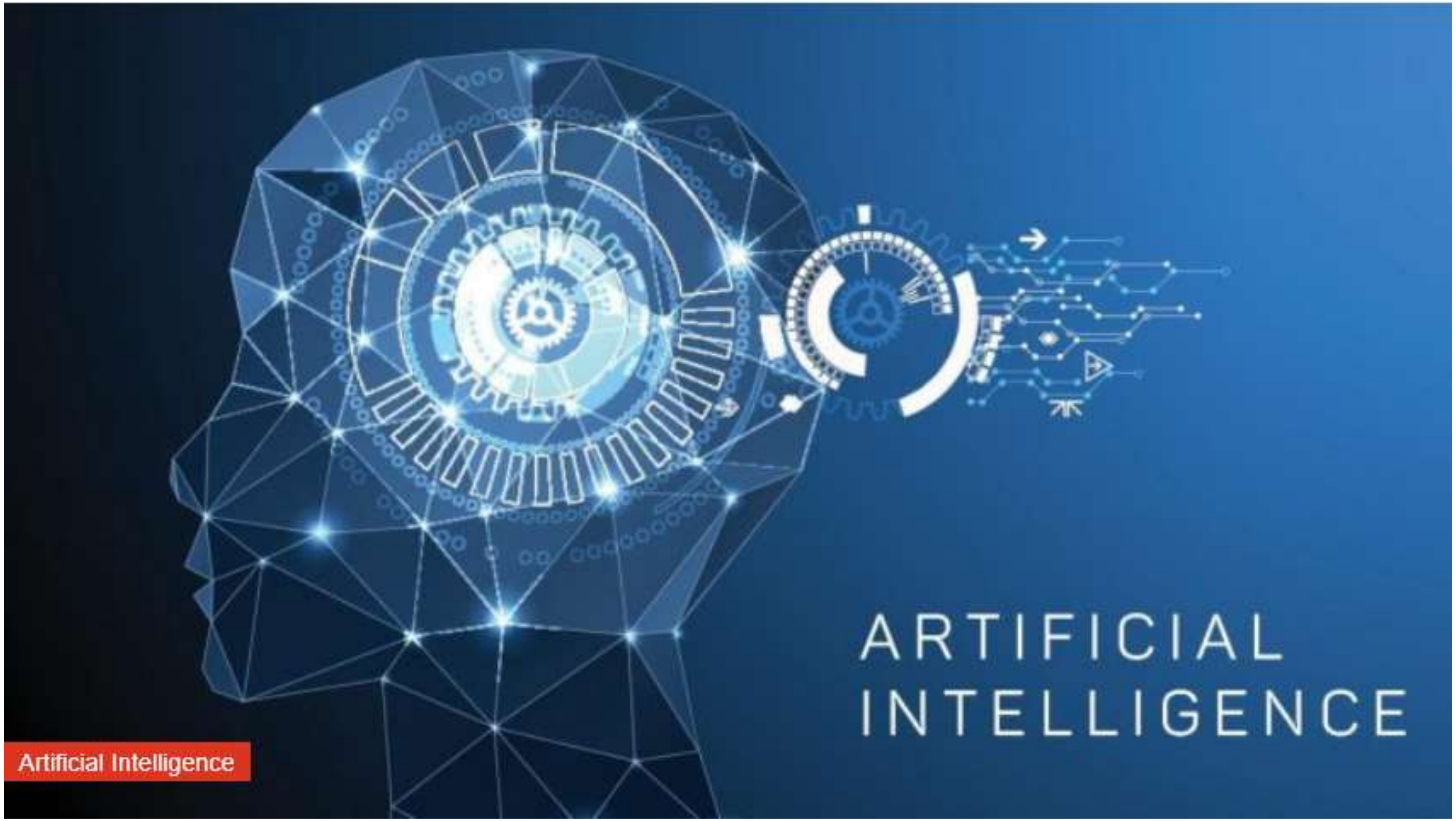
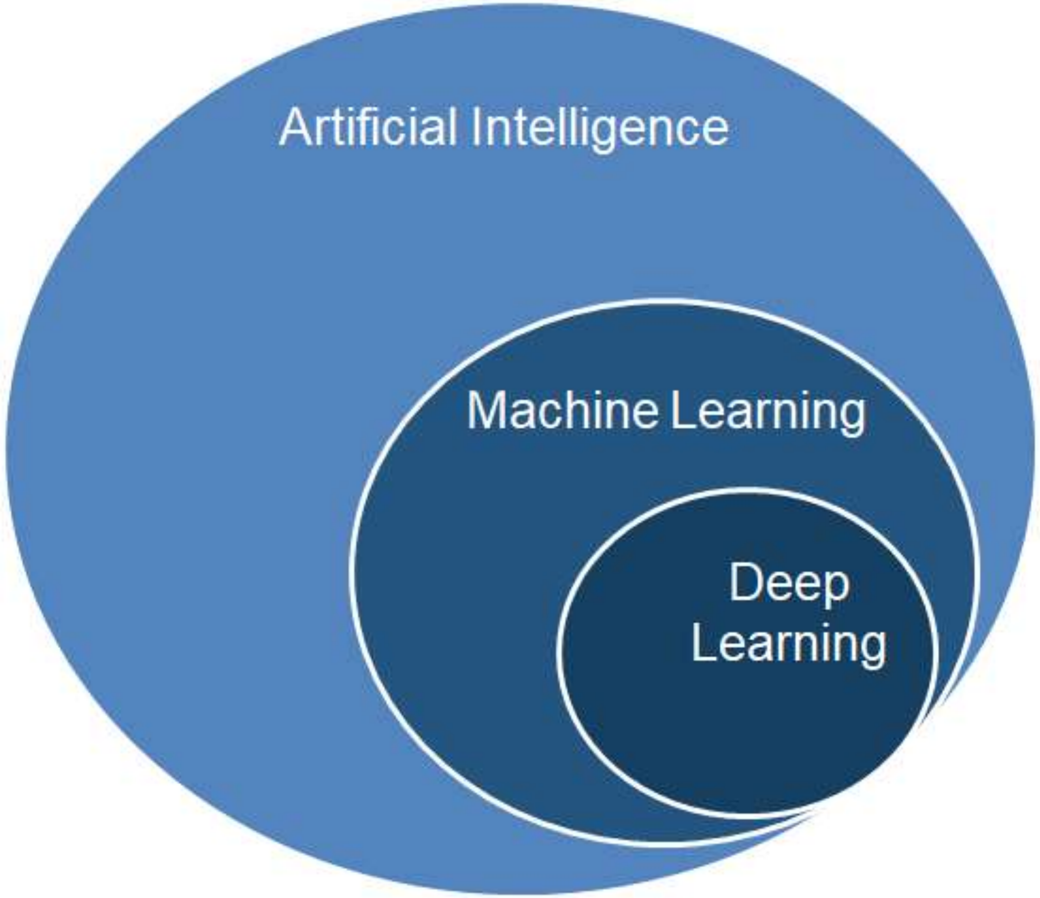
AI를 위한 데이터 활용 및
Machine Learning & Deep Learning Workflow

Artificial Intelligence

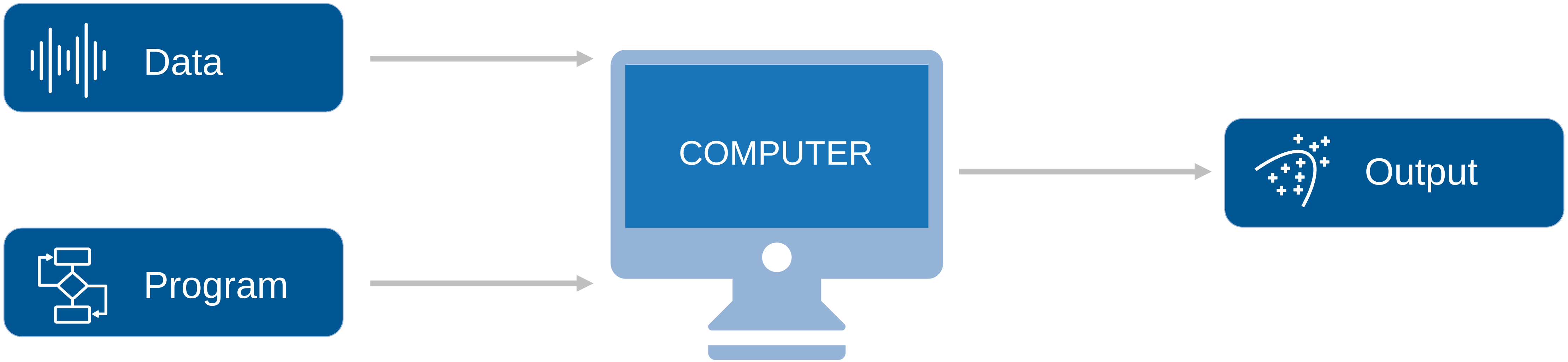


MATLAB for Artificial Intelligence

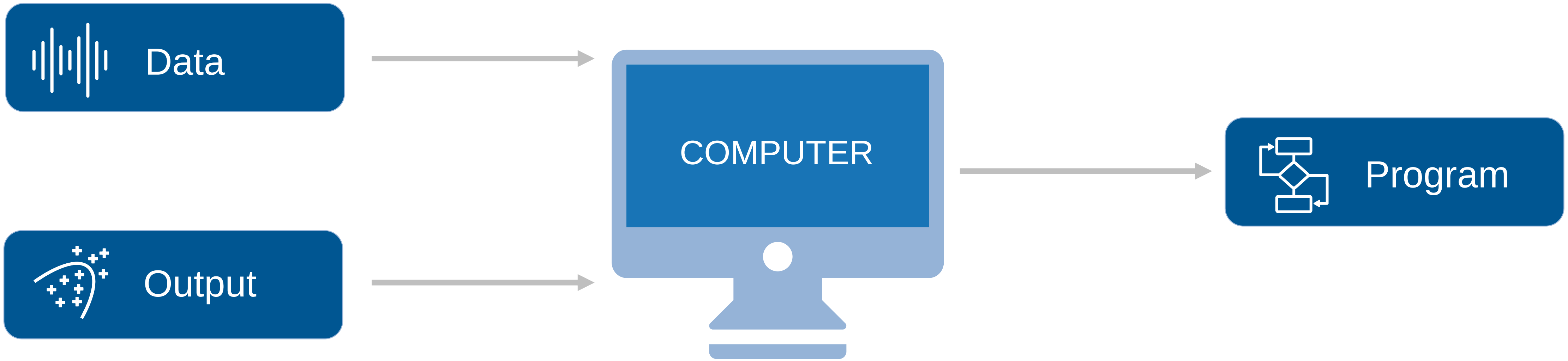


- Machine Learning
- Deep Learning
- Reinforcement Learning
- Predictive Maintenance
- Data Science / Data Analytics
- Signal Processing
- Image Processing
- …and more

There are two ways to get a computer to do what you want

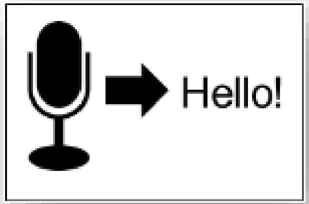


There are two ways to get a computer to do what you want



Machine Learning is Everywhere

Solution is too complex for hand written rules or equations



Speech Recognition

Object Recognition

Engine Health Monitoring

learn complex non-linear relationships

Solution needs to adapt with changing data

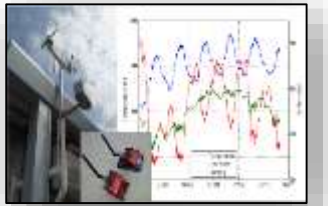
Weather Forecasting

Energy Load Forecasting

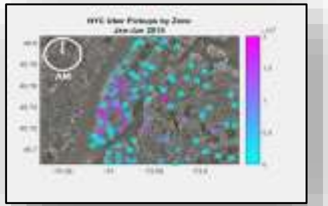
Stock Market Prediction

update as more data becomes available

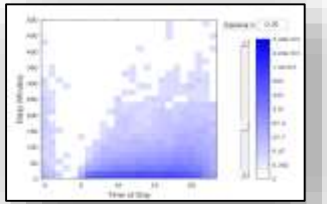
Solution needs to scale



IoT Analytics



Taxi Availability

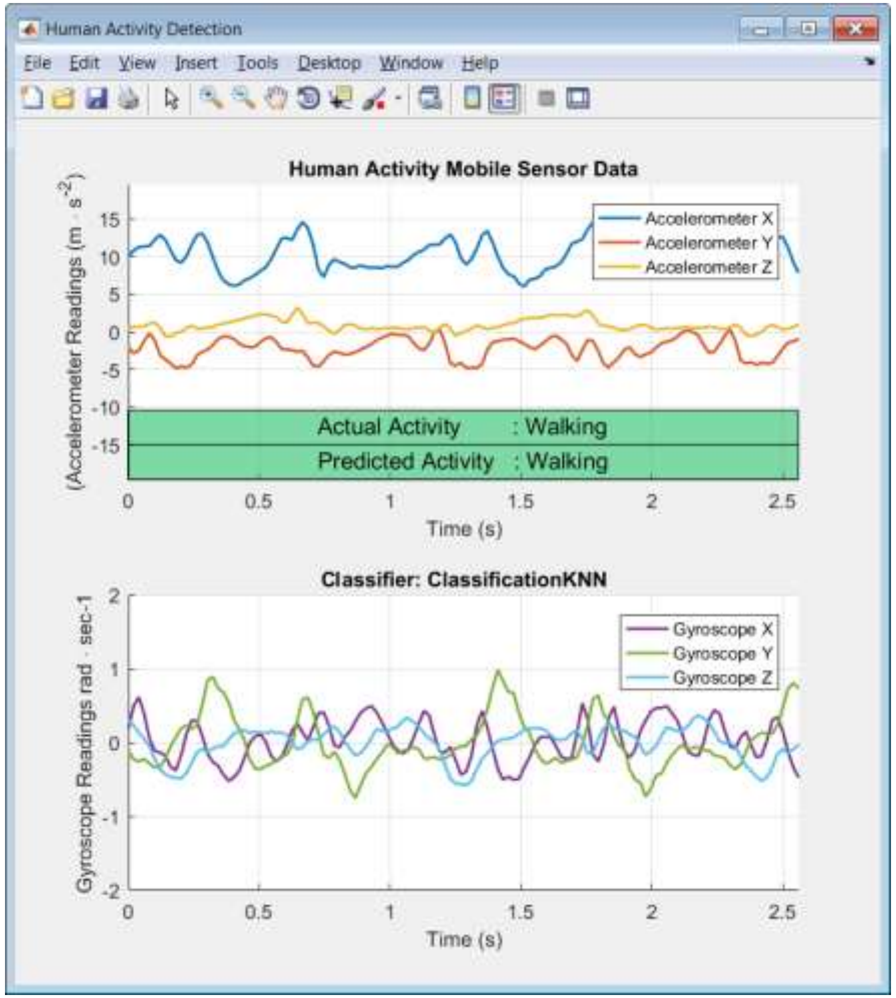


Airline Flight Delays

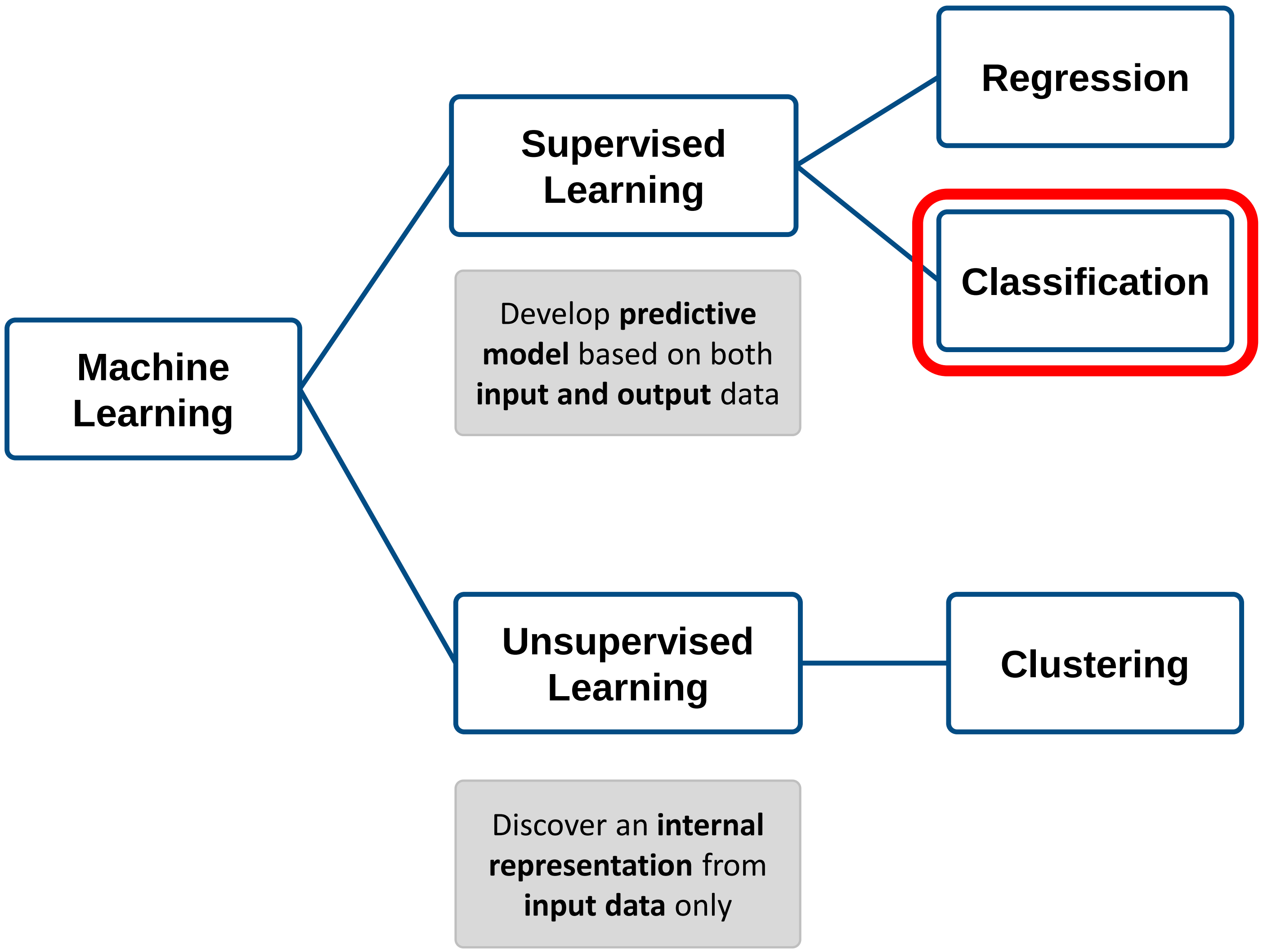
learn efficiently from very large data sets

Agenda 1

- What is Machine Learning?
- Supervised Learning
  - Feature Engineering
  - Model Selection and Training
  - Optimization and AutoML
- Unsupervised Learning
- Deployment
- Resources



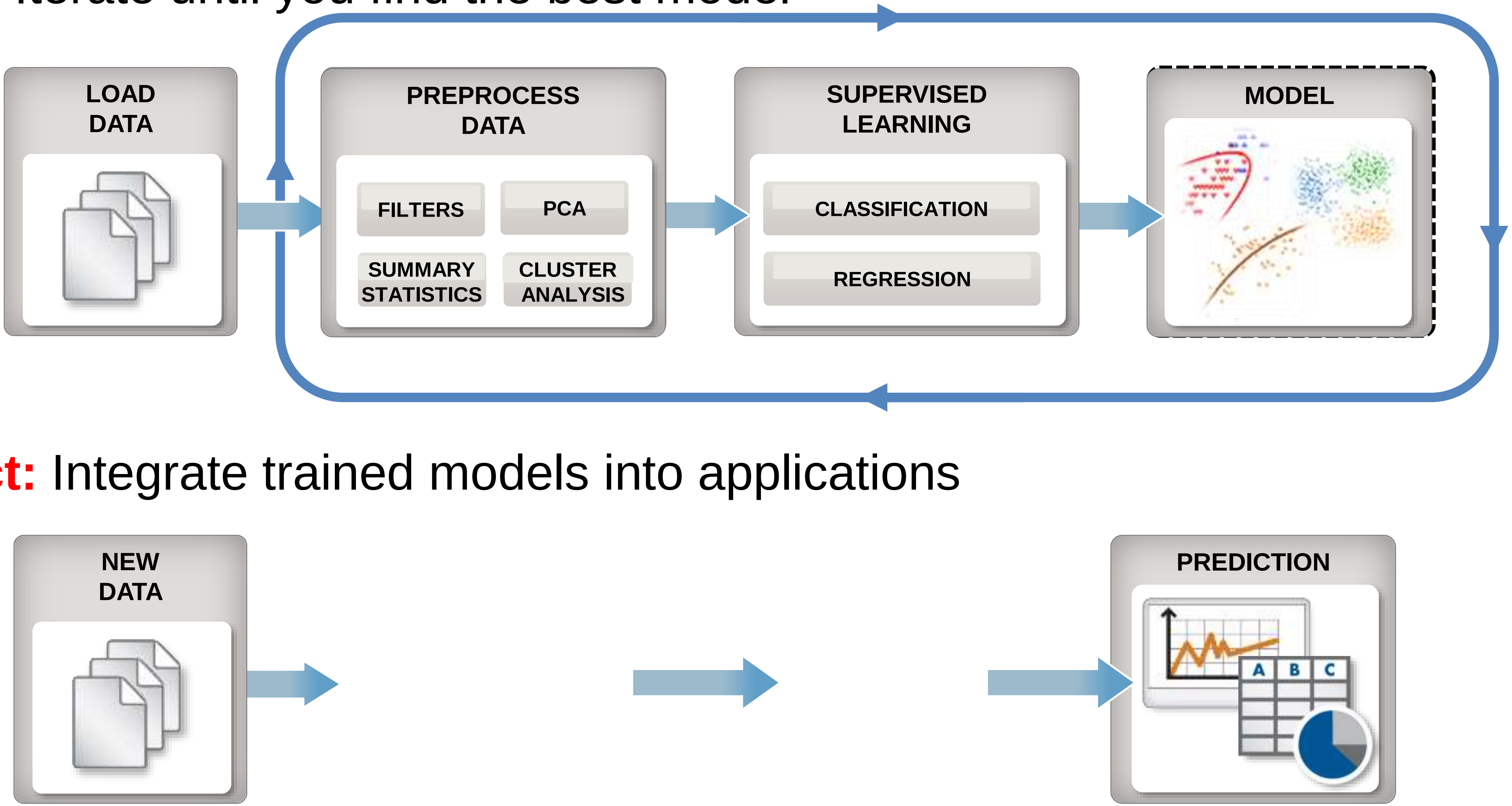
Types of Machine Learning



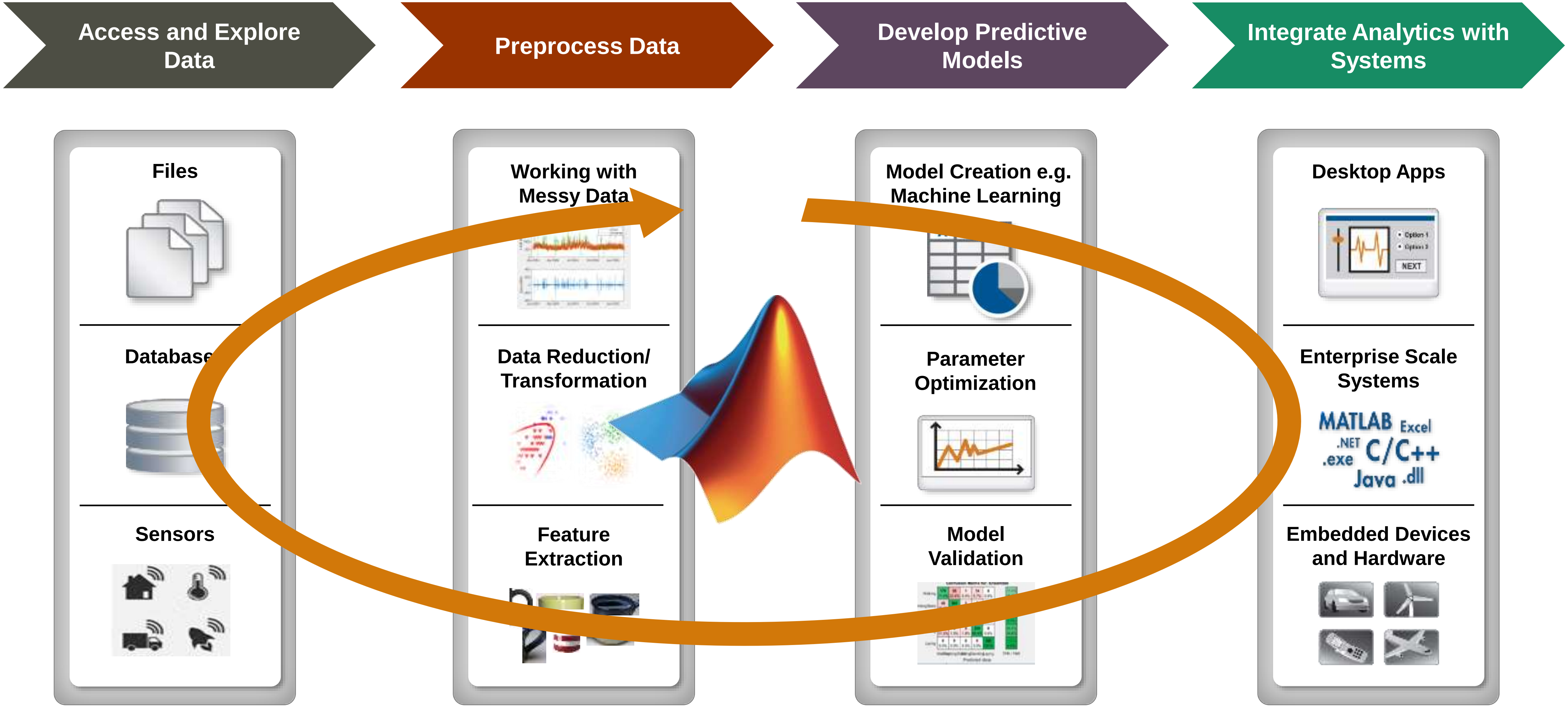
Training a predictive model

Train: Iterate until you find the best model

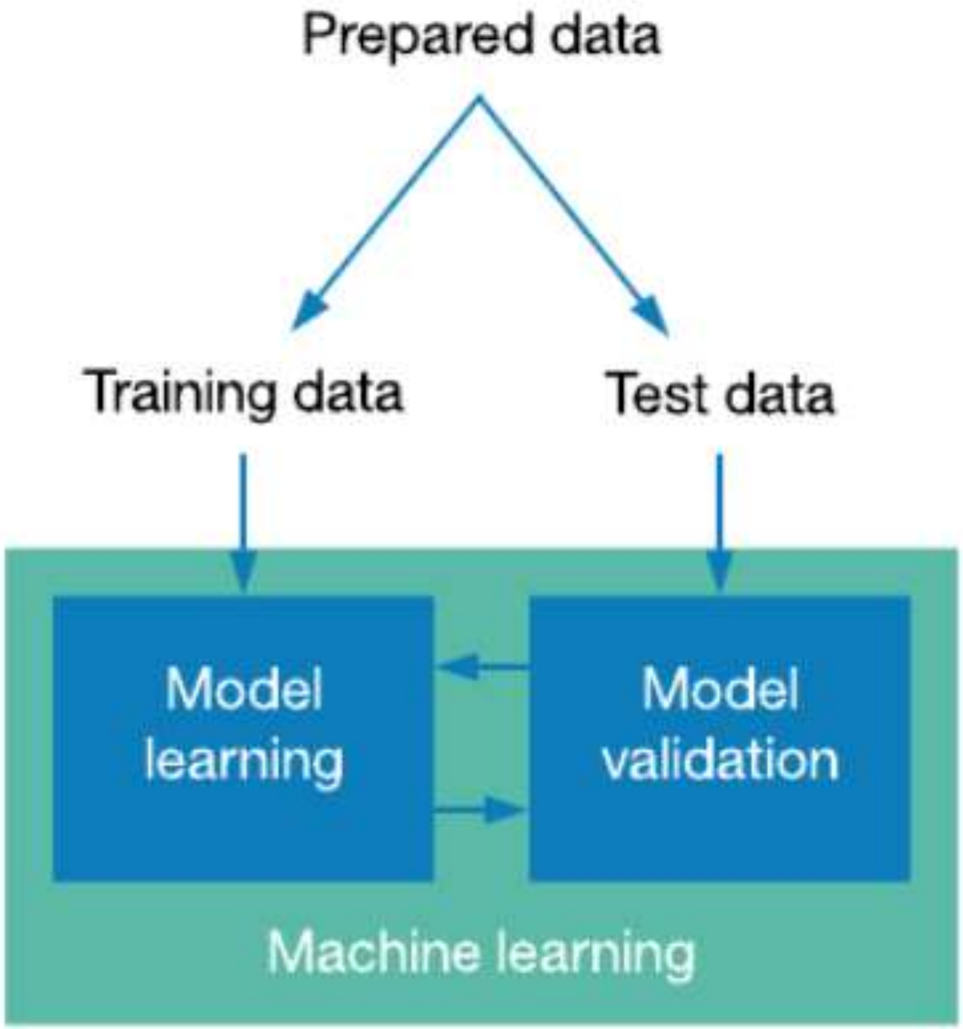
Predict: Integrate trained models into applications



Machine Learning Workflow

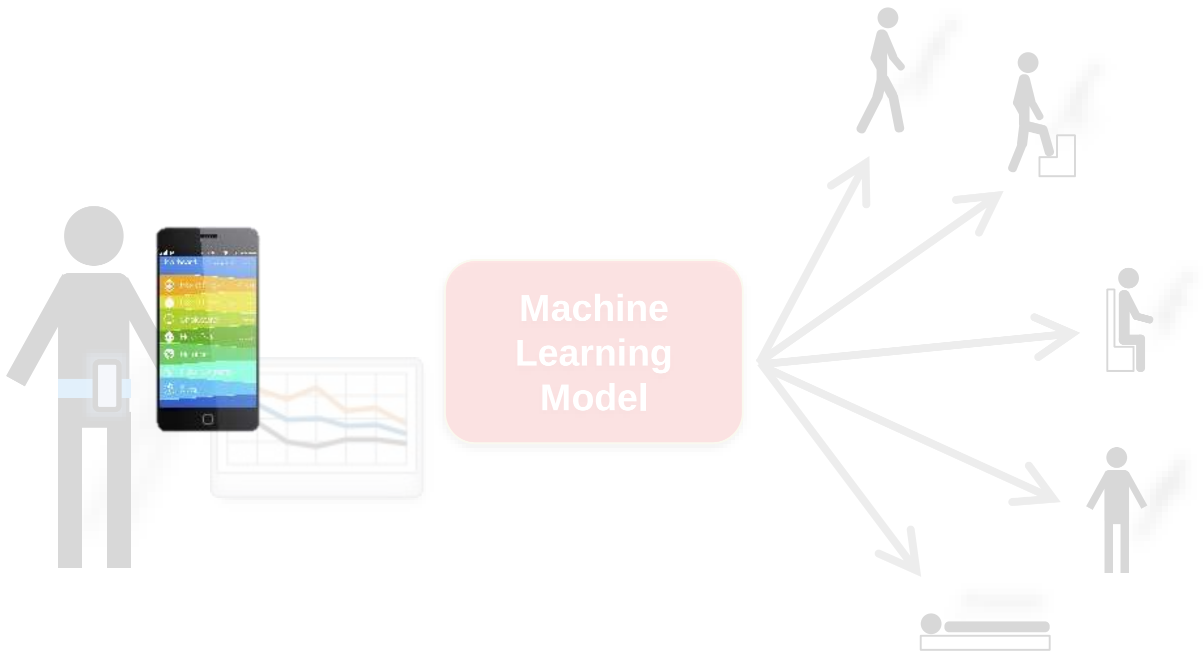


Data Preparation



Demo Example 1: Human Activity Recognition

Classification



Data:

- 3-axial Accelerometer data
- 3-axial Gyroscope data

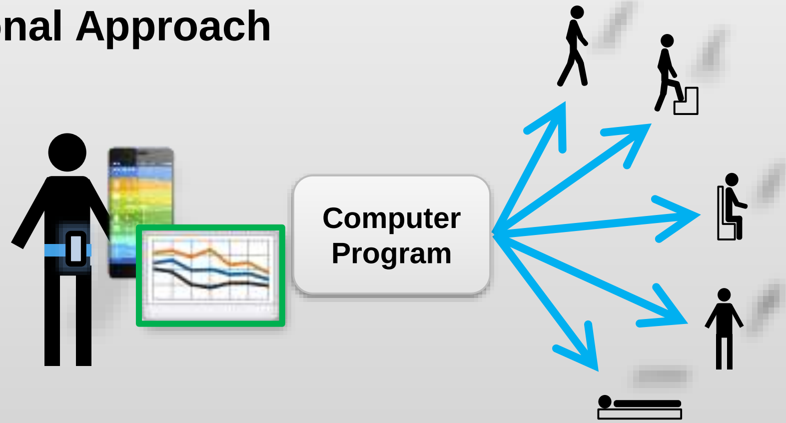
Dataset courtesy of:
Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra and Jorge L. Reyes-Ortiz.
Human Activity Recognition on Smartphones using a Multiclass Hardware-Friendly Support Vector Machine.
International Workshop of Ambient Assisted Living (IWAAL 2012). Vitoria-Gasteiz, Spain. Dec 2012
http://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

What is Machine Learning?

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

Task: Human Activity Detection

Traditional Approach



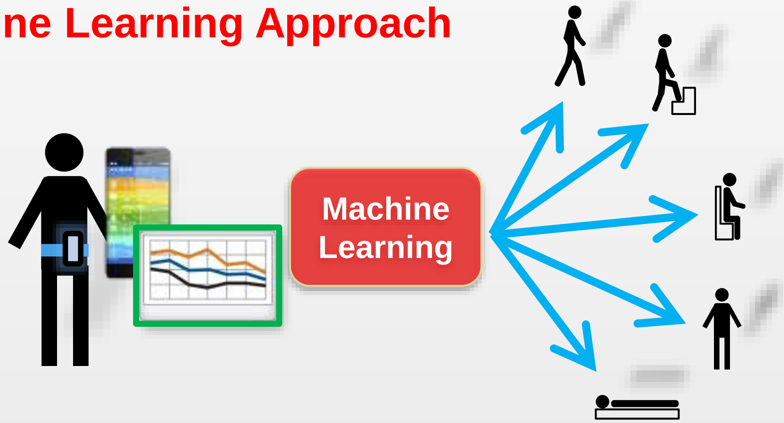
Hand Written Program

If X_acc > 0.5
then "SITTING"
If Y_acc < 4 and 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

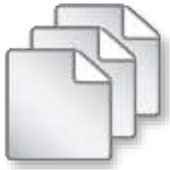


$$model: \text{Inputs} \rightarrow \text{Outputs}$$

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

Machine Learning Workflow – step 1

Access and Explore Data → Preprocess Data → Develop Predictive Models → Integrate Analytics with Systems

Access and Explore Data	Preprocess Data	Develop Predictive Models	Integrate Analytics with Systems
Files 	Working with Messy Data	Model Creation e.g. Machine Learning	Desktop Apps 
Databases 	Data Reduction/ Transformation	Parameter Optimization 	Enterprise Scale Systems
Sensors 	Feature Extraction 	Model Validation	Embedded Devices and Hardware 

Machine Learning Workflow – step 2

Access and Explore Data → **Preprocess Data** → Develop Predictive Models → Integrate Analytics with Systems

Access and Explore Data

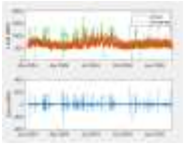
Files

Databases

Sensors

Preprocess Data

Working with Messy Data



Data Reduction/ Transformation

Feature Extraction

Develop Predictive Models

Model Creation e.g. Machine Learning

Parameter Optimization

Model Validation

Integrate Analytics with Systems

Desktop Apps

Enterprise Scale Systems

Embedded Devices and Hardware

Feature Engineering

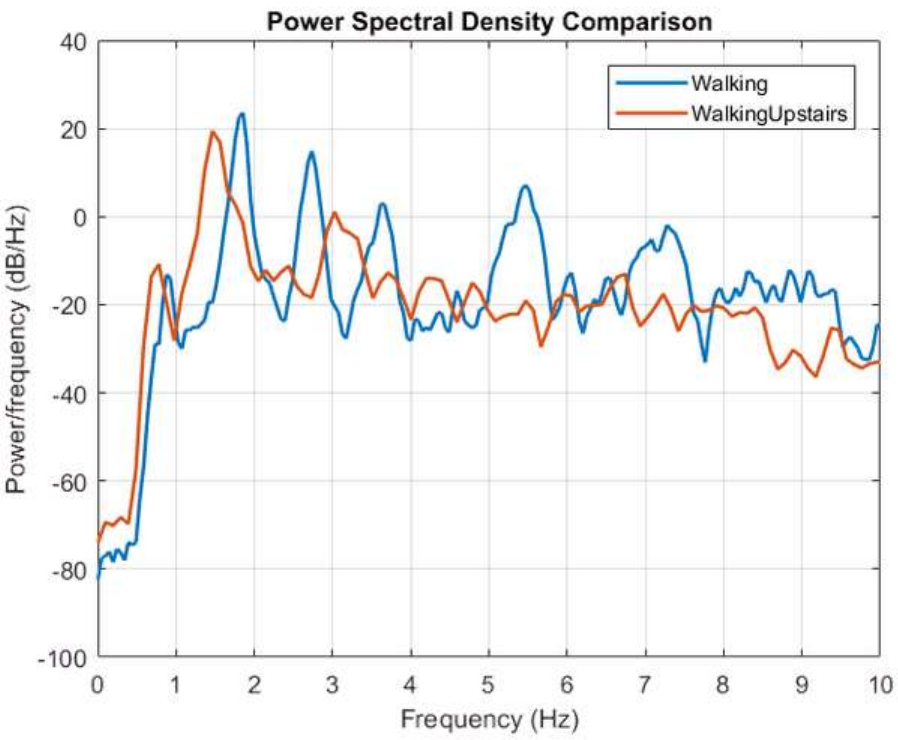
Using domain knowledge to create features for machine learning algorithms

Feature transformation: **Reduce dimensionality**

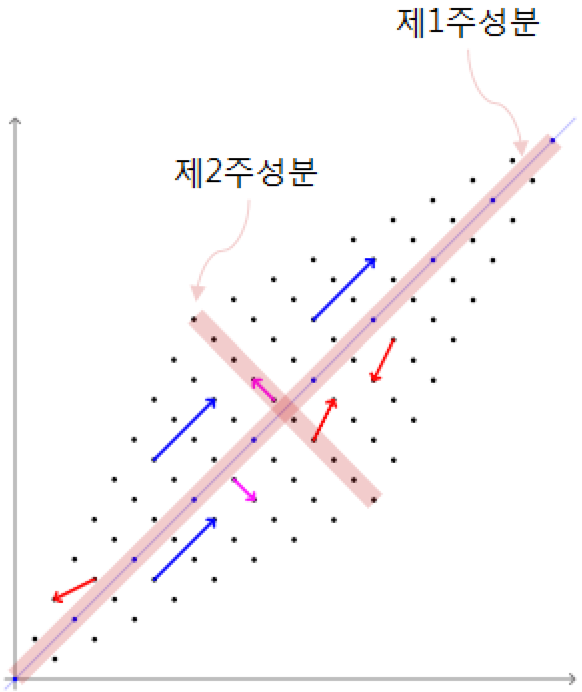
Feature selection: Choose subset of most relevant features

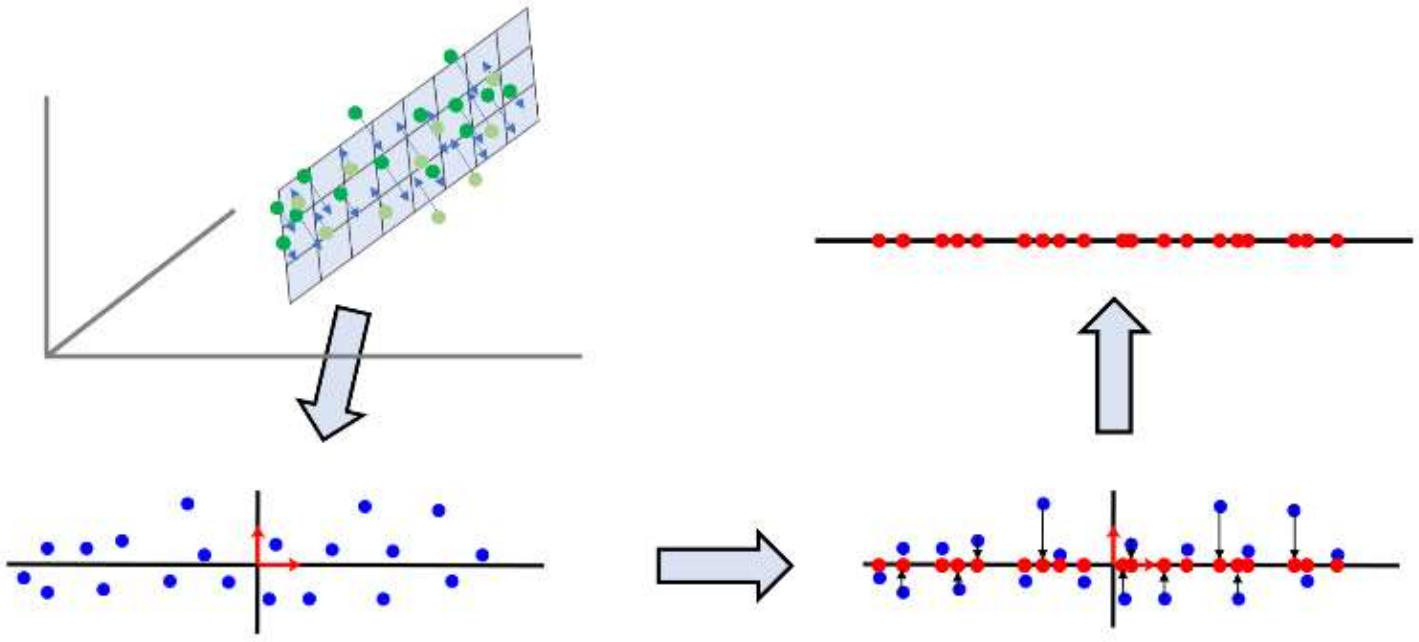
Possible feature engineering ideas:

- Additional statistics – PCA, NCA etc.
- Signal Processing Techniques – power spectral density, wavelets etc.
- Image Processing Techniques – bag of words, pixel intensity etc.
- Get creative!

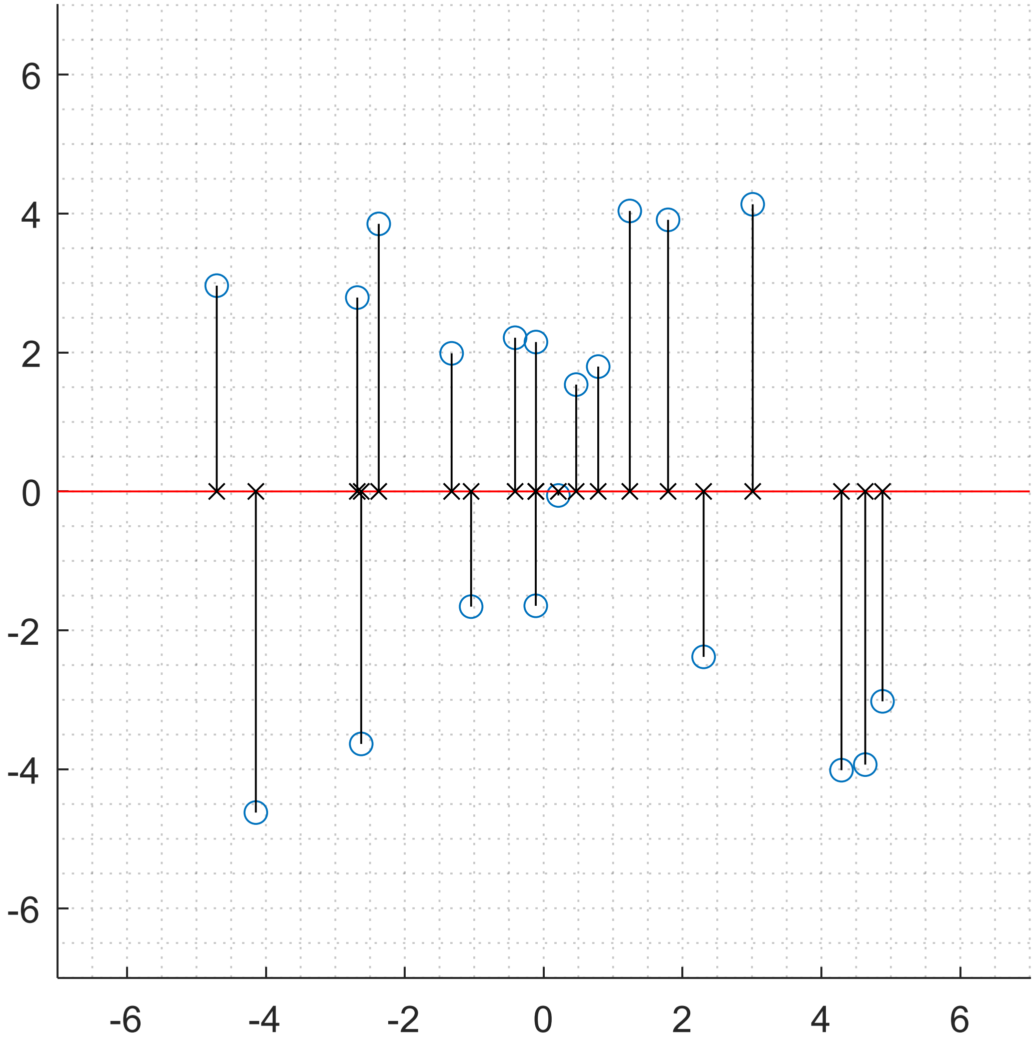


Principal Components Analysis(PCA)





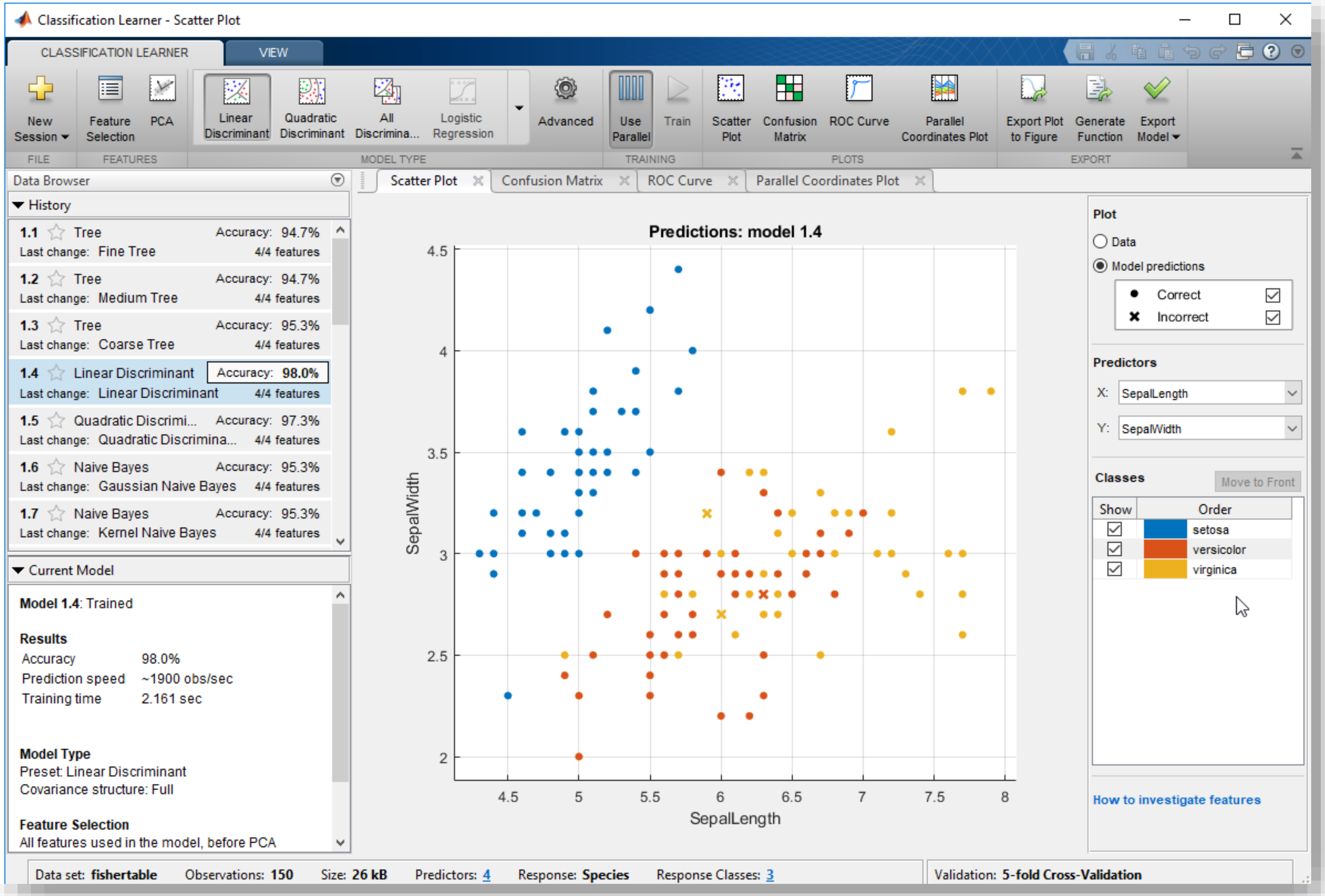
Demo Example 2: PCA



Comparison of Feature Selection Methods

Functions	Predictors	Machine Learning	Training Speed	Types of Models	Accuracy	Caveats
NCA	Continuous	Classification Regression	Medium	KNN SVM (can use for others)	Strong	Requires manual tuning of lambda
MRMR	Continuous Categorical Mix of both	Classification	Fast	Model Independent	Strong	
ReliefF	Continuous Categorical	Classification Regression	Medium	KNN SVM (can still use for others)	Moderate	Unable to differentiate correlated predictors
Sequentialfs	Continuous Categorical	Classification Regression	Very Slow	Model Independent (define custom loss function)	Strong	Doesn’t rank all features
F Test	Continuous Categorical Mix of both	Regression	Very Fast	Model Independent	Weak	Unable to differentiate correlated predictors
Chi Squared	Continuous Categorical Mix of both	Classification	Very Fast	Model Independent	Weak	Unable to differentiate correlated predictors

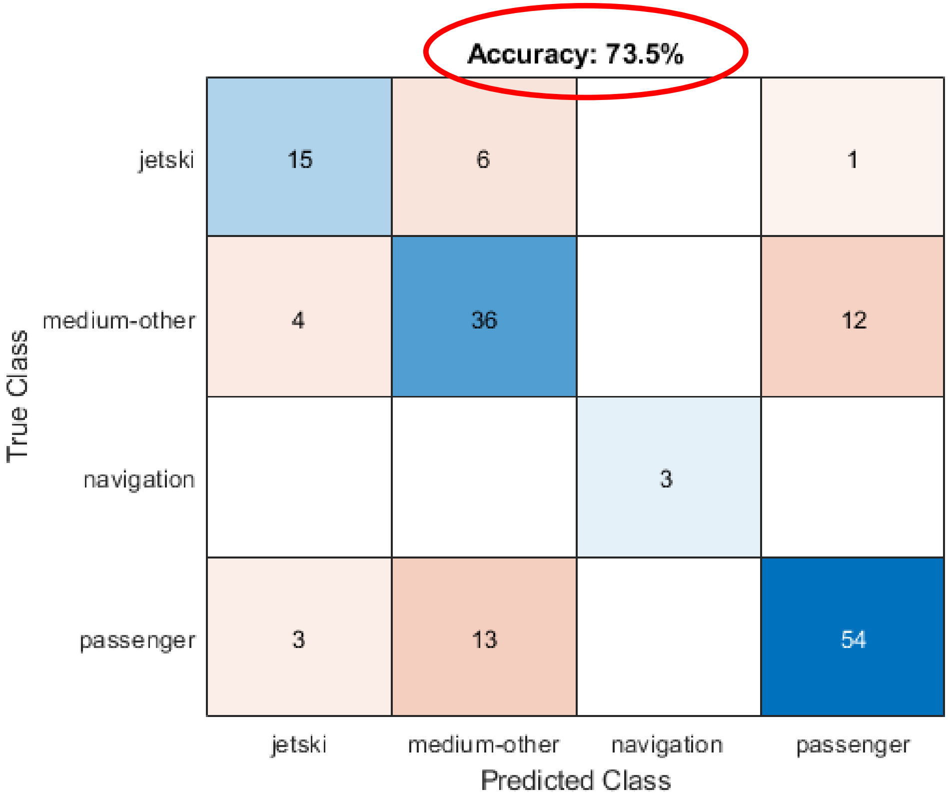
Demo Example 3: Classification Learner App



Data Augmentation to Improve Accuracy

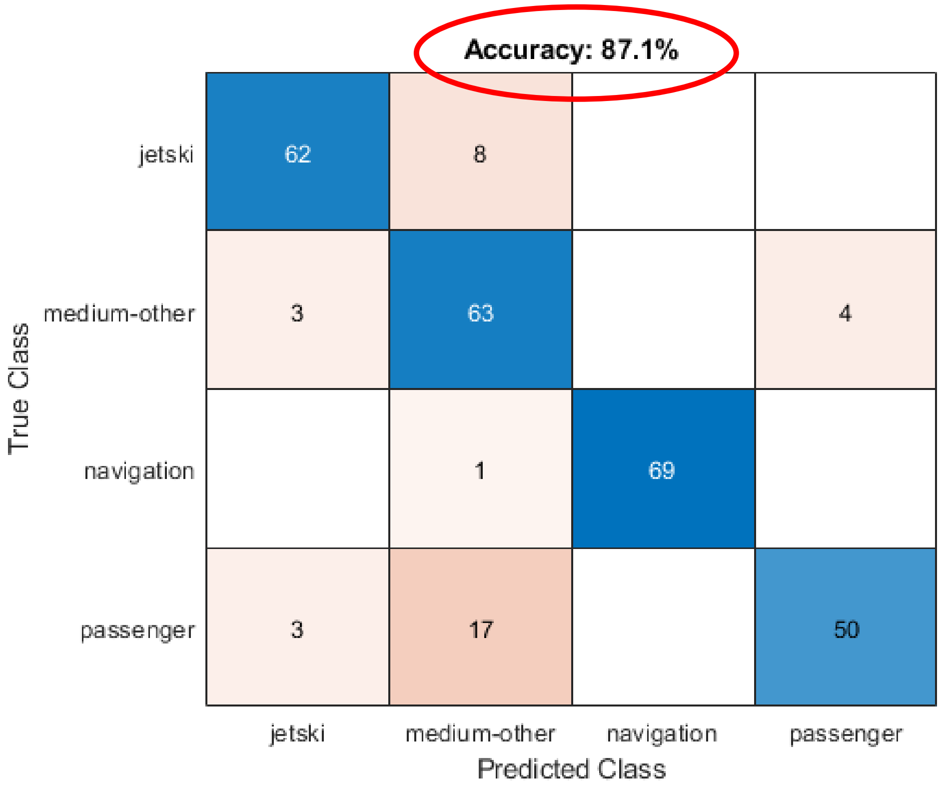
With original image set

	Label	Count
1	jetski	74
2	medium-other	172
3	navigation	11
4	passenger	233



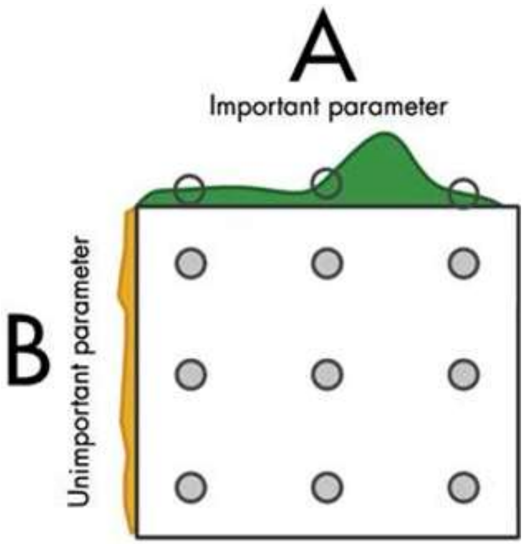
With augmented image set

	Label	Count
1	jetski	233
2	medium-other	233
3	navigation	233
4	passenger	233

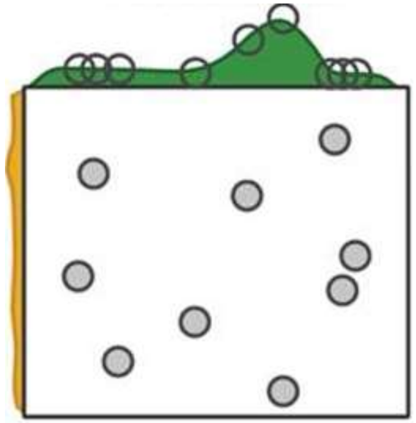


Hyperparameter Tuning

Standard:
Grid Search

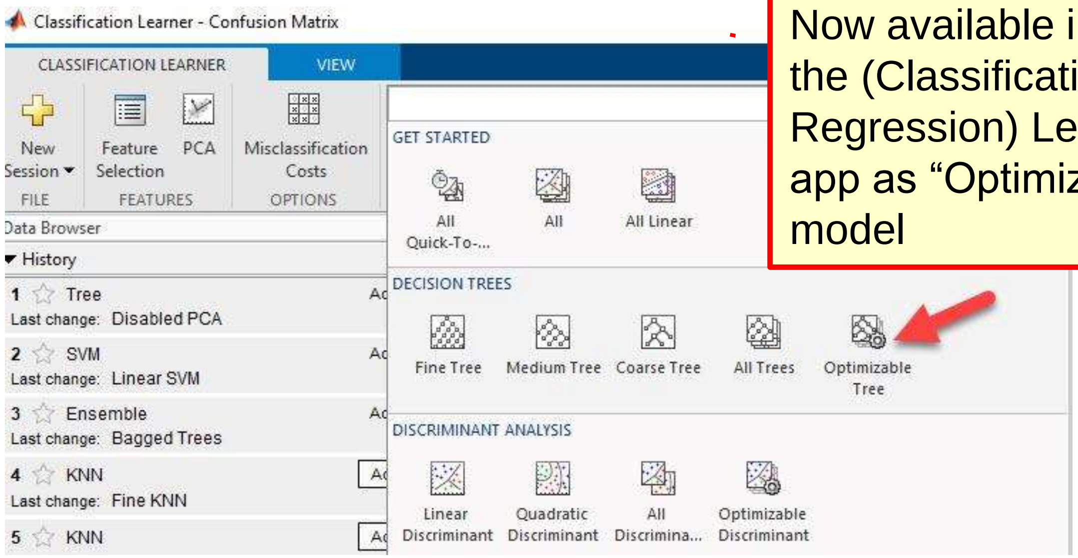


Better:
Random Search



Best: Bayesian Optimization

- Bayesian model indicates impact of change
- Model picks “good” point to try next
- Much more efficient!
- Scale to multi-cores (using PCT) for larger datasets



Now available inside the (Classification/ Regression) Learner app as “Optimizable” model

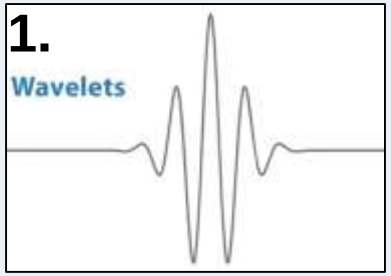
AutoML Machine Learning Workflow

Access Data → **AutoML** → **Integrate Analytics and Deploy**

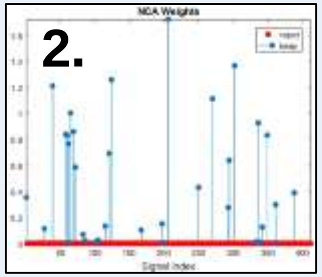
Preprocess Data

Develop Predictive Models

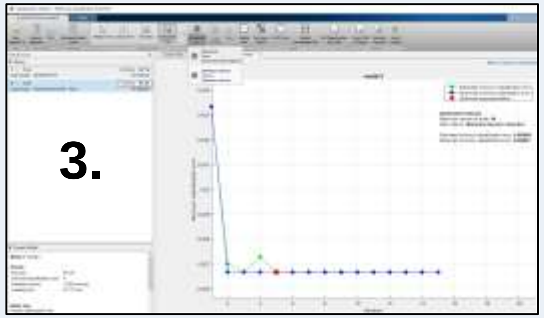
Data Preparation → **Feature Extraction** → **Model Training** → **Parameter Tuning**

1.


`Wavelets`

2.


`fscnca`

3.


`fitcauto`

4. `codegen`

AutoML Workflow in MATLAB

1. Generate features by applying Wavelet scattering

   Note: other (manual) feature generation methods exist!

2. Apply Feature Selection techniques

3. Select and Optimize Model

   3a. Train and optimize various models in Learner App **R2019b**

   3b. Or, automatic model selection `fitcauto` **R2020a**

4. Generate C-code or Compile to deploy: `codegen`

MathWorks

Model Selection and Optimization

R2019b

1. Open up Classification (or Regression) Learner App
2. Train multiple models
3. Perform hyperparameter tuning on top models
4. Other advanced optimization maneuvers are manual…

R2020a

1. Run fitcauto on your features
2. If good enough, DONE. ELSE continue with iterative process above.

Additional Resources:

Tech Talk: Hyperparameter Optimization [4:43 min video]
Bayesian Optimization Workflow [doc category page]
Hyperparameter Optimization in Classification Learner [doc]

Automated Feature Generation with Wavelet Scattering

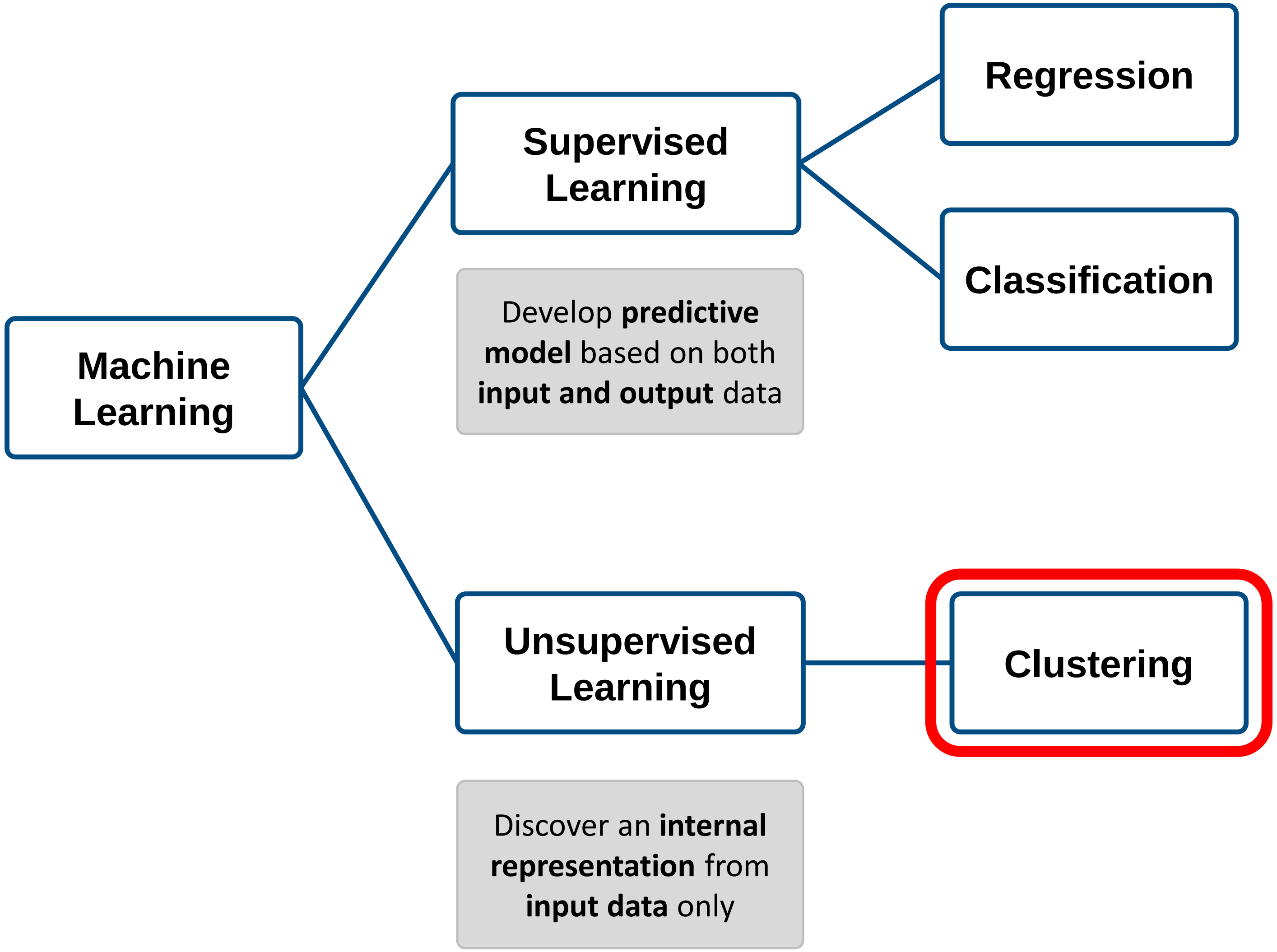
Wavelet Scattering Framework [Bruna and Mallat 2013]

- Automatic Feature Extraction
- Great starting point if you don't have a lot of data
- Reduces data dimensionality and provides compact features

Additional Resources:

- Wavelet scattering for ECG [doc example]
- Applying Deep Learning to Signals [3 min video]
- Blog about Wavelet scattering on towardsdatascience.com

Types of Machine Learning



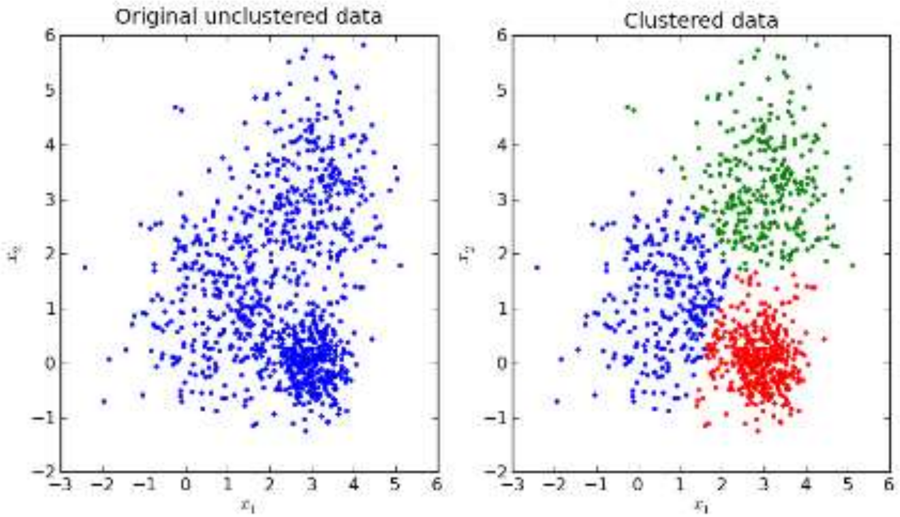
Clustering Motivation 1

- 2명의 대통령 후보가 있는 상황에서 여러분의 당선 전략은?
  - 상대와 나의 여론조사 지지율은 50.2% Vs 47.7% (2.5% p 차이)
  - 현재 결과의 역전을 위해서는 상대편 지지자의 1.3% p 가져와야 함.
- 그렇다면 상대편 지지자 중에서 나를 지지할 수 있는 1.3% p는 누구인가??
  - 유권자들을 세분화하여 군집화
  - 상대편으로부터 나에게 올 수 있는 유권자(군집)에게 집중
  - 즉, 군집을 통해 유권자별 맞춤형 선거전략이 가능
- 유권자들을 적절하게 군집하여 맞춤형 선거전략을 통해 선거 승리 가능함.

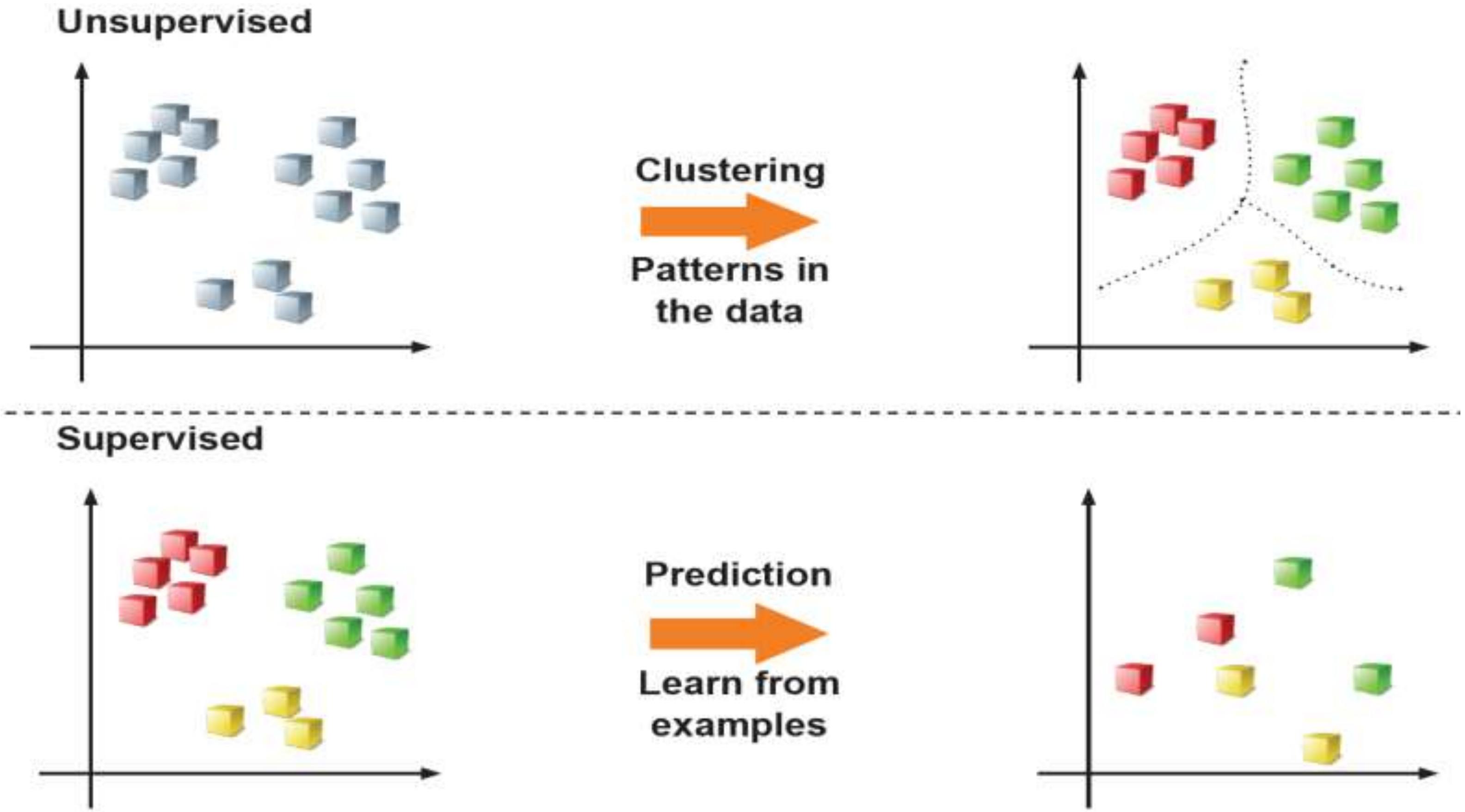
Clustering Motivation 2

- 왼쪽 데이터
  – 기존 데이터에 아무 것도 없는 상태(No Labeling)

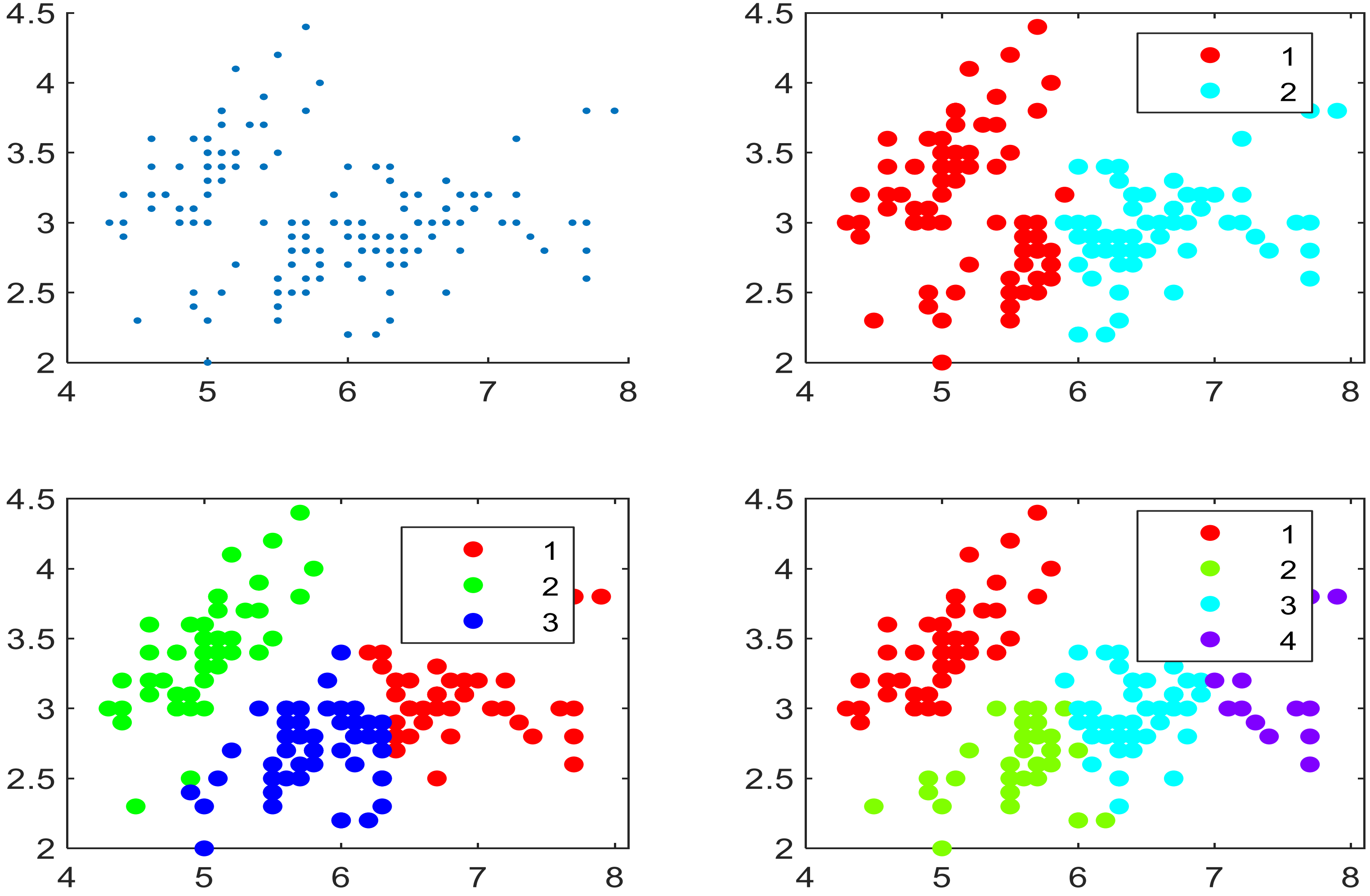
- 오른쪽 데이터
  – 데이터를 군집한 결과(Labeling 가능해짐)



Unsupervised Learning Vs Supervised Learning



Demo Example 1: K-means Clustering Algorithm using MATLAB



Clustering

- 데이터를 다음의 두가지 조건을 만족하는 군집함.
  - 같은 Cluster 내부의 데이터들 간의 유사성: **높음**
  - 같은 Cluster 내부의 데이터와 다른 cluster 내부의 데이터 간의 유사성 : **낮음**

- 유사성(similarity)
  - 두 사물(데이터)의 다른 점을 구분하는 기준이 되는 지표
  - 두 사물(데이터) 사이의 거리(distance, dissimilarity)로 측정
    - 거리를 구하는 방법은 여러가지가 있으나,
      유클리디안 거리 측정을 많이 활용함.
    - 기타 다른 거리 구하는 방식을 활용하여 군집할 수 있음.

$\mathbf{x} = (x_1, x_2, \ldots),\ \mathbf{y} = (y_1, y_2, \ldots)$

① Euclidian distance (dissimilarity)

$$D(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

② Manhattan distance (dissimilarity)

$$D(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y}) = \sum_i |x_i - y_i|$$

③ "sup" distance (dissimilarity)

$$D(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, \mathbf{y}) = \max_i |x_i - y_i|$$

④ Correlation coefficient (similarity)

$$D(\mathbf{x}, \mathbf{y}) = s(\mathbf{x}, \mathbf{y}) = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

⑤ Cosine similarity (similarity)

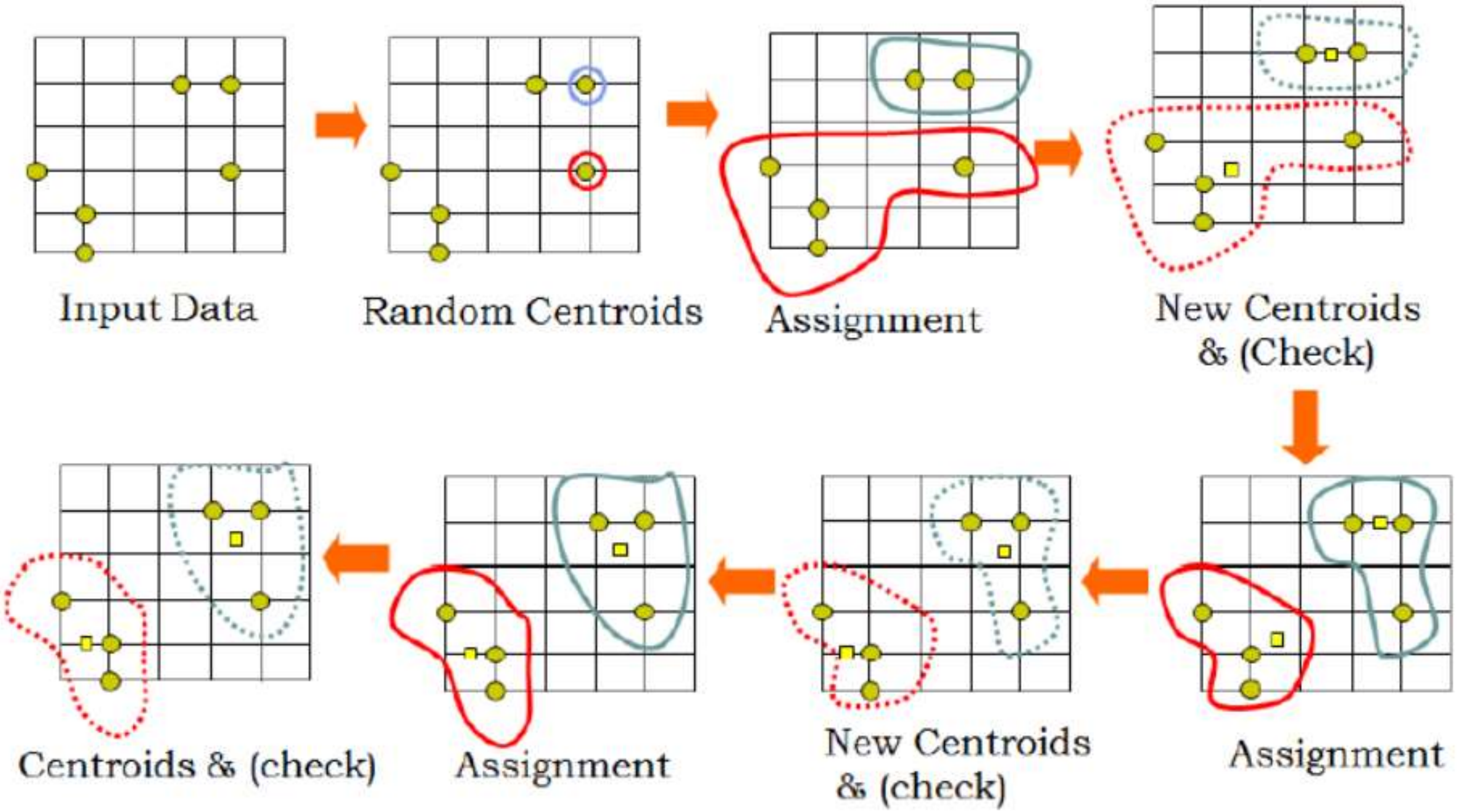
$$D(\mathbf{x}, \mathbf{y}) = \cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|} = \frac{\sum_i x_i y_i}{\sqrt{\sum_i x_i^2}\sqrt{\sum_i y_i^2}}$$

K-means Clustering Algorithm

- 개념
  - 입력 값으로 군집할 개수(K)가 필요
  - 결정된 군집 수에 따른 군집 별 중심을 설정
  - 군집 별 중심과 데이터 사이의 거리를 구함
  - 거리가 작은 중심 군집으로 해당 데이터를 속하게 함

- 특징
  - **거리기반 군집**
  - **반복작업**: 반복수행을 통해 군집 결과를 개선함.
  - **짧은 계산시간**: 직관적으로 이해되는 간단한 알고리즘으로 계산이 빠름
  - **탐색적 기법**: 주어진 자료에 대한 사전정보가 필요 없음(거리를 계산해 나가며 군집)

K-means Clustering Algorithm Process



Application Case of K-means Clustering

- 데이터 마이닝에서 데이터 군집 알고리즘으로 활용
- 트랜드 또는 성향이 불분명한 시장의 데이터 분석에 활용

- 실제 활용 사례1
  - 네이버나 카카오 뉴스 검색 클러스터링: **k-means clustering 방식 사용 추정**

- 실제 활용 사례2
  - 스타트업 기업: 언니의 파우치(**https://www.unpa.me/**)
    - 국내 화장품에 대한 소비자들의 리뷰 정보를 보유하고 있는 업체
    - 리뷰에 엄격한 기준이 적용되어 퀄리티가 높은 리뷰가 많음.
    - 고객들의 정보가 연령, 피부타입, 고객의 커뮤니티 활동, 구매 정보 등으로 나뉘어 축적되어 있음.

Application Case of K-means Clustering

– 이용자의 11개 변수를 활용해 **K-means Clustering 을 통해 5개 cluster로 군집**
  - ① 화장하기 시작한 **중고등학생**
  - ② 언니의 파우치 주축 활동 멤버로 **고등학생과 20대 초반**
  - ③ VVIP, 언니의 파우치 활동 대장 **20대**
  - ④ 언니들을 보며 배우는 **고등학생과 20대 초반**
  - ⑤ 이벤트만 관심, 조용한 **30대 이상 진짜 언니들**

[고객세분화]

Application Case of K-means Clustering

- 5개 그룹별로 '언니의 파우치' 스토어에서 구매에 영향을 주는 요인 분석
  - Tier 1인 중고등학생이 다른 그룹보다 앱 내 활동이 구매에 미치는 영향이 컸음.
    - 이들의 구매 유도를 위해 팔로워 및 팔로잉 기능 강화
    - 인적 네트워크 활성화 등 앱 내 활동 활성화 방안이 필요
  - Tier 5인 30대는 구매력은 높으나, 활동이 없는 조용한 고객군으로 분석
    - 분석 결과 이벤트에 민감한 것으로 파악
    - 30대를 위한 타깃 이벤트 기획 필요 – 안티에이징 화장품 리뷰 이벤트를 진행하고 좋은 반응을 얻음.

- <u>언니의 파우치 주요 고객층이 **10대 후반에서 20대 초반임을** 파악</u>
  - 그룹별 고객 분석 전: 주고객이 20대 후반이라는 막연한 추정
  - 이들을 타깃으로 하는 신제품 출시

Application Case of K-means Clustering

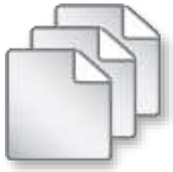
['언니의 파우치' 스토어 8~11월 매출액]

Machine Learning Workflow

Access and Explore Data → **Preprocess Data** → **Develop Predictive Models** → **Integrate Analytics with Systems**

Access and Explore Data

Files

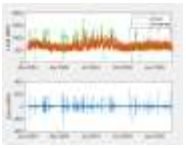


Databases

Sensors

Preprocess Data

Working with Messy Data

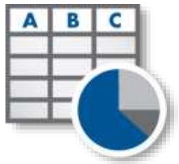


Data Reduction/ Transformation

Feature Extraction

Develop Predictive Models

Model Creation e.g. Machine Learning



Parameter Optimization

Model Validation

Integrate Analytics with Systems

Desktop Apps

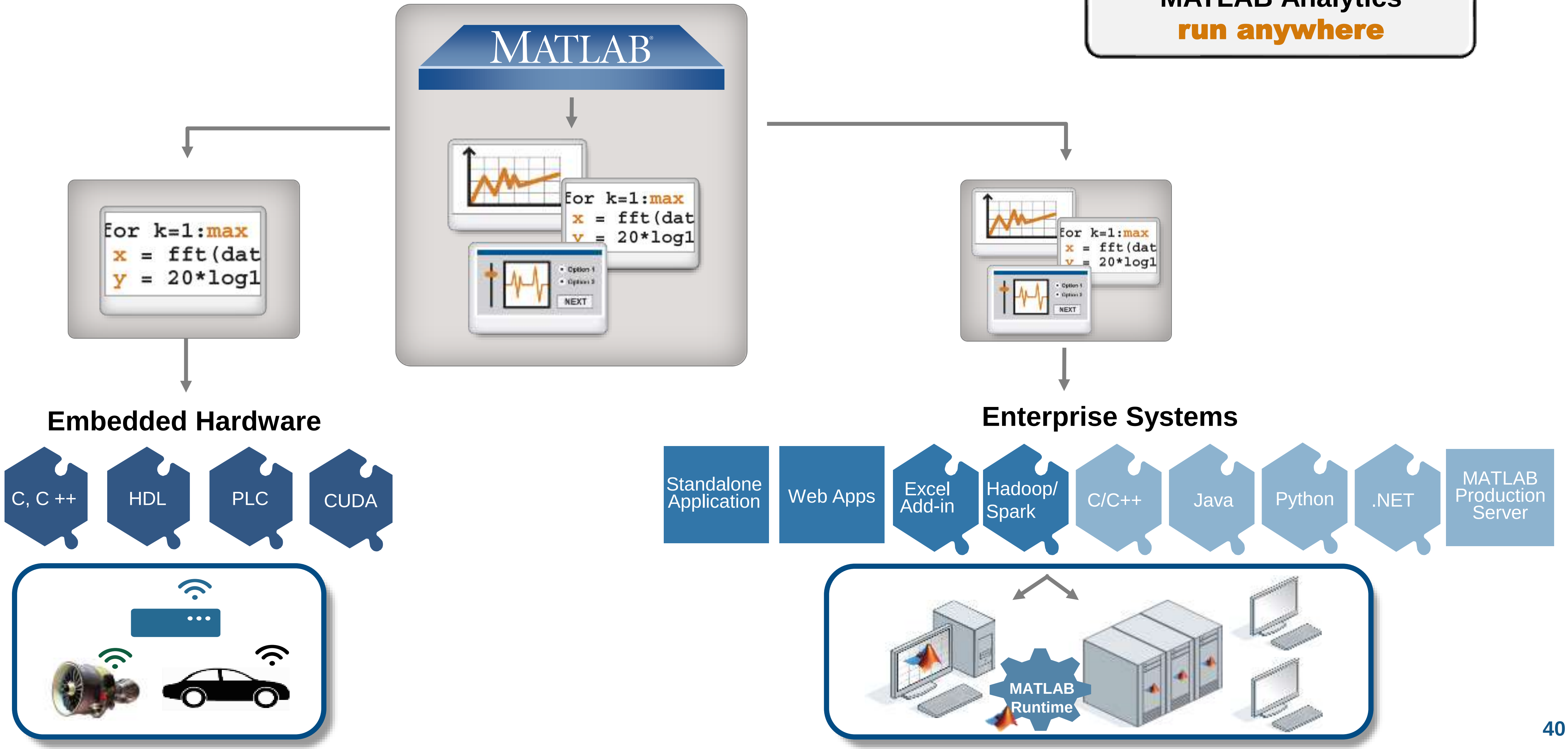
Enterprise Scale Systems

MATLAB Excel
.NET
.exe C/C++
Java .dll

Embedded Devices and Hardware

Integrate Analytics with Systems

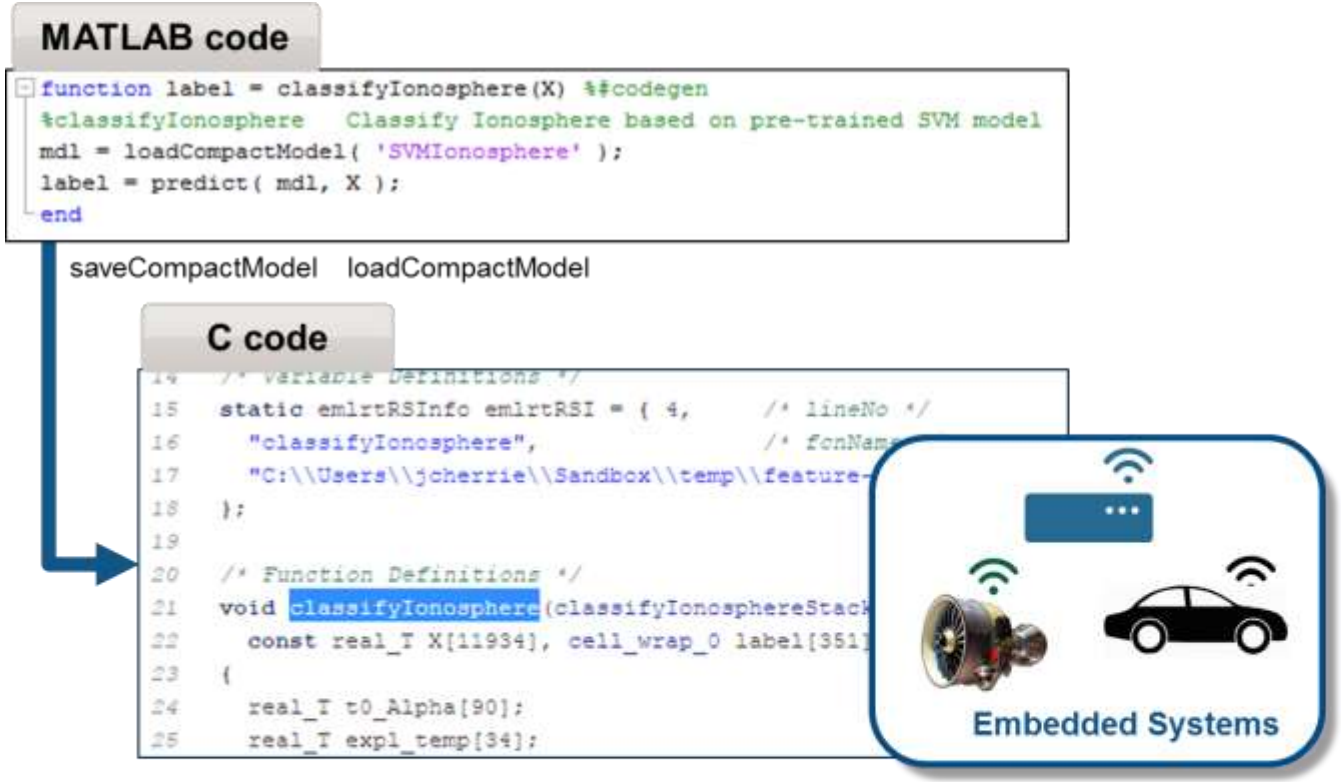
MATLAB Analytics
run anywhere



Machine Learning for Edge Analytics and Code Deployment

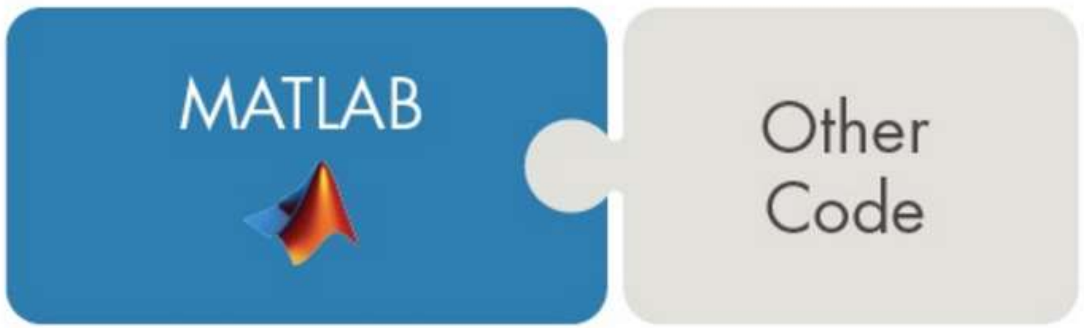
Deploy trained models as standalone C/C++ code

- Apply algorithms to out-of-memory data using tall arrays
- Generate C/C++ code for predictive models
- Generate fixed-point C/C++ code for SVM models, decision trees, and ensembles of decision trees
- Update deployed models without regenerating code



Using MATLAB with Other Languages

Calling Libraries Written in Another Language From MATLAB



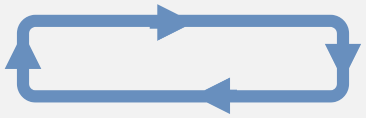
- Java
- Python
- C
- C++
- Fortran
- COM components and ActiveX® controls
- RESTful, HTTP, and WSDL web services

Call C++ libraries directly from MATLAB R2019a

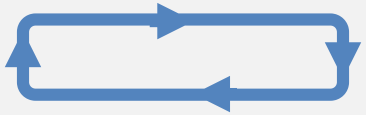
Calling MATLAB from Another Language

- Java
- Python
- C/C++
- Fortran
- COM Automation server

Challenges in Machine Learning

Steps	Challenge
Access and Explore data	**Data diversity** Numeric, Images, Signals, Text – not always tabular
Preprocess Data	**Lack of domain tools** Filtering and feature extraction Feature selection and transformation
Develop Predictive Models	**Time consuming** Train and compare several models to find the “best” Select optimal parameters and avoid overfitting
Integrate Analytics with Systems	**Platform diversity** Translate analytics to production Deploy on different target platforms
Iterate	

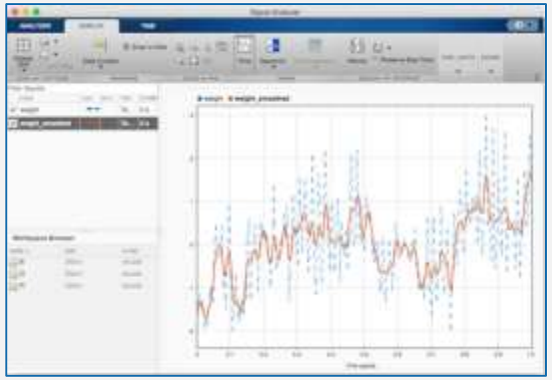
MATLAB Strengths for Machine Learning

Challenge	Solution
Data diversity	**Extensive data support** Work with signal, images, financial, textual, and others formats
Lack of domain tools	**High-quality libraries** Industry-standard algorithms for Finance, Statistics, Signal, Image processing & more
Time consuming	**Interactive, app-driven workflows** Focus on machine learning, not programing Select best model and easily fine-tune model parameters
Platform diversity	**Run analytics anywhere** Code generation for embedded targets Deploy to broad range of enterprise system architectures
	**Flexible architecture for customized workflows** Complete machine learning platform

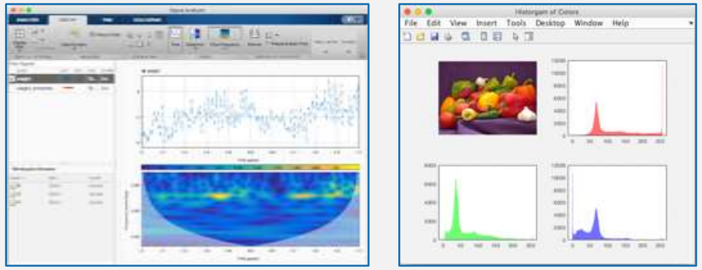
Summary: Complete Machine Learning Workflow

Access and explore data → **Preprocessing** → **Feature Engineering** → **Model Training** → **Model Tuning** → **Integrate Analytics**

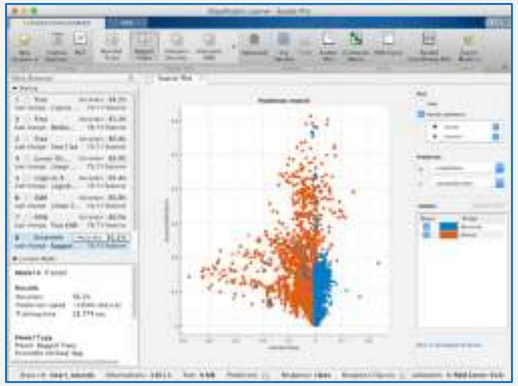
Text files, spreadsheets, databases, binary files, data feeds, web, cloud storage



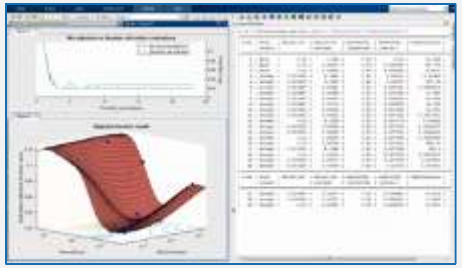
Datatypes and tools for missing data, outliers, time-alignment, etc.



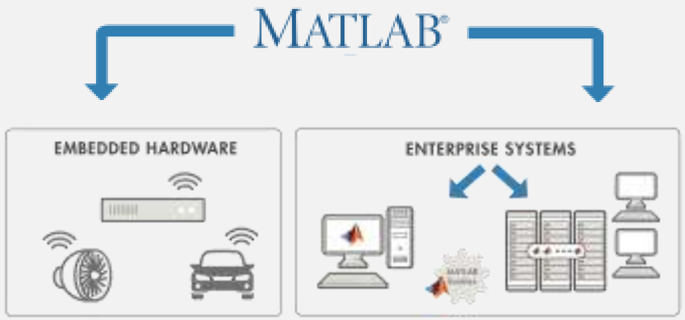
Domain-specific techniques for Signals, Images, Video, Audio, and Text



Machine Learning apps



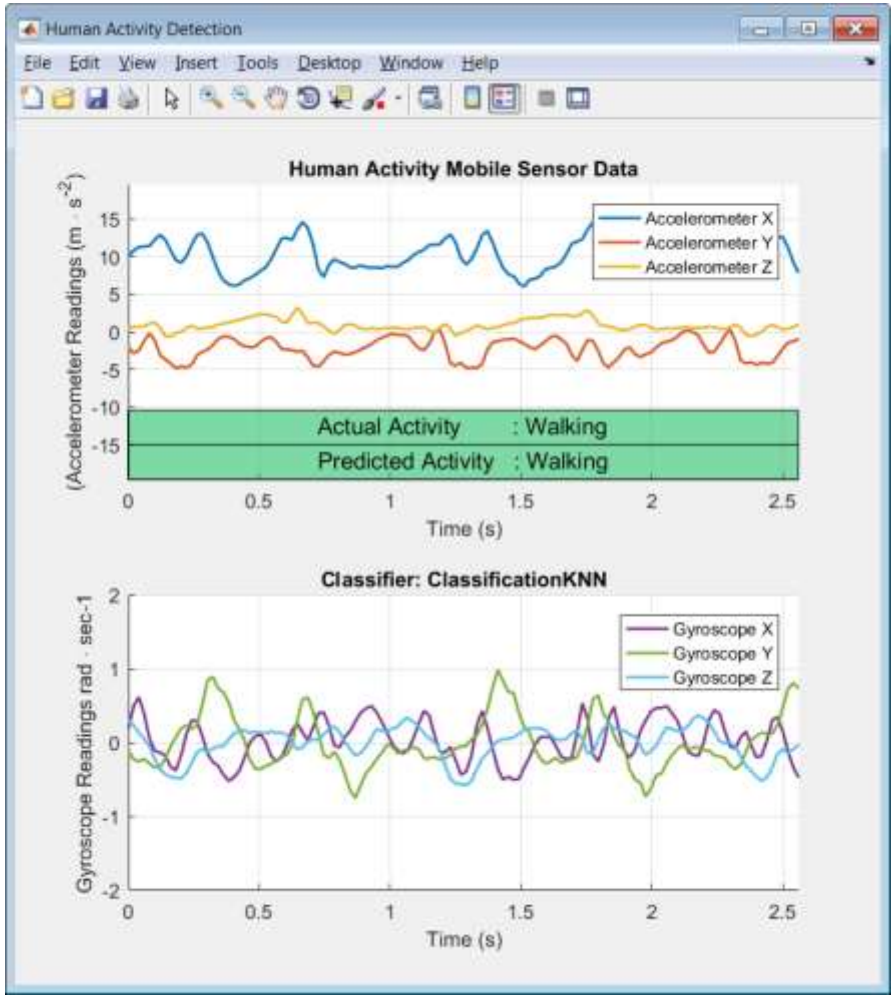
Automated Parameter Tuning



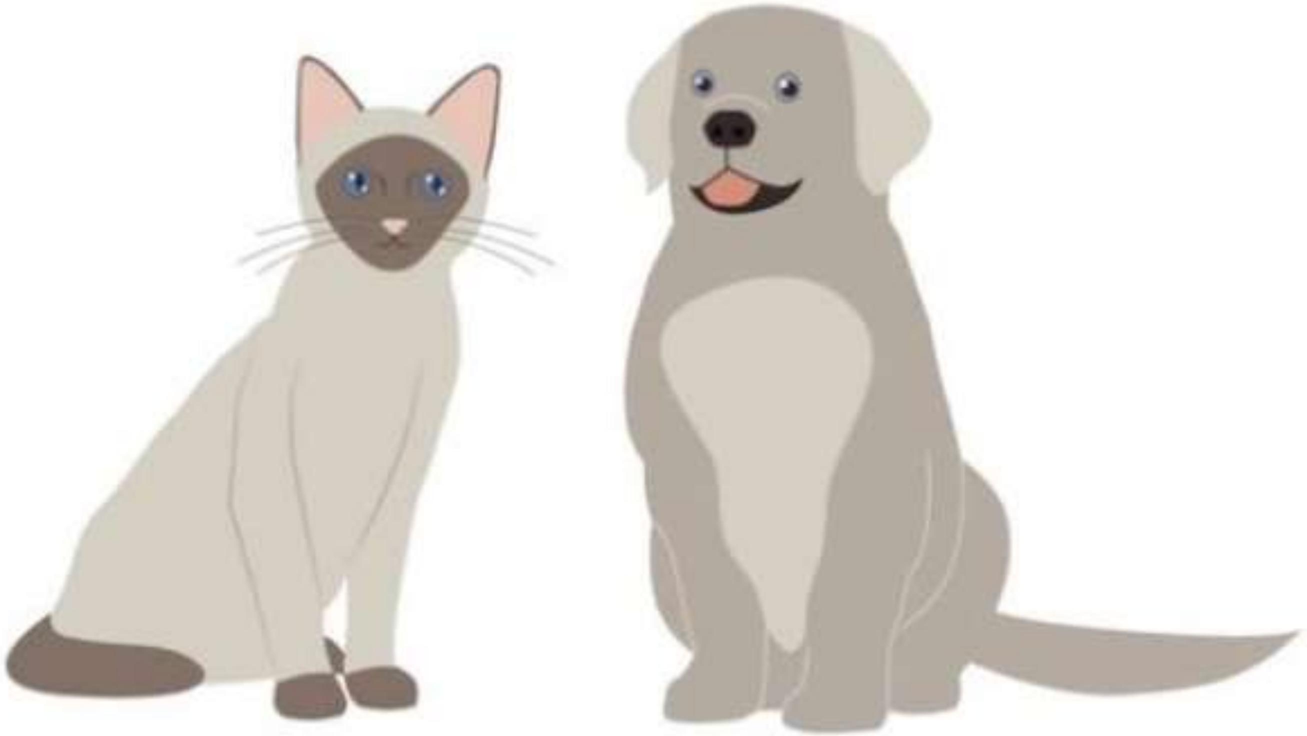
C/C++ Code Generation and Enterprise IT Integration

Agenda 2

- What is Deep Learning?

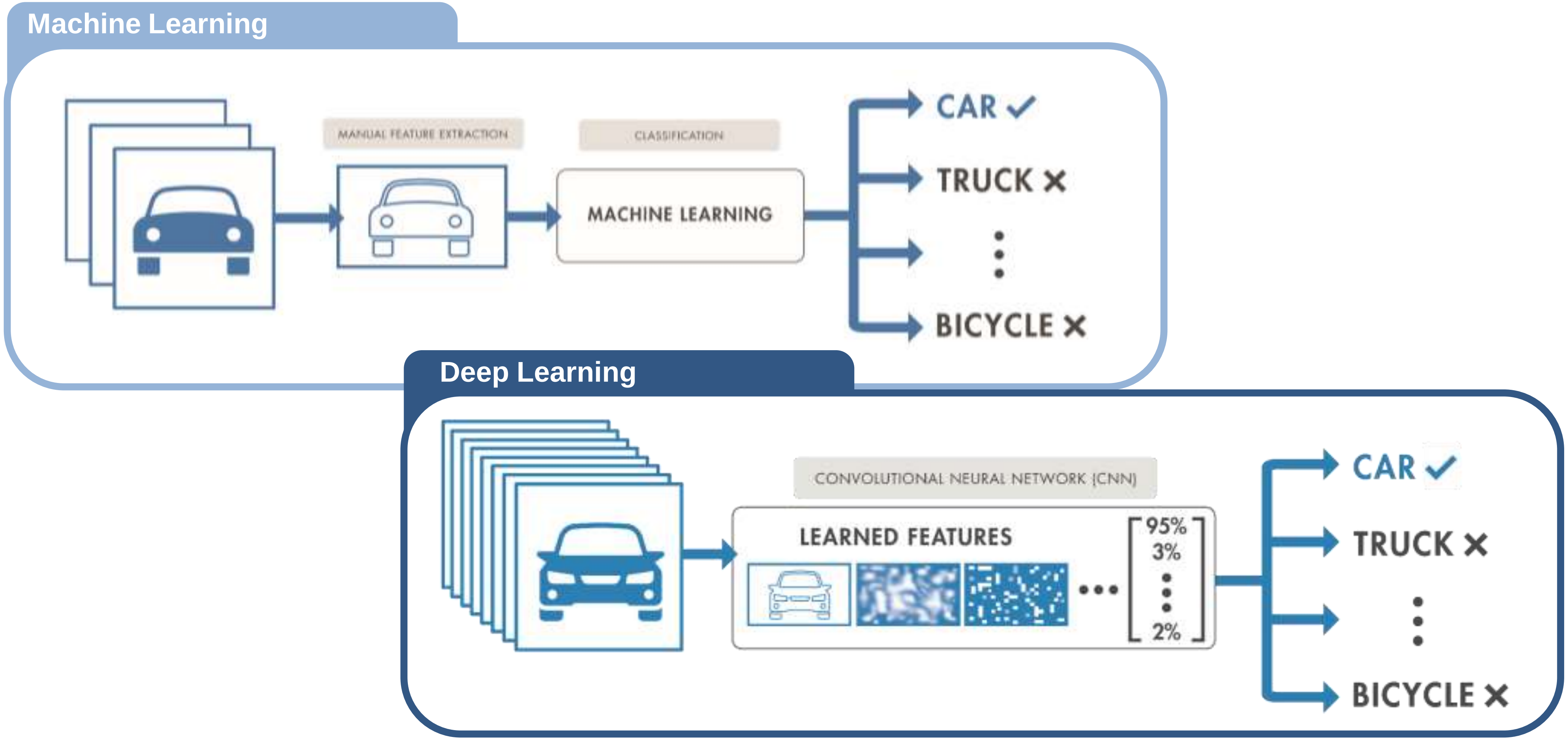


Machine Learning vs Deep Learning

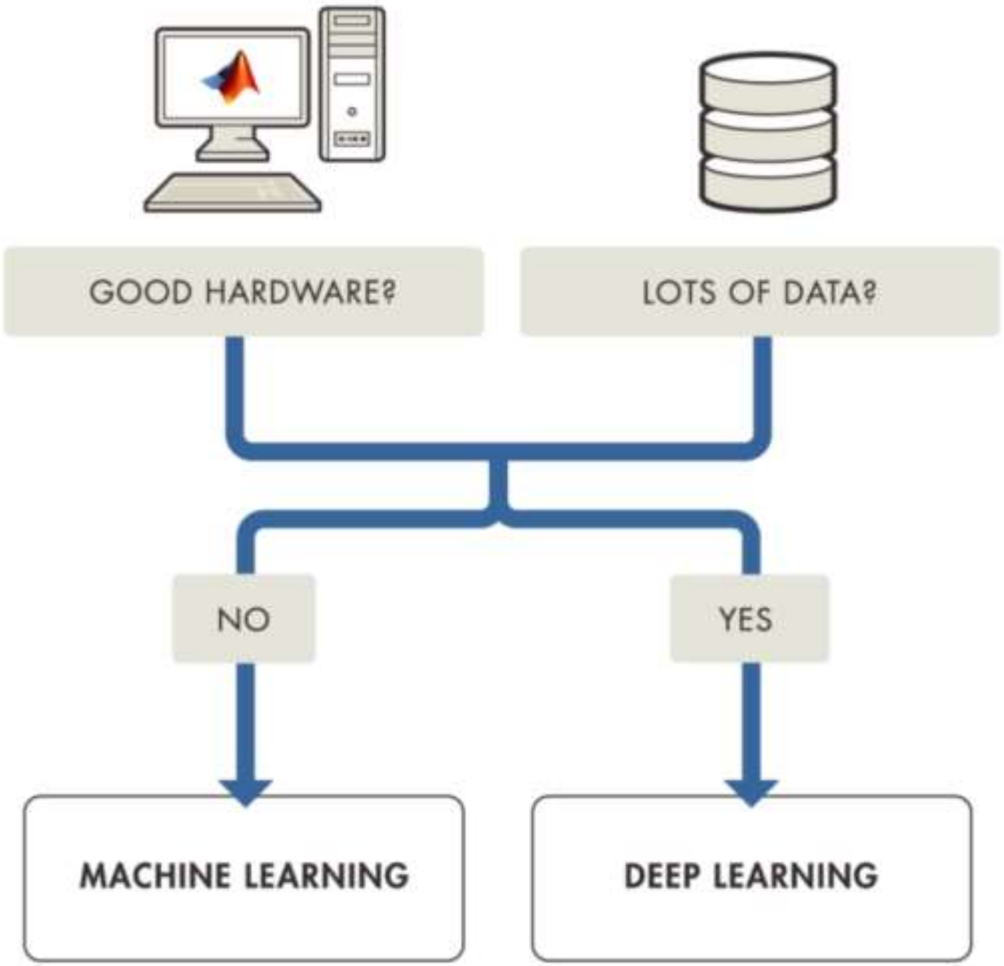


Machine Learning vs Deep Learning

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



Machine Learning vs Deep Learning

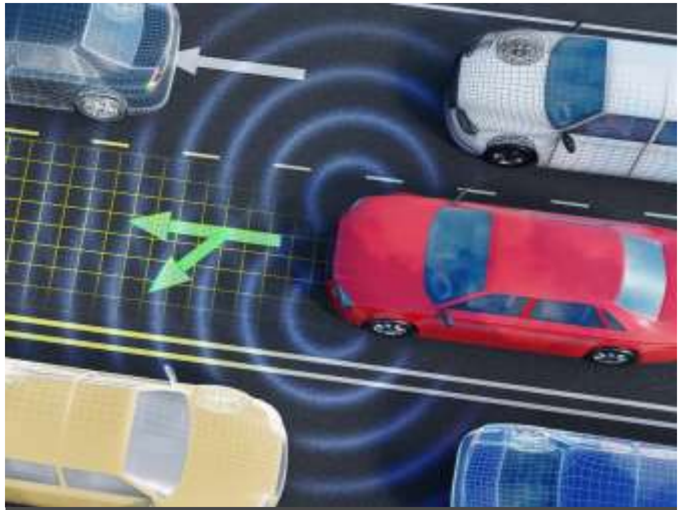


	Machine Learning	Deep Learning
Training dataset	Small	Large
Choose your own features	Yes	No
# of classifiers available	Many	Few
Training time	Short	Long

Deep learning is part of our everyday lives

Speech Recognition

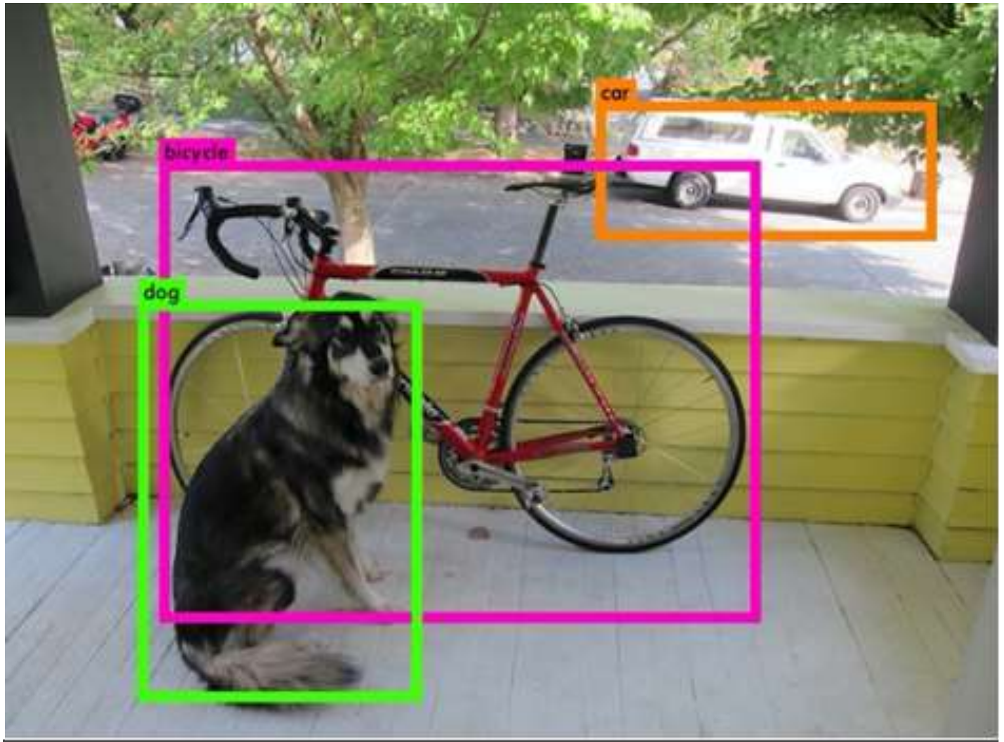
Face Detection



Automated Driving

Deep learning applications: mainstream vs. engineering

Mainstream



Detecting Objects

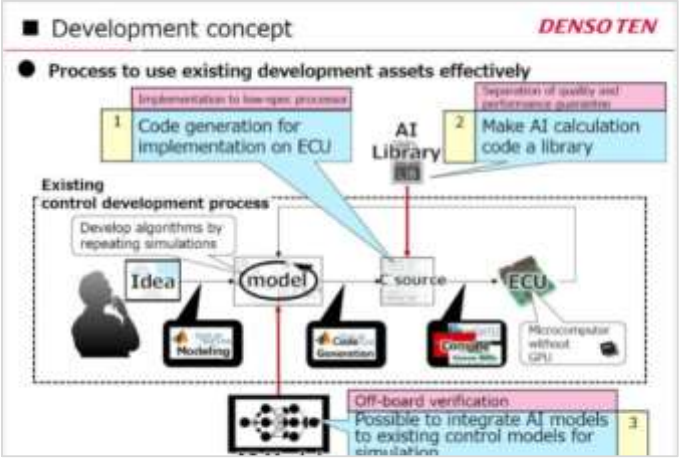
Engineering and Science

Identifying Machinery at Shell

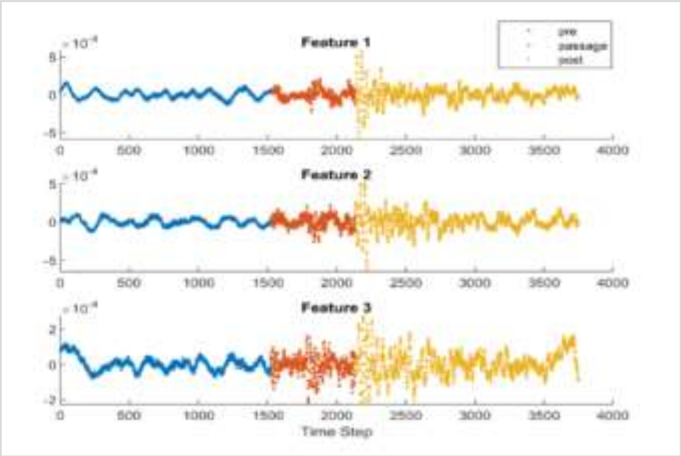
Deep Learning Detection

MATLAB Deep Learning used in Industry

Automatic Defect Detection
Airbus



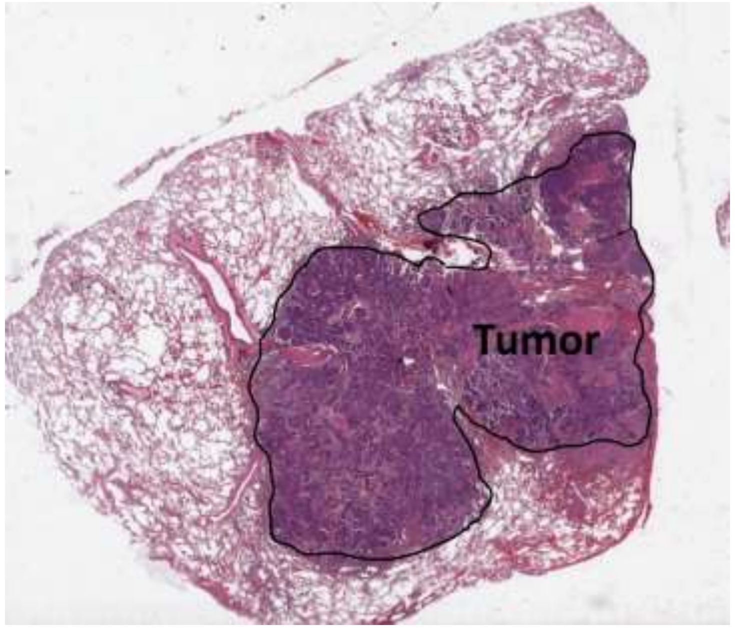
ECU Vehicle Control
Denso



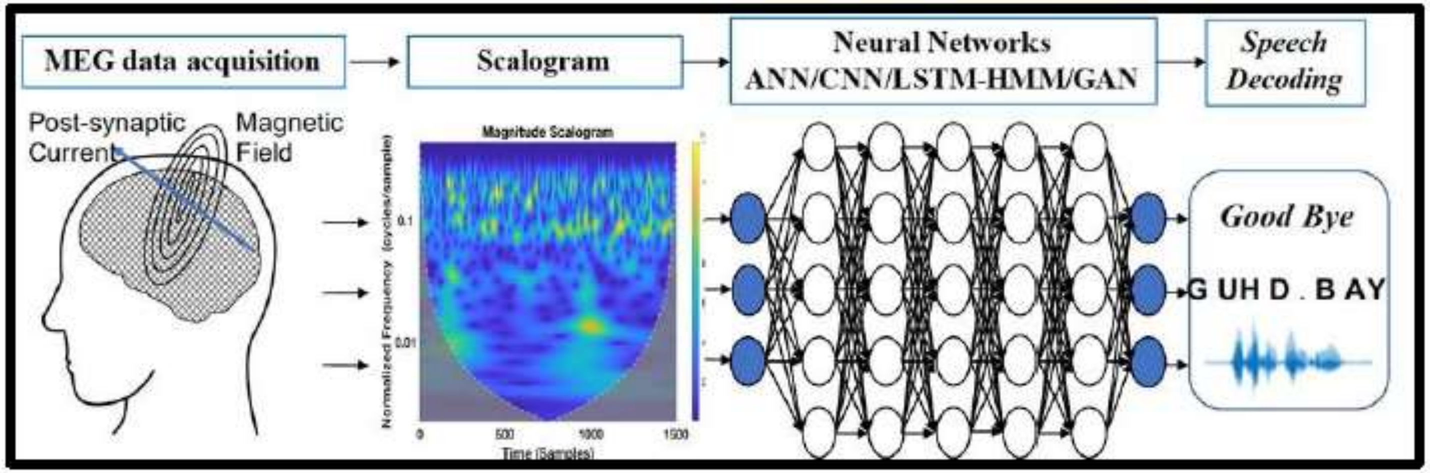
Seismic Event Detection
Shell

52

MATLAB Deep Learning used in Research

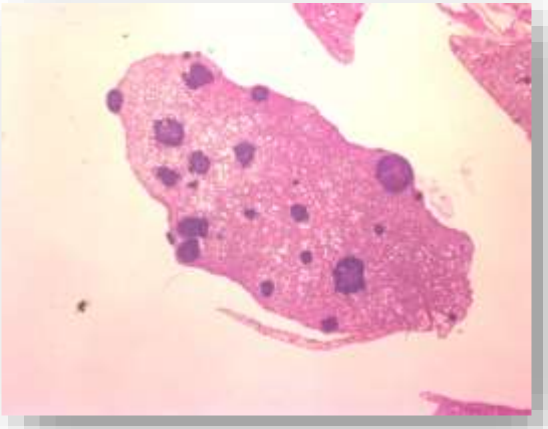


Predicting gastrointestinal cancer (July 2019)



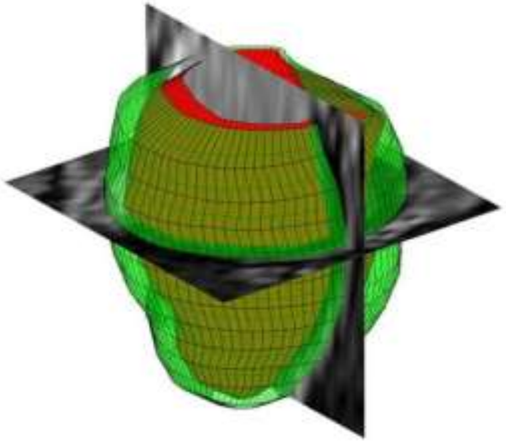
Converting brain waves to speech to help ALS patients communicate (Nov 2019)

MATLAB is used in many areas of medical imaging

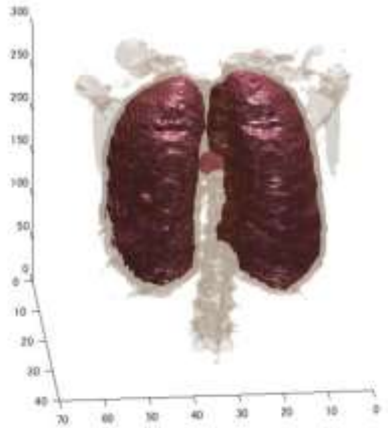


Digital pathology

Ophthalmology/OCT

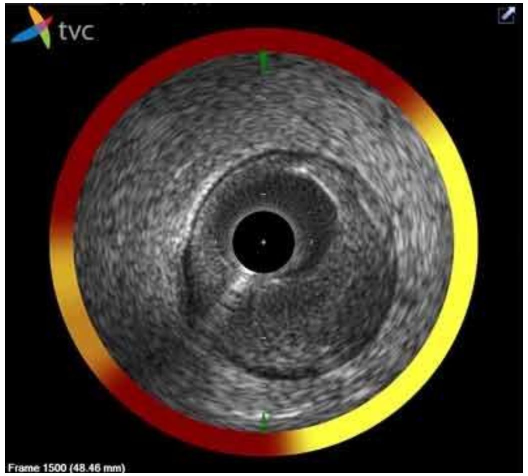


Radiology (MRI, US, X-ray, CT)



Radiotherapy planning

Endoscopy



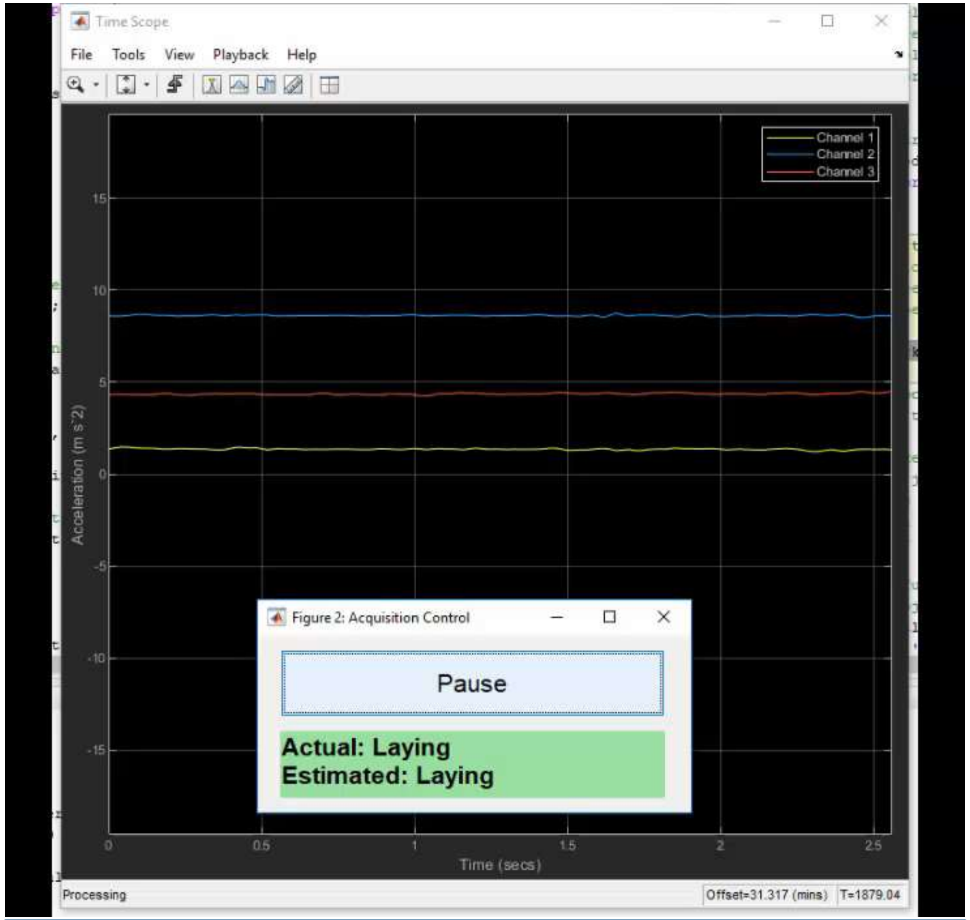
Intravascular imaging

Applications of deep learning for images and video

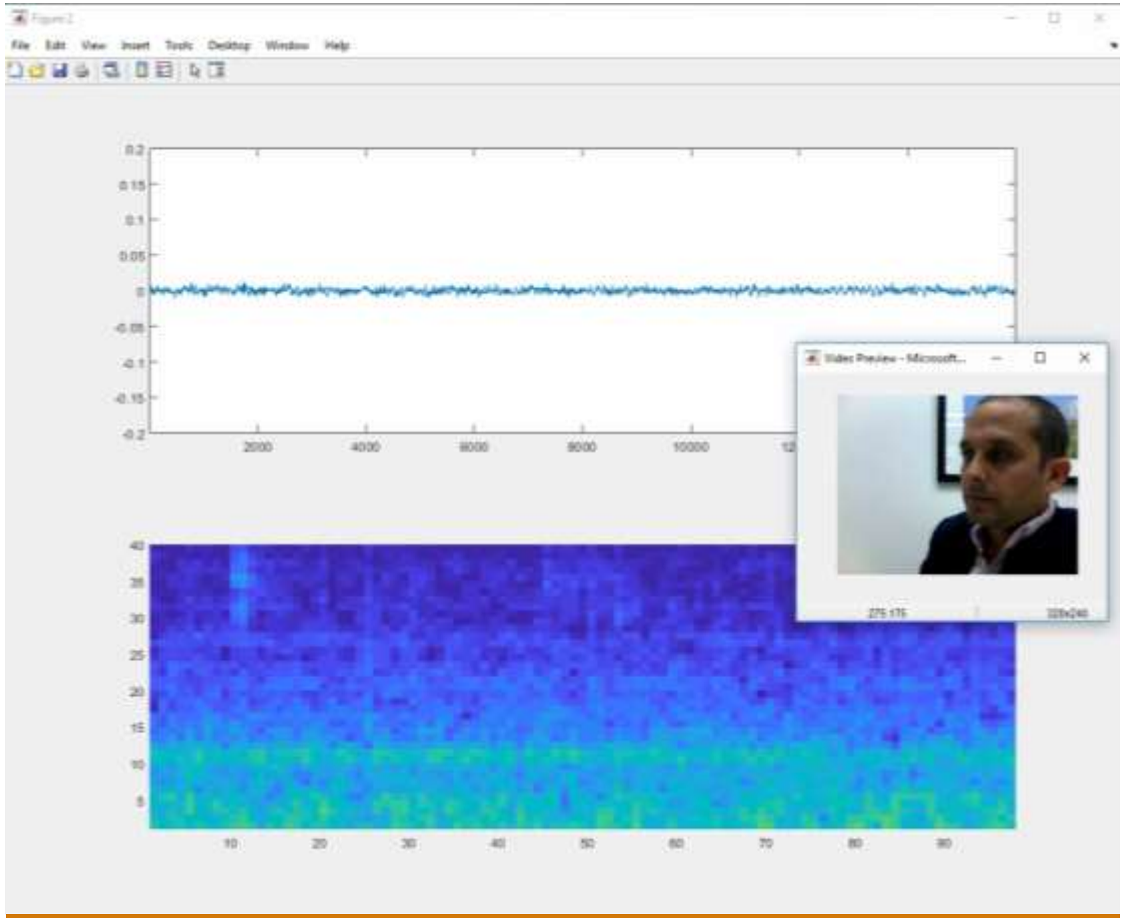
YOLO v2 (You Only Look Once)

Semantic Segmentation using SegNet

Applications of deep learning for signal processing



Signal Classification using LSTMs



Speech Recognition using CNNs

AI-driven system design

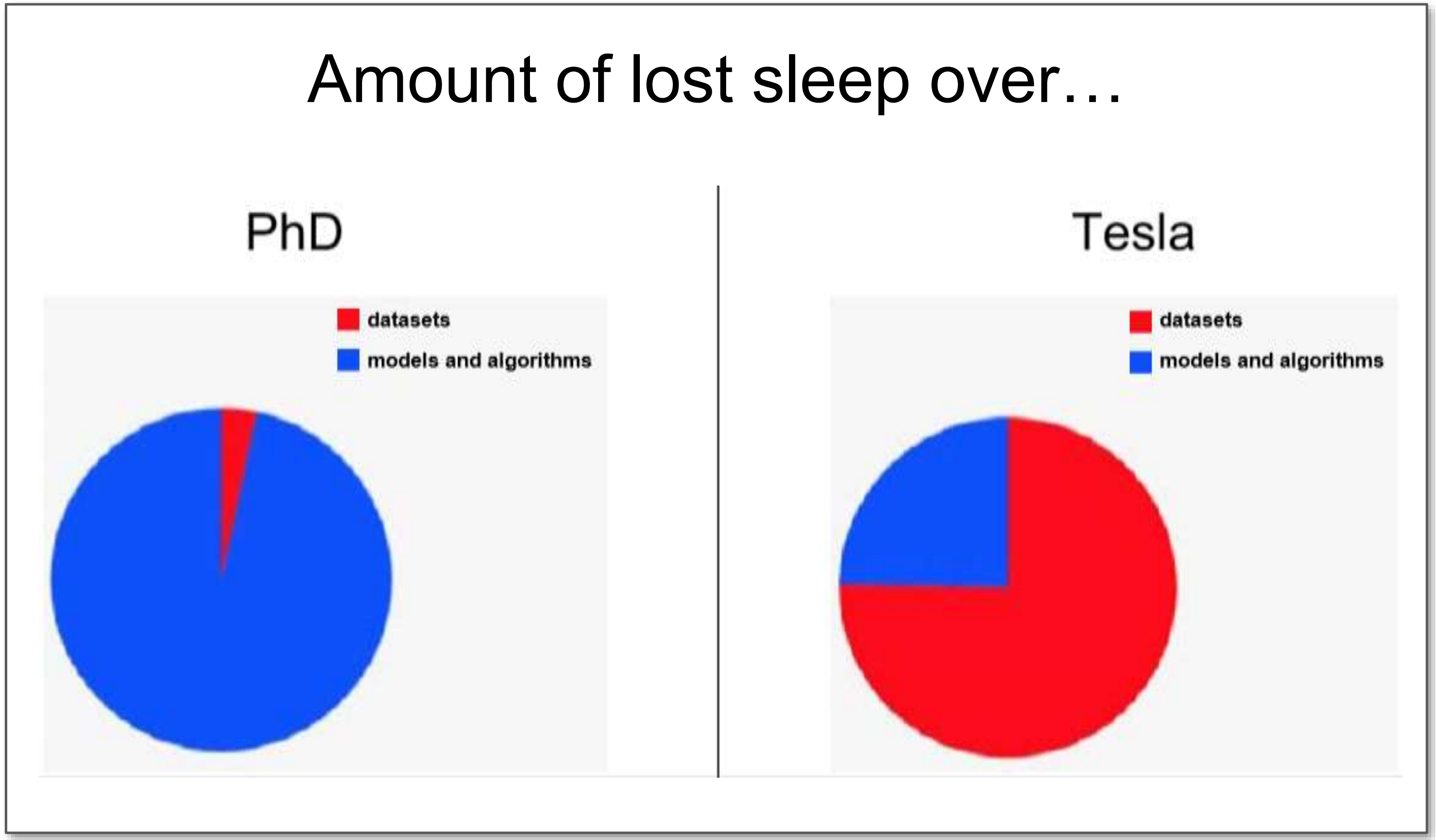
Data Preparation	AI Modeling	Simulation & Test	Deployment
Data cleansing and preparation	Model design and tuning	Integration with complex systems	Embedded devices
Human insight	Hardware accelerated training	System simulation	Enterprise systems
Simulation-generated data	Interoperability	System verification and validation	Edge, cloud, desktop

Data preparation represents most of your AI effort…

Transforming raw data for useful modeling and analysis is a critical step.

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data



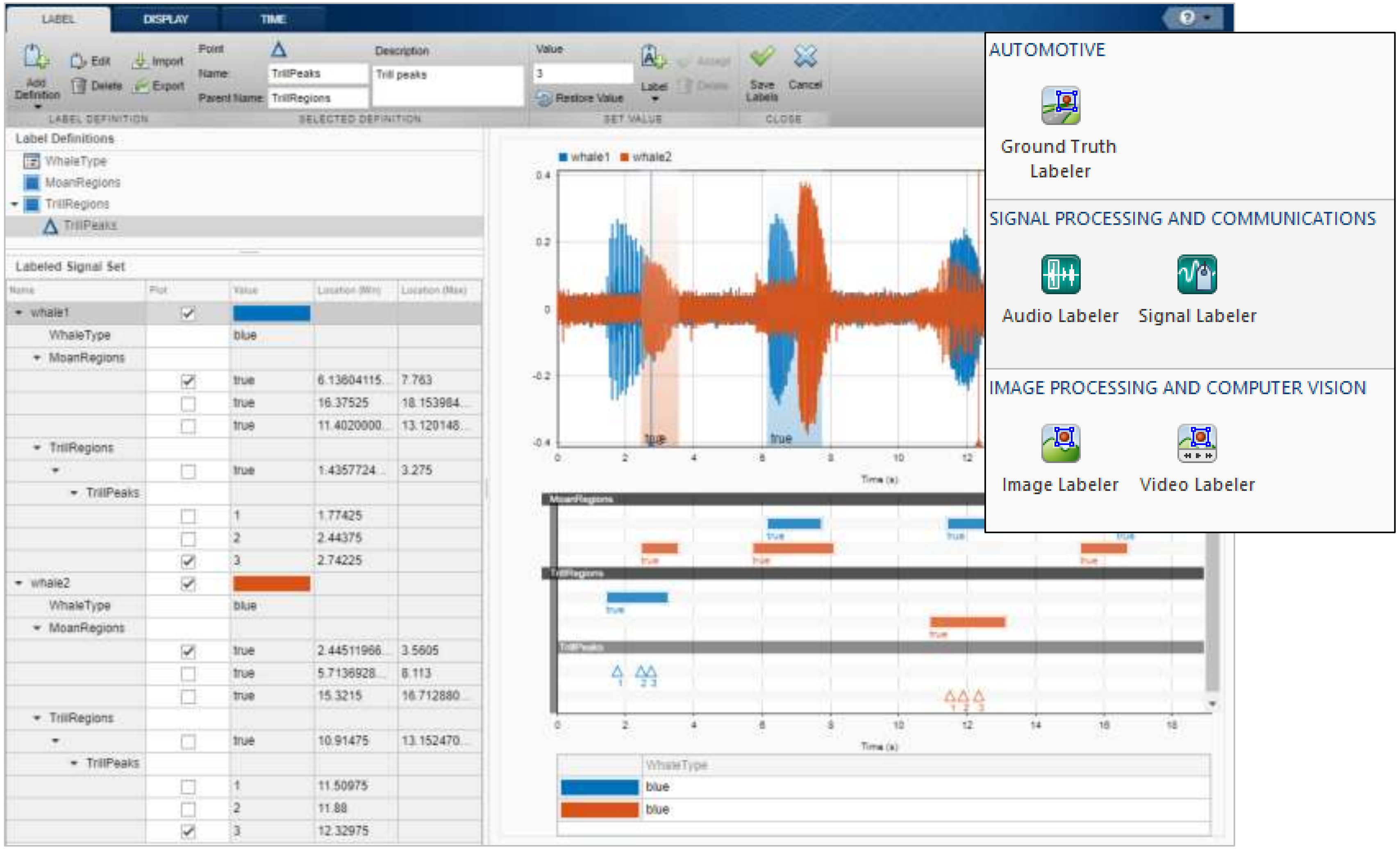
Source: Andrej Karpathy slide from TrainAI 2018

Spend less time preprocessing and labeling data

Synchronize disparate time series, filter noisy signals, automate labeling of video, and more.

Data Preparation

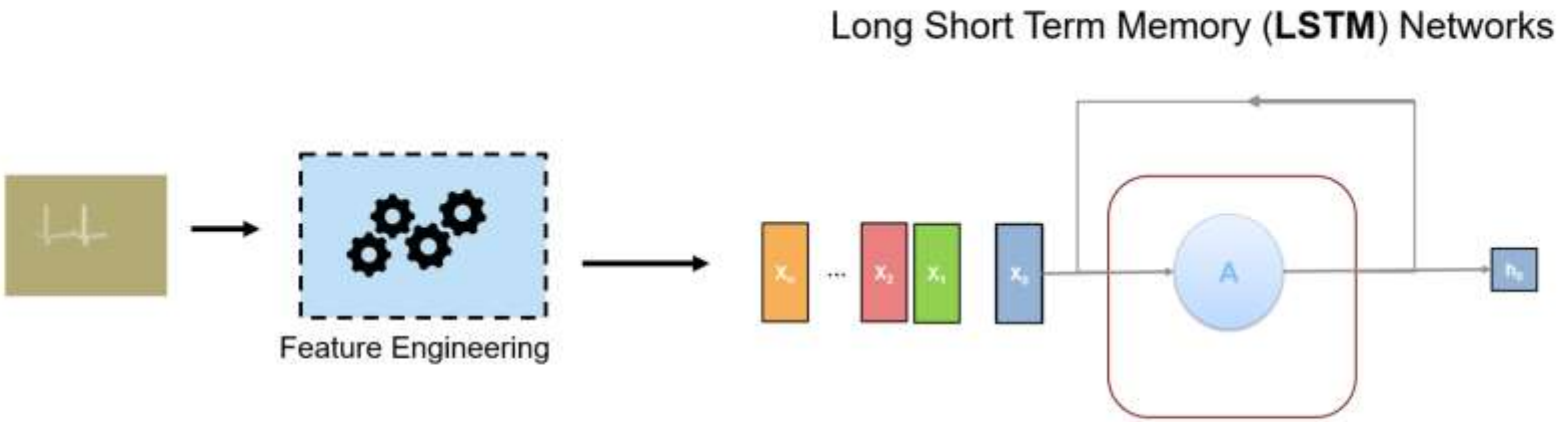
- Data cleansing and preparation
- Human insight
- Simulation-generated data



Data Preparation

Data Preparation

- Data cleansing and preparation
- Human insight
- Simulation-generated data



Start with a complete set of algorithms and pre-built models

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

Algorithms

Machine learning
Trees, Naïve Bayes, SVM…

Deep learning
CNNs, GANs, LSTM, MIMO…

Reinforcement learning
DQN, A2C, DDPG…

Regression
Linear, nonlinear, trees…

Unsupervised learning
K-means, PCA, GMM…

Predictive maintenance
RUL models, condition indicators…

Bayesian optimization

Pre-built models

Image classification models
AlexNet, GoogLeNet, VGG, SqueezeNet, ShuffleNet, ResNet, DenseNet, Inception…

Reference examples

Object detection
Vehicles, pedestrians, faces…

Semantic segmentation
Roadway detection, land cover classification, tumor detection…

Signal and speech processing
Denoising, music genre recognition, keyword spotting, radar waveform classification…

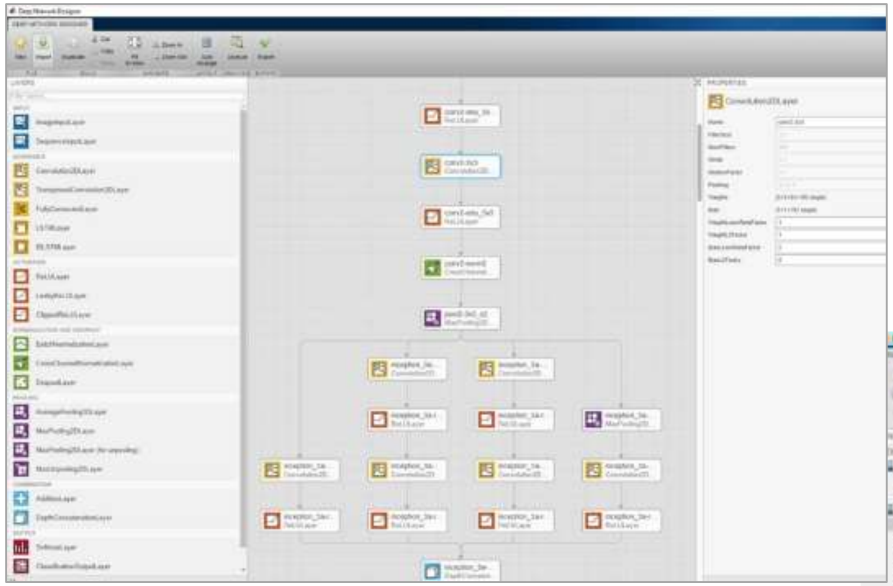
…and more…

Increase productivity using Apps for design and analysis

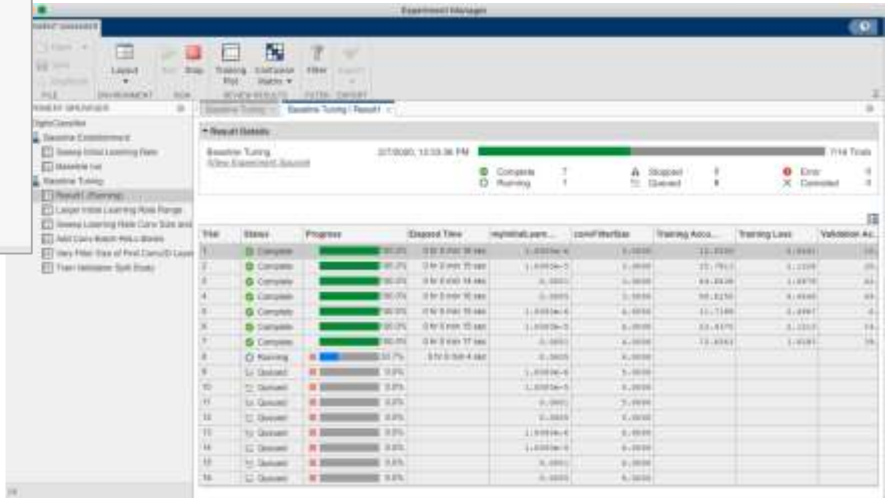
Use MATLAB Apps to design deep learning networks, explore a wide range of classifiers, train regression models, train an optical character recognition model, and more.

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability



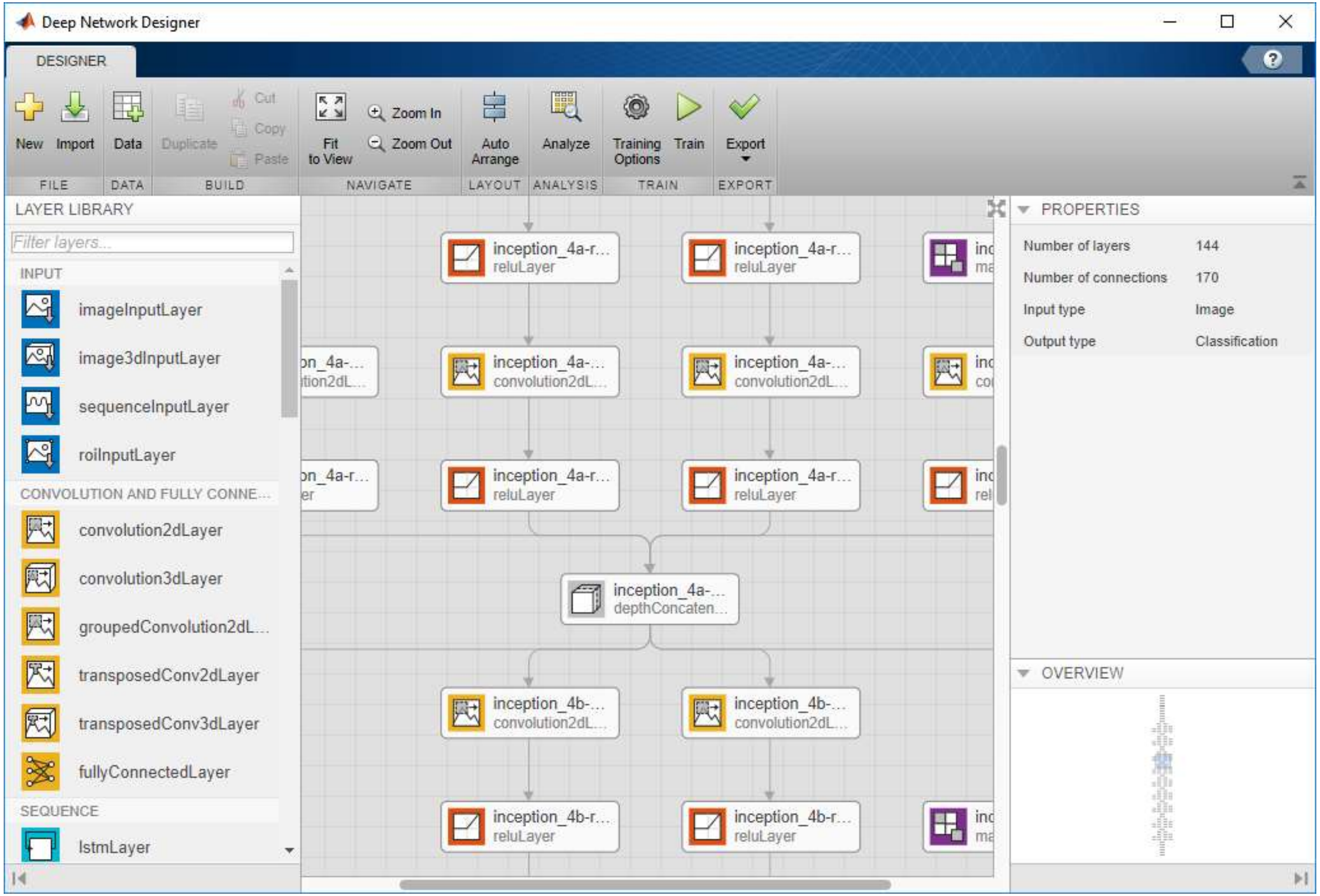
Deep Network Designer app to build, visualize, and edit deep learning networks



Experiment Manager app to manage multiple deep learning experiments, analyze and compare results and code

62

Deep Network Designer



Hardware acceleration and scaling are critical for training

MATLAB accelerates AI training on GPUs, cloud, and datacenter resources without specialized programming.

Azure

intel

NVIDIA

aws

docker

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability

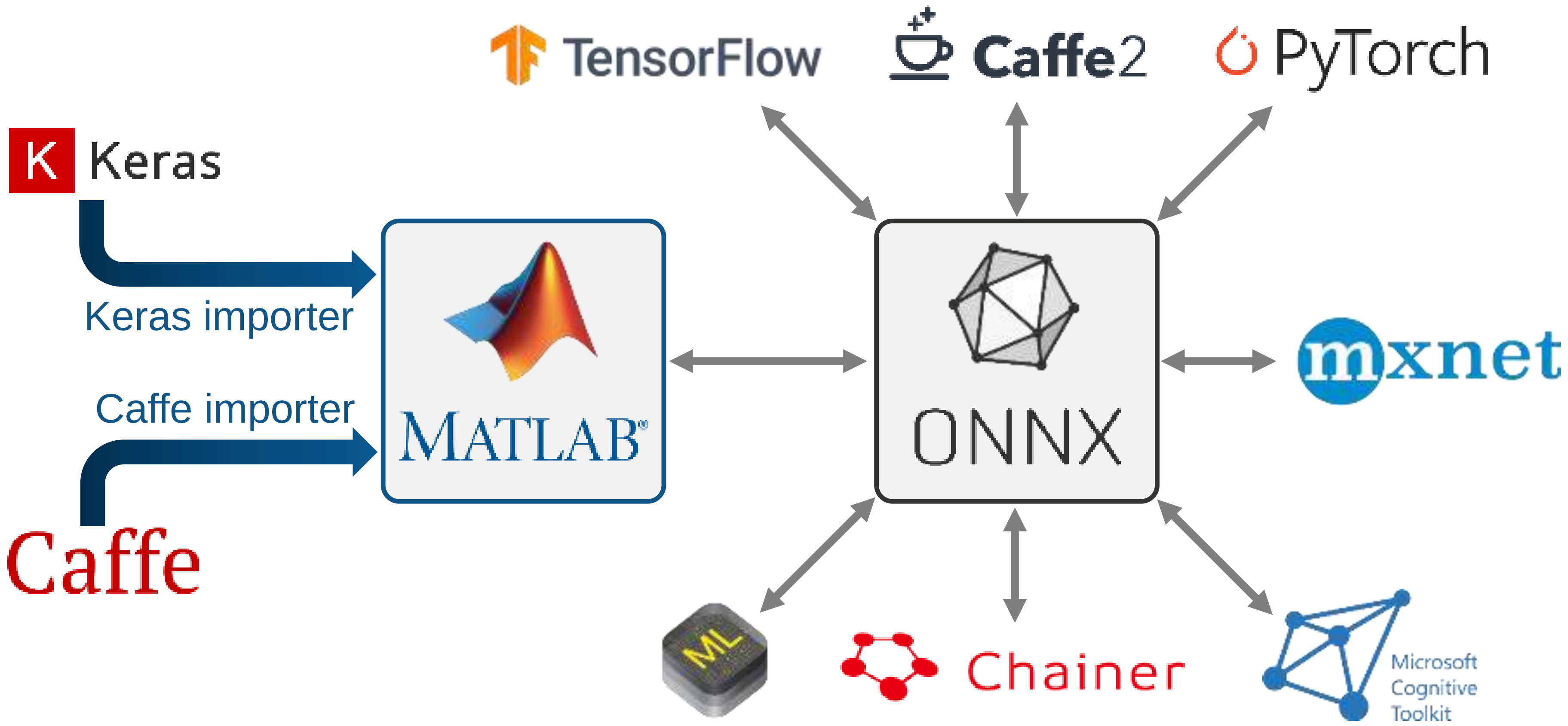
64

MATLAB interoperates with other frameworks

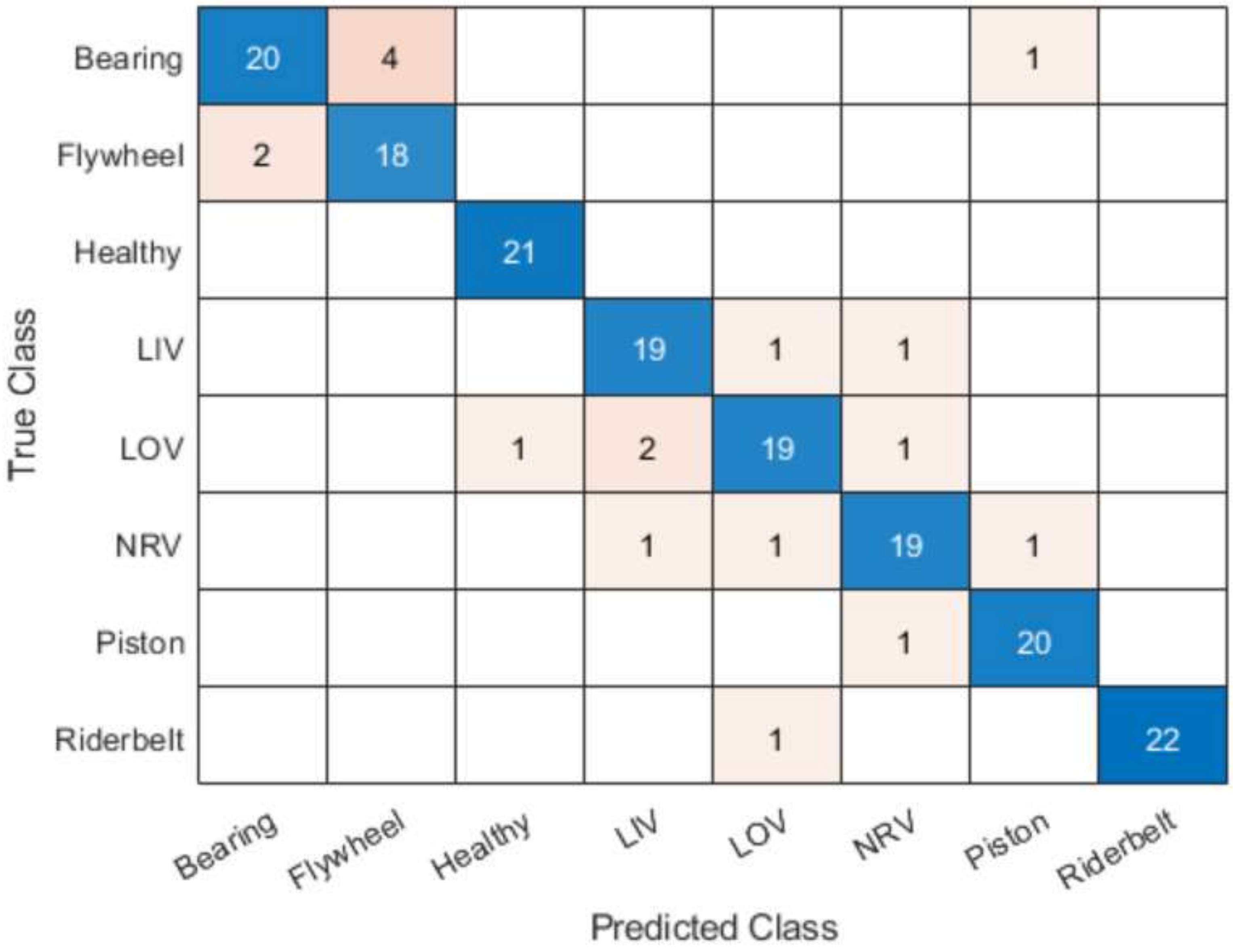
Supports ONNX and can exchange models with PyTorch, TensorFlow, and other frameworks.

AI Modeling

- Model design and tuning
- Hardware accelerated training
- Interoperability



Modeling



AI Modeling

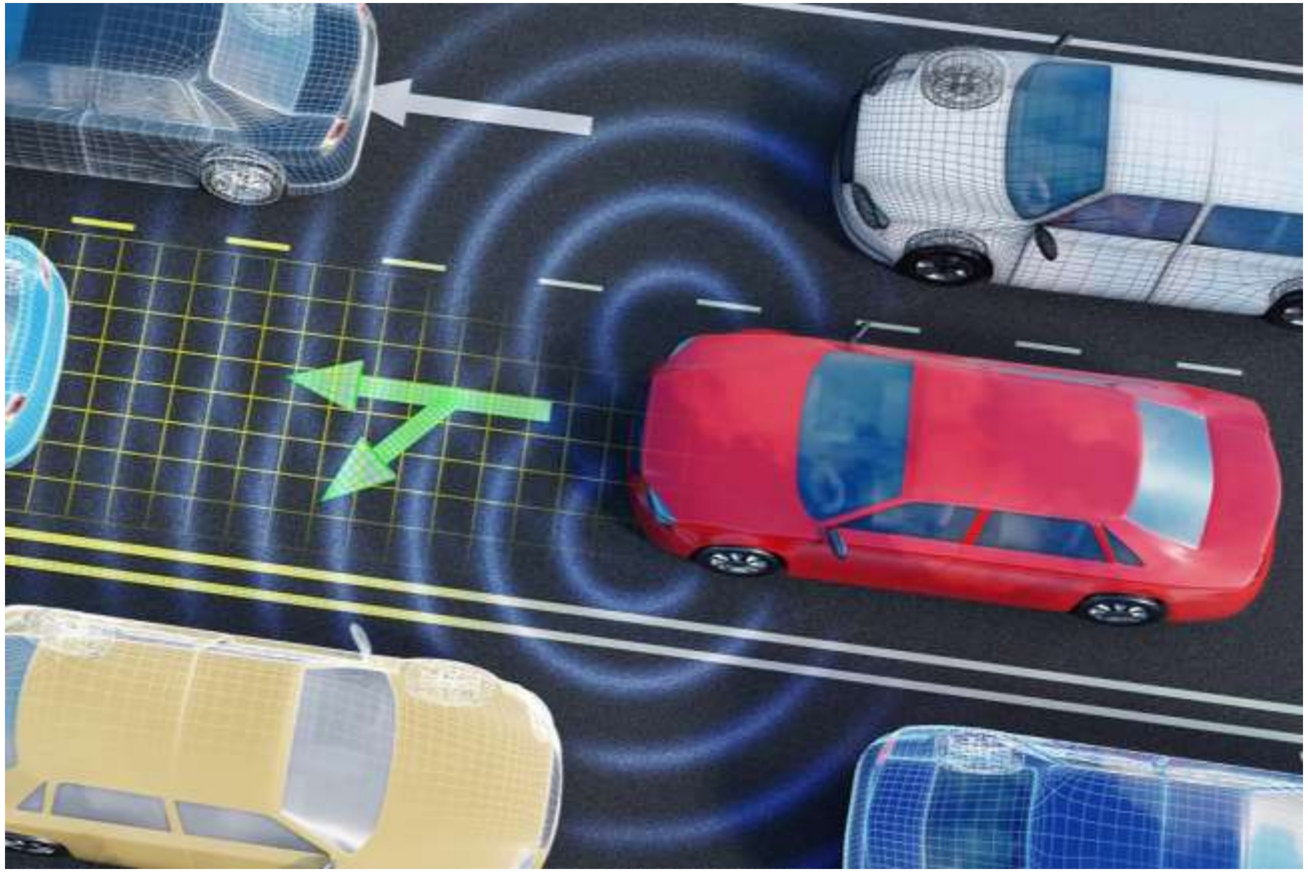
- Model design and tuning
- Hardware accelerated training
- Interoperability

Models need to exist within a complete system

In automated driving systems, AI for perception must integrate with algorithms for path planning, braking, acceleration, and other controls.

Simulation & Test

- Integration with complex systems
- System simulation
- System verification and validation

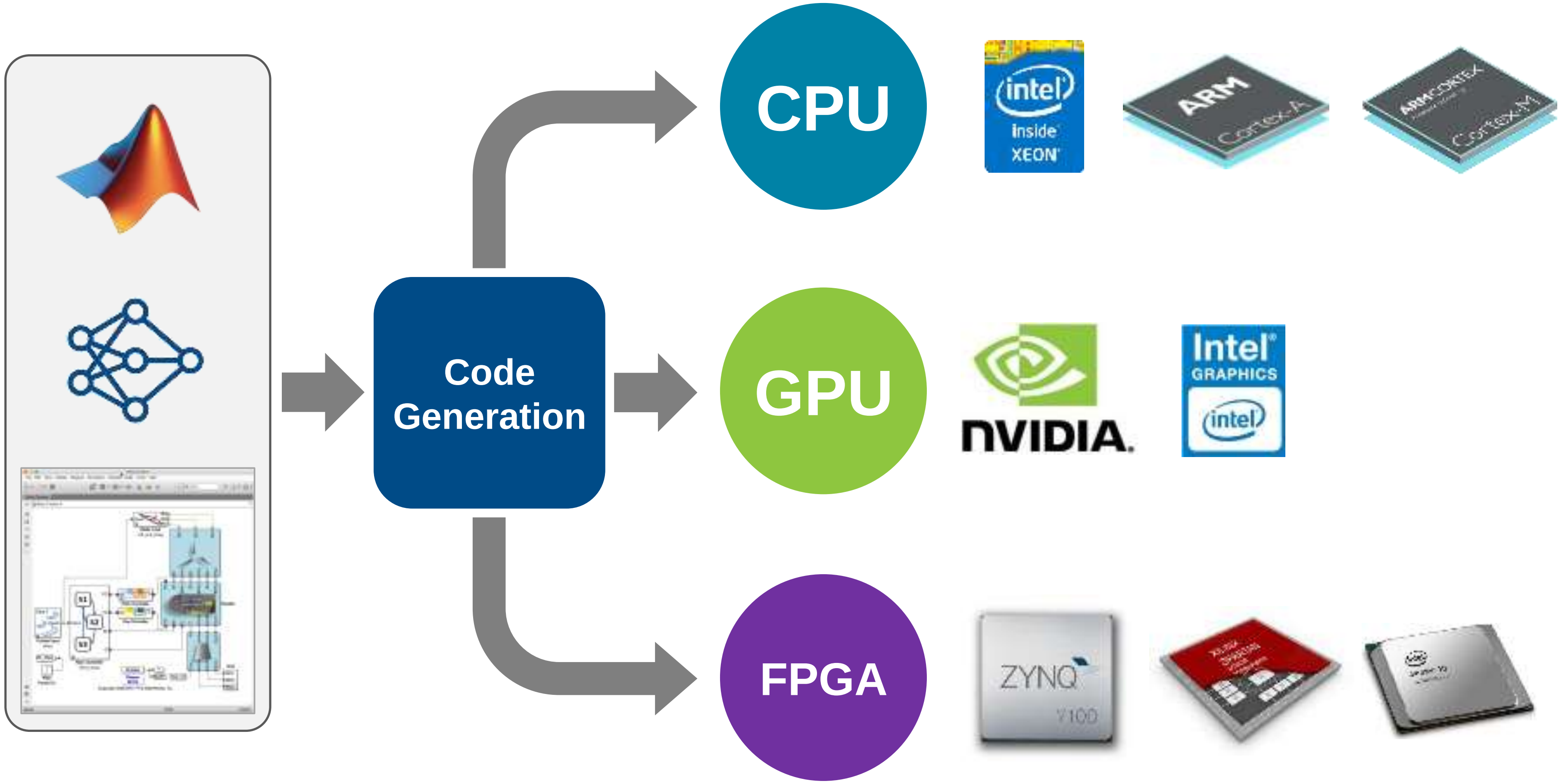


Deploy to any processor with best-in-class performance

AI models in MATLAB and Simulink can be deployed on embedded devices, edge devices, enterprise systems, the cloud, or the desktop.

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop

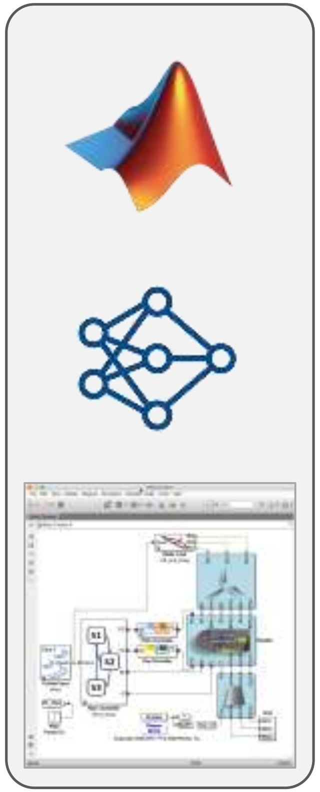


68

Deploy to enterprise IT infrastructure

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop



MATLAB Production Server

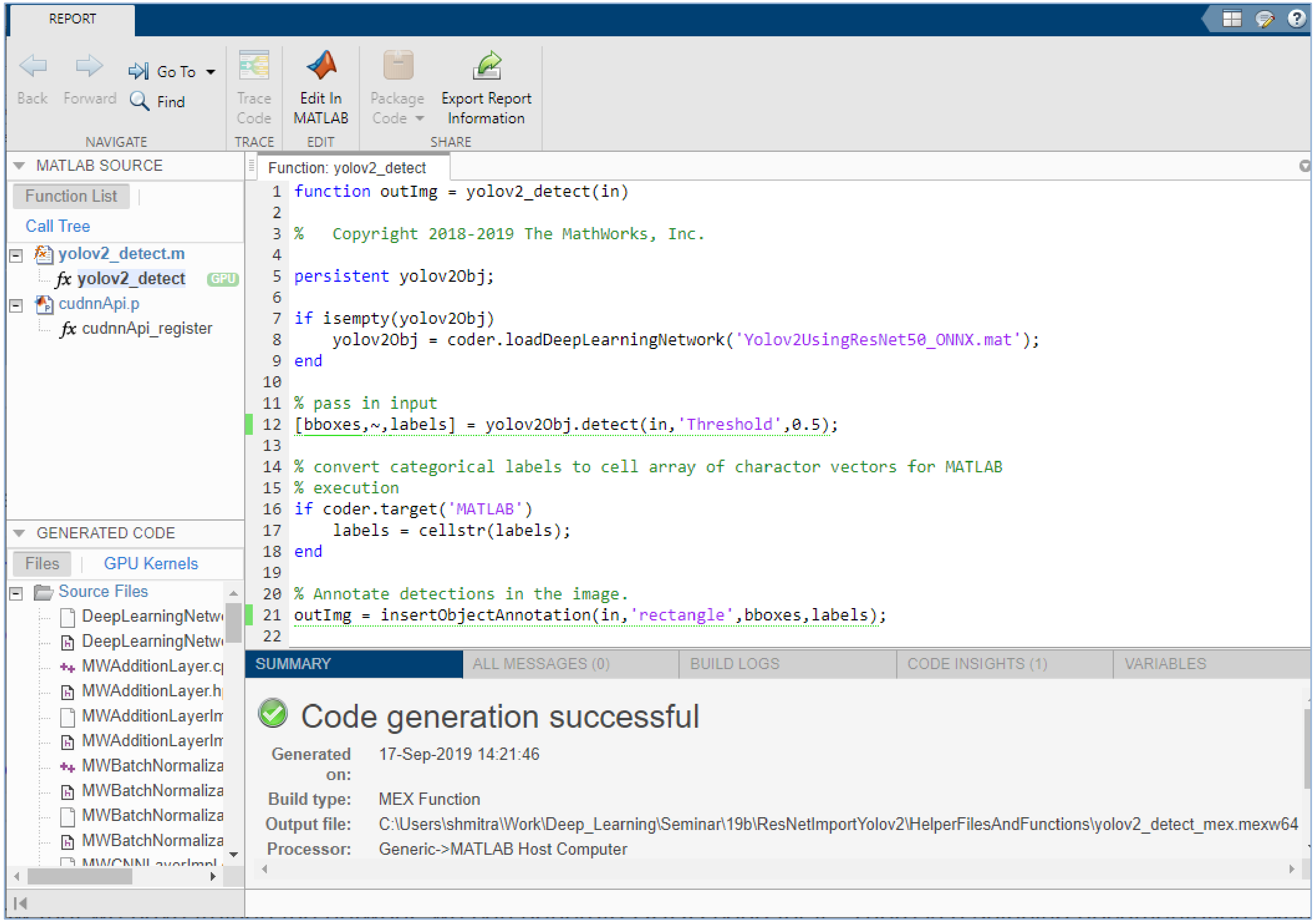
- Databases
- Streaming
- Cloud Storage
- Dashboards
- Containers
- Custom Tools
- Cloud & Datacenter Infrastructure

69

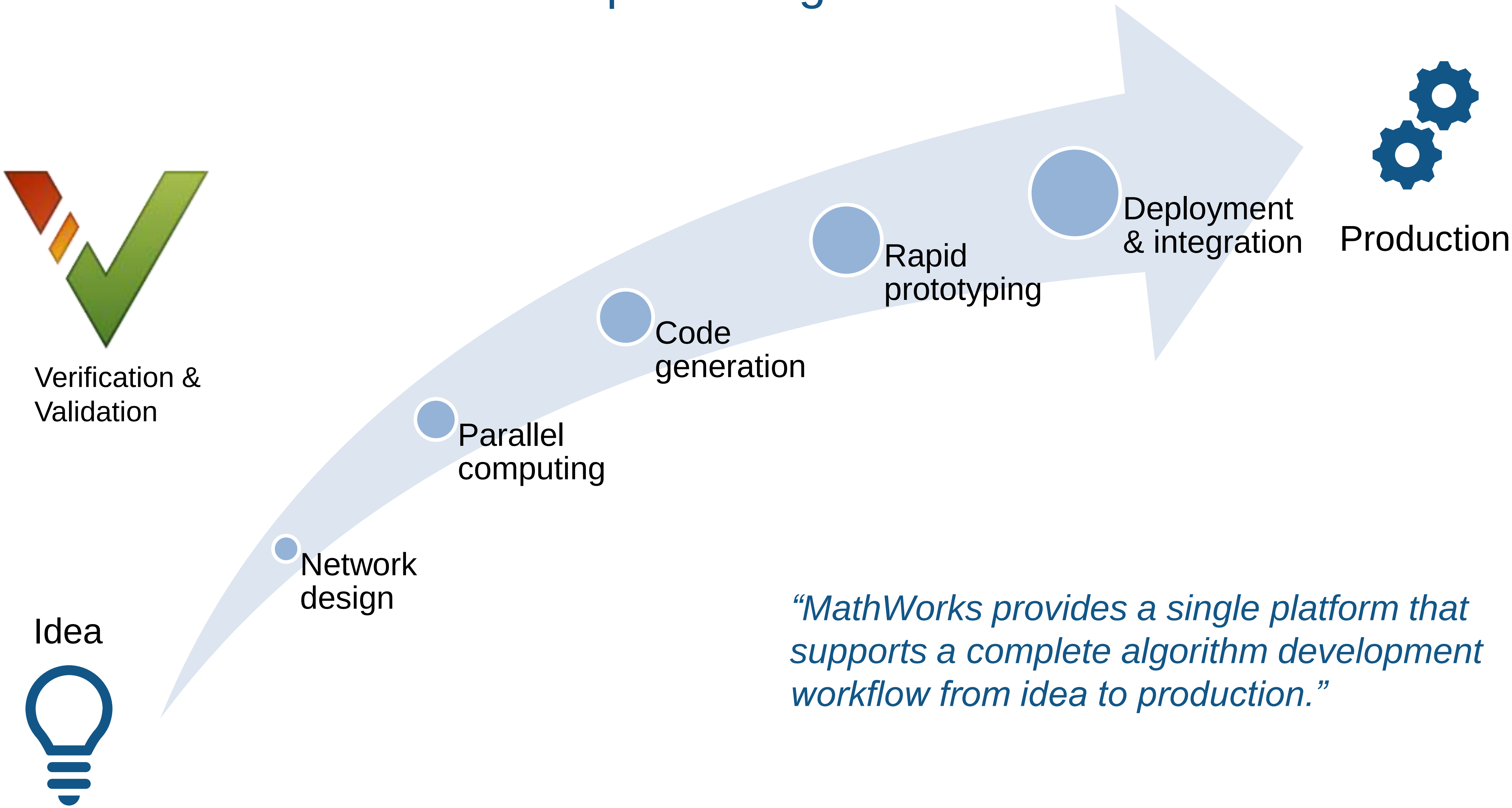
Deployment

Deployment

- Embedded devices
- Enterprise systems
- Edge, cloud, desktop



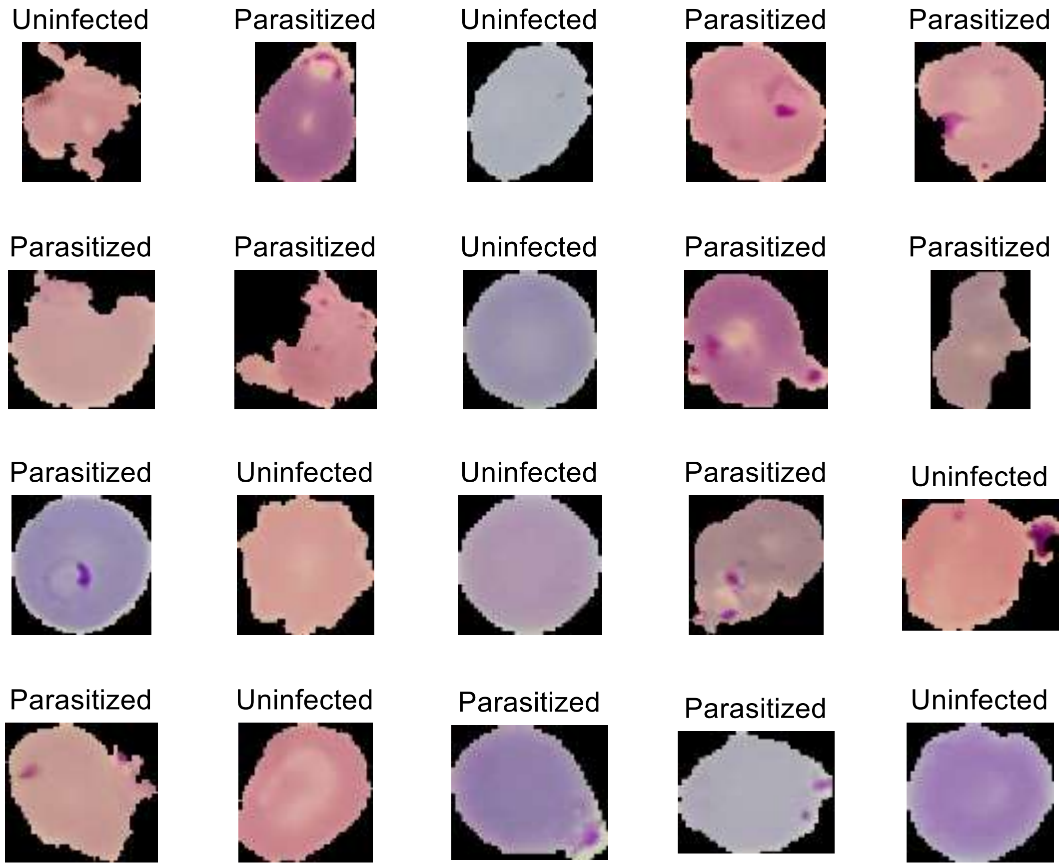
MathWorks solutions for deep learning



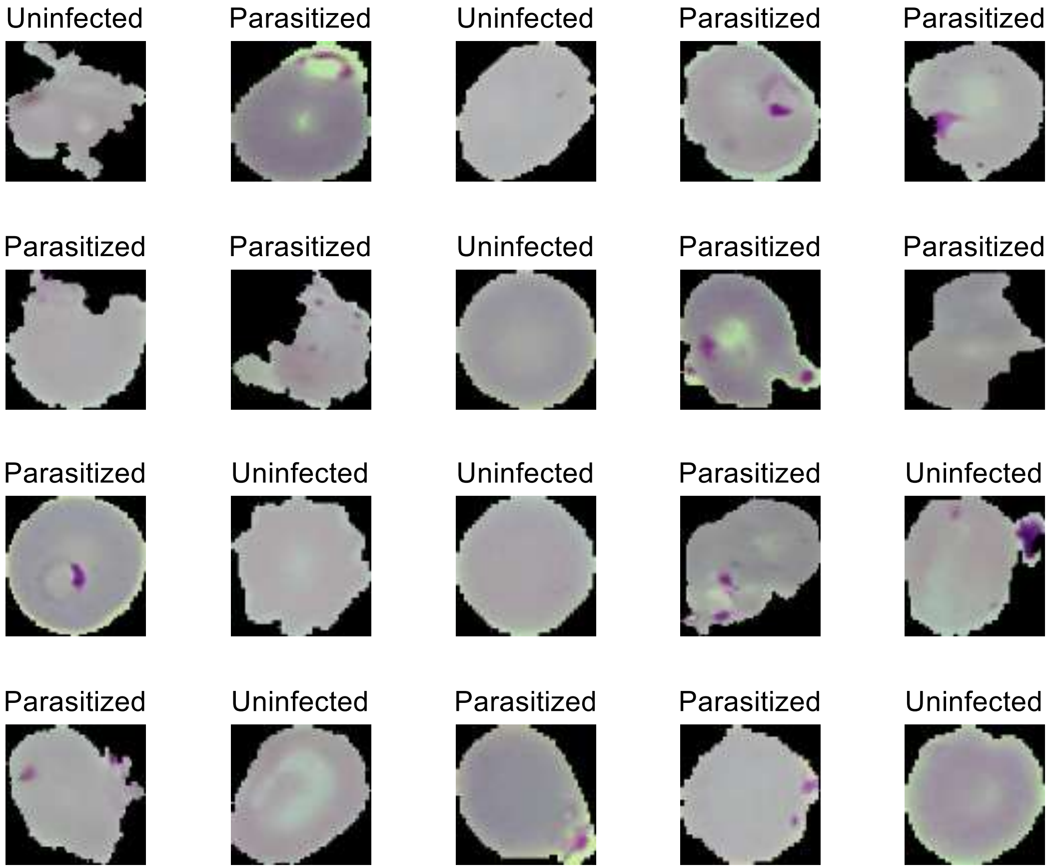
"MathWorks provides a single platform that supports a complete algorithm development workflow from idea to production."

Demo: Malaria Detection

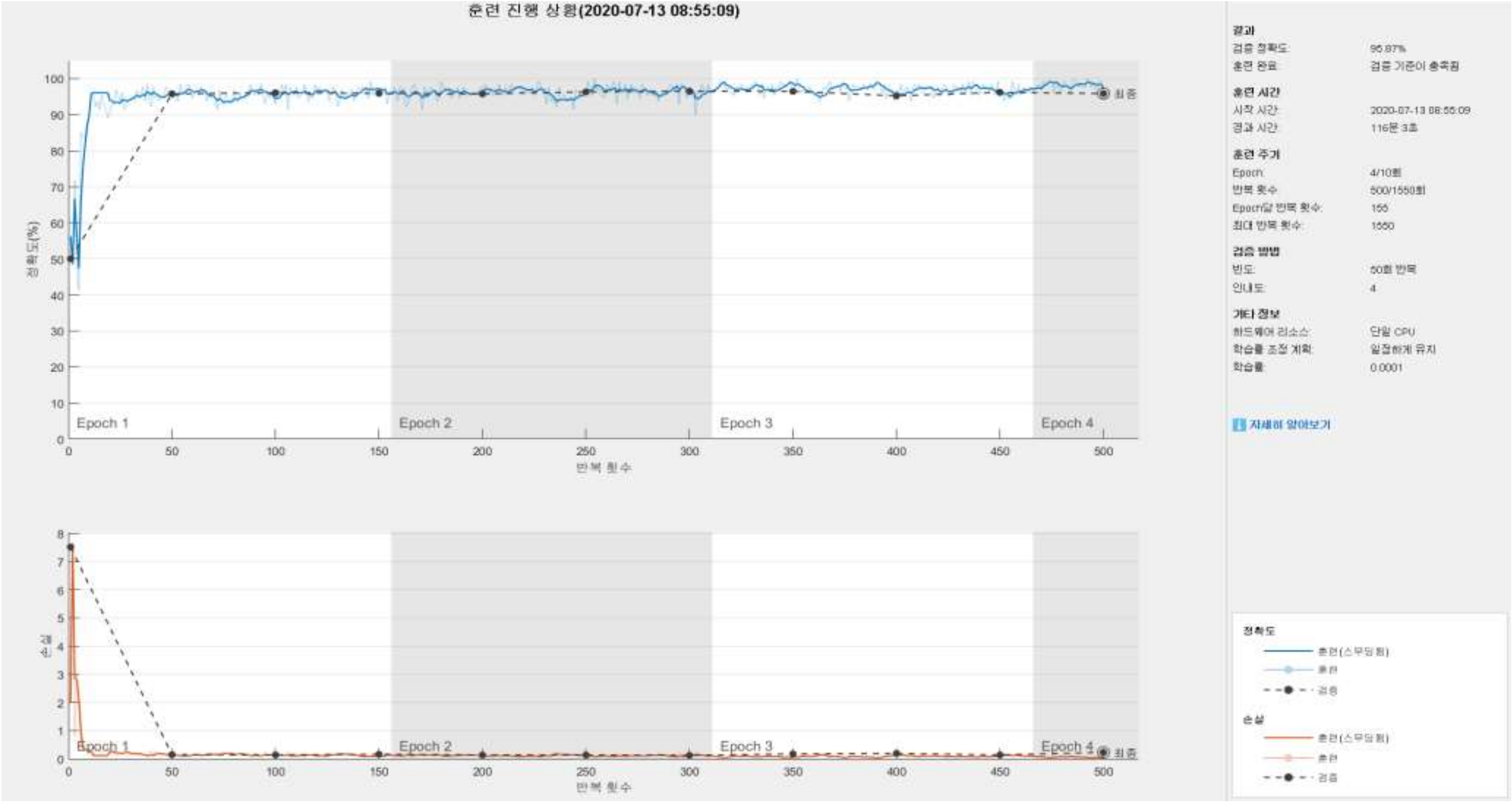
	Label	Count
1	Parasitized	13779
2	Uninfected	13779



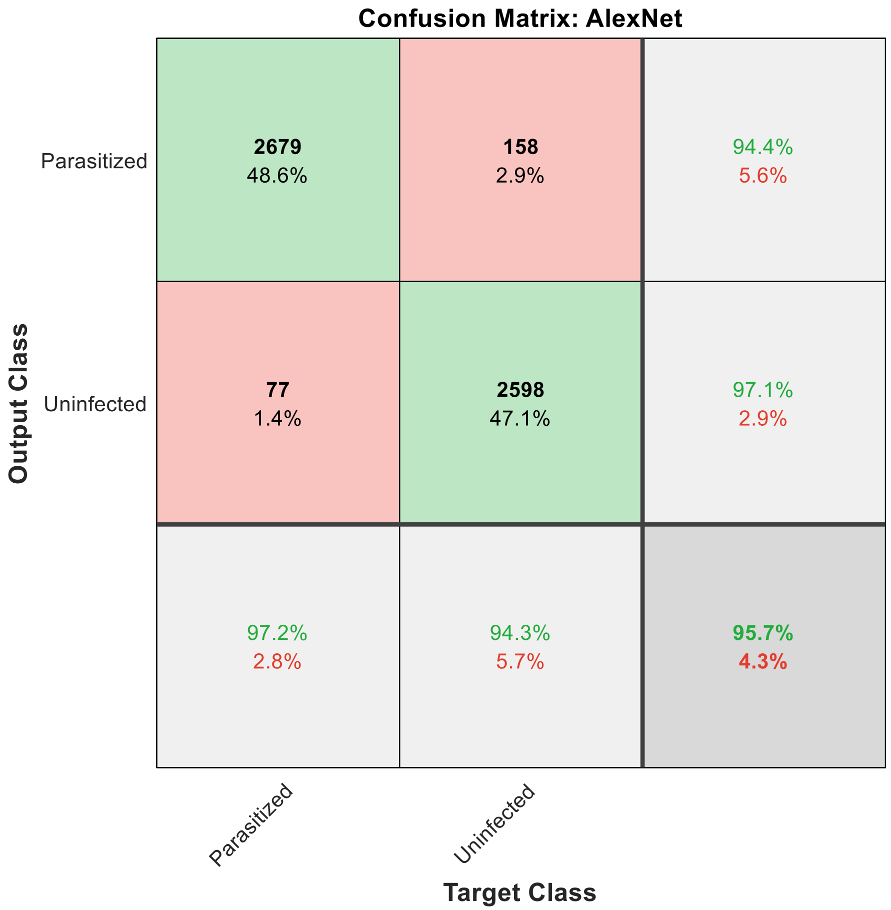
Demo: Malaria Detection

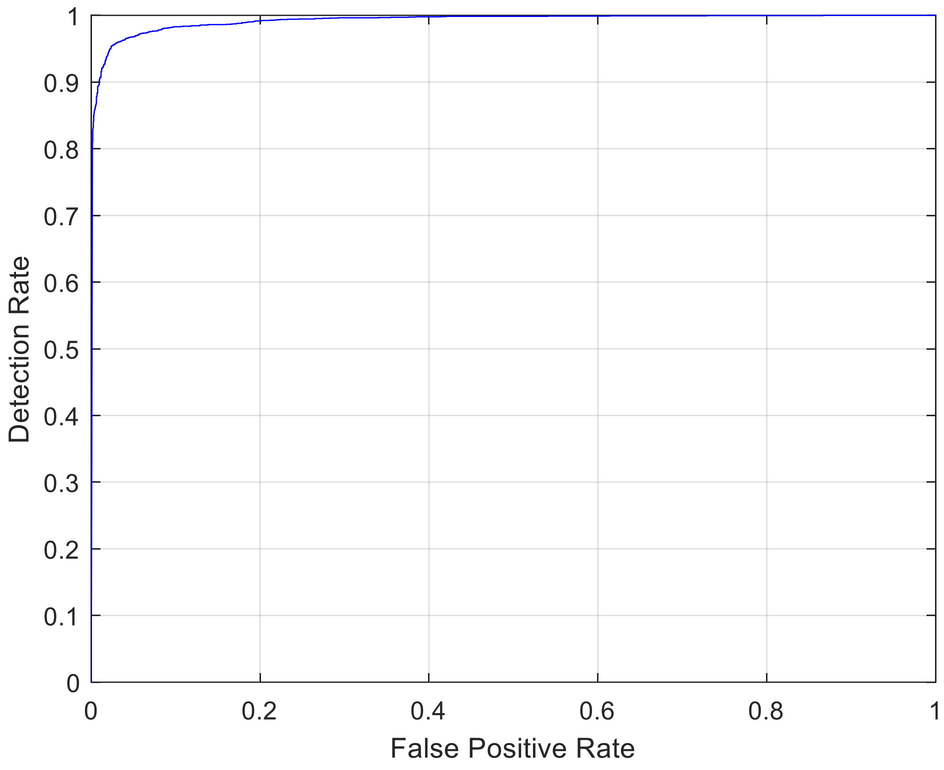


Demo: Malaria Detection



Demo: Malaria Detection





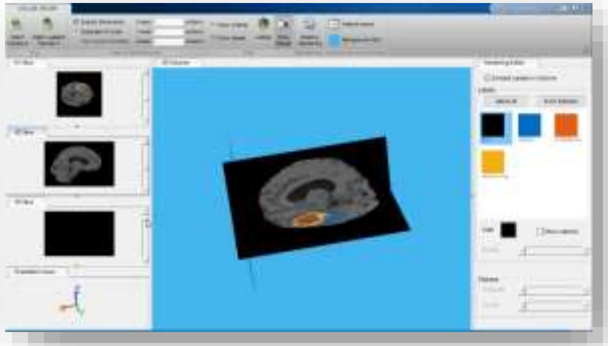
Interactively design and edit neural networks

- **Use deep learning for medical imaging tasks** such as segmentation, classification and detection

- **Interactively create and edit deep learning networks**
  - Built-in Deep Network Designer app

- Analyse network architecture to **detect errors and layer compatibility issues** before training

Interactively build and visualise network structures

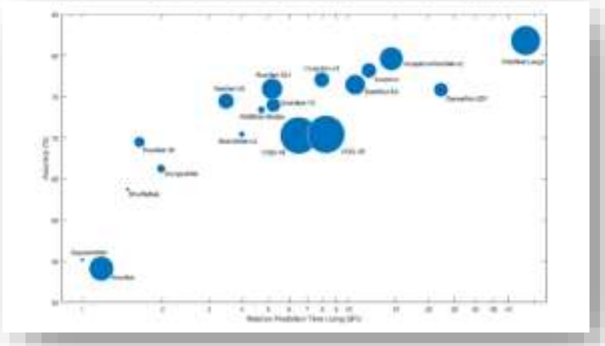


Segmentation of brain tumours in 3D images using deep learning

Import pre-trained models for fast implementation

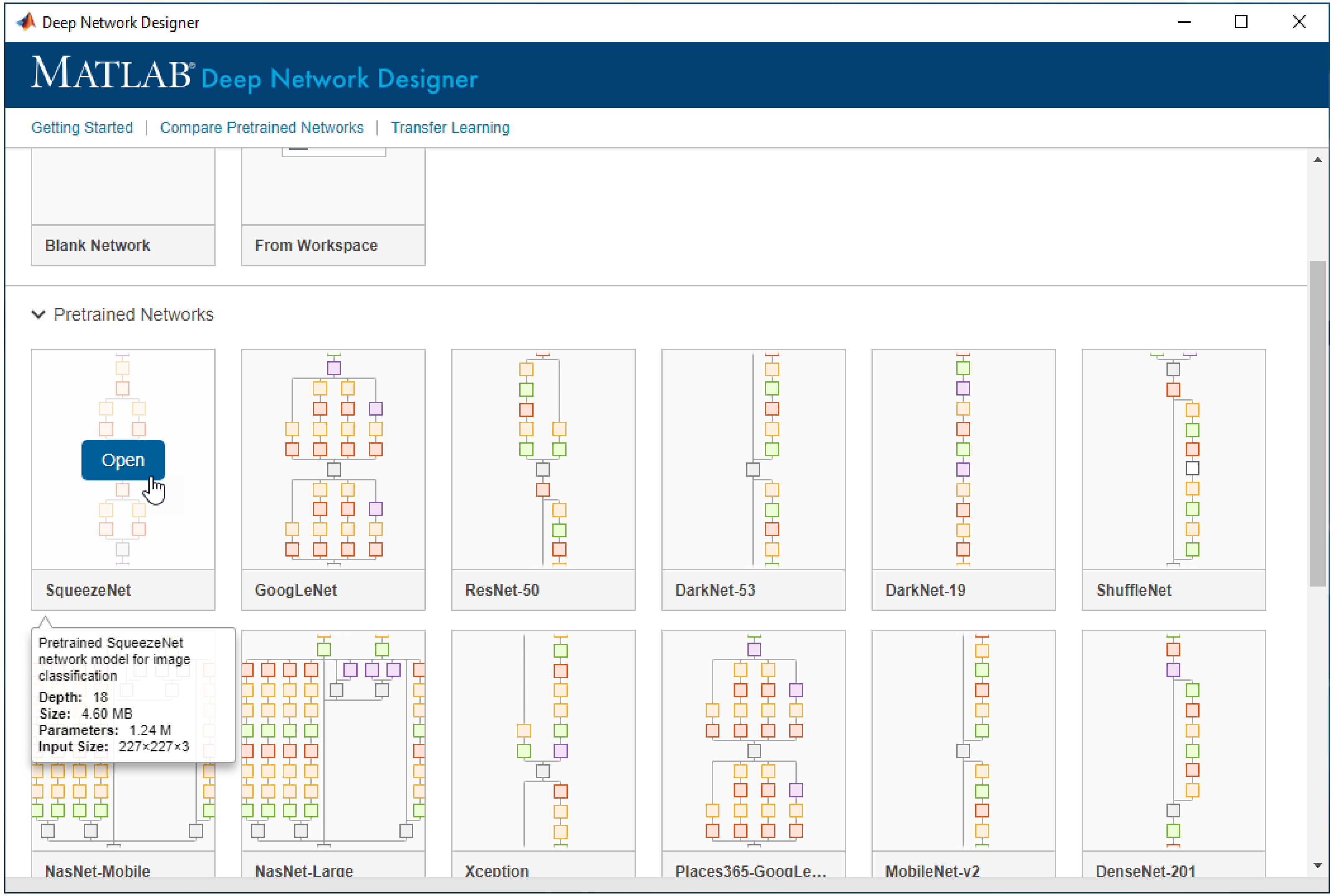
- **Access pretrained networks** and use them as a starting point for new models
  - Multiple pre-trained networks available online
- **Perform transfer learning** to use the learned features in the network for a specific task
- **Compare the accuracy of pre-trained networks** for a specific medical imaging task

A list of pretrained networks



Analysis of pretrained models

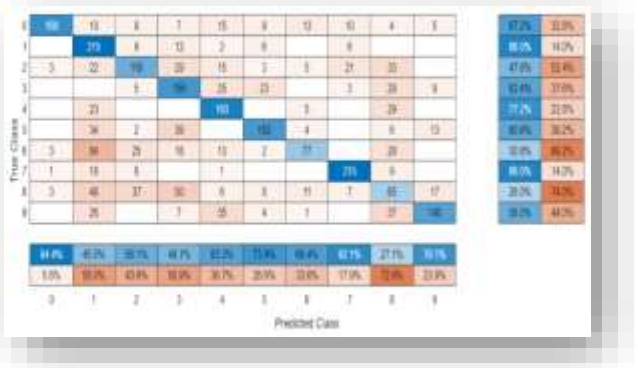
Example: A list of pretrained networks
Deep Network Designer



Track training progress and compare performance

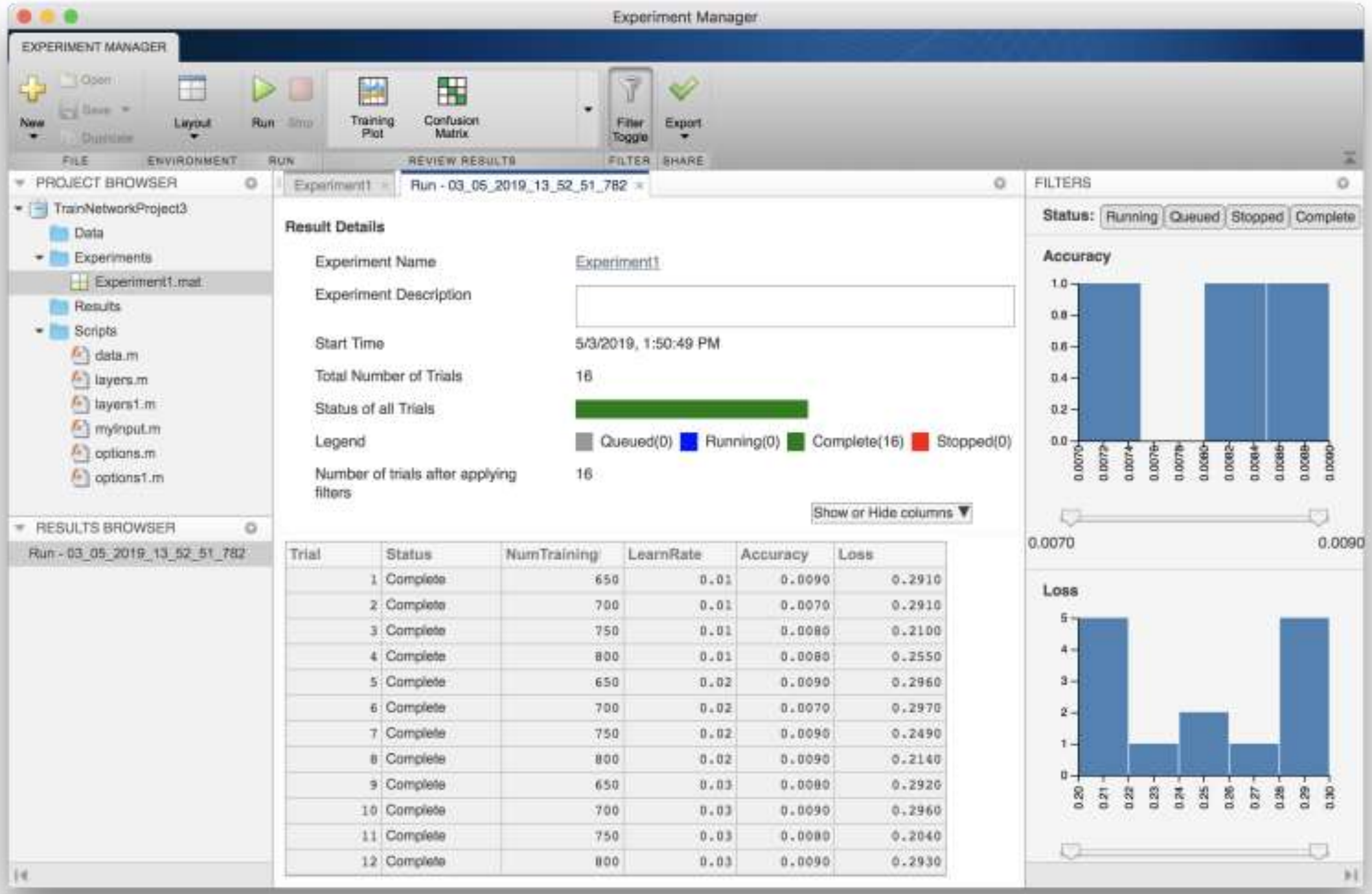
- **Train networks under various initial conditions** and compare the results
  - Built-in Experiment Manager app
- **Easily compare different network architectures** using the same training data
- **Use custom metrics** to compare the performances of deep learning modes

Design and run experiments to train and compare deep learning networks



Model performance evaluation using confusion matrix

Example: Manage multiple deep learning experiments
Experiment Manager



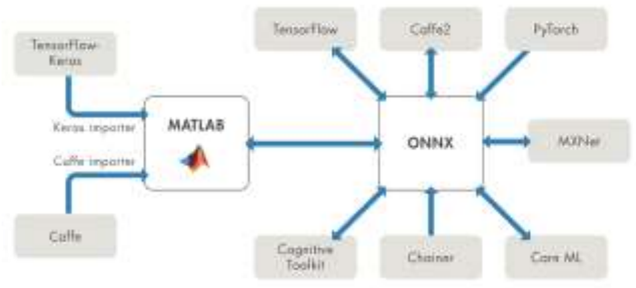
Interoperate with deep learning frameworks

- **Import models from TensorFlow-Keras and Caffe** for inference and transfer learning

- **Exchange models in ONNX format** for working with other deep learning frameworks

- **Export trained MATLAB deep learning networks** to ONNX model format for sharing models

Import models from other deep learning frameworks

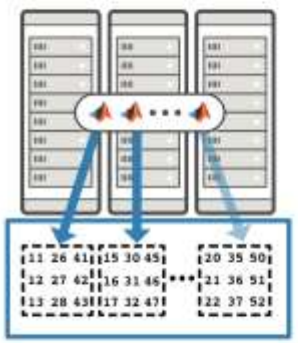


Exchange models using ONNX format

Accelerate model training in the cloud

- Train models on different environments using **MATLAB Deep Learning Container**
- **Reduce deep learning training times** with cloud instances
  - **AWS** and **NVIDIA GPU Cloud** support
- Run deep learning training across **multiple processors on multiple servers**

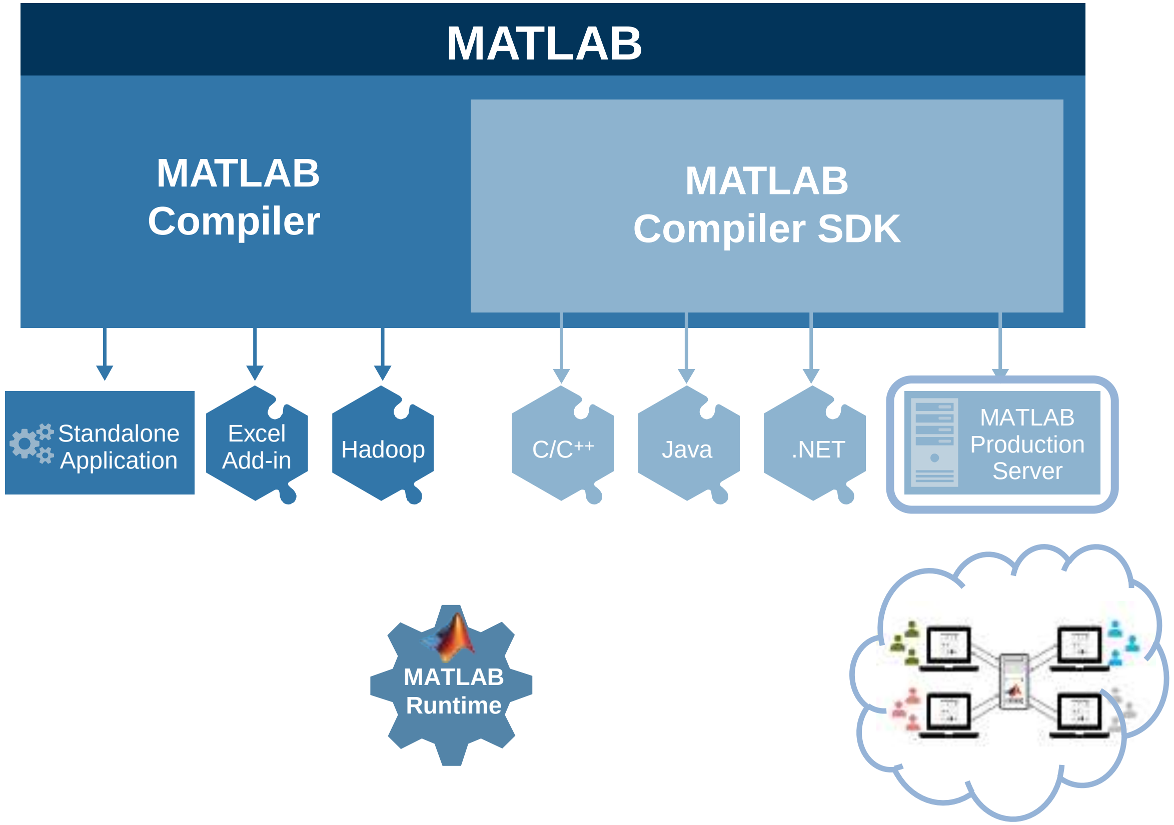
Accelerating training in the cloud



Run training across multiple processors on multiple servers

Share trained deep learning models with end users

- **MATLAB Compiler** for sharing MATLAB programs without integration programming
- **MATLAB Compiler SDK** provides implementation and platform flexibility for software developers
- **MATLAB Production Server** provides the most efficient development path for secure and scalable web and enterprise applications



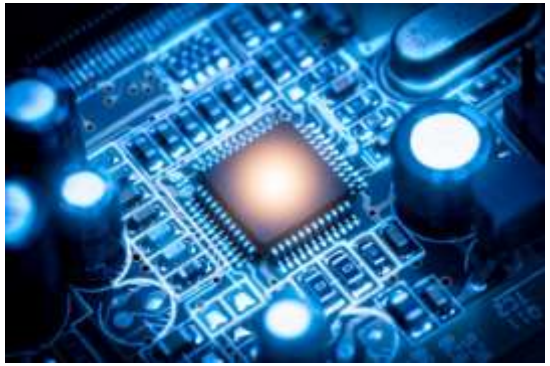
Deploy trained networks to embedded systems

- **Generate optimised CUDA code** for performance-critical applications

- **Generate C++ code** to deploy deep learning networks to **Intel** and **ARM** processors

- **Deploy trained networks** as C++ shared libraries, Microsoft .NET assemblies, Java classes and Python packages

Generate CUDA code for optimised performance on GPUs



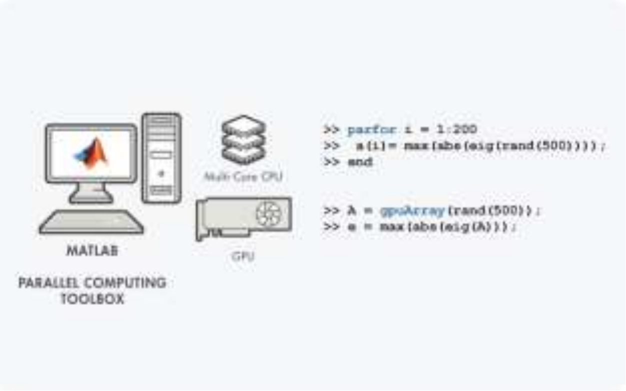
Optimise C++ code for embedded processors

Accelerate algorithm deployment by running in parallel

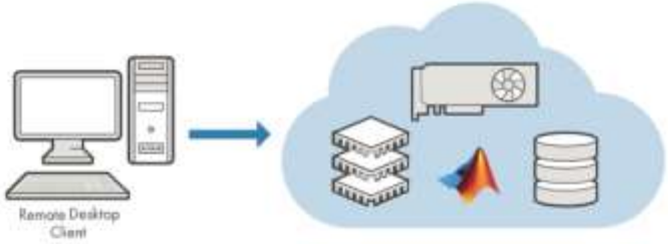
- **Speed up your deep learning models** by using GPU and multicore CPU processors

- **Legacy MATLAB algorithms** can run on GPUs with minimal code changes

- **Run deep learning models directly on virtual clouds** such as Amazon Web Services (AWS) or Microsoft Azure



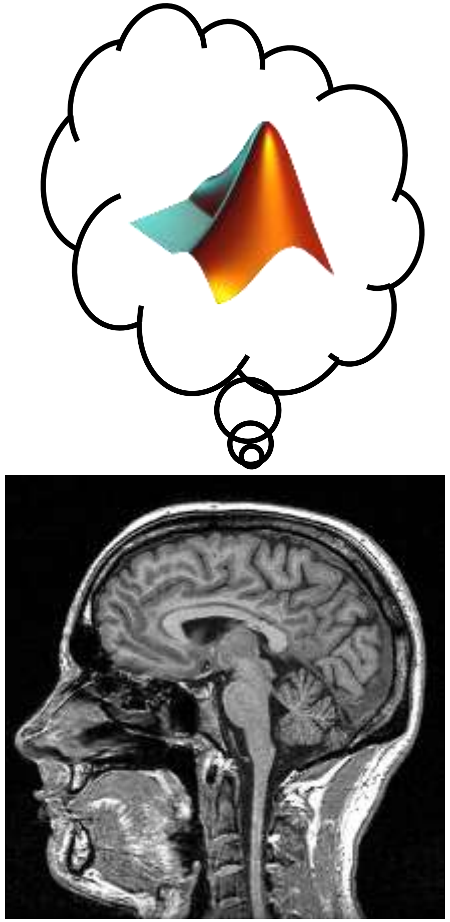
Deploy MATLAB algorithms and deep learning models on GPUs and multicore CPUs



Run MATLAB algorithms directly on EC2 instances in the Amazon Web Services (AWS)

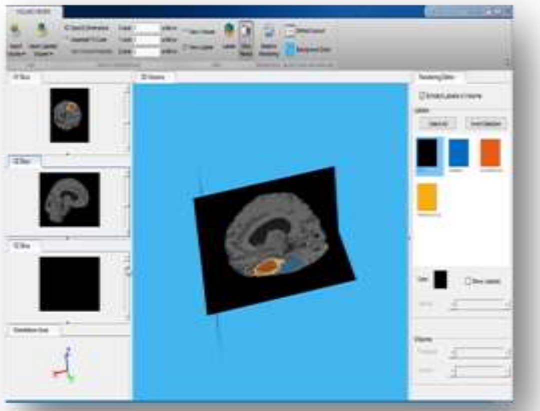
Why MATLAB & MathWorks for Deep Learning?

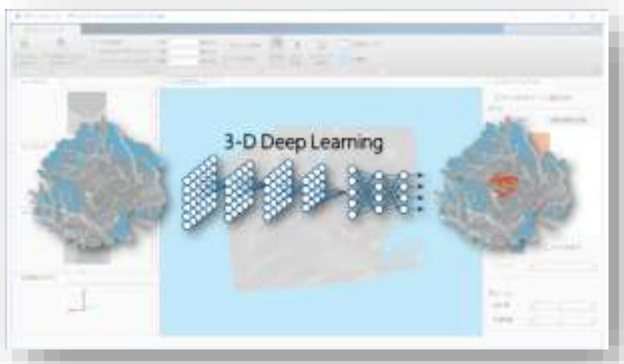
- ✓ MATLAB provides **integrated and complete workflow**
- ✓ **Pre-trained models and interactive tools** allow fast implementation from concept to code
- ✓ **Trained models can be shared to end users** easily online or using executable apps
- ✓ **Automatic code generation** enables deployment on embedded production hardware
- ✓ Support offered by **engineering support, comprehensive documentation, demos and application examples**



Online examples of deep learning in medical imaging

- [Deep Learning Toolbox]
- [3D Image Segmentation of Brain Tumors Using Deep Learning]
- [Deep Learning for Medical Imaging: Malaria Detection]
- [Medical Image Segmentation Using SegNet]
- [3-D Deep Learning : Lung Tumor Segmentation]
- [3-D Deep Learning : 3-D Volume Labeling Assist Tool]





For more information visit:

mathworks.com/medical

Evolution of Deep Learning in MATLAB

2016

CNN's

Pretrained Models

Caffe Importer

2017

Name Change

- Neural Network Toolbox to Deep Learning Toolbox

Algorithms

- LSTM's
- Directed Acyclic Graphs
- Multi-GPU Training

Code Generation

- GPU Coder

Apps

- Image Labeler

Interoperability

- TensorFlow-Keras Importer

2018

Examples

- Signal Processing
- Audio
- Text Analytics

Algorithms

- Wavelet Scattering

Code Generation

- MATLAB Coder C++

Apps

- Deep Network Designer
- Video Labeler
- Audio Labeler

Interoperability

- ONNX Support

2019

- Reinforcement Learning

Algorithms

- Automatic Differentiation
- Custom Training Loops
- Weight Sharing
- Big Image

Examples

  - GANs
  - Siamese Network
  - Autoencoders
- 3-D support
- **Explainable AI**
  - Occlusion
  - Grad-CAM

Code Generation

- MATLAB Coder (ARM)

Apps

- Signal Labeler

2020

- Deep Learning Data Sets

Apps

- Experiment Manager

Examples

- 5G Communications
- **Over 200+ examples!**

Algorithms

- Point Cloud

Code Generation

- Quantization

MathWorks

is a **Leader** in the Gartner Magic Quadrant for 2020 Data Science and Machine Learning Platforms

Figure 1. Magic Quadrant for Data Science and Machine Learning Platforms

Source: Gartner (February 2020)

*Gartner Magic Quadrant for Data Science and Machine Learning Platforms, Peter Krensky, Erick Brethenoux, Jim Hare, Carlie Idoine, Alexander Linden, Svetlana Sicular, 11 February 2020 .

Further Learning & Teaching

- Deep Learning Onramp
  - 2 hr online tutorial

- Deep Learning Workshop
  - 3 hr hands on session
  - Contact us to schedule

- Deep Learning Training
  - 16 hr in depth course
  - Online or Instructor Lead

- Teaching Deep Learning with MATLAB
  - Curriculum support

MathWorks Engineering Support

Training

Guided Evaluations

Onsite Workshops

Consulting

Technical Support

The Platform

MATLAB, Simulink, and over 100 add-on products for specialized applications

Your People

Helping you build an agile workforce today and preparing tomorrow’s engineers

Our Expertise

From onboarding and implementation to solving advanced engineering challenges

94