

Low-Code AI: Making AI Accessible to Everyone

MATLAB® & SIMULINK®

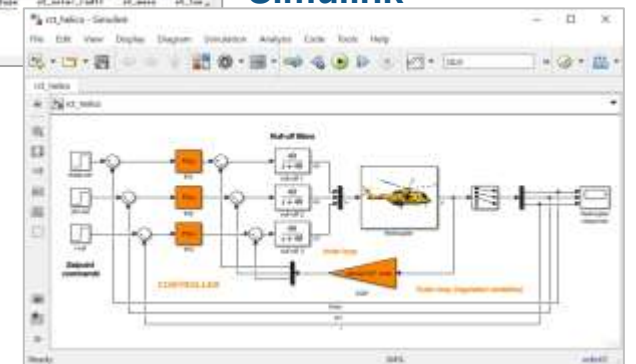


- **MATLAB** – Create algorithms and AI models for biomedical data analysis
- **Simulink** – Simulate complex medical devices with sensors and software
- **Over 100 add-on products** for specialized R&D tasks

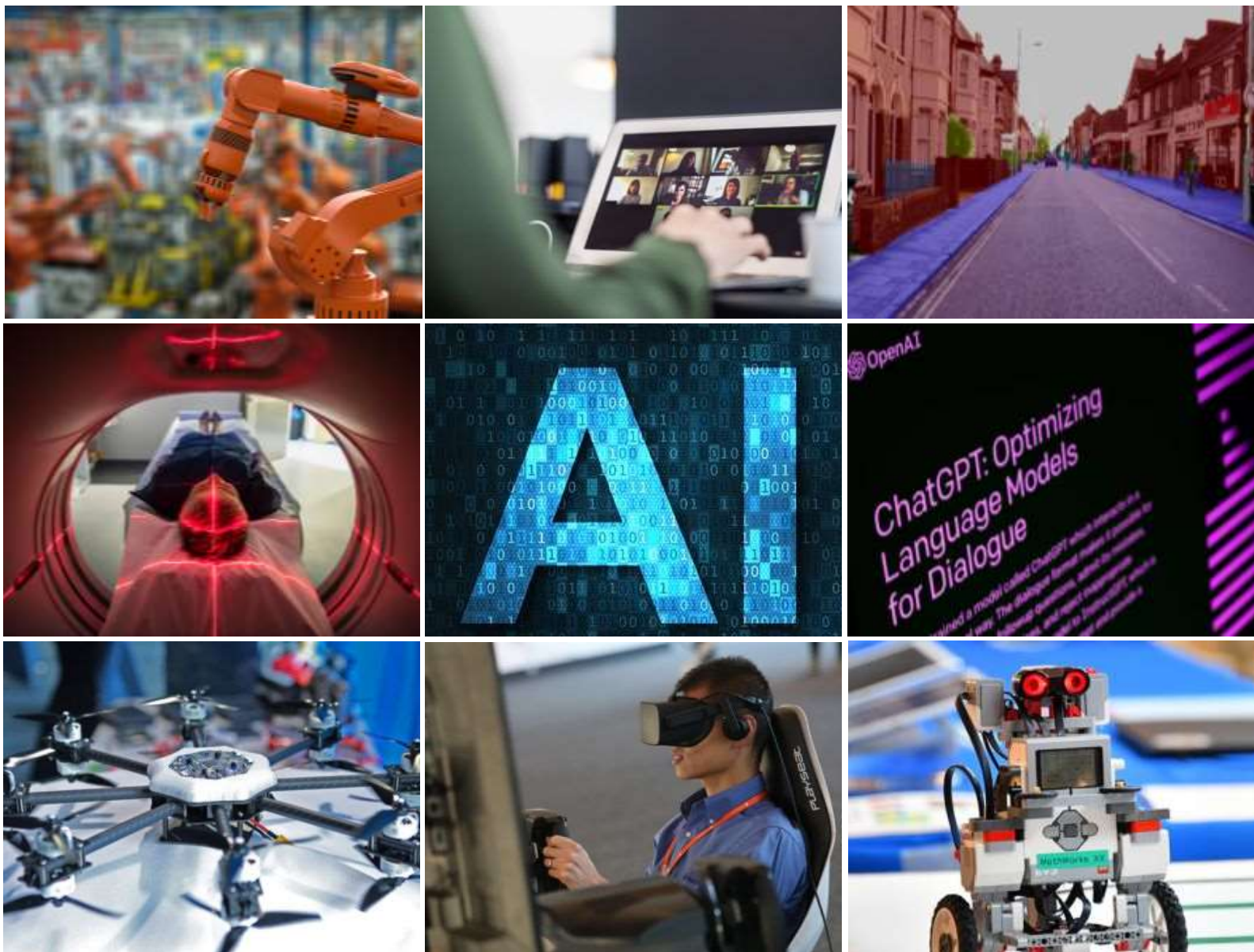
MATLAB



Simulink



AI 无处不在



有哪些缺失？



Domain Expertise



Generic AI knowledge

需要什么？

将 AI 作为服务，用于特定于领域的应用程序



Researchers
equipped
with AI skill

什么是“低代码”工具？

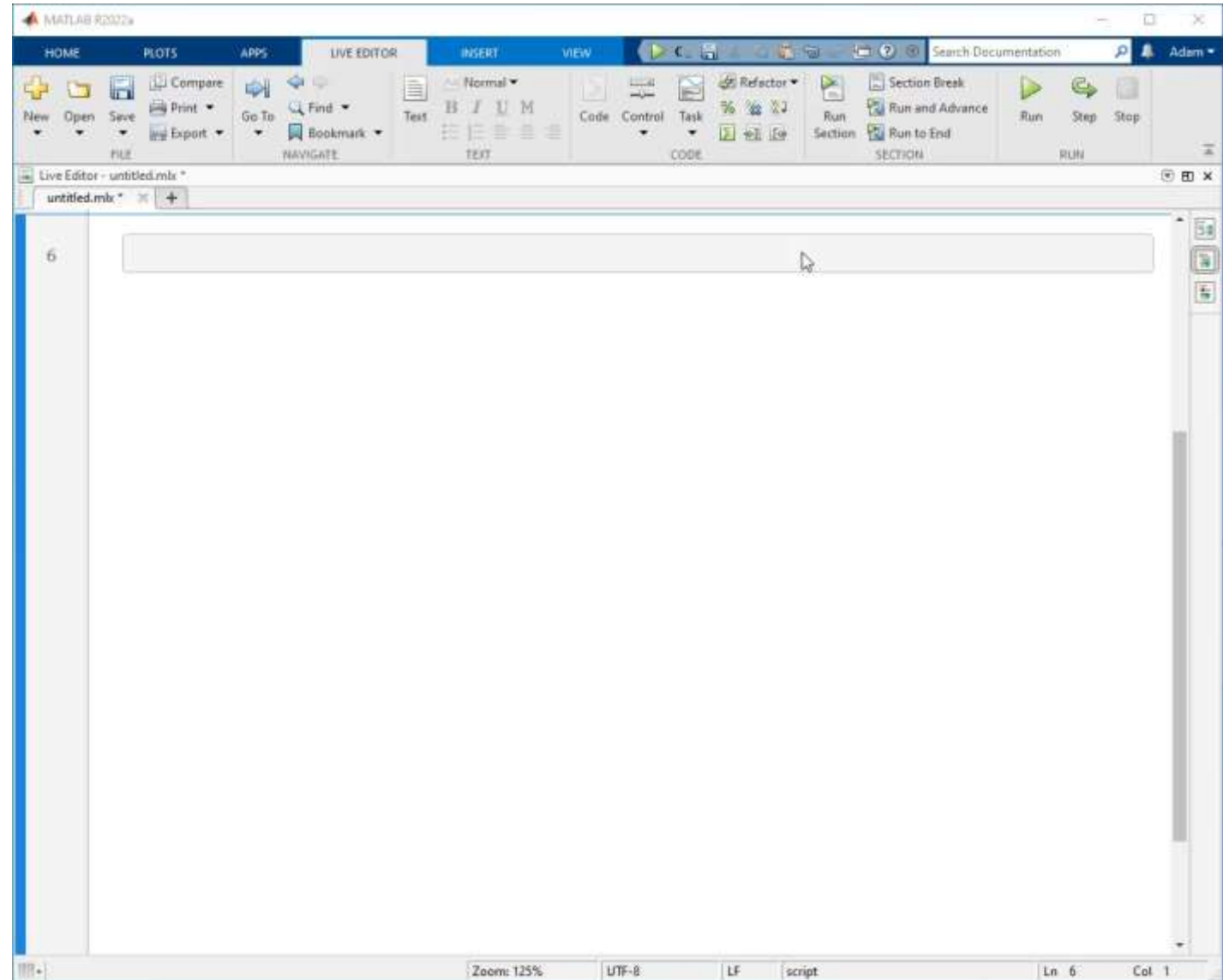
Low code tools enable:

- rapid software development
- minimal manual coding

Benefits of low code tools:

- Shallow learning curve
- Teach *how* to code
- Solve task first, code later

Not just for beginners



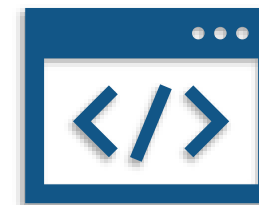
学习成果



Fundamental knowledge of
AI



Design and train AI models
with interactive tools



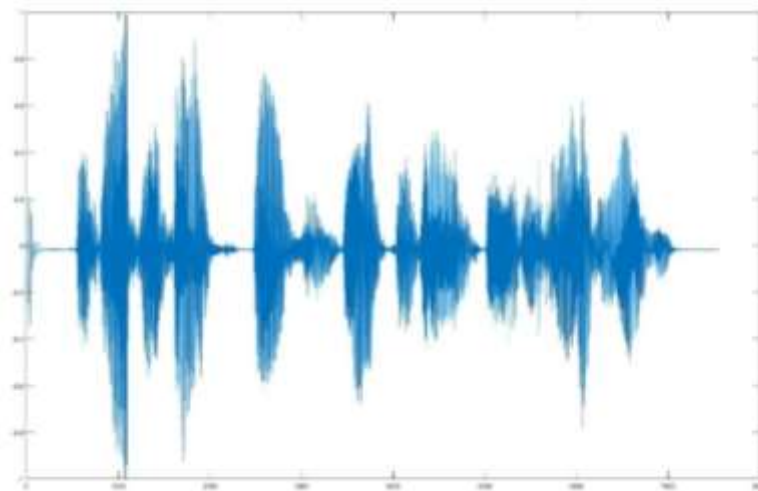
Automatically generate code
for reuse

数据类型

Image



Signal



Numeric

AgeCat	WeightQ	GroupCount	mean_BloodPressure	
Under 30	Q1	6	123.17	79.667
Under 30	Q2	3	120.33	79.667
Under 30	Q3	2	127.5	86.5
Under 30	Q4	4	122	78
30-39	Q1	12	121.75	81.75
30-39	Q2	9	119.56	82.556
30-39	Q3	9	121	83.222
30-39	Q4	11	125.55	87.273
Over 40	Q1	7	122.14	84.714
Over 40	Q2	13	123.38	79.385
Over 40	Q3	14	123.07	84.643
Over 40	Q4	10	124.6	85.1

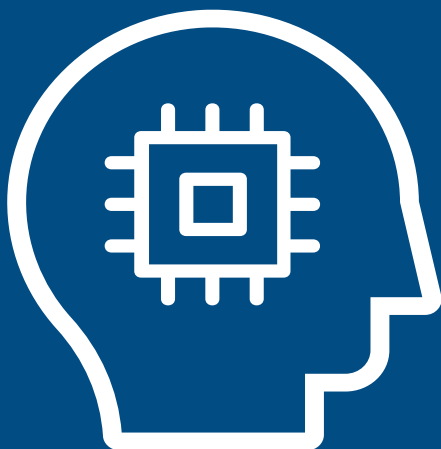
Text



机器学习是推动人工智能大趋势的关键技术

ARTIFICIAL INTELLIGENCE

Any technique that enables machines to mimic human intelligence



MACHINE LEARNING

Statistical methods that enable machines to “learn” tasks from data without explicitly programming

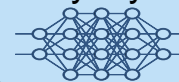
UNSUPERVISED LEARNING
(No Labeled Data)



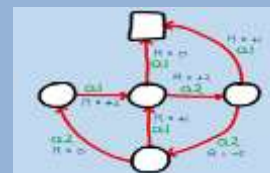
SUPERVISED LEARNING
(Labeled Data)



DEEP LEARNING
(Neural networks with many layers)



REINFORCEMENT LEARNING
(Interaction Data)

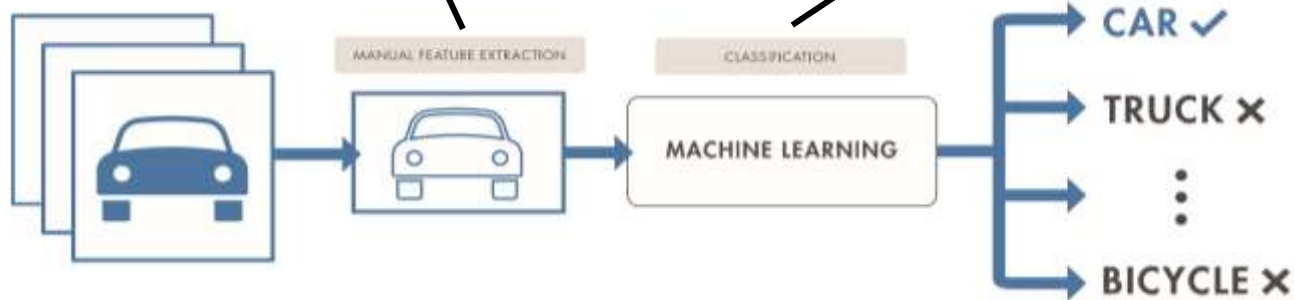


机器学习与深度学习

Decide on nature of the features ...

...and the type of model.

Machine Learning

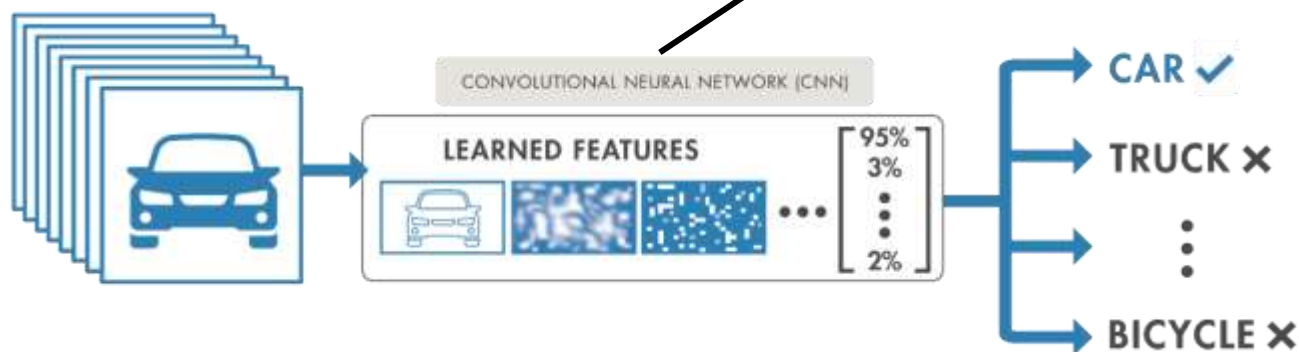


Domain knowledge required!!!

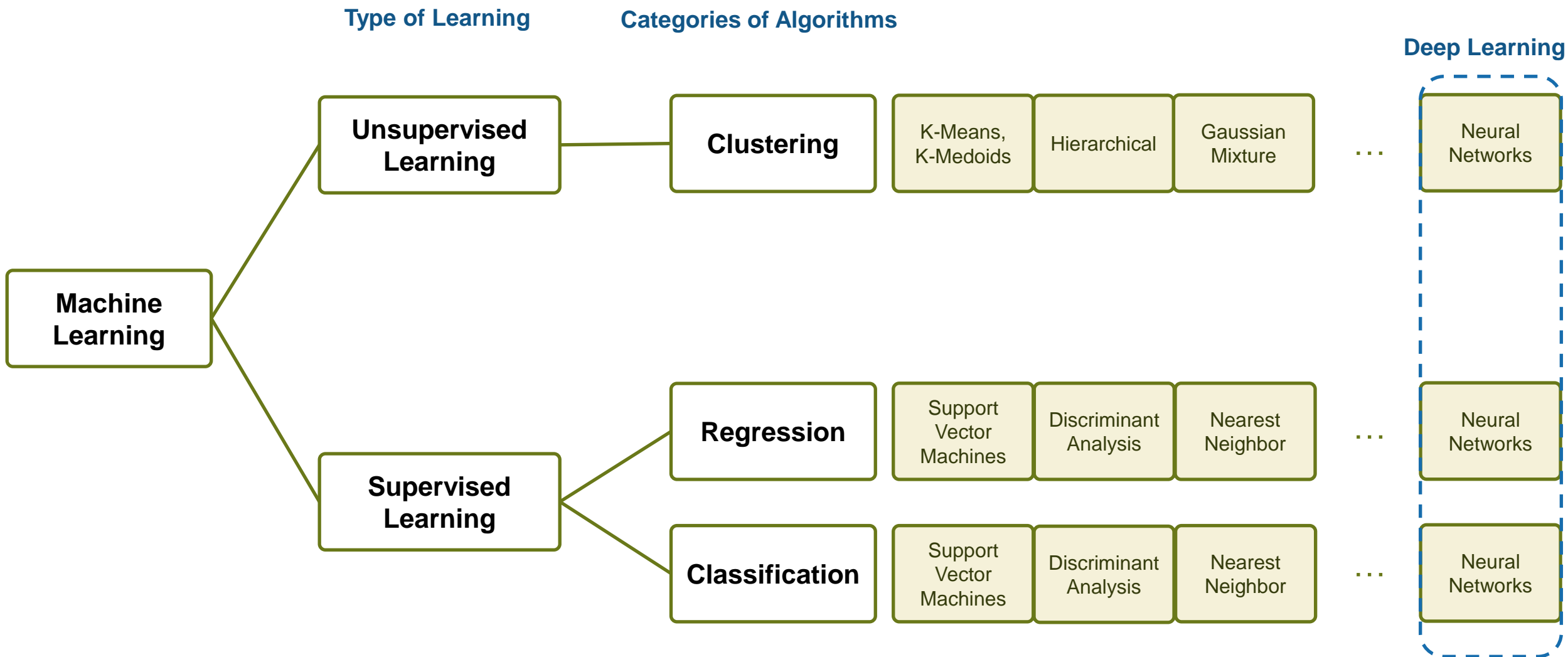
A LOT of data is required!!!

Decide on the architecture.

Deep Learning



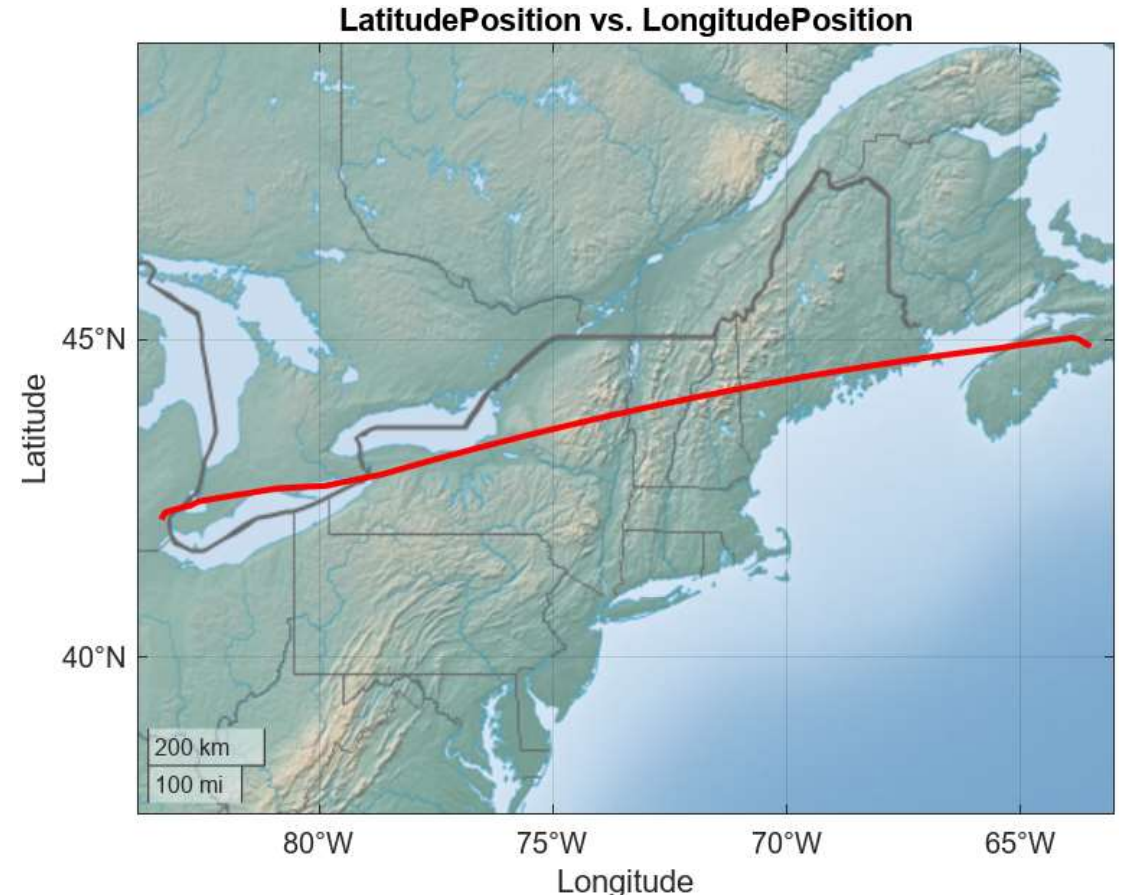
机器学习的类型



机器学习

示例：飞行传感器数据建模

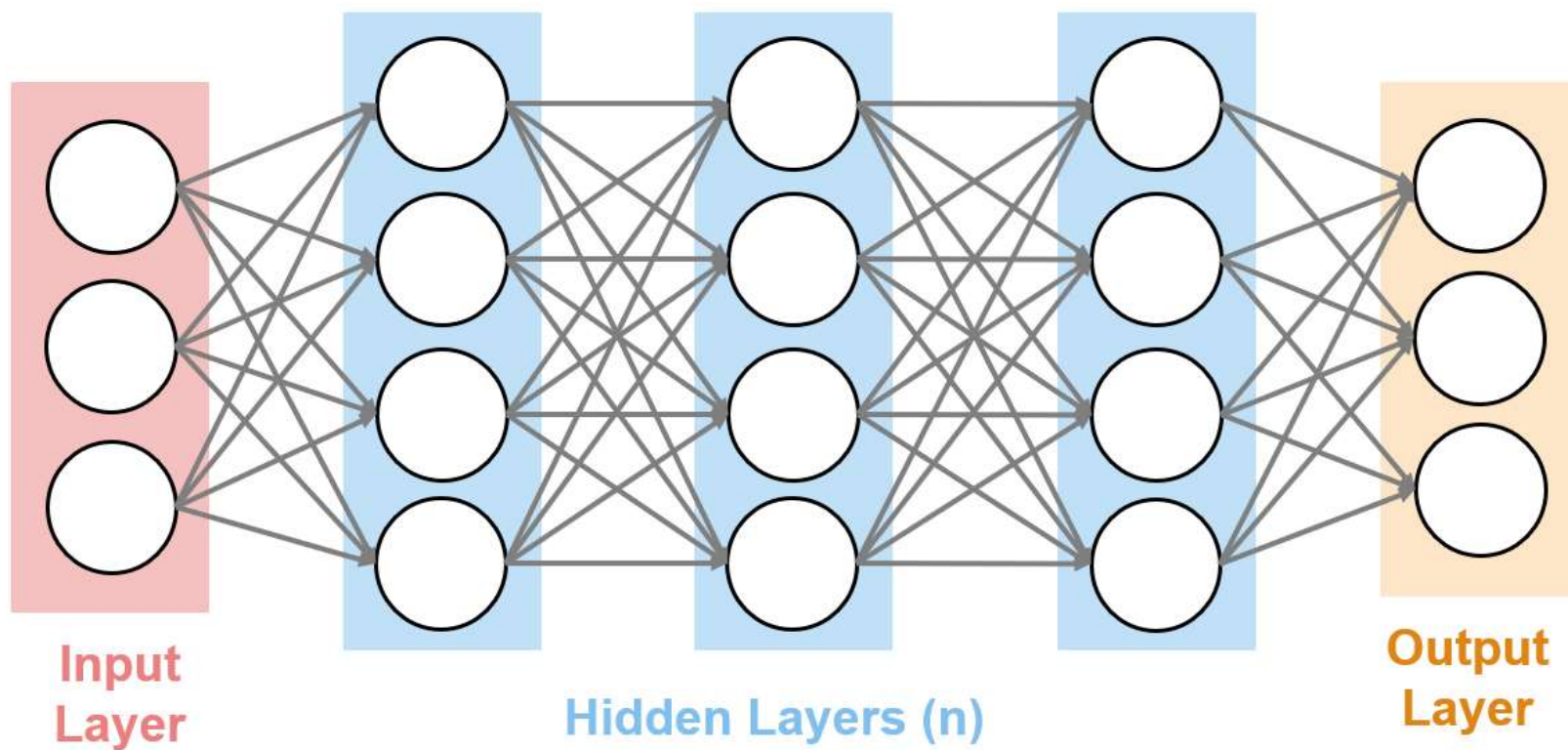
- **Objective:**
 - Create a virtual sensor model for non-observable or costly-to-observe states
- **Inputs:**
 - Excel file with 13 sensors from 1 flight
- **Approach:**
 - Visualize and explore data
 - Clean sensor anomalies
 - Train regression model to predict state
 - Share results in a report



深度学习

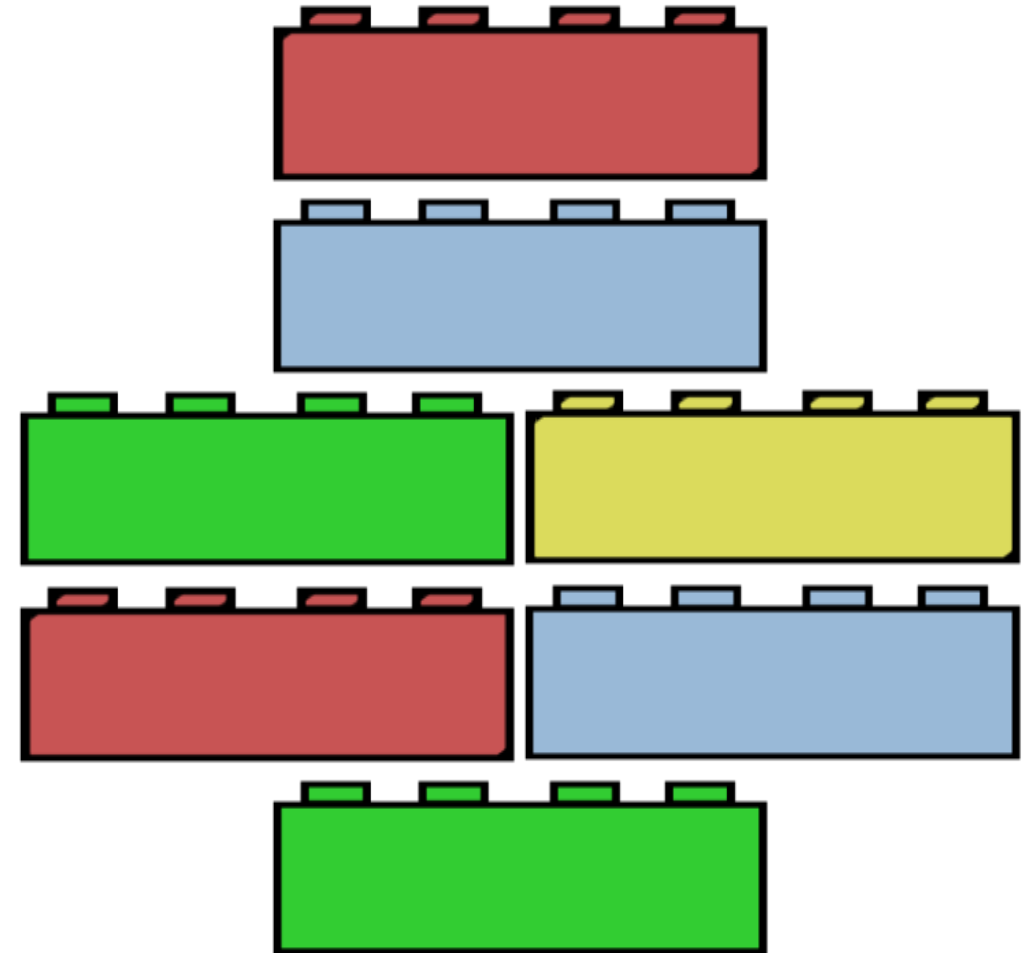
深度学习模型是神经网络

- Deep neural networks have many layers
- Data is passed through the network, and the layer parameters are updated (training)



层次

- Layers are like blocks
 - Stack on top of each other
 - Replace one block with a different one
- Each hidden layer processes the information from the previous layer
- Layers can be ordered in different ways



我们如何知道使用哪一层？

Feature Extraction - Images

- 2D and 3D convolution
- Transposed convolution (...)

Activation Functions

- ReLU
- Tanh (...)

Sequence Data

Signal, Text, Numeric

- LSTM
- BiLSTM
- Word Embedding (...)

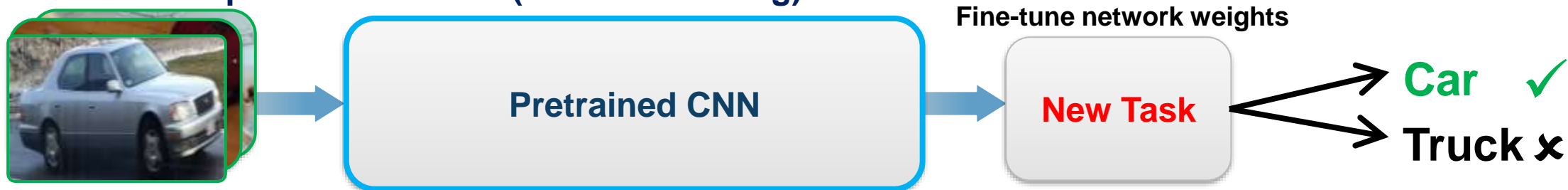
Normalization

- Dropout
- Batch normalization
- (...)

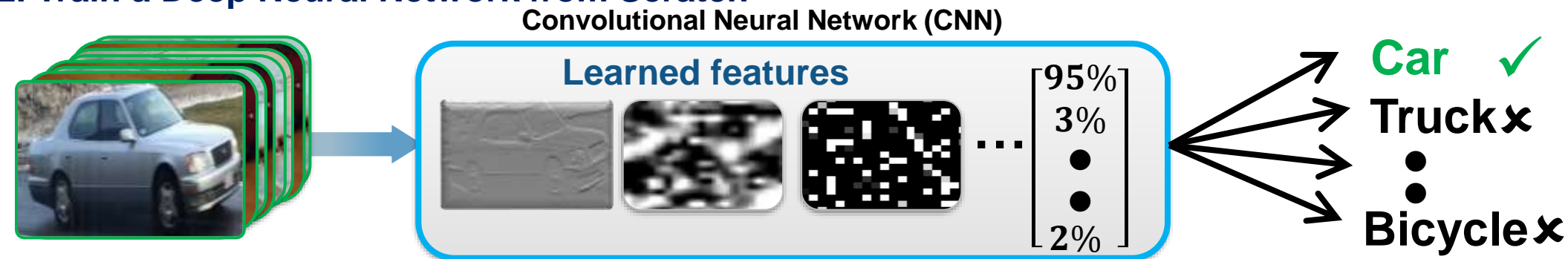
Research papers and [doc examples](#) can provide guidelines for creating architecture.

使用迁移学习 vs 从零开始搭建网络

1. Fine-tune a pretrained model (transfer learning)

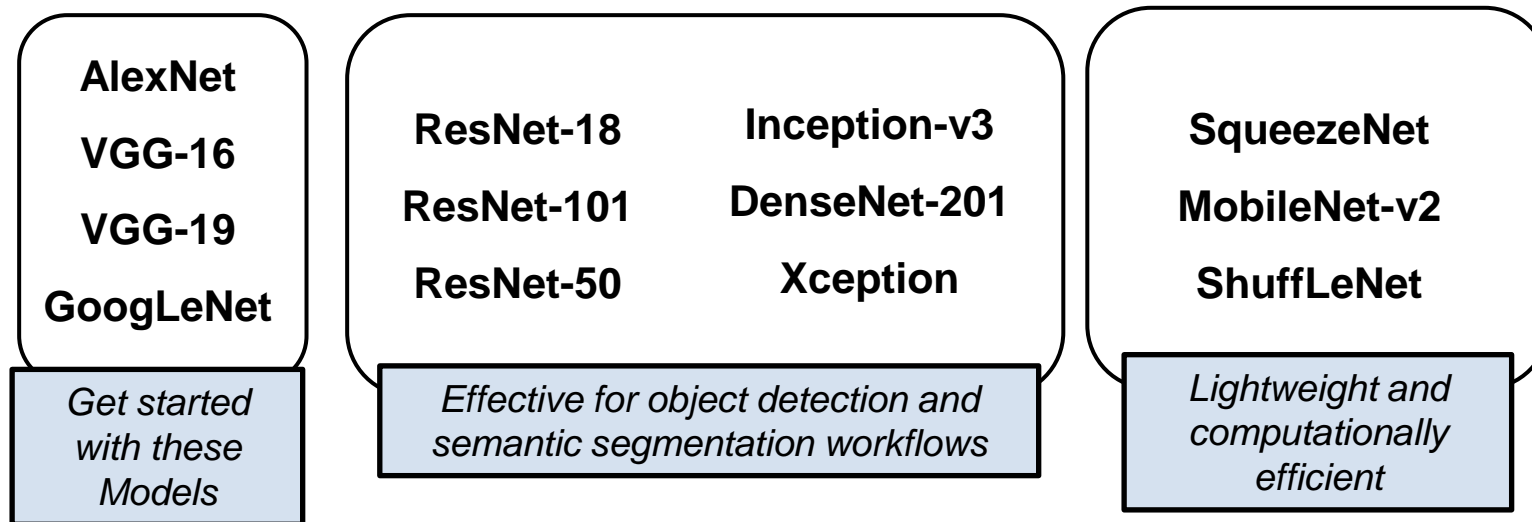


2. Train a Deep Neural Network from Scratch



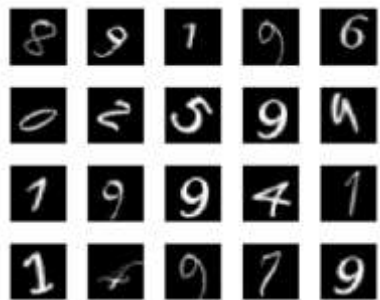
使用预训练模型构建网络

- Pretrained models have predefined layer orders and parameter values
- Can be used directly for inference (AlexNet Example)



Full list of models available [HERE](#)

使用 MATLAB 练网络



训练网络

训练参数定义

```
options = trainingOptions('sgdm', ...  
    'MaxEpochs',4, ...  
    'ValidationData',imdsValidation, ...  
    'ValidationFrequency',30, ...  
    'Verbose',false, ...  
    'Plots','training-progress');
```

```
net = trainNetwork(imdsTrain, layers, options);
```

内置网络层

```
layers = [  
    imageInputLayer([28 28 1])  
  
    convolution2dLayer(3,8,'Padding',1)  
    batchNormalizationLayer  
    reluLayer  
  
    maxPooling2dLayer(2,'Stride',2)
```

使用“深度网络设计器”修改或构建网络

The screenshot displays the 'DESIGNER' interface with three main components:

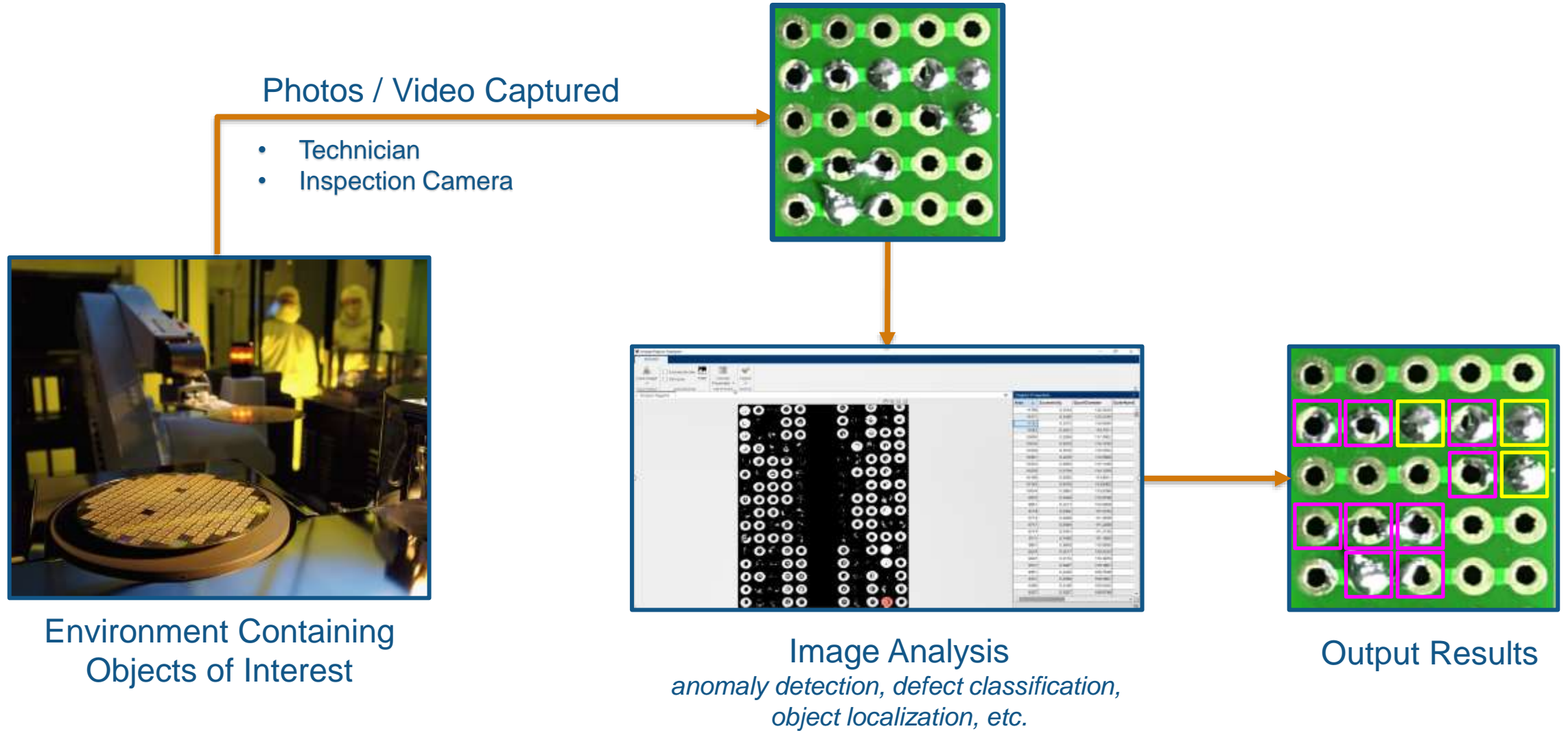
- LAYER LIBRARY:** A list of layers categorized into INPUT, CONVOLUTION AND FULLY CONNECTED, and SEQUENCE. The 'fullyConnectedLayer' is highlighted in the 'CONVOLUTION AND FULLY CONNECTED' section.
- Canvas:** A central workspace showing a neural network architecture. The layers are connected in a sequence: data imageInputLayer → conv1 convolution2d... → relu_conv1 reluLayer → pool1 maxPooling2d... → ftr2-squeeze convolution2d... → ftr2-relu_squ reluLayer. The final layer branches into ftr2-expand1x1 convolution2d... and ftr2-expand3x3 convolution2d....
- PROPERTIES:** A panel on the right showing the configuration for the selected 'fullyConnectedLayer'. Parameters include Name (fc), InputSize (auto), OutputSize (10), Weights, Bias, WeightLearnRateFactor (1), WeightL2Factor (1), BiasLearnRateFactor (1), BiasL2Factor (0), WeightsInitializer (glorot), and BiasInitializer (zeros).

Select layers from layer library

Drag and connect layers on canvas

Edit parameters of individual layers

计算机视觉 workflow



示例 1:通过迁移学习进行图像分类

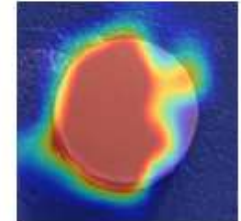
- Use transfer learning to visually inspect pills and classify them based on the type of defect
- You will learn how to:
 - Modify an existing deep learning network
 - Train a deep learning network to classify images
 - Analyze the behavior of the deep learning network



Correct normal classification



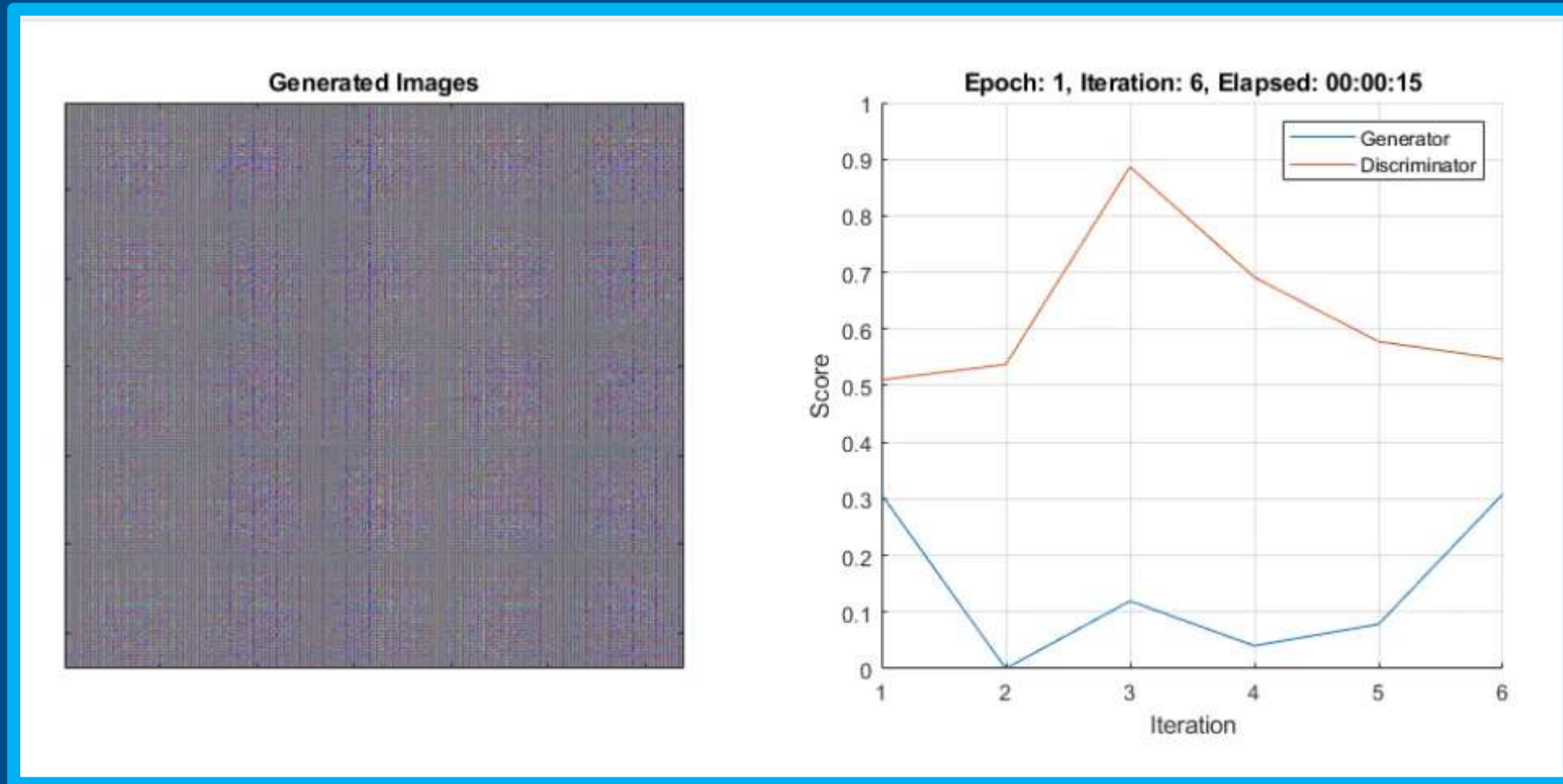
GradCam Analysis



LIME Analysis



示例2：训练生成对抗网络 (Generative Adversarial Networks, GANs)



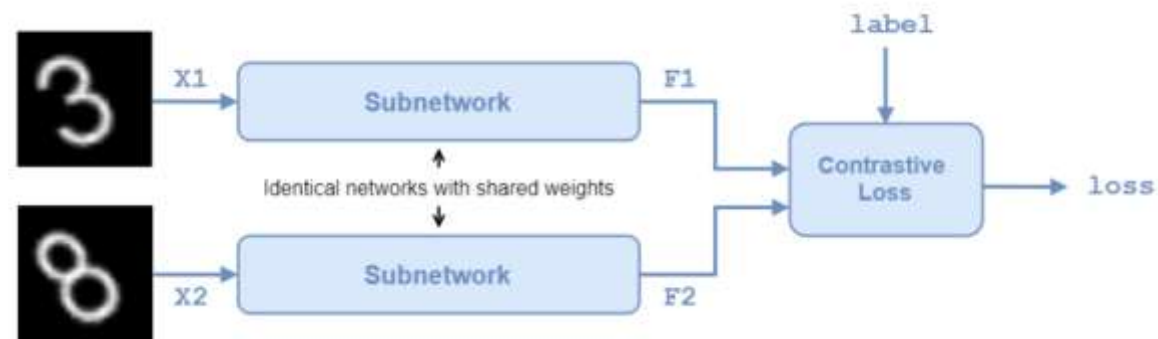
高级的神经网络

生成对抗网络 (GAN)

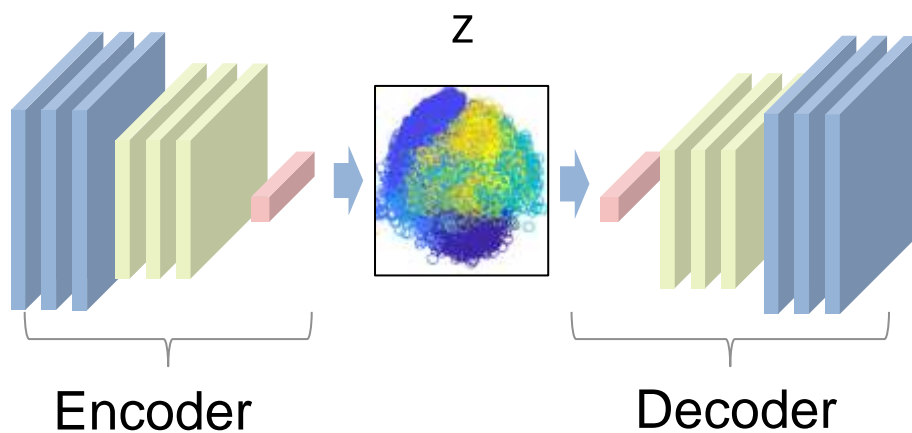


孪生网络

(Siamese networks)

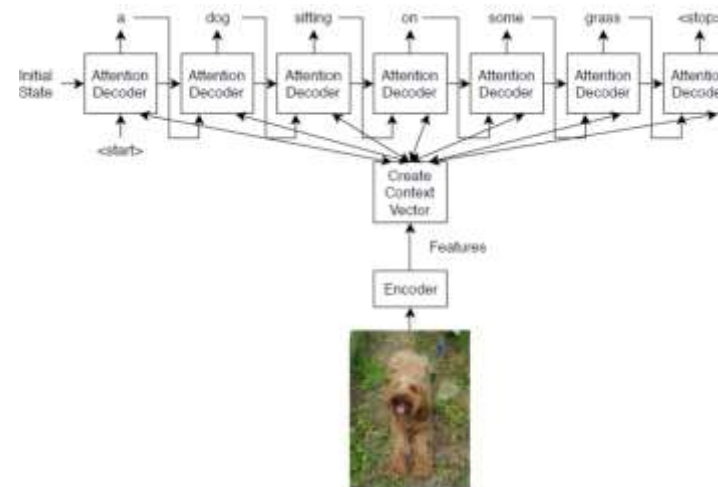


变分自编码器 (VAE)



注意力机制

(attention mechanism)



扩展的深度学习框架增加了以下内容:

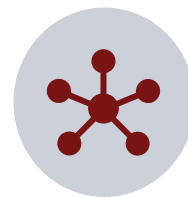
无监督式深度学习（生成模型）

多输入多输出（MIMO）神经网络

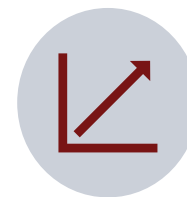
神经网络中的任意函数编程

完全定制的神经网络训练循环

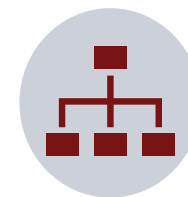
关键技术



共享的权重

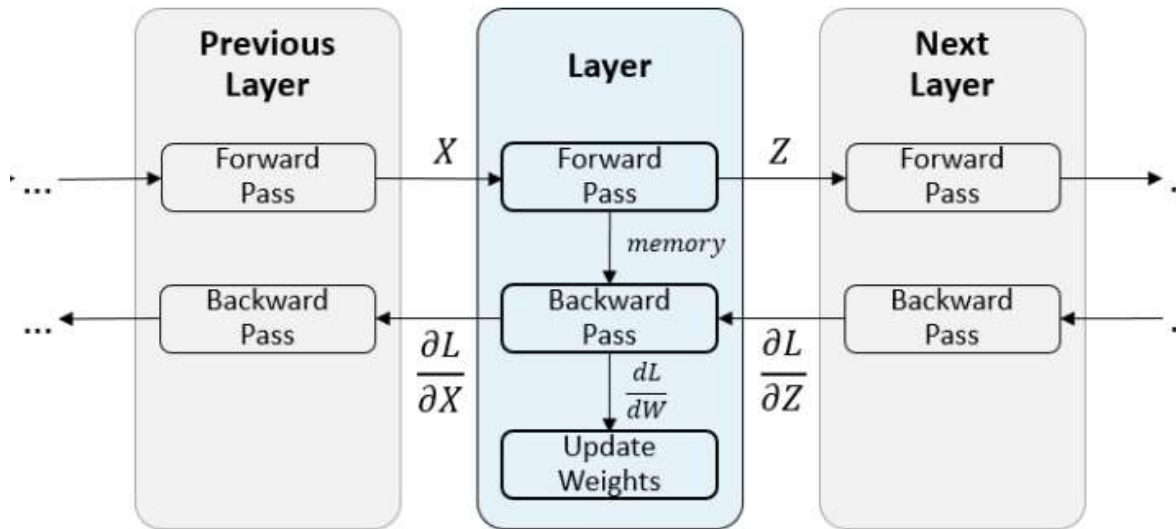


自动微分



灵活的训练
结构

定义自定义深度学习层



[Link: 定义自定义深度学习层模板](#)

```
classdef myLayer < nnet.layer.Layer % & nnet.layer.Formatable (Optional)
```

```
properties (Learnable)
```

```
% (Optional) Layer learnable parameters.
```

```
% Layer learnable parameters go here.
```

```
end
```

```
methods
```

```
function layer = myLayer()
```

```
% (Optional) Create a myLayer.
```

```
end
```

```
function [Z1, ..., Zm] = predict(layer, X1, ..., Xn)
```

```
% Forward input data through the layer at prediction time and  
% output the result.
```

```
end
```

```
function [Z1, ..., Zm, memory] = forward(layer, X1, ..., Xn)
```

```
% (Optional) Forward input data through the layer at training  
% time and output the result and a memory value.
```

```
end
```

```
function [dLdX1, ..., dLdXn, dLdW1, ..., dLdWk] = ...
```

```
backward(layer, X1, ..., Xn, Z1, ..., Zm, dLdZ1, ..., dLdZm, memory)
```

```
% (Optional) Backward propagate the derivative of the loss  
% function through the layer.
```

```
end
```

```
end
```

```
end
```

生成对抗网络 (GANs)

很多种 GAN

输入

生成器



随机向量

许多类型和不同数量的生成器

训练集



可以是图片、视频或者声音

真
 假



判别器

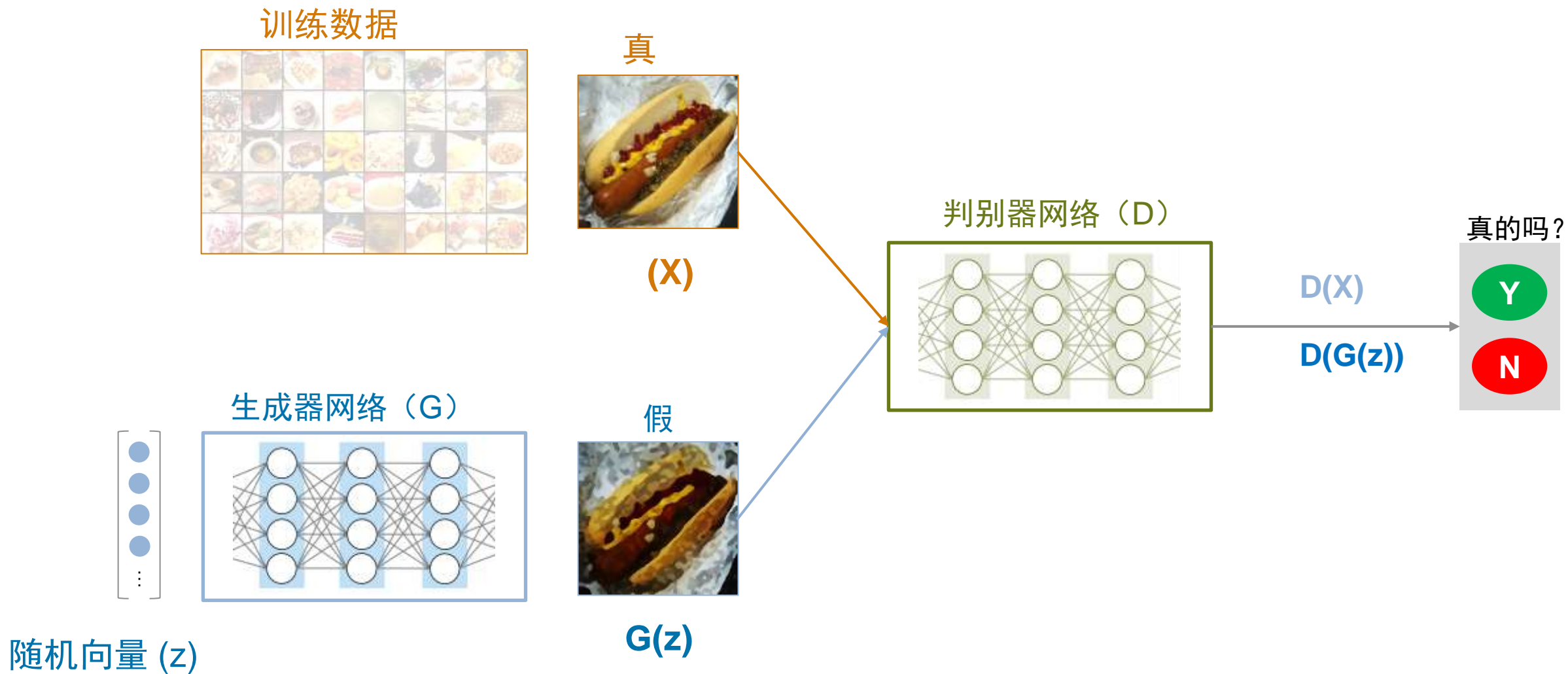
许多类型和不同数量的判别器



生成的图片

不同的分辨率

GAN 的可视化



GAN 具有竞争性的损失函数

GAN 输出 —— 介于 0 和 1 之间的值

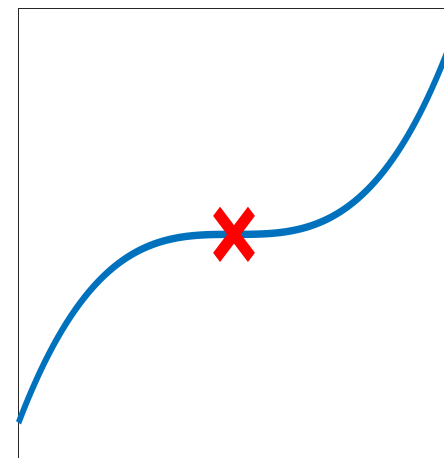
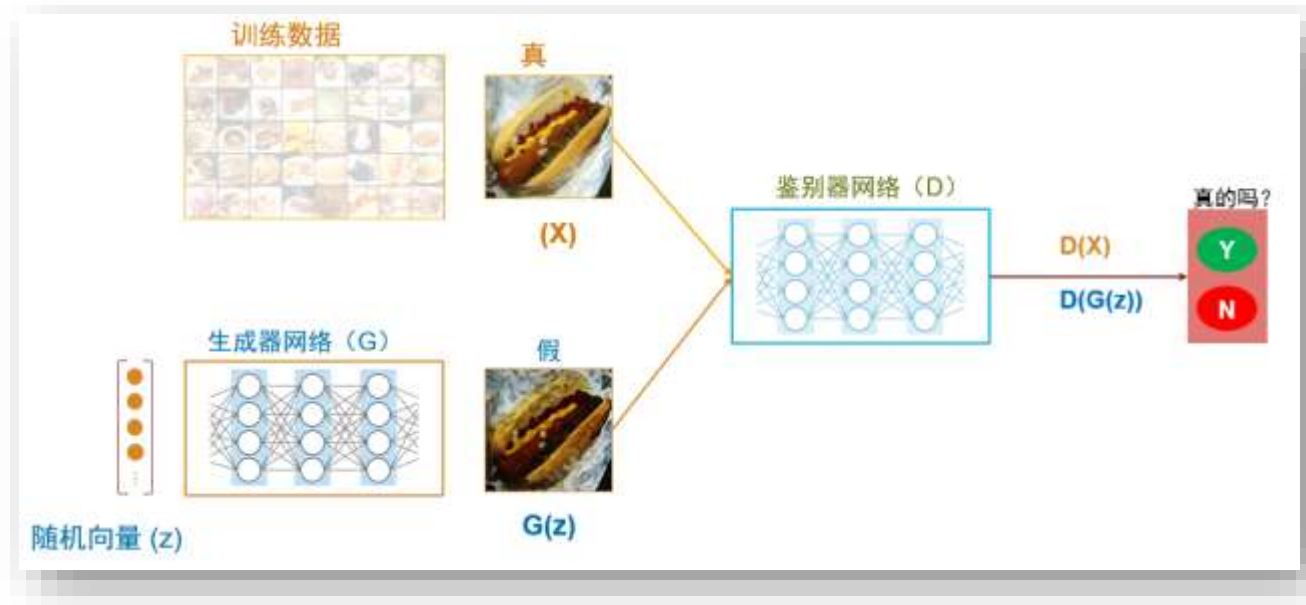
- $D(X)$ - 真 $\rightarrow 1$
- $D(G(z))$ - 假 $\rightarrow 0$

生成器的损失函数

- $(-1) * \text{mean}(\log(D(G(z))))$
- 当 $D(G(z))$ 接近 1 时, 函数最小化

判别器的损失函数

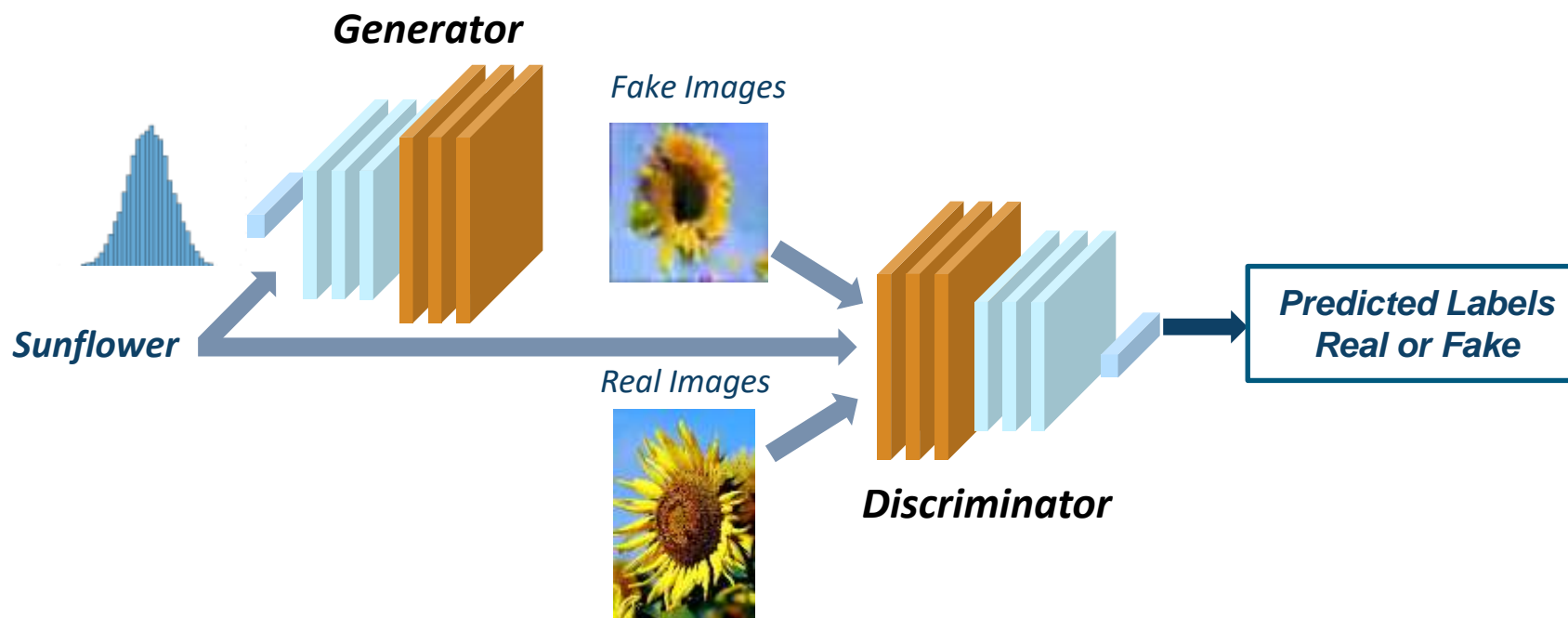
- $(-1) * \text{mean}(\log(D(X)))$ + $(-1) * \text{mean}(\log(1 - D(G(z))))$
- 判断真的图片为真 判断假的图片为假



示例：扩展的 GANs

■ C-GAN

- 条件生成对抗网络是GAN的一种类型，它在训练过程中也利用了标签



Generate Daisy flower!



[Full code available](#)

FILE NAVIGATE TEXT CODE SECTION RUN

NEW OPEN SAVE PRINT EXPORT GO TO FIND BOOKMARK

Normal B I U M

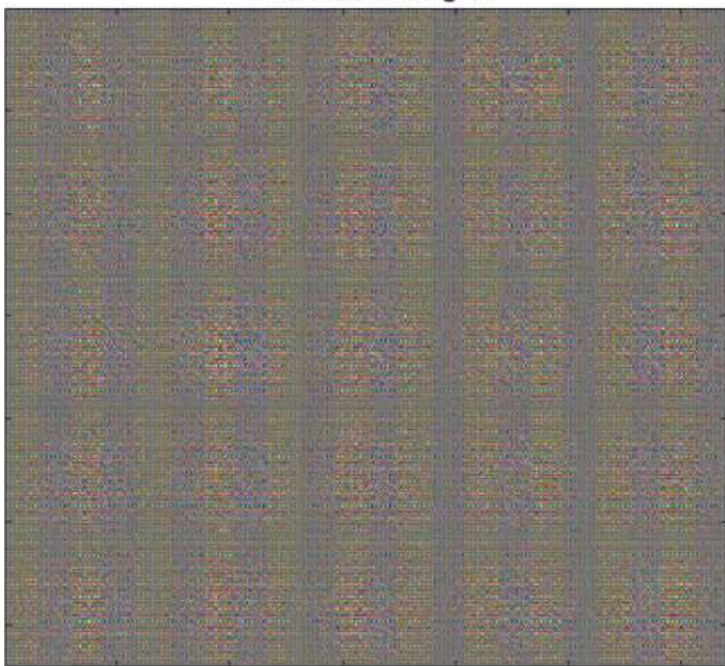
Code Control Task Refactor

Run Section Run and Advance Run to End

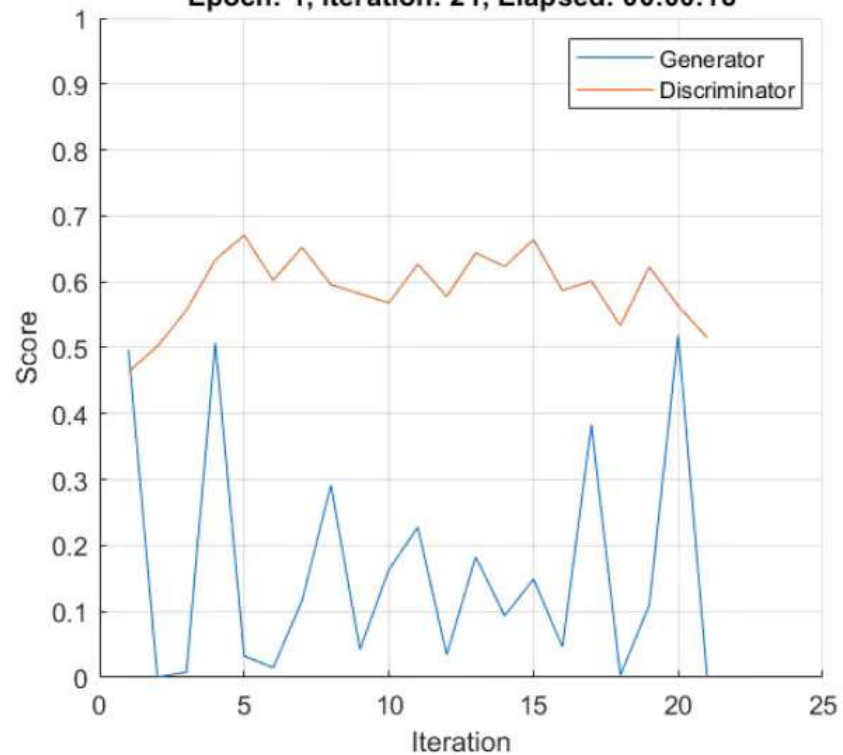
Pause Step Stop

CGAN_Example.mlx * +

Generated Images



Epoch: 1, Iteration: 21, Elapsed: 00:00:18



此时，判别器已学会在生成的图像中识别真实图像的强特征表示。顺带，生成器已学会类似的强特征表示，能够生成类似于训练数据的一类图像。

训练图显示生成器和判别器网络的分数。要了解有关如何解释网络分数的详细信息，请参阅[Monitor GAN Training Progress and Identify Common Failure Modes](#)。

生成新图像

示例 3：心电图信号分类：房颤心律（Afib）检查

[完整代码](#)

▪ 动机

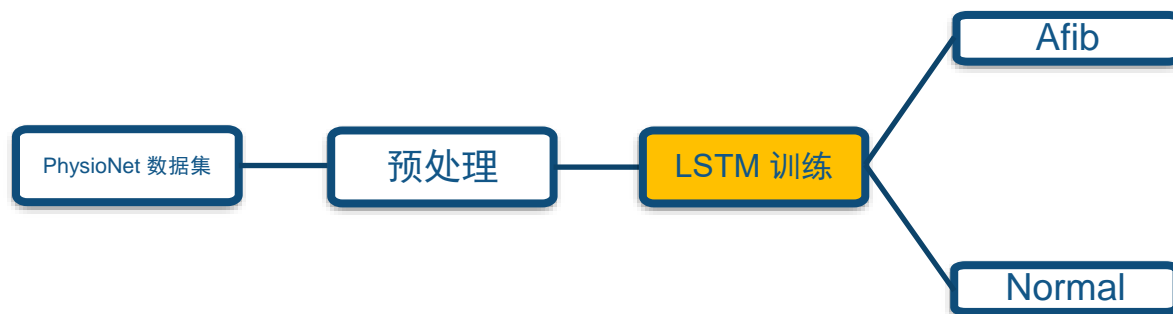
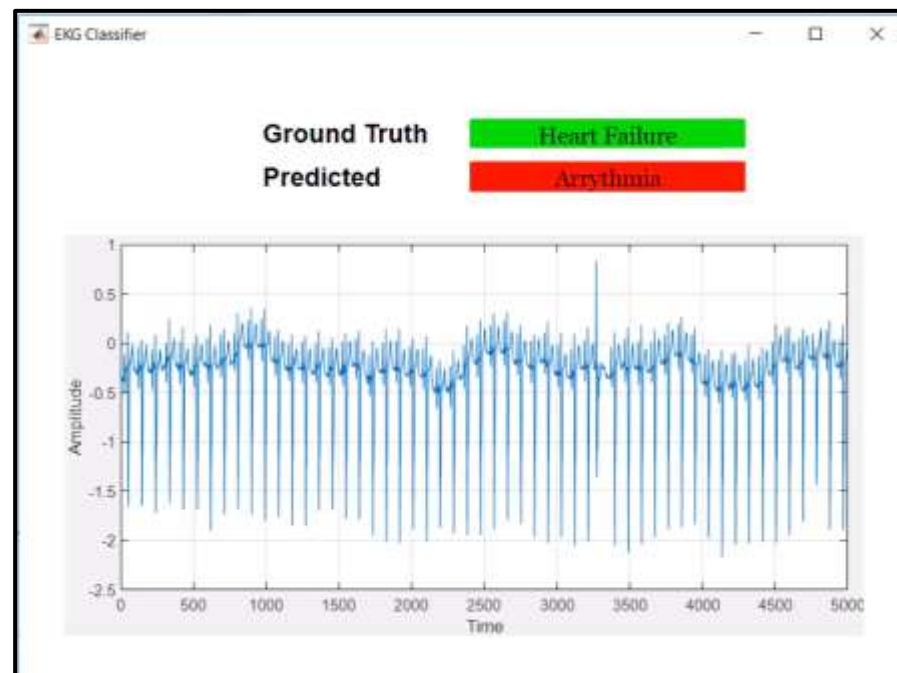
- 房颤（Afib）是一种不规则的心跳
- 创建并训练一个 LSTM 二分类网络

▪ 目的

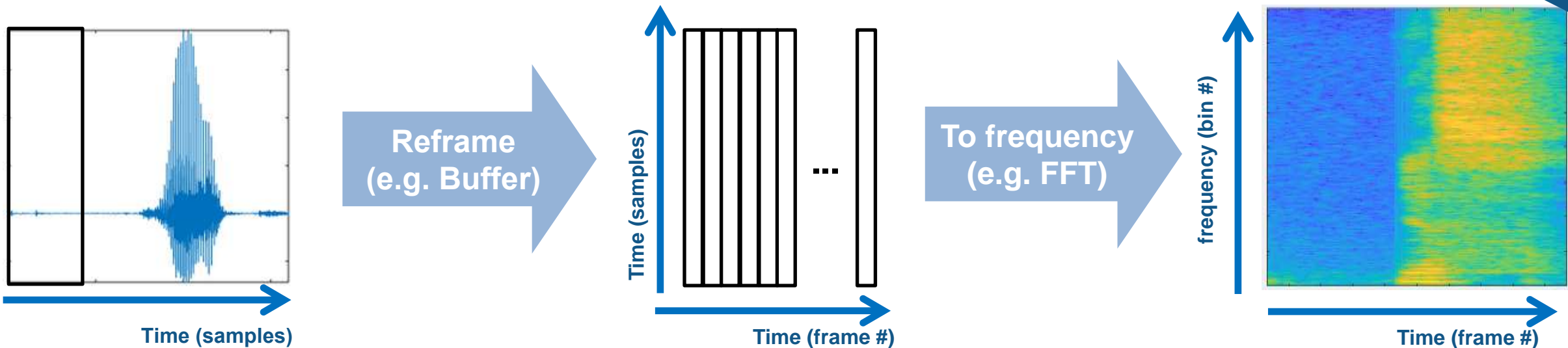
- 开发心电图的分类器

▪ 数据

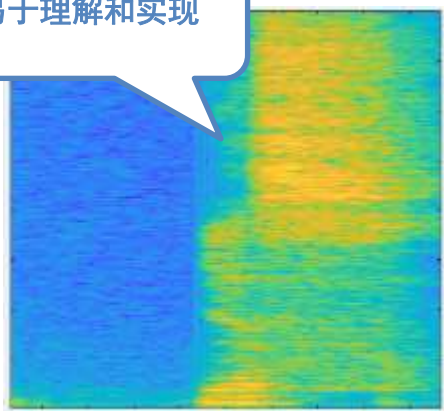
- 来源：[PhysioNet](#)
- 下载地址：
<https://archive.physionet.org/challenge/2017/training2017.zip>
- 训练数据: 8876 个信号
- 验证数据: 980 个信号
- 每个信号: 记录长短不一，类别不平衡
- 标签: 房颤心律（Afib）和正常心律（Normal）



时频转换

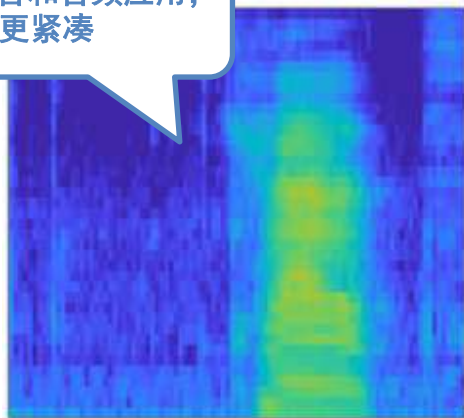


易于理解和实现



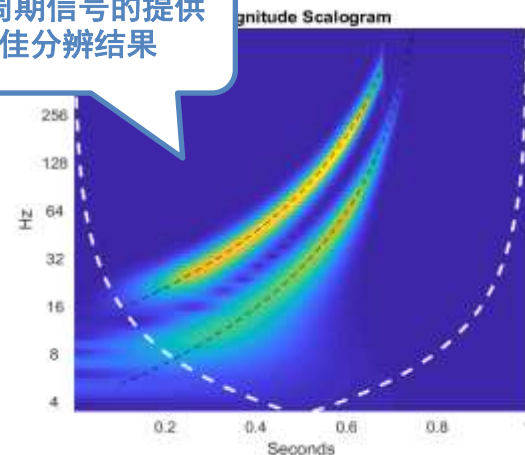
时频图

适合语音和音频应用，
更紧凑



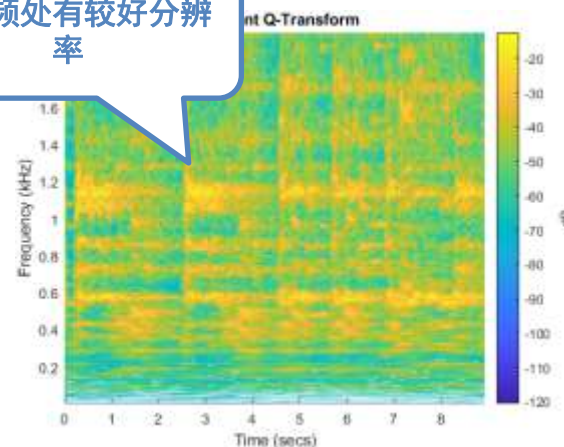
Perceptually-spaced (e.g. Mel, Bark) 时频图

为非周期信号的提供
最佳分辨结果



小波时频尺度图

在低频处有较好分辨率



常数Q变换

示例 4：基于深度学习的信号调制识别

目标：开发分类器并部署为 App

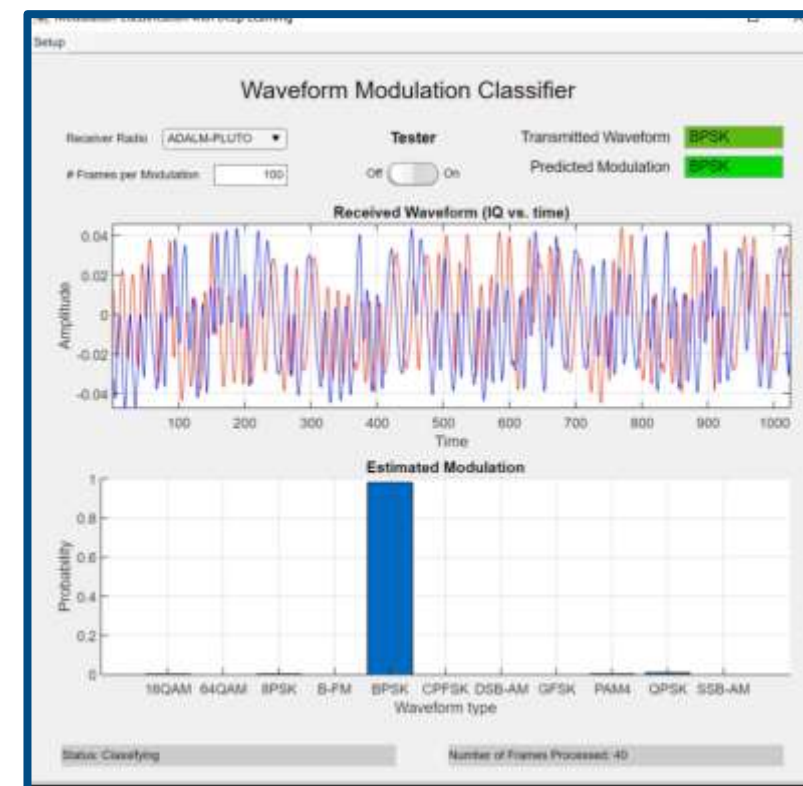


训练数据:

- 由 MATLAB 合成的11类传输损伤调制信号
- 训练、测试与验证数据：88000 v.s 11000 v.s 11000 帧
- 标签: BPSK, QPSK, 8PSK, 16QAM, 64QAM, PAM4, GFSK, CPFSK, B-FM, DSB-AM, SSB-AM
- 每帧 1024 个采样

真实信号分类:

- 使用 Adalm Pluto SDR 硬件发送和采集信号



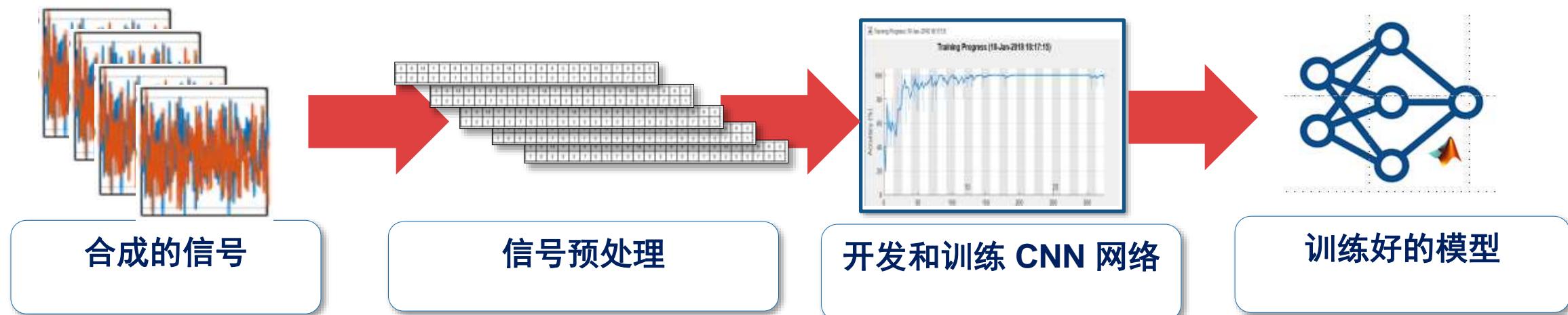


Predict Modulation Type Using CNN

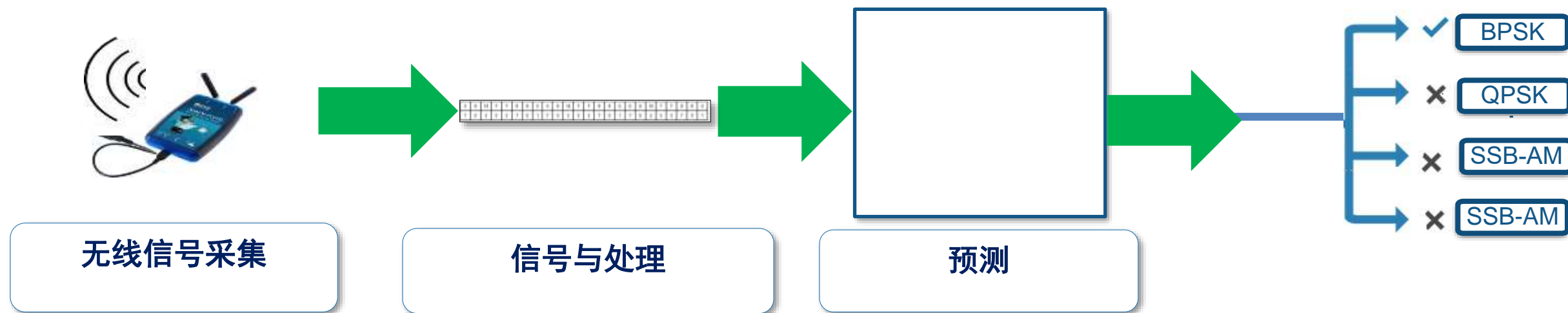
The results of the CNN model for predicting modulation types are as follows:

- Binary PSK (BPSK) - 20 dB
- Quadrature PSK (QPSK) - 20 dB
- Frequency PSK (FSK) - 20 dB
- 16-QAM (Quadrature Amplitude Modulation) - 20 dB
- 64-QAM (Quadrature Amplitude Modulation) - 20 dB
- 256-QAM (Quadrature Amplitude Modulation) - 20 dB
- 16-PSK (Polar PSK) - 20 dB
- 8-PSK (Polar PSK) - 20 dB
- Single-carrier amplitude modulation - 20 dB
- Single-carrier amplitude modulation - 20 dB

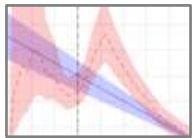
深度神经网络训练



信号识别



为什么使用MATLAB进行深度学习？



Predictive Maintenance

- [Bearing Prognosis](#)
- [Pump Fault Diagnosis](#)

Predictive Maintenance
Toolbox™



Land-Use Classification

- [Semantic Segmentation for Multispectral Images](#)

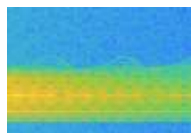
Image Processing
Toolbox™



Lidar

- [Lidar Point Cloud Semantic Segmentation](#)
- [3-D Object Detection Using PointPillars](#)

Lidar
Toolbox™



Radar

- [Radar Waveform Classification](#)
- [Pedestrian and Bicyclist Classification](#)

Phased Array
System Toolbox™



Wireless Communications

- [Modulation Classification](#)
- [Detect WLAN Router Impersonation](#)

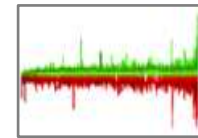
Communications
Toolbox™



Reinforcement Learning

- [Train Biped Robot to Walk](#)
- [PMSM Motor Control](#)

Reinforcement
Learning Toolbox™



Computational Finance

- [Machine Learning for Statistical Arbitrage](#)

Financial
Toolbox™



Robotics

- [Avoid Obstacles using Reinforcement Learning](#)

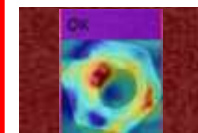
Robotics System
Toolbox™



Automated Driving

- [Deep Learning Vehicle Detector](#)
- [Occupancy Grid with Semantic Segmentation](#)

Automated
Driving Toolbox™



Visual Inspection

- [Manufacturing Defect Detection](#)
- [Anomaly Detection for Cloth Manufacturing](#)

Image Processing
Toolbox™



Audio

- [Speech Command Recognition](#)
- [Cocktail Party Source Separation](#)

Audio
Toolbox™



Medical Imaging

- [3-D Brain Tumor Segmentation](#)
- [Breast Cancer Tumor Classification](#)

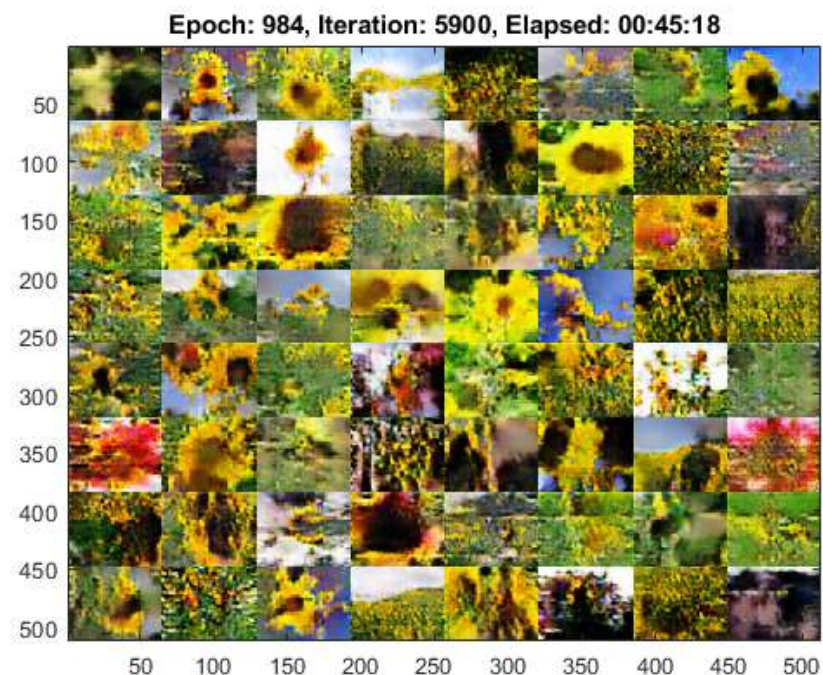
Image Processing
Toolbox™

Low-Code AI

开发你自己的自定义网络

创建你自己的自定义网络，或者重建最新研究发布的网络

Custom training loops, Custom loss functions	R2019b
Automatic Differentiation, Shared Weights	
3-D layers, MIMO	R2020a
Code generation support	R2020b
Higher Order Derivatives, 1-D Convolution	R2021a
Short-time Fourier transform layer <i>Signal Processing Toolbox</i>	R2021b
GANs, Siamese Networks	R2019b
CGANs	R2020a
Graph Convolutional Networks	R2021a



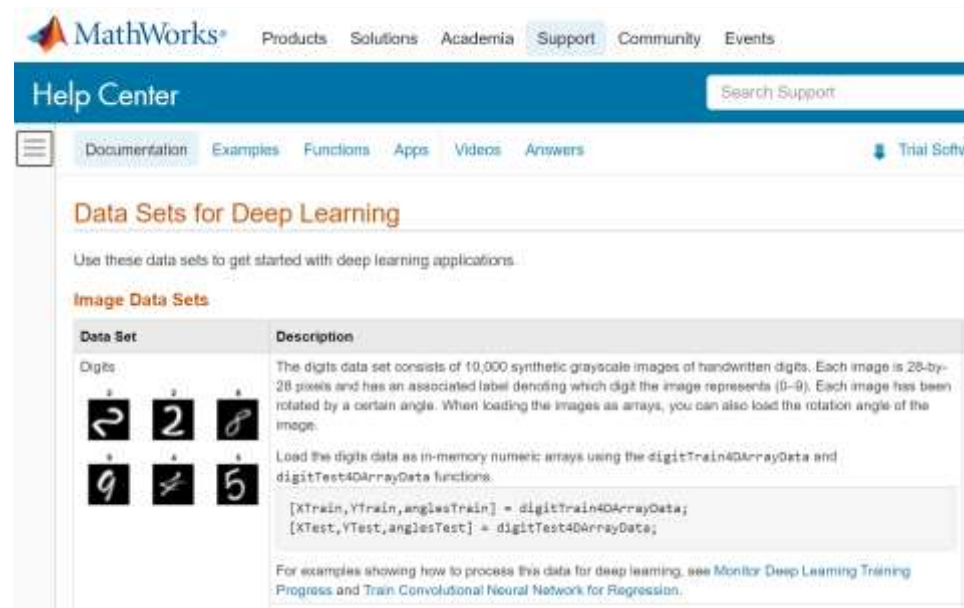
为 AI 找到有用的数据集

为训练模型节省搜索适当数据集的时间:

- 图片
- 视频
- 信号
- 时间序列
- 文本
- 音频
- 表格数据

[Data Sets for Deep Learning](#) 帮助文档

[Data Sets for Machine Learning and Statistics](#) 帮助文档



MathWorks® Products Solutions Academia Support Community Events

Help Center Search Support

Documentation Examples Functions Apps Videos Answers Trial Softw

Data Sets for Deep Learning

Use these data sets to get started with deep learning applications.

Image Data Sets

Data Set	Description
Digits	The digits data set consists of 10,000 synthetic grayscale images of handwritten digits. Each image is 28-by-28 pixels and has an associated label denoting which digit the image represents (0-9). Each image has been rotated by a certain angle. When loading the images as arrays, you can also load the rotation angle of the image.

Load the digits data as in-memory numeric arrays using the `digitTrain4DArrayData` and `digitTest4DArrayData` functions.

```
[XTrain,YTrain,anglesTrain] = digitTrain4DArrayData;  
[XTest,YTest,anglesTest] = digitTest4DArrayData;
```

For examples showing how to process this data for deep learning, see [Monitor Deep Learning Training Progress](#) and [Train Convolutional Neural Network for Regression](#).

从领域特定的例子开始

例如，工程师和科学家如何学习应用人工智能构建 [深度学习应用](#) 和 [机器学习应用](#)

- 计算机视觉(图像与视频)
- 信号处理
- 文本
- 音频
- 控制(强化学习)
- 通信系统设计(Simulink)
- 无线生物技术与医药
- 金融
- 能源
- 生产制造业
- **超过 500+ 完整示例!**

Statistics and Machine Learning Applications

Apply statistics and machine learning methods to industry-specific workflows

Statistics and Machine Learning Toolbox™ provides tools to describe, analyze, and model data. Apply these tools, in combination with other MATLAB® toolboxes, to perform industry-specific workflows. Some of the application areas include:

- **Aerospace** – Explore radar and other signals, detect anomalies, and build predictive models.
- **Biotechnology and Pharmaceutical** – Analyze clinical data, and perform modeling and simulation for drug discovery and development.
- **Communications and Signal I**
- **Energy Production** – Forecast
- **Industrial Automation and M** manufacturing processes and
- **Medical Devices** – Build inter applications while complying
- **Quantitative Finance and Risk** risk, and fraud detection.

Deep Learning Applications

Extend deep learning workflows with computer vision, image processing, automated driving, signals, audio, text analytics, and computational finance

Use Deep Learning Toolbox™ to incorporate deep learning in computer vision, image processing, automated driving, signal processing, audio, text analytics, and computational finance applications.

Deep Learning with Simulink

Extend deep learning workflows using Simulink

Computer Vision Using Deep Learning

Extend deep learning workflows with computer vision applications

Image Processing Using Deep Learning

Extend deep learning workflows with image processing applications

Automated Driving Using Deep Learning

Extend deep learning workflows with automated driving applications

Lidar Processing Using Deep Learning

Extend deep learning workflows for Lidar point cloud processing

Signal Processing Using Deep Learning

Extend deep learning workflows with signal processing applications

Audio Processing Using Deep Learning

Extend deep learning workflows with audio and speech processing applications

Wireless Communications Using Deep Learning

Extend deep learning workflows with wireless communications system applications

Reinforcement Learning Using Deep Neural Networks

Train deep neural network agents by interacting with an unknown dynamic environment

Text Analytics Using Deep Learning

Extend deep learning workflows with text analytics applications

Computational Finance Using Deep Learning

Extend deep learning workflows with computational finance applications

Aerospace

Radar Target Classification Using

Classify radar returns using macl

Biotechnology and Pharr

High-Throughput Sequencing

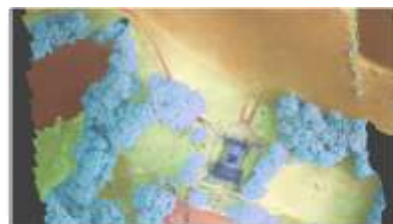
Gene Expression Profile Analysis

This example shows a number of

从领域特定的例子开始



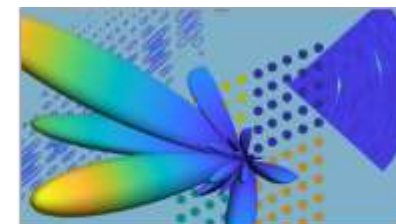
Predictive Maintenance
Anomaly Detection and Condition Monitoring



Geospatial Analysis
Hyperspectral Image Classification



Lidar
3-D Point Cloud Object Detection



Radar
Waveform Classification

AI Modeling



Model design and tuning



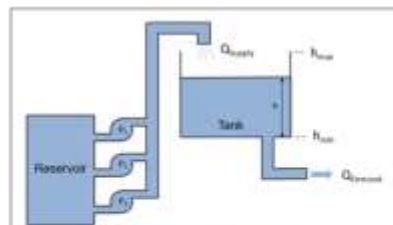
Hardware accelerated training



Interoperability



Wireless Comms
Data Synthesis for 5G Channel Estimation



Controls Systems
PID Tuning & System Scheduling



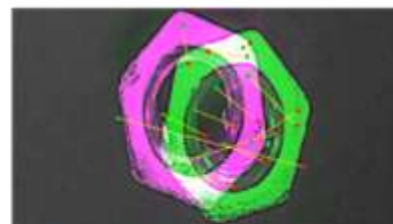
Computational Finance
Trading & Risk Management



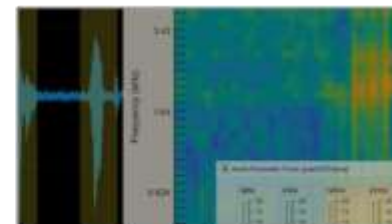
Automated Driving
Pedestrian & Vehicle Detection



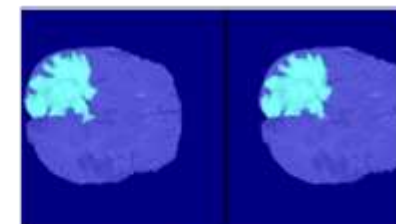
Robotics
Path Planning & Process Optimization



Visual Inspection
Defect Detection



Audio
Speech Recognition



Medical Imaging
Tumor Detection

在GitHub上找到最新的预训练模型

MATLAB 深度学习模型中心

R2022a

- 快速参考超过 50 个预训练模型：
 - 计算机视觉
 - 自然语言处理
 - 语音
 - 激光雷达
 - * 每月增加新模型

Object Detection

Object Detection is a computer vision technique used for locating instances of objects in images or videos. When humans look at images or videos, we can recognize and locate objects of interest within a matter of moments. The goal of object detection is to replicate that intelligence using a computer.

Inputs are RGB images. The output is the predicted class, bounding box and score.

Input: RGB image

Pretrained Model

Output: Classification Label, Bounding Box & Score

These networks have been trained to detect 80 objects classes from the COCO dataset. These models are suitable for training a custom object detector using transfer learning.

Network	Backbone Network	Size (MB)	Mean Average Precision (mAP)	Object Classes	Location
FRCNN	efficientnet	75.8	33.7	80	GitHub
YOLO-v4	yolov4-tiny	238	44.8	80	GitHub
YOLO-v4	yolov4-tiny-coco	21.3	39.7	80	GitHub
YOLO-v3	darknet53-coco	228.6	38.8	80	GitHub
YOLO-v3	darknet53-coco	21.3	35.1	80	GitHub
YOLO-v3	darknet53-conv	188	35.4	80	GitHub
YOLO-v3	darknet53-conv	191	39.7	80	GitHub
YOLO-v3	darknet53-conv	47	30.6	80	GitHub

Tip for selecting a model:

Pretrained object detectors have different characteristics that matter when choosing a network to apply to your problem. The most important characteristics are mean average precision (mAP), speed, and size. Choosing a network

<https://github.com/matlab-deep-learning/MATLAB-Deep-Learning-Model-Hub>

在 MATLAB 中免费开始使用 AI

无需下载，无需安装，只需您的浏览器和您...



机器学习入门之旅

初步了解面向分类问题的机器学习实用方法。

详细信息与启动



深度学习入门之旅

快速入门，运用深度学习方法执行图像识别。

详细信息与启动



强化学习入门之旅

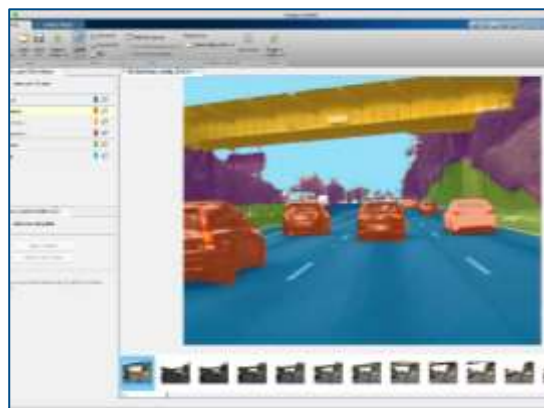
交互式入门教程，介绍了面向控制问题的强化学习方法。

详细信息与启动

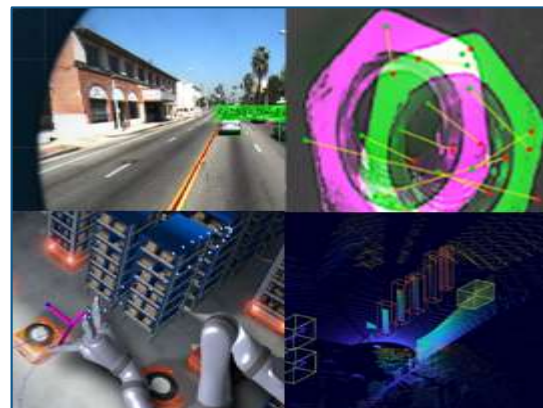
<https://matlabacademy.mathworks.com/cn>



支持领域专家做最好的工作



建立更好的数据集



利用专业领域示例



在系统内模拟 AI

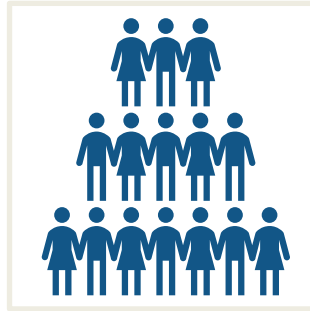


部署到任何设备

Key takeaways



Ease-of-use through
interactive tools



AI accessible to everyone



Interactive tools can
enhance programming skills

Thank you



© 2022 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://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.