
MATLAB EXPO 2018
KOREA

MATLAB EXPO 2018

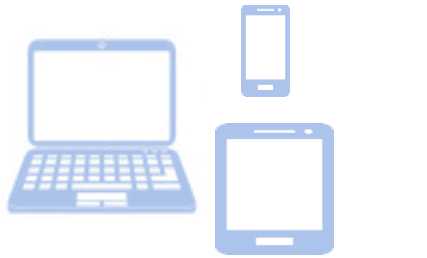
딥러닝을 활용한 영상 인식 응용프로그램 개발 워크플로우

송완빈
Application Engineer
MathWorks Korea

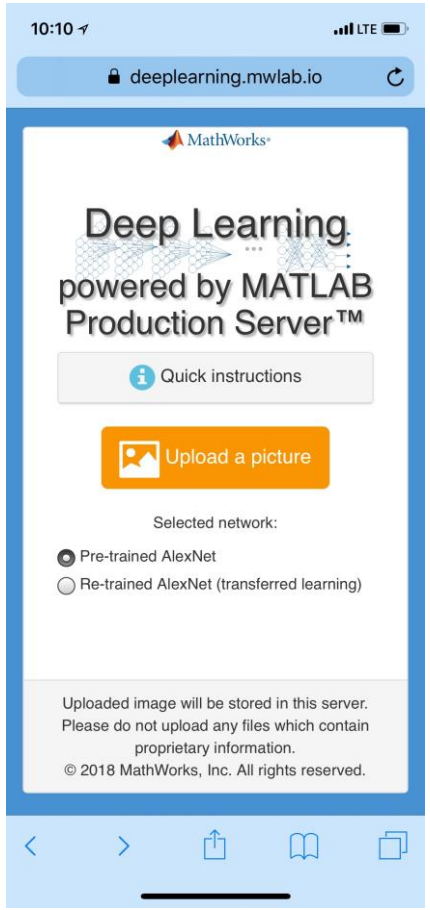
Agenda

- Deep Learning Application for object recognition
- What is Deep Learning?
- Object Recognition using Deep Learning
  - Convolutional Neural Network
  - Regions with Convolutional Neural Network
  - Semantic Segmentation
- Key Takeaways

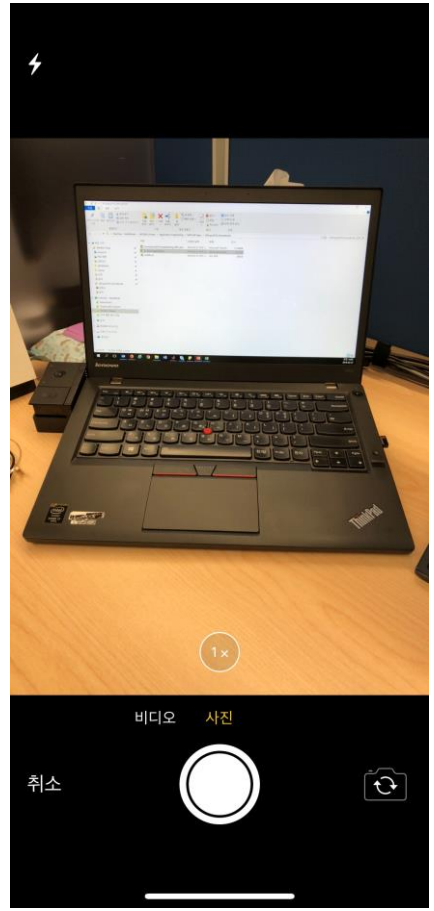
Smart Phone Application with Deep Learning



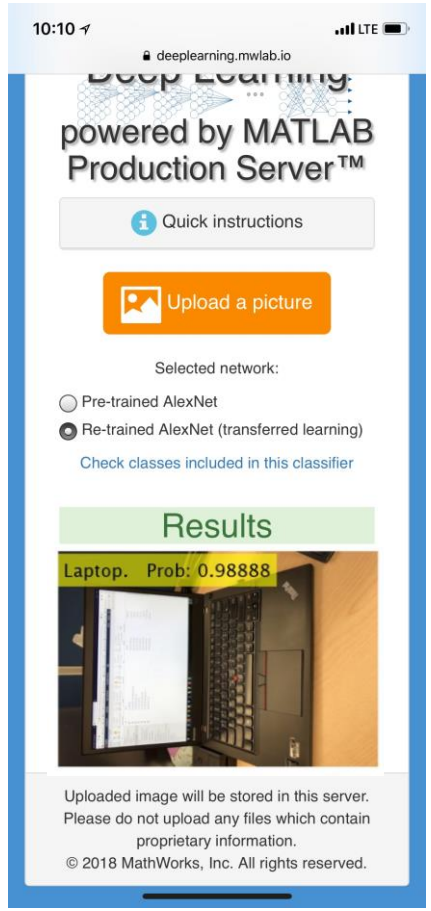
Users with Mobile Devices



Mobile Internet Browser

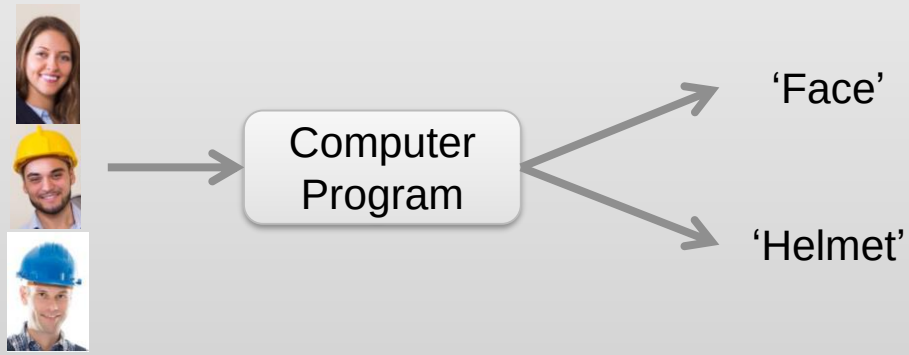


Mobile Camera



What is a scene where machine learning should be used?

Image processing algorithm



Hand Written Program

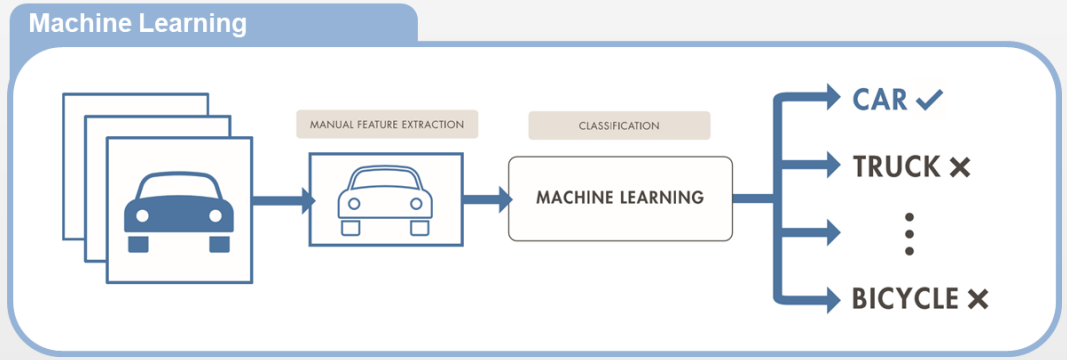
If **brightness** > 0.5
then 'Helmet'
If (**edge_density** < 4 and **major_axis** > 5)
then …

Formula or Equation

$$Y_{Classify} = \beta_1 X_{\text{edge}} + \beta_2 Y_{bright} + \beta_3 Z_{intensity} + \dots$$

- Specify condition by numerical value and carve
- When clear division is possible

Machine Learning Approach



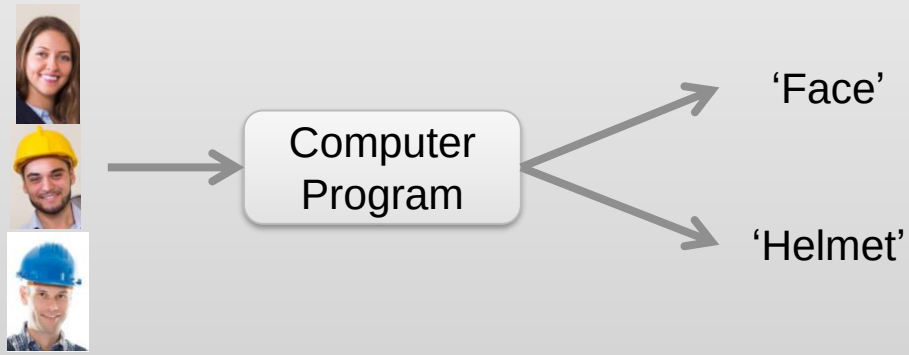
$$model : \text{Inputs} \rightarrow \text{Outputs}$$

$$model = < \textit{Machine Learning Algorithm} > (\text{images}, \text{label})$$

- Learn classifiers using image data
- Flexible separation can be done if it works

What is a scene where machine learning should be used?

Image processing algorithm



Hand Written Program

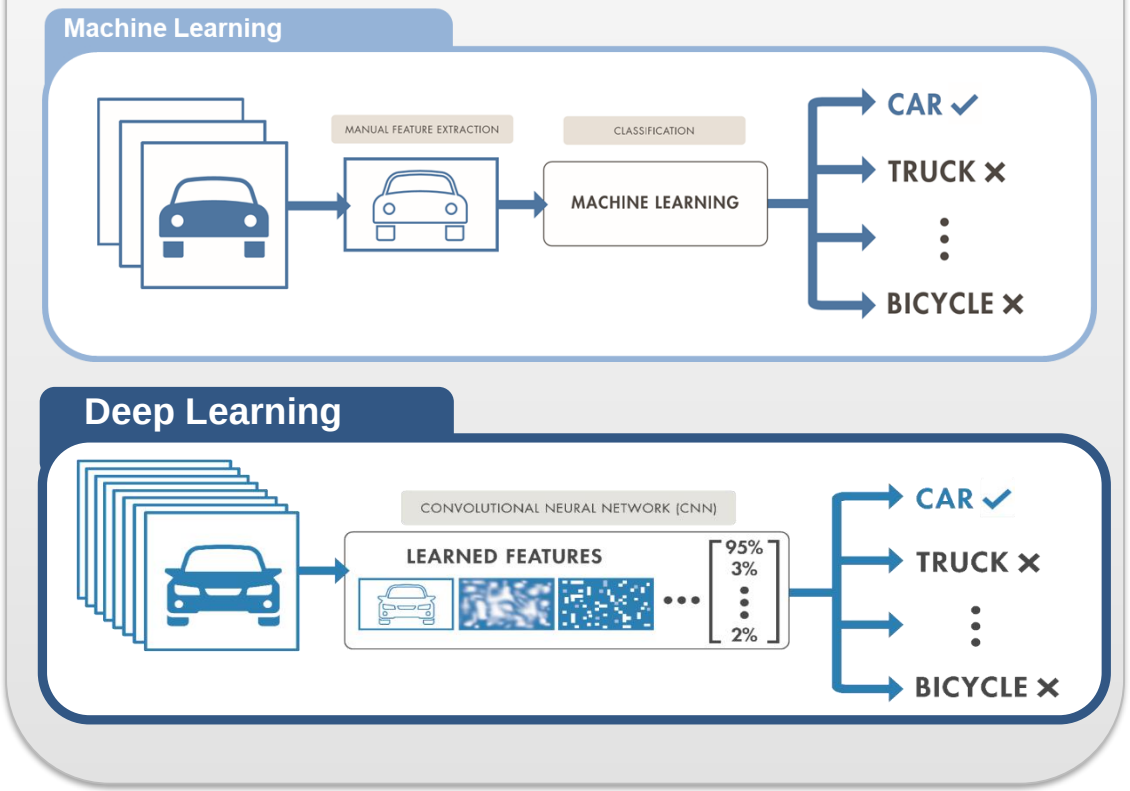
If **brightness** > 0.5
then 'Helmet'
If (**edge_density** < 4 and **major_axis** > 5)
then …

Formula or Equation

$$Y_{Classify} = \beta_1 X_{\text{edge}} + \beta_2 Y_{bright} + \beta_3 Z_{intensity} + \dots$$

- Specify condition by numerical value and carve
- When clear division is possible

Machine Learning Approach

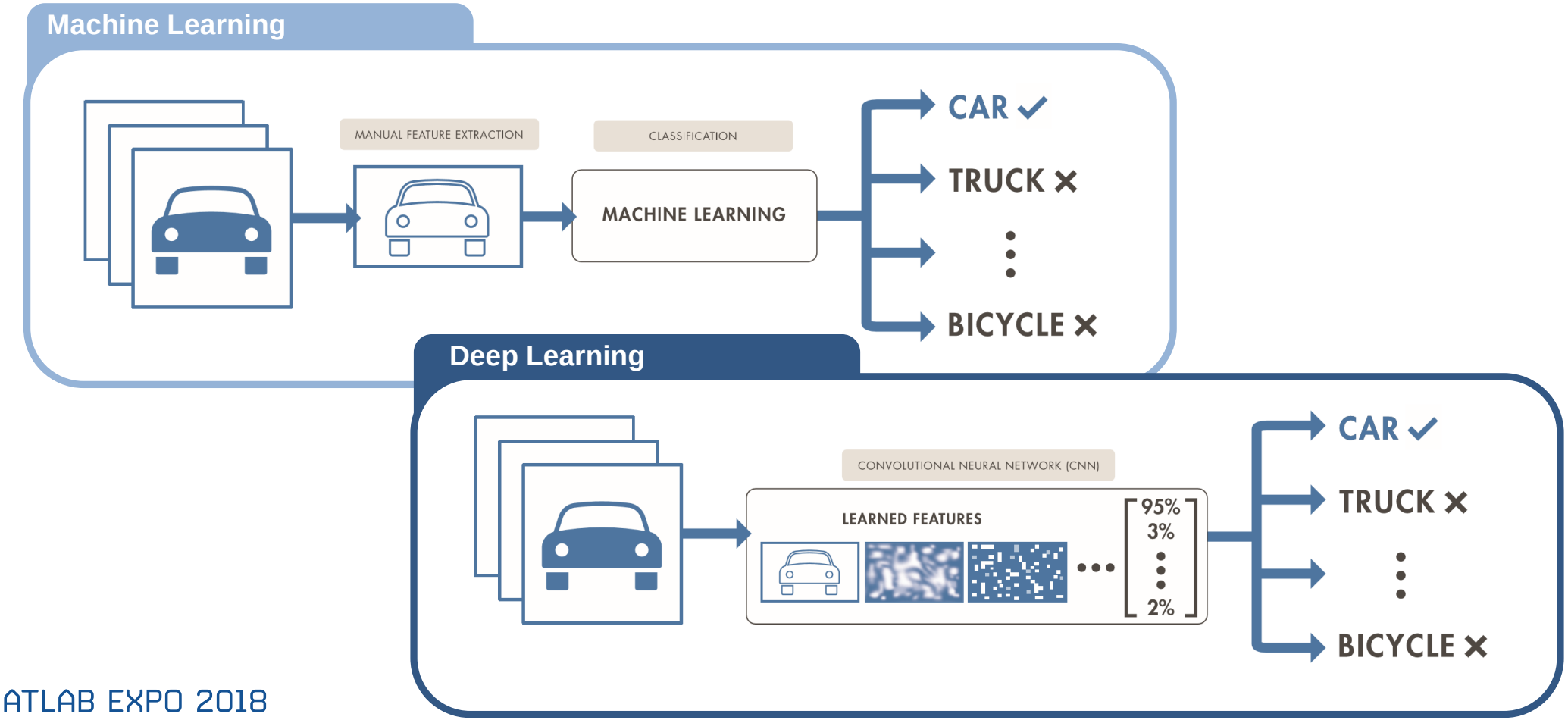


- Learn classifiers using image data
- Flexible separation can be done if it works

Machine learning vs Deep learning

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

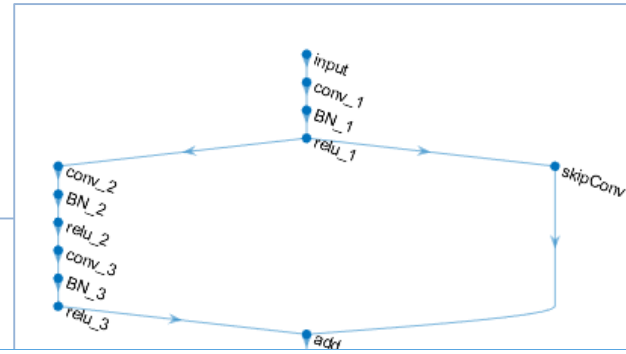
Deep learning algorithms also **scale with data –** traditional machine learning **saturates**

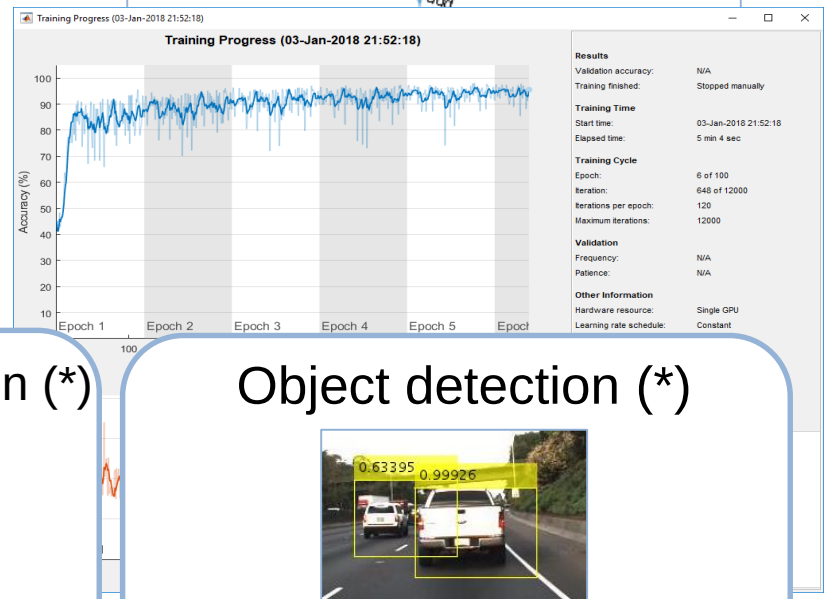


MATLAB deep network in a nutshell

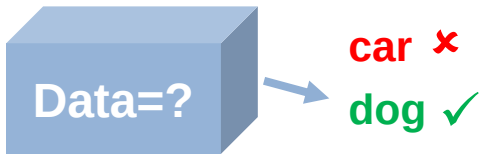
- A MATLAB deep network (**) is a MATLAB object that contains an array of trained layer objects.
- Layers array can be created, imported, edited, plotted in MATLAB
- Layers are trained with a lot of data and `net = trainNetwork(…,layers)`, most of the time.
- MATLAB deep networks have different usages.

```
layers = [ ...
    imageInputLayer([28 28 1])
    convolution2dLayer(12,25)
```



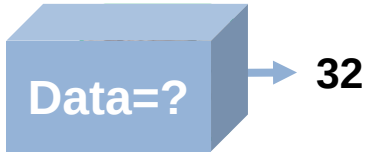


Classification




```
YPred = classify(net,X);
```

Regression




```
YPred = predict(net,X);
```

Semantic segmentation (*)


```
C = semanticseg(I,net);
```

Object detection (*)

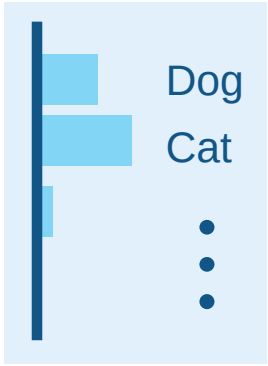

```
[bboxes,scores] = detect(net,I);
```


(**) Neural Network Toolbox

(*) Computer Vision Syst. Toolbox

Object Recognition using Deep Learning

Object recognition (whole image)	Object detection and recognition	Object recognition (in pixels)
CNN (Convolutional Neural Network)	**R-CNN / Fast R-CNN / Faster R-CNN**	**SegNet / FCN**



Image

Probability

Front of Car

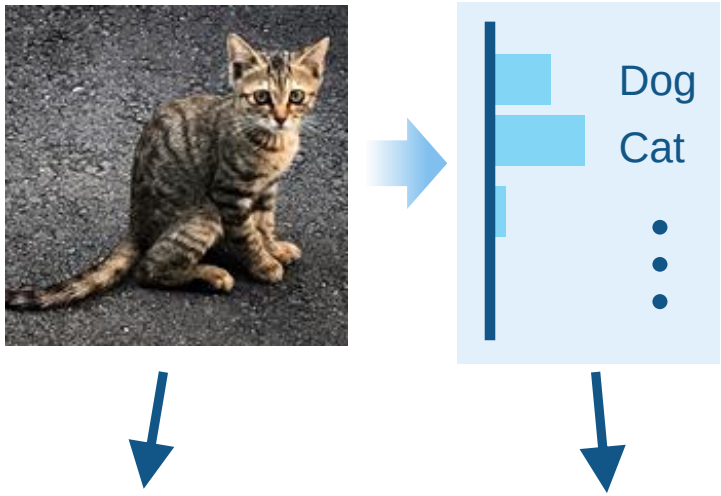
Stop Sign

Road

Vehicle

Object Recognition using Deep Learning

Object recognition (whole image)	Object detection and recognition	Object recognition (in pixels)
CNN (Convolutional Neural Network)	**R-CNN / Fast R-CNN / Faster R-CNN**	**SegNet / FCN**



Image

Probability

Front of Car

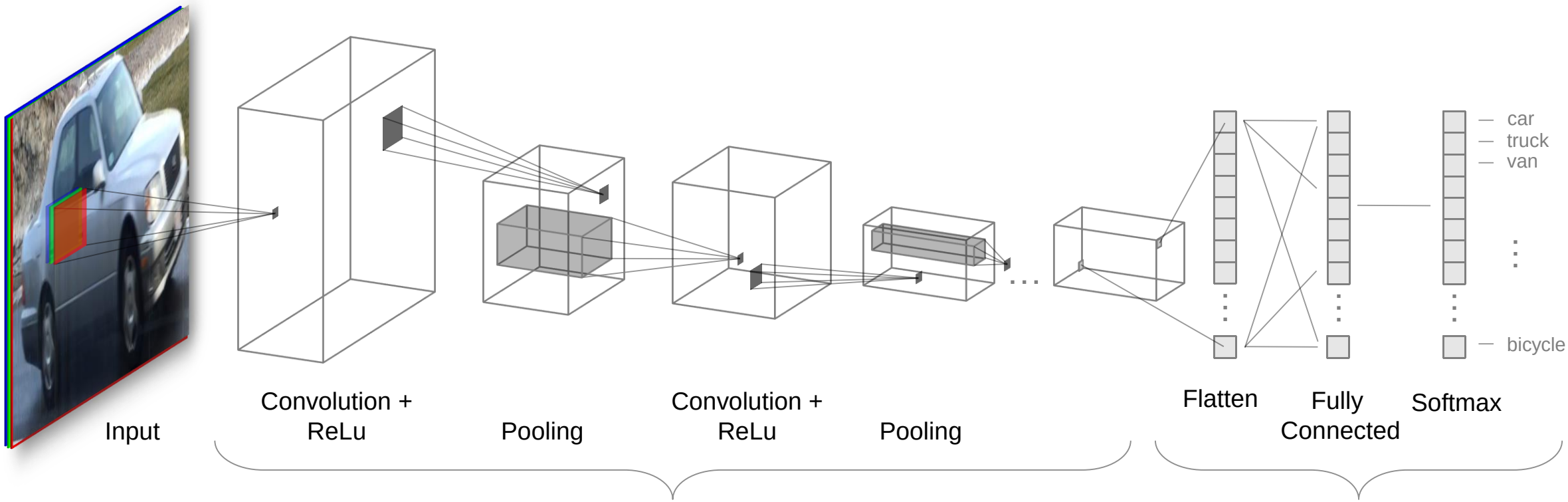
Stop Sign

Road

Vehicle

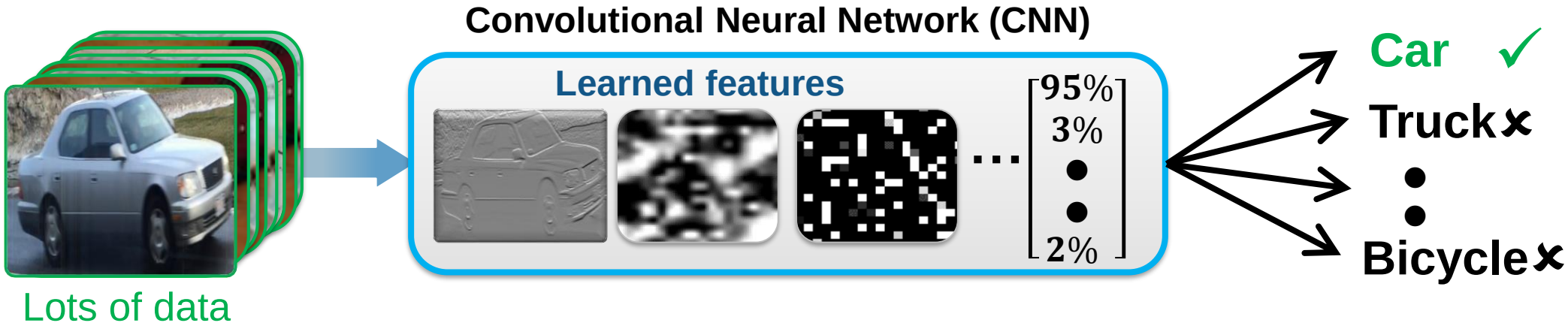
Convolutional Neural Networks

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

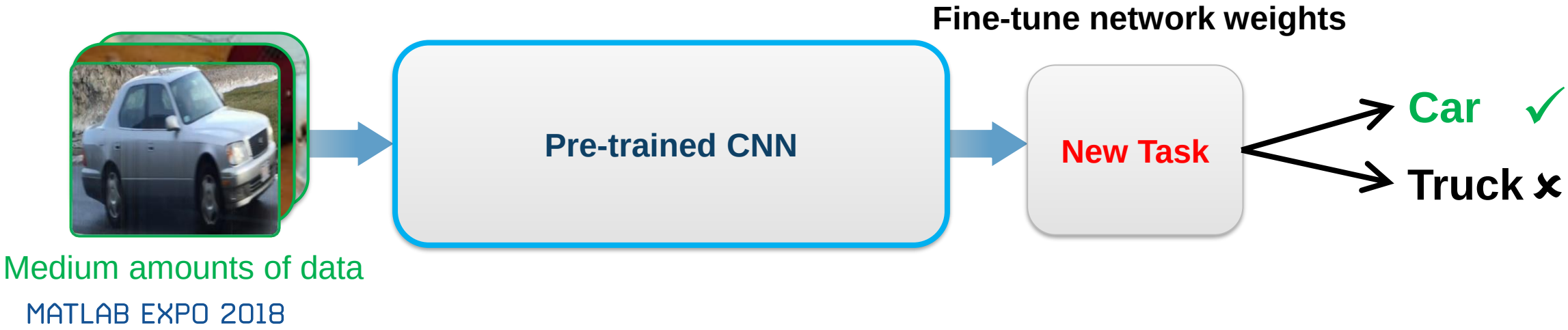


Two approaches for Deep learning

Approach 1. Train a Deep Neural Network from Scratch



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



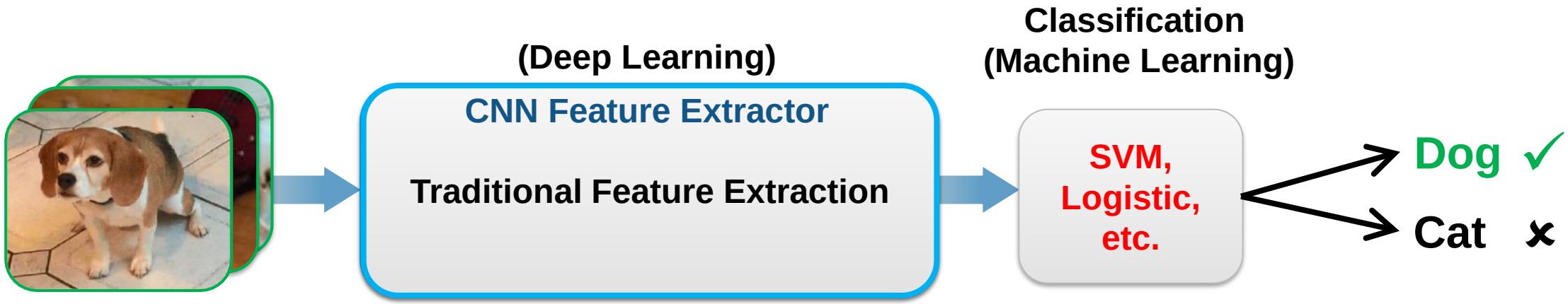
Another Deep Learning Workflows: Feature Extraction

Use a pretrained CNN as an automatic feature extractor

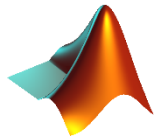
Recommended when:

the accuracy is not high enough using transfer learning

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



Deep Learning Application Approaches

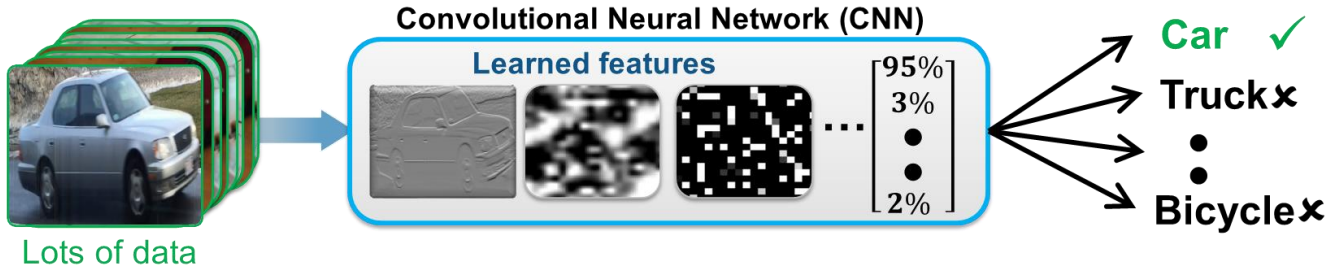


Train a Deep Neural Network from Scratch

Fine-tune a pre-trained model (Transfer learning)

Use a pretrained CNN as an automatic feature extractor

Deep Learning Application Approaches

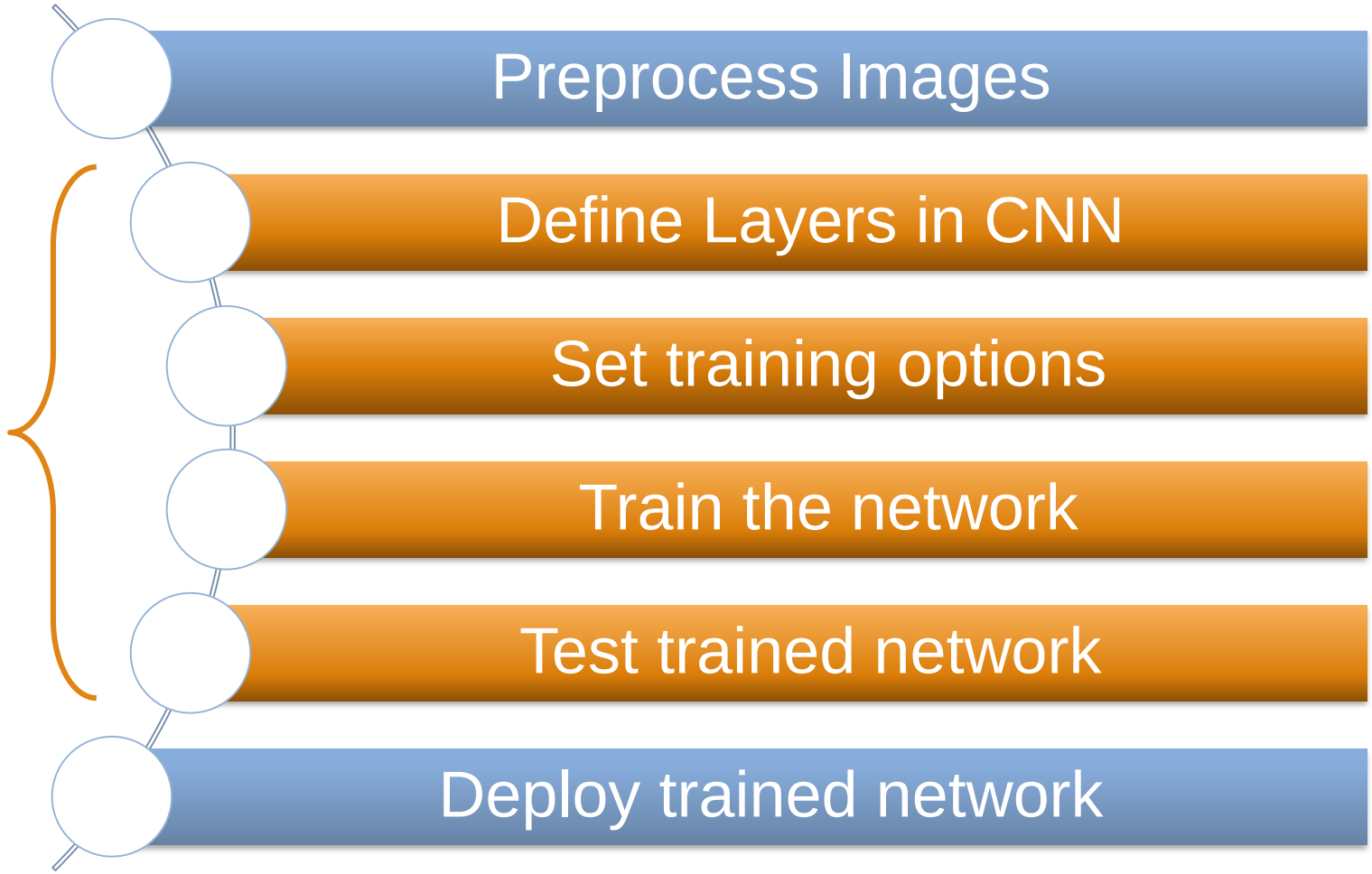


Fine-tune a pre-trained model (Transfer learning)

Use a pretrained CNN as an automatic feature extractor

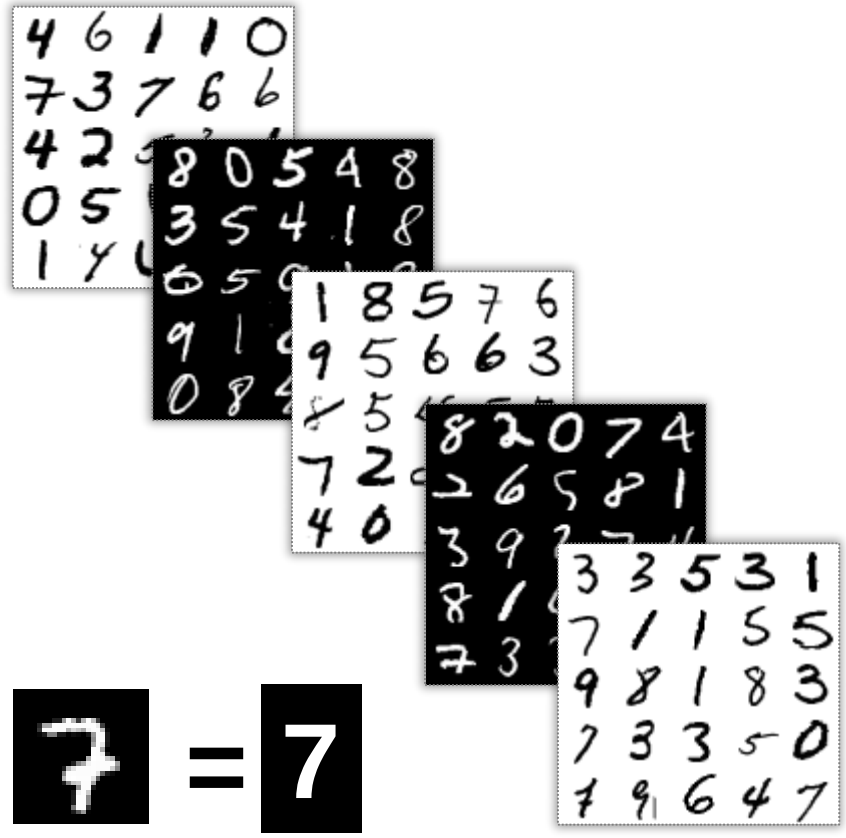
Deep Learning Workflow from Scratch

Repeat these steps until network reaches desired level of accuracy



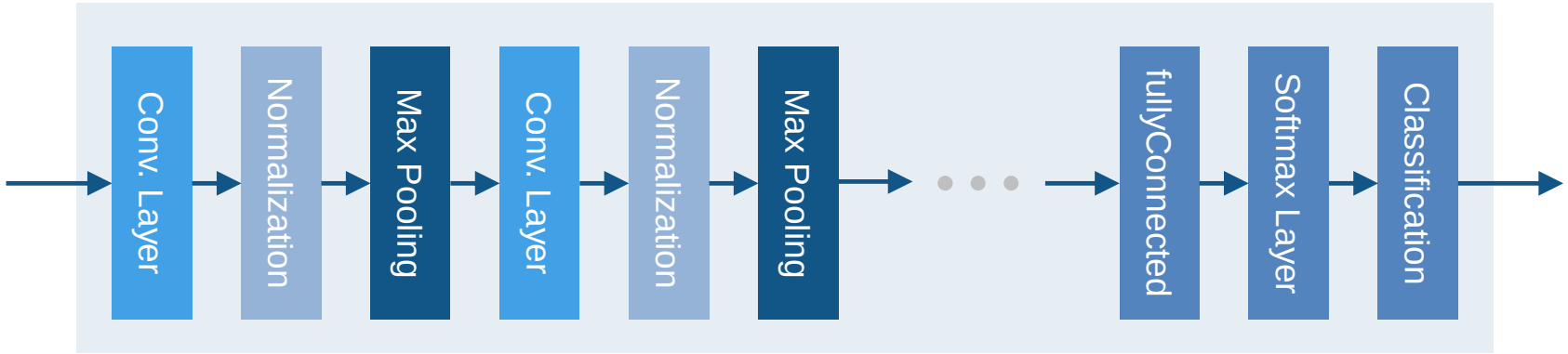
Example : MNIST, The “Hello, World!” of computer vision

What?	A set of handwritten digits from 0-9
Why?	An easy task for machine learning beginners
How many?	60,000 training images 10,000 test images
Best results?	99.79% accuracy



Sources: http://yann.lecun.com/exdb/mnist/
https://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results

Example : MNIST, The “Hello, World!” of computer vision



1
9
8
2

- Typical CNN Architecture
- Learning is performed by error back propagation method

Handwritten Character
28 x 28 pixel

Number (0-9)

Train a Deep Neural Network from Scratch

- Example of network construction with 28x28 sized image

```
layers = [ ...
    imageInputLayer([28 28 1], 'Normalization', 'none');
    convolution2dLayer(5, 20);
    reluLayer();
    maxPooling2dLayer(2, 'Stride', 2);
    fullyConnectedLayer(10);
    softmaxLayer();
    classificationLayer()];
```

Define Layers as in order

```
opts = trainingOptions('sgdm', 'MaxEpochs', 50);
net = trainNetwork(XTrain, TTrain, layers, opts);
```

Define learning rate and maximum iteration number

Call learning functions

Automatically judge the presence or absence of GPU. If there is GPU, learning with CPU if not.

Demo

Hurdle to create a network from scratch

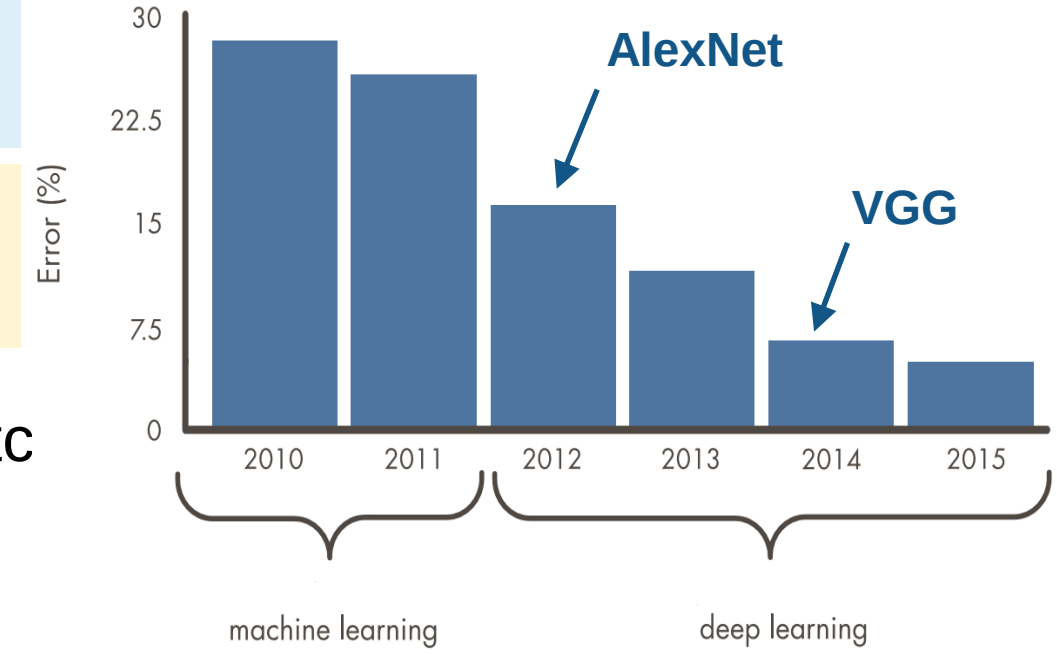
- What kind of learning does the network with high precision do?

AlexNet

■ 5 to 6 days of learning with 2 NVIDIA® GeForce ® GTX 580

VGG Net

■ 2 to 3 weeks study with NVIDIA® GeForce ® TITAN Black 4 aircraft



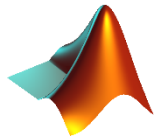
- A hurdle when creating a network with scratc
  - Knowledge of network construction
  - Large number of image sets
  - Extensive calculation cost

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks" In NIPS, pp.1106-1114, 2012

K. Simonyan, A. Zisserman "Very Deep Convolutional Networks for Large-Scale Image Recognition" arXiv technical report, 2014

Source: ILSVRC Top-5 Error on ImageNet

Deep Learning Application Approaches

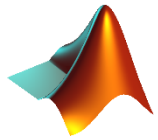


Train a Deep Neural Network from Scratch

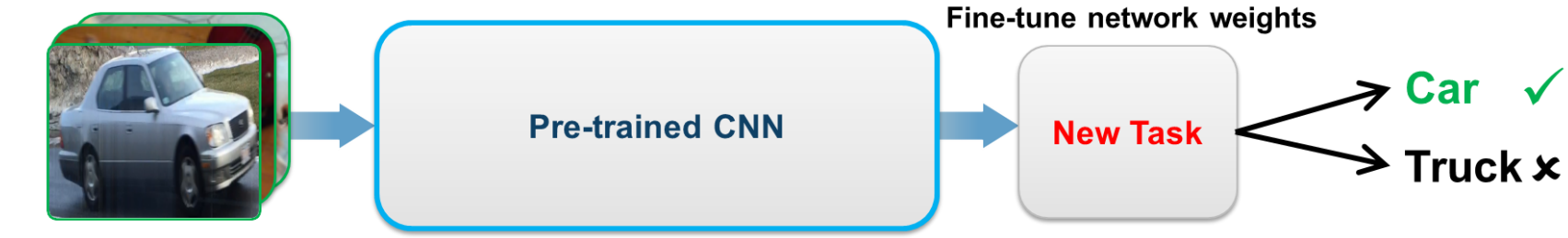
Fine-tune a pre-trained model (Transfer learning)

Use a pretrained CNN as an automatic feature extractor

Deep Learning Application Approaches

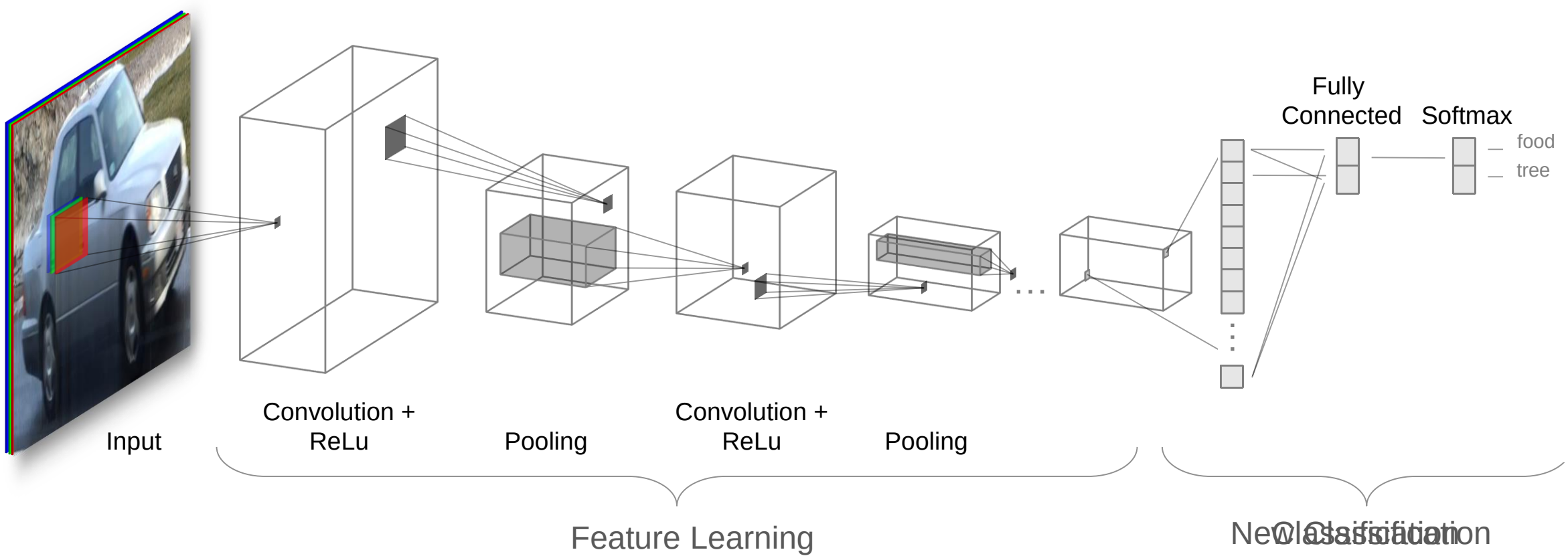


Train a Deep Neural Network from Scratch

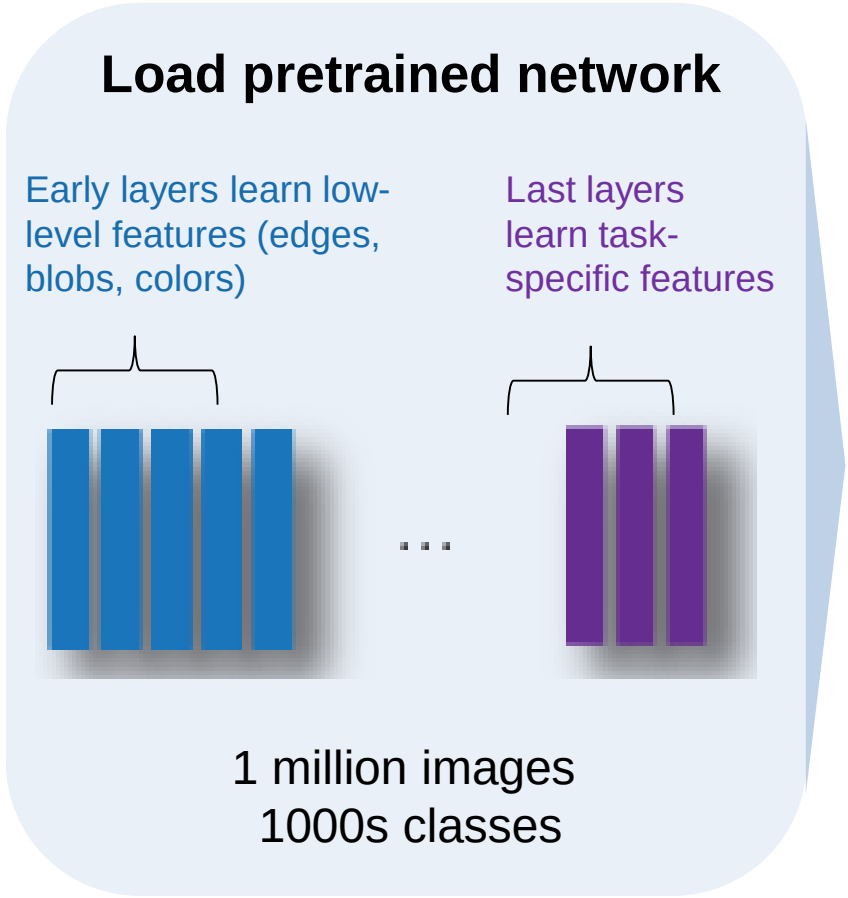


Use a pretrained CNN as an automatic feature extractor

Transfer Learning in CNN: Replace final layers



Transfer Learning Workflow



Transfer Learning Workflow

Load pretrained network

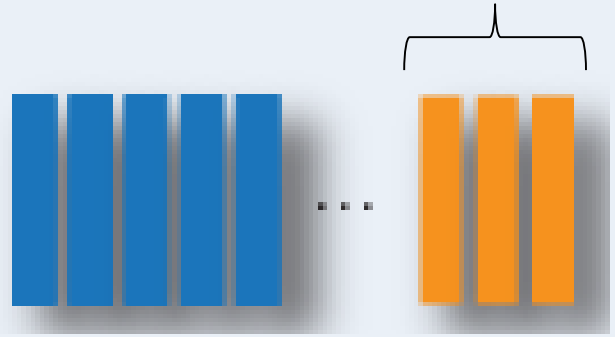
Early layers that learned low-level features (edges, blobs, colors)

Last layers that learned task specific features

1 million images
1000s classes

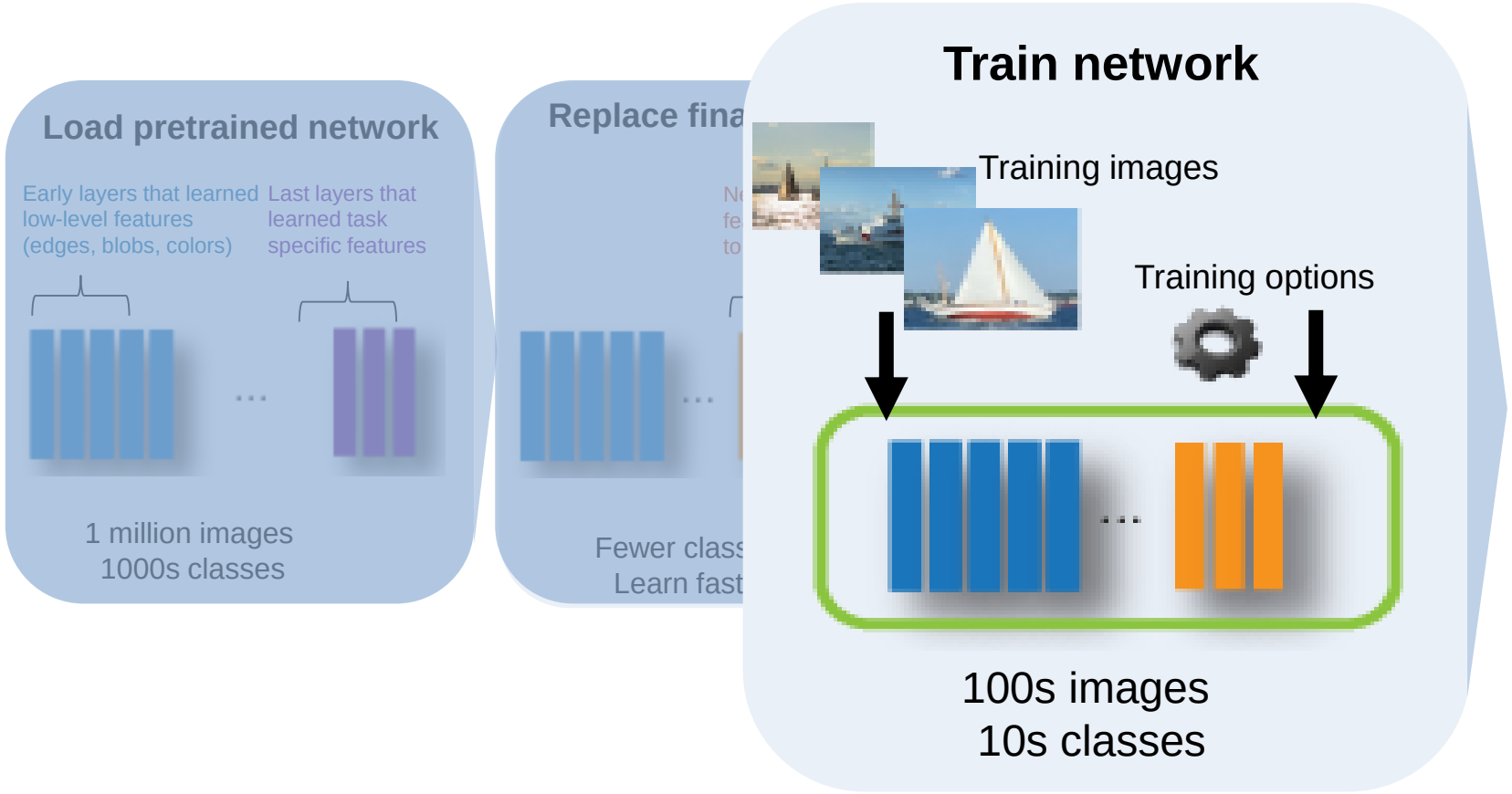
Replace final layers

New layers learn features specific to your data

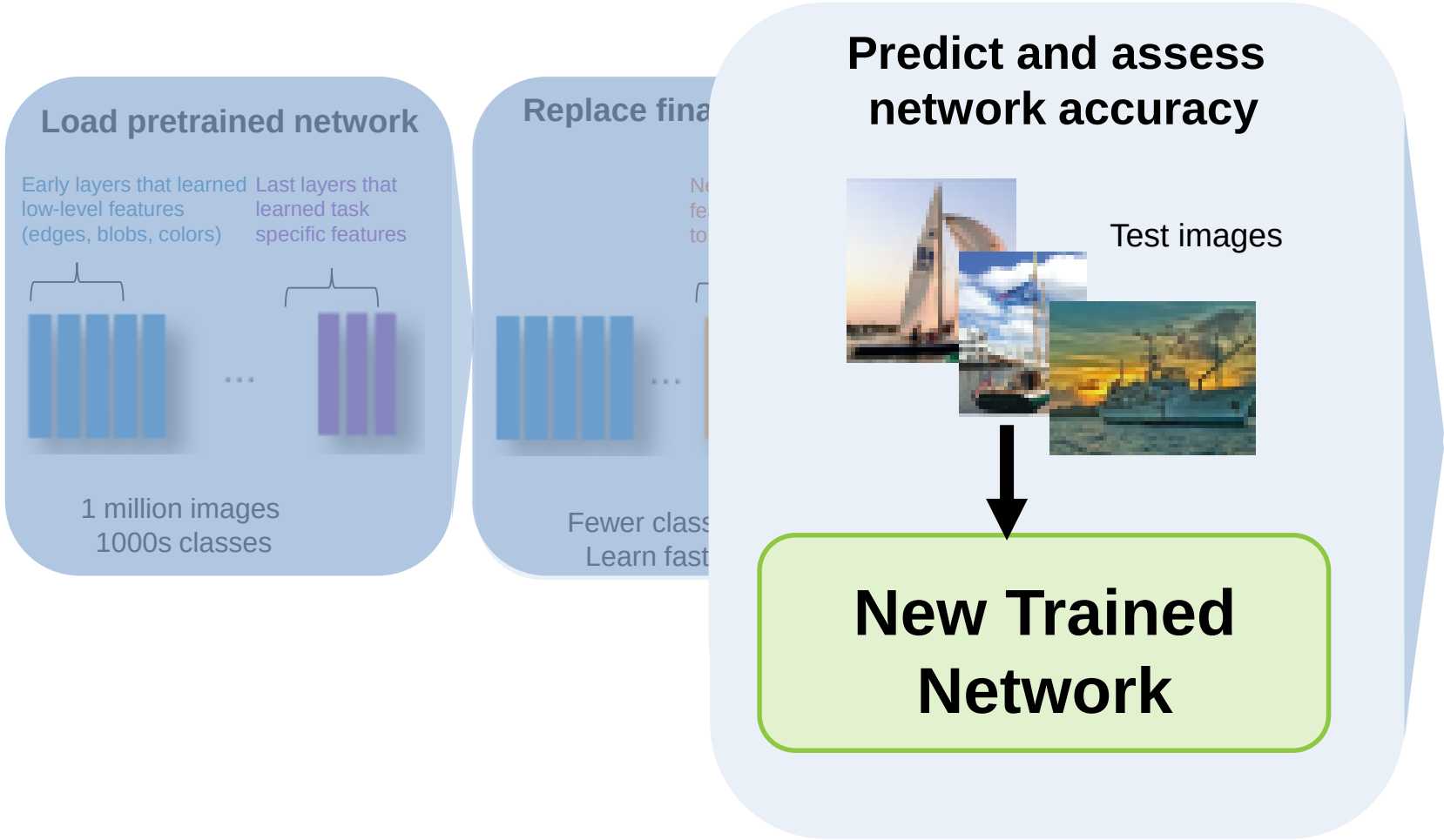


Fewer classes
Learn faster

Transfer Learning Workflow



Transfer Learning Workflow



Transfer Learning Workflow

Load pretrained network

Early layers that learned low-level features (edges, blobs, colors)

Last layers that learned task specific features

1 million images
1000s classes

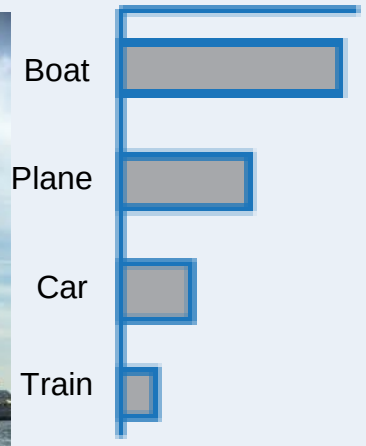
Replace fina

Ne
fe
to

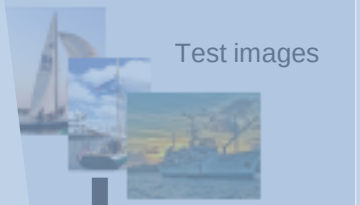
Fewer class
Learn fast

Deploy results

Probability



redict and assess etwork accuracy



Test images

ained Network

Transfer Learning Workflow

Load pretrained network

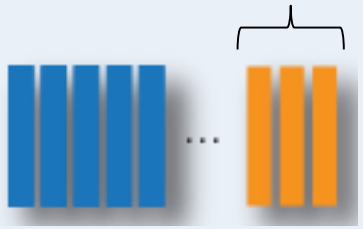
Early layers that learned low-level features (edges, blobs, colors)

Last layers that learned task specific features

1 million images
1000s classes

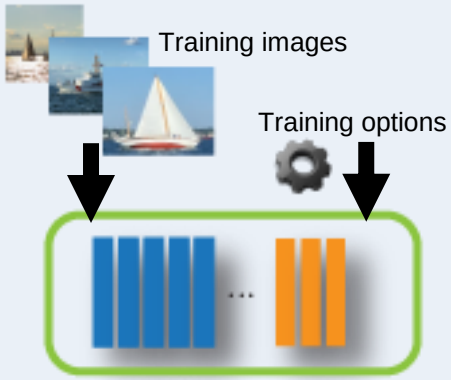
Replace final layers

New layers to learn features specific to your data



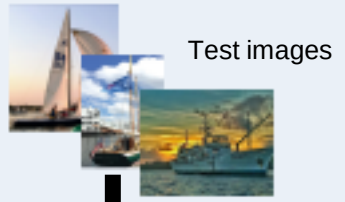
Fewer classes
Learn faster

Train network



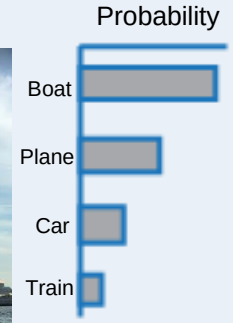
100s images
10s classes

Predict and assess network accuracy

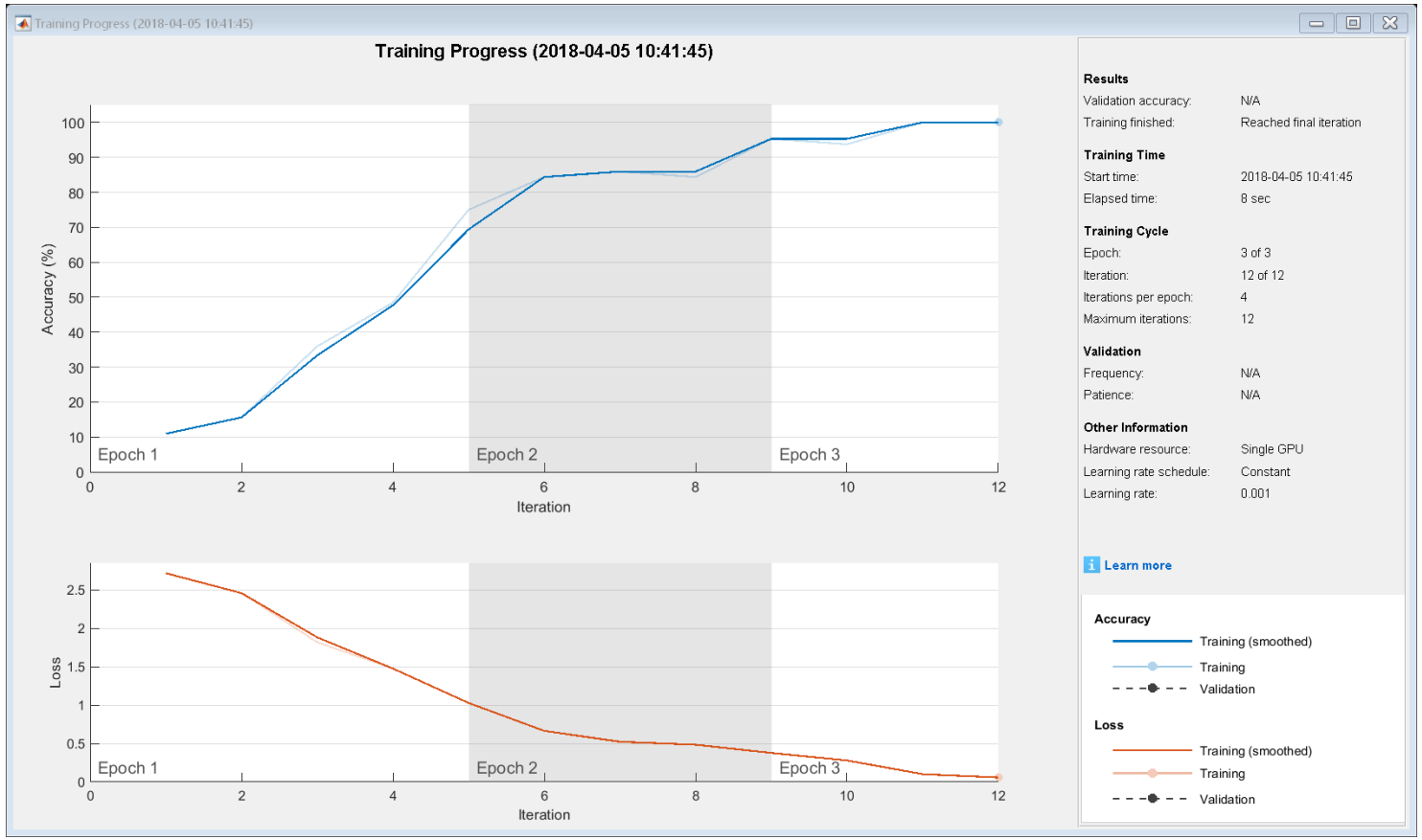


New Trained Network

Deploy results

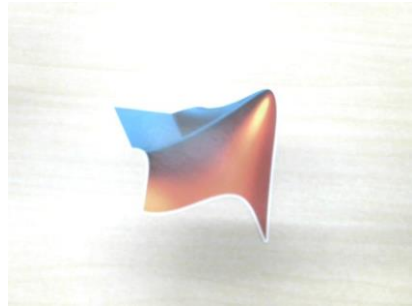


Example: Transfer Learning



Validation Accuracy : 100%

MathWorks Logo



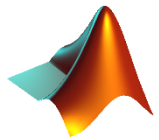
Laptop

Smart Phone

Clip

Deep Learning Application Approaches

Train a Deep Neural Network from Scratch



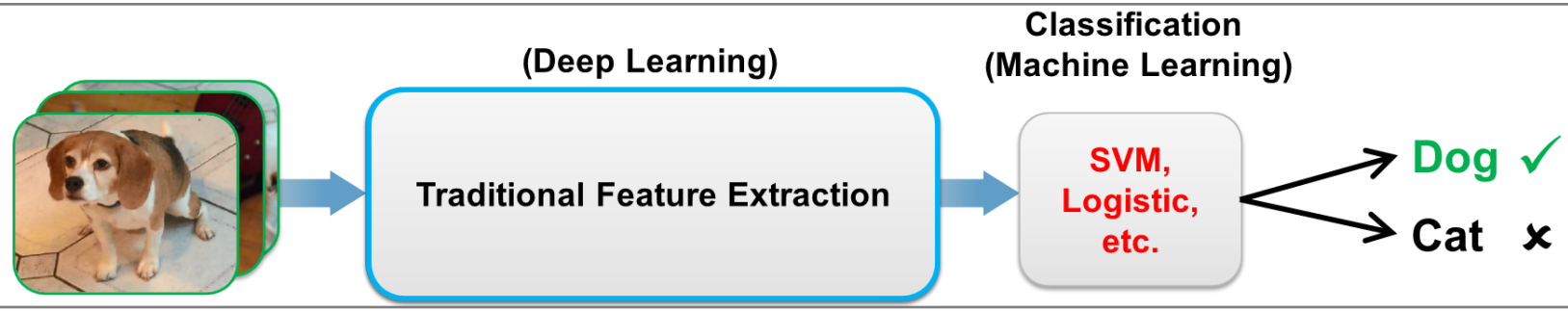
Fine-tune a pre-trained model (Transfer learning)

Use a pretrained CNN as an automatic feature extractor

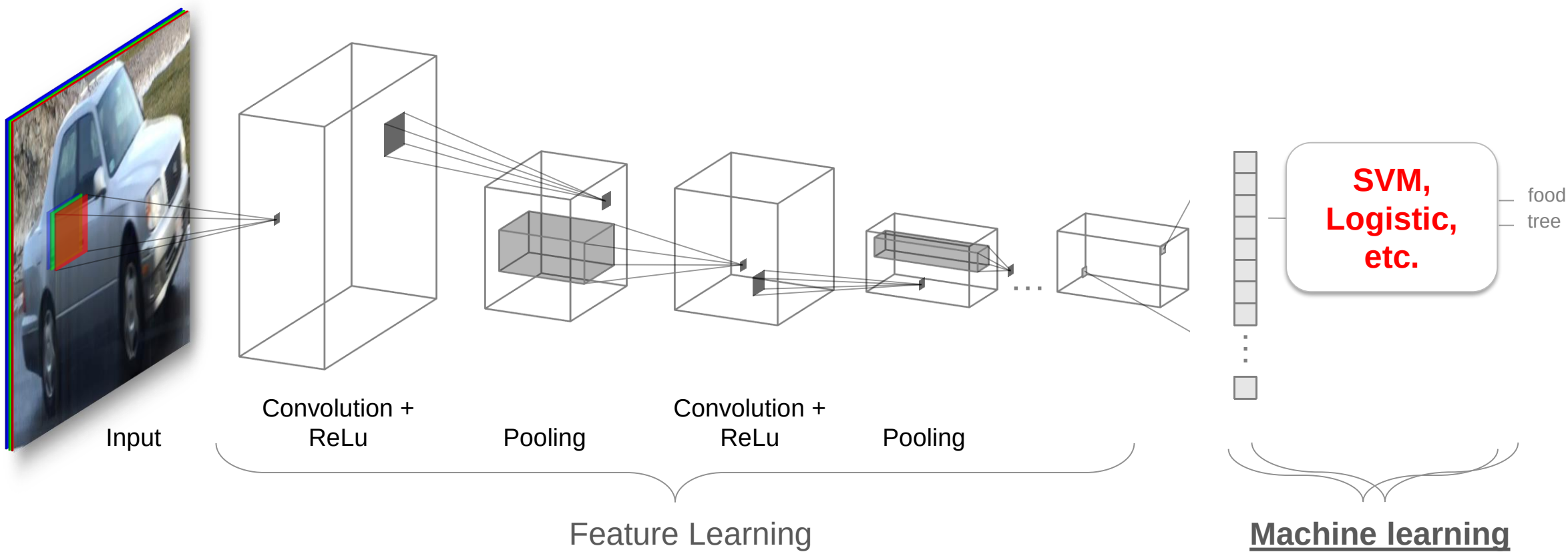
Deep Learning Application Approaches

Train a Deep Neural Network from Scratch

Fine-tune a pre-trained model (Transfer learning)

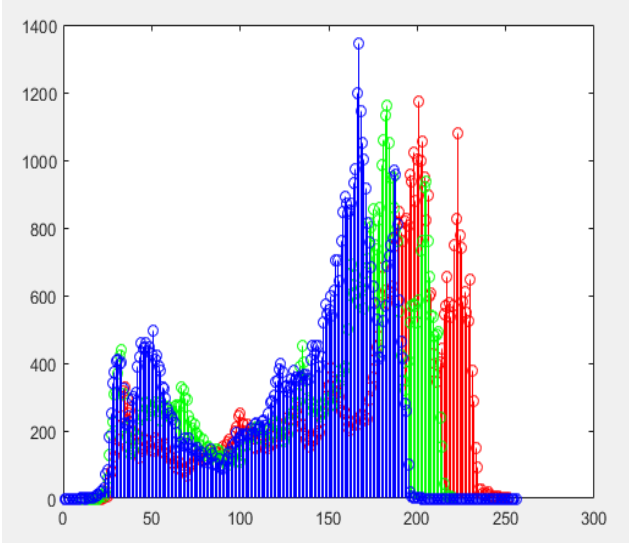


Transfer Learning: Feature extraction

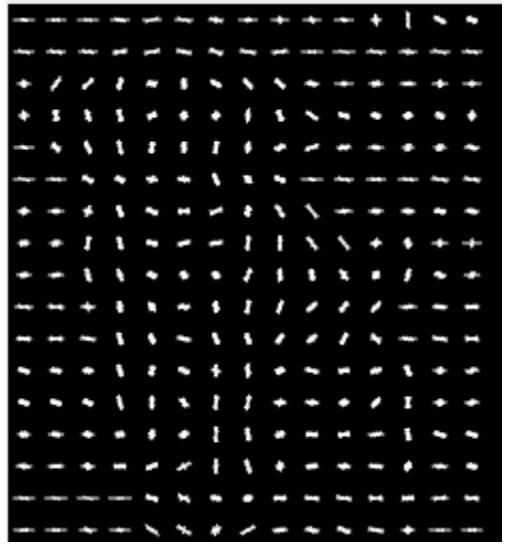


What is Feature Extraction ?

Image Pixels



Color



Histogram of Oriented Gradients

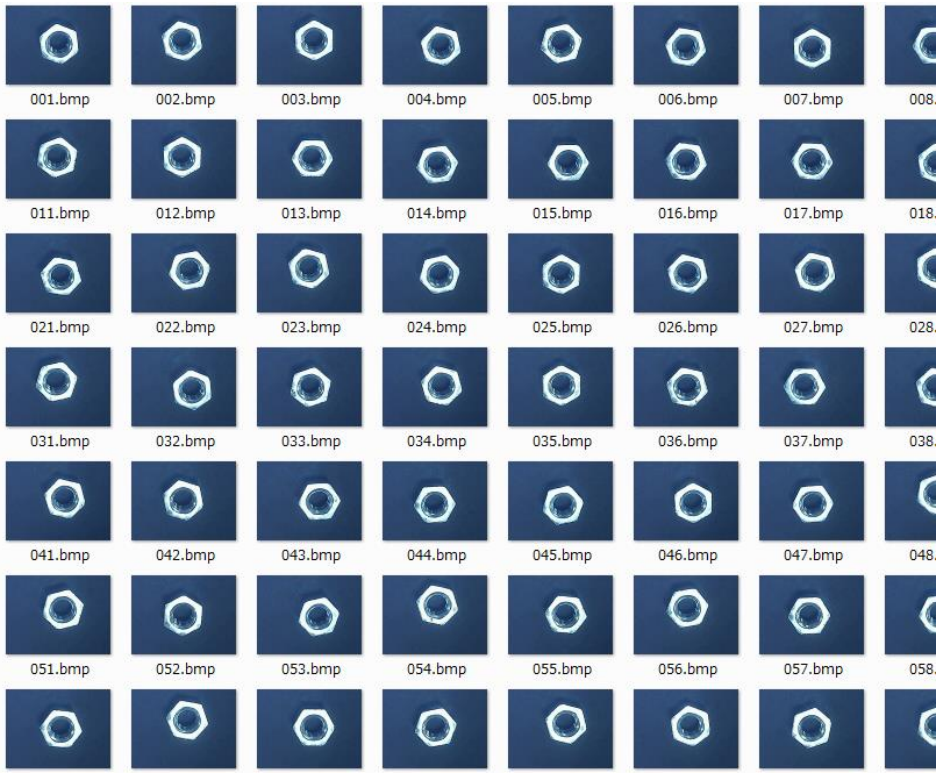
Edges

Dense

Sparse

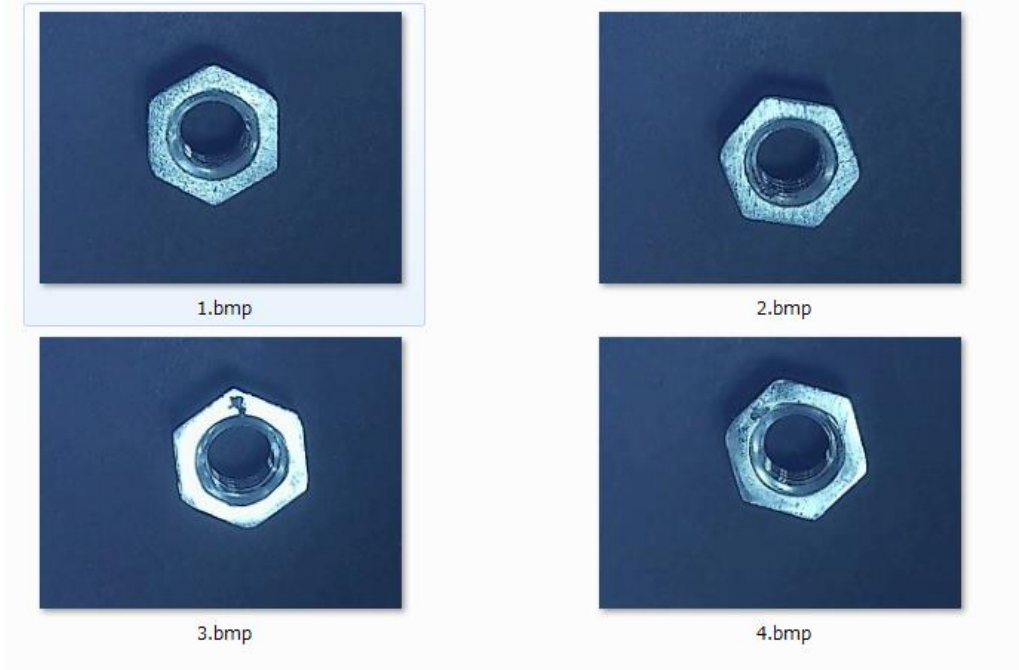
Example : Image anomaly Detection Task

100 hexagon nuts



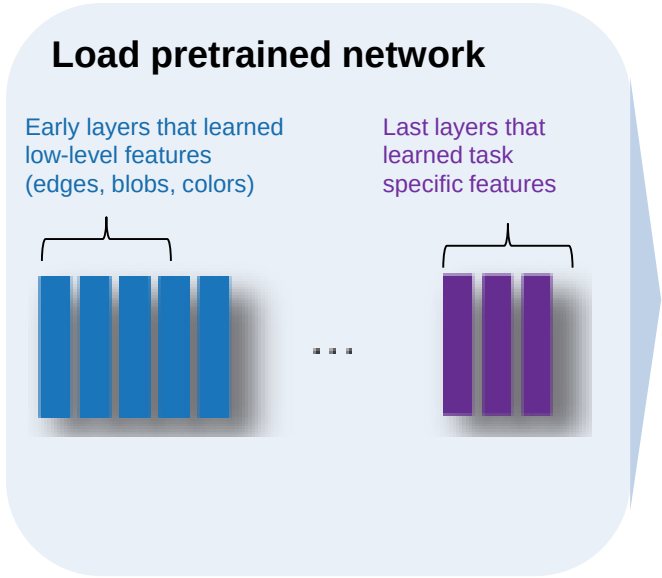
Task: Find 4 defective units in 100 test images.

4 Defective units

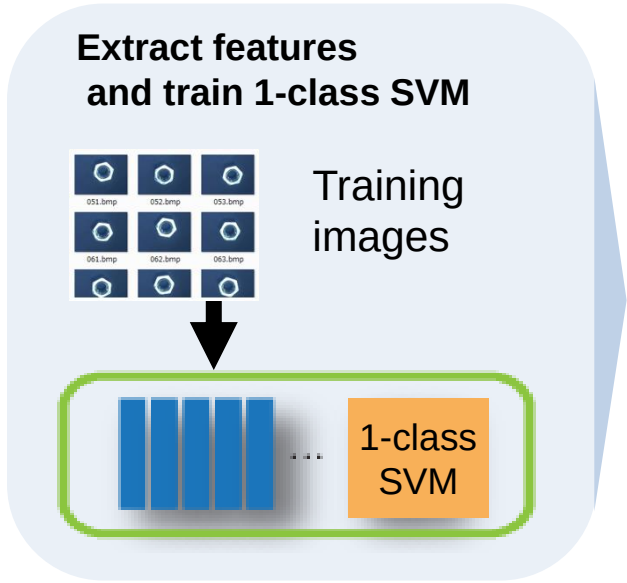


- Challenge
  - Number of defective units is very small.
  - Difficult to apply supervised learning to this task.

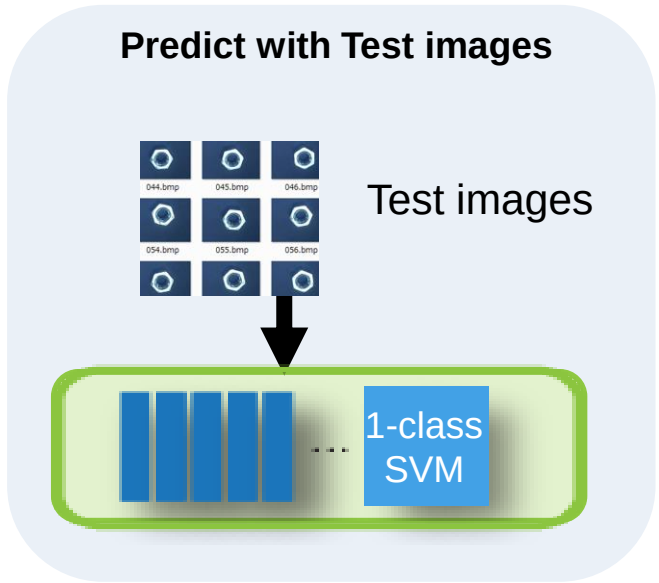
Example : Image anomaly Detection Task



・Load pretrained AlexNet



・Extract features with pre-trained model（AlexNet）
・Train 1-class SVM with 100 images.
・Unsupervised training

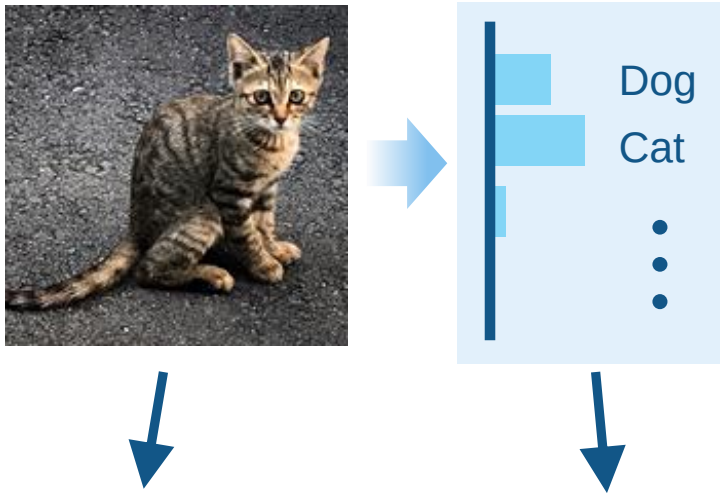


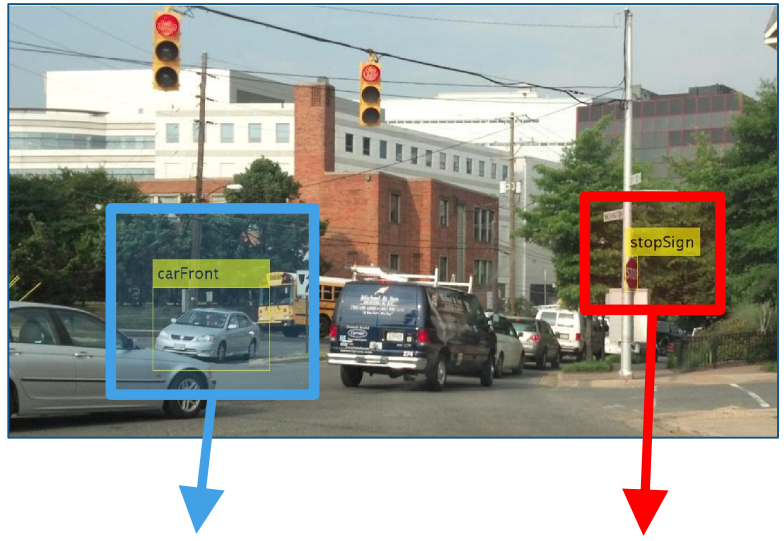
・Predict
・List sorted by predicted score
・Find 4 defective units from 100 test images

Demo

Object Recognition using Deep Learning

Object recognition (whole image)	Object detection and recognition	Object recognition (in pixels)
CNN (Convolutional Neural Network)	**R-CNN / Fast R-CNN / Faster R-CNN**	**SegNet / FCN**

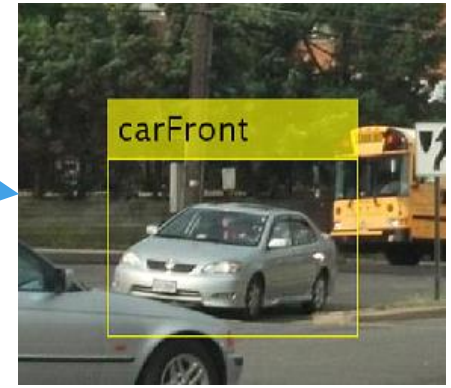




R-CNNs : Regions with Convolutional Neural Networks

- Object detection and identification method combining CNN with computer vision method

Stop Sign

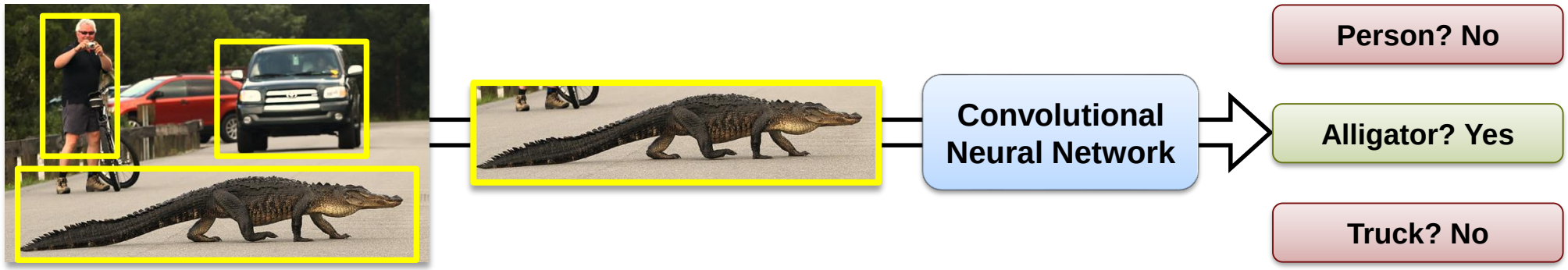


Front of Car

Detection example when R-CNN learns car front and stop signs

R-CNNs : Regions with Convolutional Neural Networks

- Take a neural network trained for image classification and modify it for object detection.



1. For a given input image, region proposals (ROIs) are generated
2. Each region proposal is then independently send through the CNN to compute features
3. The neural network makes a classification for each region proposal

The differences between the R-CNN methods occur at the region proposal stage.

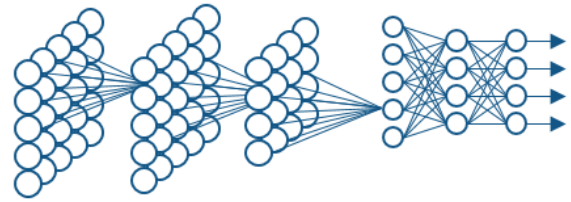
R-CNNs Learning

- >> **detector** = trainRCNNObjectDetector(groundTruth, network, options)

FastRCNN
FasterRCNN

1 imageFilename	2 stopSign	3 carRear
'stopSignImages/image001.jpg'	[856,318,39,41]	[398,378,315,210]
'stopSignImages/image002.jpg'	[445,523,52,54]	[332,633,691,287]
'stopSignImages/image003.jpg'	[897,365,49,48]	[718,409,74,66;1...
'stopSignImages/image004.jpg'	[948,424,34,44]	[757,503,143,69]
'stopSignImages/image005.jpg'	[980,393,31,56]	[]
'stopSignImages/image006.jpg'	[1.0408e+03,35...	[]

Ground Truth



Series Network or Array of layers

- >> [bbox, score, label] = detect(**detector**, image);

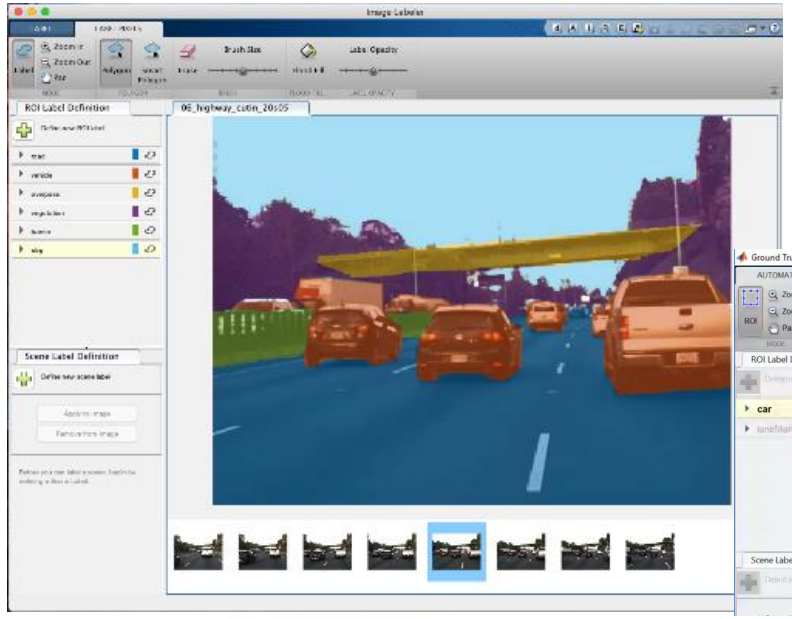
Ground truth Labeling

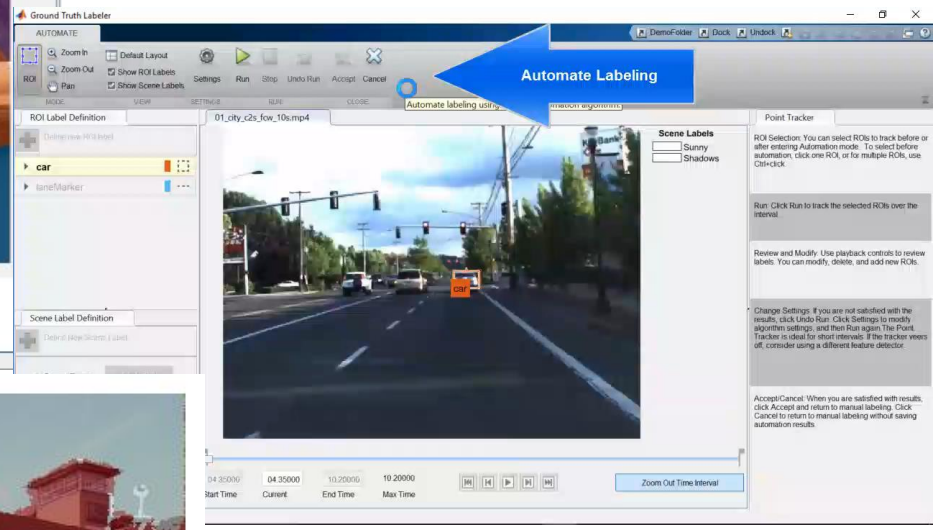
"How do I *label* my data?"

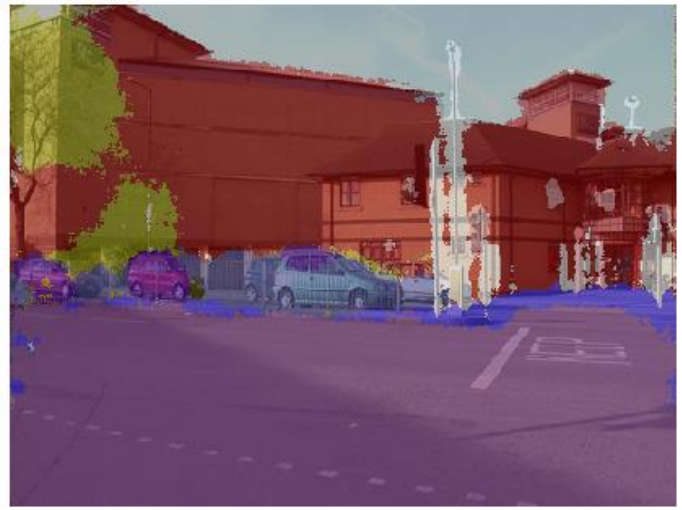
New App for Ground Truth Labeling

Label pixels and regions for semantic segmentation

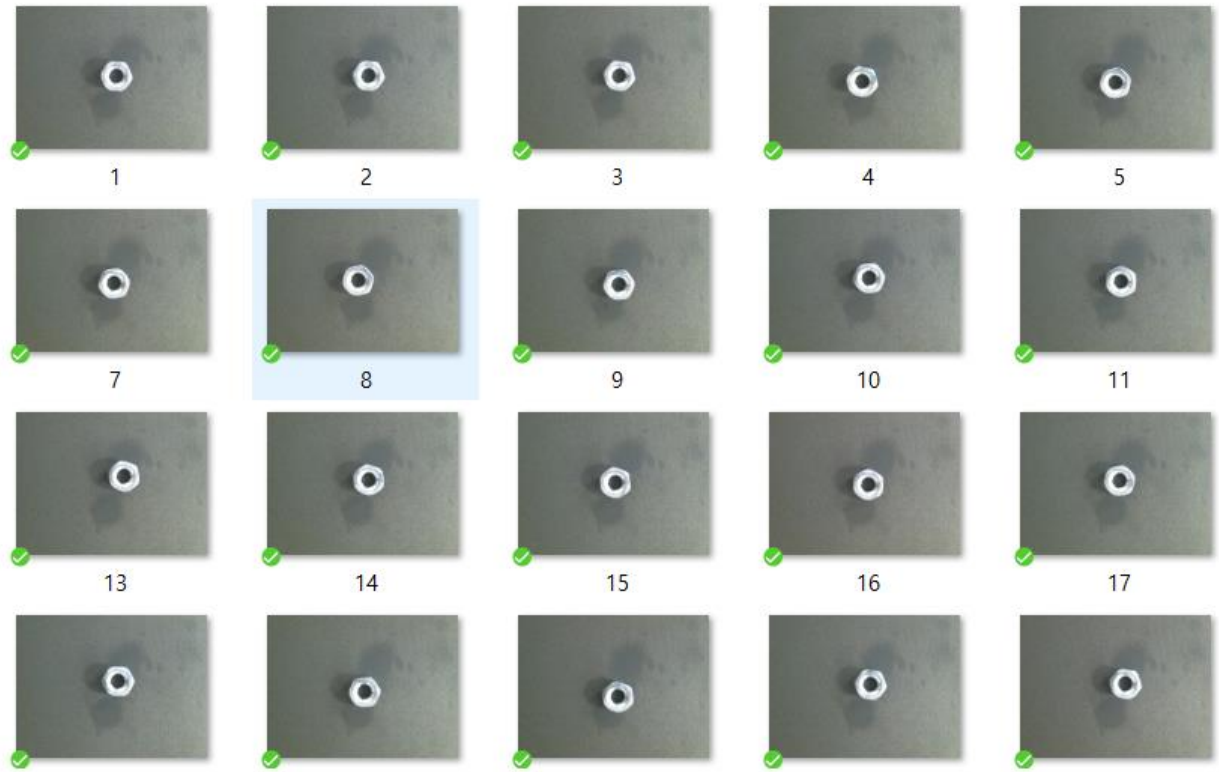
Data







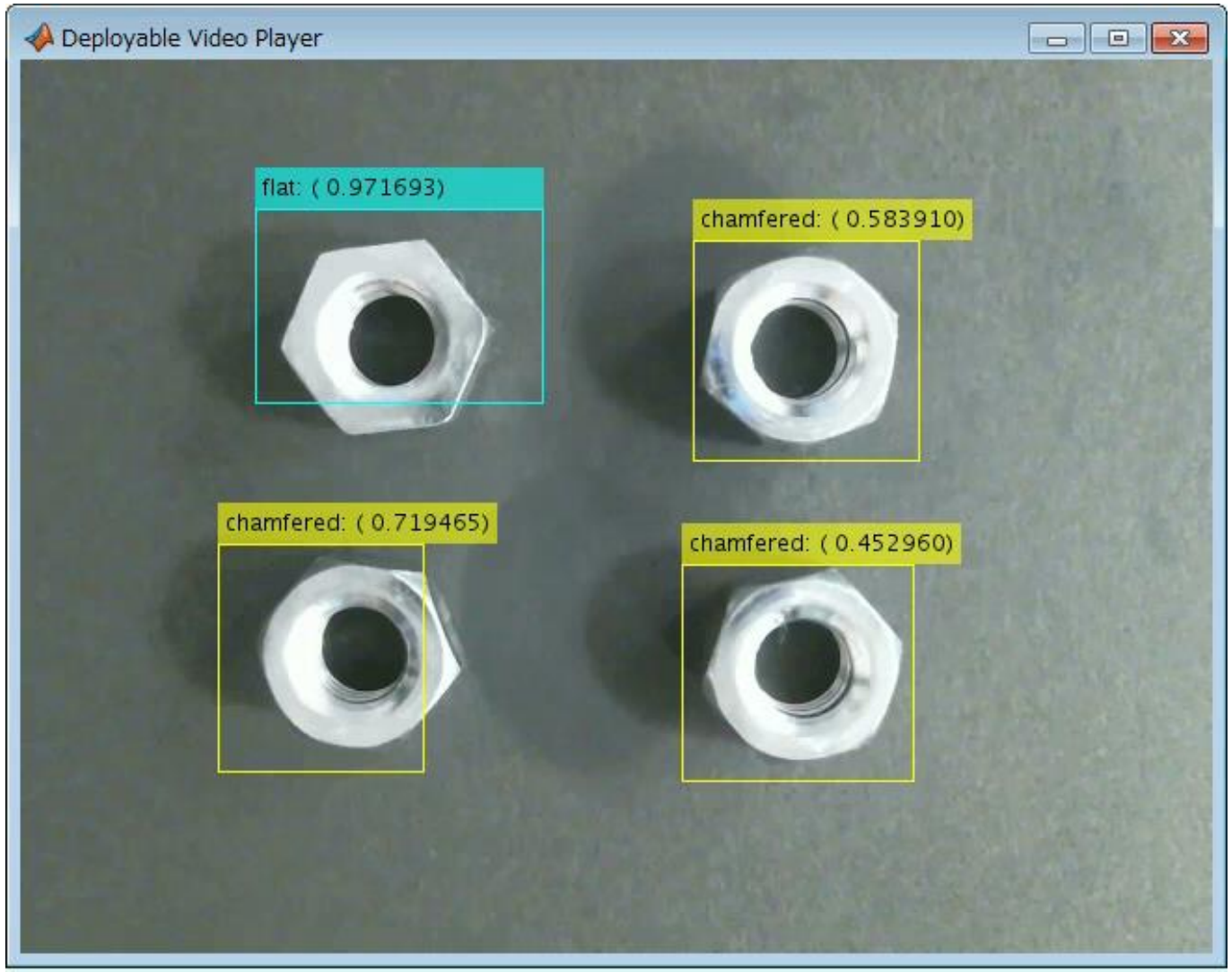
Example : Detect and Classify hexagon nut using Faster RCNN



- Detect and classify hexagon nut into Top/Bottom using Faster R-CNN.
- Features of Top / bottom are Same shape, Same color, Same size and Tiny difference in texture

Demo

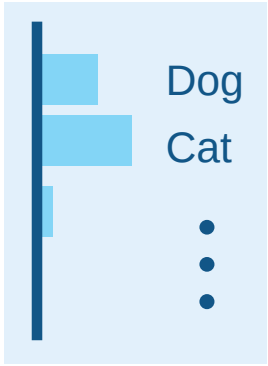
Example : Detect and Classify hexagon nut using Faster RCNN



Control manipulator robot using this object recognition.

Object Recognition using Deep Learning

Object recognition (whole image)	Object detection and recognition	Object recognition (in pixels)
CNN (Convolutional Neural Network)	**R-CNN / Fast R-CNN / Faster R-CNN**	**SegNet / FCN**



Image

Probability

Front of Car

Stop Sign

Road

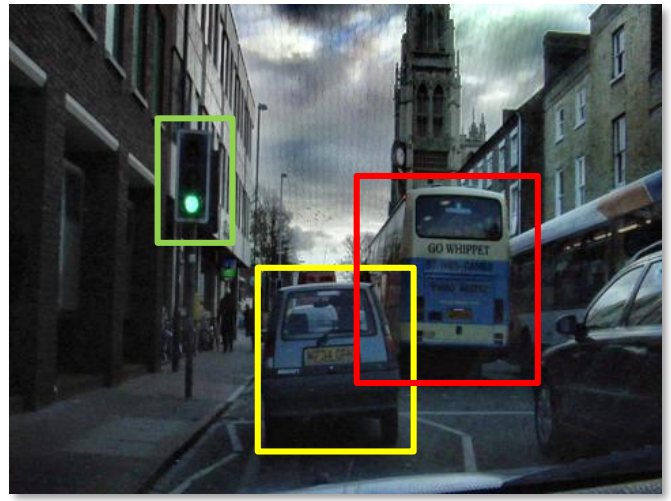
Vehicle

Semantic Segmentation

R2017b

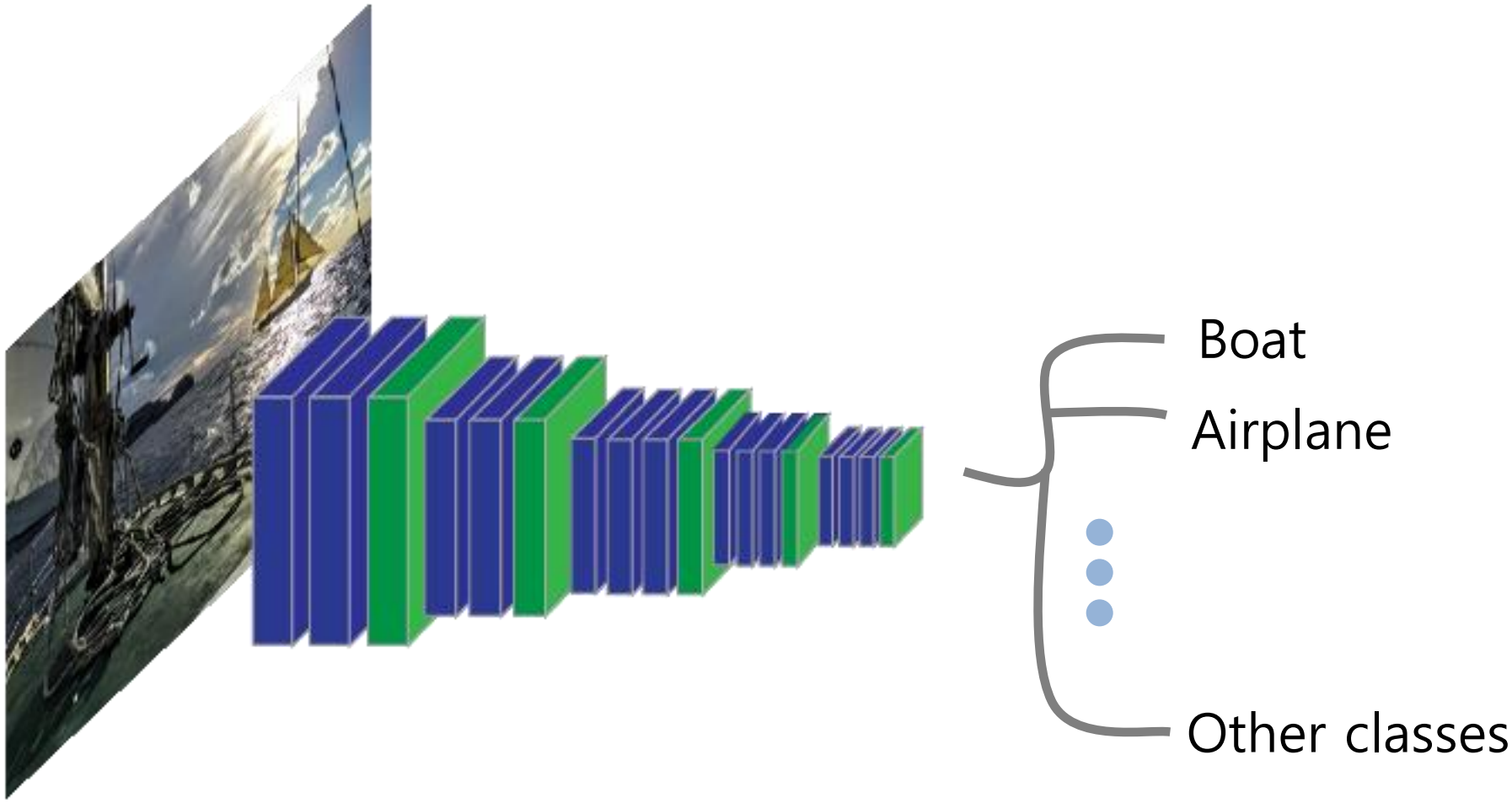
Original Image

ROI detection

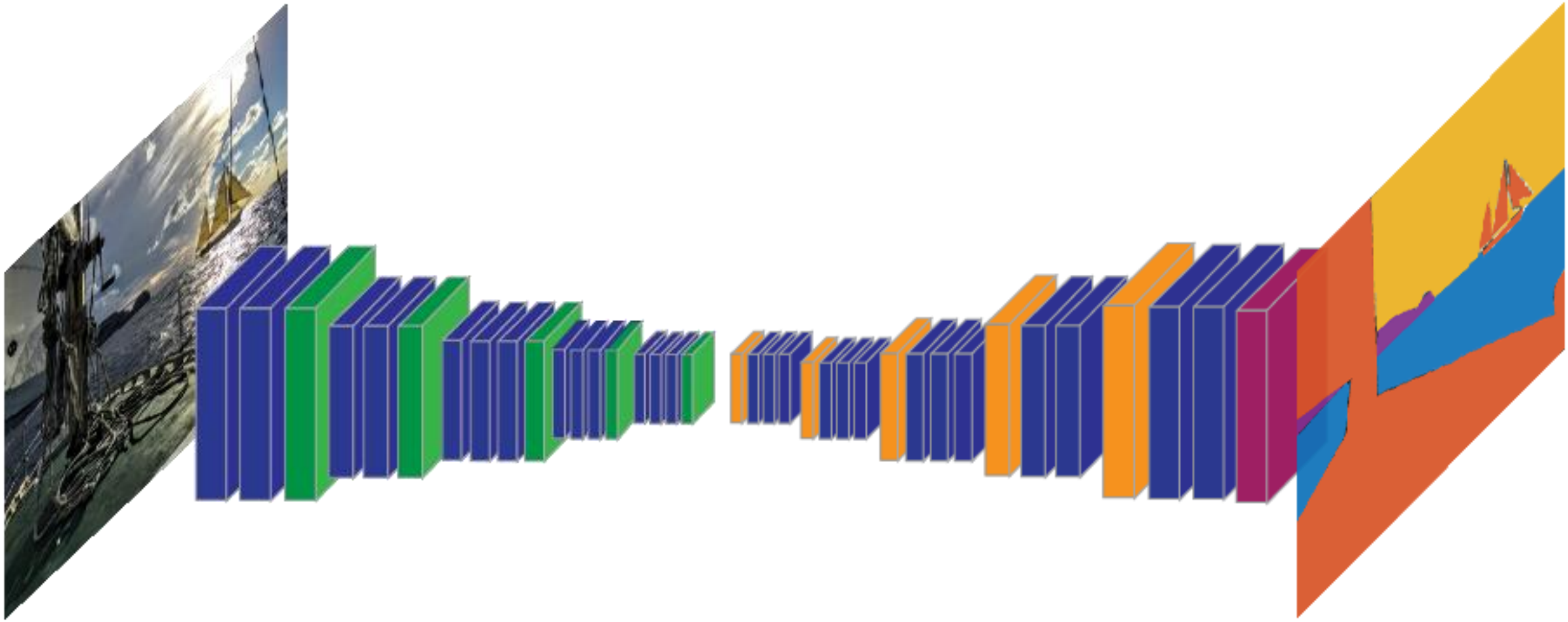


Pixel classification

Semantic Segmentation Network



Semantic Segmentation Network



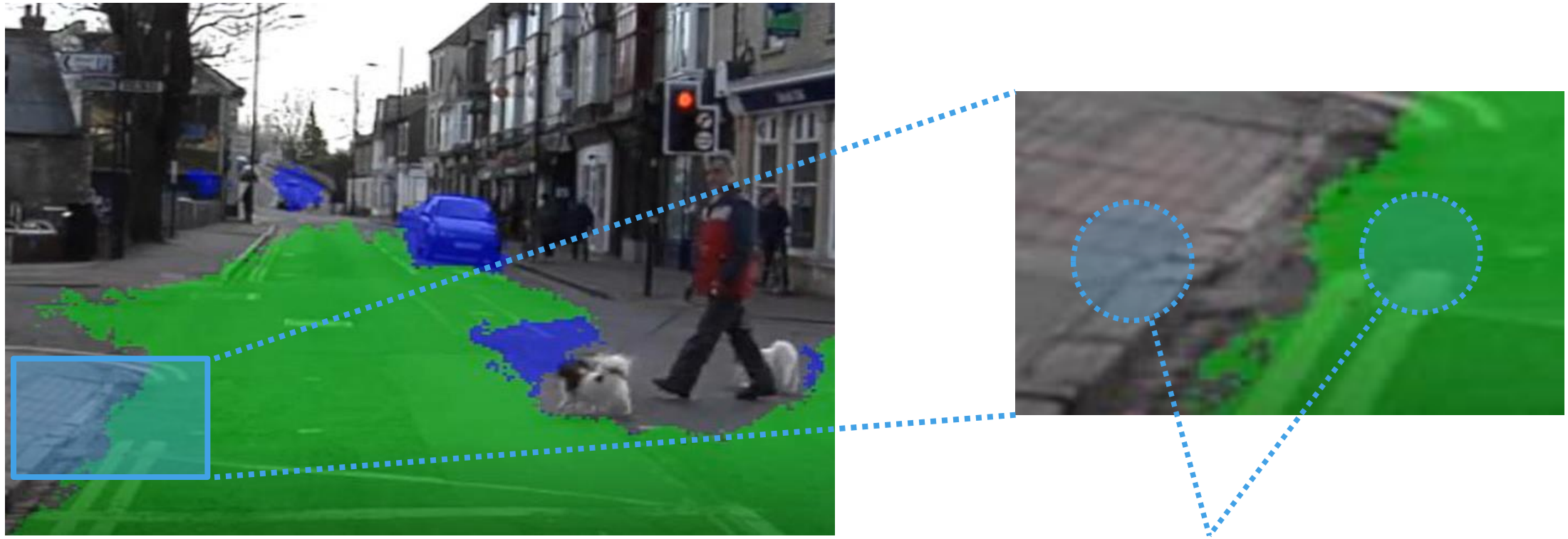
Semantic Segmentation

CamVid Dataset

1. Segmentation and Recognition Using Structure from Motion Point Clouds, ECCV 2008
2. Semantic Object Classes in Video: A High-Definition Ground Truth Database ,Pattern Recognition Letters

Semantic Segmentation

- A method of categorizing each pixels based on its meaning.



- Distinguish between sidewalks and roadways
- It is not just looking at colors

Example : Semantic Segmentation for Free road detection

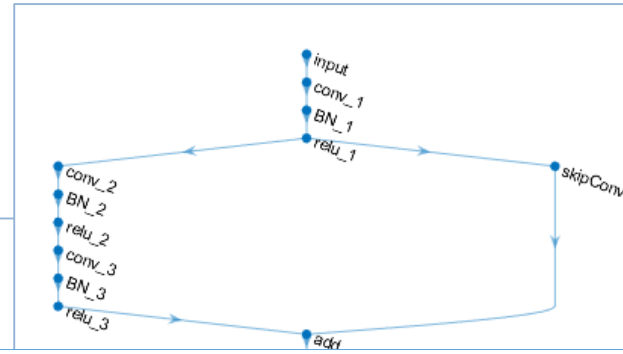
- Use Semantic Segmentation to detect the free space on the road as well as lanes and pavements

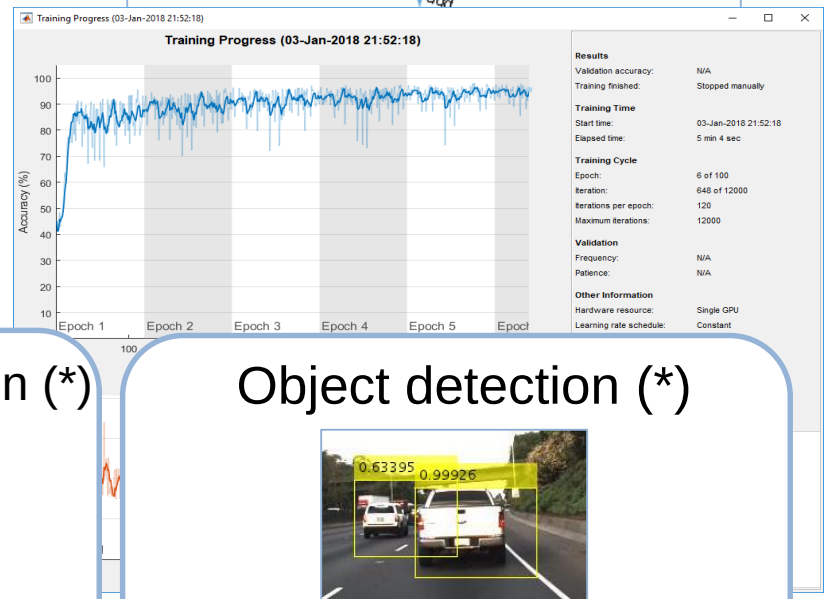
Demo

MATLAB deep network in a nutshell

- A MATLAB deep network (**) is a MATLAB object that contains an array of trained layer objects.
- Layers array can be created, imported, edited, plotted in MATLAB
- Layers are trained with a lot of data and `net = trainNetwork(…,layers)`, most of the time.
- MATLAB deep networks have different usages.

```
layers = [ ...
    imageInputLayer([28 28 1])
    convolution2dLayer(12,25)
```





Classification

car ✗
dog ✓

```
YPred = classify(net,X);
```

Regression

32

```
YPred = predict(net,X);
```

Semantic segmentation (*)

sky
car
road

```
C = semanticseg(I,net);
```

Object detection (*)


```
[bboxes,scores] = detect(net,I);
```


(**) Neural Network Toolbox

(*) Computer Vision Syst. Toolbox

MathWorks® can help you do Deep Learning

Free resources

- **Guided evaluations with a MathWorks deep learning engineer**
- Proof-of-concept projects
- **Deep learning hands-on workshop**
- Seminars and technical deep dives
- Deep learning onramp course

More options

- Consulting services
- Training courses
- Technical support
- Advanced customer support
- Installation, enterprise, and cloud deployment
- MATLAB for Deep Learning

© 2018 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.