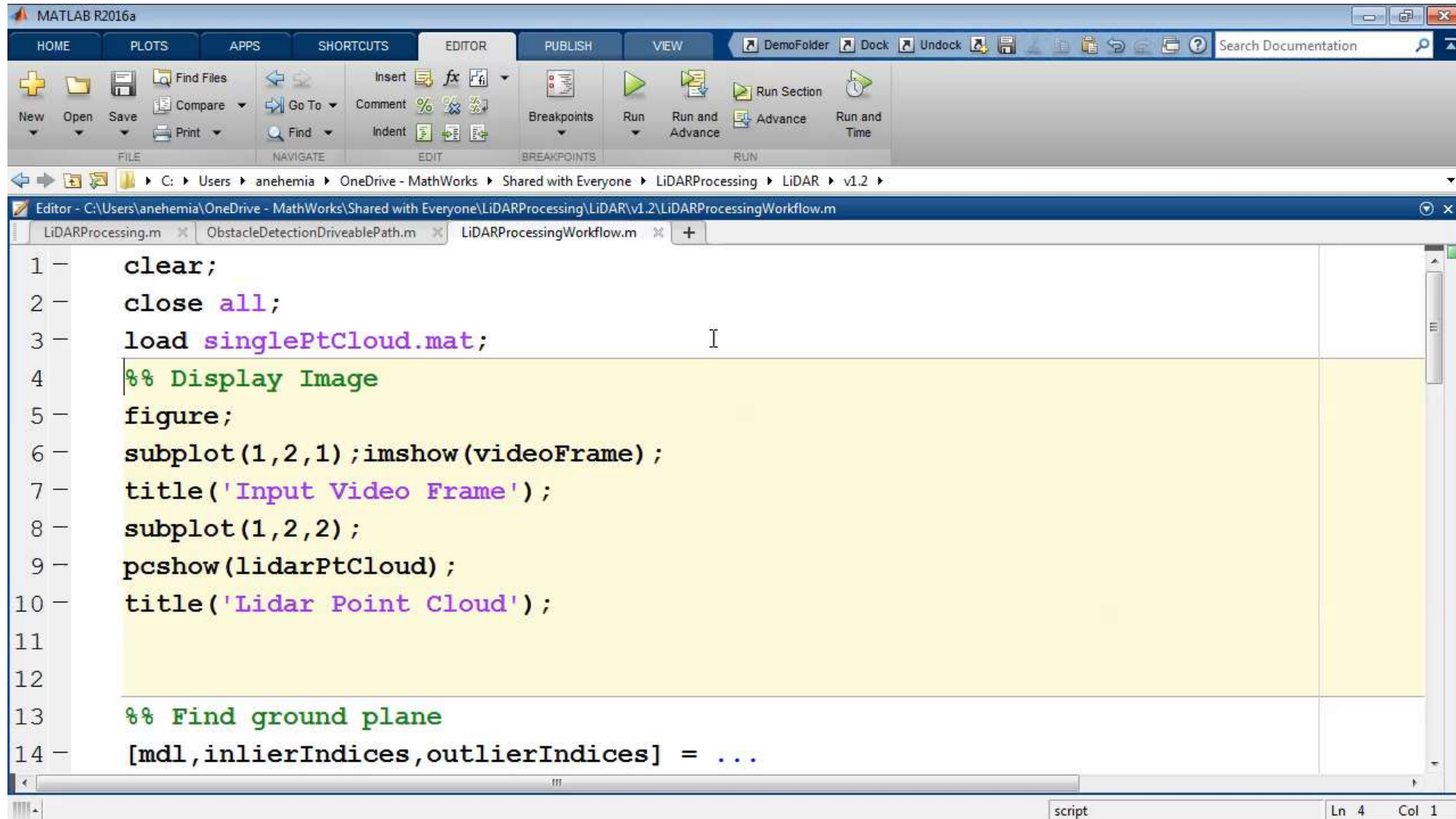


Common Sources of Point Cloud Data





3:D Vision: Design LiDAR Processing



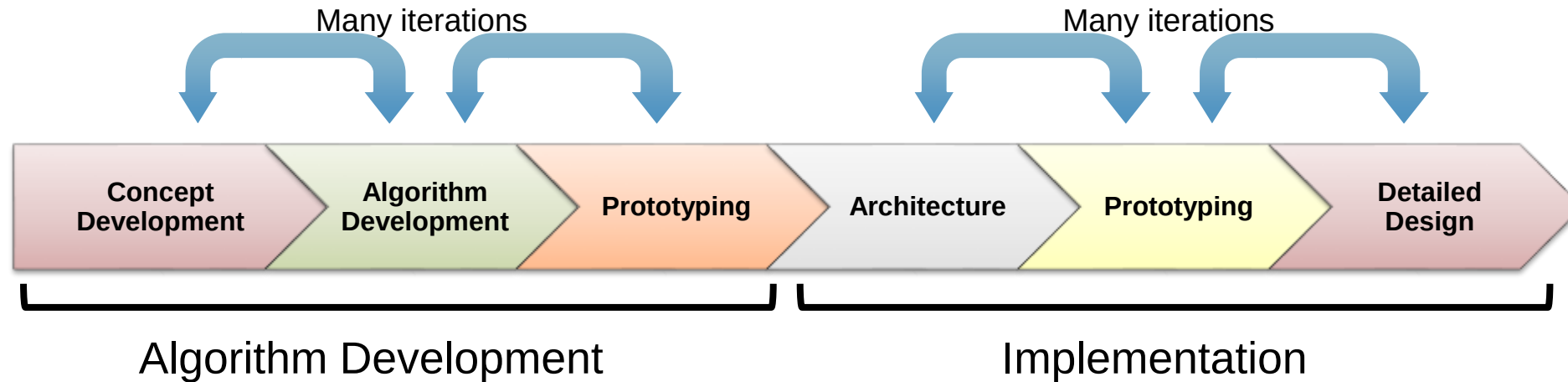
The image shows the MATLAB R2016a Editor window. The title bar indicates the version is R2016a. The interface includes a ribbon with tabs for HOME, PLOTS, APPS, SHORTCUTS, EDITOR, PUBLISH, and VIEW. The EDITOR tab is active, showing a script named 'LiDARProcessingWorkflow.m' located at 'C:\Users\anehemia\OneDrive - MathWorks\Shared with Everyone\LiDARProcessing\LiDAR\v1.2\LiDARProcessingWorkflow.m'. The script content is as follows:

```
1 clear;
2 close all;
3 load singlePtCloud.mat;
4 %% Display Image
5 figure;
6 subplot(1,2,1); imshow(videoFrame);
7 title('Input Video Frame');
8 subplot(1,2,2);
9 pcshow(lidarPtCloud);
10 title('Lidar Point Cloud');
11
12
13 %% Find ground plane
14 [mdl,inlierIndices,outlierIndices] = ...
```

The status bar at the bottom right shows 'script', 'Ln 4', and 'Col 1'.

Embedded Vision System Development using Automatic Code Generation

Typical Workflow for Embedded Vision System Development



- Is my idea new? What is required?
- Is it robust to all kinds of conditions? (lighting noise, etc.)



- Consideration of HW platform
 - FPGA? CPU? DSP? GPU?
- Speed and resource requirement
 - Resolution, Frame-rate constraint
 - Memory constraint

Development of the algorithm and implementation are often done by different groups

MATLAB Coder app with Integrated Editor and Simplified Workflow

R2015a

New user interface simplifies code generation workflow

The image displays two screenshots of the MATLAB Coder application interface for a project named 'kalmanfilter.prj'.

Left Screenshot: Finish Workflow

- Header:** Finish Workflow
- Status:** A green checkmark icon indicates 'Source Code Generated Successfully'. Below it, text states: 'You can now use the C code in your applications. [Learn more](#)'.
- Project Summary:**
 - Functions:** kalmanfilter.m
 - Project Type:** MATLAB Coder
 - Fixed-Point Conversion:** Disabled
 - Project File:** kalmanfilter.prj
- Generated Output:**
 - C Code:** C:\Work\coderdemo_kalman_filter2\codegen\lib\kalmanfilter
 - Example main Files:** C:\Work\coderdemo_kalman_filter2\codegen\lib\kalmanfilter\examples
 - Reports:** Code Generation Report
- Navigation:** A 'Back' button is at the bottom left.

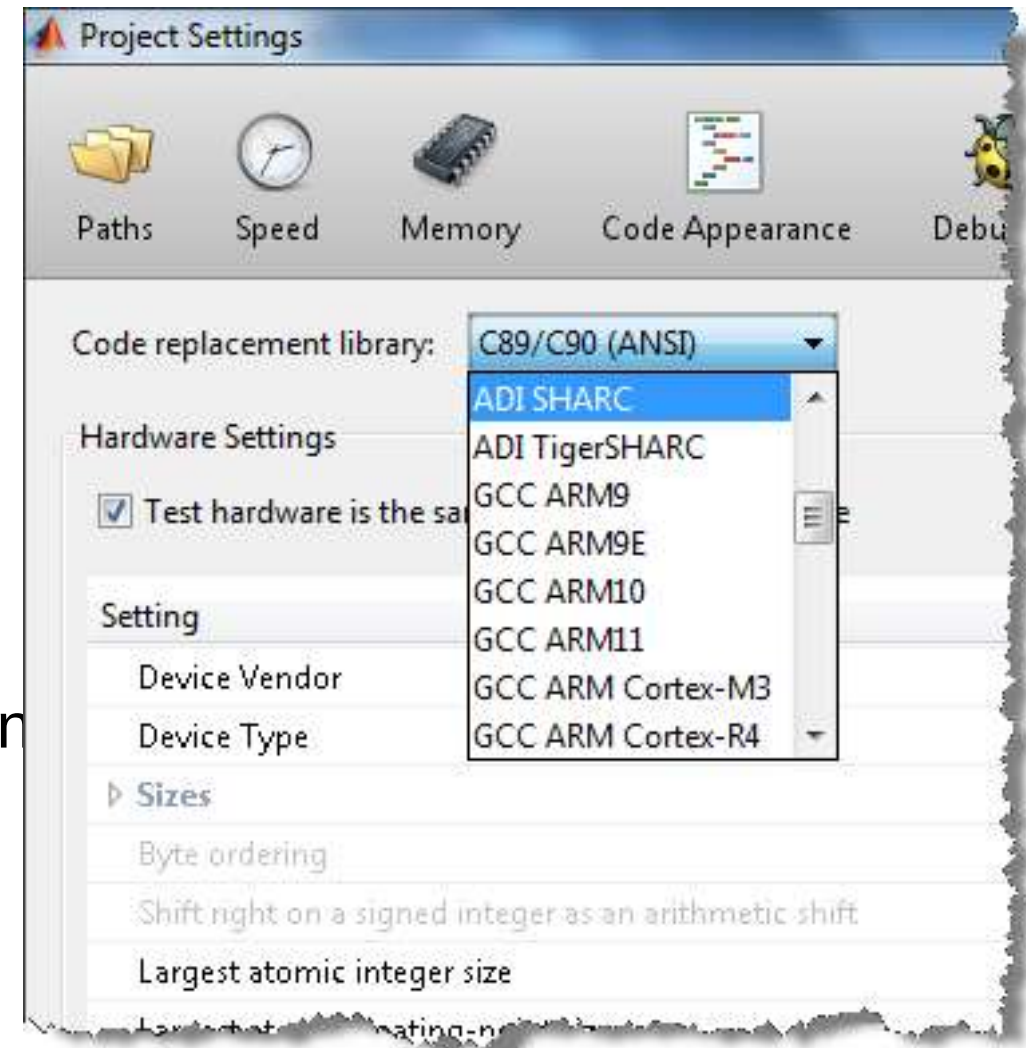
Right Screenshot: MATLAB Coder

- Header:** MATLAB Coder
- Workflow Steps:** A sequence of steps: Select, Define, Check, Generate, Finish. The 'Select' step is currently active.
- Fixed-Point Conversion:** A toggle switch labeled 'Fixed-Point Conversion' is currently set to 'Disabled'.
- Entry-Point Functions:**
 - A list box contains the function name 'kalmanfilter'.
 - A button '+ Add Entry-Point Function' is located below the list box.
- Project location:** A text field displays the path 'C:\Work\coderdemo_kalman_filter2\kalmanfilter.prj'.
- Navigation:** A 'Next' button is at the bottom right.

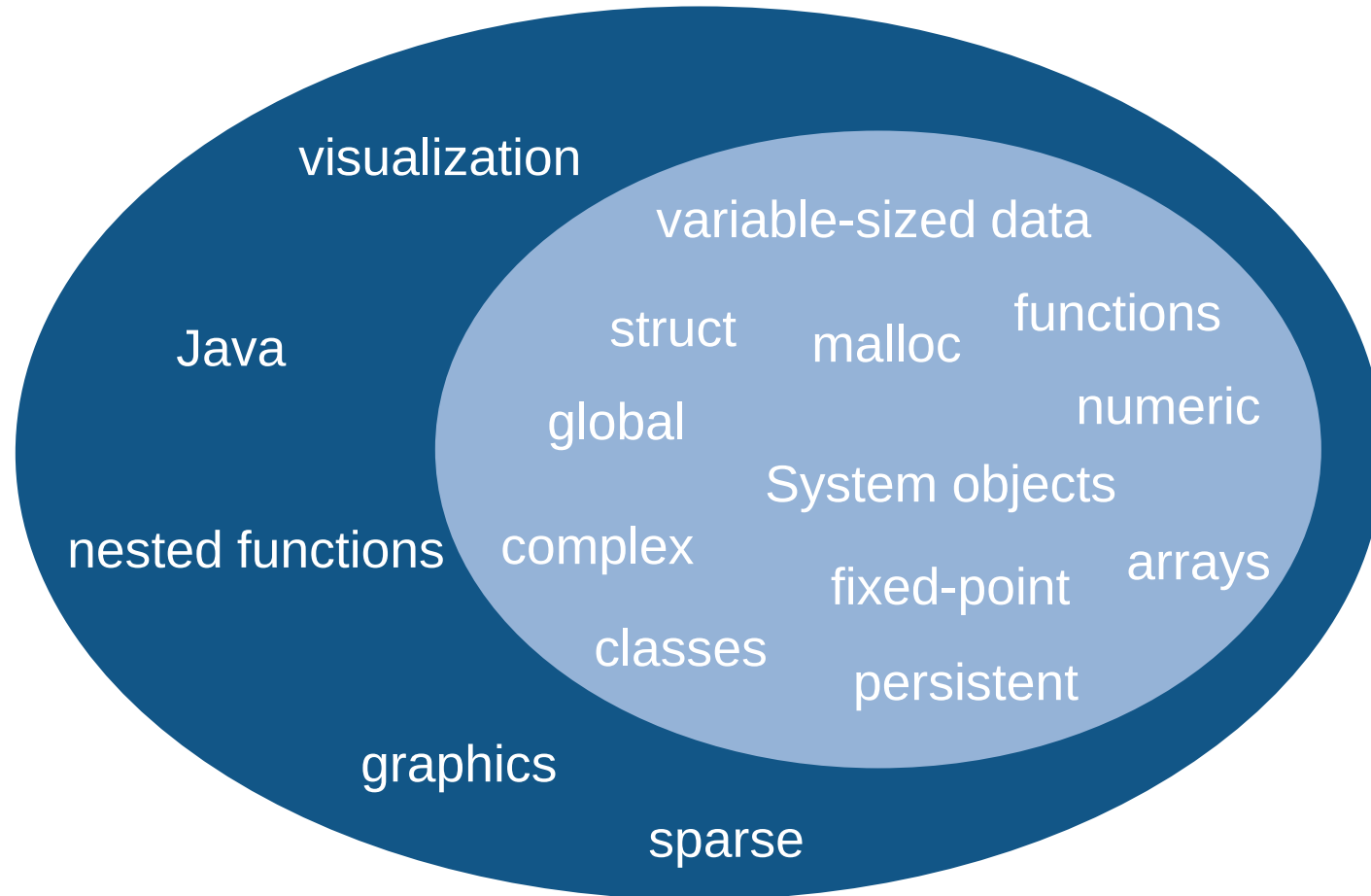
Embedded Coder for Optimized Code

Embedded Coder extends MATLAB Coder with

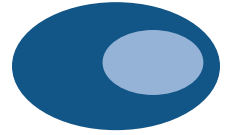
- Processor-specific code generation
 - Built-in support for select processors
 - Open APIs for use with any processor
- Speed, memory, and code appearance advanced features



MATLAB Language Support for Code Generation



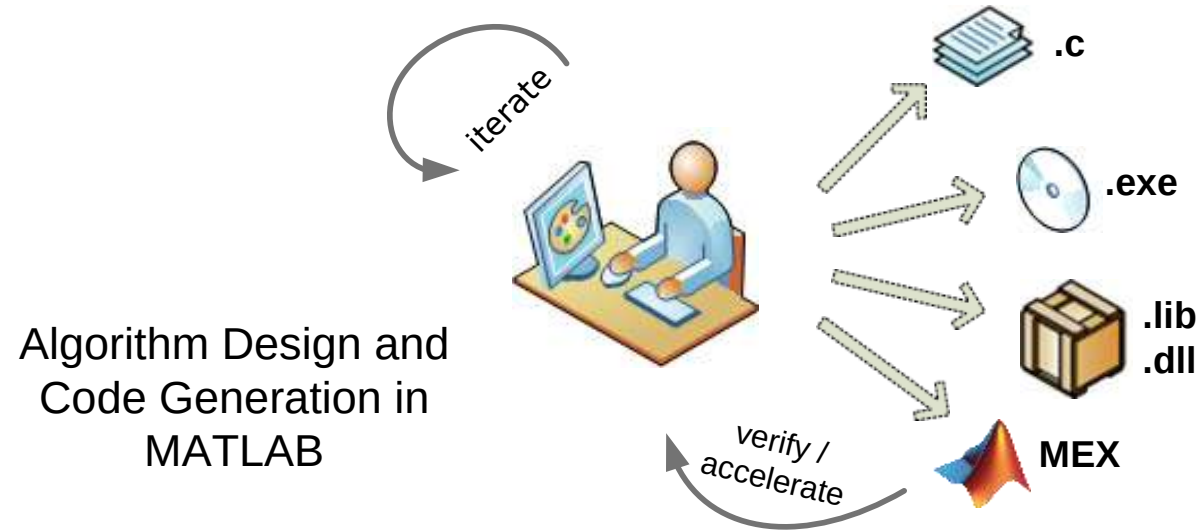
Supported MATLAB Language Features and Functions



Broad set of language features and functions/system objects supported for code generation

Matrices and Arrays	Data Types	Programming Constructs	Functions
• Matrix operations • N-dimensional arrays • Subscripting • Frames • Persistent variables • Global variables	• Complex numbers • Integer math • Double/single-precision • Fixed-point arithmetic • Characters • Structures • Cell arrays • Numeric class • Variable-sized data • MATLAB Class • System objects	• Arithmetic, relational, and logical operators • Program control (if, for, while, switch)	• MATLAB functions and subfunctions • Variable-length argument lists • Function handles Supported algorithms • More than 1300 MATLAB operators, functions, and System objects for: • Communications • Computer vision • Image processing • Neural networks • Phased Array signal processing • Robotics • Signal processing • Statistics and machine learning

Automatic Translation of MATLAB to C



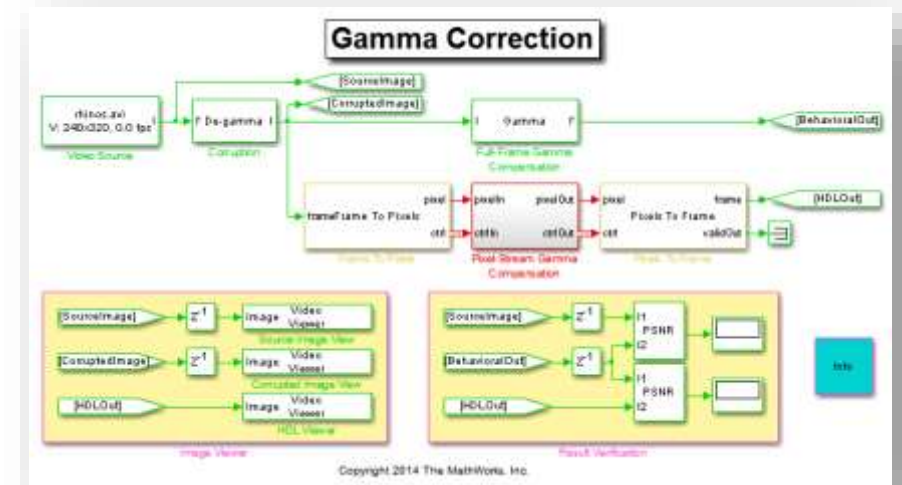
With MATLAB Coder, design engineers can:

- Maintain one design in MATLAB
- Design faster and get to C quickly
- Test more systematically and frequently
- Spend more time improving algorithms in MATLAB

Vision HDL Toolbox

Design and prototype video image processing systems

- Modeling hardware behavior of the algorithms
 - Pixel-based functions and blocks
 - Conversion between frames and pixels
 - Standard and custom frame sizes
- Prototyping algorithms on hardware
 - (With HDL Coder) Efficient and readable HDL code
 - (With HDL Verifier) FPGA-in-the-loop testing and acceleration



Pixel Based Video Image Algorithms

- **Analysis & Enhancement**

- Edge Detection, Median Filter

- **Conversions**

- Chroma Resampling, Color-Space Converter
- Demosaic Interpolator, Gamma Corrector, Look-up Table

- **Filters**

- Image Filter, Median Filter

- **Morphological Operations**

- Dilation, Erosion,
- Opening, Closing

- **Statistics**

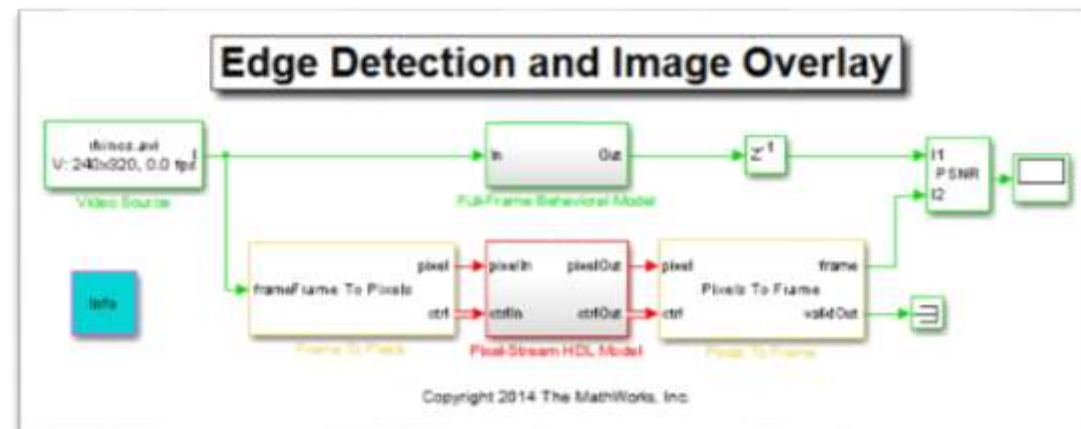
- Histogram
- Image Statistics

- **I/O Interfaces**

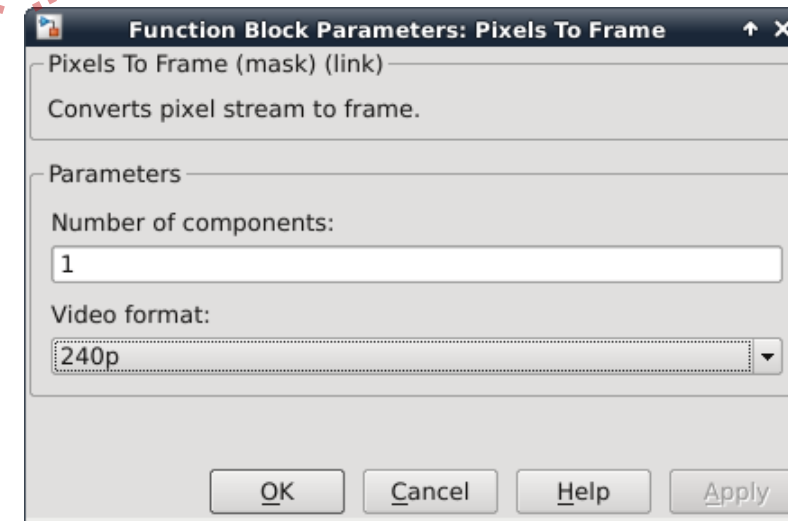
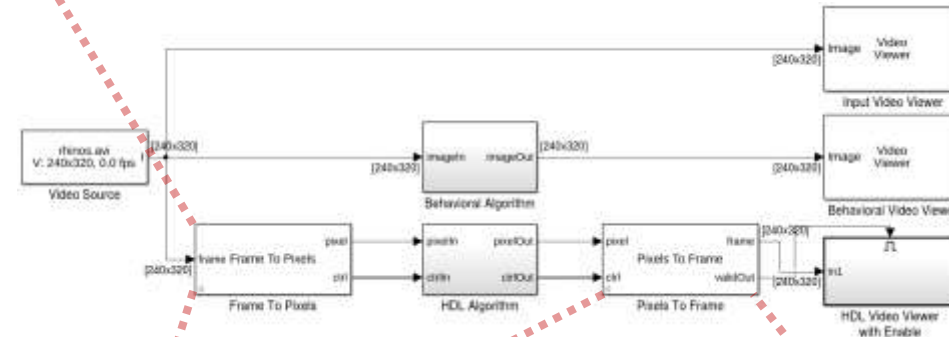
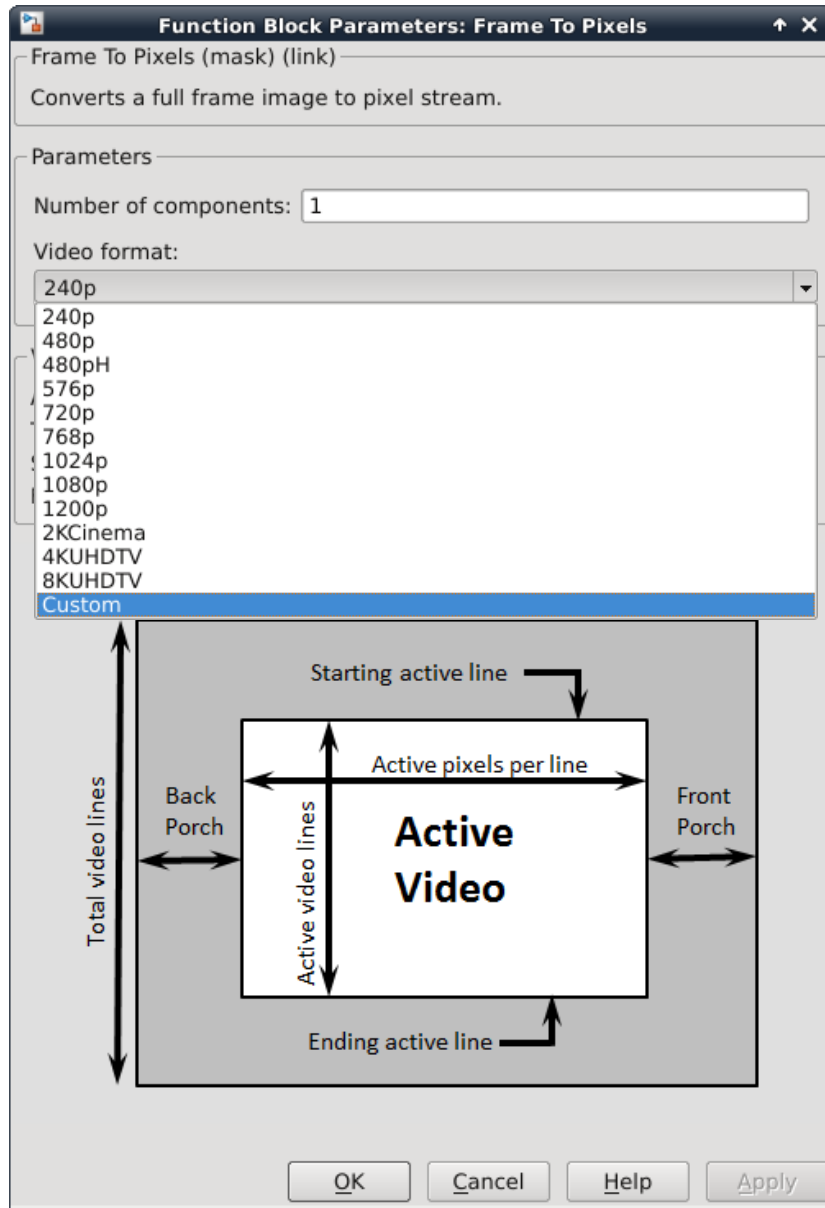
- Frame to Pixels, Pixels to Frame, FIL versions

- **Utilities**

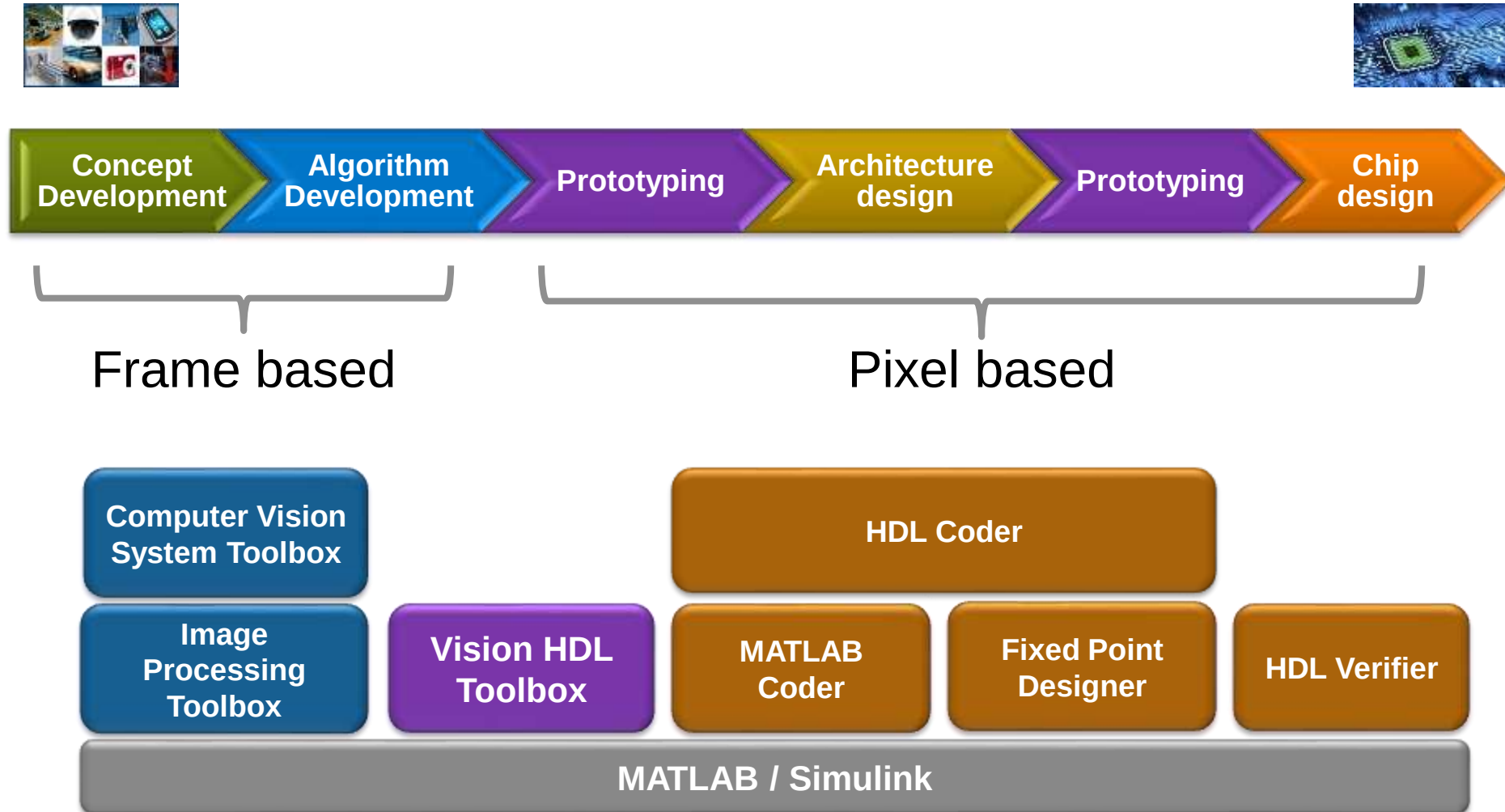
- Pixel Control Bus Creator
- Pixel Control Bus Selector



Frame To Pixels and Pixels To Frame



A Complete Solution for Embedded Vision



FLIR Accelerates Development of Thermal Imaging FPGA

Challenge

Accelerate the implementation of advanced thermal imaging filters and algorithms on FPGA hardware

Solution

Use MATLAB to develop, simulate, and evaluate algorithms, and use HDL Coder to implement the best algorithms on FPGAs

Results

- Time from concept to field-testable prototype reduced by 60%
- Enhancements completed in hours, not weeks
- Code reuse increased from zero to 30%



Raw image (left) and image after applying filter developed with HDL Coder (right).

“With MATLAB and HDL Coder we are much more responsive to marketplace needs. We now embrace change, because we can take a new idea to a real-time-capable hardware prototype in just a few weeks. There is more joy in engineering, so we’ve increased job satisfaction as well as customer satisfaction.”

Nicholas Hogasten
FLIR Systems