Agenda

- Introduction
  - Applications
  - Computer vision tasks
  - Choosing an approach

- Examples
  - ‘Traditional’ image processing
  - Deep learning

- Getting started
Computer Vision Tasks

- Automated Driving
- Manufacturing
- Medical Imaging
Computer Vision Tasks

- Where are the cars?
- Where can I drive?
- How many parts?
- Are they damaged?
- Is this a tumour?
- How large is it?
Computer Vision Tasks

- Image classification
- Object detection
- Semantic segmentation
- Instance segmentation

Adapted from arXiv:1704.06857
Two approaches to computer vision

‘Traditional’ Image Processing
Two approaches to computer vision

Machine Learning

Images

Output

Model
Machine Learning v Deep Learning

**Machine Learning** learns tasks using manually extracted features.

Deep Learning learns both features and tasks directly from data.

---

*MATLAB EXPO 2019*
Examples

Two examples to demonstrate these approaches:

1. Traditional image processing for segmentation
2. Deep learning for object detection
Example 1: Part Inspection

Challenge:
- Find all of the items in the image
- Classify them - hook, nut or washer

Data
- Small number of images, unlabelled
- Taken from fixed position, controlled lighting
Two approaches to computer vision

‘Traditional’ Image Processing
Load Image

Import image

\[ I = \text{imread}('nutsAndBolts.png'); \]

Convert to grayscale

\[ I_{\text{gray}} = \text{rgb2gray}(I); \]
\[ \text{imshow}(I_{\text{gray}}) \]
Segmentation

• Next we want to segment the image

• Two routes:
  – Writing MATLAB code
  – Apps (and then generating code)
▪ Next segment out the parts

▪ Two routes:
  – Writing MATLAB code
  – Apps (and then generating code)
In code

```matlab
BW = imbinarize(Igray, 'adaptive', 'Sensitivity', 0.52,...
    'ForegroundPolarity', 'dark');
```
In code

```matlab
BW = imbinarize(Igray,'adaptive','Sensitivity',0.52,...
    'ForegroundPolarity','dark');
```

```matlab
BW = imcomplement(BW);
```
In code

```matlab
BW = imbinarize(Igray, 'adaptive', 'Sensitivity', 0.52,...
    'ForegroundPolarity', 'dark');
BW = imcomplement(BW);
radius = 9;
decomposition = 0;
se = strel('disk', radius, decomposition);
BW = imclose(BW, se);
```
In code

```matlab
BW = imbinarize(Igray, 'adaptive', 'Sensitivity', 0.52,...
    'ForegroundPolarity', 'dark');

BW = imcomplement(BW);

radius = 9;
decomposition = 0;
se = strel('disk', radius, decomposition);
BW = imclose(BW, se);

se2 = strel('disk', 3, 0);
BW = imopen(BW, se2);
```
In code

```matlab
BW = imbinarize(Igray, 'adaptive', 'Sensitivity', 0.52,...
    'ForegroundPolarity', 'dark');

BW = imcomplement(BW);

radius = 9;
decomposition = 0;
se = strel('disk', radius, decomposition);
BW = imclose(BW, se);

se2 = strel('disk', 3, 0);
BW = imopen(BW, se2);

BW = imfill(BW, 'holes');
```
Classification

Going to classify the parts based on their area

```matlab
[regions, numPixelRegions] = bwlabel(BW);
imshow(label2rgb(regions))
```
Classification

Going to classify the parts based on their area

```matlab
[regions, numPixelRegions] = bwlabel(BW);
imshow(label2rgb(regions))

stats = regionprops(regions, 'all');
for k=1:length(stats)
    text(stats(k).Centroid(1),stats(k).Centroid(2),... 
        sprintf('%04d',stats(k).Area), 'Hor','Center','Vert','middle')
end
```
Classification

```matlab
histogram(Area, 1000:100:4000)
xlabel('Area (pixels)')
ylabel('Number of Parts')
```

```matlab
minArea = [1300 1900 3200];
maxArea = [1800 2200 4100];
partNames = {'nut', 'ring', 'screw'};
partColors = {'magenta', 'green', 'cyan'};
Ipants = 1;
for k=1:3
    idx = Area > minArea(k) & Area < maxArea(k);
    Ipants = insertObjectAnnotation(Ipants,...
        'rectangle', vertcat(stats(idx).BoundingBox),...  
        partNames{k}, 'Color', partColors{k});
end

imshow(Ipants)
```
Example 1: Part Inspection

- ‘Basic’ image processing can solve this problem well
- Single feature (area) can be used to classify
- Fast, and east to interpret

MATLAB provides:
- High-level functions to chain together
- Apps to get started/learn functions
- Simple route to deployment
Adding more features

- More complex classifications will require more features
- More features leads to a more complicated model
  - Machine Learning
- Other ways to extract features
  - e.g. Visual bag of words
Example 2: Deep Learning

Challenge:
- Build an object detector to find cars

Data:
- Many images, each containing one or more cars
- Large variations in angle, lighting etc
- Labelled with bounding boxes (hopefully)
Two approaches to computer vision

Machine Learning

Images → COMPUTER → Model → Output
Deep Learning Workflow

**Prepare Data**
- Data access and preprocessing
- Ground truth labeling

**Train Model**
- Model design, Hyperparameter tuning
- Model exchange across frameworks
- Hardware-accelerated training

**Deploy**
- Multiplatform code generation (CPU, GPU)
- Edge deployment
- Enterprise Deployment
Dedicated MATLAB apps for automating and simplifying the labelling process.
Prepare Data

- Dedicated MATLAB apps for automating and simplifying the labelling process
- Split data in training and test sets
- Datastore objects to manage collections of data

```matlab
imdTrain = imageDatastore(trainingDataTbl{:, 'imageFilename'});
blsTrain = boxLabelDatastore(trainingDataTbl{:, 'vehicle'});
trainingData = combine(imdsTrain, bldsTrain);
```
Train Model

- Using YOLOv2 model architecture
  - Start-of-the-art object detector
  - Capable of running on real time video
  - Documented example

- Two stages:
  - Feature extraction layers – use a pretrained research network
  - Detector layers – build ourselves

- Use Deep Network Designer to build the network graphically
Train Model
Train Model

Next define training options

\[
\begin{align*}
\text{options} &= \text{trainingOptions('sgdm', ...}
\text{'MiniBatchSize', 16, ....}
\text{'InitialLearnRate', 1e-3, ...}
\text{'MaxEpochs', 20});
\end{align*}
\]

And train the model

\[
\text{detector} = \text{trainYOLOv2ObjectDetector}(
\text{trainingData, lgraph, options});
\]
Testing

Evaluate model performance on validation images

```matlab
I = imread(testDataTbl.imageFilename{1});

% Run the detector.
[bboxes,scores] = detect(detector,I);

% Annotate detections in the image.
I = insertObjectAnnotation(I,'rectangle',bboxes,scores);
imshow(I)
```
Testing

Evaluate model performance across training and test sets:
- Recall – what proportion of the cars do I detect?
- Precision – of the detections I make, what proportion are correct

Look out for:
- Underfitting – performance poor on training and test data
- Overfitting – good performance on training data, poor on test data

Iterate to improve the model
Example 2: Deep Learning

- Deep learning a good fit because of variation in the data
- Learns both a feature representation and a detection model

MATLAB provides:
- Graphical tools for labelling and network design
- Pretrained models to build on top of
Deploying Algorithms

Excel Add-in Java .NET MATLAB Production Server

Standalone Application C/C++ Java Python .NET MATLAB Production Server

MATLAB Compiler

MATLAB Coder

Embedded Hardware

C, C++, CUDA

MATLAB EXPO 2019
Musashi Seimitsu Industry Co., Ltd.
Detect Abnormalities in Automotive Parts

Automated visual inspection of 1.3 million bevel gear per month

MATLAB use in project:
- Preprocessing of captured images
- Image labelling and annotation
- Deep learning based analysis
  - Various transfer learning methods
    - Combinations of CNN models, Classifiers
  - Estimation of defect area using Class Activation Map
  - Abnormality/defect classification
- Deployment to NVIDIA Jetson using GPU Coder
Summary

- Segmentation and object detection form the basis of many common computer vision tasks
- Select image processing or machine learning approaches based on specifics of your problem

- MATLAB supports full workflow for both routes:
  - Easy data management
  - Apps to get started
  - Robust implementations of mathematical methods
  - Visualisations tools
  - Deployment to enterprise and embedded systems
  - Wide range of examples to adapt to your projects
What Next?

▪ Deep Learning Onramp

▪ Other talks
  – AI techniques for Signal, Time-series and Text Data
  – Automated Driving System Design

▪ Demo stands
  – Deep Learning and Reinforcement Learning
  – Driverless – Science Museum exhibition stand

▪ Doc examples

▪ Application Engineer support