

# 農業に画像解析を

ースマート農業を支える必須技術を実際に体験しよう！ー

MathWorks Japan

アプリケーションエンジニアリング部

草野 駿一



# アジェンダ

- 画像処理の基礎
- 画像処理による画像のセグメンテーション
- 転移学習による画像の分類
- 点群処理による個々の樹木のセグメンテーション

# アジェンダ

- 画像処理の基礎
- 画像処理による画像のセグメンテーション
- 転移学習による画像の分類
- 点群処理による個々の樹木のセグメンテーション



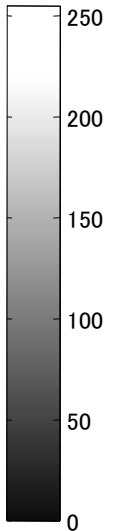
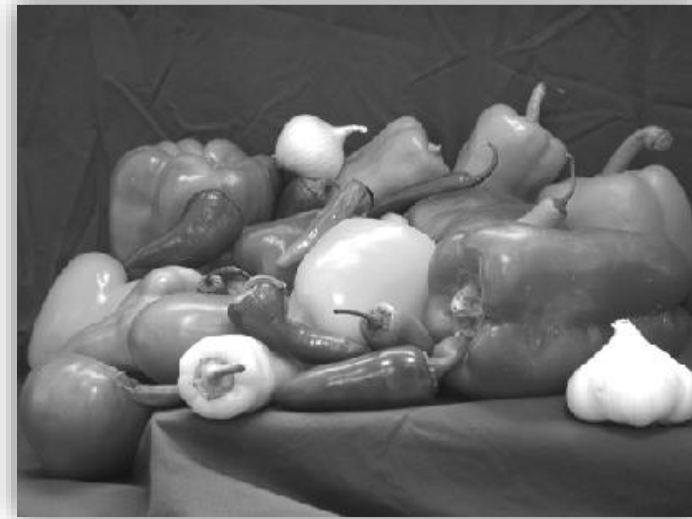
# デジタル画像とは？

画像	2次元行列
ピクセル	各要素
画像の高さ・幅	行列の縦・横方向
ピクセル個数	行列の要素数
ピクセル輝度値	行列の要素ごとのデータ値

123	163	213	123	123	163	213	123	121	128	123	126	138	23	123	226	213	123
138	23	123	226	138	23	123	226	47	72	113	123	34	111	186	123	186	123
34	111	186	123	34	111	186	123	13	23	83	91	253	38	161	173	213	123
253	38	161	173	253	38	161	173	54	44	121	103	121	128	123	126	121	128
123	123	123	123	123	163	213	123	47	72	47	72	47	72	113	123	47	72
123	123	123	123	138	23	123	226	13	23	13	23	13	23	83	91	13	23
123	123	123	123	34	111	186	123	54	44	54	44	54	44	121	103	54	44
121	128	123	126	253	38	161	173	123	163	213	123	121	128	123	126	123	163
47	72	113	123	123	226	123	226	138	23	123	226	47	72	113	123	138	23
13	23	83	91	186	123	186	123	34	111	186	123	13	23	83	91	34	111
54	44	121	103	161	173	161	173	253	38	161	173	54	44	121	103	253	38

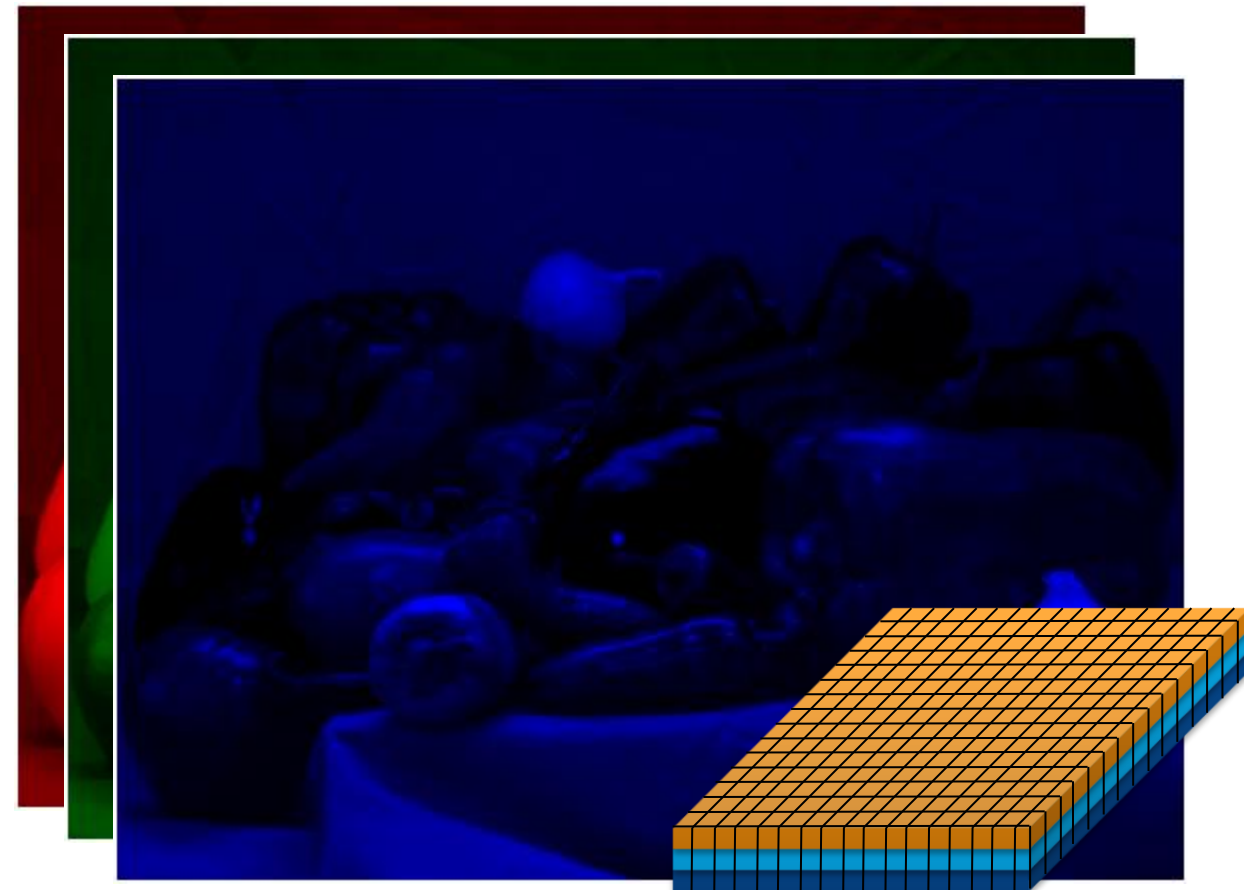
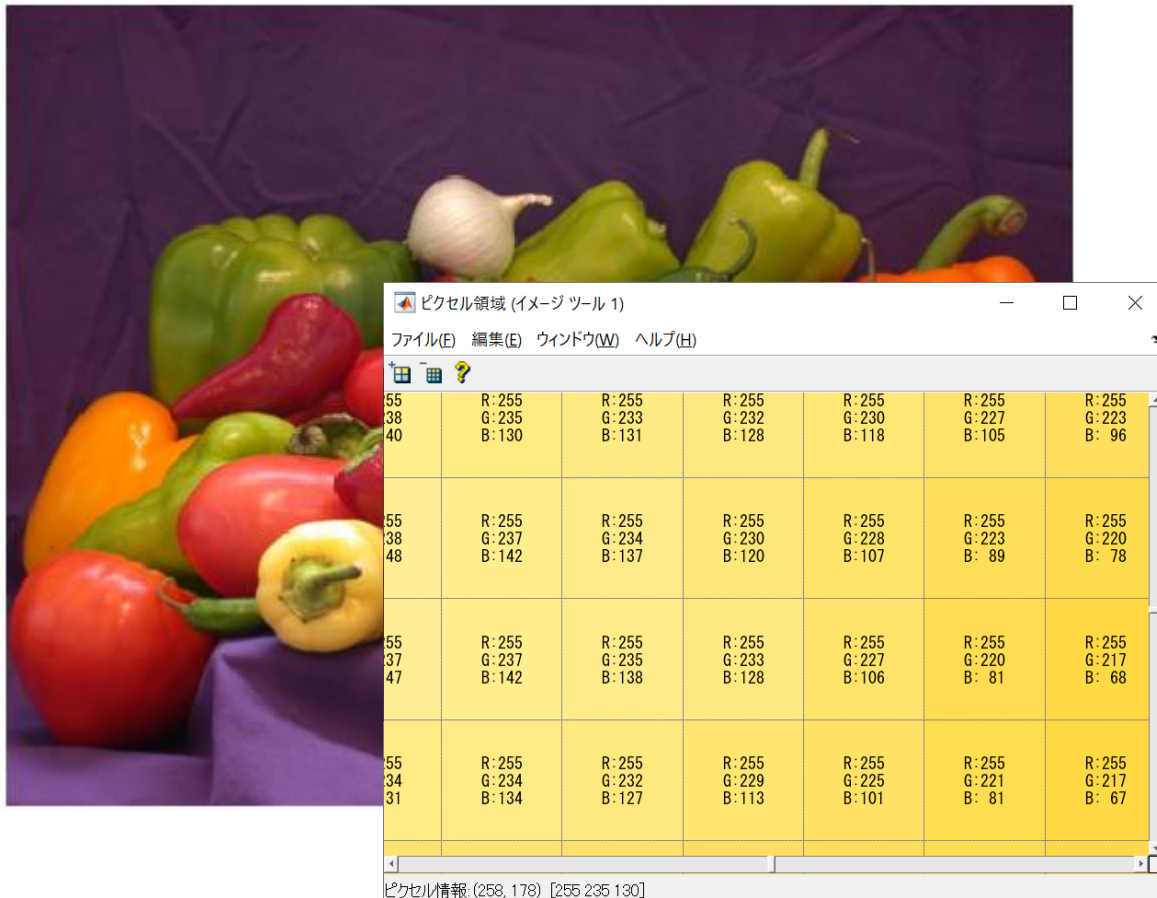
# グレースケール画像

- 各ピクセルの輝度値(Intensity)情報を持つ
- 輝度値を一定の範囲にマッピング
  - 範囲が 0 ~ 255 の場合、256階調
  - 最小輝度値: 黒、最大輝度値: 白
- 2次元行列、サイズ:  $M \times N$   
(M:縦方向ピクセル数、N:横方向ピクセル数)



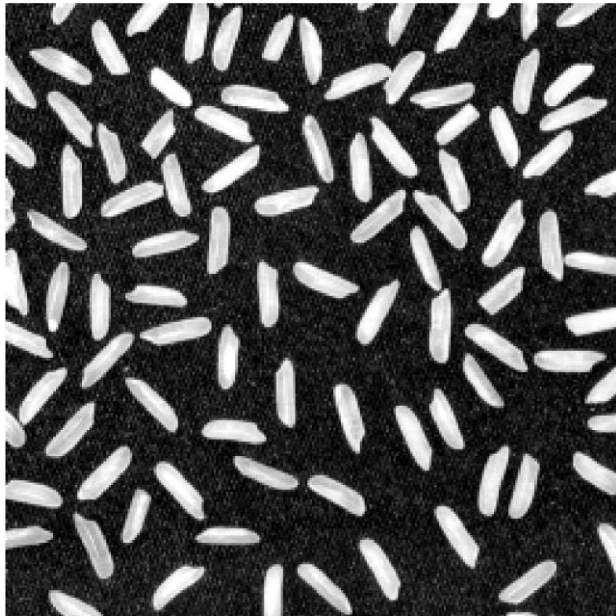
# 画像データの構造

- 画像はRGBチャンネルごとの輝度配列（M行N列の数値データ）を重ねたもの

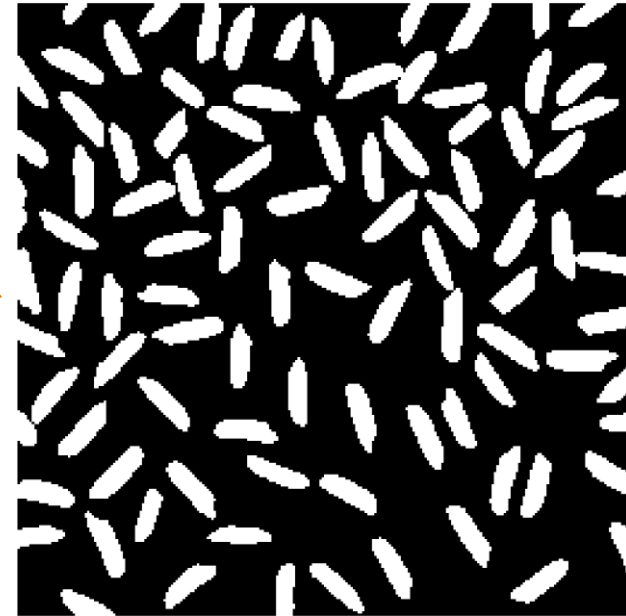


## バイナリー（2値化）画像

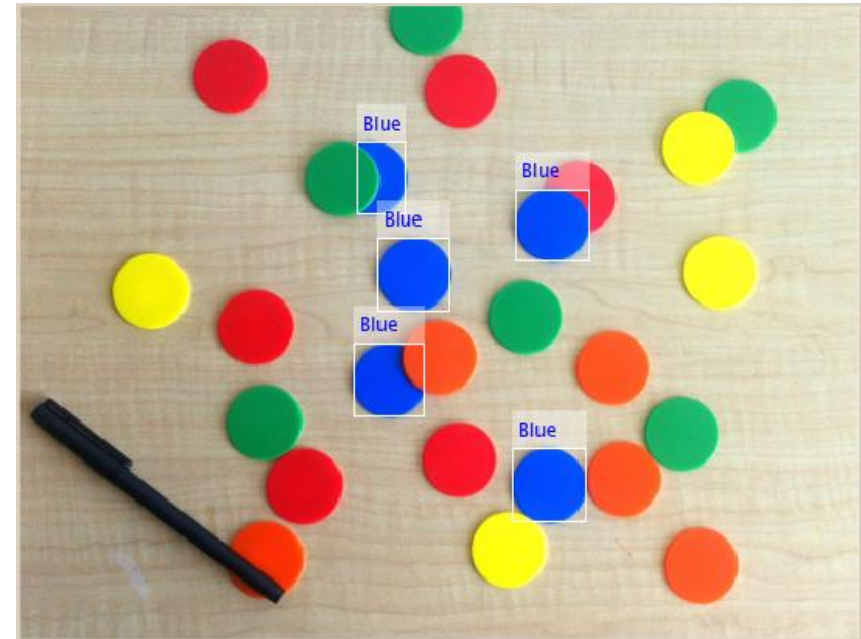
- 画素値が2つ固定した値、一般的に【0, 1】のどれかを持つ
- 「0」は黒（背景）、「1」は白（物体）を代表
- 用途: 画像内物体の識別、閾値化、ディザリング



一定以上の輝度値で  
二値化



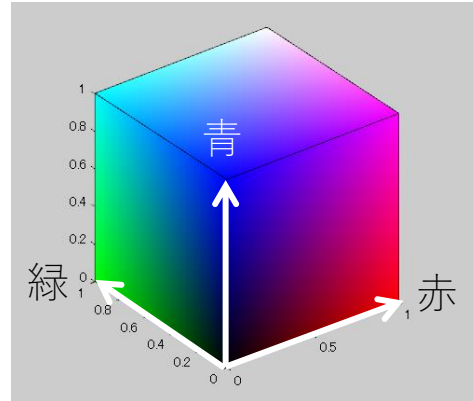
# 特定の色のチップのみ、抽出してみましょう





# 色空間を変えることで「色」を一つの次元で表現可能

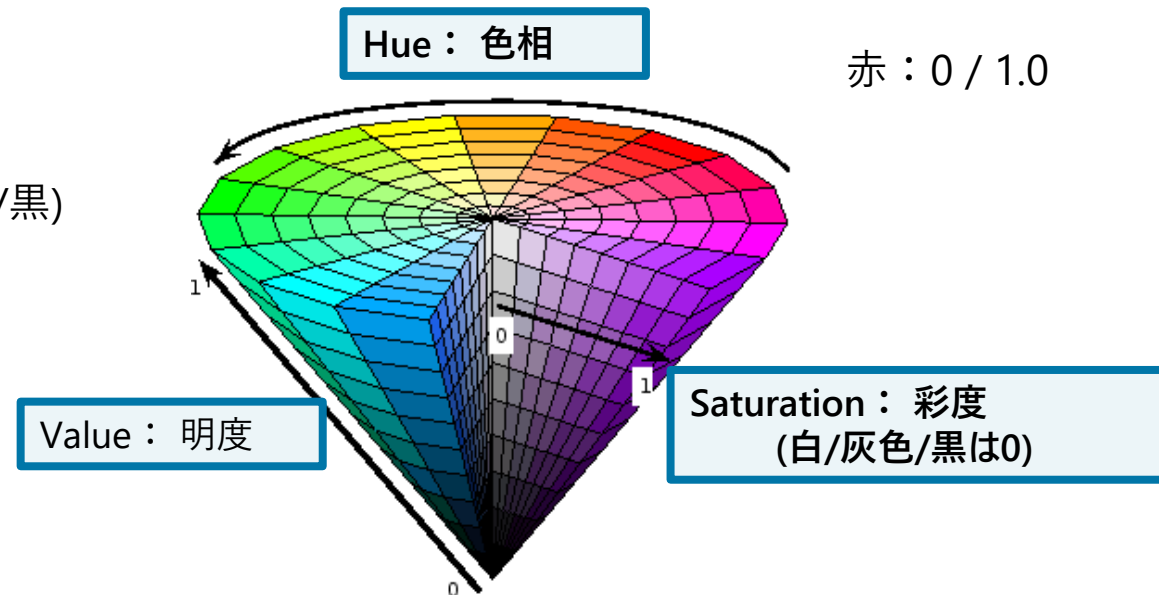
## RGB色空間



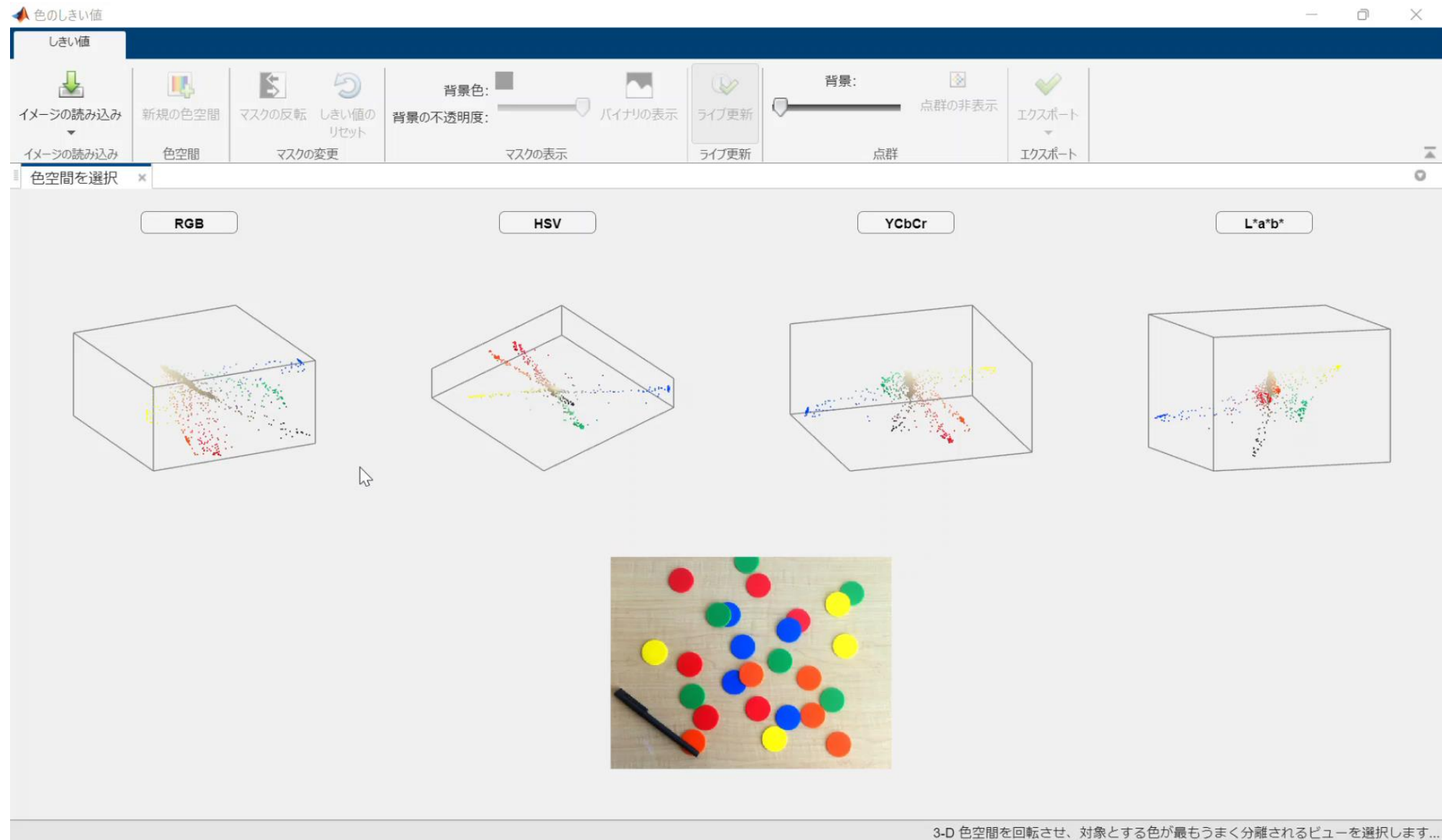
白	: (R,G,B) = (1, 1, 1)
黄色	: (R,G,B) = (1, 1, 0)
シアン (～水色)	: (R,G,B) = (0, 1, 1)
緑	: (R,G,B) = (0, 1, 0)
マゼンタ (紅紫色)	: (R,G,B) = (1, 0, 1)
赤	: (R,G,B) = (1, 0, 0)
青	: (R,G,B) = (0, 0, 1)
黒	: (R,G,B) = (0, 0, 0)

## HSV色空間

Hue : 色相  
 Saturation : 彩度 (0 : 白/灰色/黒)  
 Value : 明度

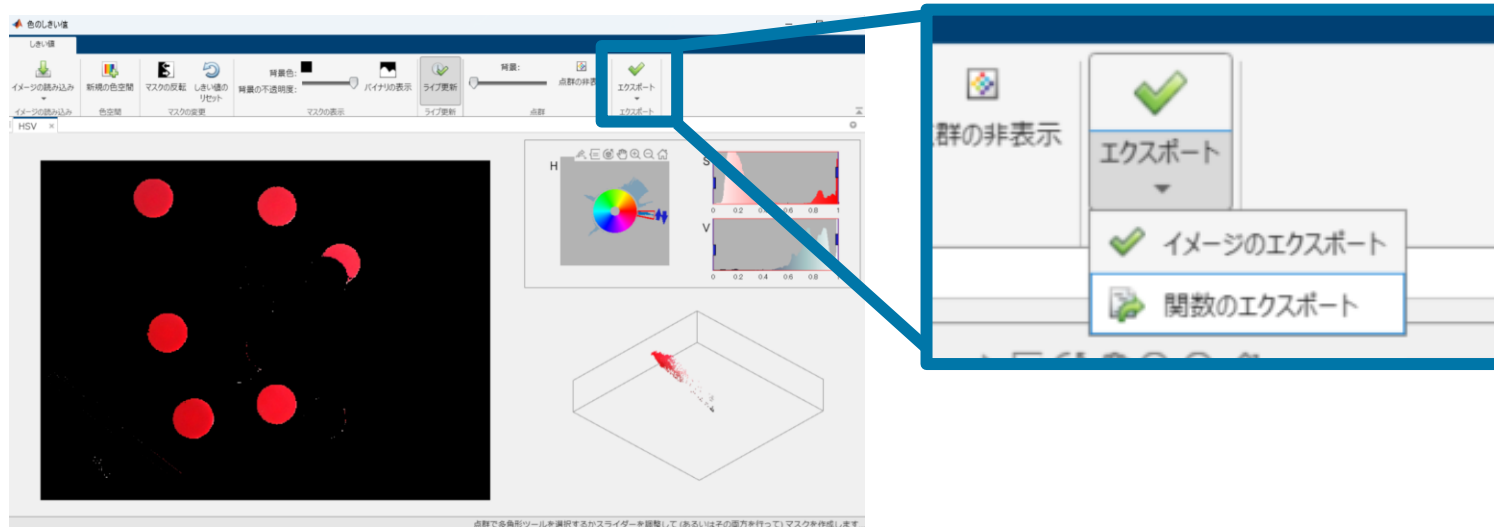


# 色のしきい値アプリでのチップ抽出の様子



# 色のしきい値アプリでのチップ抽出の様子

アプリ



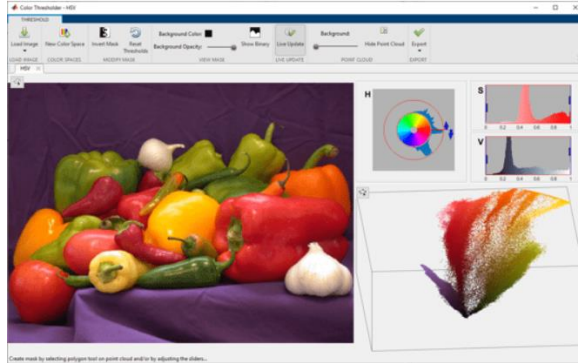
MATLAB関数

```

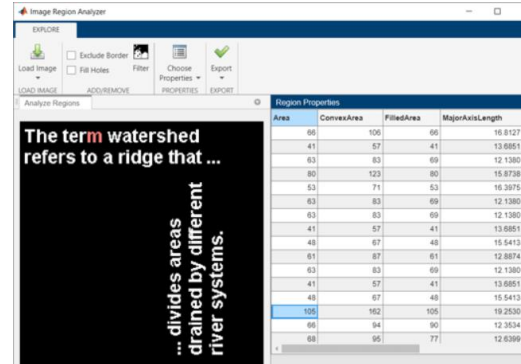
1 function [BW,maskedRGBImage] = createMask(RGB)
2 %createMask Threshold RGB image using auto-generated code from colorThresholder app.
3 % [BW,MASKEDRGBIMAGE] = createMask(RGB) thresholds image RGB using
4 % auto-generated code from the colorThresholder app. The colorspace and
5 % range for each channel of the colorspace were set within the app. The
6 % segmentation mask is returned in BW, and a composite of the mask and
7 % original RGB images is returned in maskedRGBImage.
8
9 % Auto-generated by colorThresholder app on 18-Jul-2023
10 %-----
11
12
13 % Convert RGB image to chosen color space
14 I = rgb2hsv(RGB);
15
16 % Define thresholds for channel 1 based on histogram settings
17 channel1Min = 0.968;
18 channel1Max = 0.000;
19
20 % Define thresholds for channel 2 based on histogram settings
21 channel2Min = 0.000;
22 channel2Max = 1.000;
23
24 % Define thresholds for channel 3 based on histogram settings
25 channel3Min = 0.000;
26 channel3Max = 1.000;
27
28 % Create mask based on chosen histogram thresholds
29 sliderBW = ( (I(:,:,1) >= channel1Min) | (I(:,:,1) <= channel1Max) ) & ...
30 (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
31 (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
32 BW = sliderBW;
33
34 % Initialize output masked image based on input image.
35 maskedRGBImage = RGB;
36
37 % Set background pixels where BW is false to zero.
38 maskedRGBImage(repmat(~BW,[1 1 3])) = 0;
39
40 end
  
```

アプリで行った処理はMATLABスクリプトに自動生成可能！

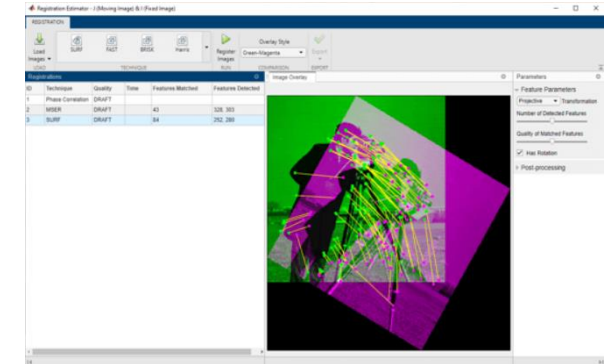
# アプリによる画像処理



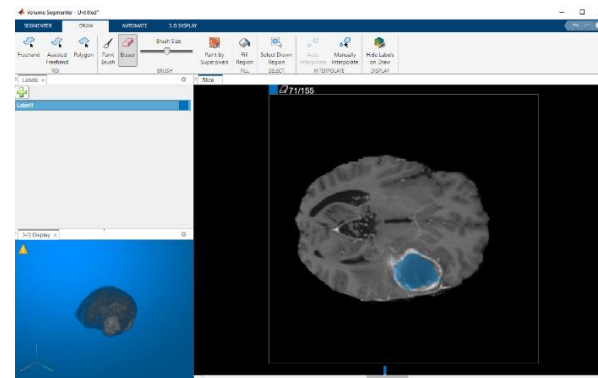
色のしきい値アプリ  
複数色空間を使った画像のセグメント化



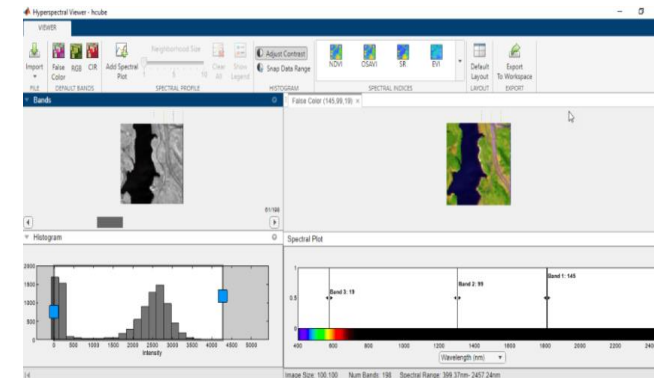
イメージの領域解析アプリ  
二値画像のプロパティ解析・フィルタリング



レジストレーション推定アプリ  
複数手法による画像の位置合わせ



ボリュームの領域分割アプリ  
3次元グレースケールのセグメンテーション



ハイパースペクトルビューアアプリ  
ハイパースペクトルデータの可視化・解析

# 様々な画像処理手法を解説するヘルプ機能と例題

逆引き形式で  
やりたい事から機能を検索

The screenshot shows the MathWorks Help Center page for the Image Processing Toolbox. The left sidebar contains a 'カテゴリ' (Categories) list with items like 'Computer Vision Toolbox', 'Image Processing Toolbox', 'Image Processing Toolbox 入門', 'インポート、エクスポートおよび変換', '表示と調査', '幾何学的変換とイメージ レジストレーション', 'イメージ処理の深層学習', '3次元ボリューム イメージ処理', 'ハイパースペクトル イメージ処理', 'コード生成と GPU サポート', 'Lider Toolbox', and 'Medical Imaging Toolbox'. The main content area is titled 'Image Processing Toolbox' and includes a description of the toolbox, a list of 'Image Processing Toolbox 入門' (Getting Started) topics, and a list of 'Image Processing Toolbox 入門' (Getting Started) topics. The 'Image Processing Toolbox 入門' section includes 'インポート、エクスポートおよび変換' (Import, Export, and Conversion), '表示と調査' (Display and Investigation), '幾何学的変換とイメージ レジストレーション' (Geometric Transformations and Image Registration), and 'イメージのフィルター処理と強調' (Image Filtering and Enhancement).

動画コンテンツにもヘルプからアクセス可能

ヘルプにあるコード例は  
MATLABでそのまま開いて実行可能

The screenshot shows the MATLAB Help Center page for the 'bwareaopen' function. The page title is 'bwareaopen' and the subtitle is 'バイナリ イメージからの小さなオブジェクトの削除'. The '構文' (Syntax) section shows the function signature: `BW2 = bwareaopen(BW,P)` and `BW2 = bwareaopen(BW,P,conn)`. The '説明' (Description) section explains that the function removes small objects from a binary image BW. The '例' (Example) section shows a code snippet: `BW2 = bwareaopen(BW,P,conn)` and a button to 'この例を開く' (Open this example). Below the code, there is a text box with the command `BW = imread('text.png');` and a button to 'Copy Command'.

アルゴリズムの実装詳細や文献情報も記載

実用的ですぐに試せる  
例題を豊富にご用意

The grid displays 16 thumbnails of image processing results and examples. The thumbnails are arranged in a 4x4 grid. The first row shows: 'Image Processing Toolbox 入門' (Introduction), 'イメージのインポート、処理およびエクスポートの基礎' (Basics of Importing, Processing, and Exporting Images), 'イメージ内の円形オブジェクトの検出と測定' (Detection and Measurement of Circular Objects in Images), and '不均一な照度の補正と輪縁オブジェクトの解析' (Correction of Non-uniform Illumination and Analysis of Edge Objects). The second row shows: 'マルチスペクトル イメージ内の物体の検出' (Detection of Objects in Multispectral Images), 'イメージのインポート、処理およびエクスポートの基礎' (Basics of Importing, Processing, and Exporting Images), 'イメージ内の円形オブジェクトの検出と測定' (Detection and Measurement of Circular Objects in Images), and '不均一な照度の補正と輪縁オブジェクトの解析' (Correction of Non-uniform Illumination and Analysis of Edge Objects). The third row shows: 'RGB 色空間と HSV 色空間の変換' (Conversion between RGB and HSV Color Spaces), 'L\*a\*b\* 値が RGB の色域にあるかどうかの判定' (Determining if L\*a\*b\* values are within the RGB color gamut), 'デスト チャート上の色の CIE94 色差の計算' (Calculation of CIE94 color differences for colors on a target chart), and '自動ホワイト バランス アルゴリズムの比較' (Comparison of Automatic White Balance Algorithms). The fourth row shows: 'イメージのセグメンテーションと解析' (Image Segmentation and Analysis), 'L\*a\*b\* 色空間を使った色ベースのセグメンテーション' (Color-based Segmentation using L\*a\*b\* Color Space), 'k-means クラスタリングを使った色ベースのセグメンテーション' (Color-based Segmentation using k-means Clustering), 'マーカール コントロール付き watershed セグメンテーション' (Watershed Segmentation with Marker Control), 'エッジ検出とモルフロジーを使用した領域の検出' (Detection of Regions using Edge Detection and Morphology), 'L\*a\*b\* 色空間を使った色ベースのセグメンテーション' (Color-based Segmentation using L\*a\*b\* Color Space), 'k-means クラスタリングを使った色ベースのセグメンテーション' (Color-based Segmentation using k-means Clustering), 'マーカール コントロール付き watershed セグメンテーション' (Watershed Segmentation with Marker Control), 'エッジ検出とモルフロジーを使用した領域の検出' (Detection of Regions using Edge Detection and Morphology), 'L\*a\*b\* 色空間を使った色ベースのセグメンテーション' (Color-based Segmentation using L\*a\*b\* Color Space), 'k-means クラスタリングを使った色ベースのセグメンテーション' (Color-based Segmentation using k-means Clustering), 'マーカール コントロール付き watershed セグメンテーション' (Watershed Segmentation with Marker Control), and 'エッジ検出とモルフロジーを使用した領域の検出' (Detection of Regions using Edge Detection and Morphology).

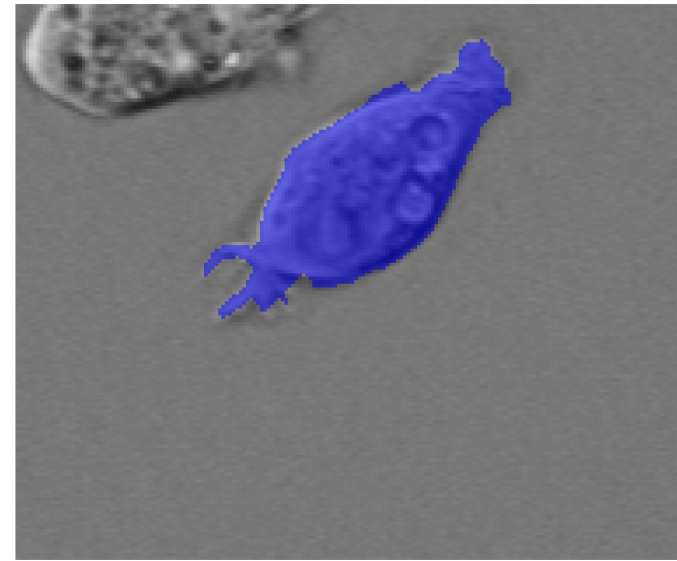
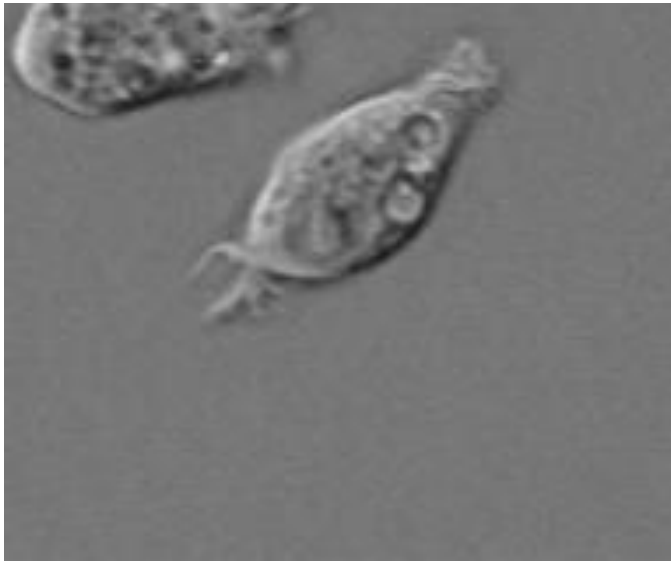


# アジェンダ

- 画像処理の基礎
- 画像処理による画像のセグメンテーション
- 転移学習による画像の分類
- 点群処理による個々の樹木のセグメンテーション

# 画像から細胞を抽出

色によるセグメンテーションが使えない時どのような方法があるか



- 応用

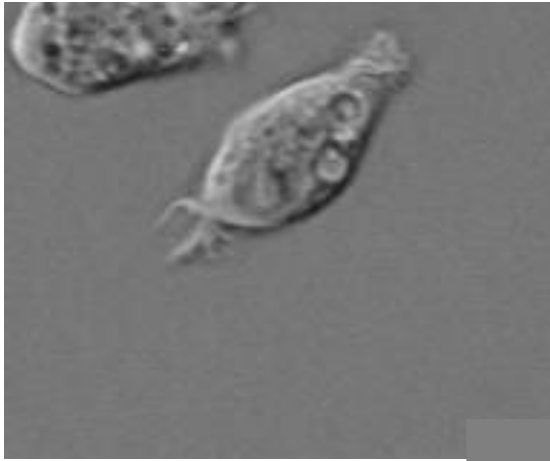
- 面積の計測
- 数の計測

- 形の計測
- ビデオでのトラッキング

# 手順



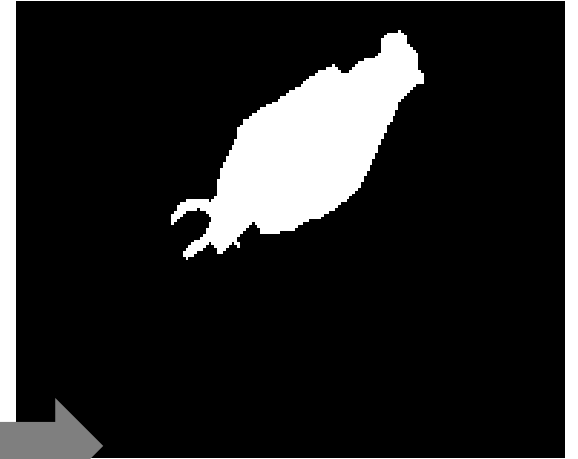
元画像



エッジ検出

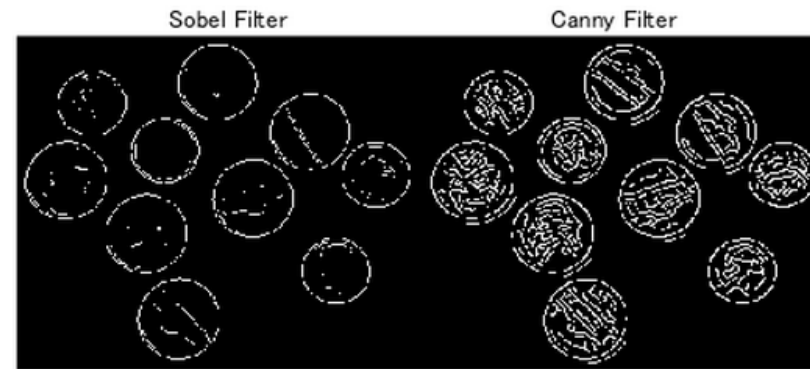


モルフォロ  
ジー処理



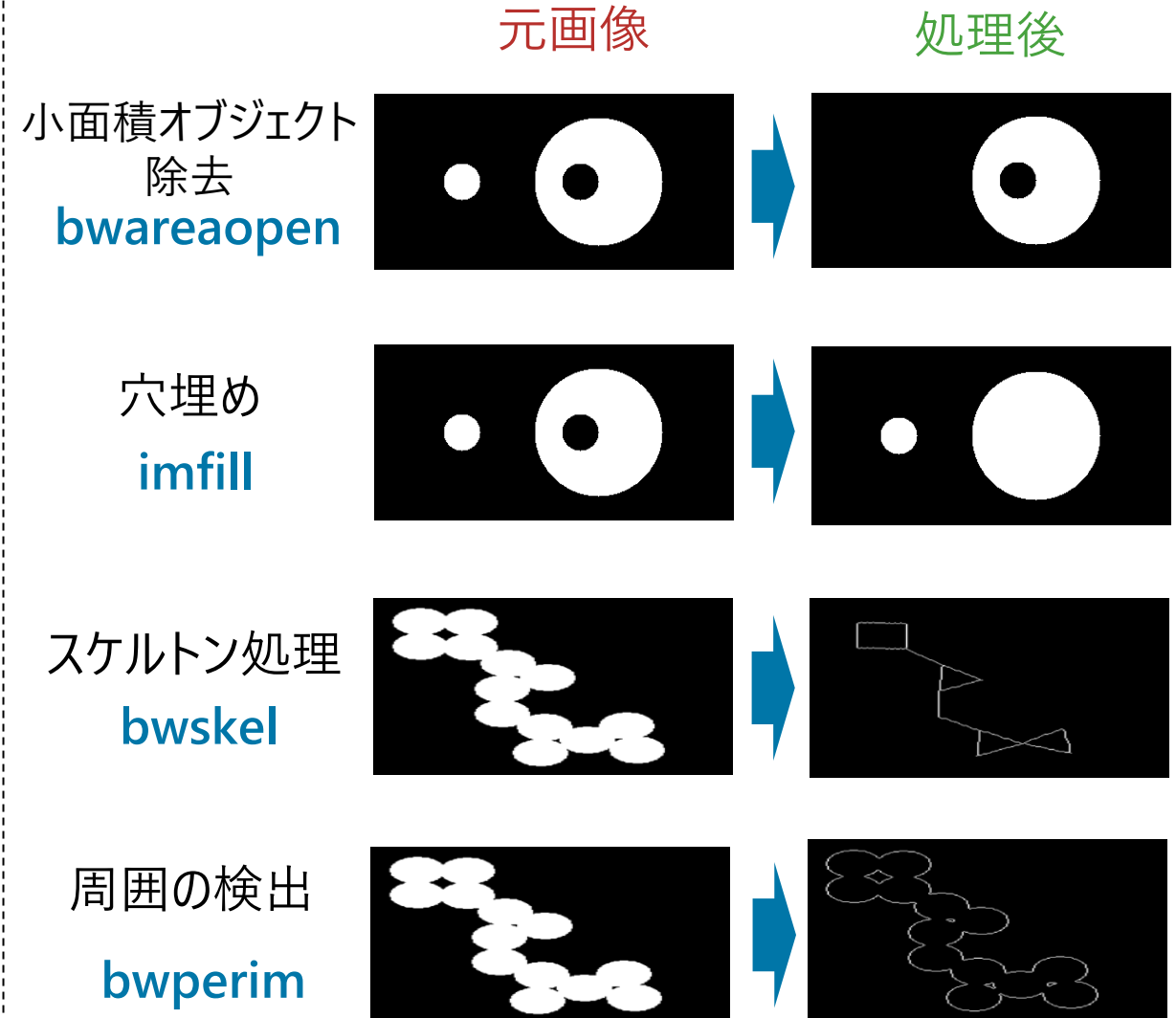
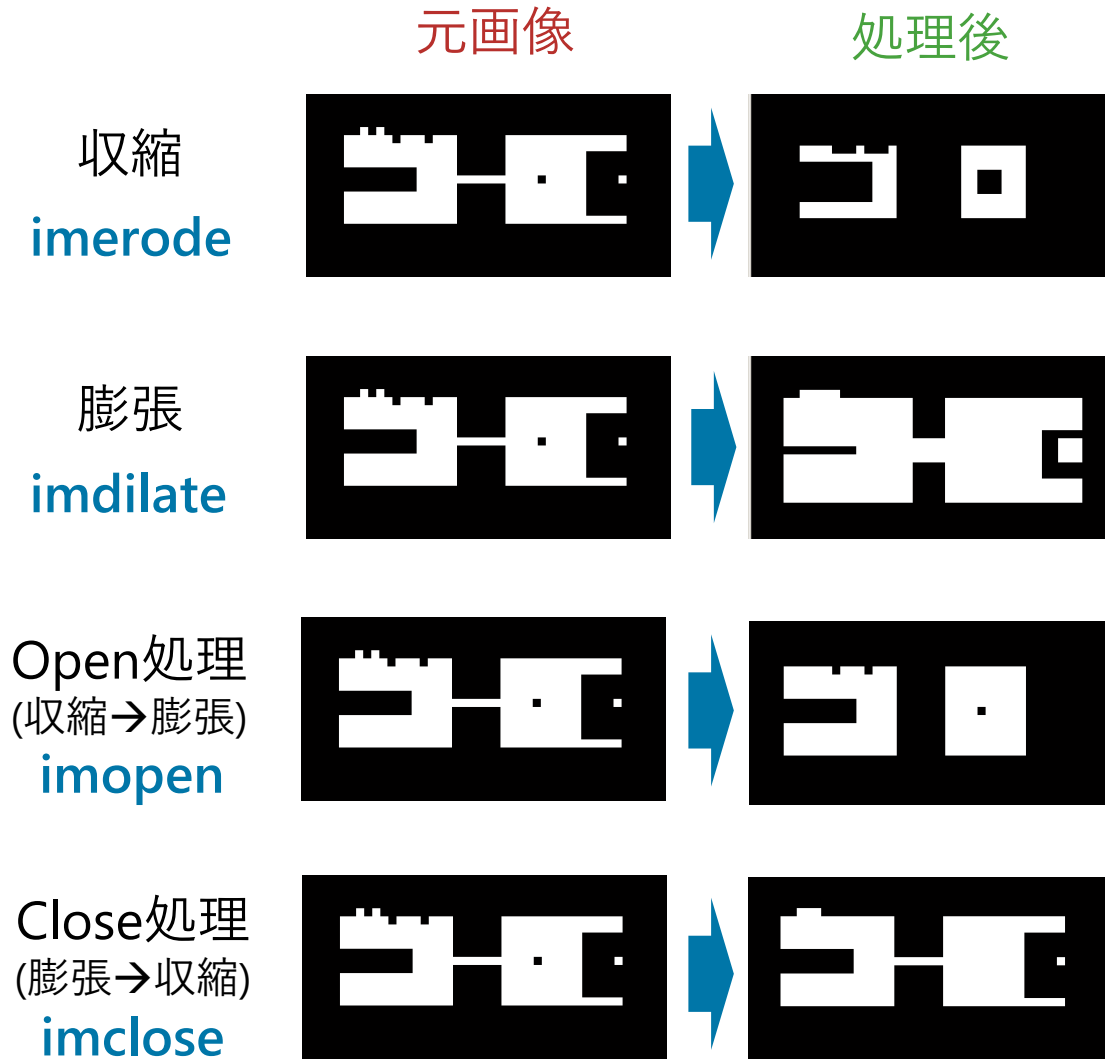
# エッジ検出 / 二値化

- エッジの検出 **edge**



隣り合う画素の変化の大きさを計算→二値化→エッジ

# モルフォロジー処理



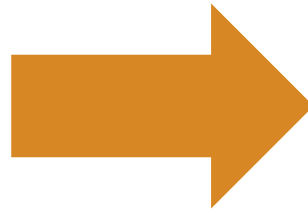


# フィルタリングとは？

- 原画像に何らかの演算を施し、必要な情報を取り出すこと
- 目的に応じて様々なフィルタがある



原画像



メディアンフィルタ

# 2次元フィルタリングの実行

## imfilter関数

```
f = fspecial('xxx',c); % 係数の作成  
fimg = imgfilter(img,f); % フィルタリング
```

<https://jp.mathworks.com/help/images/ref/fspecial.html>



原画像



Sobelエッジ検出 : sobel



平均化 : average



モーション : motion



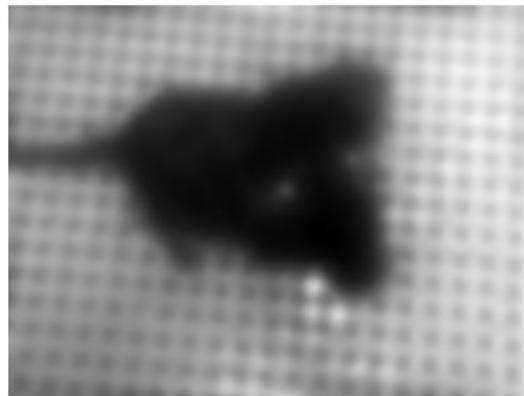
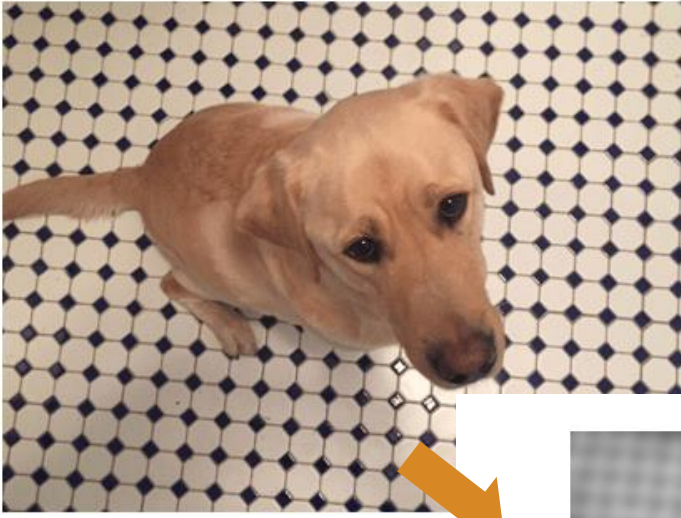
エッジ・コントラスト強調 : unsharp mask



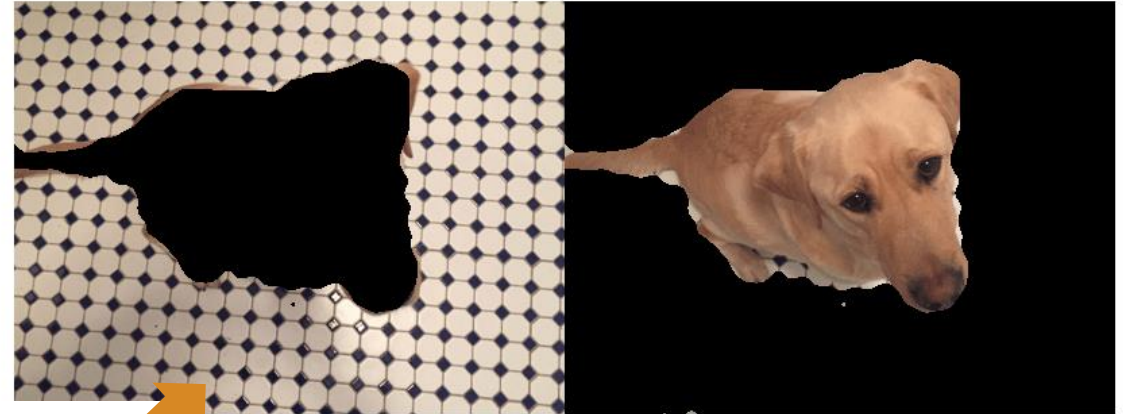
巡回平均化 : disk

# テクスチャによるセグメンテーション

元画像



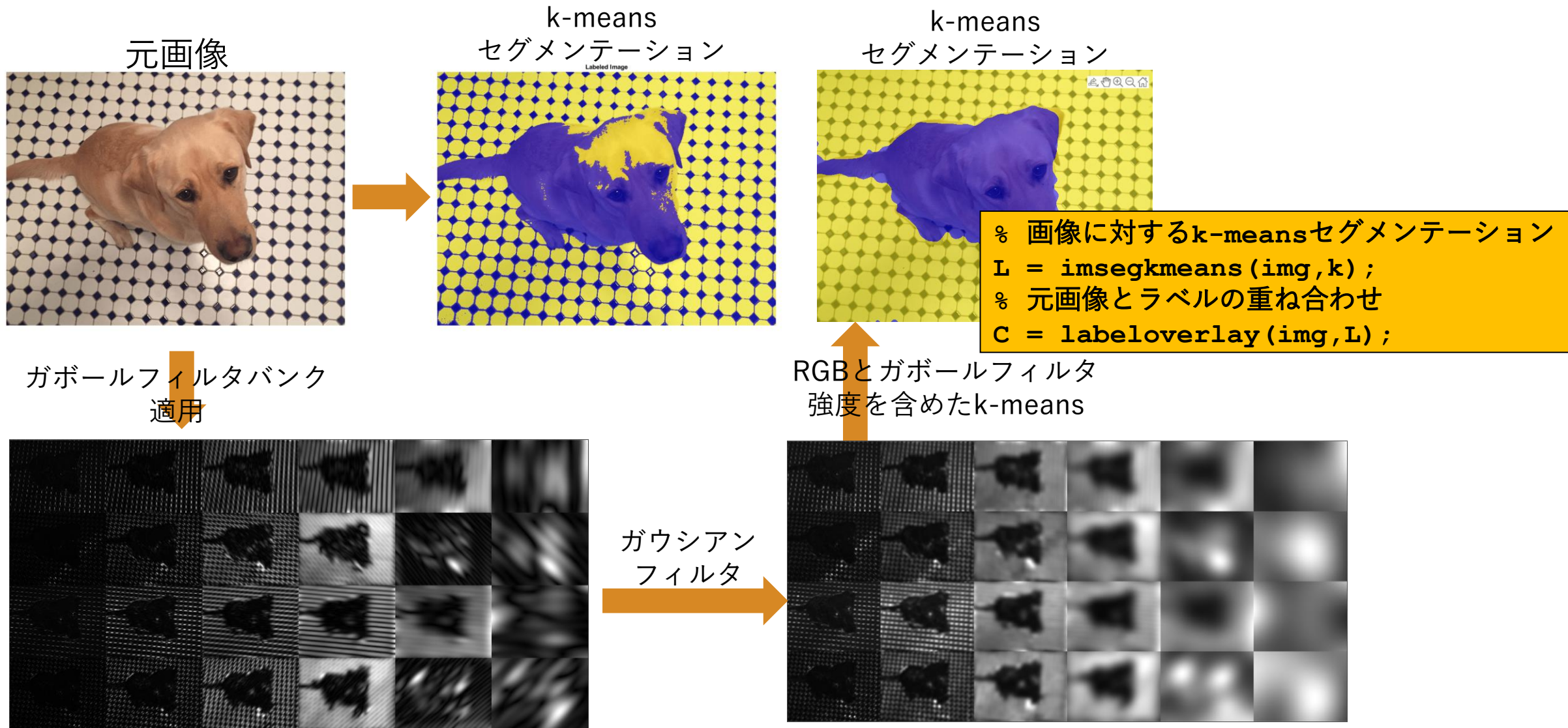
ガボールフィルタによるテクスチャ分類



k-means  
セグメンテーション



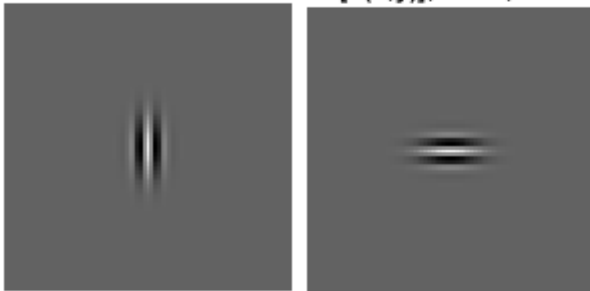
# テクスチャによるセグメンテーションの例のフロー



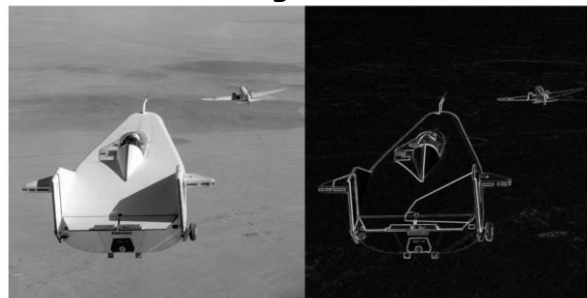
# テクスチャ計算用の種々の関数

関数名	名称	内容
<code>gabor / imgaborfilt</code>	ガボールフィルタ	特定の向き・大きさを持つ線を強調することができる。複数種類を組合わせてセグメンテーションに活用。
<code>rangefilt</code>	レンジフィルタ	局所領域内の輝度の最大値と最小値の差によるテクスチャ表現
<code>stdfilt</code>	標準偏差フィルタ	局所領域内の輝度の標準偏差によるテクスチャ表現
<code>entropyfilt</code>	エントロピーフィルタ	局所領域内の輝度を行列と見た場合のエントロピーによるテクスチャ表現
<code>graycomatrix</code>	グレイレベルの同時生起行列を作成	同時生起行列によるテクチャ表現

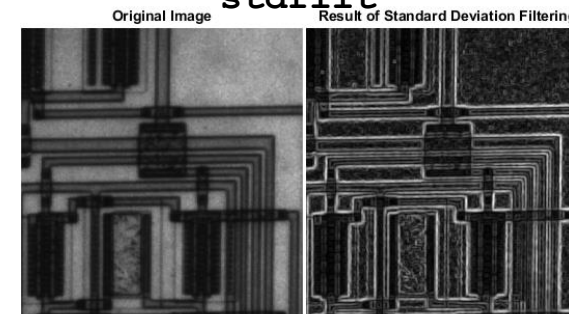
$\text{Re}[h(x,y)], \lambda = 5, \theta = 0$     $\text{Re}[h(x,y)], \lambda = 5, \theta = 90$



`rangefilt`



`stdfilt`



## サポートされている静止画フォーマット (R2019b)

### **imread()** によりサポートされている静止画フォーマット

**BMP** — Windows ビットマップ  
**CUR** — Cursor ファイル  
**GIF** — Graphics Interchange Format  
**HDF4** — Hierarchical Data Format  
**ICO** — Icon ファイル  
**JPEG** — Joint Photographic Experts Group  
**JPEG 2000** — Joint Photographic Experts Group 2000  
**PBM** — Portable Bitmap  
**PCX** — Windows Paintbrush  
**PGM** — Portable Graymap  
**PNG** — ポータブル ネットワーク グラフ  
**PPM** — Portable Pixmap  
**RAS** — Sun ラスター  
**TIFF** — Tagged Image File Format  
**XWD** — X Window Dump

詳細 <http://www.mathworks.co.jp/jp/help/matlab/ref/imread.html>



# サポートされている動画フォーマット (R2019b)

## VideoReader によりサポートされている動画フォーマット

### Windows

**MPG - MPEG-1**

**WMV, ASF, ASX - Windows Media® ビデオ**

**Microsoft® DirectShow® がサポートするすべての形式**

### Windows7以降

**MP4, M4V - H.264 エンコード ビデオ(Windows 7 のみ)**

**MOV - Apple QuickTime Movie**

**Microsoft Media Foundation がサポートするすべての形式。**

### Macintosh

**MPEG-1, MP4, M4V**

**MOV – Apple QuickTime Movie**

**3GPP, 3GPP2, AVCHD, DV**

**MOV - QuickTime Movie および QuickTime がサポートするほとんどの形式**

### Linux®

**GStreamer 0.10 以降用のプラグインでサポートされる任意の形式**

**全てのプラットフォームに共通**

**AVIファイル**

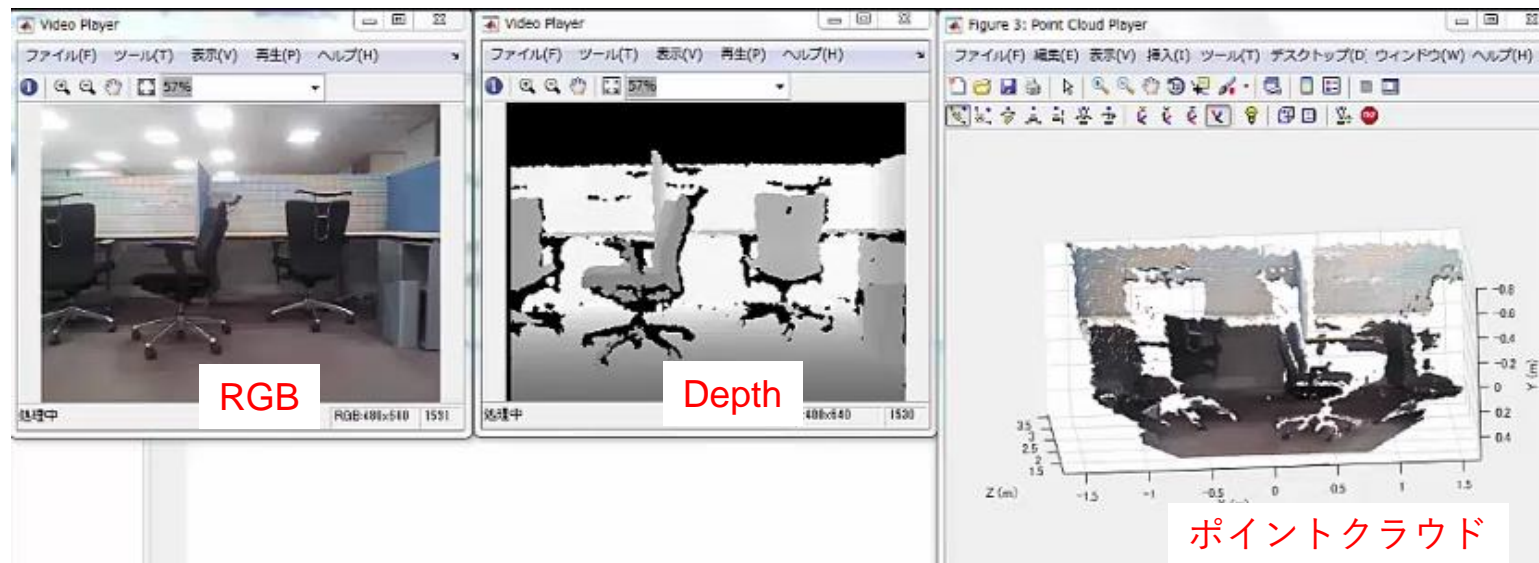
**Motion JPEG2000**

詳細 <http://www.mathworks.co.jp/jp/help/matlab/ref/videoreaderclass.html>

# 各種カメラデバイスからのデータ取り込み

## Image Acquisition Toolbox

- 業界標準のHWからの動画像取込み機能を提供
  - フレームグラバ
    - Analog
    - Camera Link
  - DCAM 互換 FireWire (IIDC 1394)
  - GigE Vision
  - USB3 Vision
  - IPカメラ
- Microsoft Kinect

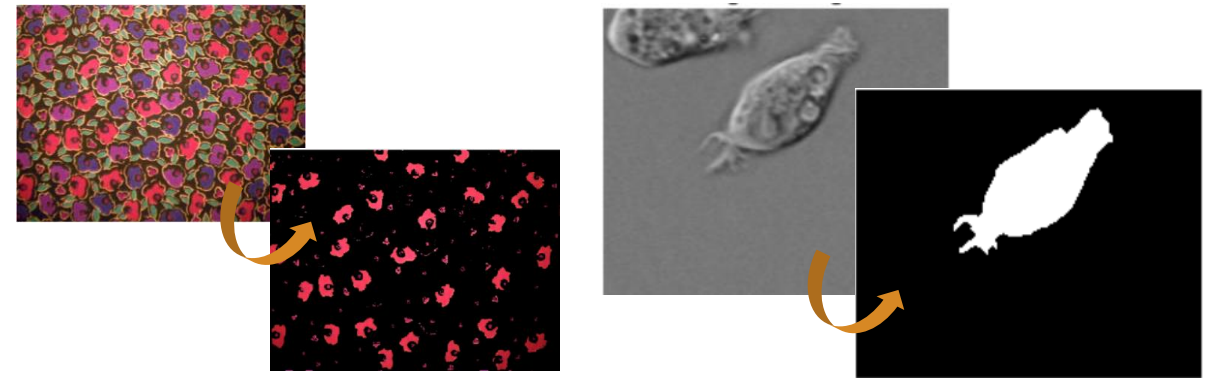


# 前処理/後処理に利用できる様々な画像処理

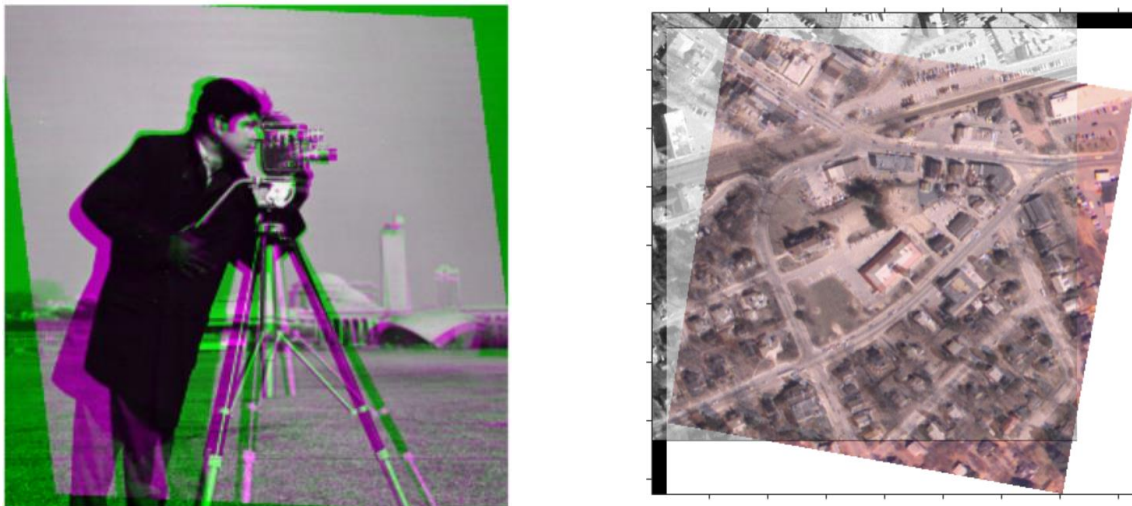
## モルフォロジー処理



## セグメンテーション



## レジストレーション



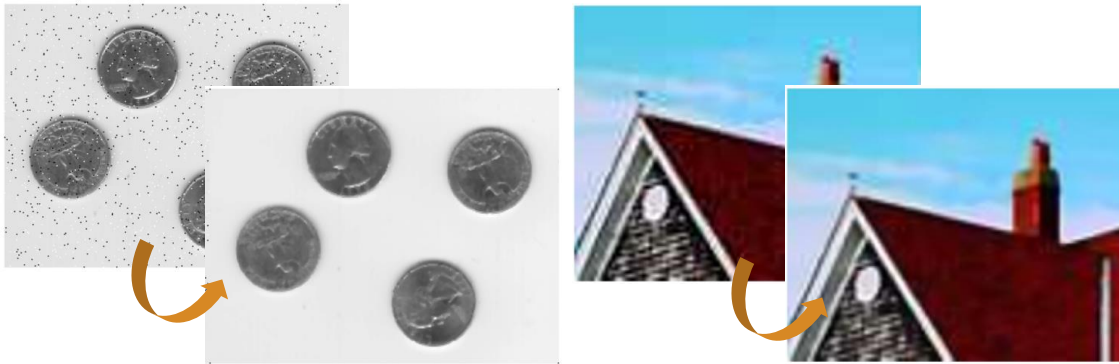
## 高解像度化





# 前処理/後処理に利用できる様々な画像処理

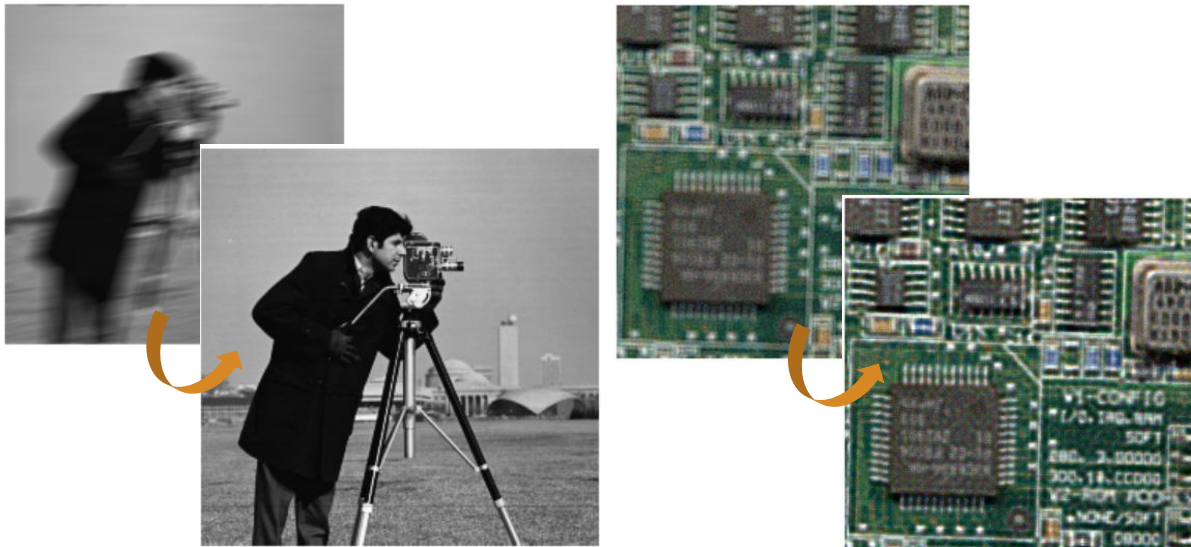
## ノイズ除去



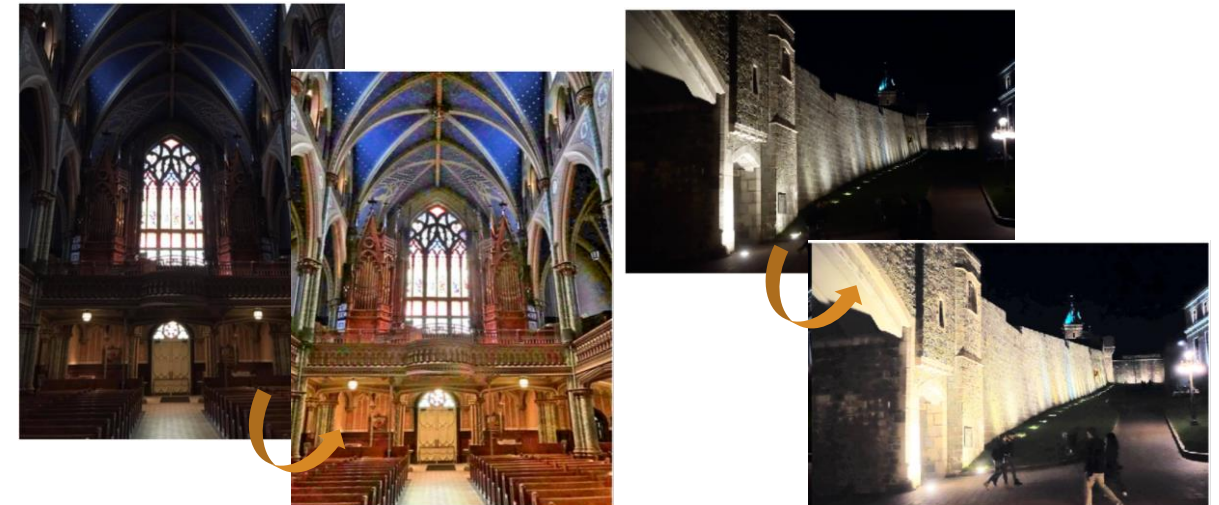
## 明るさ調整



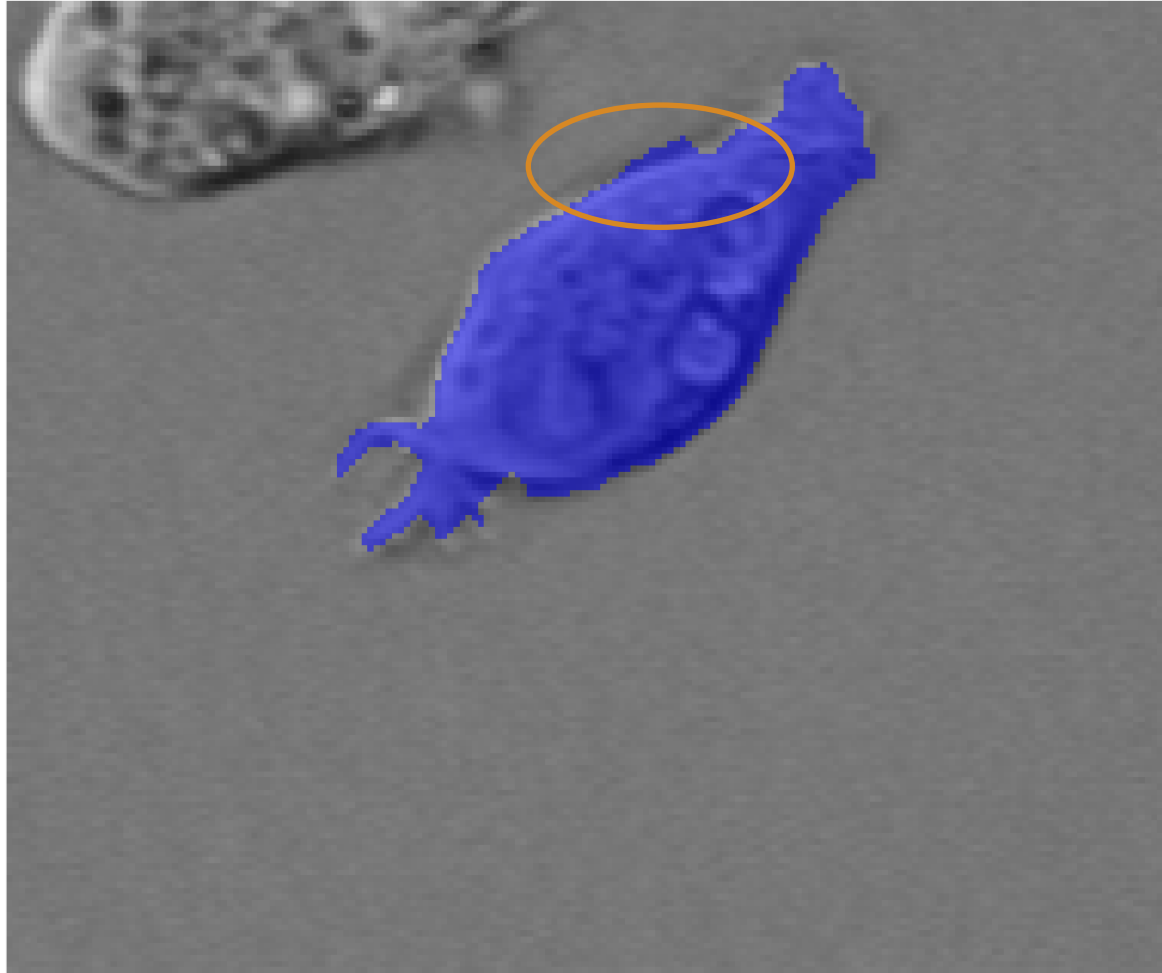
## ブレ・ぼやけ除去



## 低光量画像の強調

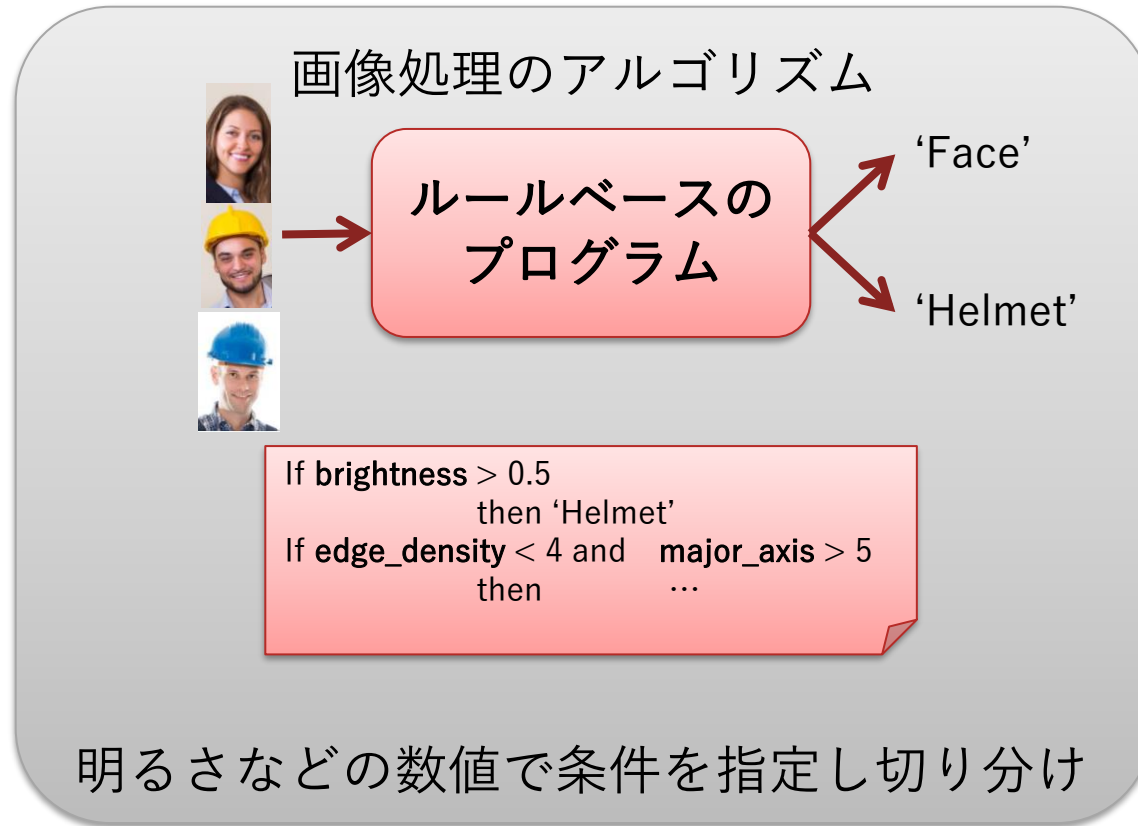


## 画像処理がうまく動かないケース



しきい値によって局所的にうまくいかない箇所が出てくるのはよくある

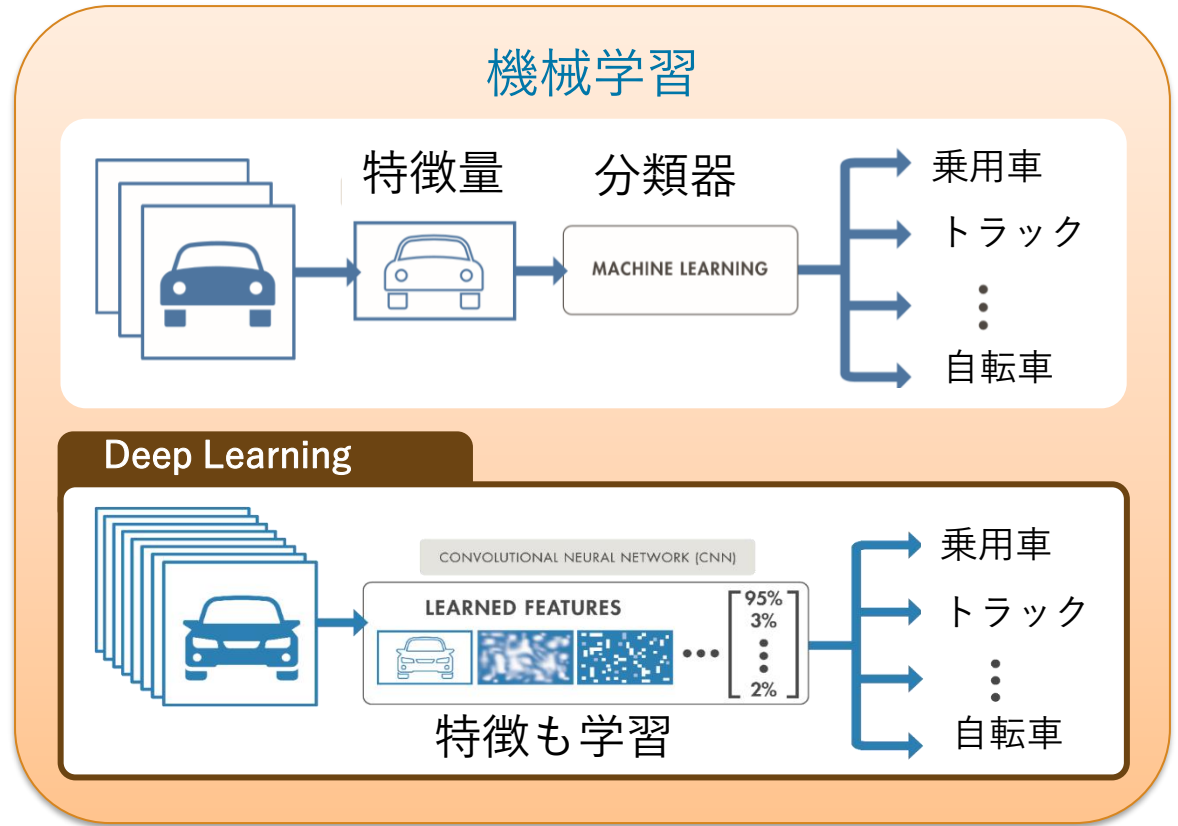
# ルールベースの画像処理と機械学習の比較



直感的で明確な切り分け、計算コスト小

条件が複雑になるとコーディングが複雑に

制御された環境、シンプルな計測に強い



柔軟な切り分けに強い

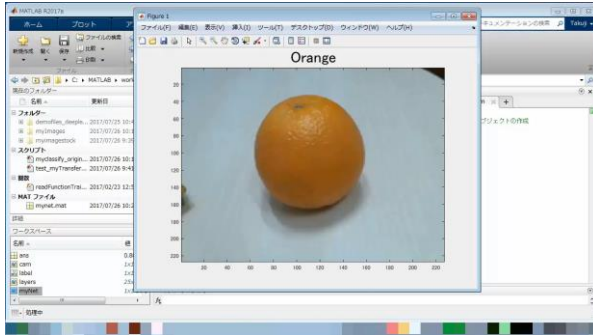
画像データが多く必要

自然環境、複雑な条件に適用可能

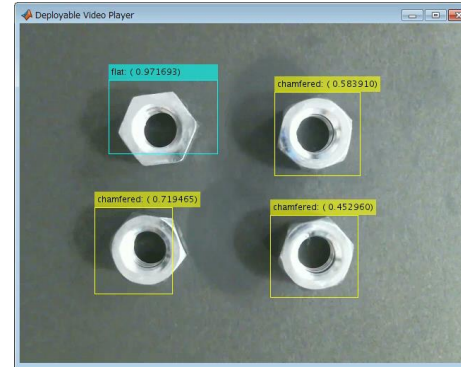


# ディープラーニングの画像への適用でできること

## 分類



## 検出



## セグメンテーション



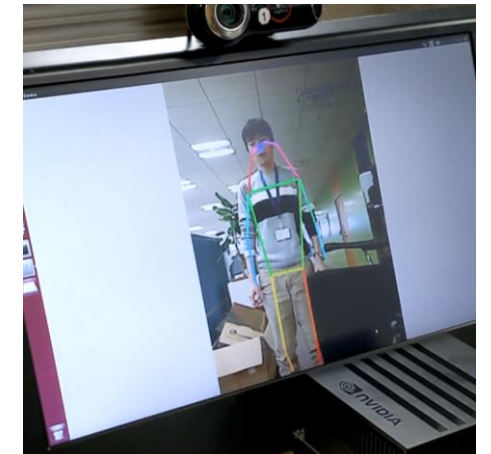
## 回帰



## 文字認識



## 骨格検出など

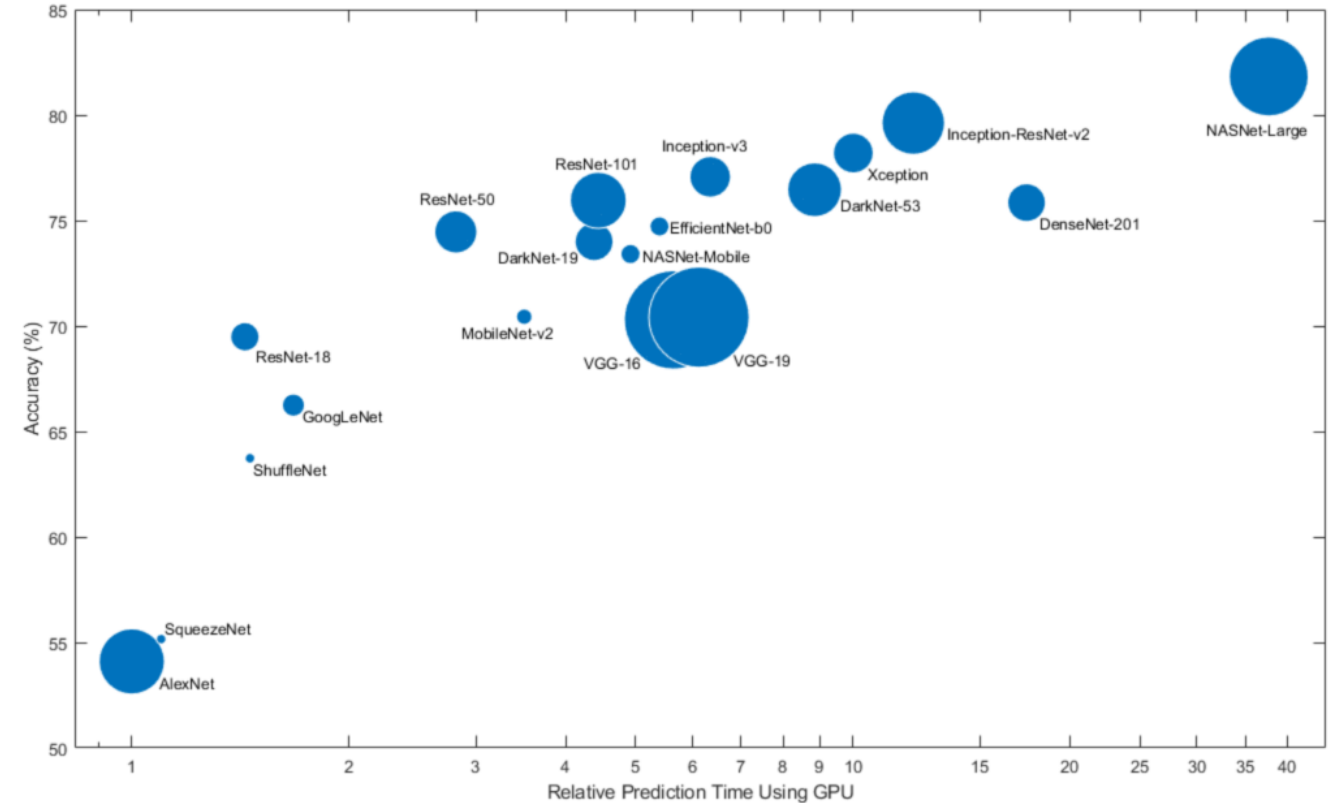


# ディープラーニングを利用する2つの方法

- 学習済みネットワークの利用
- 転移学習
- スクラッチから学習

## 事前学習済みの分類モデル

- ImageNetデータセットで学習
- 1000クラスの分類
- 性能と速度のトレードオフ



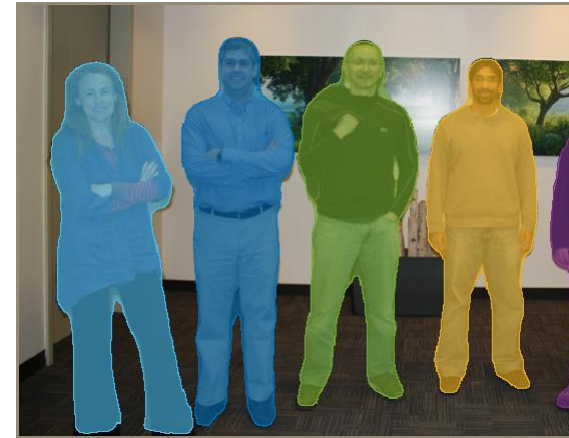
# その他のタスクでの事前学習済みモデル (COCOデータセットで学習)

## 物体検出



- R-CNN ファミリー
- SSD
- YOLOv2/v3/v4/X

## インスタンスセグメンテーション



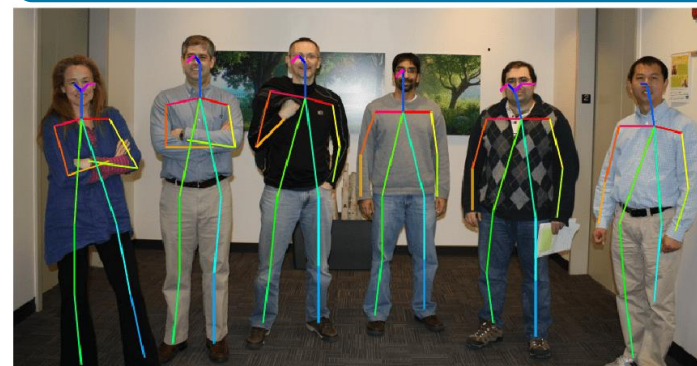
- Mask R-CNN
- SOLOv2

## セマンティックセグメンテーション

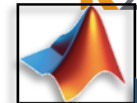


- [Segnet](#)
- [U-Net](#)
- [DeepLabv3+](#)

## 骨格検出



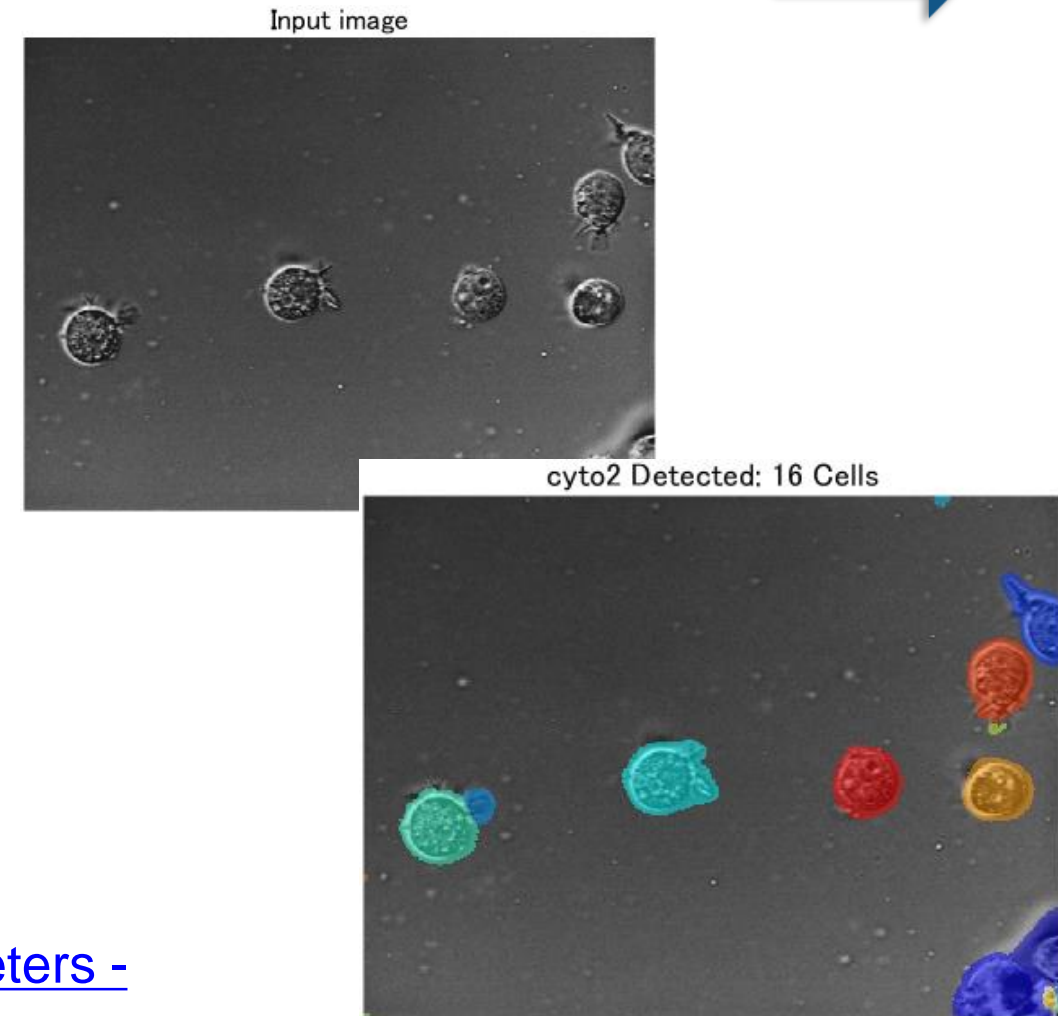
- OpenPose
- HRNet



Demo

# Cellpose : 顕微鏡画像の細胞セグメンテーション

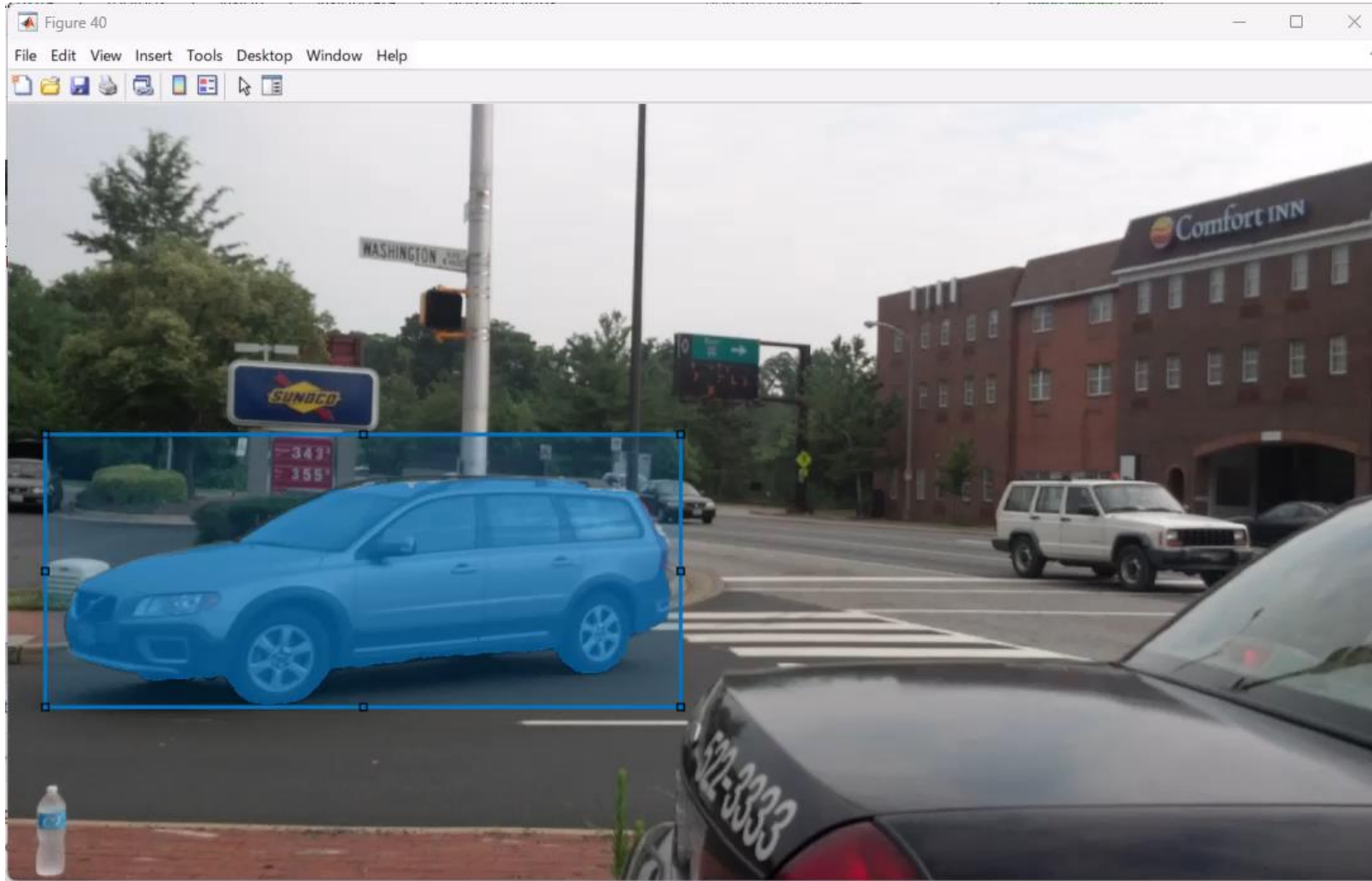
- 顕微鏡画像の細胞のセグメンテーションのための事前学習済みモデル
- オリジナルデータで学習も可能
- 2D / 3Dのセグメンテーション



[Refine Cellpose Segmentation by Tuning Model Parameters - MATLAB & Simulink - MathWorks 日本](#)

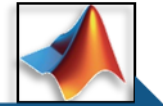
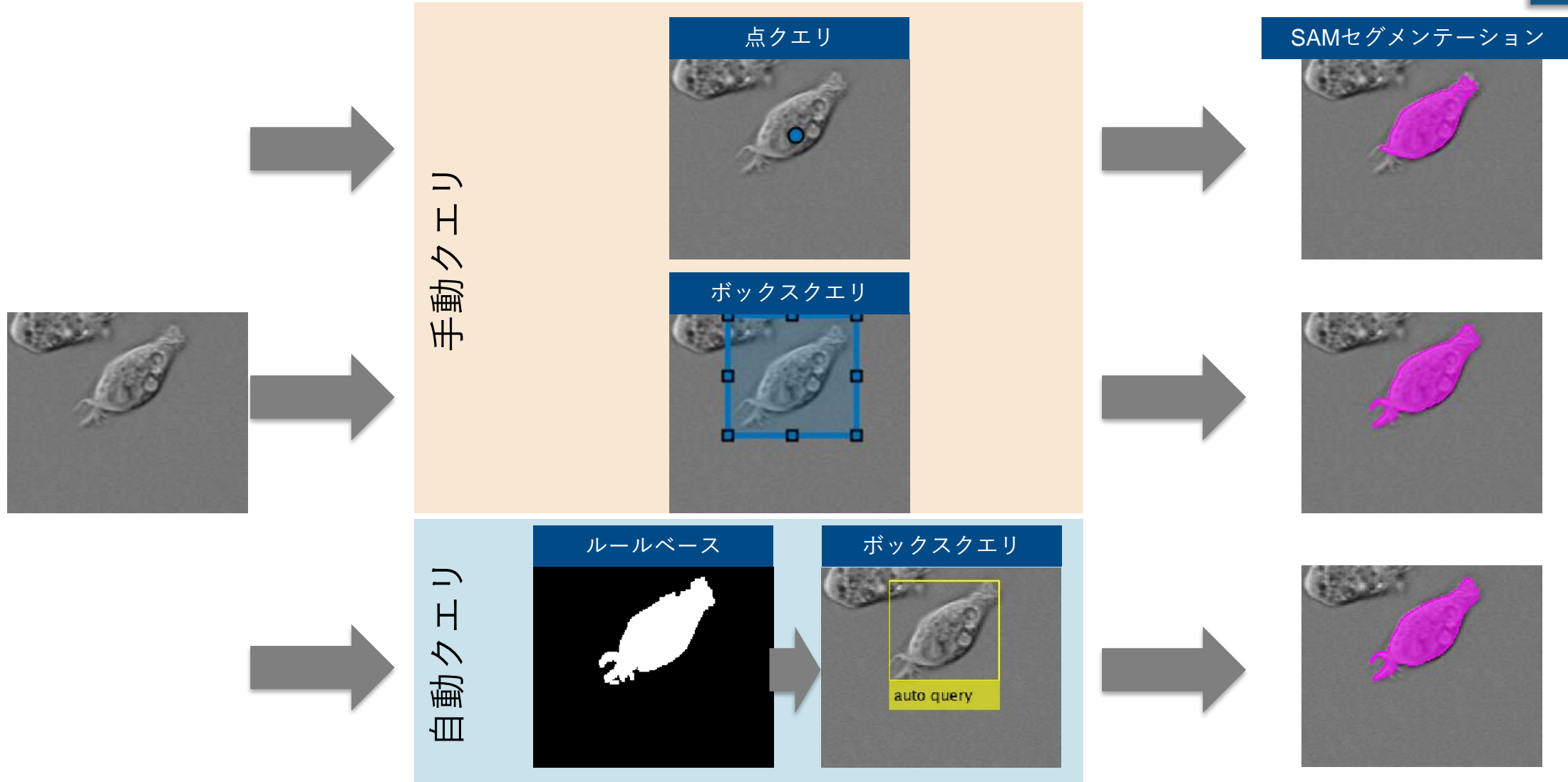


# ゼロショットセグメンテーションモデル





# SAMによる細胞のセグメンテーション

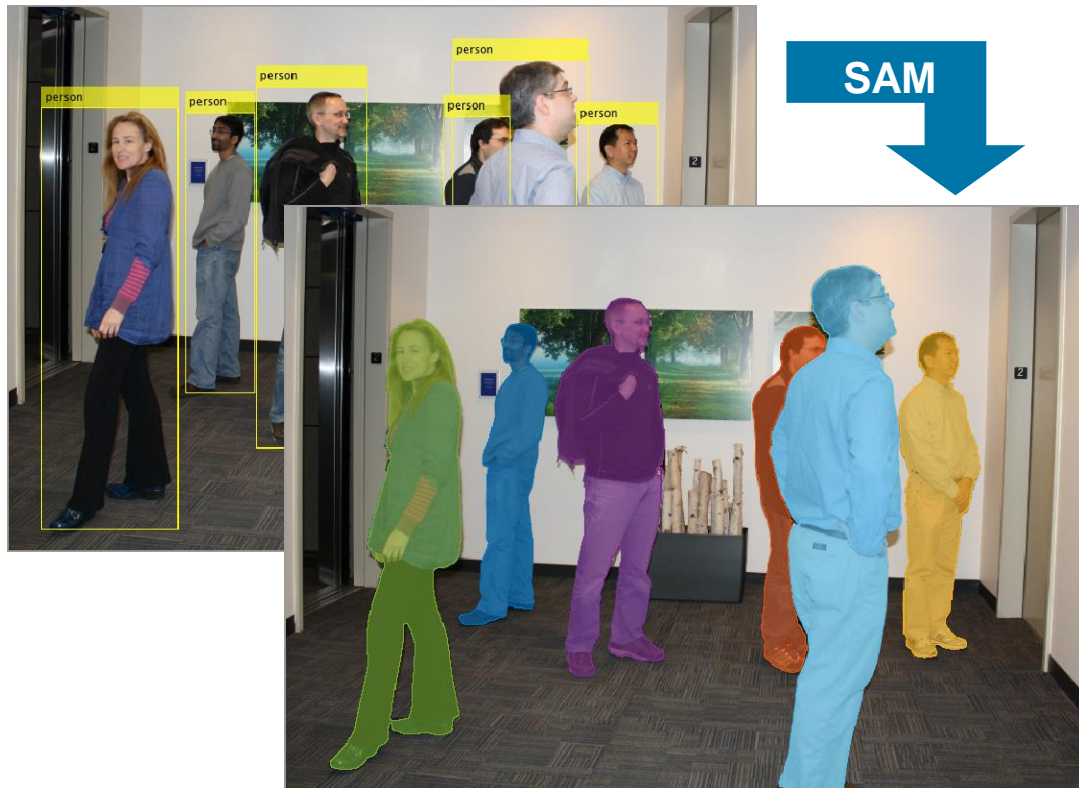
**Demo**

# Segment Anything ModelによるZero-shotセグメンテーション

Image Processing Toolbox  
Deep Learning Toolbox

- クエリの自動化

YOLOで検出したボックスをクエリに



画像処理で粗く推定したマスクの重心をクエリに



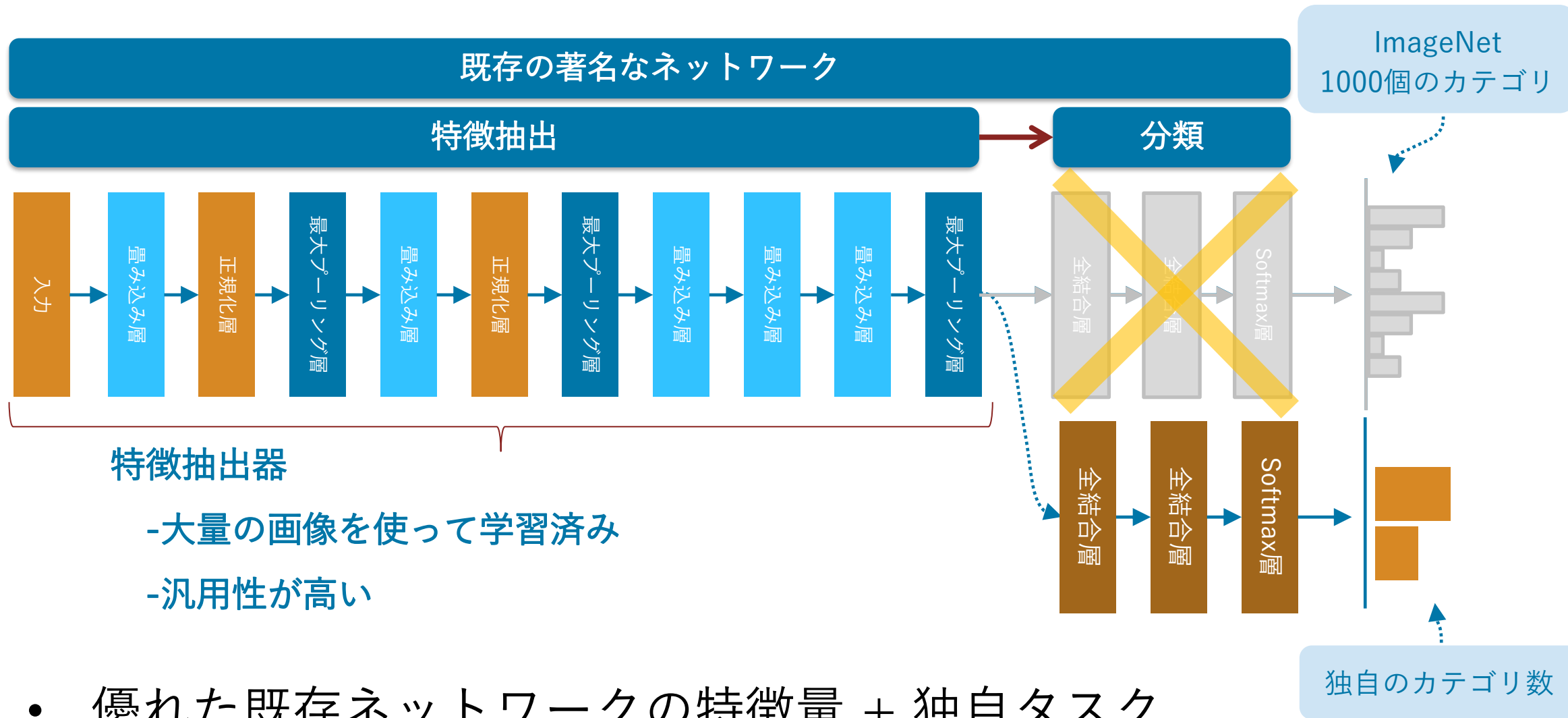
# アジェンダ

- 画像処理の基礎
- 画像処理による画像のセグメンテーション
- 転移学習による画像の分類
- 点群処理による個々の樹木のセグメンテーション

# ディープラーニングを利用する2つの方法

- 学習済みネットワークの利用
- 転移学習
- スクラッチから学習

# 転移学習

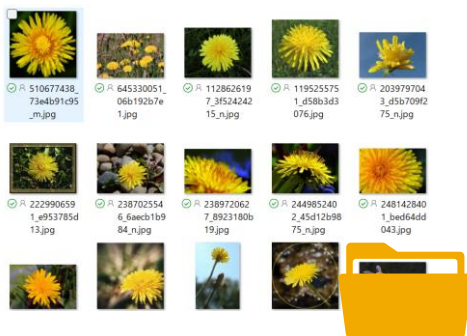


- 優れた既存ネットワークの特徴量 + 独自タスク
- 学習時間・データ収集コストの低減

# ひまわりとタンポポの分類

## 転移学習の例

画像の準備 /  
ラベリング

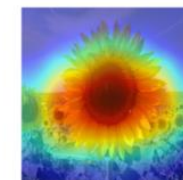
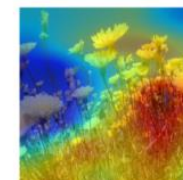
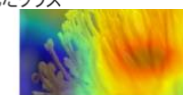


ネットワークの準備 /  
学習



性能評価

	dandelion	9
	sunflowers	9
真のクラス	dandelion	sunflowers
	予測されたクラス	





# 画像認識AI開発の流れ

## 画像の準備 ラベリング



## ネットワークの準備



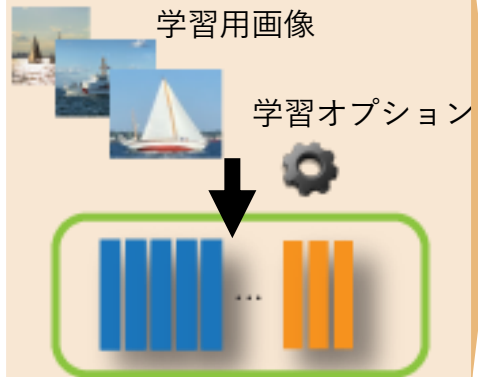
スクラッチからの構築



転移学習



## ネットワークの学習



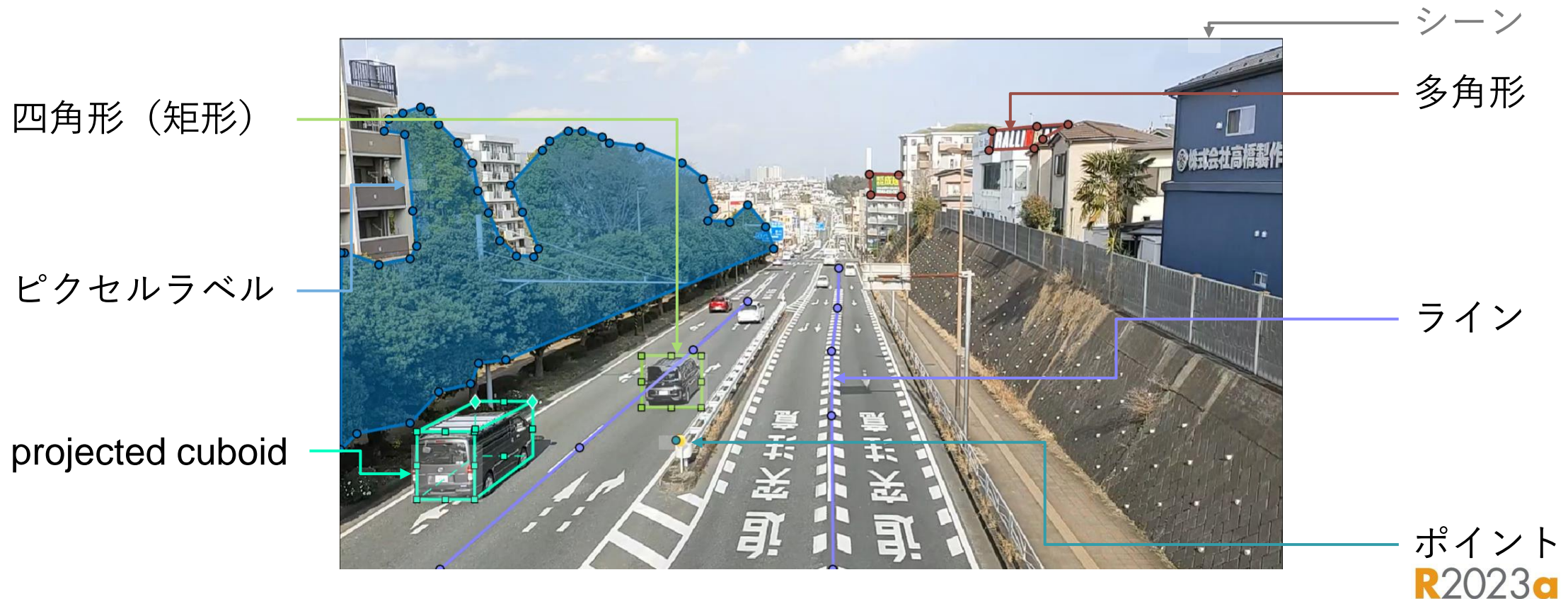
## 予測とネットワークの 性能評価



## システムへの統合



# ラベルタイプ (Image Labeler & Video Labeler)



※各ラベルタイプのサブラベル (子ラベル) も対応

多様なタスク・マルチラベルに対応可能

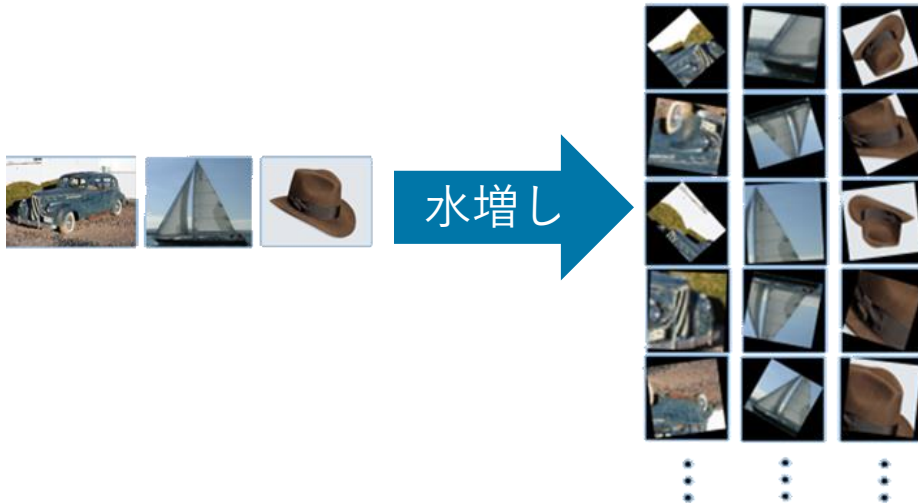
# 画像の疑似的な拡張（水増し）

データ量の不足を補う

## imageDataAugmenter

- 画像データの拡張

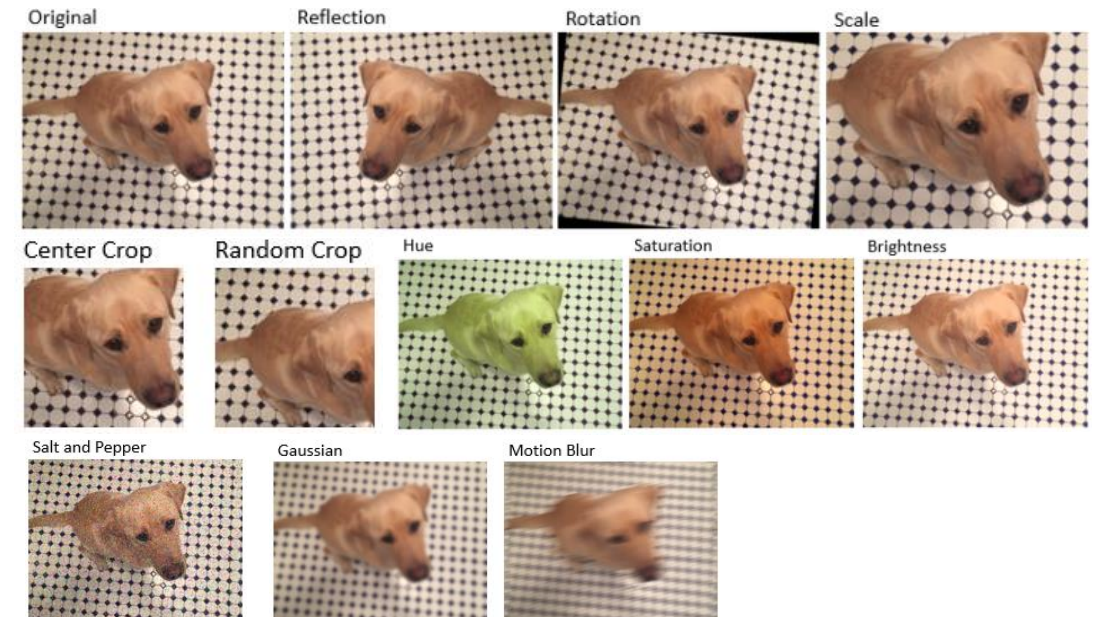
```
augmenter = imageDataAugmenter(...  
    'RandRotation',[0 360], ...  
    'RandScale',[0.5 1])
```



## TransformedDatastore

- 関数により任意の画像変換を定義可能

- アフィン変換（回転、スケール、せん断、平行移動）
- トリミング
- 色の変換
- ノイズ付加
- ブレ、ボケ



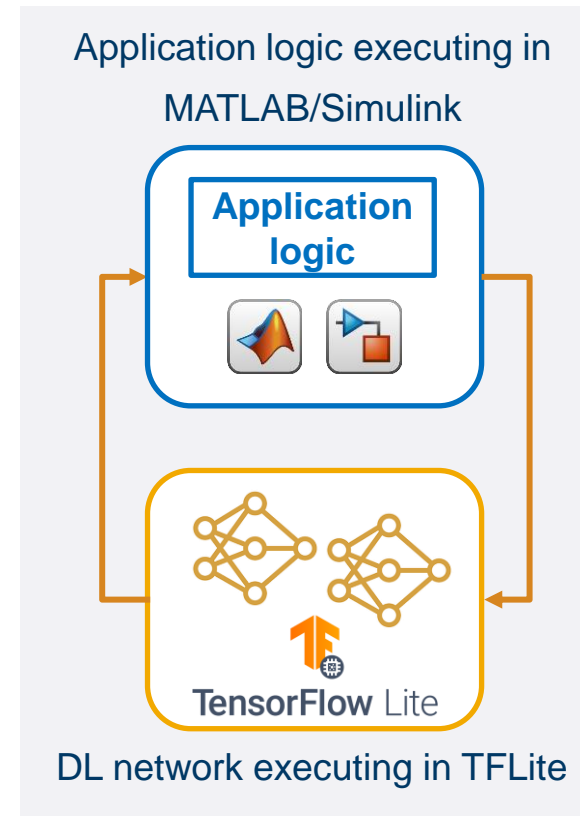
# TensorFlow Lite サポート

## TensorFlow Lite :

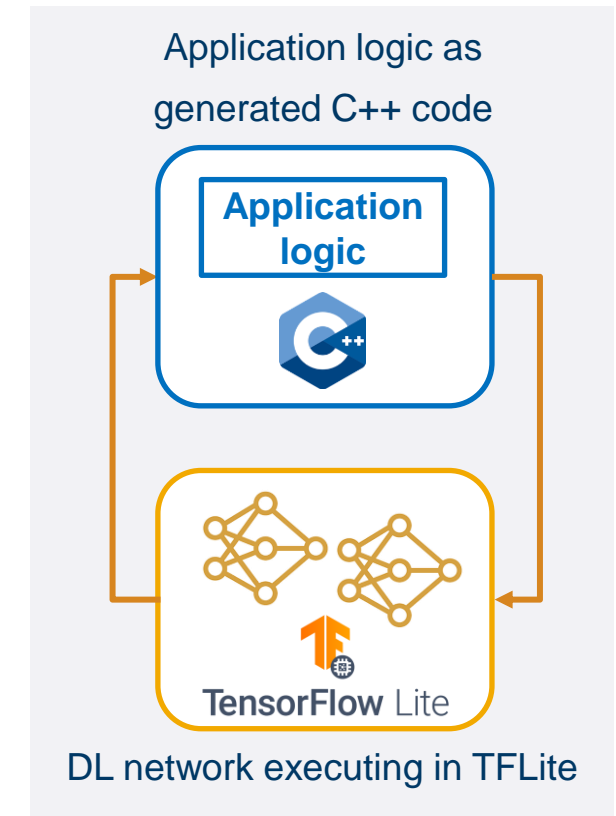
エッジ実装を意識した軽めの学習済みモデル

- 学習済みのモデルを使ったシミュレーション
  - 推論以外はMATLAB
  - MATLAB から TFLite を呼び、DLシミュレーション
- TFLiteモデルの実装
  - MATLAB Coder でTFLiteを含むアプリケーション全体をコード生成
  - MATLAB Coder は推論以外のC++コードを作成
  - TFLite インタープリタが TFLite モデルをターゲット上で呼び出し

## シミュレーション



## コード生成



[Subset of pretrained models](#) on TensorFlow Hub supported

» [Deploy Super Resolution Application That Uses TFLite Model on Host and Raspberry Pi](#)

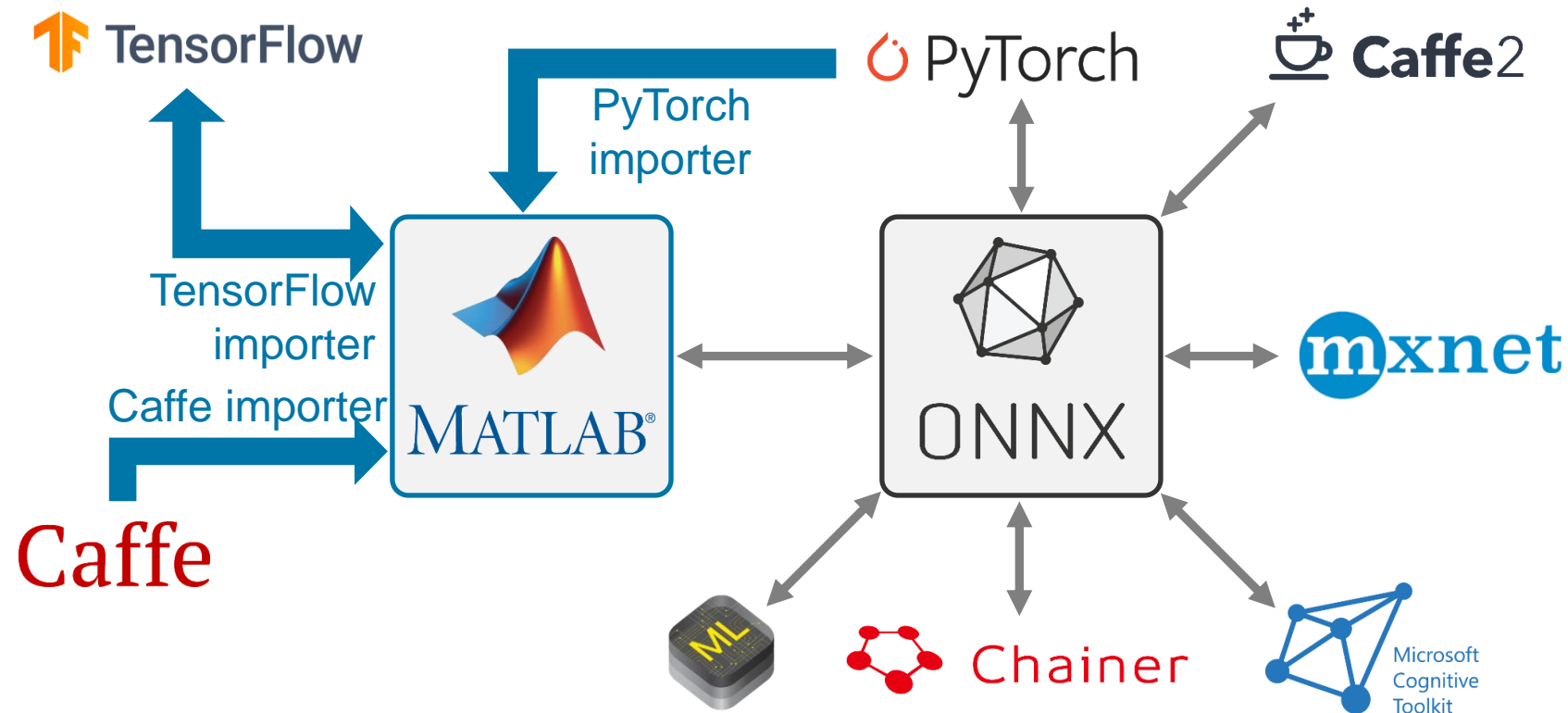


## OSSフレームワークとの連携

- 他フレームワークによる学習済みモデルの取り込み
  - Tensorflow, Caffe, PyTorchモデル
  - ONNXフォーマット

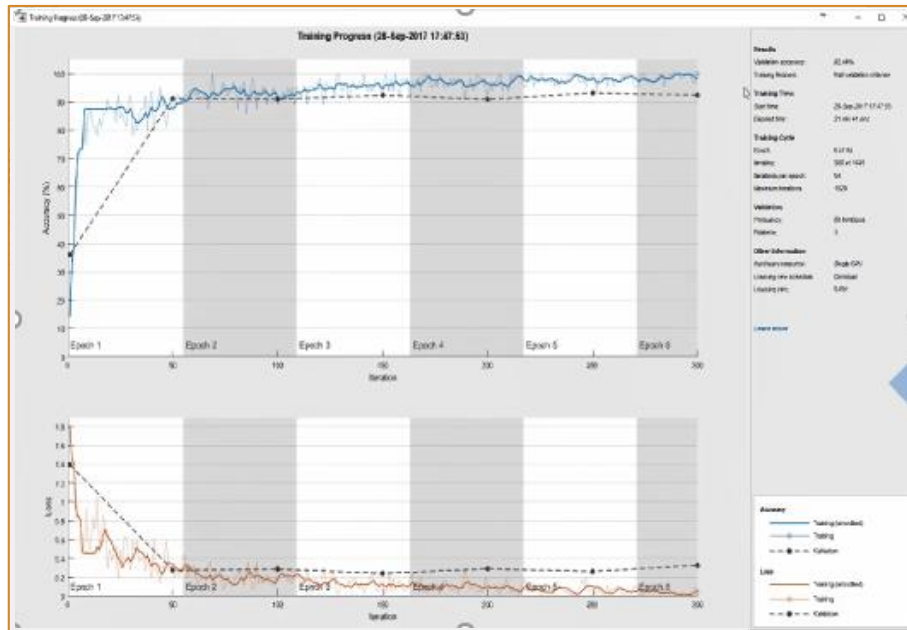
### 他フレームワークとの連携

```
net = importKerasNetwork(modelfile)
net = importNetworkFromPyTorch(modelfile)
net = importONNXNetwork(modelfile, ...
    'OutputLayerType', outputtype)
```



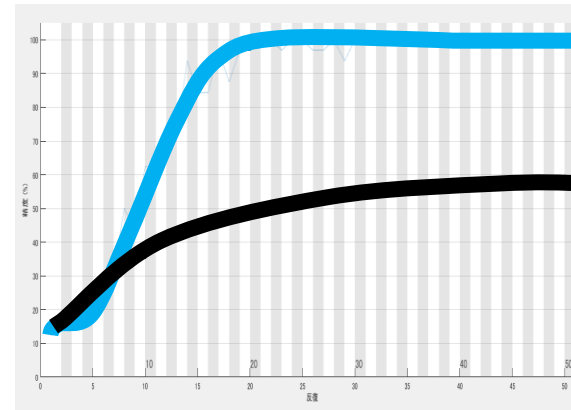
# 学習曲線の可視化

## 学習進捗の可視化と学習停止



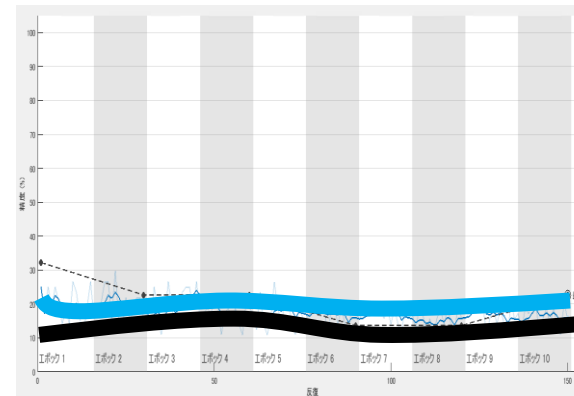
```
opts = trainingOptions(...
    'Plots','training-progress',...
    'ValidationPatience', 3);
```

## 過学習(Overfitting)



- 画像の追加
- 正則化項を大きく
- ユニット数を減らす
- 学習率(調整)

## 未学習(Underfitting)



- ネットワークの変更
- 正則化項を小さく
- ユニット数を増やす
- 学習率(初期値)

学習曲線を見ながら、対策を考えていくことが必要



# ネットワークのチューニングによる最適化

## 実験マネージャー app

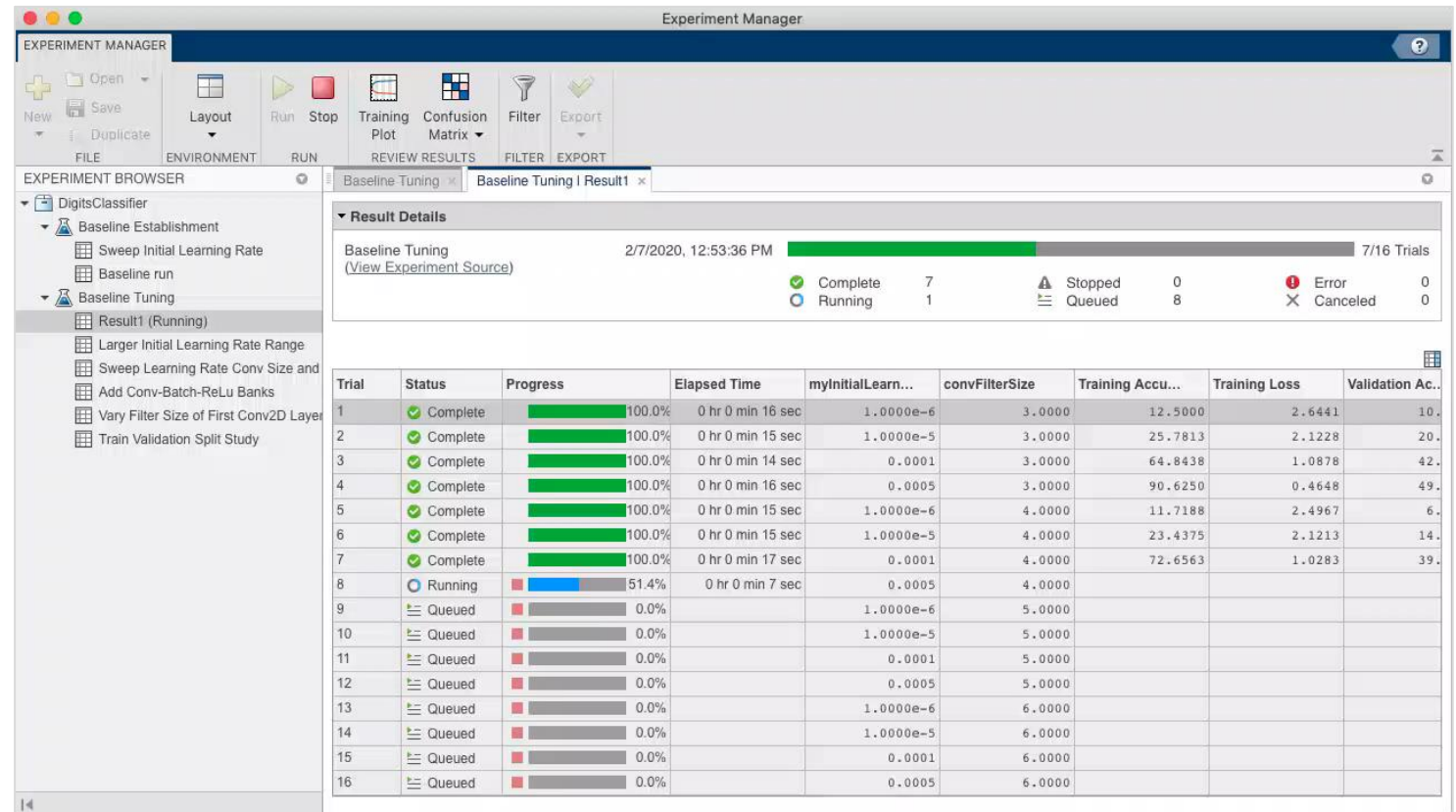
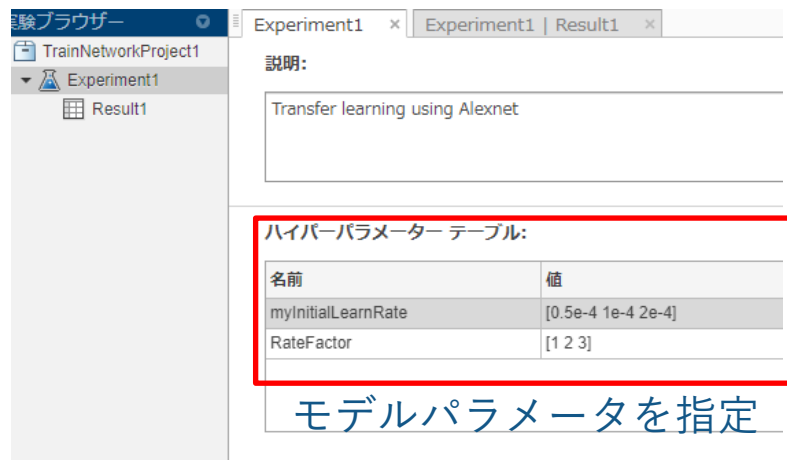
- ✓ 複数の実験を管理
- ✓ カスタムトレーニングに対応

R2021a

- ✓ パラメータの履歴をトラッキング
- ✓ グリッドサーチ or ベイズ最適化
- ✓ 結果を解析 & 比較
- ✓ 並列計算が可能

R2020b

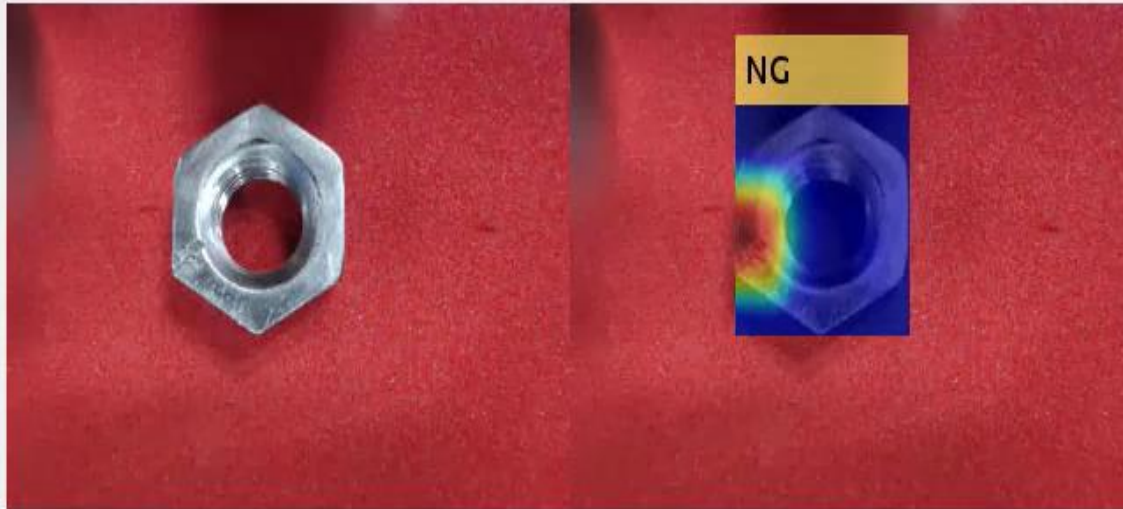
(Parallel Computing Toolboxが必要)



# 説明可能なAIをサポートする機能

元映像

分類とCAM表示

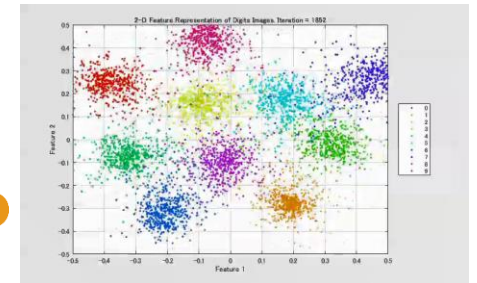


■ OK→ 表面全体に反応  
■ NG→ 局所的についた傷に反応

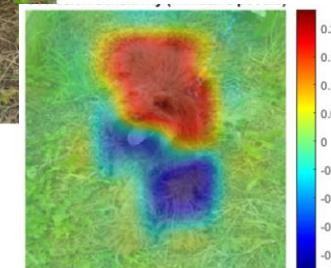
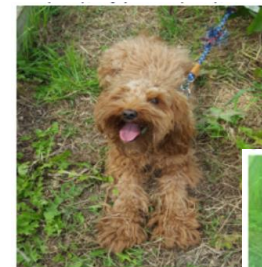
Grad-CAM R2019b



LIME R2020b

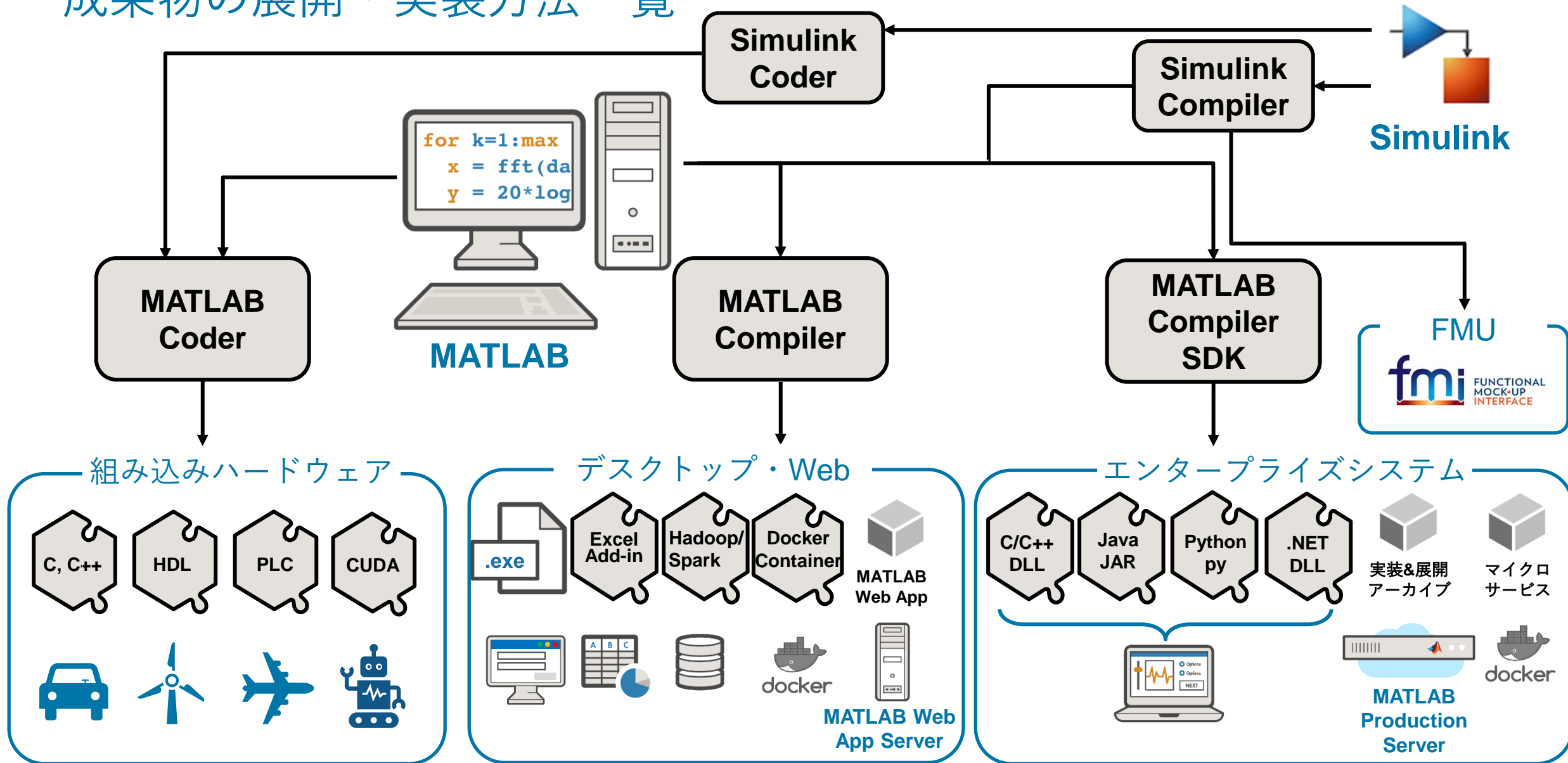


tsneを使った  
特徴ベクトルの可視化



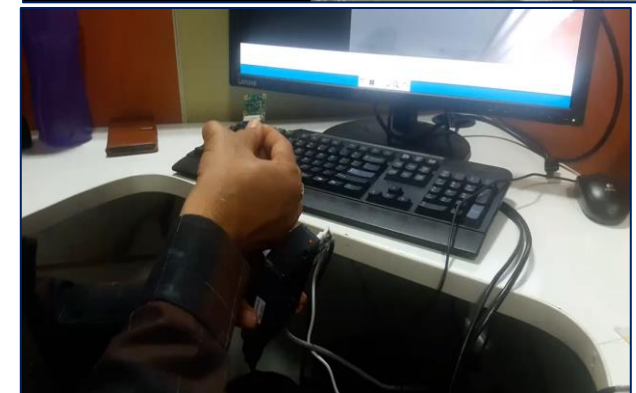
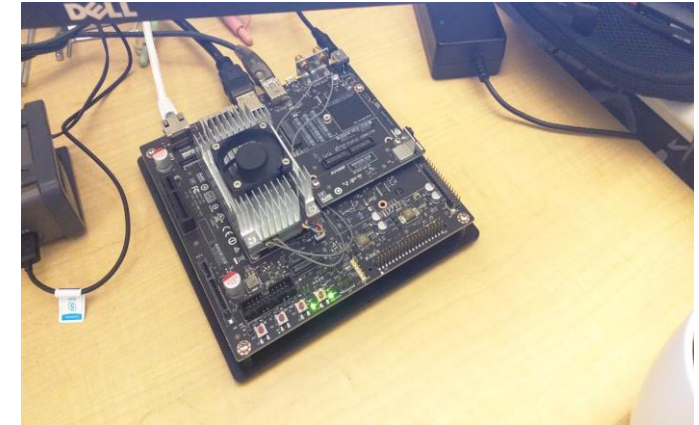
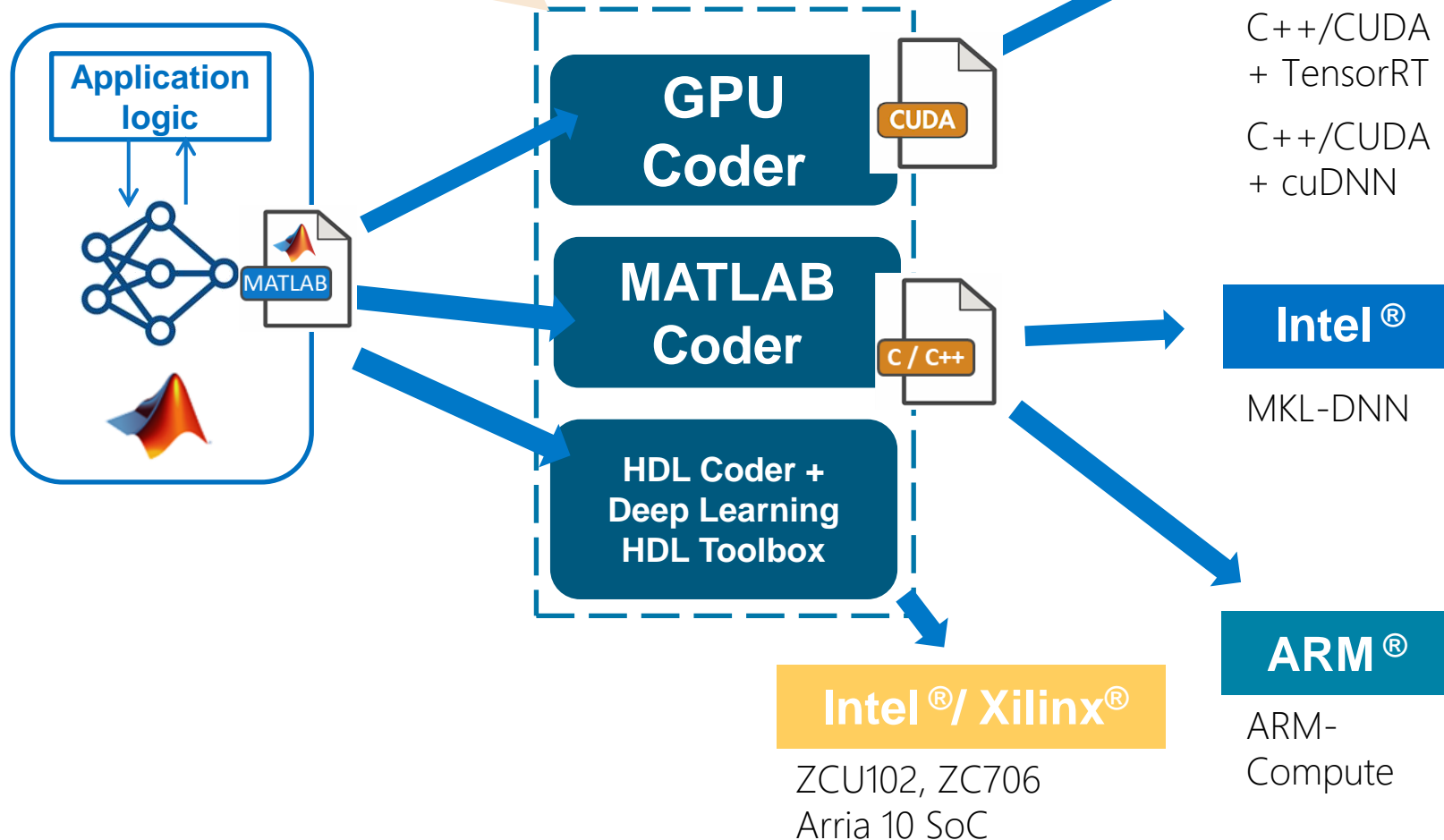
Occlusion sensitivity R2019b

## 成果物の展開・実装方法一覧



# Coderによる実装ソリューション

- 前処理・後処理を含めたコードの自動生成
- 生成されるコードの最適化
- 複数のターゲットへの実装





# 現場にフィットするツールを簡単にカスタマイズ作成可能 AppDesigner

画像フォルダを選択

対象画像を表示

ディープラーニングが予測した結果

判定の正誤表示  
(正→緑 誤→赤)

スコアのグラフ表示

判定時結果

ファイルを選択

Errorimagesというフォルダを作り  
誤判定の画像のみを保存

File List:

- Orange\_13.jpg
- Orange\_14.jpg
- Orange\_15.jpg
- Orange\_16.jpg
- Orange\_2.jpg
- Orange\_3.jpg
- Orange\_4.jpg

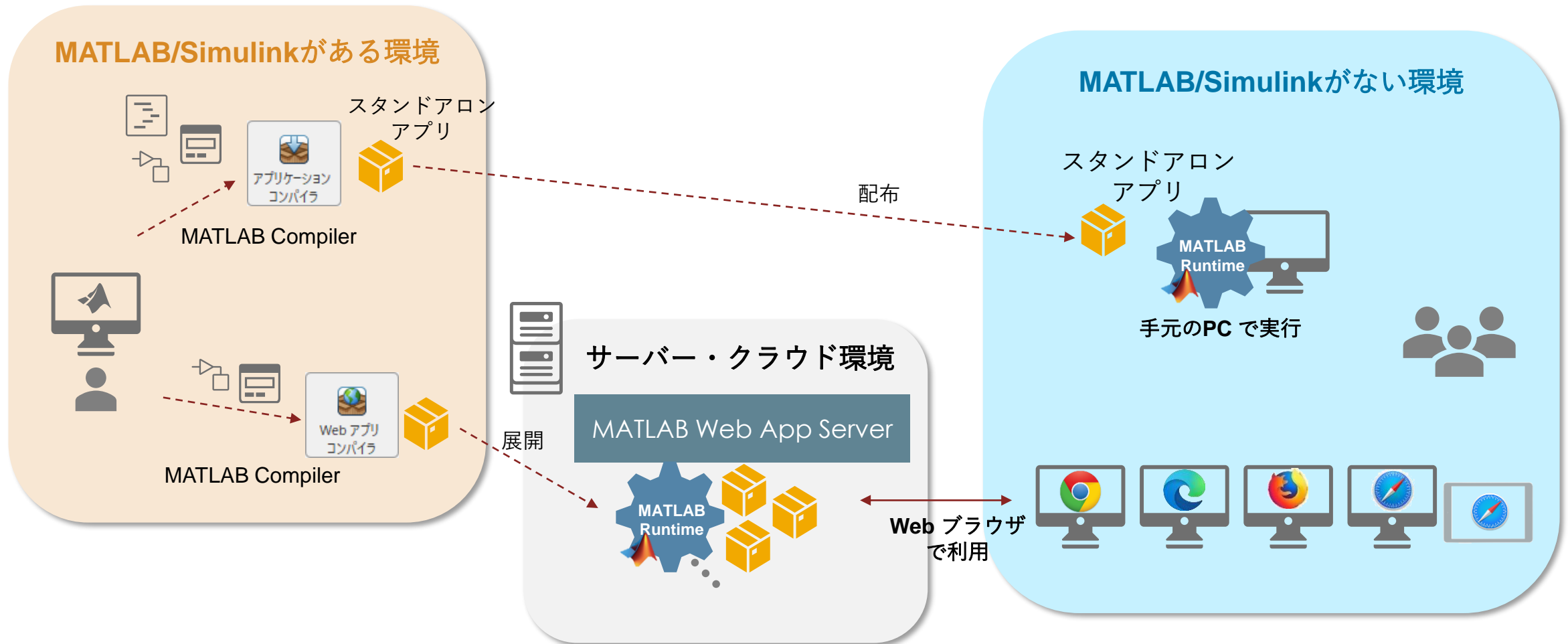
Score Graph Data:

Class	Score
Lemon	0.0
Orange	0.1
grapefruit-ruby	0.0
grapefruit-white	0.0
mandarin-orange	0.9

AppDesignerを使うことで、使われるAIツールの開発を進めることができます！



# MATLAB の環境が無い方へのGUI アプリの共有



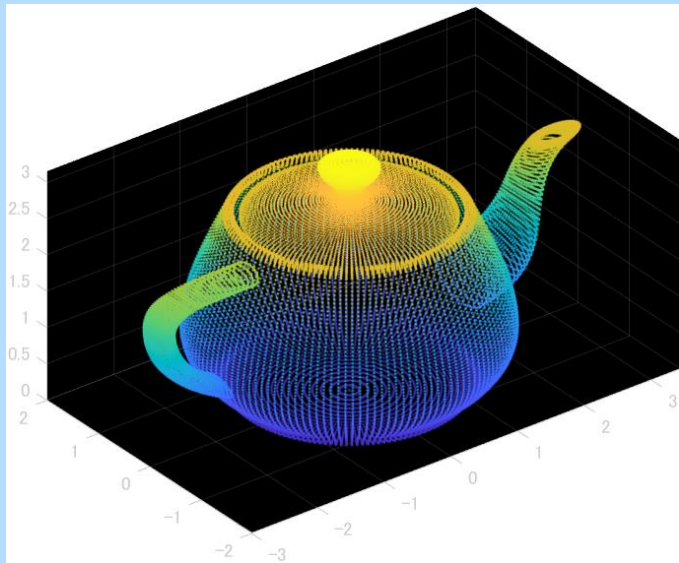
コンパイルして配布・サーバー展開することで、  
MATLAB / Simulink が無い環境でもGUI アプリを利用可能に

# アジェンダ

- 画像処理の基礎
- 画像処理による画像のセグメンテーション
- 転移学習による画像の分類
- 点群処理による個々の樹木のセグメンテーション

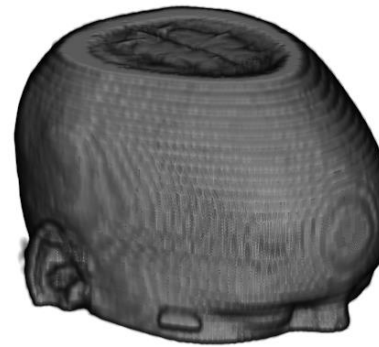
# 点群は様々な3次元データ表現の中の一つ

## 点群



- オブジェクトの表面形状を点の集合として表現
- 各点は3次元座標値を持つ
- 局所的な形状もそのまま表現が可能
- Lidarなどのセンサから直接取得可能

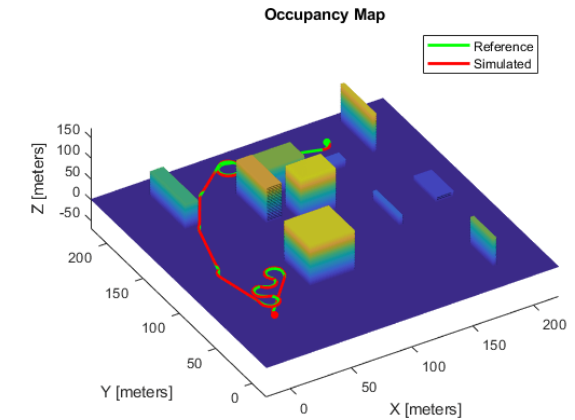
## ボクセル



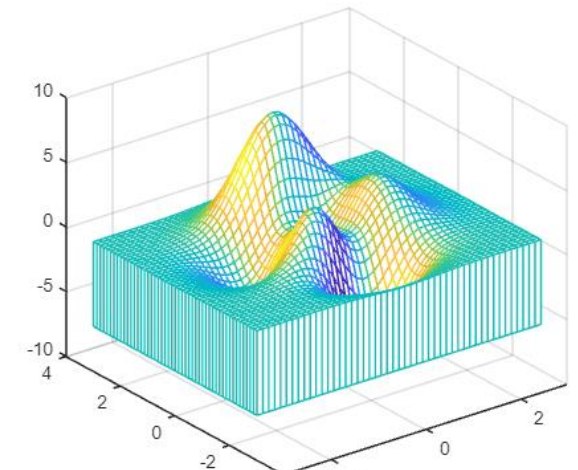
## 深度画像(RGB-D)



## オクトマップ

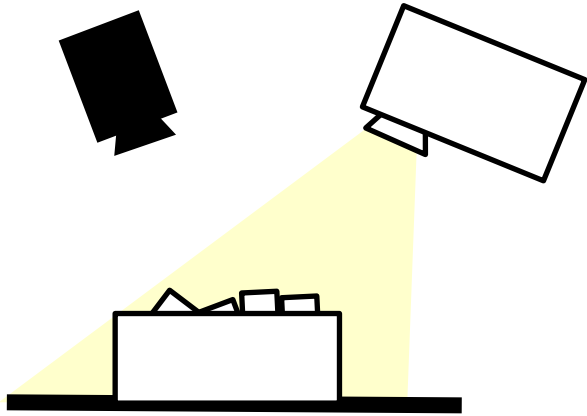


## メッシュ



# 点群を計測できるセンサ

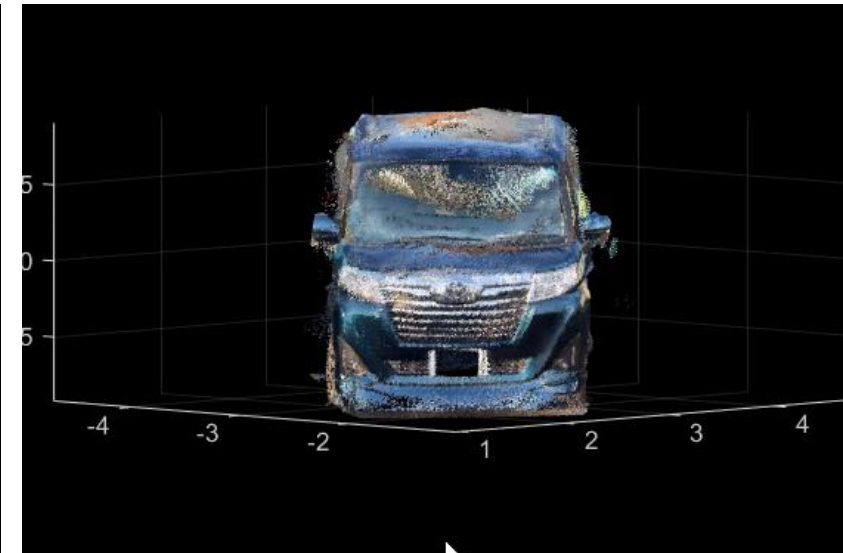
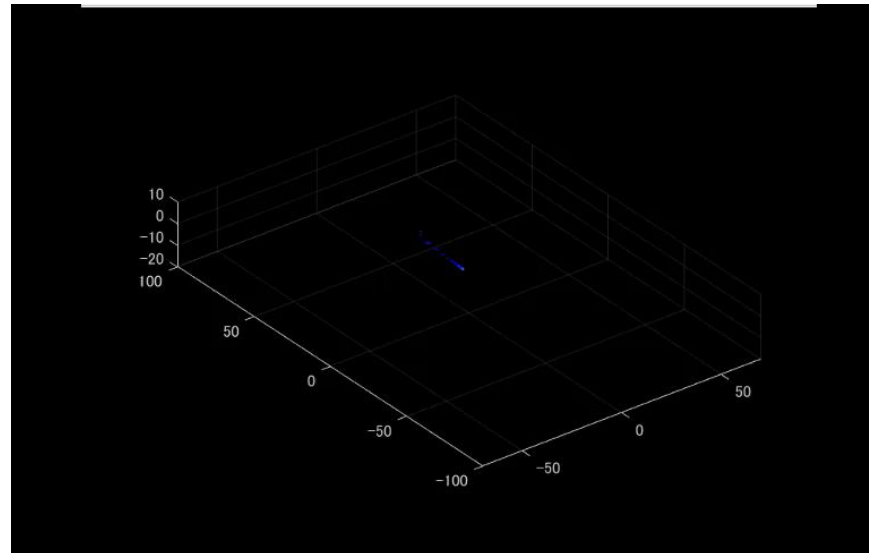
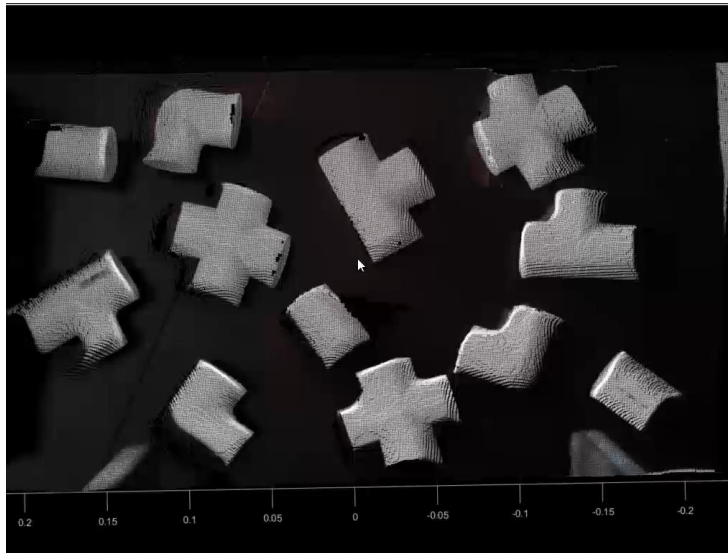
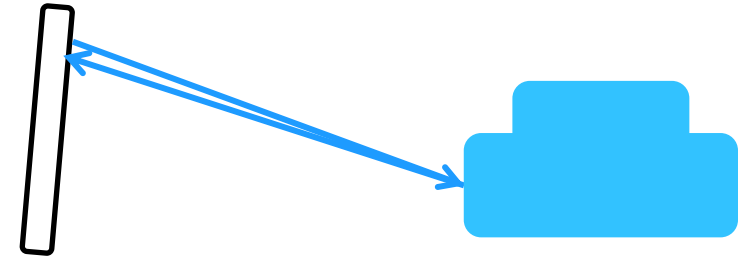
## RGB-D system (camera)



## 回転式LiDAR

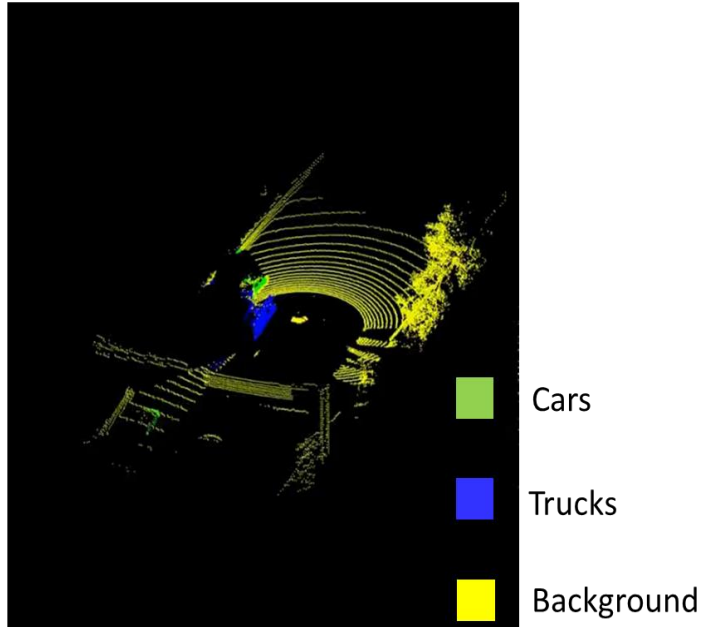


## 非回転式LiDAR (iPhone等)

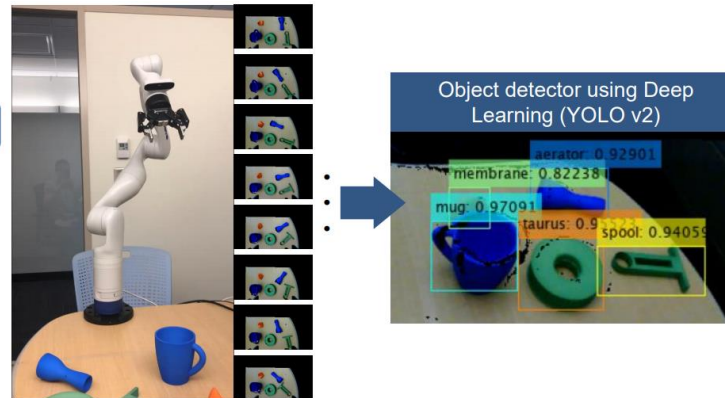


# 点群データの利活用が進んでいます！

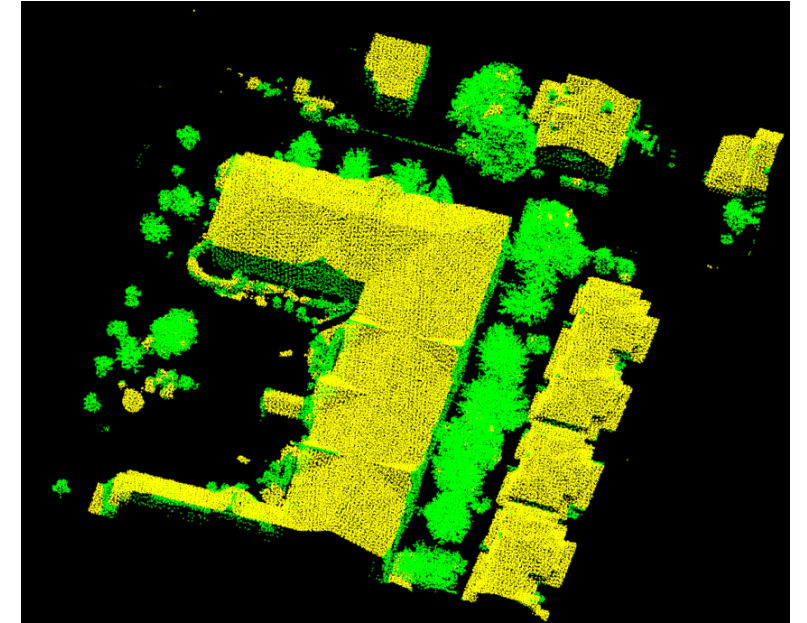
## 自動運転・移動ロボット



## ロボットアーム



## 航空 (UAV) 測量

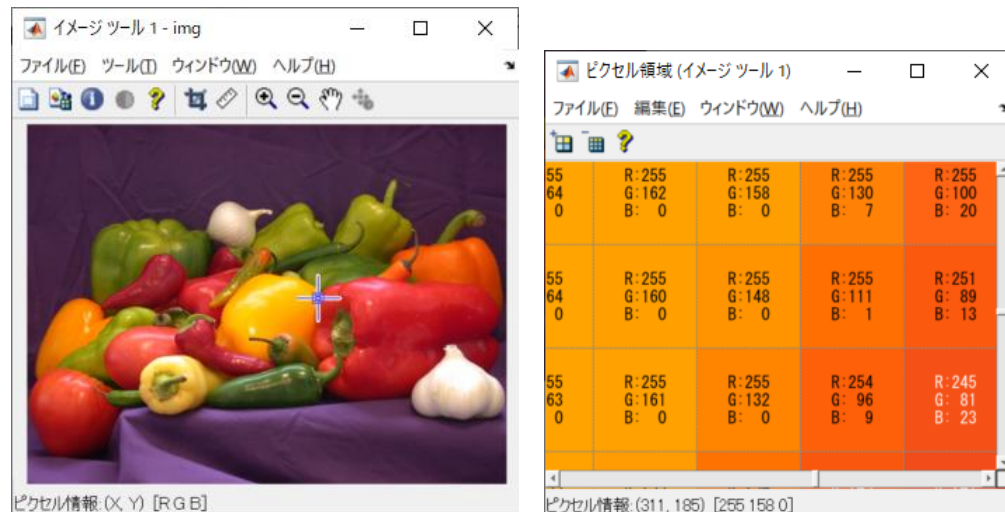




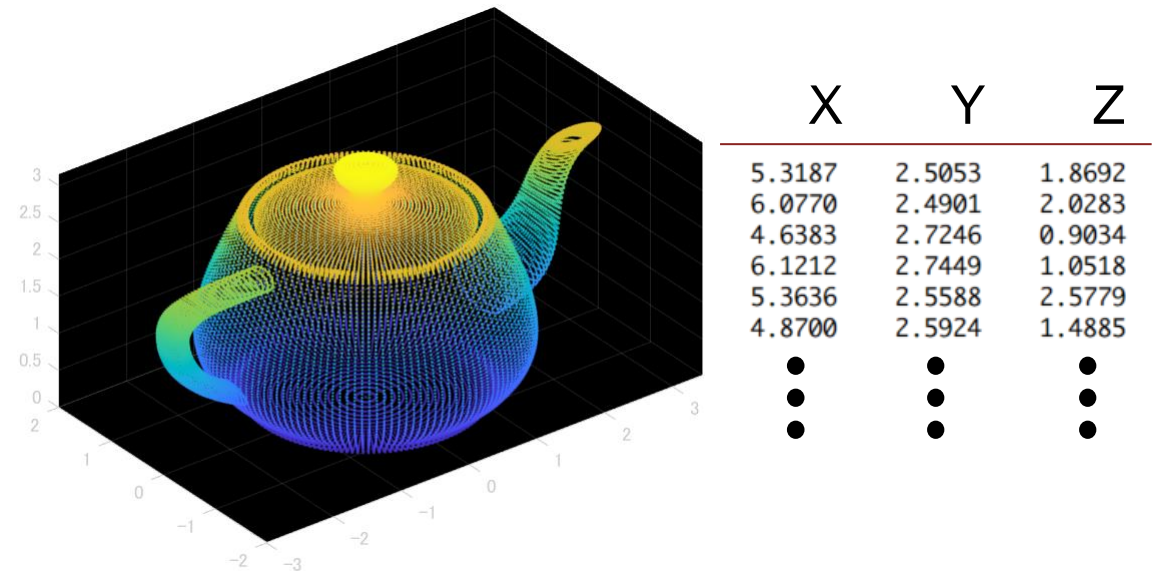
# 点群データの特徴（画像データと比較して）

	空間の次元	位置の規則性
画像データ	2次元	有（グリッド状）
点群データ	3次元	無（物体の形、密度による）

画像

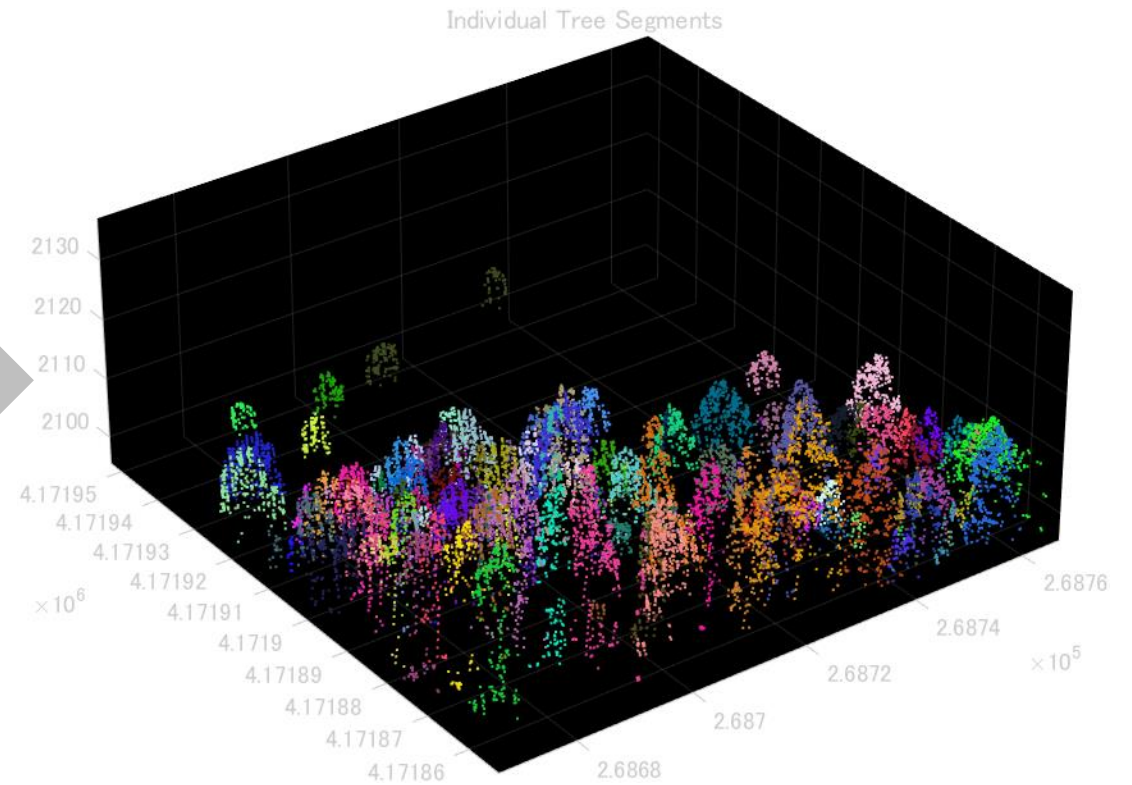
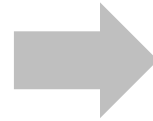
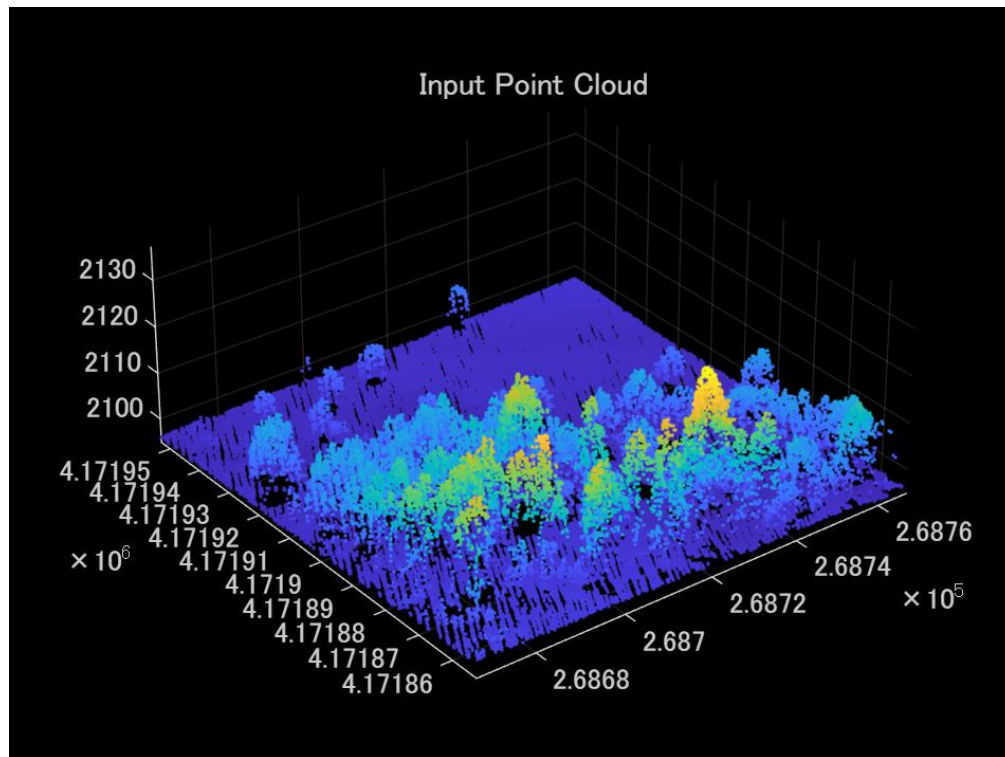


点群



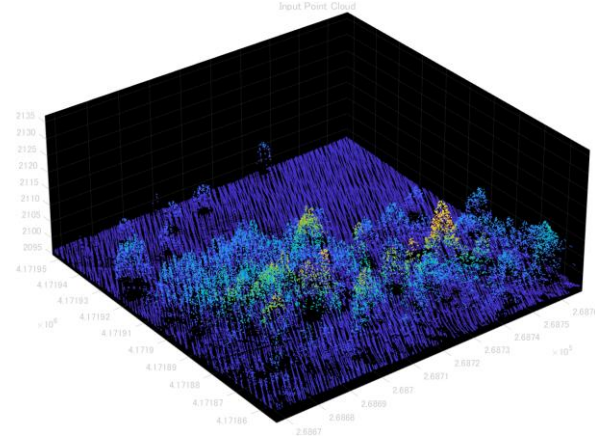
点群はXYZの各座標が保存(N点ある場合は $N \times 3$ の行列)

# 樹木のセグメンテーション

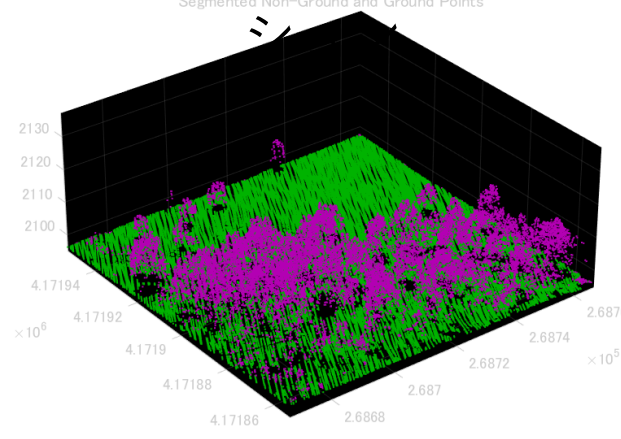


# 手順

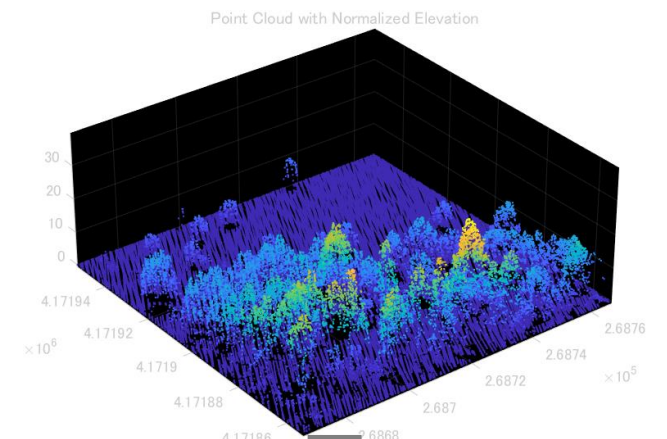
## 元の点群



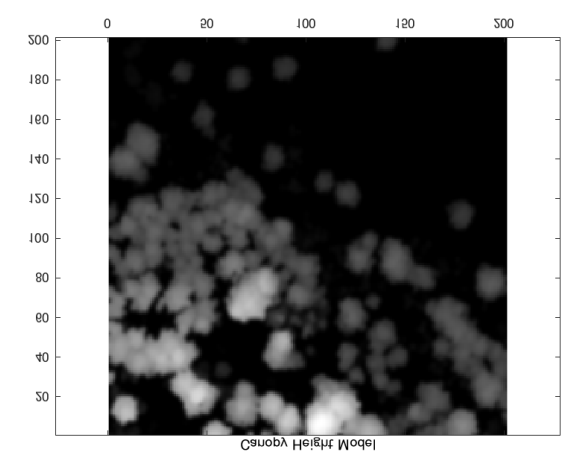
## 地面のセグメンター



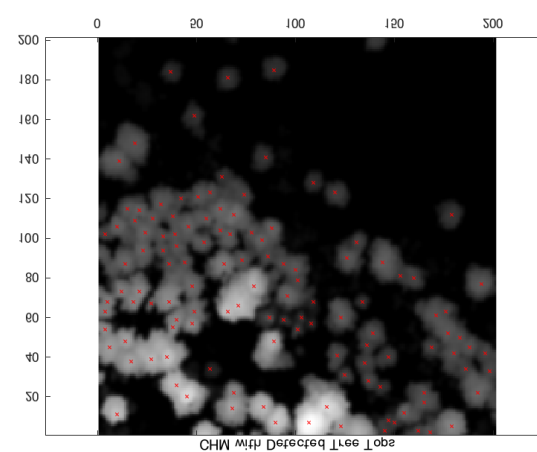
## 標高の正規化



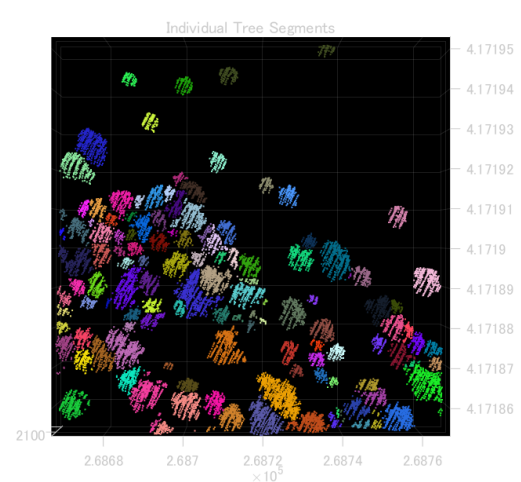
## 画像化



## 頂点検出



## セグメンテーション



# pointCloud オブジェクト: 点群データコンテナ

pointCloud オブジェクト: 点群データのためのコンテナオブジェクト

```
ptCloudObj =
```

```
pointCloud with properties:
```

```
Location: [32×1084×3 single]
```

```
Color: []
```

```
Normal: []
```

```
Intensity: [32×1084 single]
```

```
Count: 34688
```

```
XLimits: [-80.0677 93.4626]
```

```
YLimits: [-82.5912 84.3181]
```

```
ZLimits: [-8.7008 10.6333]
```

[x,y,z] 座標  
色データ

反射強度データ

pointCloud オブジェクトのメソッド

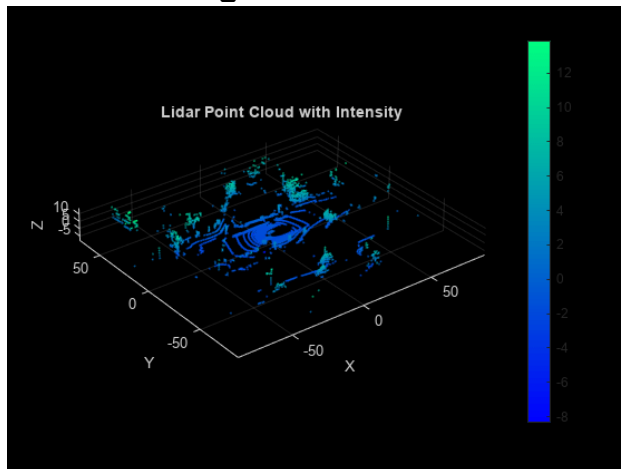
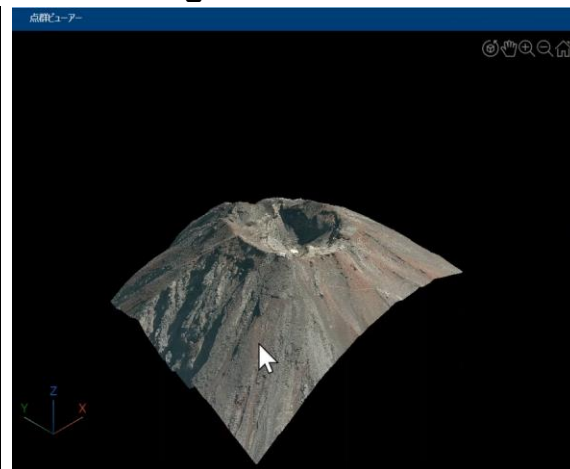
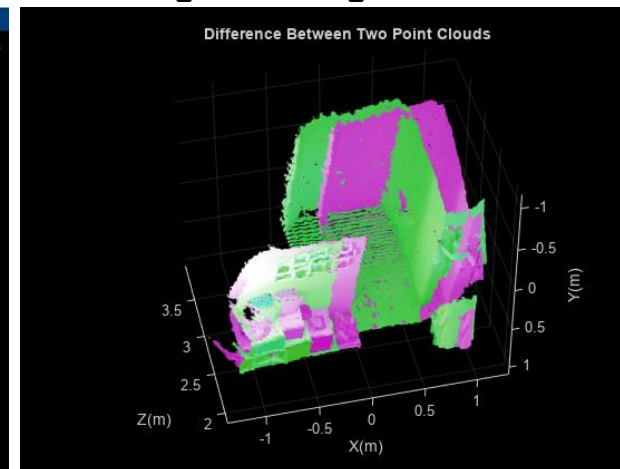
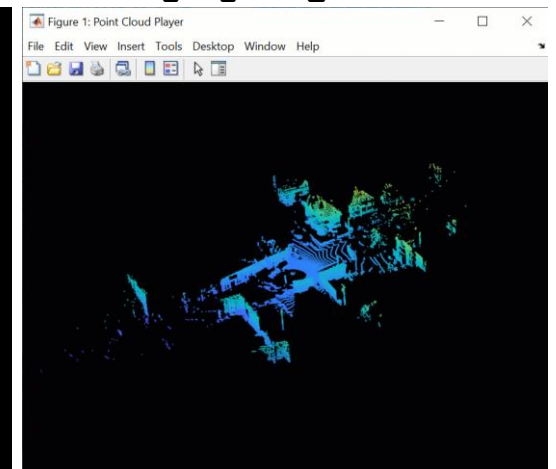
<a href="#">findNearestNeighbors</a>	最近傍の点群の選択
<a href="#">findNeighborsInRadius</a>	指定点のある半径内の点群の選択
<a href="#">findPointsInROI</a>	直方体ROI内の点群の選択
<a href="#">findPointsInCylinder</a>	円筒形ROI内の点群を選択
<a href="#">removeInvalidPoints</a>	無効な点群の除去
<a href="#">select</a>	指定の点群の抽出

ドキュメンテーション  
[Point Cloud Object](#)



# 可視化のための様々なツール

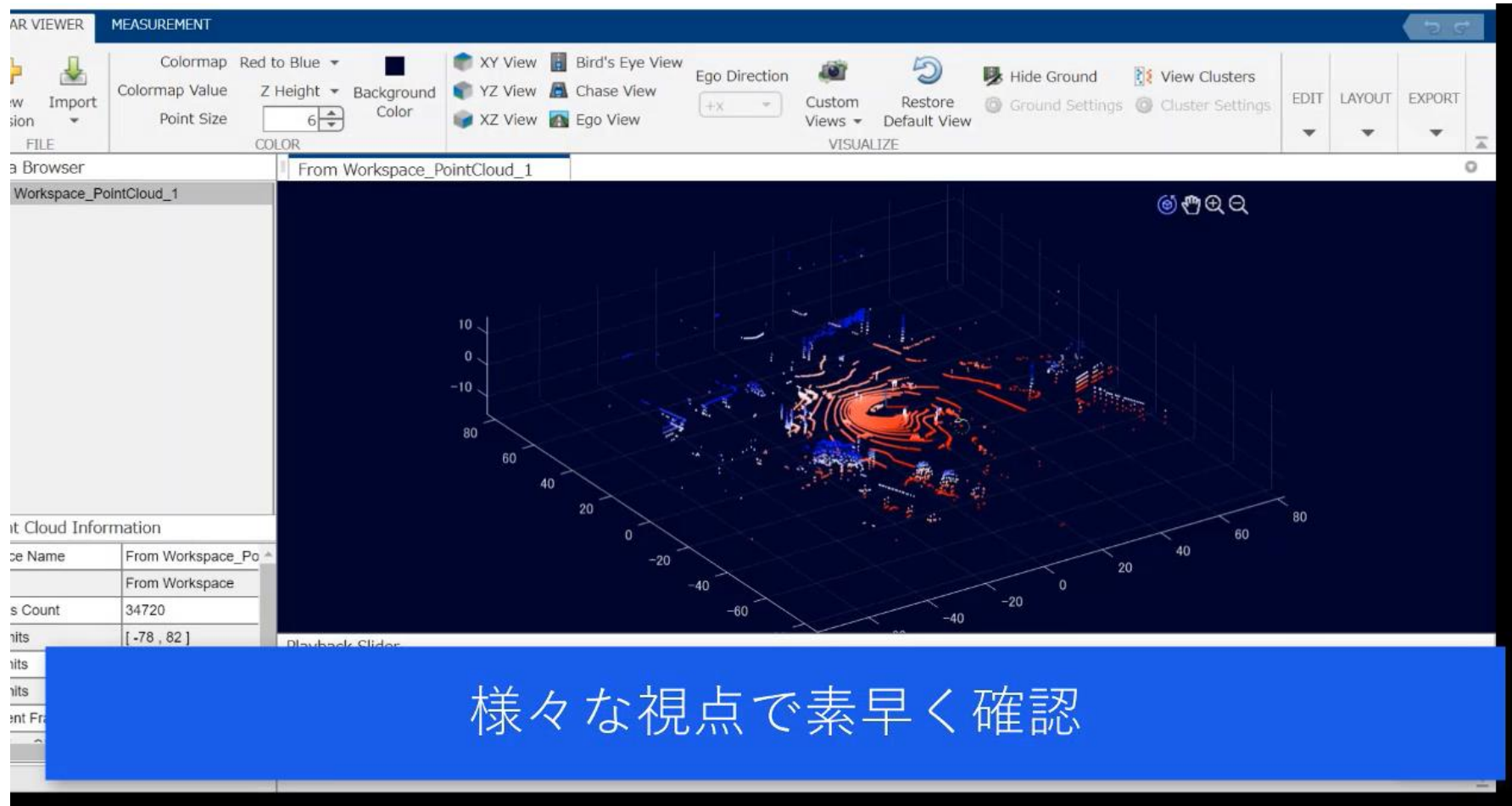
関数名	機能
<b>pcshow</b>	点群表示のための標準ツール
<b>pcviewer</b>	大規模点群（処理が重い）をスムーズに表示するためのツール
<b>pcshowpair</b>	2セットの点群の比較ツール
<b>pcplayer</b>	点群の動画を表示するためのツール

**pcshow****pcviewer****pcshowpair****pcplayer**



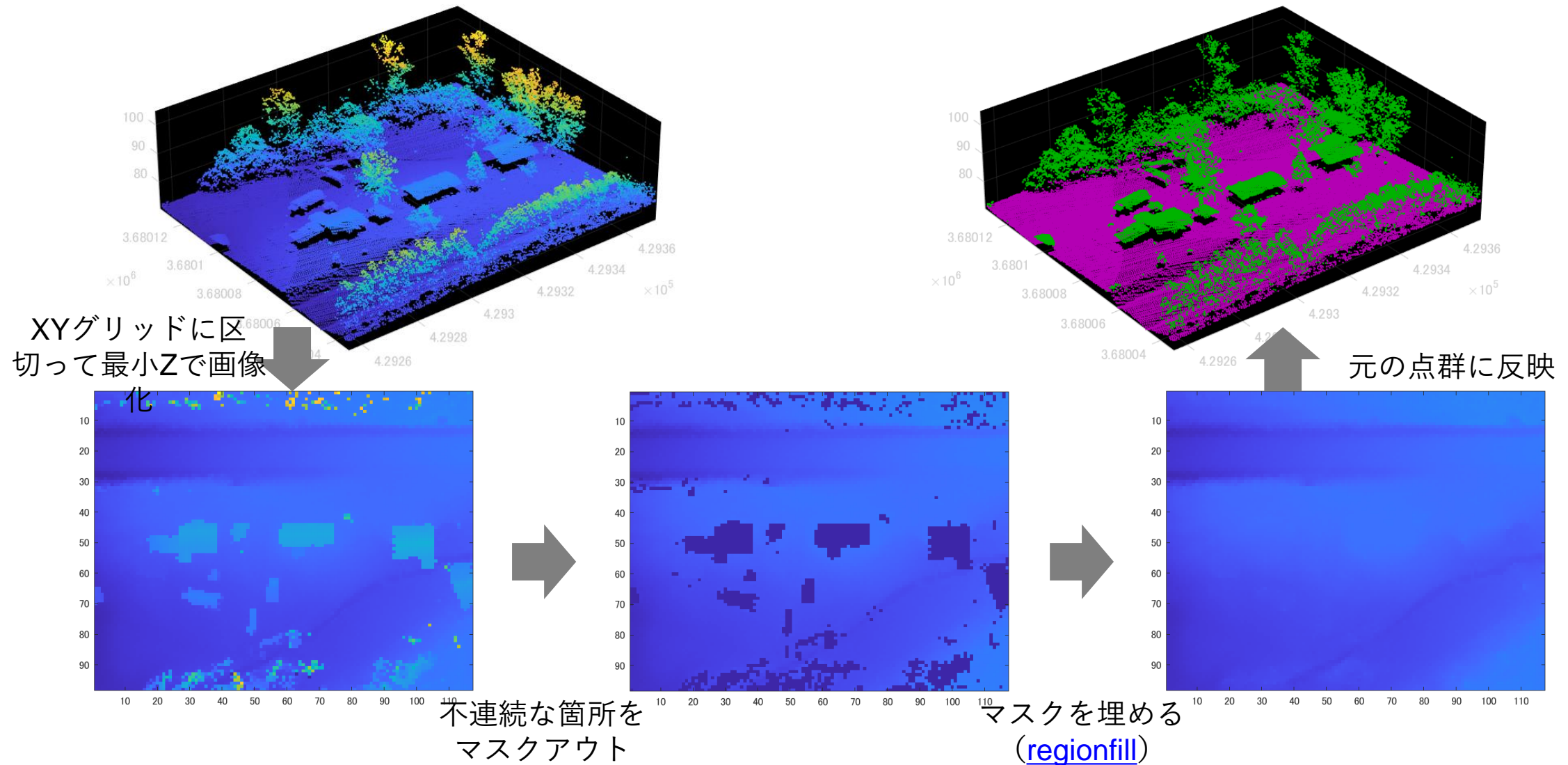
# Lidar Viewer App

点群の可視化、前処理、切り抜き、解析



# 地面のセグメンテーション

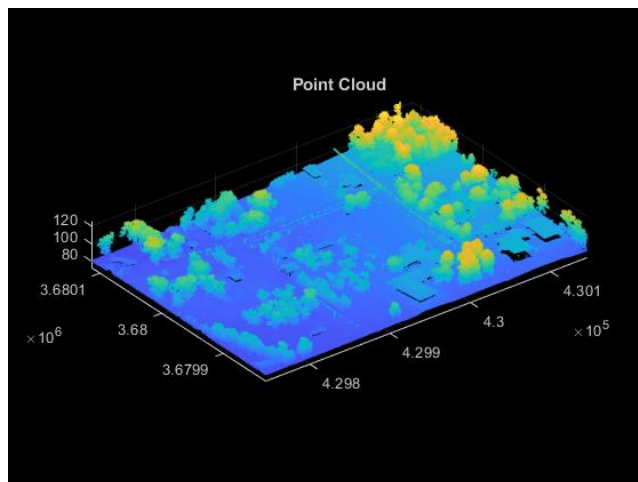
## SMRFのアプローチ



# Digital Elevation Model (デジタル標高モデル)

## 点群データのデジタル標高モデル (DEM) を作成

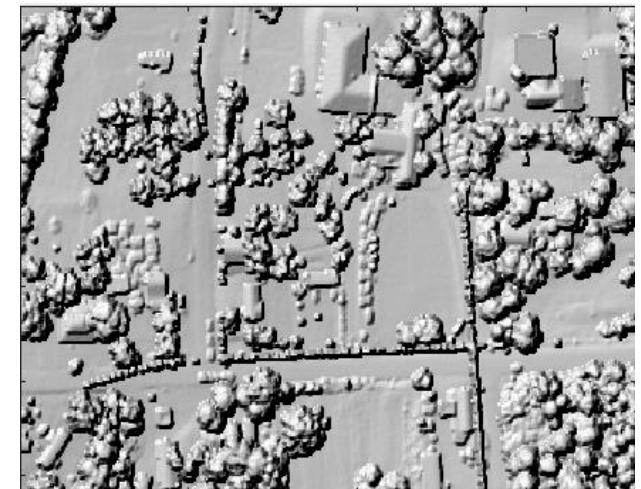
- デジタル標高モデル: 地表面形状を表す
- デジタル表面モデル: 木や建物などを考慮した表面形状
- アプリケーション: 地形のモデリング、地形解析、フライトシミュレーション



入力する点群

pc2dem

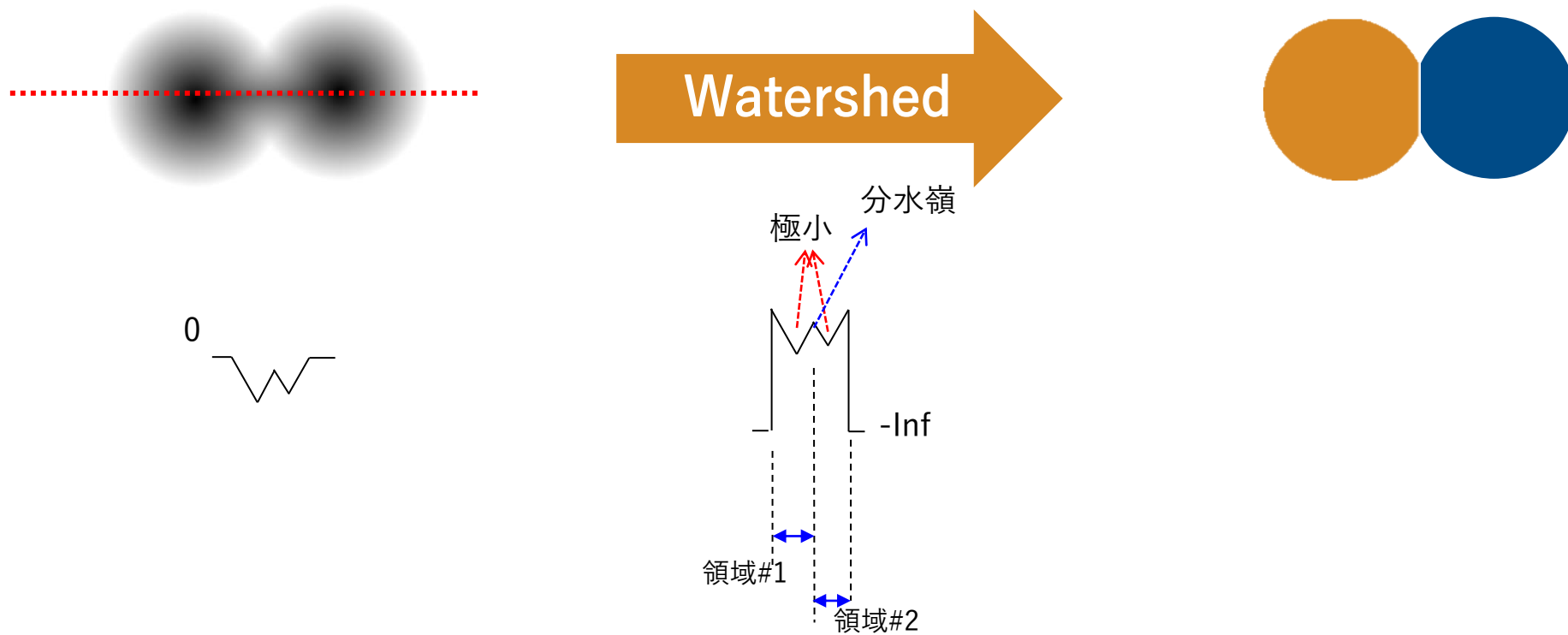
visualize



DEM

# Watershed (分水嶺) セグメンテーション

領域の境で分割



# 点群のDeep Learningによるセグメンテーション例

## RandLA-Net ネットワーク用API

### % 学習と推論

```
>> segmenter = randlanet("dales", classnames);
```

“dales”: 航空LiDAR用データセット  
“pandaset”: 自動運転用データセット

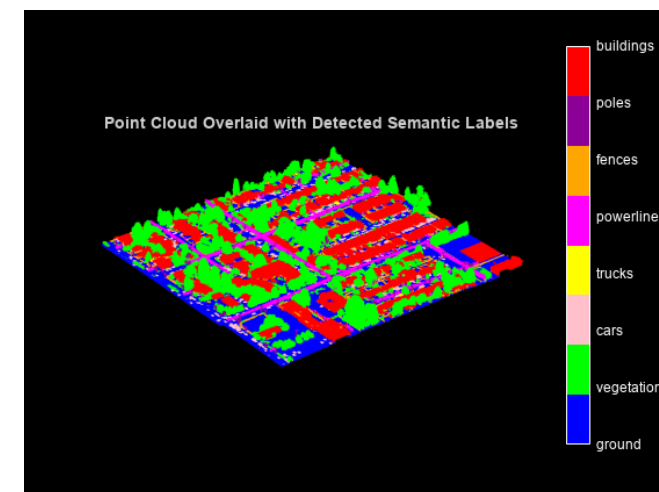
これらの事前学習パラメータを使った転移学習が可能

### % ネットワークの学習

```
>> net = trainRandlanet(datastore, segmenter, options);
```

### % ネットワークのを使った推論

```
>> labels = semanticObjects(segmenter, ptCloud);
```



セグメンテーション  
結果



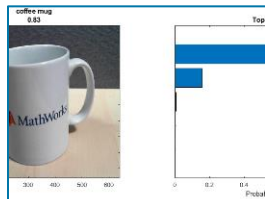
## まとめ

- 関数やアプリを使ったノーコード・ローコードで、コーディングの手間を削減
- AIを利用したロバストなアルゴリズム開発
- 点群処理などドメインをまたいだ開発も可能

MATLABで画像処理・点群処理の研究・開発を加速

# 豊富な例題からトライできます

## 分類



### 深層学習を使用した Web カメライメージの分類

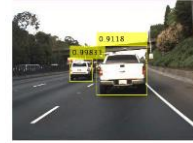
この例では、事前学習済みの深層畳み込みニューラル ネットワーク GoogLeNet を使用して、Web カメラのイメージをリアルタイムで分類



### 深層学習を使用した複数ラベル イメージ分類

この例では、複数ラベル イメージ分類用に、転移学習を使用して深層学習モデルに学習させる方法を説明します。

## 検出



### YOLO v4 深層学習を使用したオブジェクトの検出

この例では、You Only Look Once version 4 (YOLO v4) 深層学習ネットワークを使用して、イメージ内のオブジェクトを検出する方法を説明



### 深層学習を使用した Simulink® での車線検出と車両検出

この例では、Simulink® モデル内で深層畳み込みニューラル ネットワークを使用して、車線検出と車両検出を実行する方法を示します。こ

## セマンティックセグメンテーション



### 深層学習を使用したセマンティック セグメンテーション

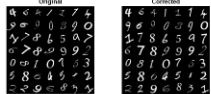
この例では、セマンティック セグメンテーション ネットワークを使用してイメージをセグメント化する方法を説明します。



### Perform Instance Segmentation Using Mask R-CNN

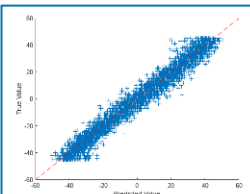
Segment individual instances of people and cars using a multiclass mask region-based convolutional neural network (R-CNN).

## 回帰



### 分類ネットワークの回帰ネットワークへの変換

この例では、学習済み分類ネットワークを回帰ネットワークに変換する方法を説明します。



### 回帰用の畳み込みニューラルネットワークの学習

この例では、畳み込みニューラルネットワークを使用して回帰モデルを当てはめ、手書きの数字の回転角度を予測する方法を示します。

## 画像生成



### 条件付き敵対的生成ネットワーク (CGAN) の学習

この例では、条件付き敵対的生成ネットワークに学習させてイメージを生成する方法を説明します。



### 深層学習を使用したニューラル スタイル転送

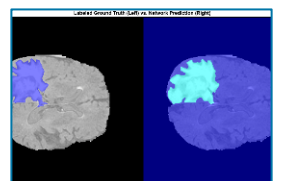
この例では、事前学習済みの VGG-19 ネットワークを使用して、あるイメージのスタイル外観を別のイメージのシーン コンテンツに適用す

## 骨格検出など



### 深層学習を使用した体の姿勢の推定

OpenPose アルゴリズムを使用し、1 人以上の人物の体の姿勢の推定。



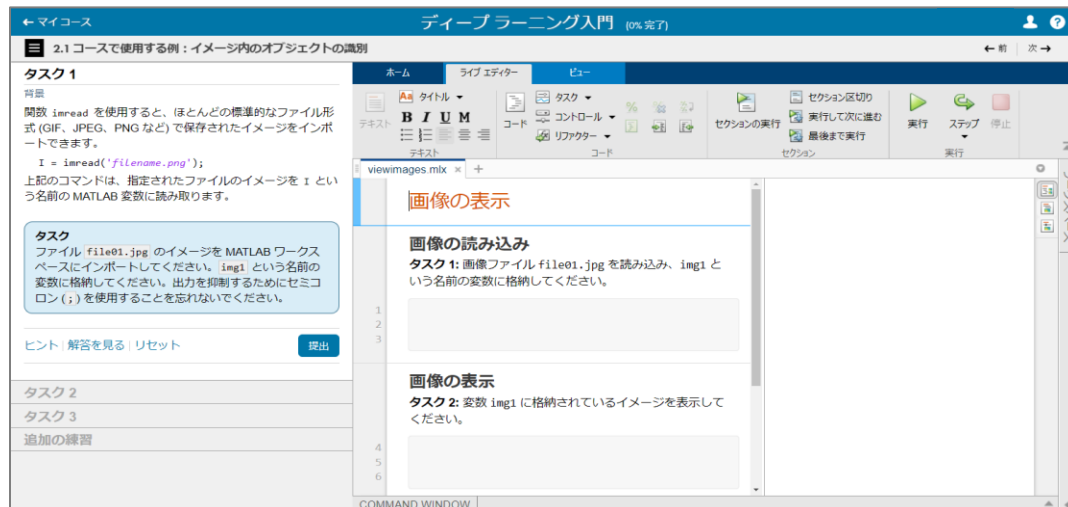
### 深層学習を使用した脳腫瘍の 3 次元セグメンテーション

この例では、3 次元医用画像から脳腫瘍のセマンティック セグメンテーションを実行する方法を説明します。

# 自己学習形式：無料オンライントレーニング

## MathWorks製品のスタートポイント

- 全23コースを無料で受講可能
  - ライセンス不要！
  - 製品インストール不要！
- 各約2時間程度
- 好きな時に開始、中断、再開
- ゲーム感覚



**MATLAB 入門**  
 2 時間 | 英語  
 最短で MATLAB の基礎を学習します。

**Simulink 入門**  
 2 時間 | 英語  
 最短で Simulink の基礎を学習します。

**機械学習入門**  
 2 時間 | 英語  
 分類問題の実用的な機械学習手法の基礎を学習します。

**電気回路シミュレーション入門**  
 2 時間 | 英語  
 Simscape で電気回路をシミュレーションするための基礎を学習します。

**ディープラーニング入門**  
 1 時間 | 英語  
 最短でディープラーニングを使用した画像認識技術を学習します。

**Simscape Battery Onramp**  
 1 時間 | 英語  
 Learn the basics of simulating battery systems in Simscape.

**Stateflow 入門**  
 2 時間 | 英語  
 Stateflow でのステートマシンの作成、編集、およびシミュレーションの基礎を学習します。

**統計入門**  
 1.5 時間 | 英語  
 MATLAB で統計解析の基礎を学習します。

**強化学習入門**  
 2.5 時間 | 英語  
 経験から学習するインテリジェントコントローラーの作成の基礎を学習します。

**System Composer Onramp**  
 1 時間 | 英語  
 Learn how to perform model-based systems engineering by creating simulatable architecture models using System Composer.

**Power Systems Simulation Onramp**  
 1.5 時間 | 英語  
 Learn how to progressively build and validate power systems using Simscape Electrical.

**画像処理入門**  
 2 時間 | 英語  
 MATLAB で実用的な画像処理の基礎を学習します。

**Simscape 入門**  
 1.5 時間 | 英語  
 Simscape における物理システムのシミュレーションの基礎を学習します。

**パワーエレクトロニクスシミュレーション入門**  
 1 時間 | 英語  
 Simscape でパワーエレクトロニクスコンバーターをシミュレーションするための基礎を学習します。

**最適化入門**  
 1 時間 | 英語  
 MATLAB で最適化問題を解くための基礎を、問題解決型のアプローチで学習します。

**曲線近似入門**  
 1.5 時間 | 英語  
 曲線フィッターアプリを使用した曲線近似の基礎を学習します。

**MATLAB Coder Onramp**  
 1 時間 | 英語  
 Learn the basics of C code generation from MATLAB functions.

**アプリ開発入門**  
 1 時間 | 英語  
 App Designer を使用して MATLAB で動作する対話型アプリを効率的に開発する方法を学習します。

**オブジェクト指向プログラミング入門**  
 2 時間 | 英語  
 MATLAB でオブジェクト指向プログラミングを使用して、実世界のオブジェクトをモデル化し、ソフトウェアの複雑度を管理するための基礎を学習します。

**Simulink による制御設計入門**  
 1 時間 | 英語  
 最短で Simulink のフィードバック制御設計の基礎を学習します。

**コンピュータービジョン入門**  
 2 時間 | 英語  
 コンピュータービジョンの基本および、オブジェクトの検出と追跡について学習します。

**無線通信入門**  
 1 時間 | 英語  
 MATLAB で無線通信リンクをシミュレーションするための基礎を学習します。

**信号処理入門**  
 1 時間 | 英語  
 スペクトル解析に関する低号処理技術を対話形式で学習できます。



© 2024 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://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.