

ROBO-ONE Conference第二弾

<ロボットの自律化>

- ROBO-剣/ROBO-ONE AutoへのMATLAB活用法 -

MathWorks: Hiroumi Mita

2020/05/31 15:15-16:00

目次

- 1. 導入
- 2. MATLABの環境構築
- 3. 情報リソース
- 4. サンプルの説明

1. 導入

- MathWorks社は、Rono-One に参加されるプレイヤーの皆さんに
- MATLABプロダクト体験版を御貸出ししています。

- 今回は、MATLABプロダクト体験版を使うときに
- 役に立つであろう知見や
- 画像処理とディープラーニングのサンプル等をご紹介します。

1. 導入

MathWorks 世界の拠点

MathWorks のお客様は、7 大陸 (南極大陸を含む)、175 を超える国々の 200 万人以上の技術職の方です。これらのお客様は、最先端技術を誇る企業や政府の研究機関、金融機関、3,500 校を超える大学で、MathWorks の製品をご利用になっています。このように MathWorks が広く信頼されているのは、MATLAB と Simulink が科学界および産業界における標準となっているからです。

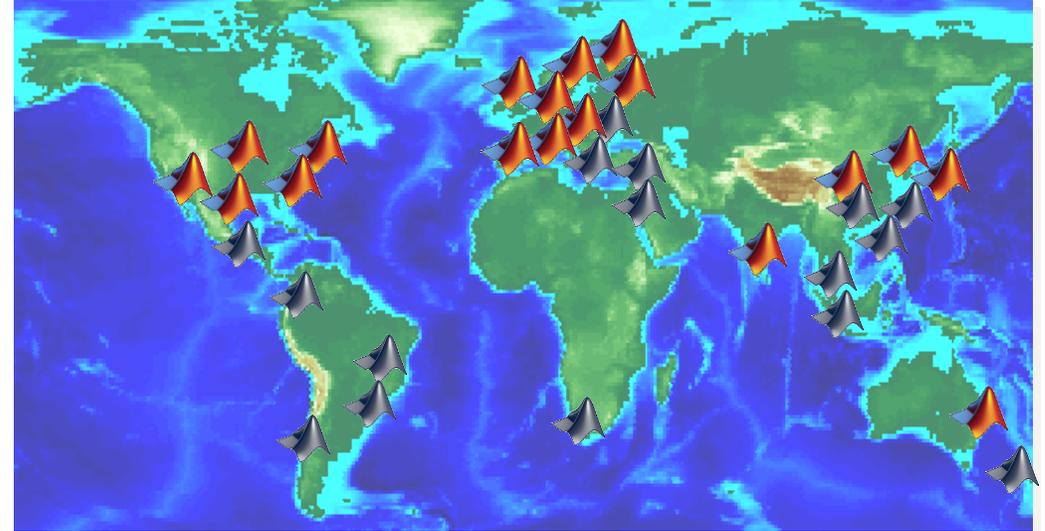
MathWorks は、世界中にある支社と代理店、販売業者のネットワークを通じてお客様をサポートしています。

本社 (米国マサチューセッツ州、ネイティック)
508-647-7000
www.mathworks.com
support@mathworks.com

世界の拠点

各国拠点の詳細を見るには、下記 URL の MathWorks Web サイトにアクセスし、該当する国を選択してください。

<http://www.mathworks.com/company/worldwide/>



この地図は、MATLAB と Mapping Toolbox™ を使用して地球を正距円筒図法で表したものです。

1. 導入

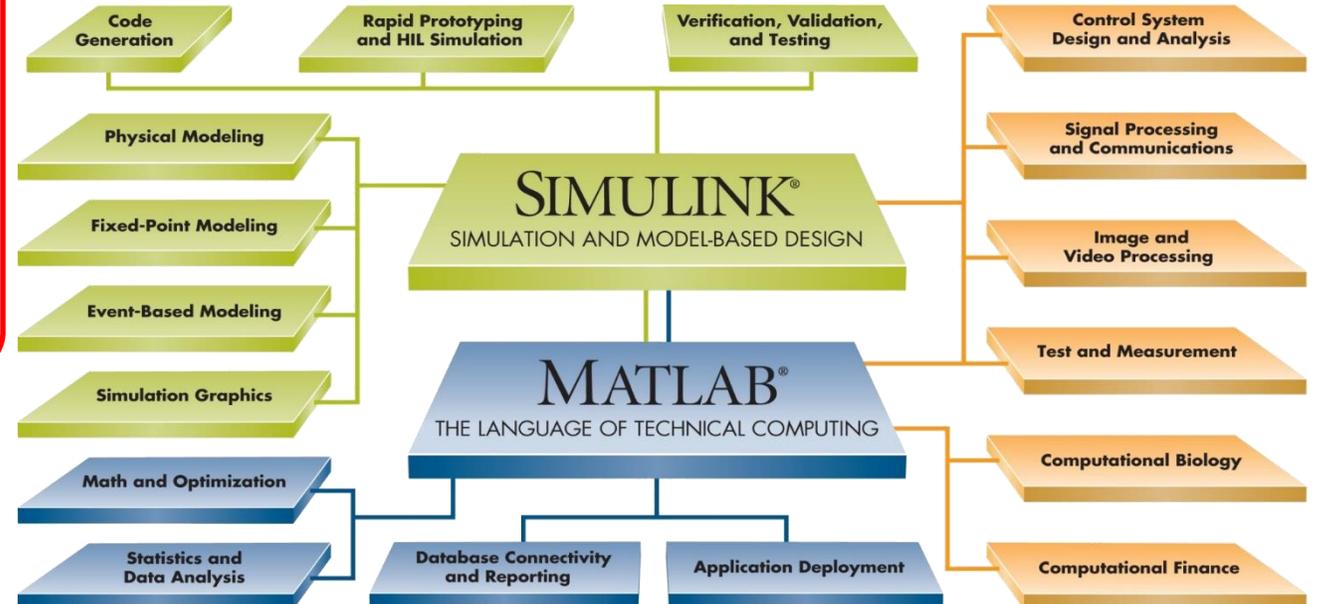
MathWorks製品概要

MATLAB® 言語の主な特徴

- 使いやすく簡潔な構文 (短くシンプルでありながら、ユーザーにその意味がわかりやすくなっています)
- 質の高い数値アルゴリズム (数値解析研究者のコミュニティと密接に協力して開発されています)
- 強力で使いやすいグラフィックスおよび可視化機能
- 高水準言語 (Fortran や C などの言語では数百行のコードが必要な演算をわずか 1、2 行で実行できます)
- 高い拡張性 (ユーザーが自分で拡張することも、MATLAB のコードとアプリを用途別にまとめたパッケージであるツールボックスを利用することもできます)
- スパース行列を含む実数型および複素数型のベクトルと行列 (これらは基本的なデータ型になります)

Simulink® の主な特徴

- 動的および組み込みシステムのモデル化、シミュレーション、および実装のための完全な環境
- 線形、非線形、離散時間、連続時間、ハイブリッド、およびマルチレートシステムの設計とテスト
- 制御、デジタル信号処理 (DSP)、通信、およびシステムエンジニアリングで利用可能
- 他の環境のモデルと簡単に統合できるオープン アーキテクチャ



1. 導入

MATLAB(MAtrix LABOratory)とは一言で言うと何か？

- 独断と偏見で言えば・・・
- 文法的に似たコンピュータ言語環境は
- BASIC (beginner's all-purpose symbolic instruction code) (初心者向け汎用記号命令コード) 1970年代のPCによく使われた。基本的に逐次解釈型言語(インタープリター)
- FORTRAN(1950年代に登場。“FORmula TRANslation”の略。数値計算プログラム作成に適しています)
- PASCAL(1970年に発表されたプログラミング言語。構造化プログラミング環境として設計された)
- かなあ・・・？
- ⇒ つまり、言語構造としては、コンピュータ黎明期の言語のDNAを継承するので**易しい**。
- ⇒ C言語系(ハードウェア志向)、JAVA系(オブジェクト指向、プラットフォーム非依存型)ではない。
- ※念のため、オブジェクト指的な表現でもプログラムは書けます。C言語とI/Fする機能(MEX,S-Function等)もあります。

- さらに、**数値計算を極めて得意**とし、制御、信号処理、画像処理、機械学習、数学など、様々なオプションライブラリで、基本関数を提供するので、**プログラマーは、基本機能を開発しないで、その労力を本来のアルゴリズム開発に集中できる**のです。
- ※オプションライブラリ(Toolbox,Blockset)100以上あります。

1. 導入

簡単な例

```
% {  
A=[1 2 ...  
   3 4]  
B=[5 6 ...  
   7 8]  
のとき、C=A+Bを計算  
%}
```

%% 普通の言語の書き方(成分)

```
A=[1 2;3 4]  
B=[5 6;7 8]
```

```
for jj=1:2  
    for ii=1:2  
        C(ii, jj)=A(ii, jj)+B(ii, jj)  
    end  
end
```

%% MATLABの書き方(行列/ベクトル)

```
C=A+B
```

※
MATLABでは、
数の計算は成分ではなく
ベクトル、行列扱いで計算できます。
大規模データの扱いに
大きなメリットが得られます。

1. 導入

MATLABのプログラムの実行

拡張子m

mファイル

例えば

plus_sample.m

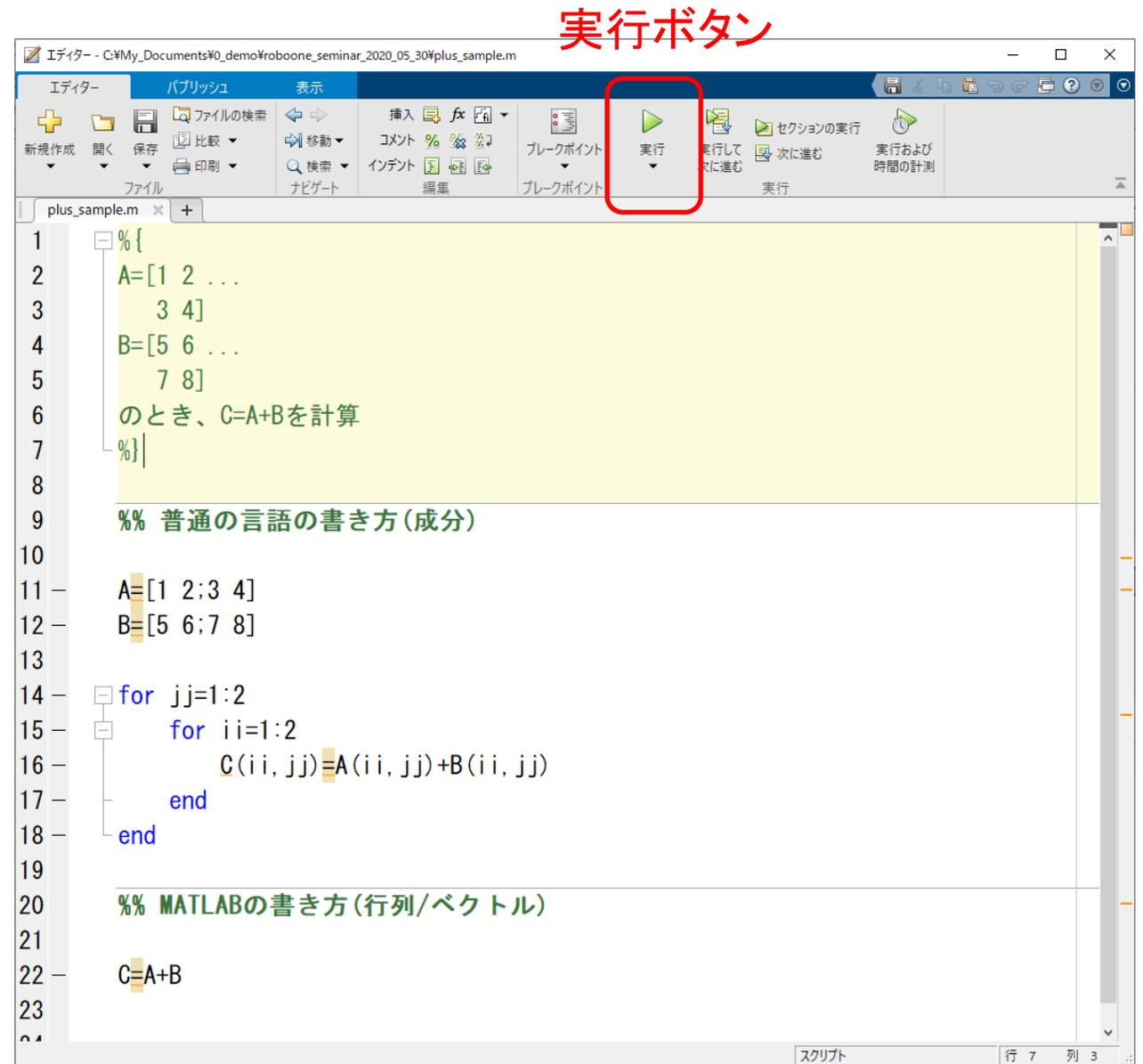
というプログラムでは

MATLABコマンドウィンドウから

>> plus_sample

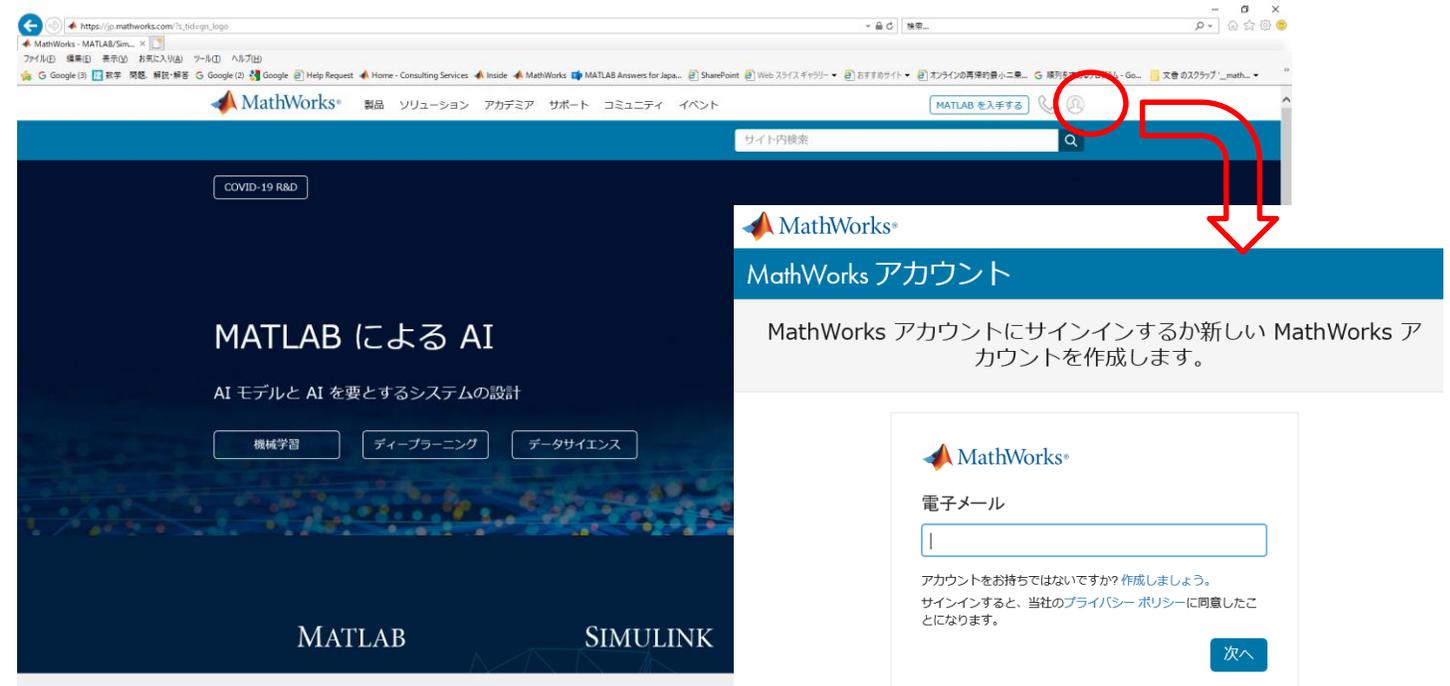
あるいは、plus_sample.mを開き

実行ボタンをクリック



2. 環境構築

- MathWorksは、期間限定でStudent Competition (学生競技)版体験ライセンスを貸し出しています。インストールしてROBO-ONEのエンジニアリング作業に使うことができます。
 - 【手順】
 - [1] MathWorks HPを参照します。
 - <https://jp.mathworks.com/>
 - [2] MathWorks アカウントを作ります。
- アカウントがあれば・・・
- (1) Student Competition用の学生競技ライセンスがDLできます。
 - (2)補助用のサポートパッケージ(無料)がDLできます。



2. 環境構築

Student Competition (学生競技)版体験ライセンスの貸し出し

- 次のHPを参照します。
- <https://jp.mathworks.com/academia/student-competitions/robo-one.html>

The screenshot shows the MathWorks Academia website. The top navigation bar includes 'MathWorks®', '製品', 'ソリューション', 'アカデミア', 'サポート', 'コミュニティ', and 'イベント'. A search bar is present with the text 'サイト内検索'. Below the navigation bar, there are links for 'Student Home', 'MATLAB Student', 'MATLAB の例', '競技会とコンテスト', '書籍', and 'ハードウェア サポート'. The main content area is titled 'ROBO-ONE' and contains a paragraph of Japanese text describing the competition. To the right of the text is a photograph of two humanoid robots on a stage.

ここから
学生向けライセンスを
DLします。



無償ソフトウェア

MathWorks は、このコンテストに無償でソフトウェアを提供しています。このコンテストの参加チームでソフトウェアをご希望の場合は、ソフトウェア申し込みフォームをご記入のうえ、お申し込みください。

ソフトウェアの利用を申し込む



Student Tutorials and Videos

Learn how to use MATLAB and Simulink to design algorithms, create simulations, deploy code, and speed up software development for your projects.

ビデオを見る

2. 環境構築 サポートパッケージのinstall

- (1) ネット環境に接続
- (2) MATLAB起動
- (3) MATLABのGUIから
- [アドオン]→[アドオンの入手]
- [MATLAB Support Package for USB Webcams]をダウンロードします。
- これで、ラップトップPCに付く、カメラやUSBカメラ等で画像をMATLAB上に
- 取り込めます。
-

The image illustrates the steps to install a support package in MATLAB:

- Step 1:** In the MATLAB GUI, click on the **アドオン (Add-ons)** menu, then select **アドオンの入手 (Get Add-ons)**.
- Step 2:** In the Add-on Explorer, search for **ハードウェア サポート パッケージ (Hardware Support Packages)**. The **MATLAB Support Package for USB Webcams** is highlighted.
- Step 3:** Click on the **インストール (Install)** button for the selected package.

3. 情報リソース

Student Competition (学生競技)版 向け 自己学習教材

次のHPを参照します。参加チーム向けビデオチュートリアル

<https://jp.mathworks.com/academia/student-competitions/robo-one.html>

The screenshot shows the 'Academia' page on the MathWorks website. The page features a navigation bar with 'Student Home', 'MATLAB Student', 'MATLAB の例', '競技会とコンテスト', '書籍', and 'ハードウェア サポート'. A search bar is located on the right. The main content area is titled '参加チーム向けビデオ チュートリアル' and contains six cards, each with an icon, a title, a brief description, and a button to view the tutorials.

Icon	Title	Description	Action Button
	インタラクティブ オンライン チュートリアル	使いながら覚えよう！この2時間のオンラインコースは、WebベースのMATLABを使ったハンズオン形式になっており、コンテキストに応じたフィードバックをその場で受けることができます。	すべてのチュートリアルを表示します
	物理モデリング ビデオ チュートリアル	この無料トレーニング教材で、自動車システムのモデリング、シミュレーション、解析を始めましょう。	トレーニング ビデオの利用を申し込む
	コンピューター ビジョン ビデオ チュートリアル	この無料トレーニング教材で、MATLAB と Simulink を使用してコンピューター ビジョン アルゴリズムを設計し実装する方法を学びましょう。	ビデオ チュートリアルの利用を申し込む (12 ビデオ)
	コード生成 ビデオ チュートリアル	MATLAB 関数および Simulink モデルを C/C++ コードに変換する方法を学びます。	ビデオ チュートリアルの利用を申し込む (7 ビデオ)
	モバイル ロボティクス ビデオ チュートリアル	この無料トレーニング教材で、MATLAB と Simulink を使用して自律型ロボットのアルゴリズムを設計しテストする方法を学びましょう。	ビデオ チュートリアルの利用を申し込む (6 ビデオ)

3.情報リソース

Student Competition (学生競技)版 向け 自己学習教材

次のHPを参照します。その他のリソース

<https://jp.mathworks.com/academia/student-competitions/robo-one.html>

その他のリソース

全て閉じる

▼ Getting Started with Code Examples

以下の例は、物理モデリングがロボット設計にどのように役立つかをご紹介します。これらのデモにアクセスするには、コマンド ラインから該当する関数を入力してください。

SimMechanics を使用したロボットのモデリング、
mech_robot_vr

SimElectronics を使用したアクチュエーターのモデリング、
elec_controlled_dcmotor、elec_inact_control、または elec_kers

SimPowerSystems を使用したアクチュエーターのモデリング、
cs_robot、または power_dcmotor

Stateflow を使用したシーケンス ロジックのモデリング、
sf_launchabort

コード生成、
web([docroot '/toolbox/ecoder/ecoder_example_index.html'])

▼ ビデオ

- ◎ LEGO MINDSTORMS EV3 プログラミングをSimulinkで (26:13)
- ◎ Simulinkで衝突回避のためのモバイルロボットシミュレーション (45:02)
- ◎ Stateflowを用いたLEGOプログラミング (32:17)
- ◎ Simulink と Stateflow によるステートマシンと制御ロジック, Part 1 (35:24)

Racing Lounge ブログ

学生向けコンテストにおける成功事例やチームの体験談を紹介しています。

ブログを読む



MATLAB Centralのユーザー コミュニティに参加する

このコンテストに関して質問する、もしくはファイルを送信する

MATLAB Answers に寄せられた質問を見る

MathWorks サポートページの資料を見る

MATLAB Answers で質問する

The Winner's Circle

MATLAB と Simulink を活用してコンテストを勝ち抜いた世界各地の学生チームをご紹介します。

学生プロジェクトの一覧を見る

Connect with Us

✉ このコンテストについてMathWorksに問い合わせる

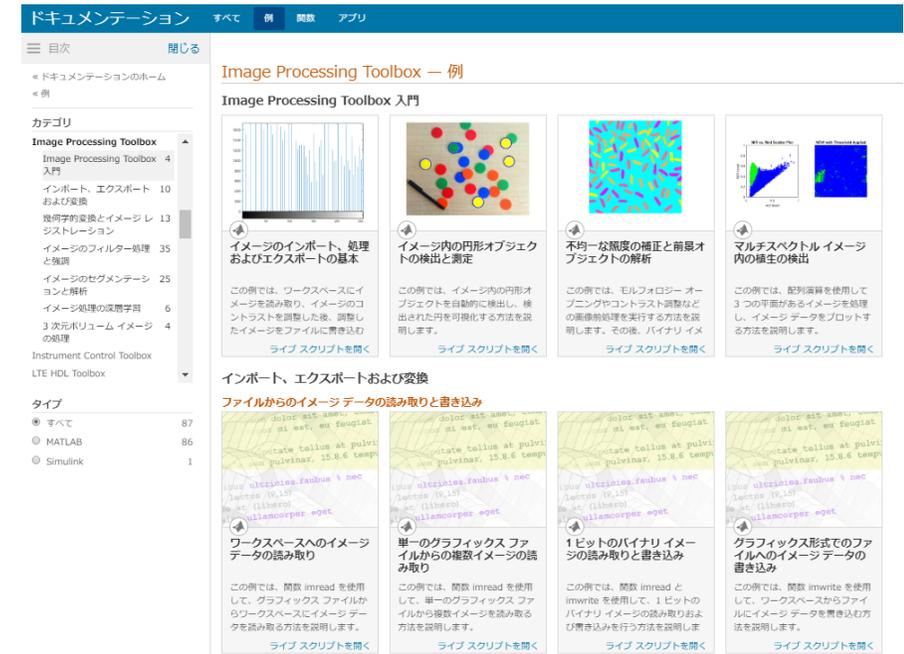
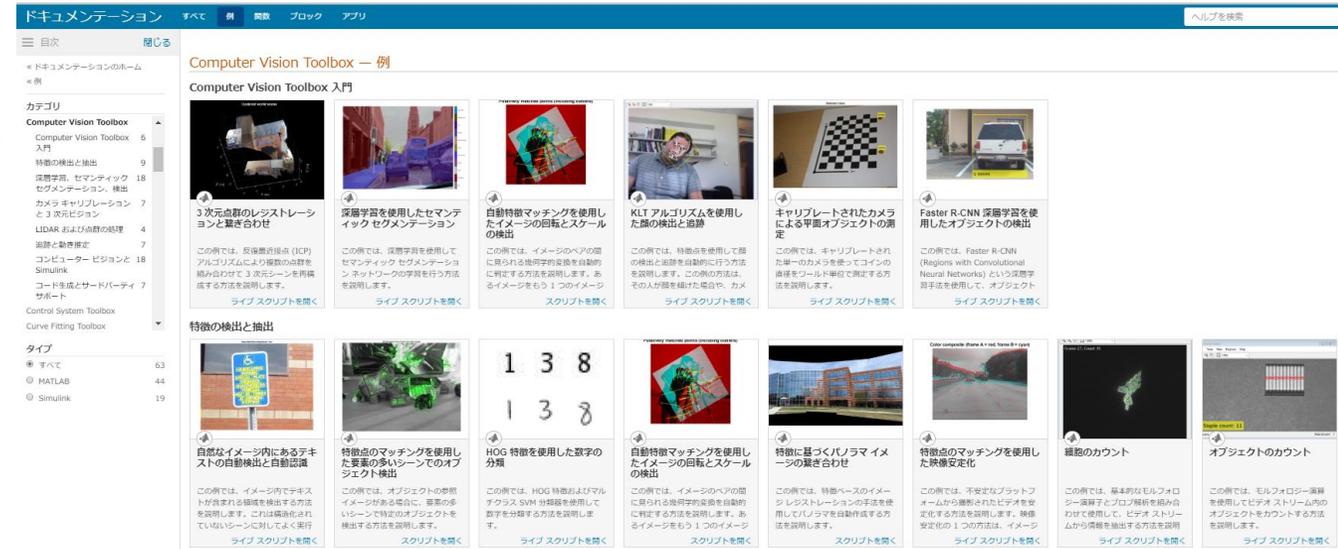
f MATLAB and Simulink Robotics Arena

3. 情報リソース MATLAB情報の取得(オフライン)

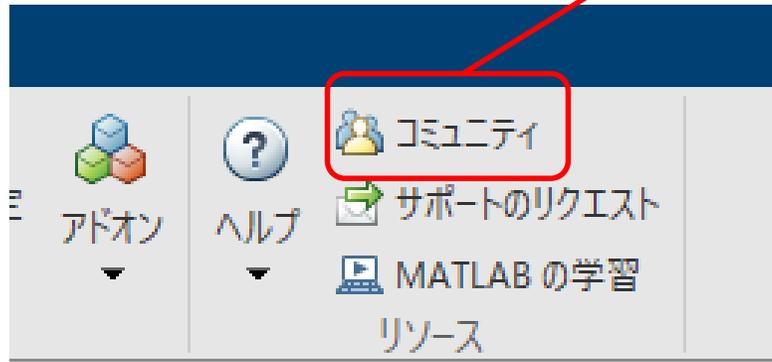
- オフラインでは
- MATLABから
- >>demo

- 各Toolboxのサンプルデモが見れます。
- ソース付

- 各Toolbox の関数や機能を調べるのは
- MATLABから
- >>doc 関数名 例: doc detectSURFFeatures
- >>doc 機能名 例: doc 画像



3. 情報リソース MATLAB情報の取得 (オンライン)



MathWorks®

製品 ソリューション アカデミア サポート コミュニティ イベント

MATLAB® Central

Search Support Support

MATLAB Answers File Exchange

An open exchange for the MATLAB and Simulink user community

Explore the Community

Can matlab work with NVIDIA 2gb Graphics card?

MathWorks® Products Solutions Academia Support Community Events

File Exchange

Filter by Source

Community	37,946
MathWorks	269

Filter by Category

Using MATLAB	
Get Started with MATLAB	36
Language Fundamentals	1,887
Data Import and Analysis	1,666
Mathematics	2,962
Graphics	3,217
Programming	682
App Building	680
Software Development Tools	268
External Language Interfaces	637
Environment and Settings	280
Installation, Licensing, and Activation	21
Parallel Computing	189
Application Deployment	122
Database Access and Reporting	179

Most Recent

- General Adsorbent Library and Evaluation (GALE)
- eeglabdevelopers/eeglab
- Guaranteed Automatic Integration Library

Community Toolboxes

File Exchange: ユーザの作成したプログラムのシェア

3. 情報リソース

- ***MATLABへ通じる道は一夜にしてならず！！***
- ***僕の前に道はない。僕の後には道はできる！！***
- ということでは…
- 自分がやりたいことと似ているサンプルを探しましょう。
- サンプル(Demo)をしっかりと調べましょう。
- (プログラムの構造と使っているライブラリ関数)
- 気になる関数はhelpで、manualを調べましょう。

4. サンプルの紹介

- Demo [1]
静止画像の連続取得
 - Image Acquisition Toolbox
- Demo [2]
画像処理の基本
 - Image Processing Toolbox
- Demo [3]
円を数えて結果を表示
 - Image Processing Toolbox
- Demo [4]
顔面検出の例
 - Computer Vision Toolbox
- Demo [5]
ディープラーニングによる画像の判別
 - Deep Learning Toolbox

ある環境の例

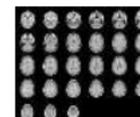
- 注: カメラはラップトップPCの内臓カメラと外付けUSBカメラの場合
- %カメラのリスト
- `camList = webcamlist`
- `camList =`
- `2 × 1 の cell 配列`
- `{'Integrated Camera'}`
- `{'HD USB Camera' }`

画像処理用 関数の一例

画像表示・調査用各種ツール

画像の表示・調査用の関数

<code>imshow()</code>	イメージの表示	(MATLAB基本関数) R2014b
<code>imshowpair()</code>	イメージ間の差異の比較、横に並べて表示	(合成は <code>imfuse()</code>)
<code>montage()</code>	複数のイメージフレームを四角形モンタージュとして表示	(異なる画像サイズサポート R2018a)
<code>warp()</code>	テクスチャ マッピングされたイメージの表示	
<code>volshow()</code>	3次元ボリュームデータの表示	R2018b
<code>imshow()</code>	画像ビューアー アプリケーション	
<code>colorThresholder()</code>	色の閾値 アプリケーション	
<code>imcontrast()</code>	コントラスト調整ツール	
<code>imcrop()</code>	画像のトリミング (3次元対応: <code>imcrop3</code>)	R2019b
<code>imdistline()</code>	距離ツール	
<code>getpts()</code>	マウスを使用した、ピクセル位置の指定	
⋮		



```
A = imread('peppers.png')
figure; imshow(A) % Figureを一つ開き、画像を表示
```

画像処理用 関数の一例



Image Acquisition Toolbox™

USBカメラからの動画像の取込み

Image Acquisition Toolboxの
System objectによる取込み

R2012a

```

%% 画像を取り込むためのオブジェクトの生成
hCamera = imaq.VideoDevice('winvideo', 1)
hCamera.ReturnedDataType = 'uint8';

%% ビデオを表示するためのオブジェクトの生成
viewer = vision.DeployableVideoPlayer;

%% 1フレーム毎に処理するためのループ処理
for i=1:200
    I = step(hCamera);    %1フレーム取込み
    % 各種処理
    step(viewer, Itxt);  %1フレーム表示
end

%%
release(hCamera);
release(viewer);

```

必要なハードウェアサポートパッケージ：
Image Acquisition Toolbox Support Package for
OS Generic Video Interface

MATLABの関数による取込み

注) USB Video Class (UVC) compliant
Webcam の場合のみ

R2014a

```

%% 画像を取り込むためのオブジェクトの生成
camera = webcam(1)

%% ビデオを表示するためのオブジェクトの生成
viewer = vision.DeployableVideoPlayer;

%% 1フレーム毎に処理するためのループ処理
for i=1:200
    I = snapshot(camera); %1フレーム取込み
    % 各種処理
    step(viewer, Itxt);   %1フレーム表示
end

%%
clear('camera');
release(viewer);

```

必要なハードウェアサポートパッケージ：
USB Webcams

画像処理用 関数の一例

高速動画ストリーミング処理

Computer Vision Toolbox

動画入出力

`vision.VideoFileReader()`

動画ファイルの読み込み（音声も可）

`vision.VideoPlayer()`

動画表示

`vision.DeployableVideoPlayer()`

動画表示（Windows用Cコード生成対応）

更に高速化 **R2015a**`vision.VideoFileWriter()`

動画ファイルの書き出し（音声も可）

`vision.BinaryFileReader()`

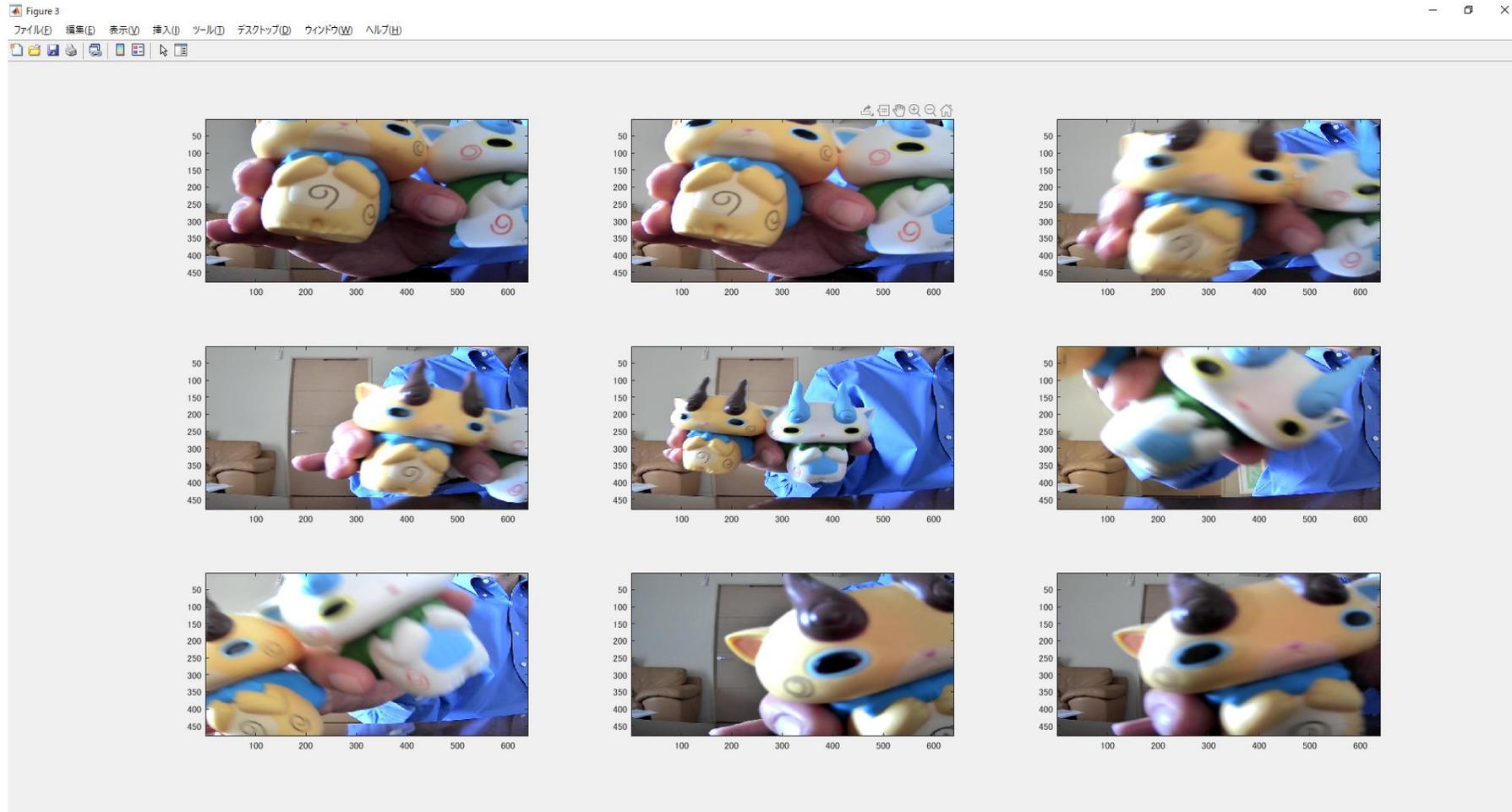
バイナリ動画ファイルの読み込み

`vision.BinaryFileWriter()`

バイナリ動画ファイルの書き出し

4. サンプルの紹介

Demo [1] 静止画像の連続取得



Demo [1] 静止画像の連続取得

Still_image.m

```
% Still_image.m
```

```
%サンプル：静止画像の取得
```

```
%カメラのリスト
```

```
camList = webcamlist
```

```
%WEBカメラの接続、設定確認
```

```
cam = webcam(2)
```

```
%カメラのプレビュー
```

```
figure(1)
```

```
preview(cam);
```

```
%フレーム取得
```

```
img = snapshot(cam);
```

```
%フレームの表示
```

```
figure(2)
```

```
image(img);
```

```
%
```

```
figure(3)
```

```
for idx = 1:9
```

```
    img = snapshot(cam);
```

```
    subplot(3,3,idx)
```

```
    image(img);
```

```
    pause(1)
```

```
end
```

```
%画像処理オブジェクトの消去
```

```
clear cam
```

```
%Copyright 2014 The MathWorks, Inc.
```

4. サンプルの紹介

Demo [2] 画像処理の基本

(1) 画像の読み取り



(2) 画像の二値化



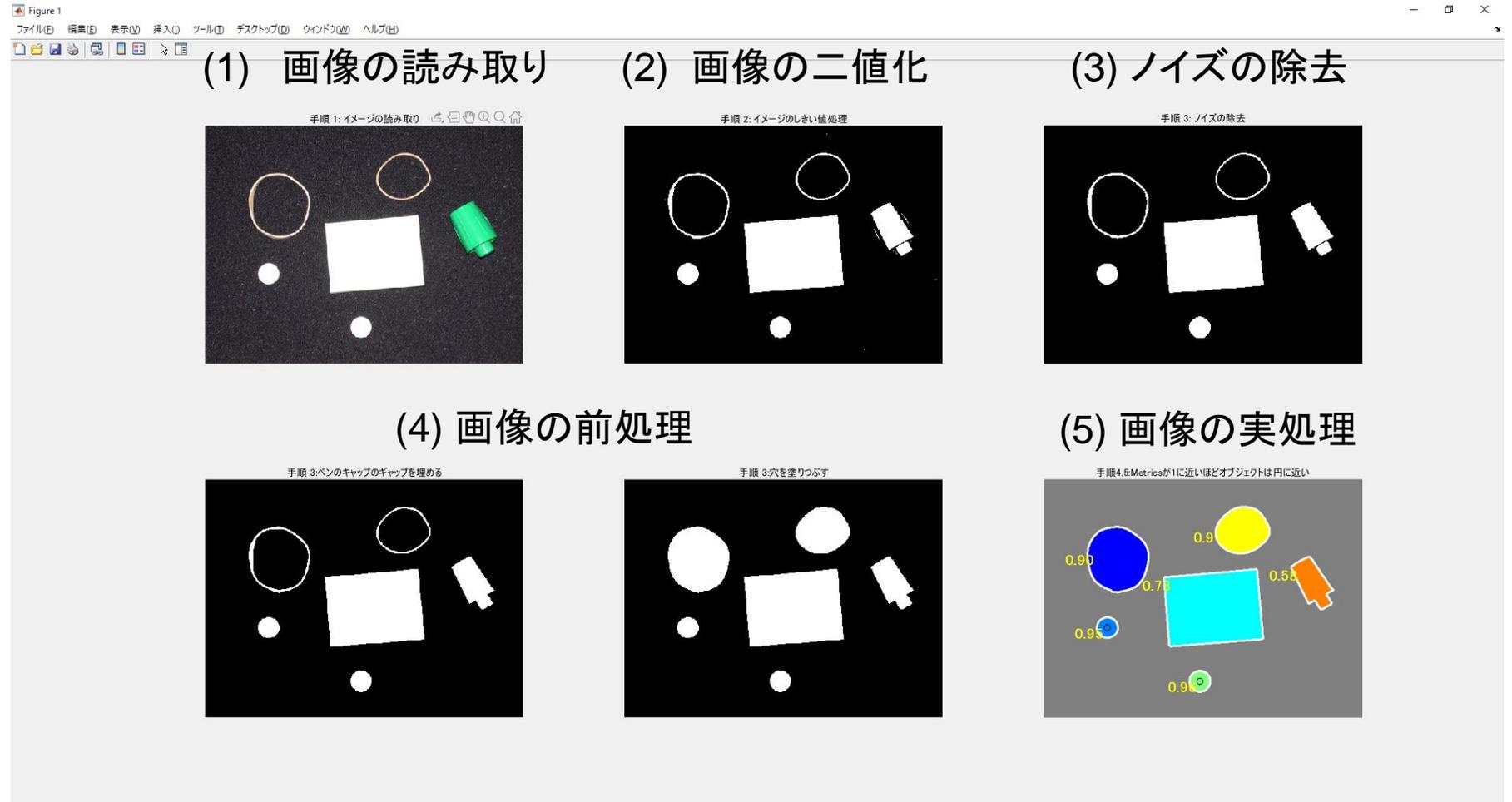
(3) ノイズの除去



(4) 画像の前処理



(5) 画像 実処理



Demo [2]

画像処理の基本

%円形オブジェクトの認識

%この例では、境界をトレースするルーチンである `bwboundaries` を使用して、
%真円度に基づきオブジェクトを分類する方法を示します。

% 手順 1: イメージの読み取り

% `pills_etc.png` を読み取ります。

```
RGB = imread('pillsetc.png');
```

```
figure(1)
```

```
subplot(2, 3, 1)
```

```
imshow(RGB)
```

```
title('手順 1: イメージの読み取り')
```

%手順 2: イメージのしきい値処理

%イメージを白黒に変換して、`bwboundaries` を使用した境界トレースの準備をします。

```
I = rgb2gray(RGB);
```

```
bw = imbinarize(I);
```

```
subplot(2, 3, 2)
```

```
imshow(bw)
```

```
title('手順 2: イメージのしきい値処理')
```

%手順 3: ノイズの除去

%モルフォロジー関数を使用して、対象オブジェクトに属さないピクセルを除去します

%30 ピクセル未満のすべてのオブジェクトを削除します。

```
bw = bwareaopen(bw, 30);
```

```
subplot(2, 3, 3)
```

```
imshow(bw)
```

```
title('手順 3: ノイズの除去')
```

%ペンのキャップのギャップを埋める

```
se = strel('disk', 2);
```

```
bw = imclose(bw, se);
```

```
subplot(2, 3, 4)
```

```
imshow(bw)
```

```
title('手順 3: ペンのキャップのギャップを埋める')
```

%`regionprops` を使用してそれぞれの境界で囲まれた領域を推定できるように、穴を塗りつぶします。

```
bw = imfill(bw, 'holes');
```

```
subplot(2, 3, 5)
```

```
imshow(bw)
```

```
title('手順 3: 穴を塗りつぶす')
```

%Step 4: 境界の検出

%外部境界線のみ焦點を当てます。オプション '`noholes`' を指定すると、
%`bwboundaries` による内部輪郭の探索が回避されるため、処理が迅速になります。

%ラベル行列を表示して、各境界を描きます。

```
[B, L] = bwboundaries(bw, 'noholes');
```

%ラベルの表示

```
subplot(2, 3, 6)
```

```
imshow(label2rgb(L, @jet, [.5 .5 .5]))
```

```
hold on
```

```
for k = 1:length(B)
```

```
    boundary = B{k};
```

```
    plot(boundary(:, 2), boundary(:, 1), 'w', 'LineWidth', 2)
```

```
end
```

Demo [2]

画像処理の基本

```
%Step 5: 円のオブジェクトの決定
%各オブジェクトの面積および周囲の長さを推定します。
%これらの結果を使用して、次のようにオブジェクトの丸さを示す簡単なメトリクスを作成します。
%metric=(4*pi*area)/perimeter^2
%このメトリクスは円の場合に 1 で、他の形状の場合は 1 未満になります。
%適切なしきい値を設定することにより、識別処理を制御できます。
%この例では、0.94 のしきい値を使用しているため、錠剤の形状だけが円形として分類されます。
%関数 regionprops を使用して、オブジェクトのすべての領域を推定します。関数 bwboundaries によって返されたラベル行列は、関数 regionprops によって再利用できることに注意してください。
stats = regionprops(L, 'Area', 'Centroid');

threshold = 0.94;

% 境界毎に繰り返し計算
for k = 1:length(B)

    % label 'k' に対する境界の (X, Y) 座標を取得
    boundary = B{k};

    % オブジェクトの周囲長を推定
    delta_sq = diff(boundary).^2;
    perimeter = sum(sqrt(sum(delta_sq, 2)));
```

```
% label 'k' に対する面積を計算
area = stats(k).Area;

% 丸みのメトリックを計算
metric = 4*pi*area/perimeter^2;

% 結果表示
metric_string = sprintf('%2.2f', metric);

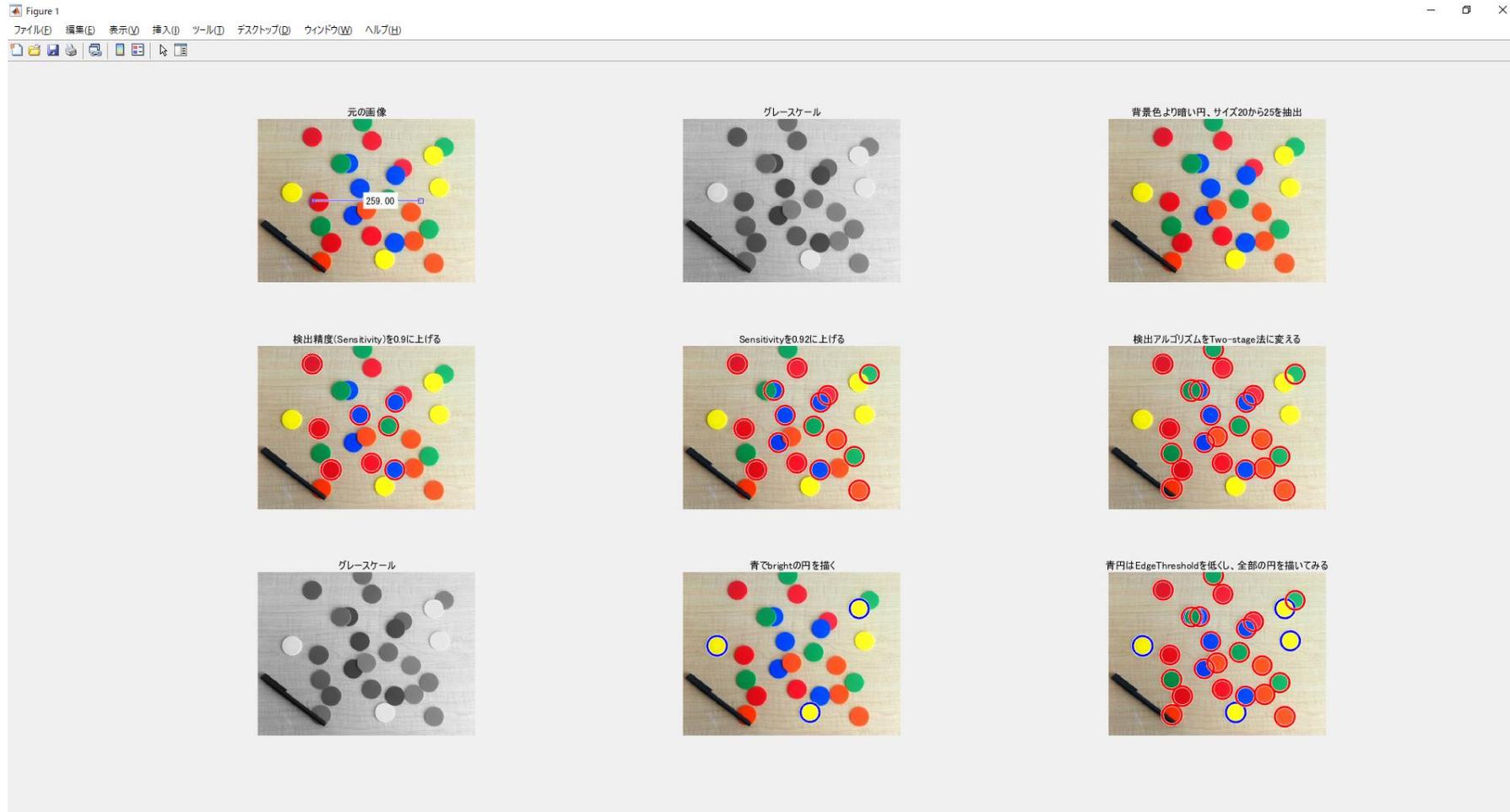
% 円に近いオブジェクトをマーク
if metric > threshold
    centroid = stats(k).Centroid;
    plot(centroid(1), centroid(2), 'ko');
end

text(boundary(1, 2)-35, boundary(1, 1)+13, metric_string, 'Color', 'y', ...
     'FontSize', 14, 'FontWeight', 'bold')
end

title(['手順4, 5: Metricsが1に近いほどオブジェクトは円に近い'])
%Copyright 1993-2015 The MathWorks, Inc.
```

4. サンプルの紹介

Demo [3] 円を数えて結果を表示 実行結果(Count_circle.m)



Demo [3]

円を数えて結果を表示 プログラムリスト1(Count_circle.m)

```

% カラー円のカウンタ
% Count_circle.m

close all;
%{
Step 1.
画像読み込み、表示
%}
rgb = imread('coloredChips.png');
figure(1);
subplot(3,3,1)
imshow(rgb)
title('元の画像')
%{
Step 2.
距離測定ツール設置
%}
d = imdistline;

%{
距離測定ツール消去
%}
delete(d)

%{
Step 3.
画像をグレイメージに変更
%}
gray_image = rgb2gray(rgb);
subplot(3,3,2)
imshow(gray_image)
title('グレースケール')
%{
背景色より暗い円、サイズ20から25を抽出
%}
[centers, radii] = imfindcircles(rgb, [20
25], 'ObjectPolarity', 'dark')
subplot(3,3,3)
imshow(rgb)
title('背景色より暗い円、サイズ20から25を抽出')
h = viscircles(centers, radii);

%{
検出される円無し
Step 4: 検出精度を上げる。
%}
[centers, radii] = imfindcircles(rgb, [20
25], 'ObjectPolarity', 'dark', ...
'Sensitivity', 0.9)

%{
Step 5: 検出された円を描いてみる。
%}
subplot(3,3,4)
imshow(rgb)
title('検出精度(Sensitivity)を0.9に上げる')
h = viscircles(centers, radii);

%{
探せていない円があるので検出精度をさらに上げる。
%}
[centers, radii] = imfindcircles(rgb, [20
25], 'ObjectPolarity', 'dark', ...
'Sensitivity', 0.92);

length(centers)
%{
'Sensitivity'を上げると、発見した円が増える。
%}
subplot(3,3,5)
imshow(rgb)
h = viscircles(centers, radii);
title('Sensitivityを0.92に上げる')

```

Demo [3]

円を数えて結果を表示 プログラムリスト2(Count_circle.m)

```
%{
Step 6: Use Two-stage法をやってみる。さらに発見した円が増える。
%}
[centers, radii] = imfindcircles(rgb, [20 25], 'ObjectPolarity', 'dark', ...
    'Sensitivity', 0.92, 'Method', 'twostage');
subplot(3, 3, 6)
imshow(rgb)
h = viscircles(centers, radii);
title(' 検出アルゴリズムをTwo-stage法に変える')

%{
元の方法に戻して、'Sensitivity' を上げてみる。
%}
[centers, radii] = imfindcircles(rgb, [20 25], 'ObjectPolarity', 'dark', ...
    'Sensitivity', 0.95);
subplot(3, 3, 7)
imshow(rgb)
h = viscircles(centers, radii);
title(' 検出アルゴリズムを元に戻して、Sensitivityを0.95に上げる')

%{
Step 7: 黄色の円が検出できていないことがわかる。強度が背景色と似ているのか。
%}
subplot(3, 3, 7)
imshow(gray_image)
title(' グレースケール')
```

```
%{
Step 8: 黄色の円は背景色より明るい。黄色を検出するため、'ObjectPolarity' を'bright'に変更。
%}
[centersBright, radiiBright] = imfindcircles(rgb, [20 25], ...
    'ObjectPolarity', 'bright', 'Sensitivity', 0.92);
subplot(3, 3, 8)
imshow(rgb)
title(' 黄色の円検出のため、ObjectPolarityをbrightに変更')

%{
Step 9: 違う色（青）で'Bright'な円を描いてみる。
%}
hBright = viscircles(centersBright, radiiBright, 'Color', 'b');
title(' 青でbrightの円を描く')

%{
1つ黄色の円が見つけられない。背景色に対しはっきりしてない。
%}

Step 10: 'EdgeThreshold' を低くしてみる。
%}
[centersBright, radiiBright, metricBright] = imfindcircles(rgb, [20 25], ...
    'ObjectPolarity', 'bright', 'Sensitivity', 0.92, 'EdgeThreshold', 0.1);
subplot(3, 3, 9)
imshow(rgb)
hBright = viscircles(centersBright, radiiBright, 'Color', 'b');
title(' 青円はEdgeThresholdを低くし、全部の円を描いてみる')

%{
Step 11: 全部の円を描いてみる。
%}
subplot(3, 3, 9)
h = viscircles(centers, radii);
%{
Copyright 2012–2018 The MathWorks, Inc.
%}
```

4. サンプルの紹介

Demo[4]顔面検出とトラッキング

物体検出と追跡は、多くのコンピュータビジョンアプリケーションで重要です。

例えば、活動認識、自動車の安全、監視等があります。

この例では、webcamで捉えたライブビデオストリーム上の単一の顔面を追跡するシンプルなシステムを紹介します。

この例を実施するには、webcamのハードウェアサポートパッケージのダウンロードとインストールが必要です。

顔追跡システムは2モード(検出と追跡)のうちのどちらかで実行されます。

【検出モード】

|vision.CascadeObjectDetector| オブジェクトで現在のフレーム中で顔面検出します。

顔面が検出されたら、顔面の角の点を検出します。

|vision.PointTracker| オブジェクトを初期化し、追跡モードに切り替えます。

【追跡モード】

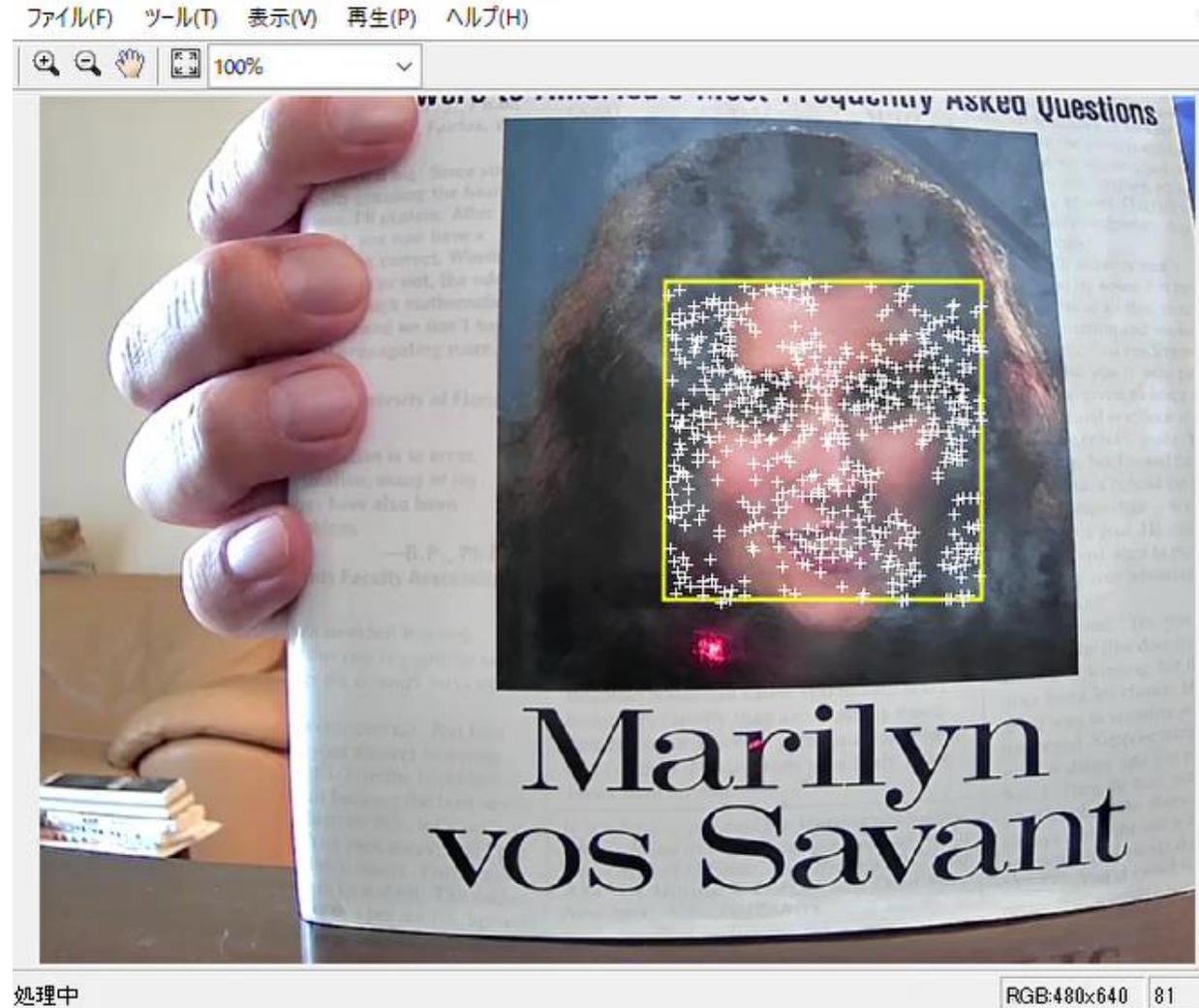
ポイントトラッカーで点を追跡します。

点を追跡するうちに、いくつかの点は排除されロストします。

追跡した点群の数がある閾値を下回ると、顔はもはや認識されないため、検出モードに戻り、再度顔認識することになります。

Demo[4]

顔面検出の例 (FaceTrackingWebcamExample_jp.m)



Demo[5] ディープラーニングによる画像の判別

ディープラーニングでWebcamの画像を分類する。

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する方法を説明します。

MATLAB、シンプルな Web カメラ、および深層ニューラル ネットワークを使用して、身の回りの物を識別します。この例では、GoogLeNet を使用します。

この事前学習済みの深層畳み込みニューラル ネットワーク (CNN または ConvNet) は、100 万枚を超えるイメージについて学習済みであり、

イメージを 1000 個のオブジェクト カテゴリ(キーボード、マグ カップ、鉛筆、多くの動物など) に分類できます。

GoogLeNet をダウンロードし、

MATLAB を使用してカメラのイメージをリアルタイムで連続的に処理できます。

GoogLeNet は広範囲のイメージに対する豊富な特徴表現を学習しています。

入力としてイメージを取り、イメージ内のオブジェクトのラベルと

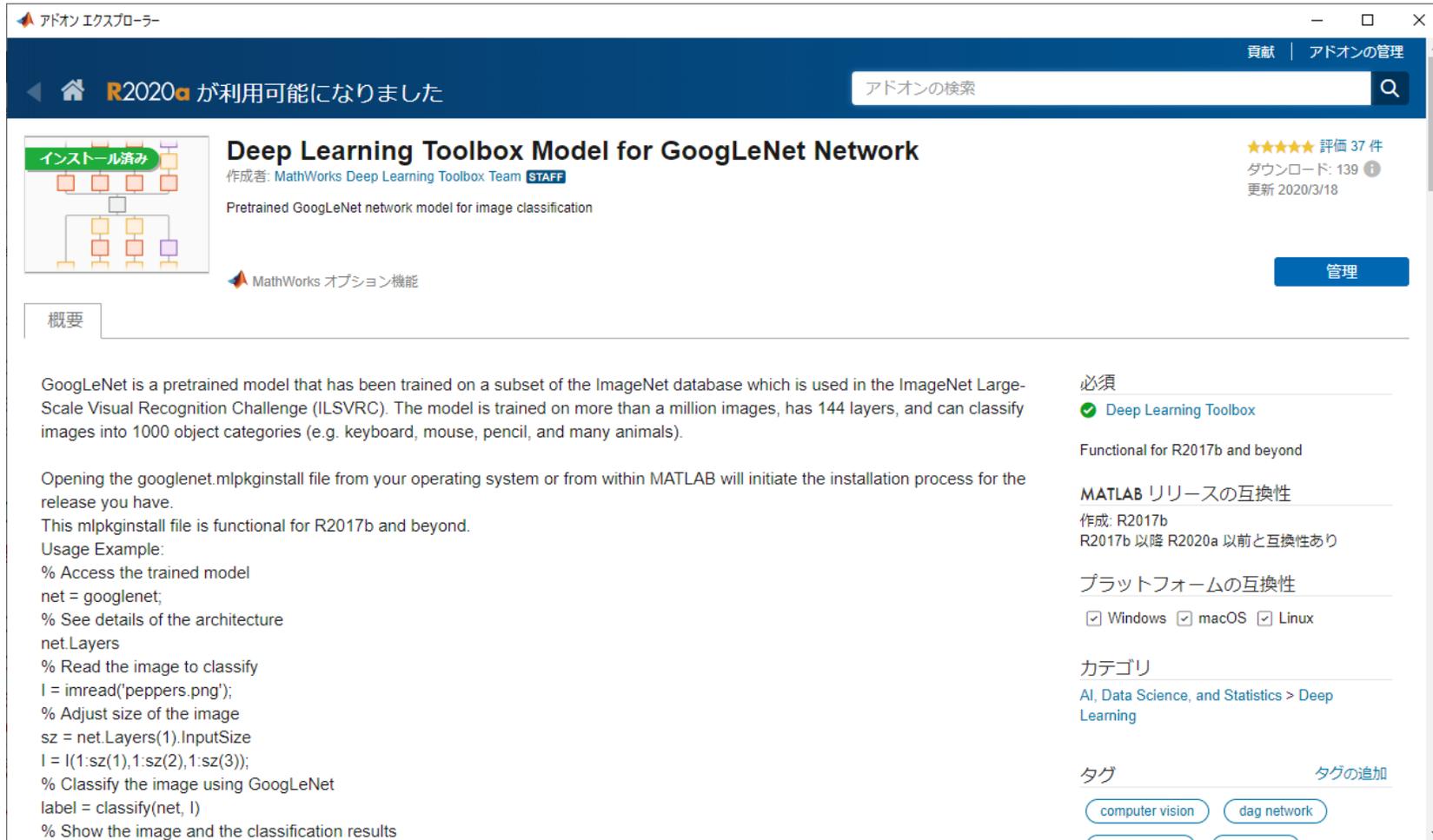
各オブジェクト カテゴリの確率を表示します。

身の回りの物で実験して、GoogLeNet によってイメージが

どの程度正確に分類されるかを確認できます。

Demo[5] ディープラーニングによる画像の判別

この例では、GoogLeNetを使って、物体画像の識別を行います。
まず、Support Packageで、以下のアドオンを追加しましょう。



アドオン エクスプローラー

貢献 | アドオンの管理

← ホーム R2020a が利用可能になりました

アドオンの検索

Deep Learning Toolbox Model for GoogLeNet Network

★★★★★ 評価 37 件
ダウンロード: 139
更新 2020/3/18

インストール済み

作成者: MathWorks Deep Learning Toolbox Team STAFF

Pretrained GoogLeNet network model for image classification

MathWorks オプション機能

管理

概要

GoogLeNet is a pretrained model that has been trained on a subset of the ImageNet database which is used in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). The model is trained on more than a million images, has 144 layers, and can classify images into 1000 object categories (e.g. keyboard, mouse, pencil, and many animals).

Opening the googlenet.mpkginstall file from your operating system or from within MATLAB will initiate the installation process for the release you have.

This mpkginstall file is functional for R2017b and beyond.

Usage Example:

```
% Access the trained model
net = googlenet;
% See details of the architecture
net.Layers
% Read the image to classify
I = imread('peppers.png');
% Adjust size of the image
sz = net.Layers(1).InputSize
I = I(1:sz(1),1:sz(2),1:sz(3));
% Classify the image using GoogLeNet
label = classify(net, I)
% Show the image and the classification results
```

必須

- Deep Learning Toolbox

Functional for R2017b and beyond

MATLAB リリースの互換性

作成: R2017b
R2017b 以降 R2020a 以前と互換性あり

プラットフォームの互換性

- Windows
- macOS
- Linux

カテゴリ

AI, Data Science, and Statistics > Deep Learning

タグ

タグの追加

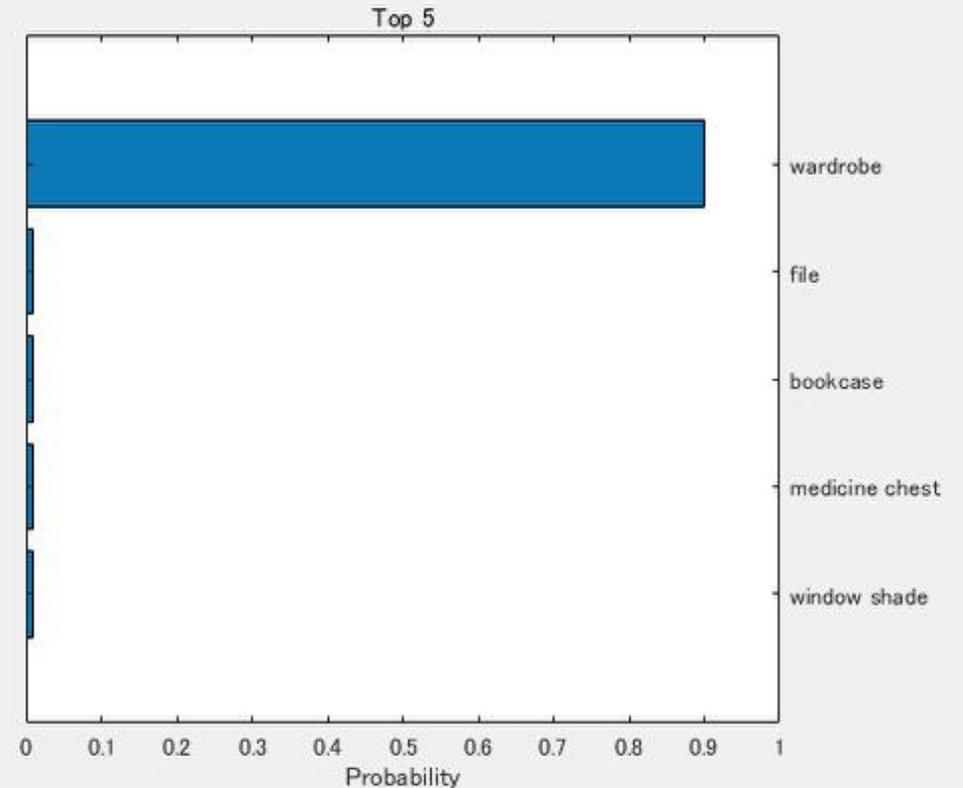
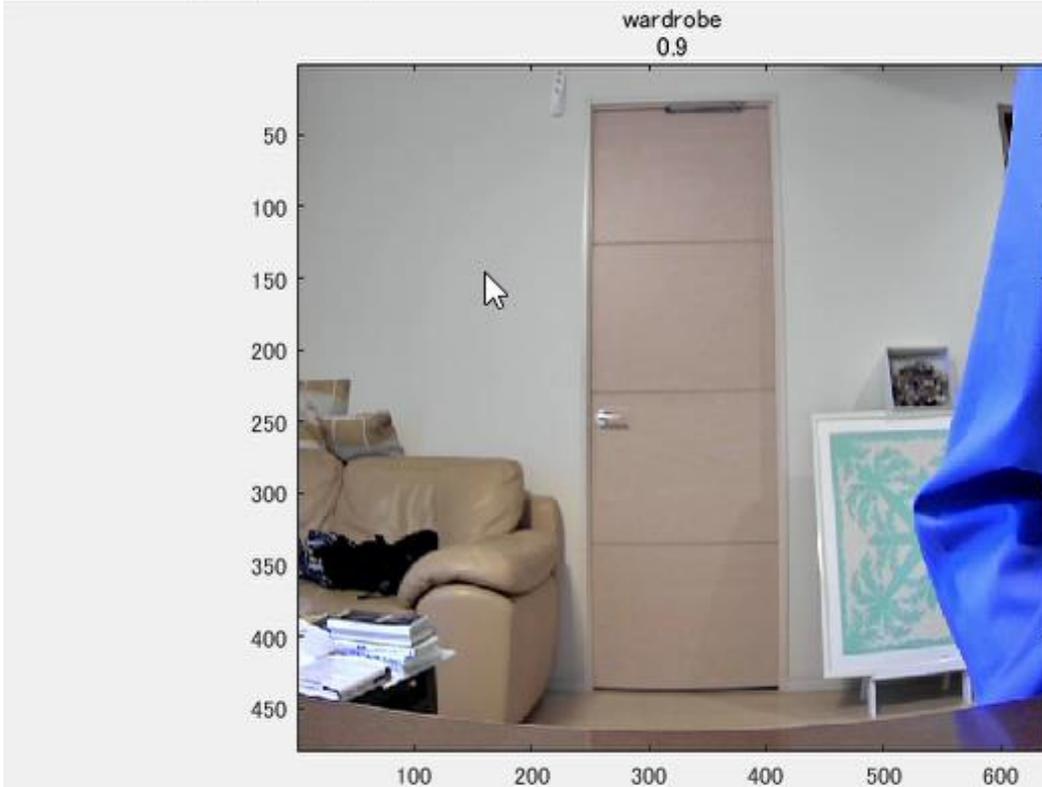
- computer vision
- dag network

4. サンプルの紹介

Demo[5] ディープラーニングによる画像の判別

(ClassifyImagesFromWebcamUsingDeepLearningExample_jp.m)

ファイル(F) 編集(E) 表示(V) 挿入(I) ツール(T) デスクトップ(D) ウィンドウ(W) ヘルプ(H)



最後に 繰り返しになりますが

- **MATLABへ通じる道は一夜にしてならず！！**
- **僕の前に道はない。僕の後には道はできる！！**
- ということですが…
- 自分がやりたいことと似ているサンプルを探しましょう。
- サンプル(Demo)をしっかり調べましょう。
- (プログラムの構造と使っているライブラリ関数)
- 気になる関数はhelpで、manualを調べましょう。
- **それでは、皆様のご健闘を期待しております。**