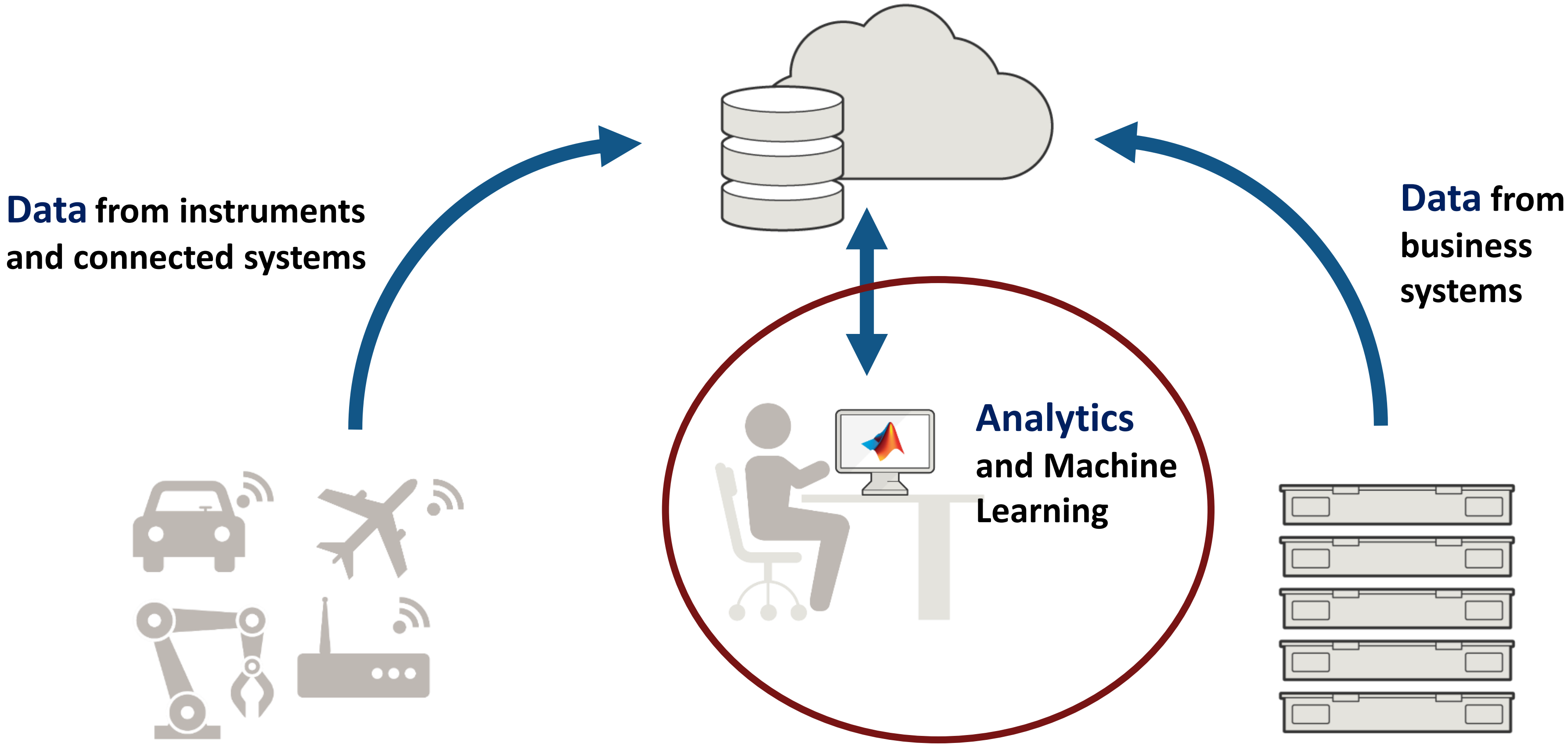
MATLAB EXPO 2016

Machine Learning and Deep Learning

Jon Cherrie

Architecture of an analytics system

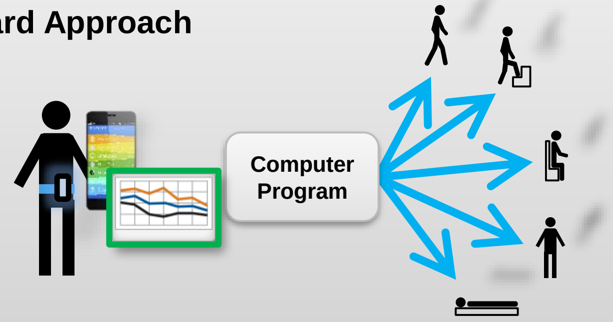


What is Machine Learning

Machine learning uses **data** and produces a **program** to perform a **task**

Task: Human Activity Detection

Standard Approach



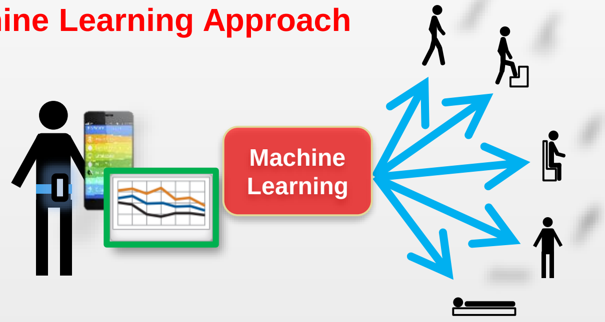
Hand Written Program

```
if X_acc > 0.5
      then “SITTING”
if Y_acc < 4 & Z_acc > 5
      then “STANDING”
      …
```

Formula or Equation

$$Y_{activity} = \beta_1 X_{acc} + \beta_2 Y_{acc} + \beta_3 Z_{acc} + \dots$$

Machine Learning Approach



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

$$model = < \begin{matrix}\textit{Machine} \\ \textit{Learning} \\ \textit{Algorithm}\end{matrix} > (sensor\_data, activity)$$

Classify images into 1000 categories

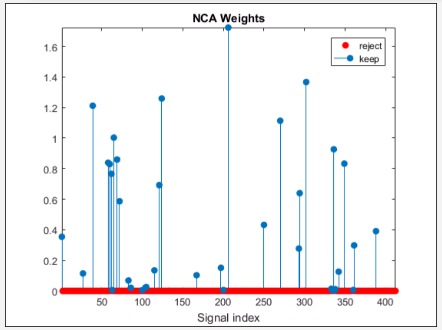
Monitor a manufacturing process

X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'
2994.1	2548.2	2195.1	1046.1	1.3204	100	103.34	'fail'
2928.8	2479.4	2196.2	1605.8	0.9959	100	97.916	'fail'
2920.1	2507.4	2195.1	1046.1	1.3204	100	103.34	'pass'
3051.4	2529.3	2184.4	877.63	1.4668	100	107.87	'pass'
2964	2629.5	2224.6	947.77	1.2924	100	104.85	'fail'

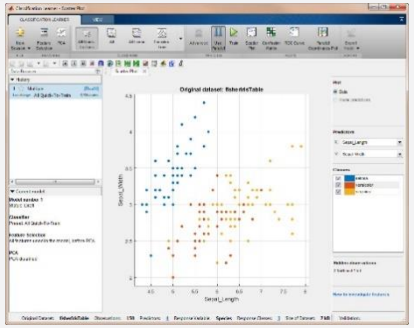
Overview

X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'
2994.1	2548.2	2195.1	1046.1	1.3204	100	103.34	'fail'
2928.8	2479.4	2196.2	1605.8	0.9959	100	97.916	'fail'
2920.1	2507.4	2195.1	1046.1	1.3204	100	103.34	'pass'
3051.4	2529.3	2184.4	877.63	1.4668	100	107.87	'pass'
2964	2629.5	2224.6	947.77	1.2924	100	104.85	'fail'

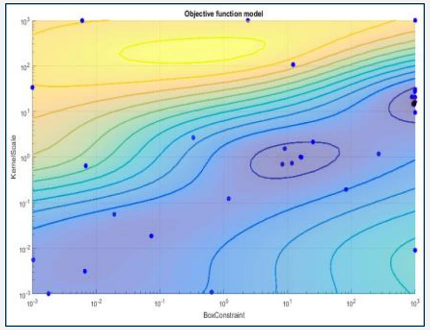
Machine Learning



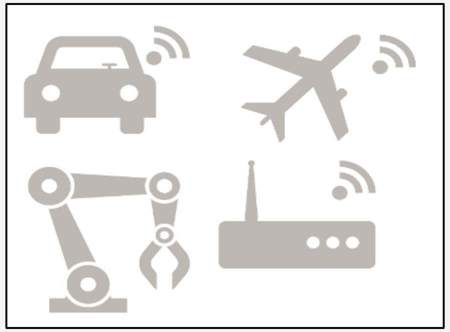
Which data to use



Choose a model



Fine tuning

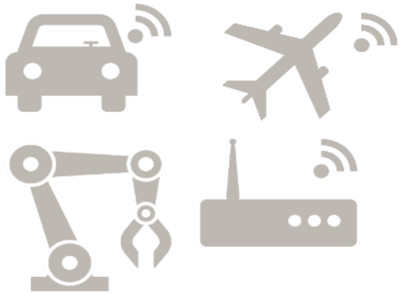


Share & Integrate

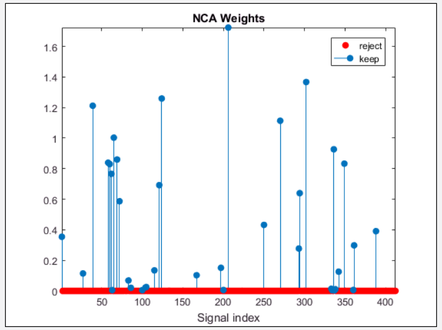
Deep Learning

X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'
2994.1	2548.2	2195.1	1046.1	1.3204	100	103.34	'fail'
2928.8	2479.4	2196.2	1605.8	0.9959	100	97.916	'fail'
2920.1	2507.4	2195.1	1046.1	1.3204	100	103.34	'pass'
3051.4	2529.3	2184.4	877.63	1.4668	100	107.87	'pass'
2964	2629.5	2224.6	947.77	1.2924	100	104.85	'fail'

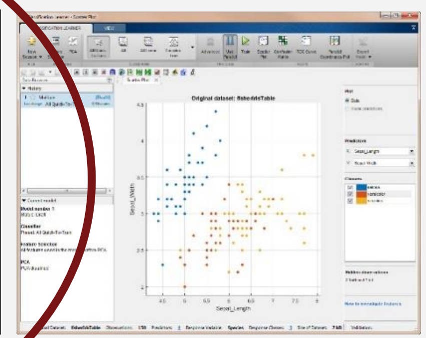
Deep Learning



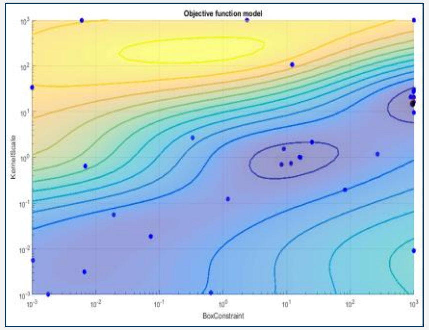
Which data to use?
Feature selection



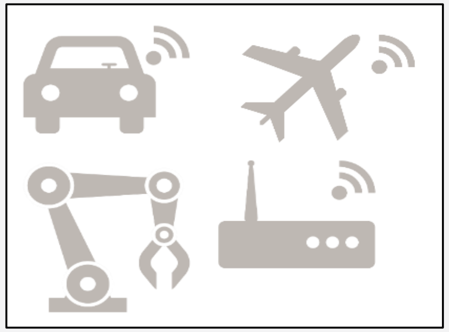
Which data to use



Choose a model



Fine tuning

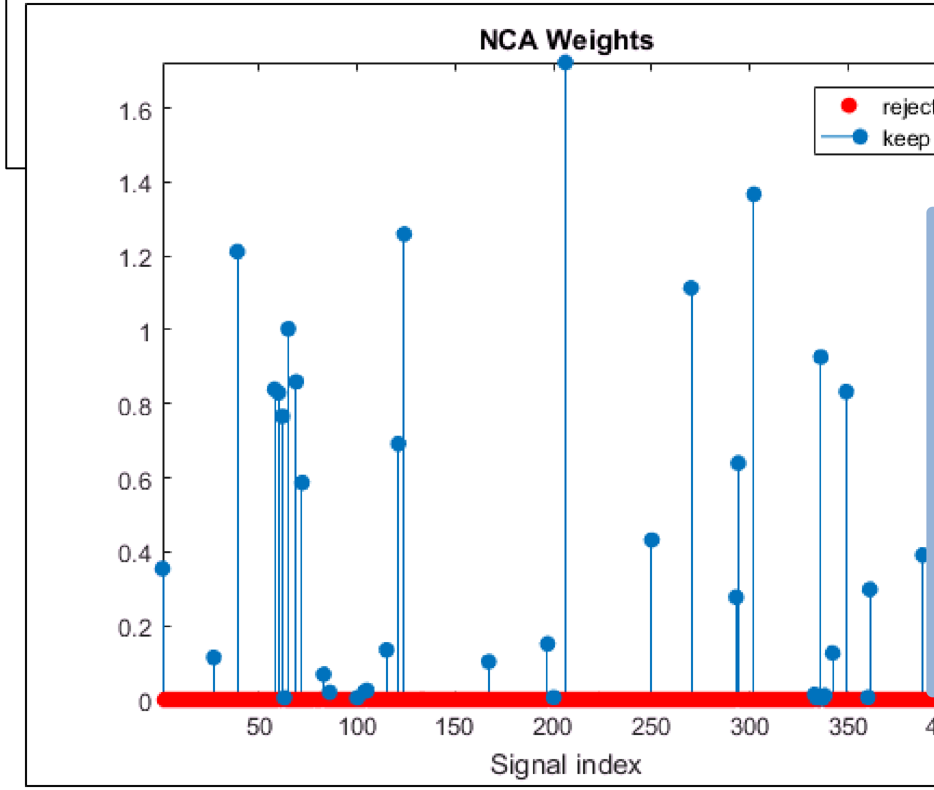


Share & Integrate

Which data to use? Feature selection

X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
						104.24	'pass'
						100.4	'pass'
						100.4	'pass'
						102.34	'pass'
						106.24	'pass'
						106.24	'pass'
						106.24	'pass'

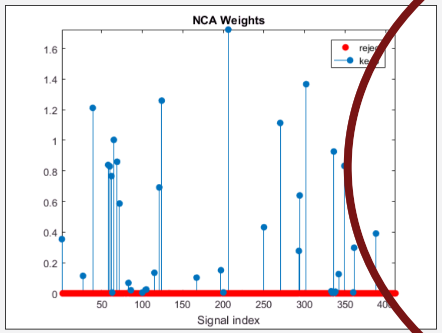
Neighbourhood Component Analysis (NCA)



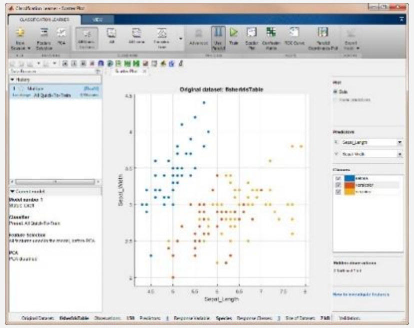
X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'

R2016b

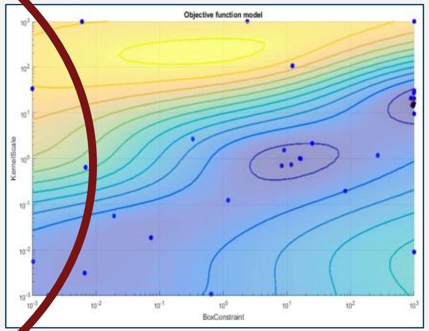
Which model to use?
Classification Learner



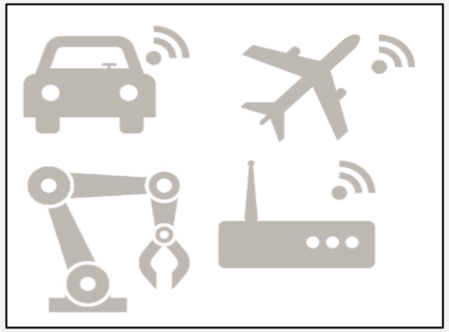
Which data to use



Choose a model

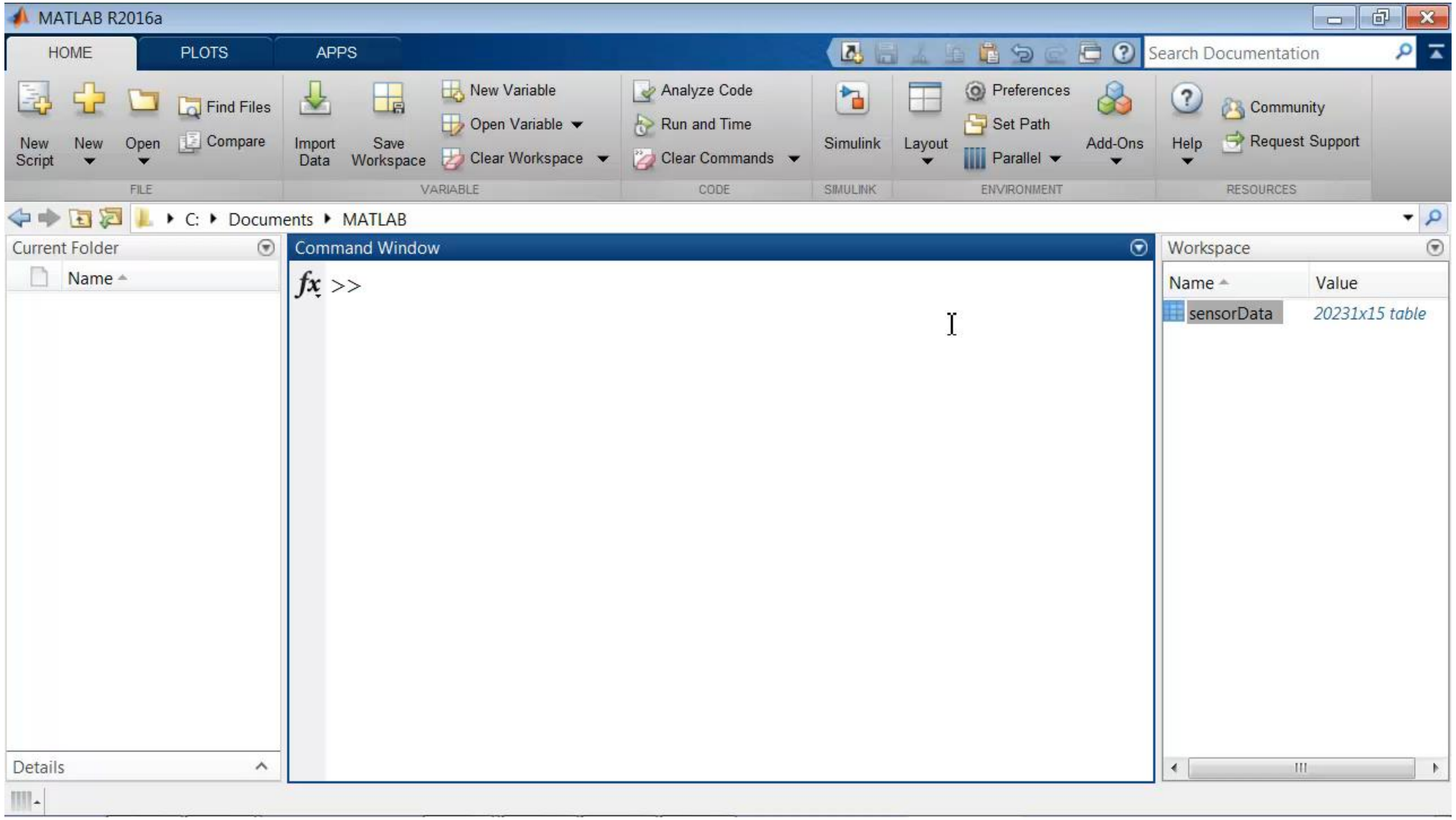


Fine tuning

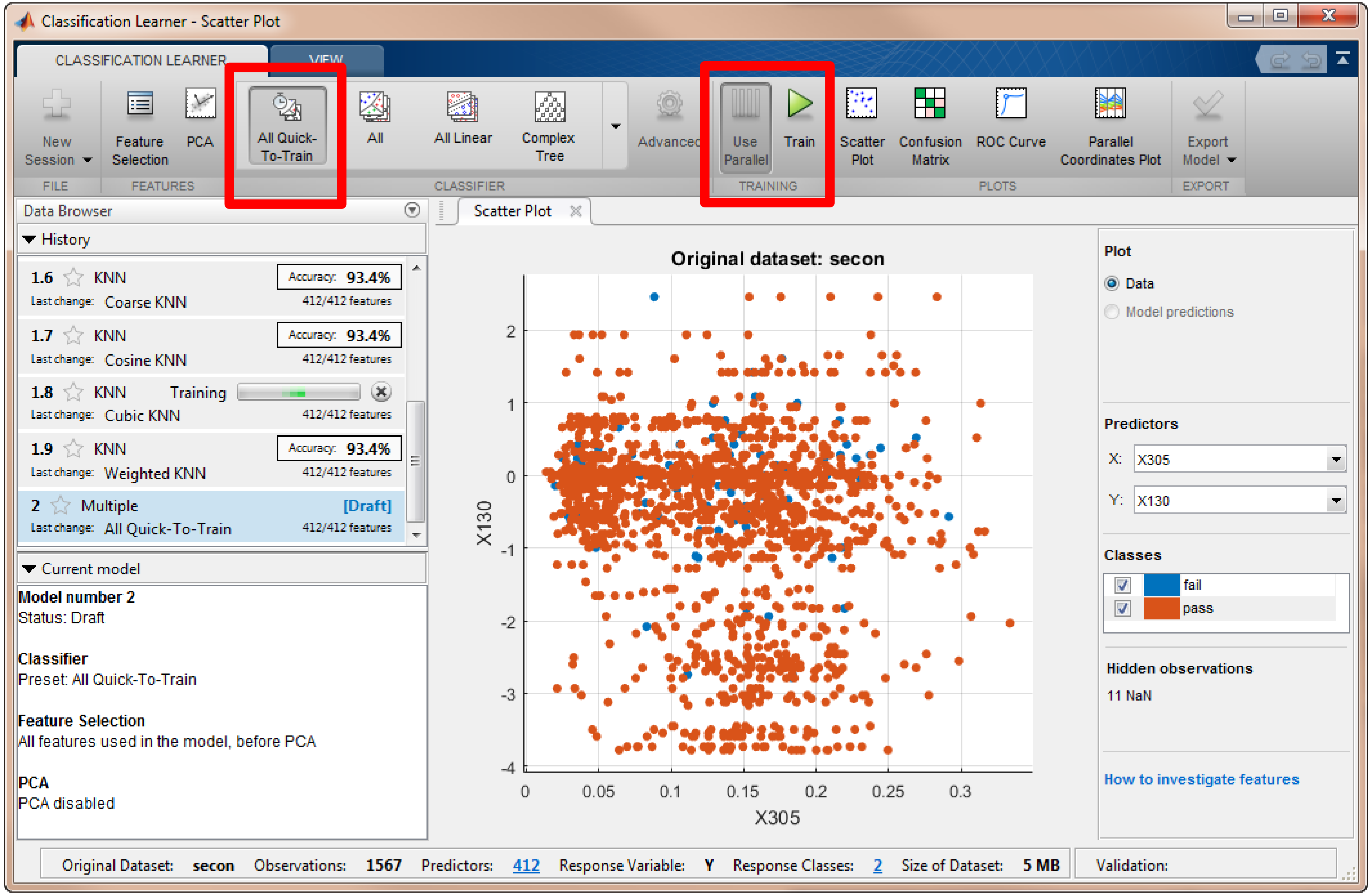


Share & Integrate

Classification Learner

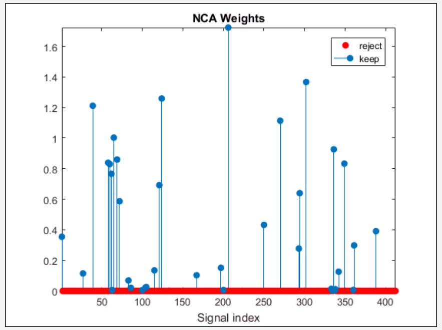


Classification Learner

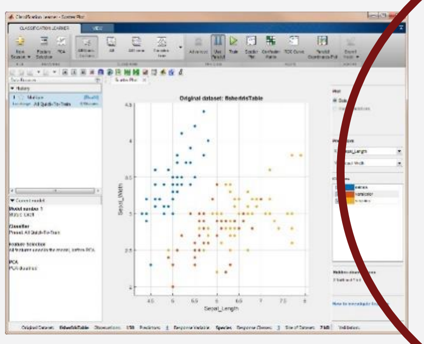


R2016a
R2016b

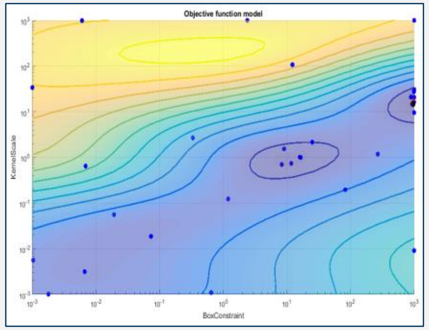
Fine Tuning a Model:
Bayesian Optimization



Which data to use



Choose a model



Fine tuning

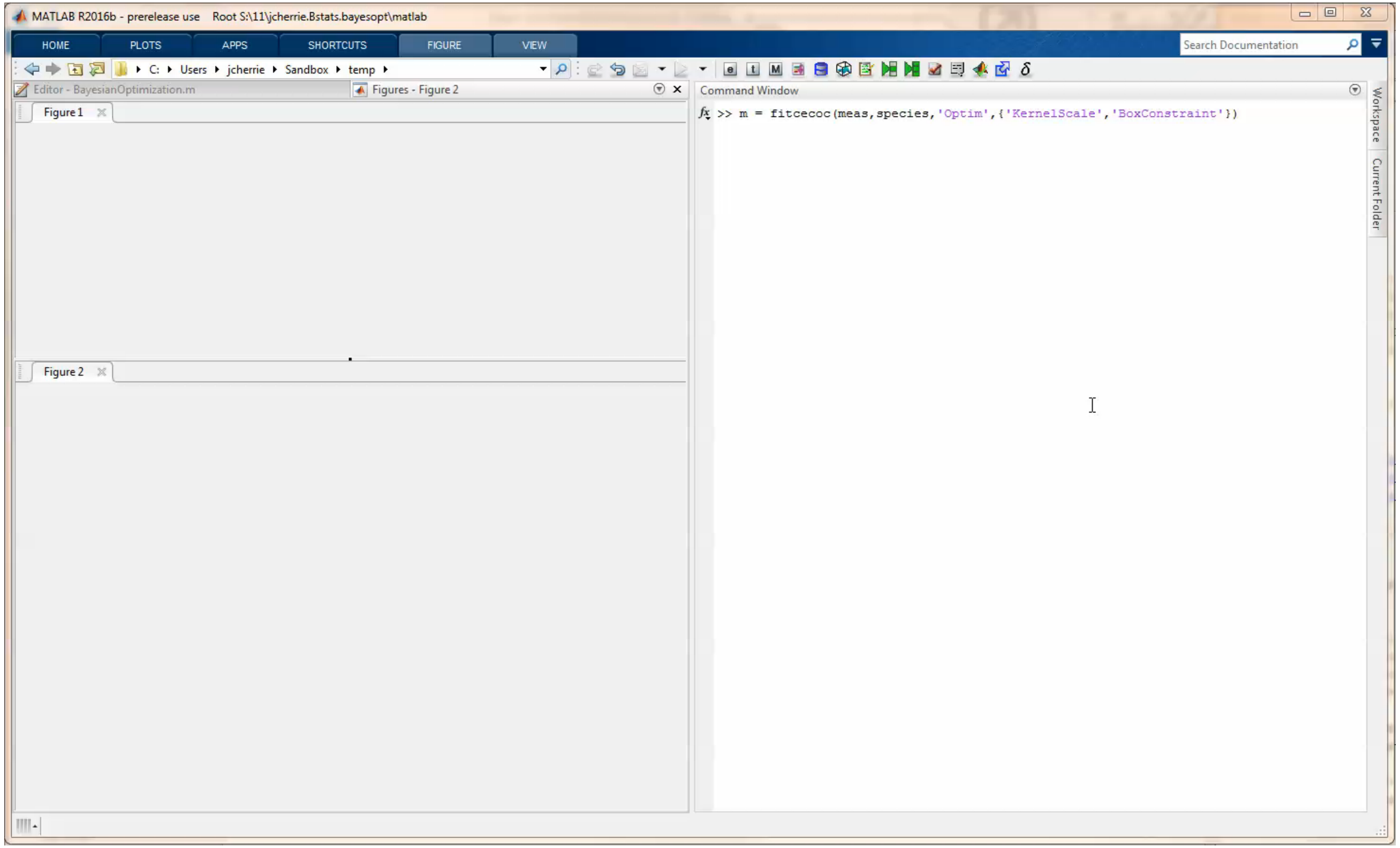
Share & Integrate

Tune Parameters with Bayesian Optimization

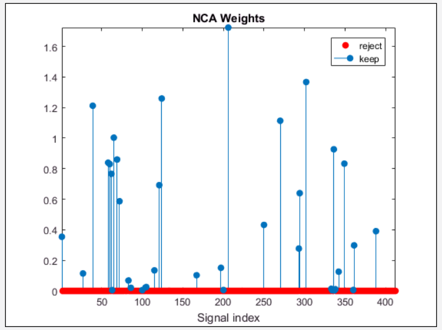
```
template = templateSVM(...
    'KernelFunction', 'linear', ...
    'PolynomialOrder', [], ...
    'KernelScale', 0.25, ...
    'BoxConstraint', 0.1, ...
    'Standardize', true);
m = fitcecoc( T, 'Species', 'Learners', template )
```

Previously tuning these parameters was a manual process

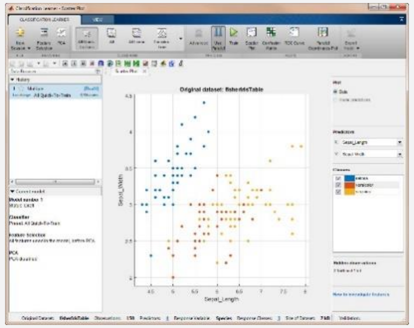
Fine tuning a model – Bayesian Optimization



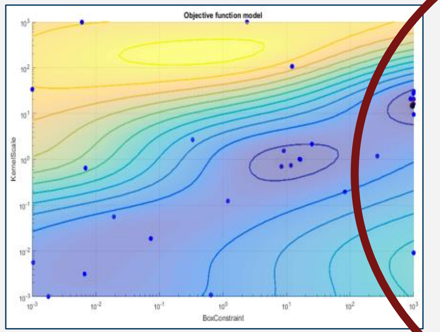
Share & Integrate



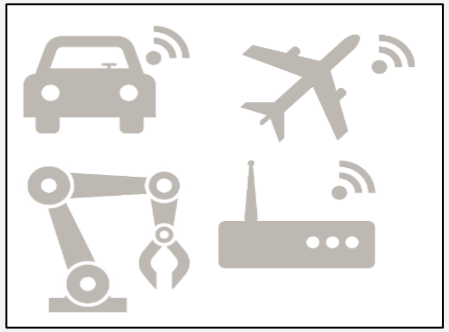
Which data to use



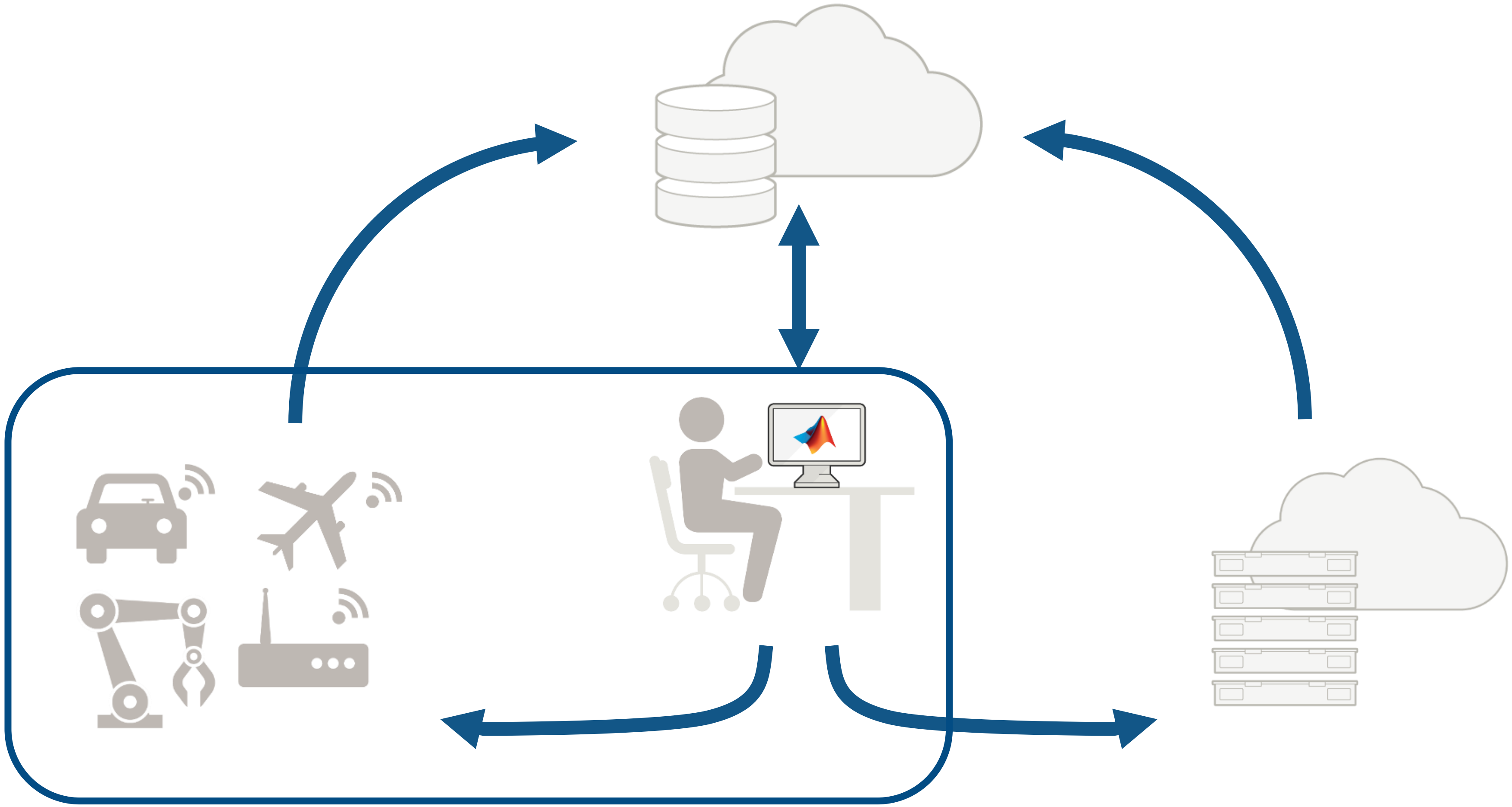
Choose a model



Fine tuning



Share & Integrate



Share & integrate: machine learning models

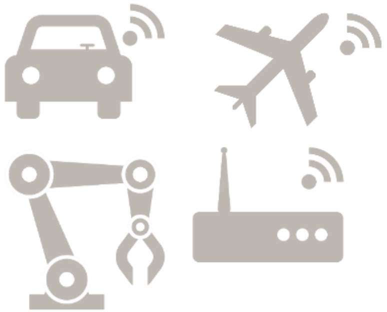
MATLAB code

```matlab
function label = classifyIonosphere(X) %#codegen
%classifyIonosphere   Classify Ionosphere based on pre-trained SVM model
mdl = loadCompactModel( 'SVMIonosphere' );
label = predict( mdl, X );
end
```

MATLAB Coder

C code

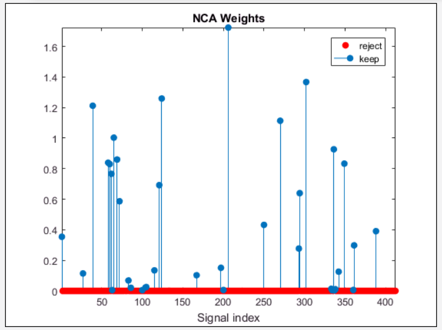
```c
/* Variable Definitions */
static emlrtRSInfo emlrtRSI = { 4,      /* lineNo */
  "classifyIonosphere",                 /* fcnName */
  "C:\\Users\\jcherrie\\Sandbox\\temp\\feature-selection\\classifyIonosphe
};

/* Function Definitions */
void classifyIonosphere(classifyIonosphereStackData *SD, const emlrtStack
  const real_T X[11934], cell_wrap_0 label[351])
{
  real_T t0_Alpha[90];
  real_T expl_temp[34];
  real_T b_expl_temp;
  char_T t0_ClassNames[2];
  real_T c_expl_temp[2];
```



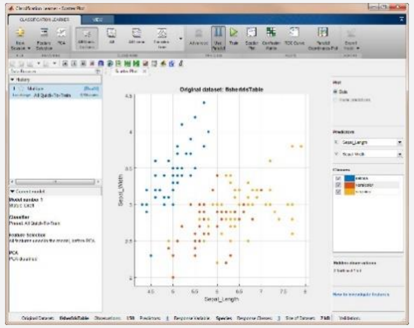
R2016b

X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'
2994.1	2548.2	2195.1	1046.1	1.3204	100	103.34	'fail'
2928.8	2479.4	2196.2	1605.8	0.9959	100	97.916	'fail'
2920.1	2507.4	2195.1	1046.1	1.3204	100	103.34	'pass'
3051.4	2529.3	2184.4	877.63	1.4668	100	107.87	'pass'
2964	2629.5	2224.6	947.77	1.2924	100	104.85	'fail'

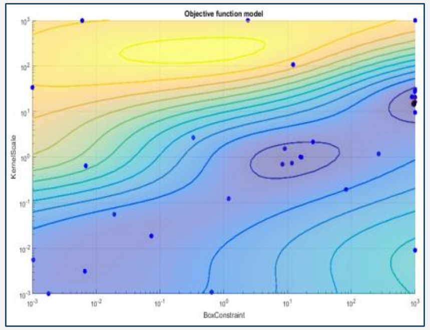
Machine Learning



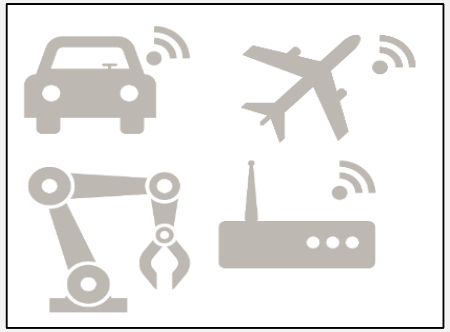
Which data to use



Choose a model



Fine tuning

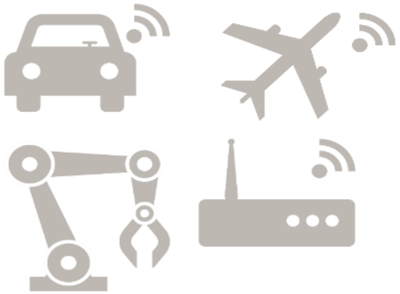


Share & Integrate

Deep Learning

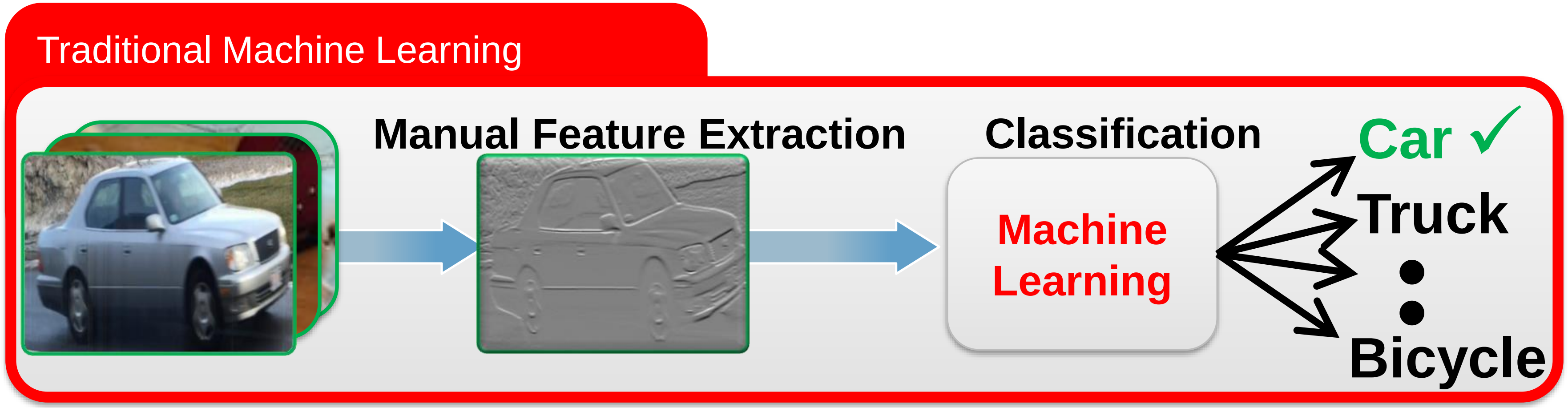
X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'
2994.1	2548.2	2195.1	1046.1	1.3204	100	103.34	'fail'
2928.8	2479.4	2196.2	1605.8	0.9959	100	97.916	'fail'
2920.1	2507.4	2195.1	1046.1	1.3204	100	103.34	'pass'
3051.4	2529.3	2184.4	877.63	1.4668	100	107.87	'pass'
2964	2629.5	2224.6	947.77	1.2924	100	104.85	'fail'

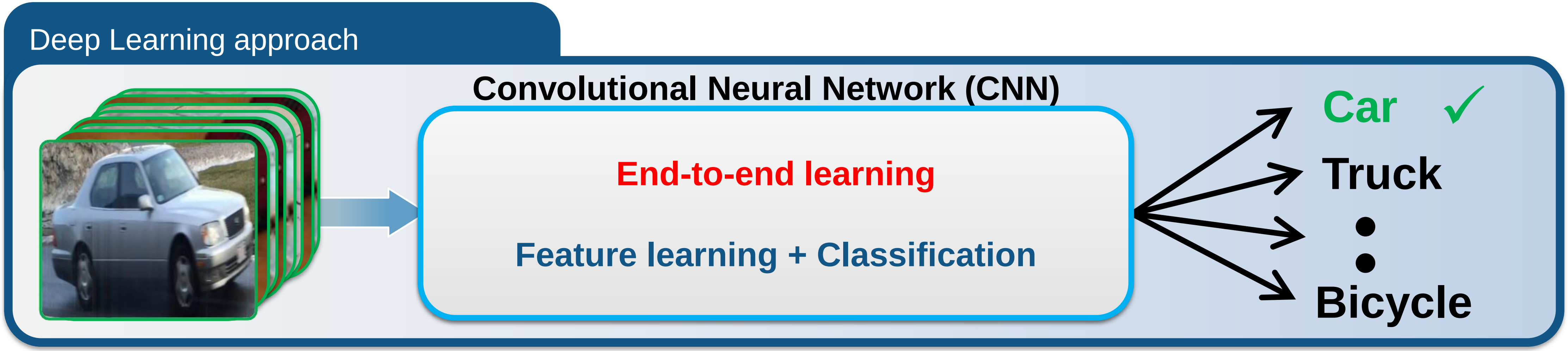
Deep Learning



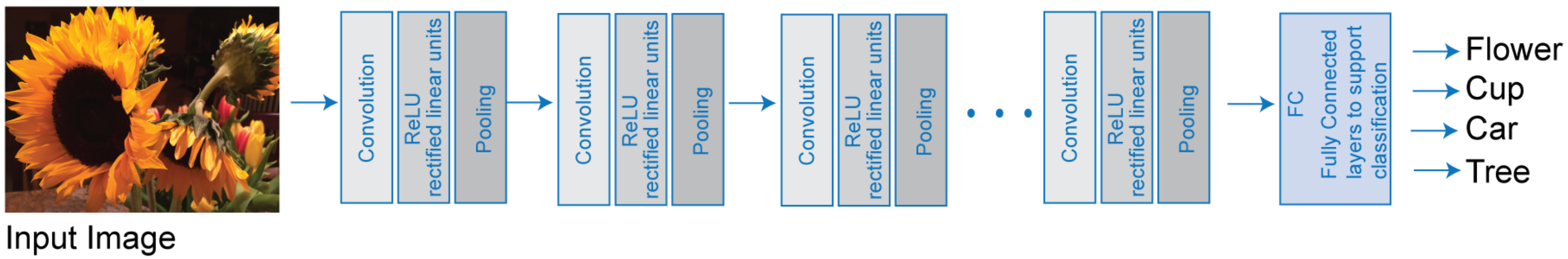
Deep Learning

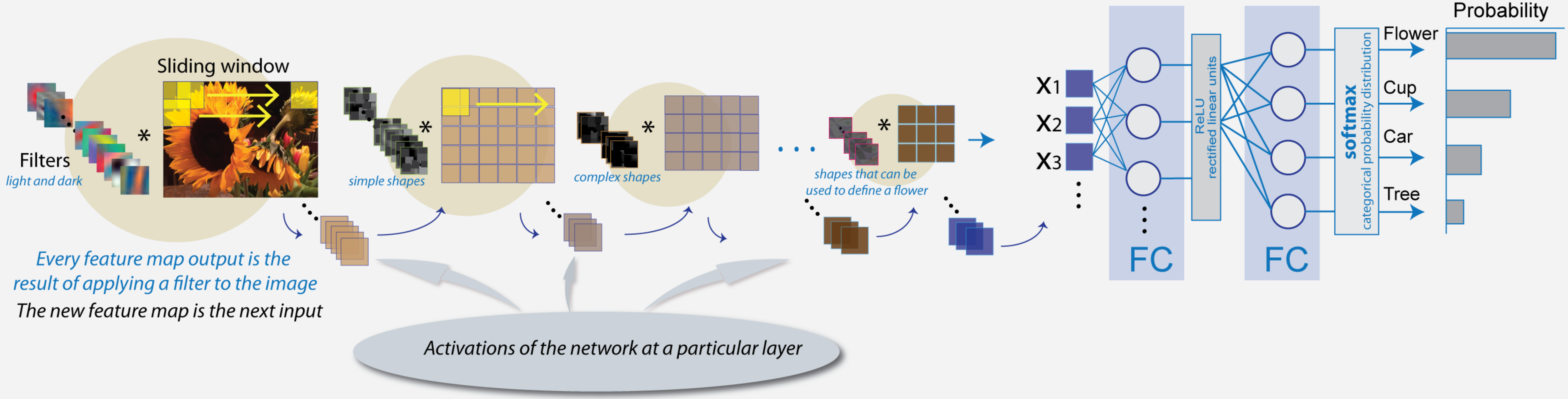
Deep learning performs **end-end learning** by learning **features, representations and tasks** directly from **images, text and sound**



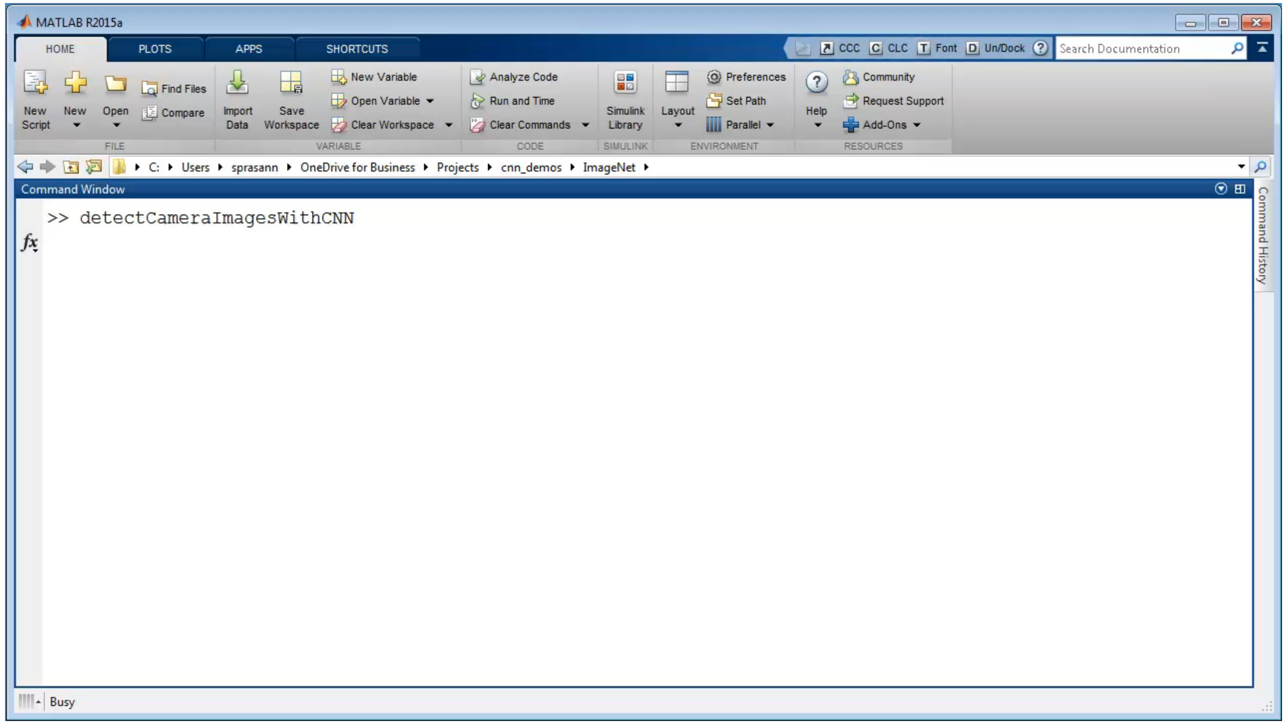


What is Deep Learning?





Object Recognition using Deep Learning



Training (using GPU)	Millions of images from 1000 different categories
Prediction	Real-time object recognition using a webcam connected to a laptop

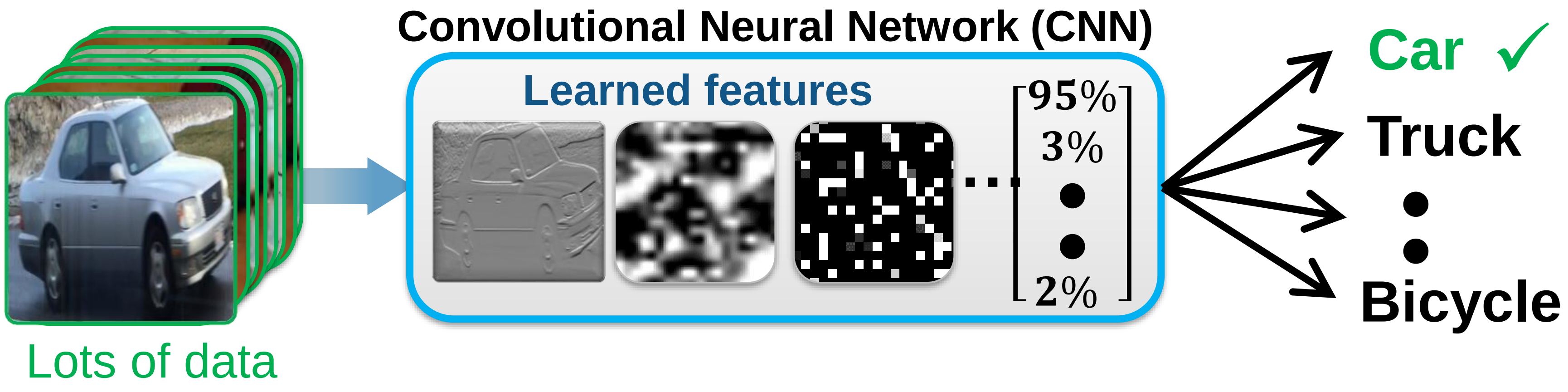
Why is Deep Learning so Popular?

- **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

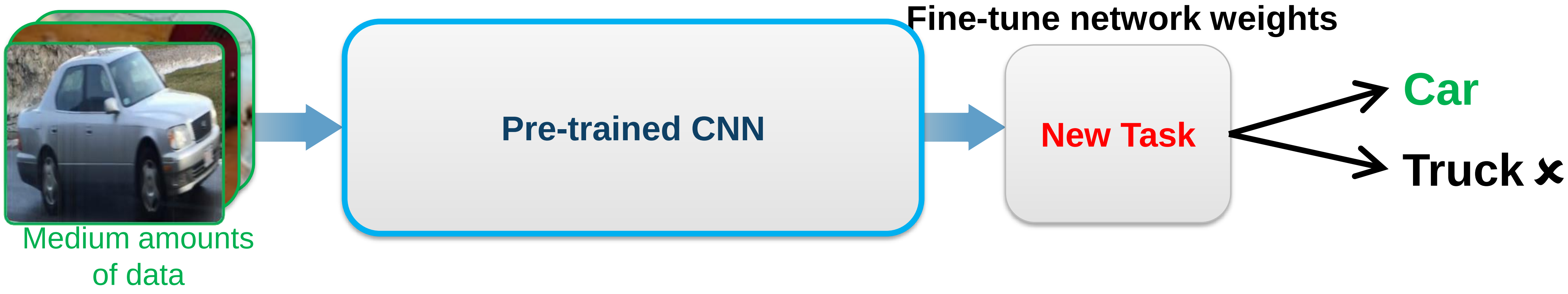
Year	Error Rate
Pre-2012 (traditional computer vision and machine learning techniques)	> 25%
2012 (Deep Learning)	~ 15%
2015 (Deep Learning)	<5 %

Two Approaches for Deep Learning

1. Train a Deep Neural Network from Scratch

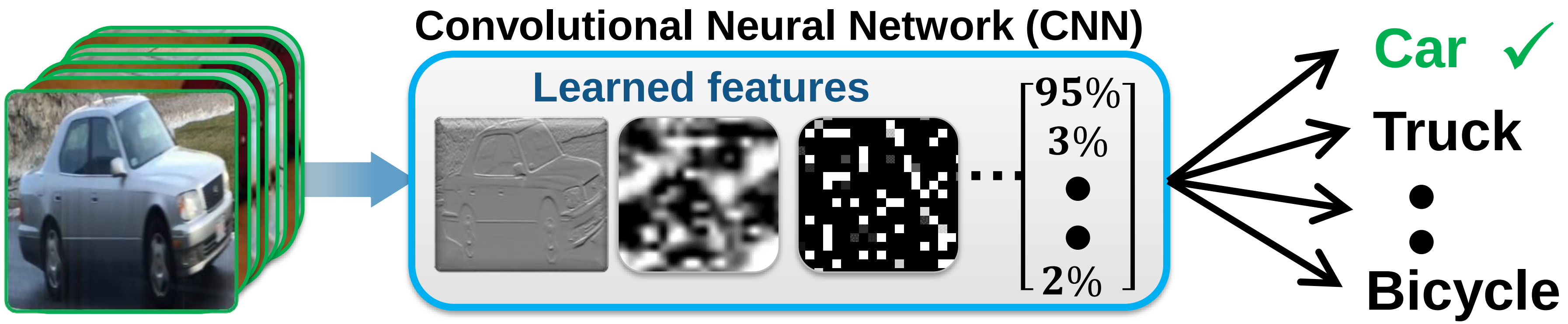


2. Fine-tune a pre-trained model (transfer learning)



Two Deep Learning Approaches

Approach 1: Train a Deep Neural Network from Scratch



Recommended <u>only</u> when:

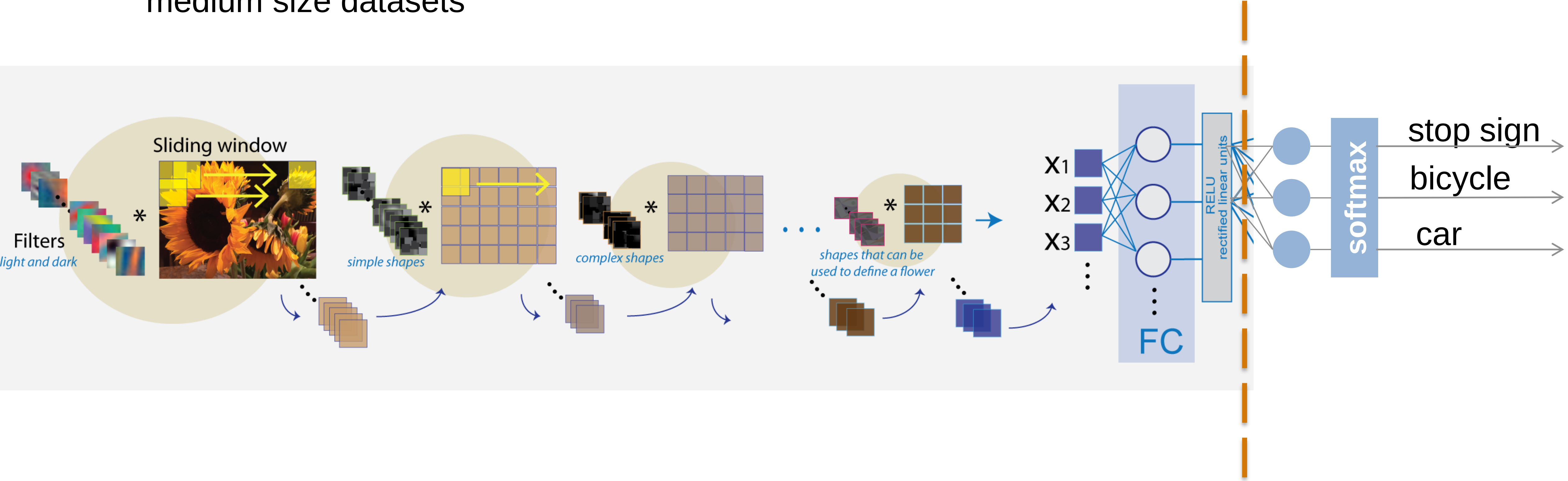
Training data	1000s to millions of labeled images
Computation	Compute intensive (requires GPU)
Training Time	Days to Weeks for real problems
Model accuracy	High (can over fit to small datasets)

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



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:

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 in MATLAB

```
% Define a CNN architecture
layers = [
    imageInputLayer([32 32 3])
    convolution2dLayer(5,32,'Padding',2)
    reluLayer()
    maxPooling2dLayer(3,'Stride',2)
      …
    fullyConnectedLayer(10)
    softmaxLayer()
    classificationLayer()
    ];

opts = trainingOptions( 'sgdm', 'InitialLearnRate', 0.001 );

% Train the CNN
[net, info] = trainNetwork(X, Y, layers, opts);
```

R2016a

Transfer Learning in MATLAB

```
% Everything except the last 3 layers.
preTrainedLayers = preTrainedNetwork.Layers(1:end-3);

% Add new fully connected layer for 2 categories,
% the softmax layer, and the classification layer which make up the
% remaining portion of the networks classification layers.
layers = [
    preTrainedLayers
    fullyConnectedLayer(2)
    softmaxLayer
    classificationLayer()
];

net = trainNetwork(X, Y, layers, opts);
```

R2016a

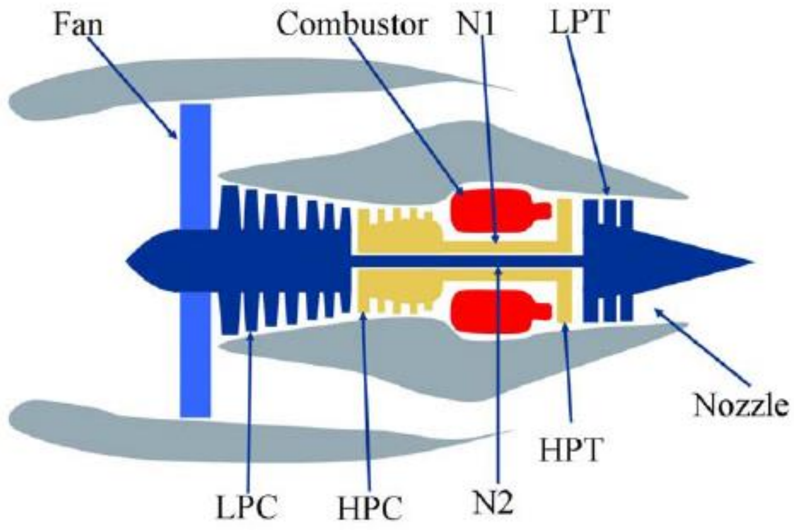
Demo Stations

1. Classification Learner Demo
Predictive Maintenance of Turbofan Engine

Sensor data from 100 engines of the same model

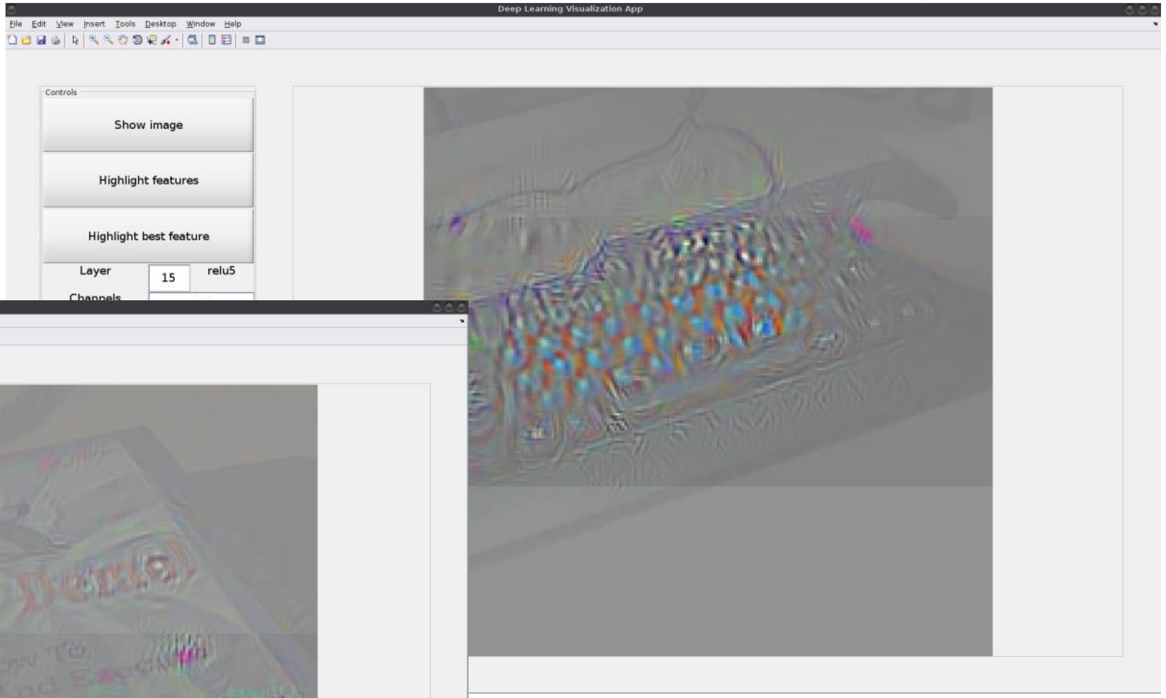
Predict and fix failures before they arise

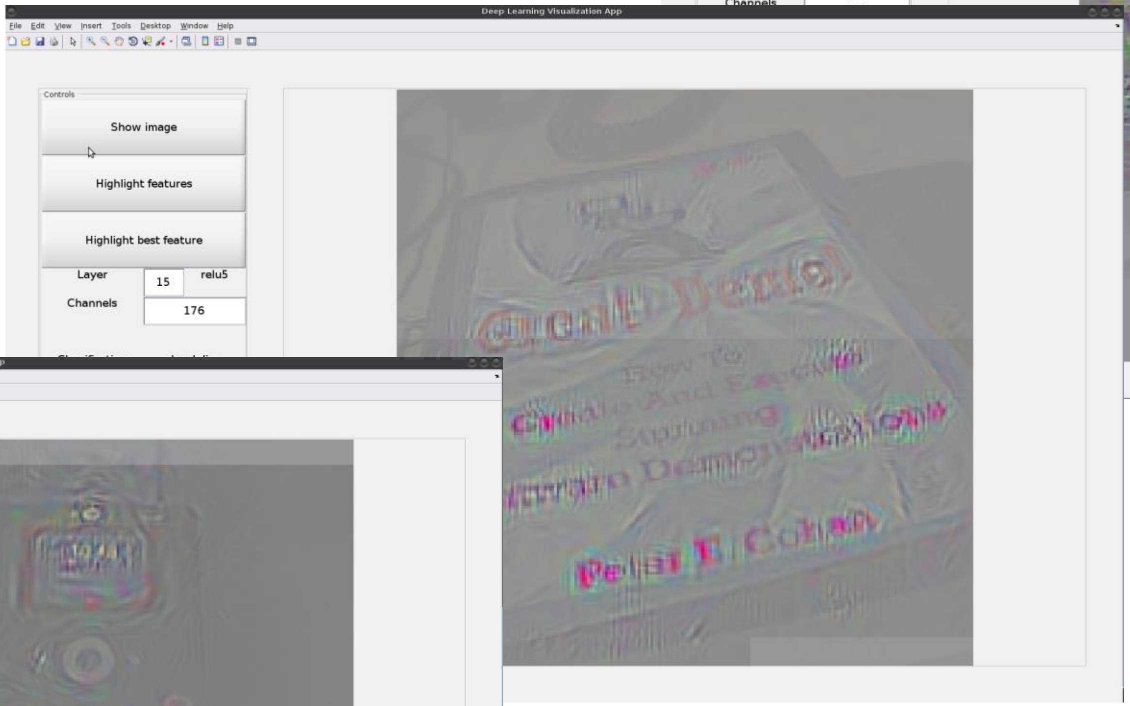
- Import and analyze historical sensor data
- Train model to predict when failures will occur
- Deploy model to run on live sensor data
- Predict failures in real time

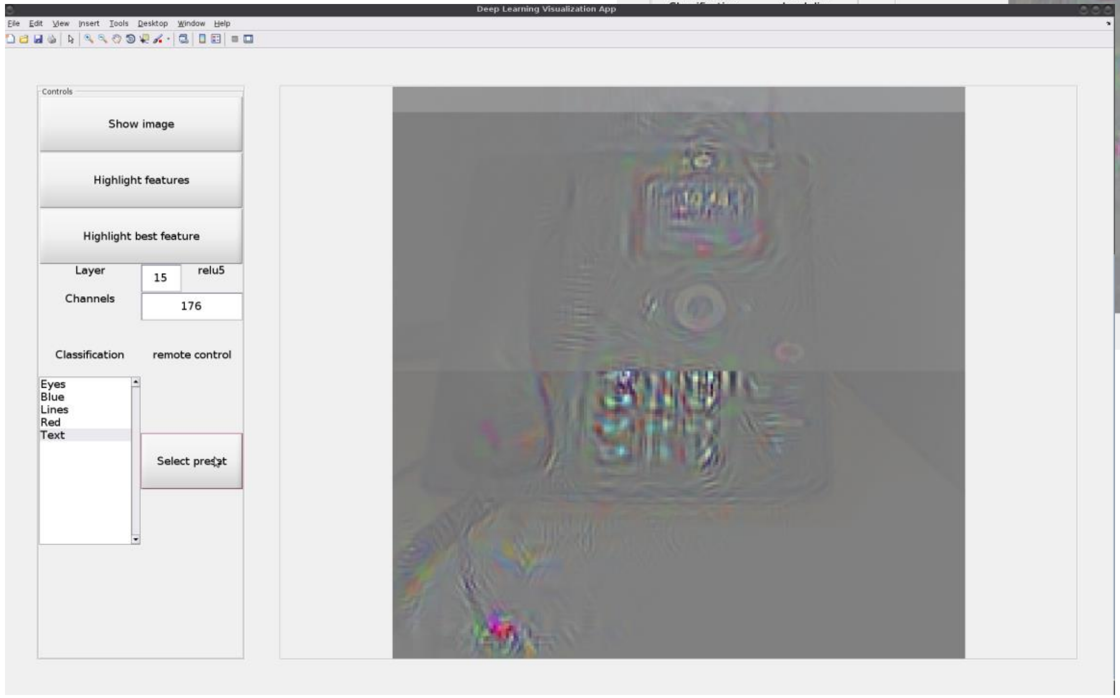


Data provided by NASA PCoE
http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/

2. Deep Learning Demo

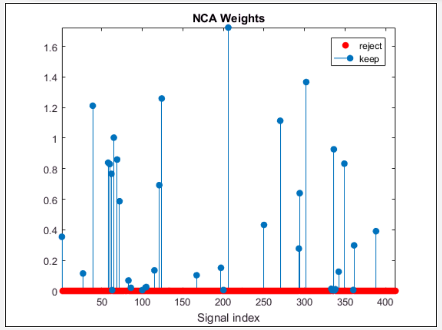




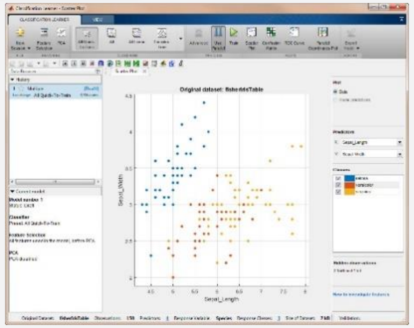


X1	X2	X3	X4	X5	X6	X7	Y
3030.9	2564	2187.7	1411.1	1.3602	100	97.613	'pass'
3095.8	2465.1	2230.4	1463.7	0.8294	100	102.34	'pass'
2932.6	2559.9	2186.4	1698	1.5102	100	95.488	'fail'
2988.7	2479.9	2199	909.79	1.3204	100	104.24	'pass'
3032.2	2502.9	2233.4	1326.5	1.5334	100	100.4	'pass'
2946.3	2432.8	2233.4	1326.5	1.5334	100	100.4	'pass'
3030.3	2430.1	2230.4	1463.7	0.8294	100	102.34	'pass'
3058.9	2690.2	2248.9	1004.5	0.7884	100	106.24	'pass'
2967.7	2600.5	2248.9	1004.5	0.7884	100	106.24	'pass'
3016.1	2428.4	2248.9	1004.5	0.7884	100	106.24	'pass'
2994.1	2548.2	2195.1	1046.1	1.3204	100	103.34	'fail'
2928.8	2479.4	2196.2	1605.8	0.9959	100	97.916	'fail'
2920.1	2507.4	2195.1	1046.1	1.3204	100	103.34	'pass'
3051.4	2529.3	2184.4	877.63	1.4668	100	107.87	'pass'
2964	2629.5	2224.6	947.77	1.2924	100	104.85	'fail'

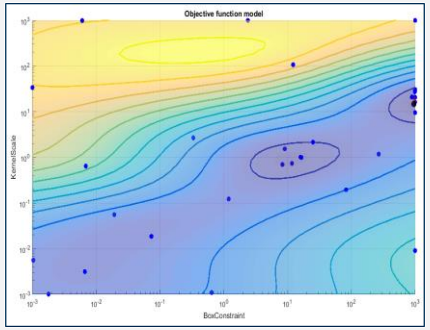
Machine Learning



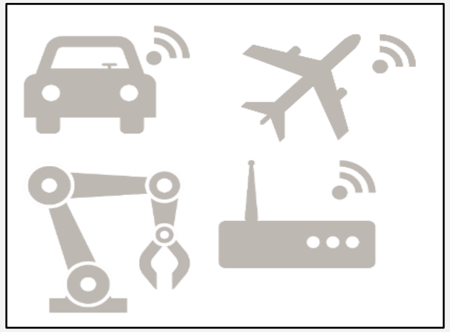
Which data to use



Choose a model

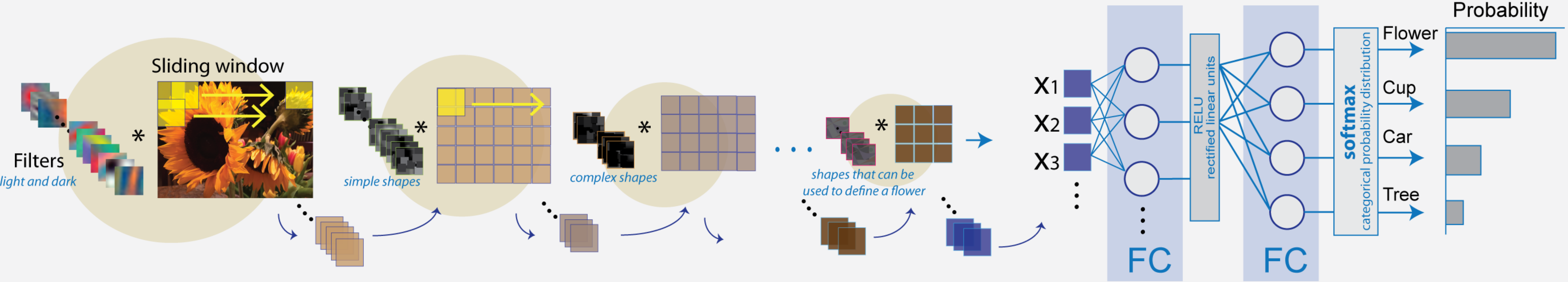


Fine tuning



Share & Integrate

Deep Learning



FIN