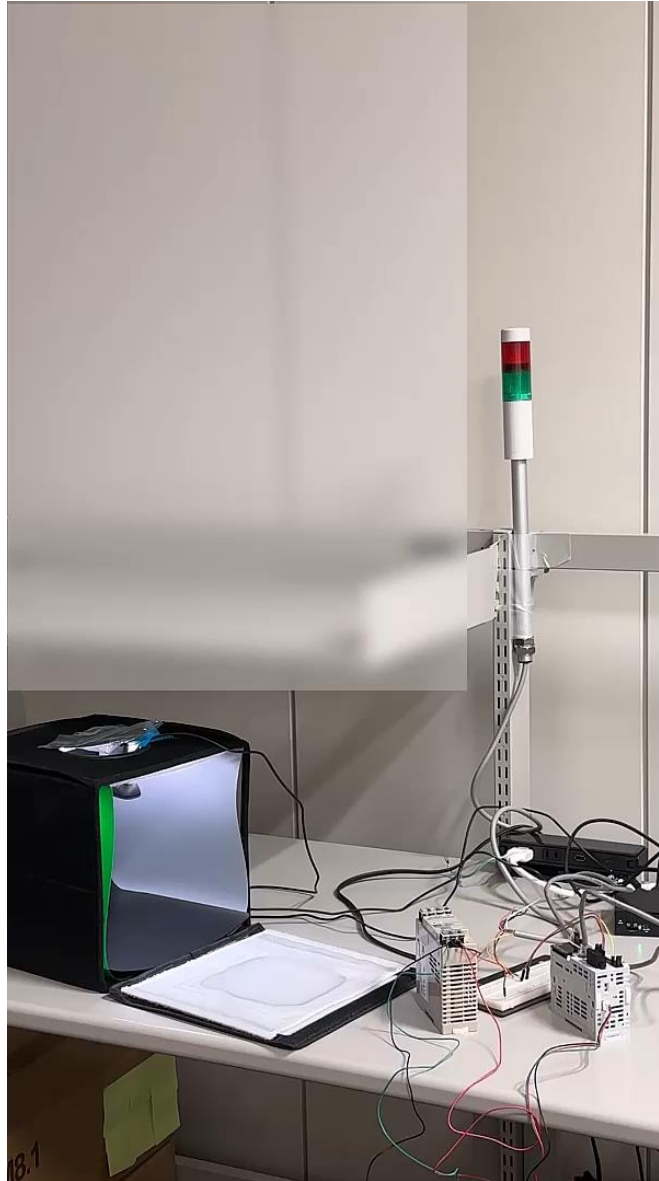


エッジデバイス上のAIによる認識結果をPLCに送り、ランプを制御



MATLABならできる！

外観検査の自動化・内製化・手の内化

AIモデル開発編

MathWorks
Application Engineer
上野 敬志

Today's Highlights

- AIモデル開発編
 - 外観検査AI開発ワークフロー
 - Automated Visual Inspection ライブラリ
- AIモデルのデプロイ編
 - AIによる機器制御の実演
- 内製化を支援するサービスご紹介

```
weather.py
import csv
import datetime
import json
import urllib.request

BASE_URL = "http://api.weather.com/v2/gc/2.5/units/{units}/?units={units}"
FORECAST_DAYS = ["current_time", "date_time", "temp", "t", "temp", "humidity",
                  "wind", "humidity", "wind", "temp", "humidity"]

def read_weather_data from a backup (if any)
    """
    will open 'weather_data.csv' if available, or create it.
    reader = csv.DictReader(csvfile)
    for s in Reader:
        if s["city"] != city:
            data = dict()
            data["city"] = city # Used for error checking below
            # Convert dates times
            for k in ["temp", "humidity", "temp min",
                      "temp max", "wind", "wind", "lat", "lon"]
                data[k] = float(data[k])
            return data
    return None

def get current weather(city, country, unit, **kwargs):
    """get current weather for specified location"""
```

非本質的作業:

プログラミング、ライブラリ間の繋ぎ合わせ、
ツール間の整合性、実装、バージョン管理、知財…

MATLAB®
& SIMULINK®

本質的な課題 + α

最小限の 非本質的作業

Accelerating the pace of Engineering and Science

MATLAB は

非本質的作業:
プログラミング、ライブラリ間の繋ぎ合わせ、
ツール間の整合性、実装、バージョン管理、知財…

を最小化!

データアクセス

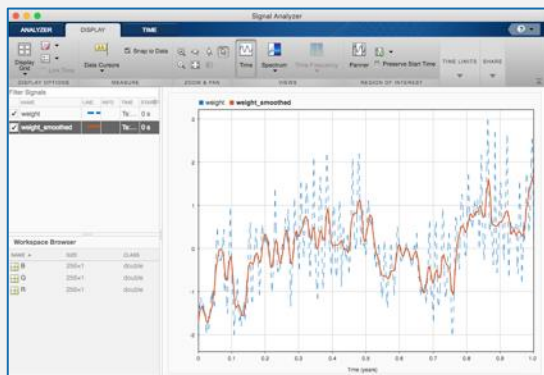
前処理

特徴抽出

モデルの学習

モデルの
チューニング

展開



欠損値、異常値などの
前処理、時刻同期

Import - (Applications\MATLAB_2019b\app\toolbox\matlab\import\outages.csv)

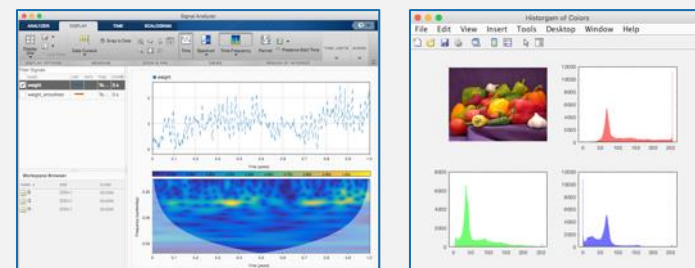
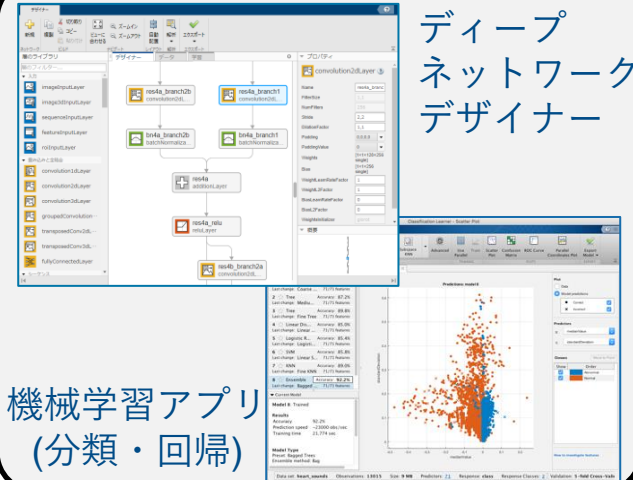
Column delimiter: Range: A1:F1469 Output Type: Table

Fixed Width: Variable Names Row: 1

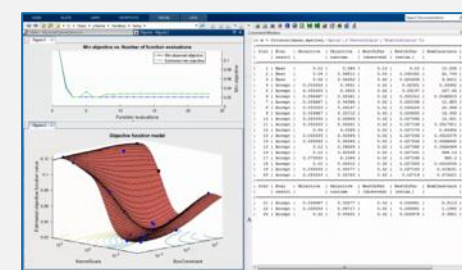
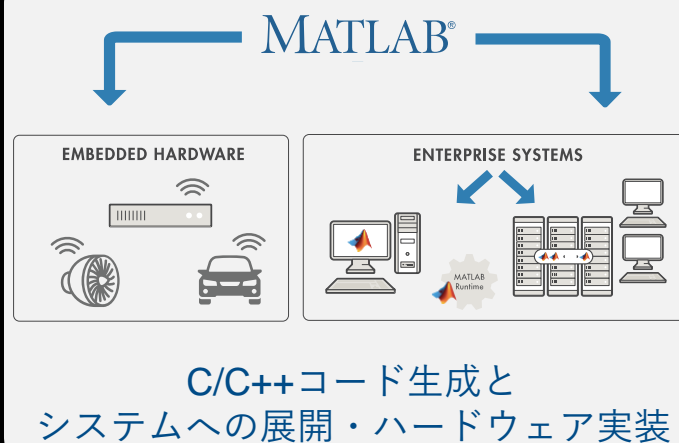
Import Selected Data

Region	OutageTime	Loss	Customers	RestorationTime	Cause
SouthWest	2002-02-01 12:18	418.977218	182019.482	2002-02-07 18:50	winter storm
SouthEast	2003-01-23 00:49	510.1399497	212015.3001	2003-02-17 08:14	winter storm
West	2004-04-06 05:44	414.8015124	340171.0338	2004-04-06 06:10	equipment fault
MidWest	2002-03-16 06:18	186.4367788	212754.0	2003-06-18 10:54	attack
West	2003-06-18 02:49	0	0.0	2003-06-18 10:54	attack
West	2004-06-20 14:39	211.2947226		2004-06-20 19:16	equipment fault

テキストファイル、スプレッドシート、データベース、
バイナリファイル、Web、クラウドストレージ



信号、画像、動画、音声、テキストなどの
ドメイン特有のためのツール



自動パラメータチューニング

MATLABでの外観検査

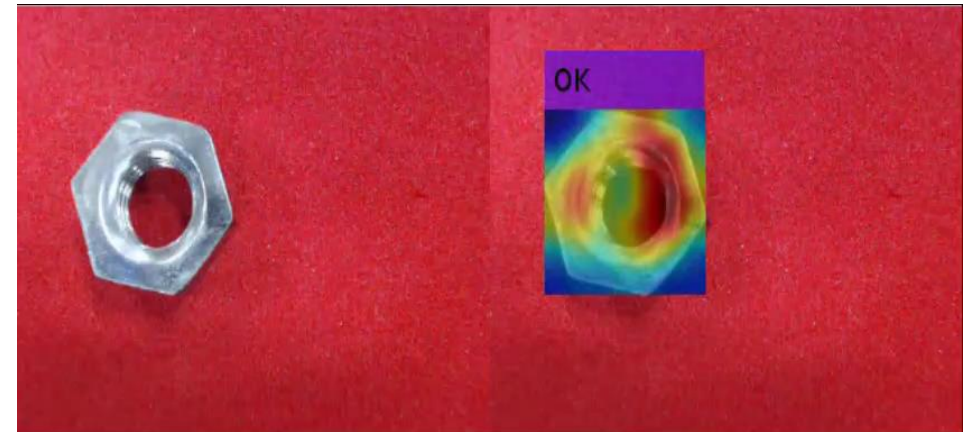
『目視による対象の検査』の自動化：分類問題(教師有り学習)

正常？異常？

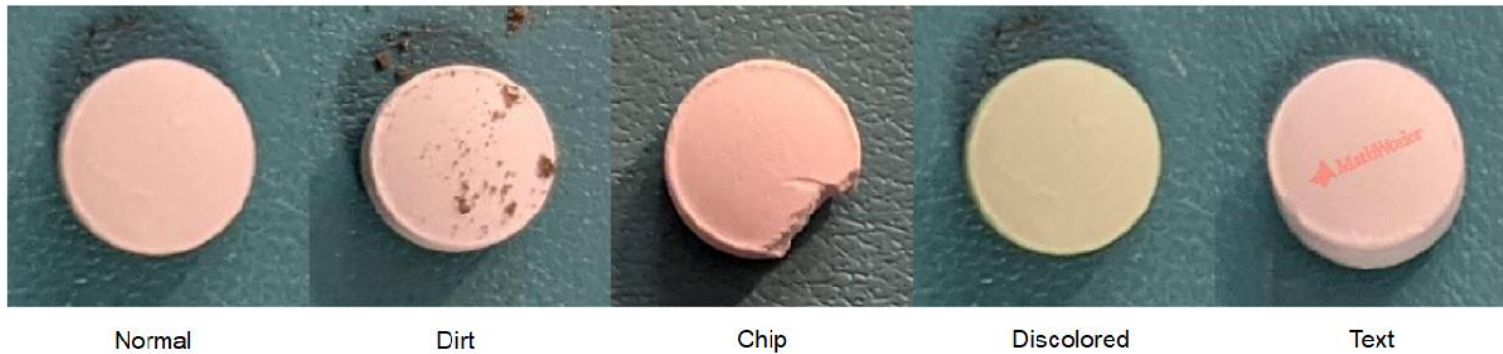


OK or NG

判断の根拠は？どこを見ている？



異常ならばどういう異常？



関西電力、配管溶接部の損傷評価に ディープラーニングを適用

木津 健一氏, 関西電力株式会社

関西電力は、高クロム鋼配管溶接部のクリープ損傷評価の実施に、ディープラーニングを適用しました。従来の外表面を観察する、非破壊検査では発見が困難だった肉厚内部での損傷を評価するため、外表面のひずみ分布計測で得られた画像を用いる手法が試みられています。

研究では、試験体のひずみ分布計測画像を使い、ディープラーニングのネットワークである、AlexNetにより抽出した特徴量を、サポートベクターマシンを使って、損傷小と損傷大に分類できることを確認しました。さらに、同様のネットワークにより損傷率を推定する回帰モデルを作成し、損傷率が大きいほど良い精度で推定できることを確認しました。今後、本手法の実機への適用性確認や、様々な材質の配管への応用などを目指しています。

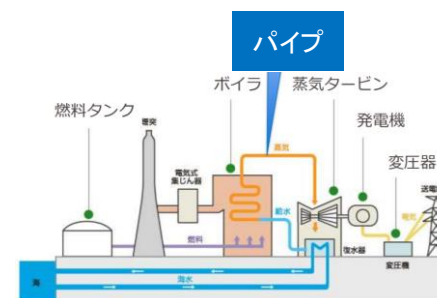
MATLAB利用のメリット:

- ・ 転移学習をすぐに利用できるサンプルコード
- ・ 学習済みネットワークの利用が容易
- ・ 専用ツールボックスによる、その他の機械学習手法との組合せ
- ・ 無料のセミナー、ヘルプドキュメントなど学習方法が充実

“

MATLABを導入して、ディープラーニングを使った開発を簡単に行えました。プログラミングが得意ではなくても、**無料セミナーとヘルプドキュメント**だけで、十分学習することができました。

”

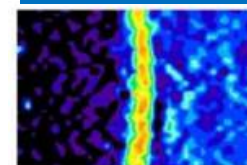


森田ほか, 日本材料学会第67期学術講演会, 609 2018
より引用

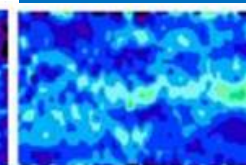
試験体



縦方向ひずみ



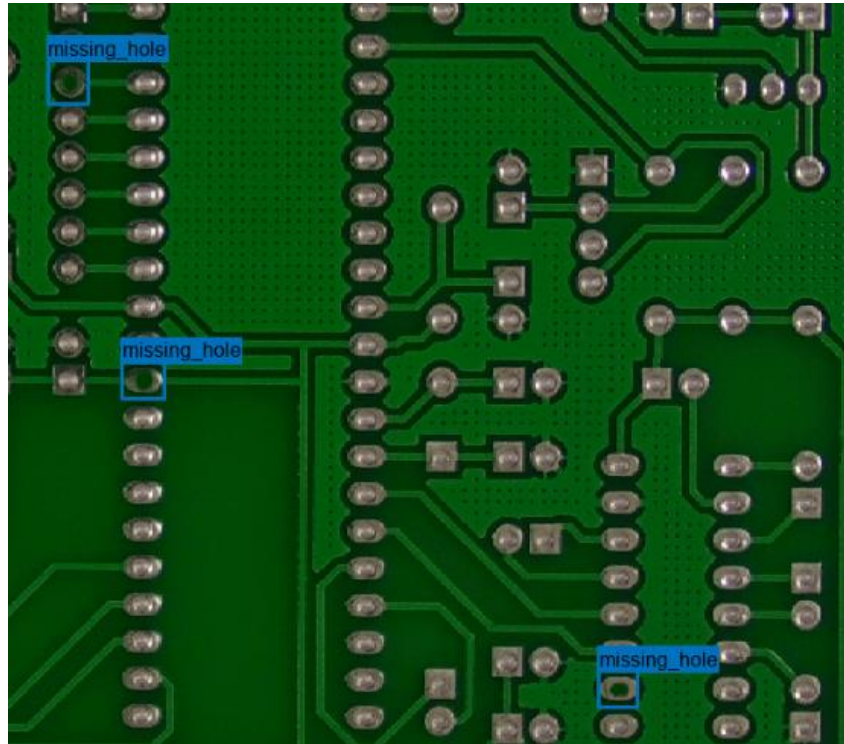
周方向ひずみ



MATLABでの外観検査

『目視による対象の検査』の自動化：物体検出/セグメンテーション(教師有り学習)

異常箇所は？ あるべき場所にあるべきものはある？



デクセリアルズ、フィルム生産の工程検査の 品質改善にディープラーニングを活用

大河原 秀之氏 松下 忍氏, デクセリアルズ株式会社

ディスプレイに使われる機能性材料の生産工程においては、外観品質を確保することが最も重要な課題ですが、高い生産性を誇るRoll to Roll方式での連続生産は、連続不良発生リスクを増加させ、品質管理を難しくしていました。

そこでデクセリアルズでは、検査装置によって検出された欠陥画像に、ディープラーニングを用いて高精度に欠陥を分類し、リアルタイムで不良を集計し、製品品質の異常検知を対策に繋げることで、歩留を向上しました。さらに、GPUで高速化を行い、既存の生産システムに統合し、24時間365日の稼働を実現しました。これらの研究開発から実装までの一連のフローにMATLABを活用することで、設計から導入までの期間を半年に短縮できました。

位置センサー情報などを付加した機械学習の2次利用や、多様な環境下での精度向上にSemantics Segmentationを活用するなど、品質・価値の向上をはじめとする新技術のトレンドへのあくなき追及を今後も行い、企業ビジョン「Value Matters 今までなかったものを。世界の価値になるものを。」の実現を目指します。

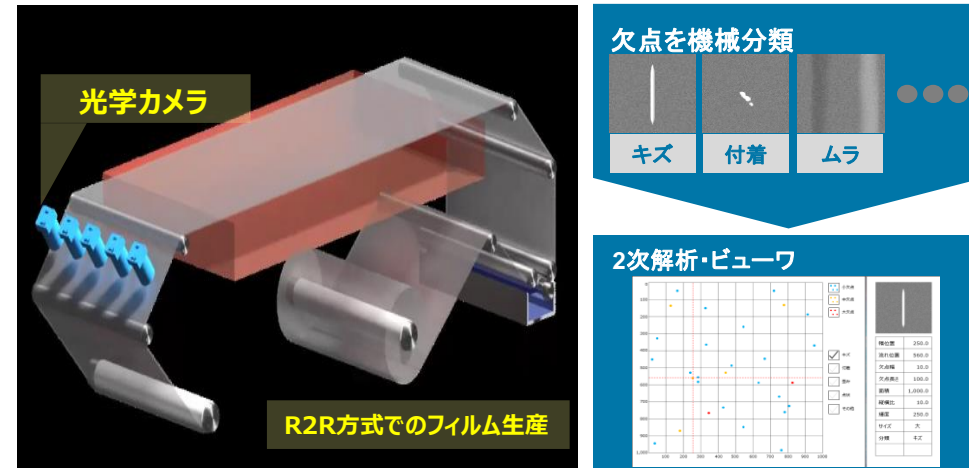
MATLAB利用のメリット:

- 様々な分野にまたがる幅広いデータタイプやアルゴリズムへの拡張性
- MATLAB Compiler SDKによる.NETシステムへの統合と複数ラインへの展開
- マニュアル操作・解析のためのGUIインターフェースの作成と提供
- 研究から実装までの研究開発を加速する統合した開発環境

“

ディープラーニングがAIにブレークスルーをもたらしたように、
**MathWorksとの出会いが、私共に
ブレークスルーをもたらしました。**

”



八千代エンジニアリング、ダム・橋梁損傷検出にディープラーニングを適用

安野 貴人氏, 八千代エンジニアリング

八千代エンジニアリングはダムのポップアウトによる損傷検出にセマンティックセグメンテーションを適用しました。この損傷は形状、大きさ、観測環境が様々なため画一的な画像処理の適用は難しく、検査員が双眼鏡で観測、スケッチで記録していました。

取り組みではまず最初の1か月でMATLABで提供される学習済みモデルとSegNetを合わせて用いることで、過去の画像処理手法より高い精度がでることが確認できました。その後ネットワークモデルの変更、画像の拡張、複数パターンのラベル作成により精度向上を目指しました。今後アプリ化による機能配布と点検の効率化を目指しています。日本国内にダムは3000以上あり、このAIによる自動検出手法の確立・適用により、大幅な損傷検出作業の効率化が期待できます。また同社では河川護岸や橋梁の損傷監視でもAI適用を進めています。

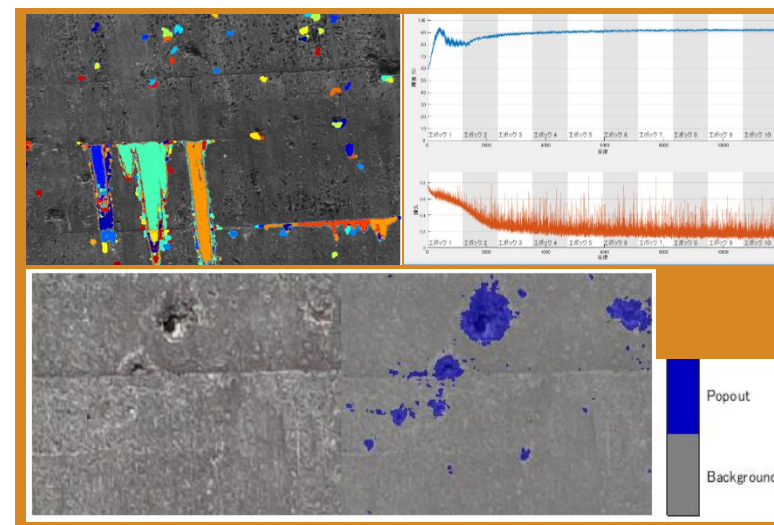
MATLAB利用のメリット

- セマンティックセグメンテーションをすぐに利用できるサンプルコード
- 多数の学習済みネットワークの追加が容易
- 複数GPUアクセス等の容易なディープラーニング用環境構築
- 任意のGUI作成と実行形式ファイルでの配布

“

MATLABを用いることで損傷の画像解析が容易に実現できました。機械学習からディープラーニングまで一つのフレームワークで実行でき、組織内で共有できることで飛躍的に業務が加速しました。

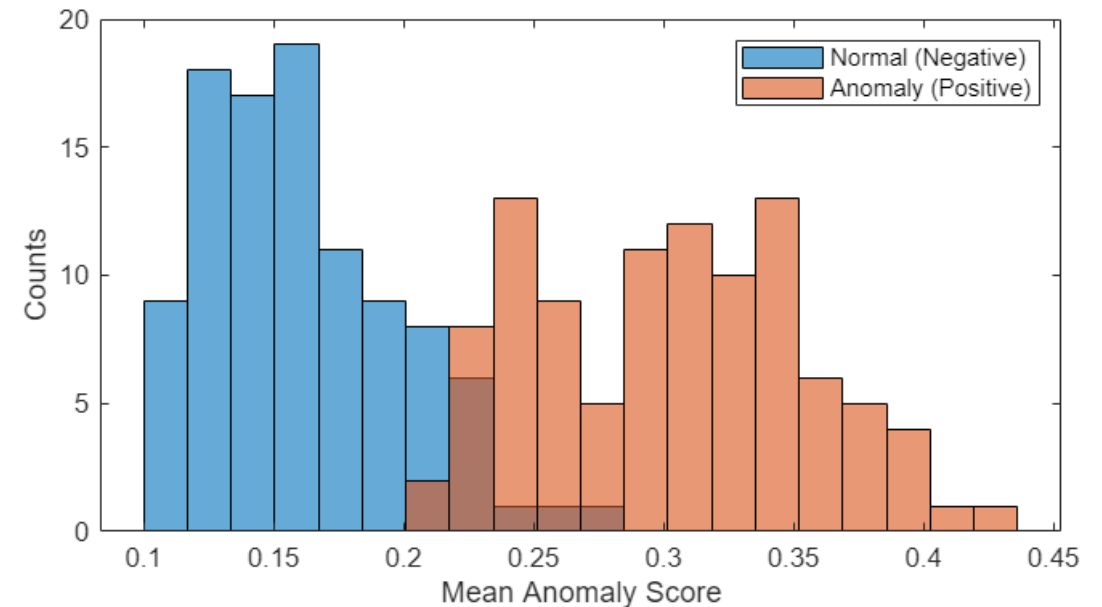
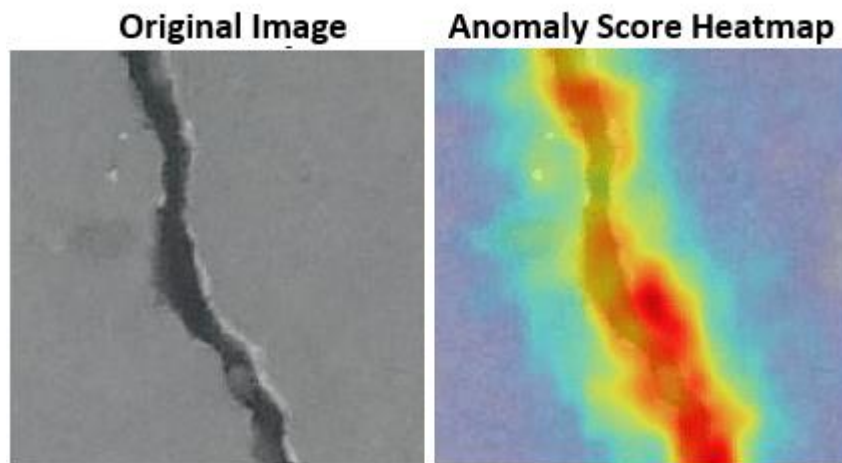
”



MATLABでの外観検査

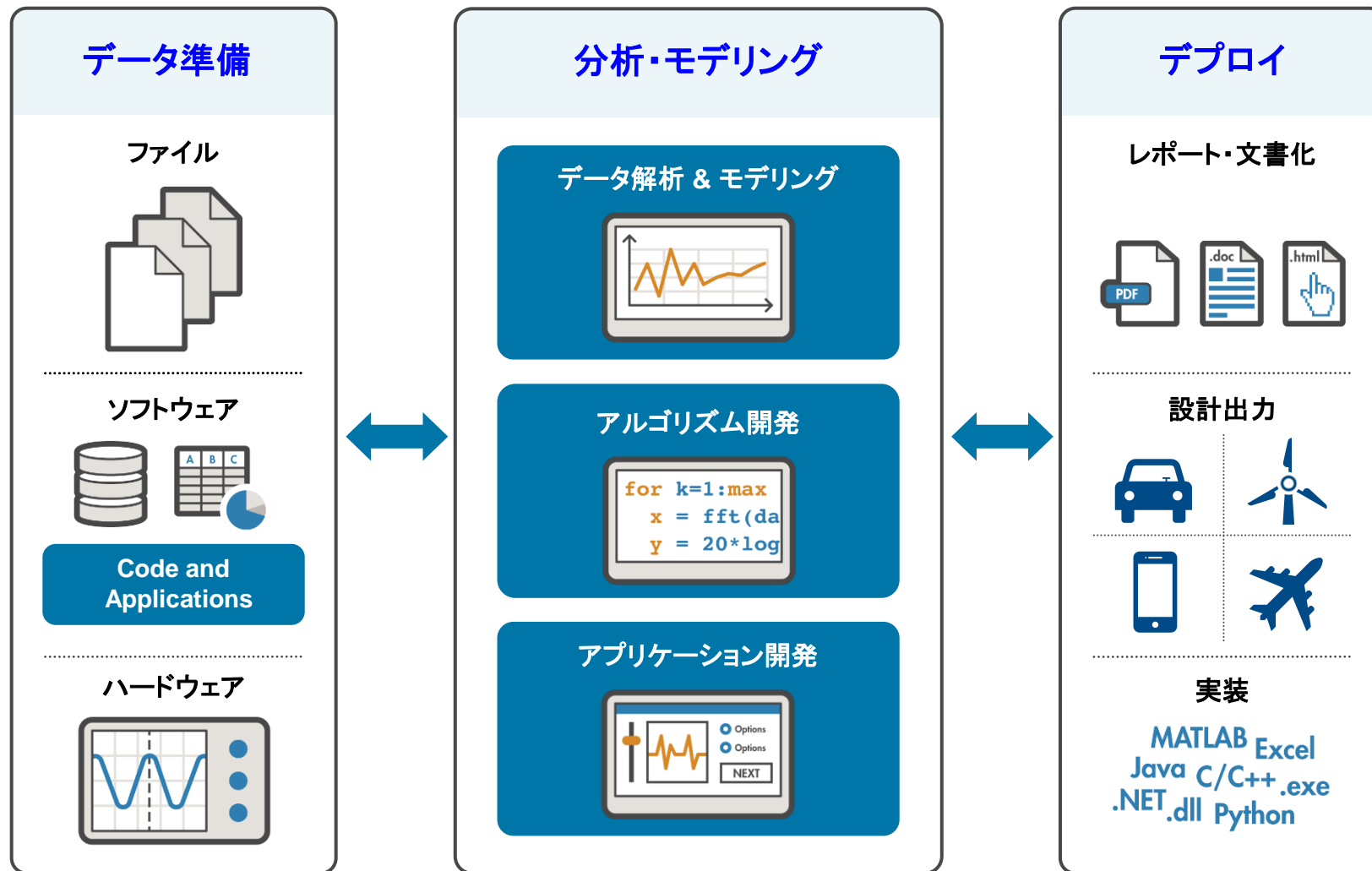
『目視による対象の検査』の自動化：正常データのみを学習に使用(教師なし学習)

異常データがあまり集められない。正常データならたくさんあるけど...



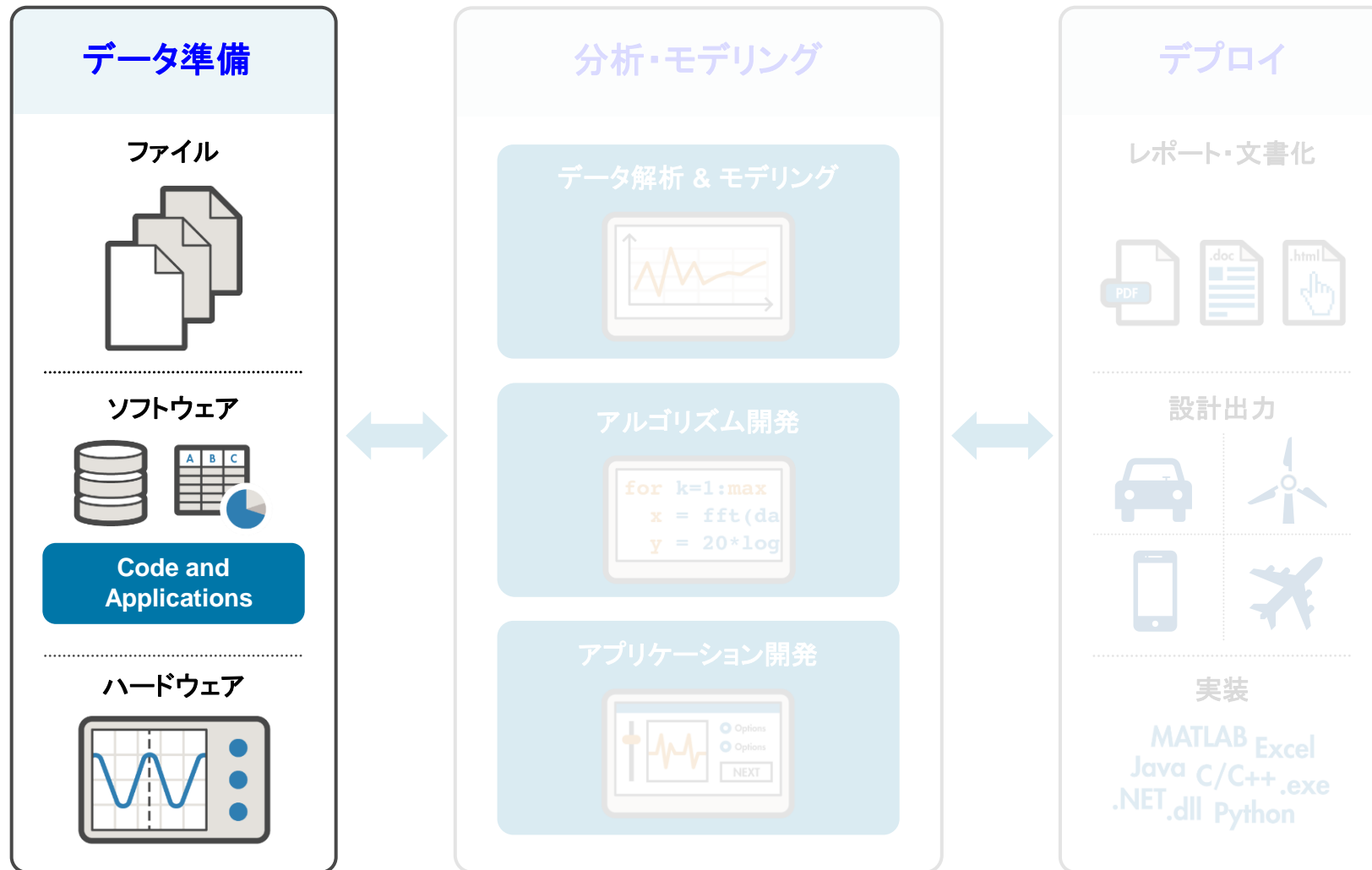
MATLABの技術計算ワークフロー

外観検査も同じワークフローで



MATLABの技術計算ワークフロー

外観検査も同じワークフローで



MATLABが提供する計測ハードウェアとのインターフェース

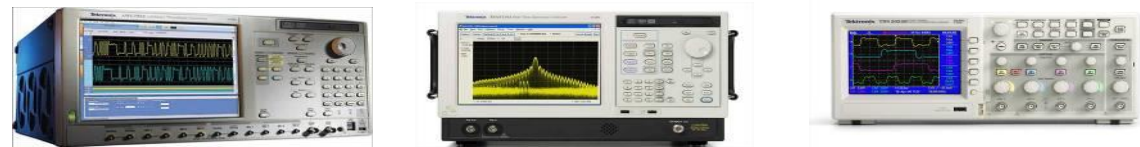
Image Acquisition Toolbox イメージキャプチャデバイス



Data Acquisition Toolbox プラグインデータ収集 カード・ボード



Instrument Control Toolbox 計測器/ RS-232 etc



MATLAB 上記ハードウェア以外との通信



カメラ接続、PLCとの通信など製造業に求められるハードウェア連携をサポート

様々な場所の様々なフォーマットのデータにアクセスする

■ タイプ

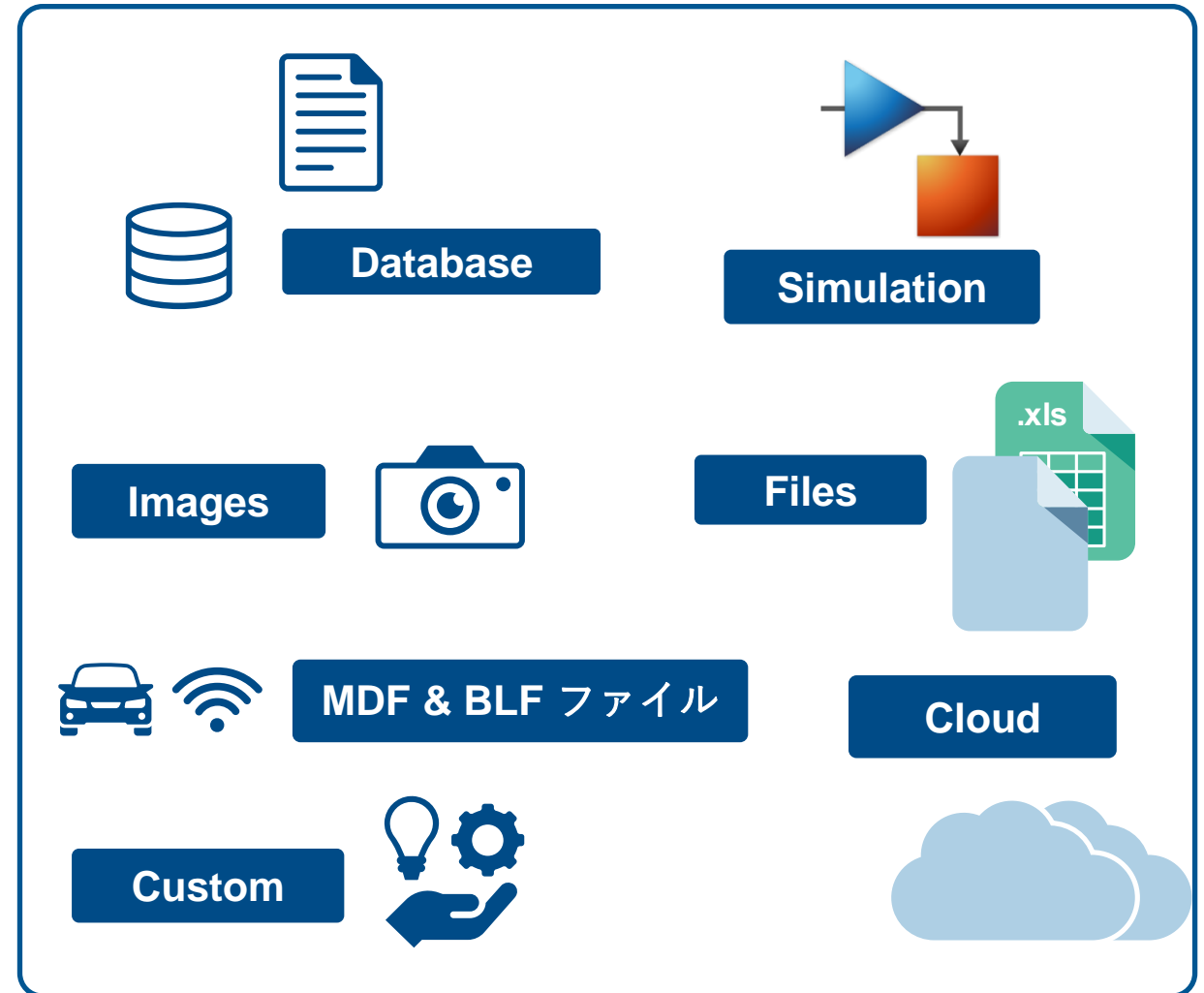
- テキスト
- バイナリ

■ 構成

- データベース
- データヒストリアン

■ 場所

- クラウド
- オンプレミス



データラベリングに時間をかけない

精度の高いラベリングに専門家の知識は必要ですが、彼らの時間もまた貴重です

領域に特化したラベリングアプリが、作業を高速化・自動化します

イメージラベラー Computer Vision Toolbox	R2017b
ビデオラベラー Computer Vision Toolbox	R2018b
信号ラベラー（オーディオ含む） Signal Processing Toolbox Audio Toolbox	R2019a
LiDARラベラー Lidar Toolbox	R2020b
イメージラベラー Big Image support	R2021a
医用画像ラベラー Medical Imaging Toolbox	R2022b

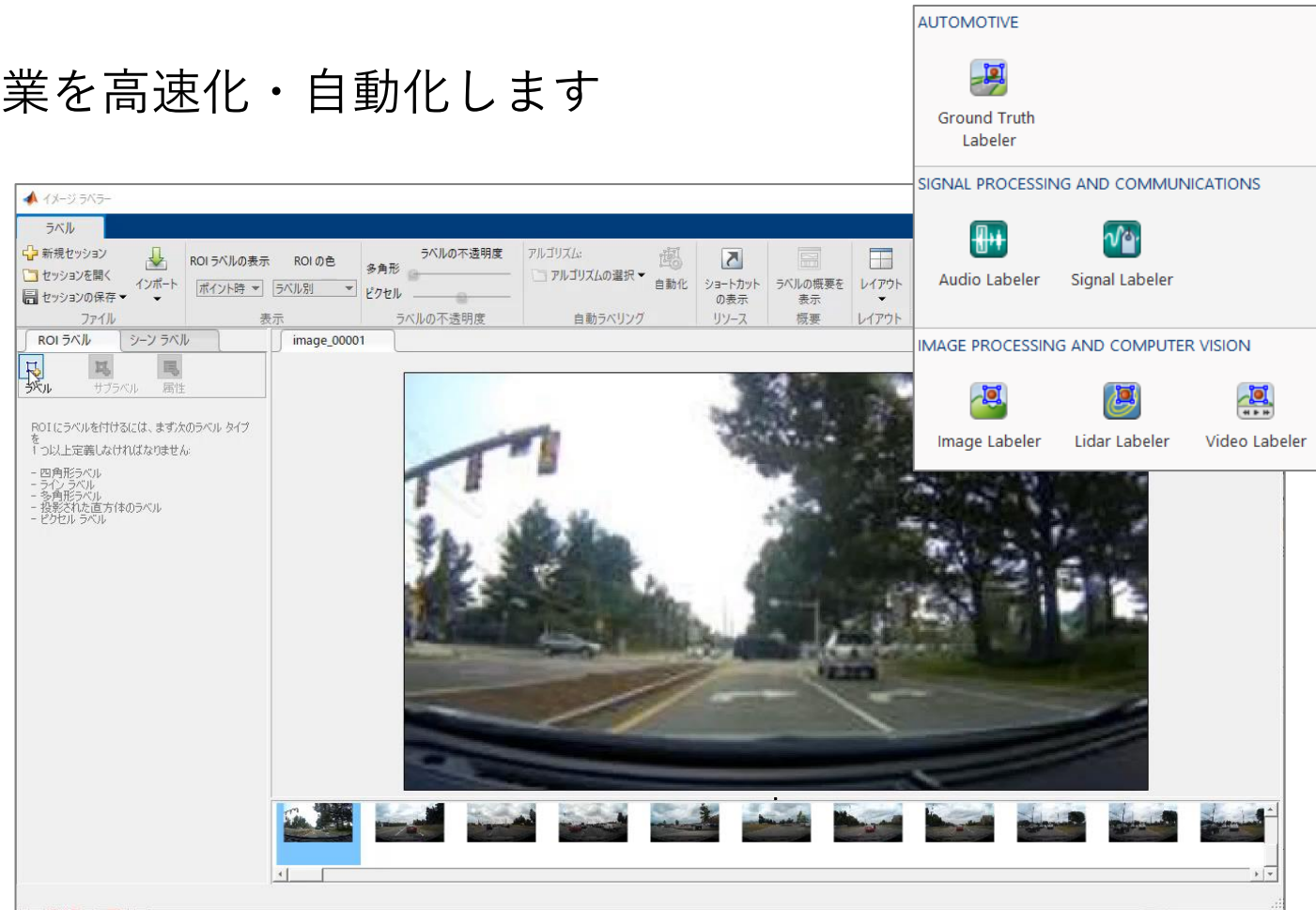
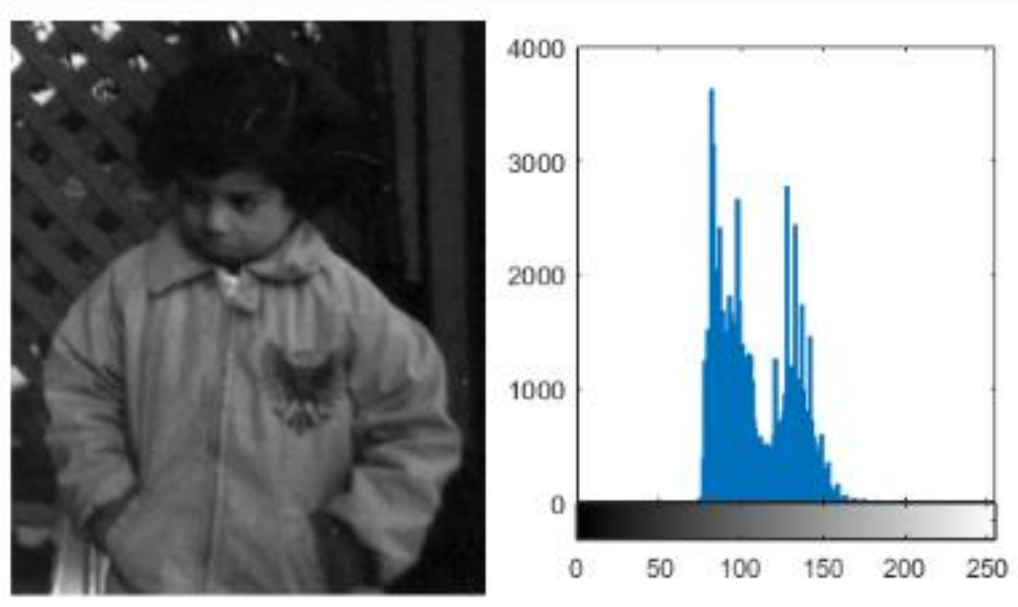
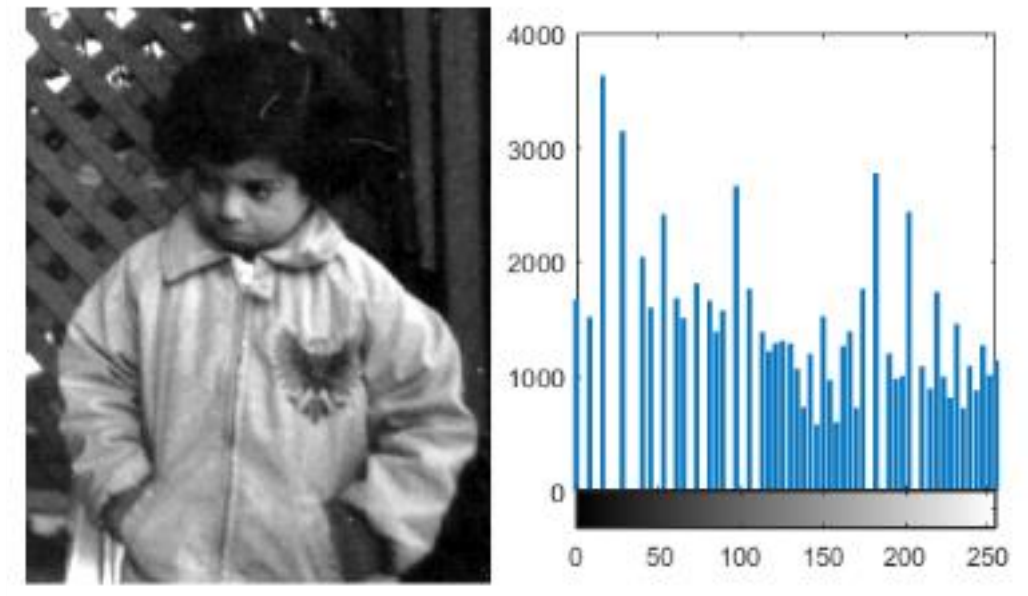


Image Labeler app

コントラスト調整

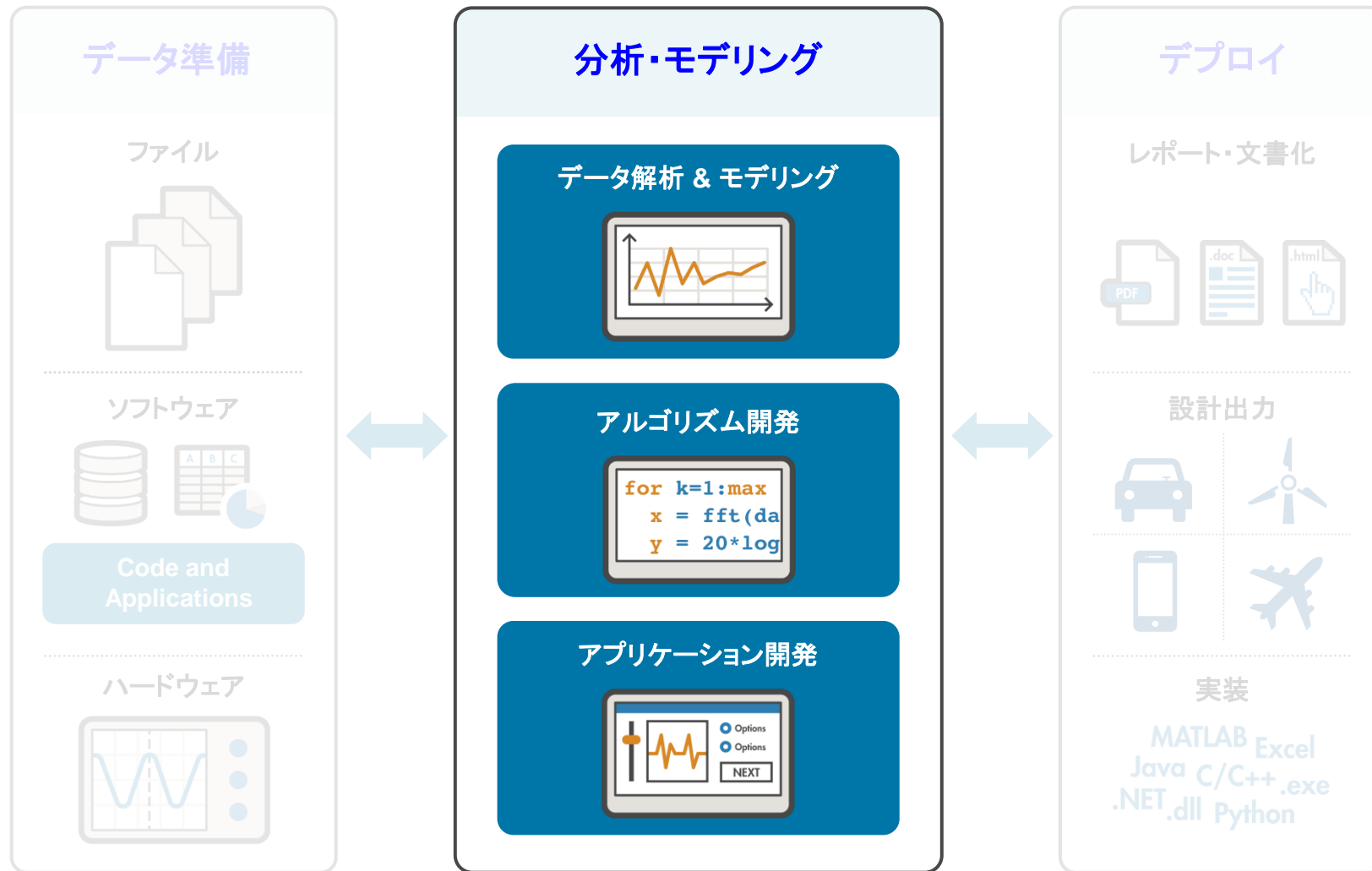


histeq,
imadjust



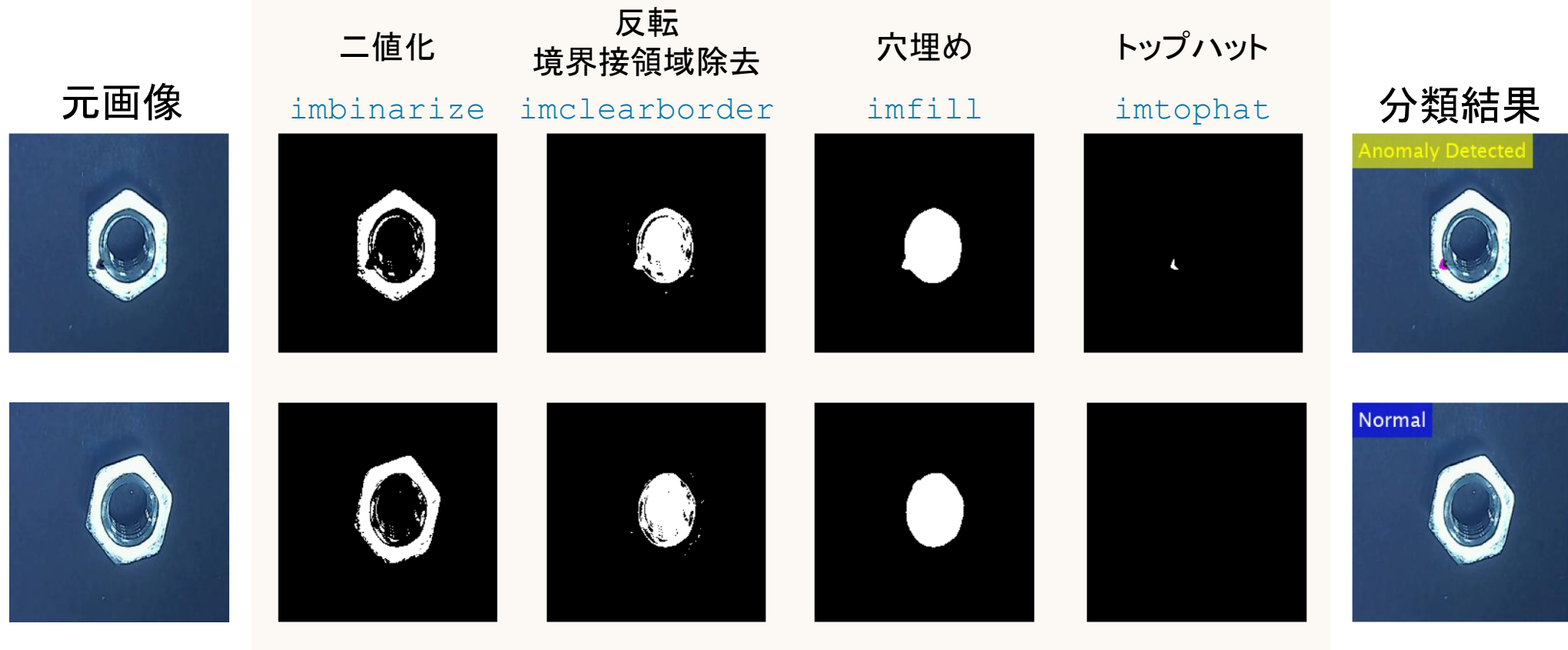
MATLABの技術計算ワークフロー

外観検査も同じワークフローで



古典的な画像処理ベース(ルールベース)の異常検出

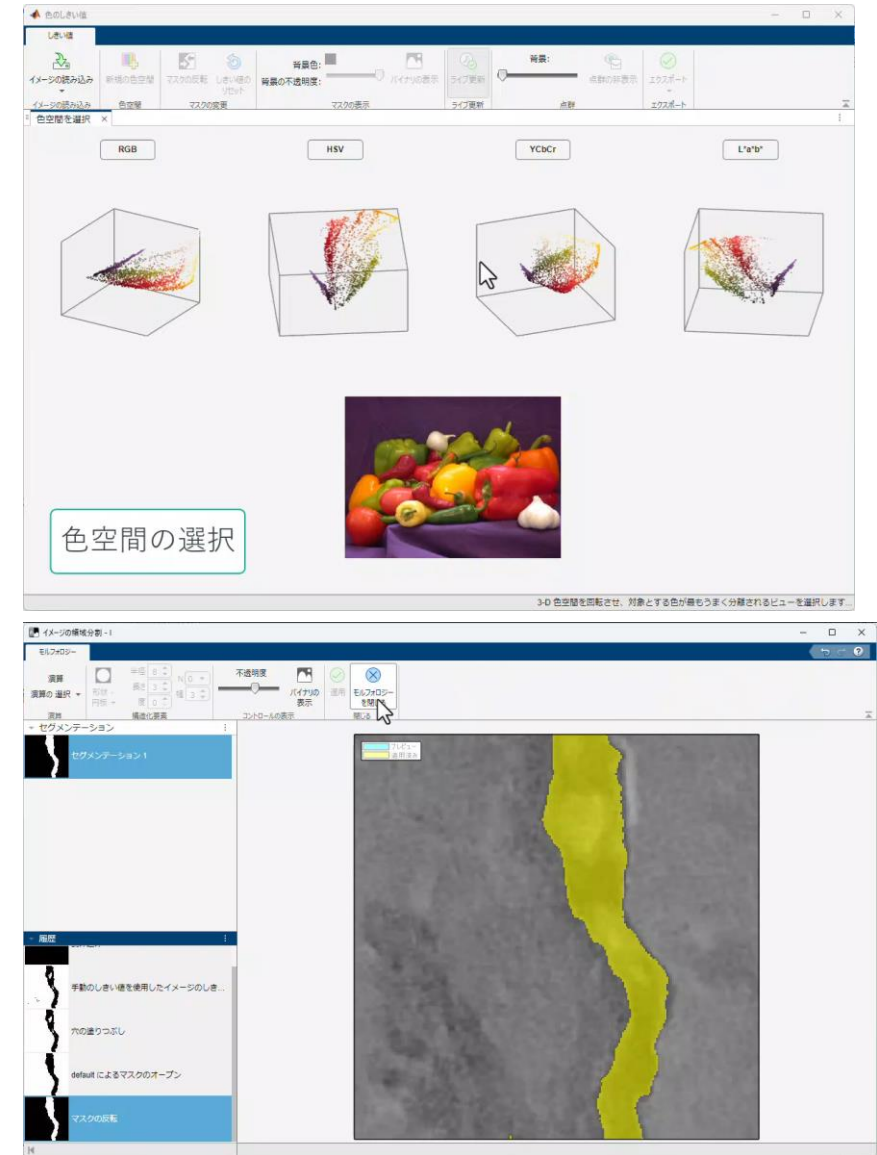
”よく使う”画像処理アルゴリズムはビルドイン



アルゴリズムを一つずつ組合わせて、正常・異常の分類が可能

画像処理関連のローコードツール

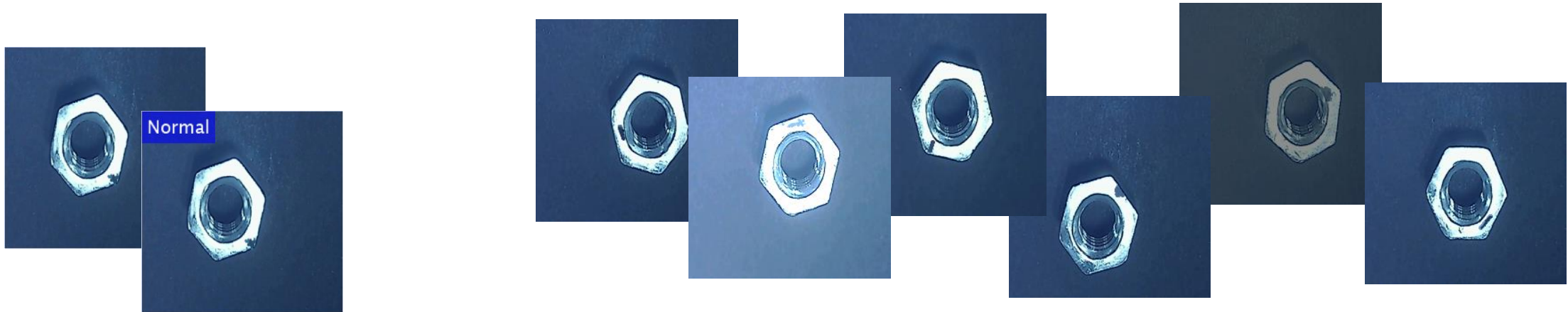
- 色のしきい値アプリ
様々な色空間上において
対話的な閾値処理によるマスク作成
- イメージの領域分割アプリ
二値化やモルフォロジー処理、グラフカット等による
対話的なセグメンテーション
- イメージの領域解析アプリ
セグメンテーション後の各領域における
面積などのプロパティ計測



対話的に結果を得て、コード生成で自動化&コード理解

画像処理(ルールベース)での異常検出の難しさ

- 異常画像は千差万別→作成したルールから少しでも外れると誤分類



- 全てに対応するルール作りは(一般的には)不可能

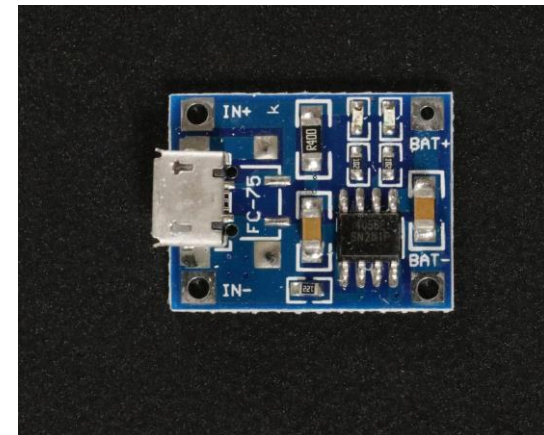
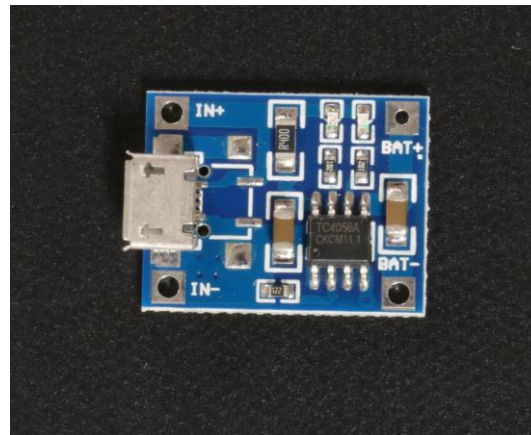
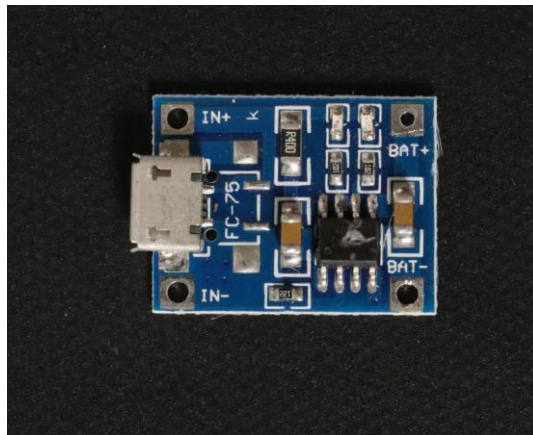
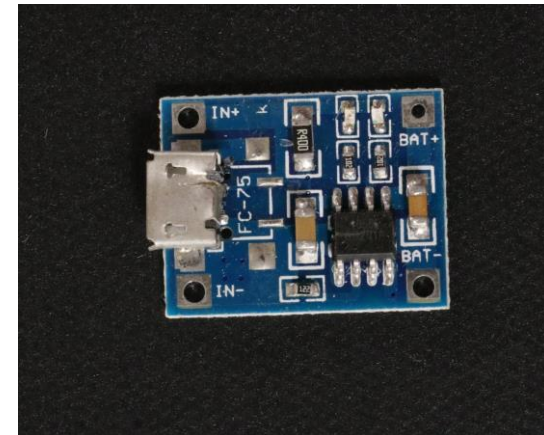
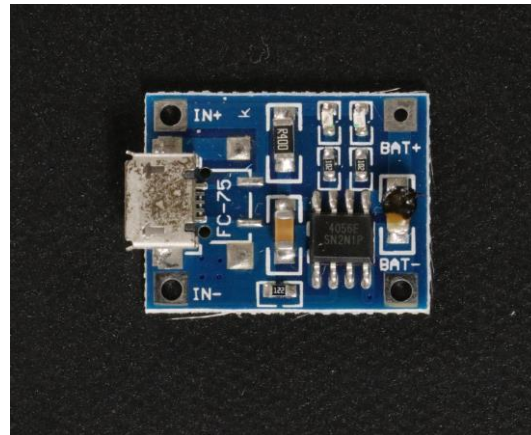
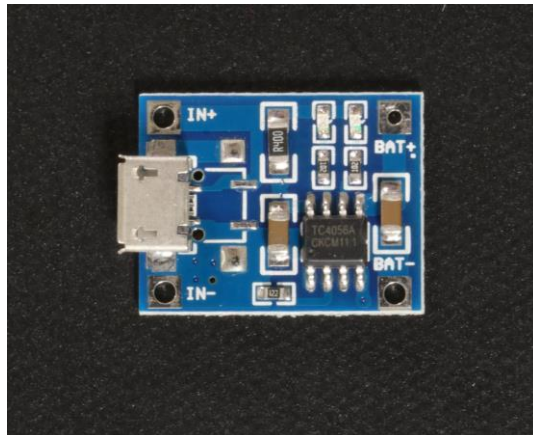


機械学習によるアプローチ

画像分類AI(正常と異常の分類)

教師有り学習

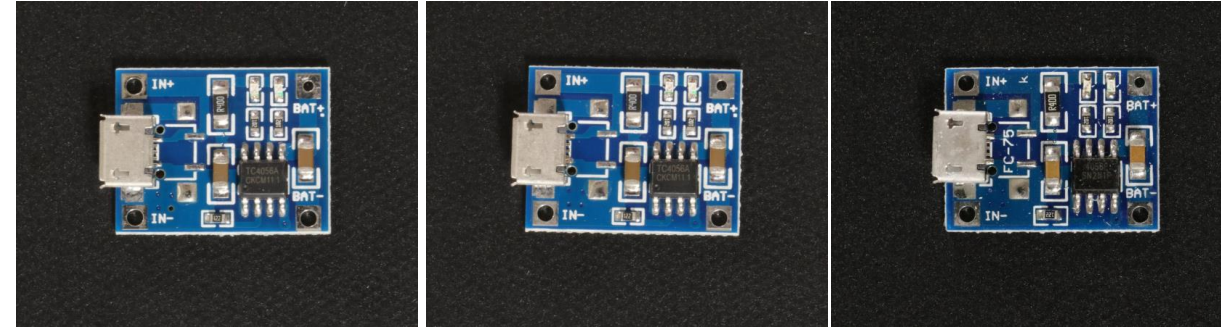
- 正常と異常の画像がそれぞれ大量に存在
- 画像に対して、「正常」「異常」がラベル付けされている(フォルダ分け)
- これらを教師データとして、正常と異常を分類するモデルを学習したい



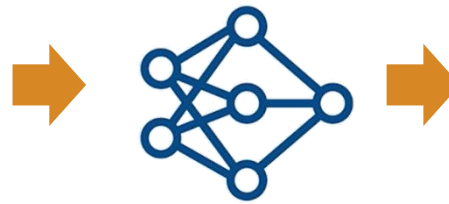
画像分類AI(正常と異常の分類)

教師有り学習

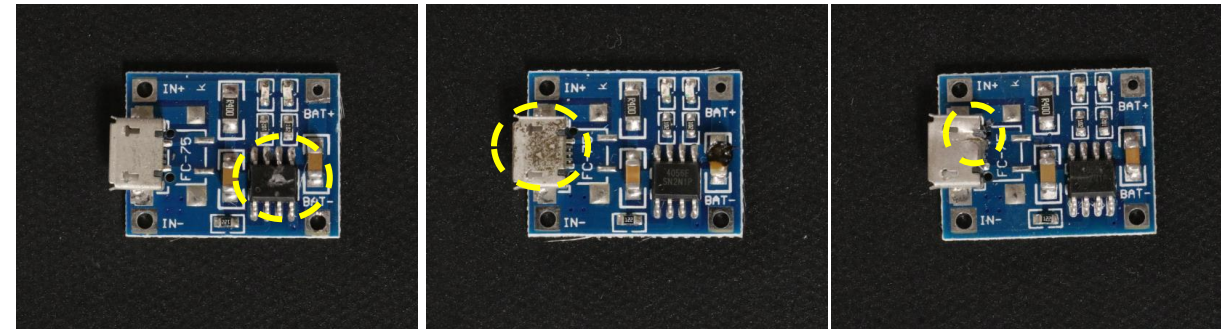
正常画像



分類CNN



異常画像



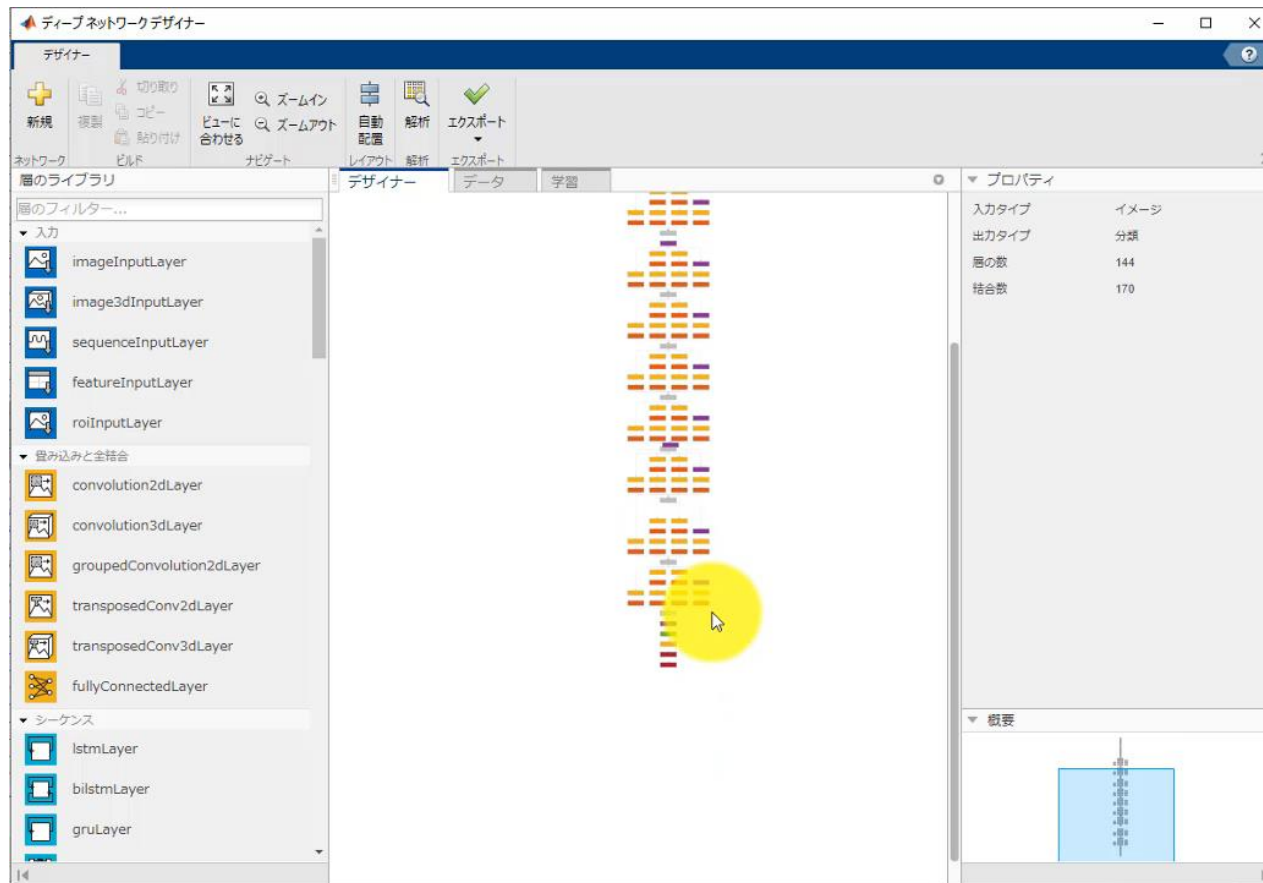
ネットワークの作成とコード自動生成

ディープネットワークデザイナー

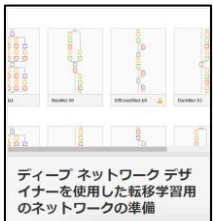
- 多数の即利用可能な学習済みモデル
- ネットワークの整合性チェック機能



- 自動コード生成機能



画像分類モデル学習の流れ



ディープ ネットワーク デザイナーを使用した転移学習用のネットワークの準備

ディープ ネットワーク デザイナーを使用した転移学習用のネットワークの準備

R2024a

この例では、ディープ ネットワーク デザイナー アプリを使用して転移学習用のネットワークを対話形式で準備する方法を示します。

この例を開く

Copy Command

転移学習は、事前学習済みの深層学習ネットワークを利用して、新しいタスクを学習するように微調整するプロセスです。通常は、転移学習を使用する方が、ネットワークにゼロから学習させるよりも簡単で時間もかかりません。少量のデータを使用して、新しいタスクに学習済みの特徴を高速に転移できます。

データの解凍

MathWorks Merch データセットを抽出します。これは、75 個の MathWorks の商品イメージから成る小さなデータセットであり、5 つの異なるクラス (cap, cube, playing cards, screwdriver, torch) に属します。データは、これら 5 つのクラスに対応するサブフォルダーにイメージが置かれる配置になっています。

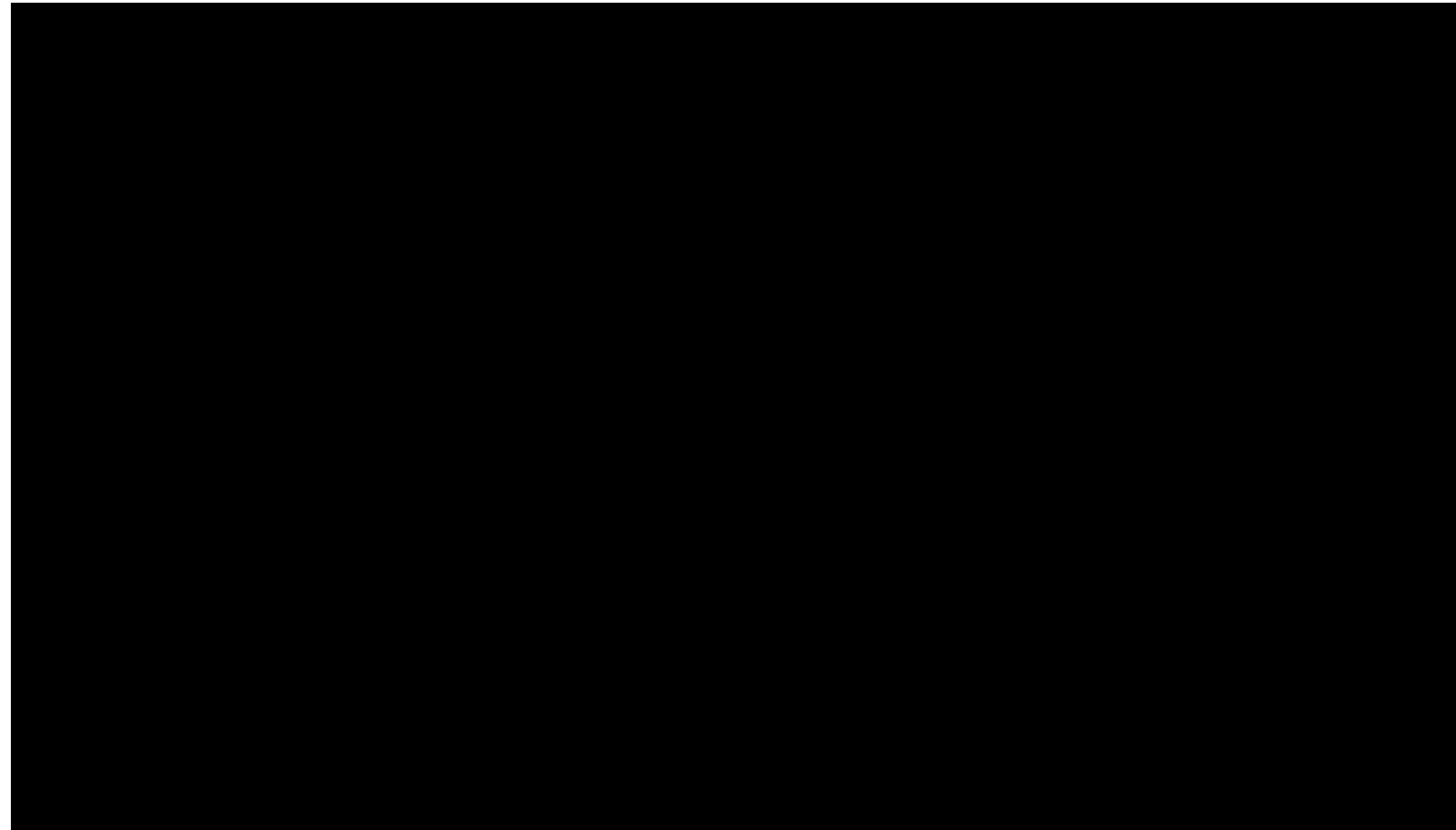
```
folderName = "MerchData";  
unzip("MerchData.zip", folderName);
```

Get ▾

イメージ データストアを作成します。イメージ データストアを使用すると、イメージ データの大規模なコレクション (メモリに収まらないデータなど) を格納し、ニューラル ネットワークの学習中にイメージをバッチ単位で効率的に読み取ることができます。抽出したイメージがあるフォルダーを指定し、サブフォルダー名がイメージ ラベルに対応していることを示します。

```
imds = imageDatastore(folderName, ...  
    IncludeSubfolders=true, ...  
    LabelSource="foldernames");
```

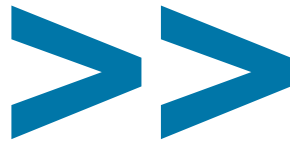
Get ▾



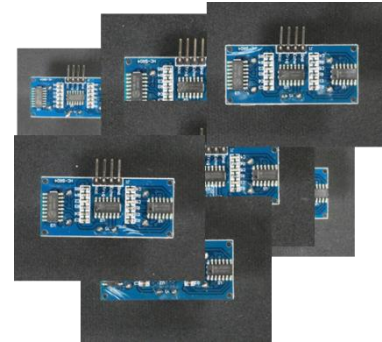
ヘルプドキュメントから必要な部分だけを抽出して高速にコードを実現

外観検査におけるよくある課題

正常画像



異常画像



異常画像が少なくて
教師あり学習ができない



Automated Visual Inspection ライブラリ

Computer Vision Toolboxのアドオン 教師なし学習をカバー

- リリース: R2022b~
- 外観検査AI開発を加速する機能を提供
 - [FCDD](#)
 - 異常とするしきい値の計算
 - 検証用のUIツール
 - アプリ化が可能
- 機能アップデート

- [PatchCore](#)

- [FastFlow](#)

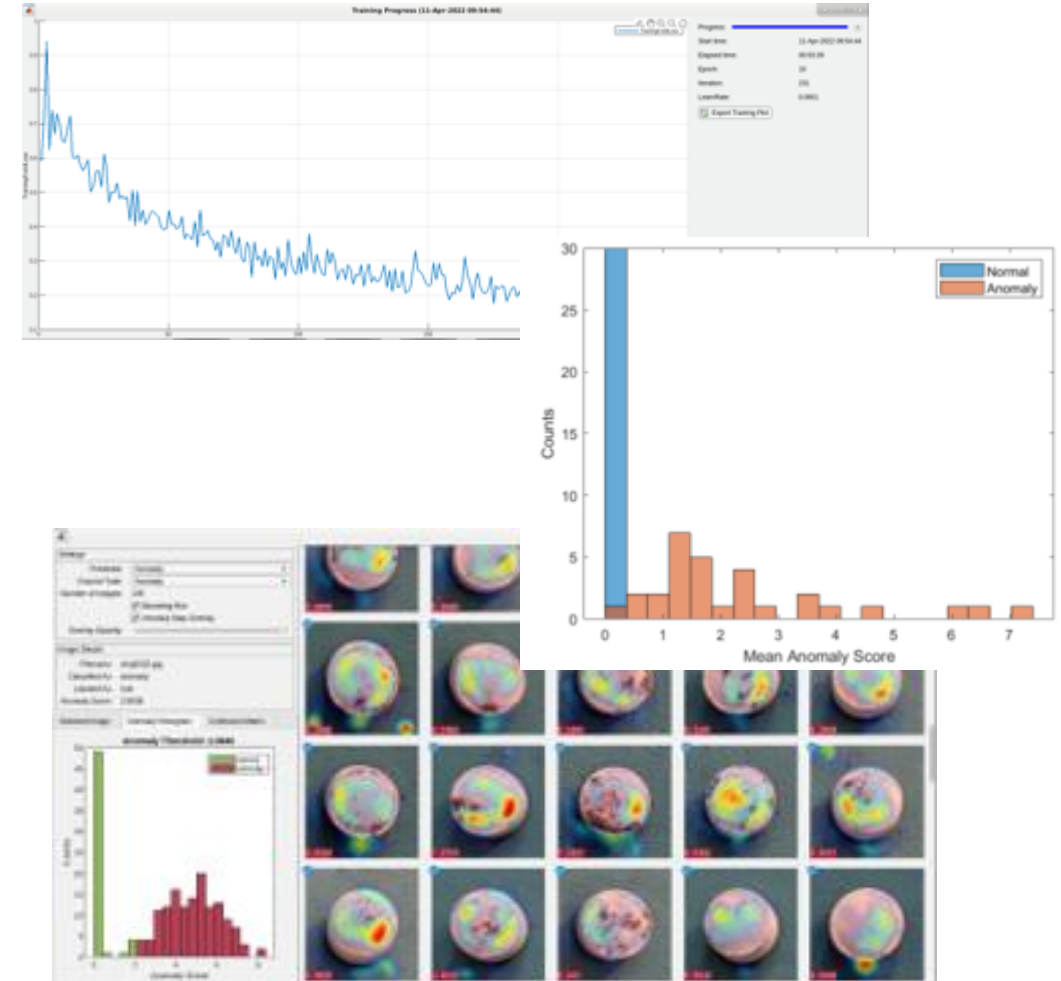
- Cコード・GPUコード生成対応

- [YOLO X](#)

- [Efficient AD](#) (R2024b Prereleaseで利用可能)

R2023a

R2023b



教師なし外観検査AIモデル開発ワークフロー

学習用の
画像準備

モデル
学習

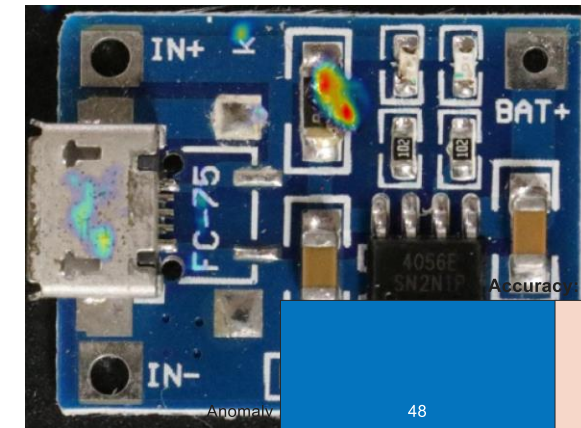
しきい値
決定

検証

展開・
実装

PatchCore
FastFlow
EfficientAD
など

anomalyThreshold
関数



True Class

Normal

Anomaly

Normal

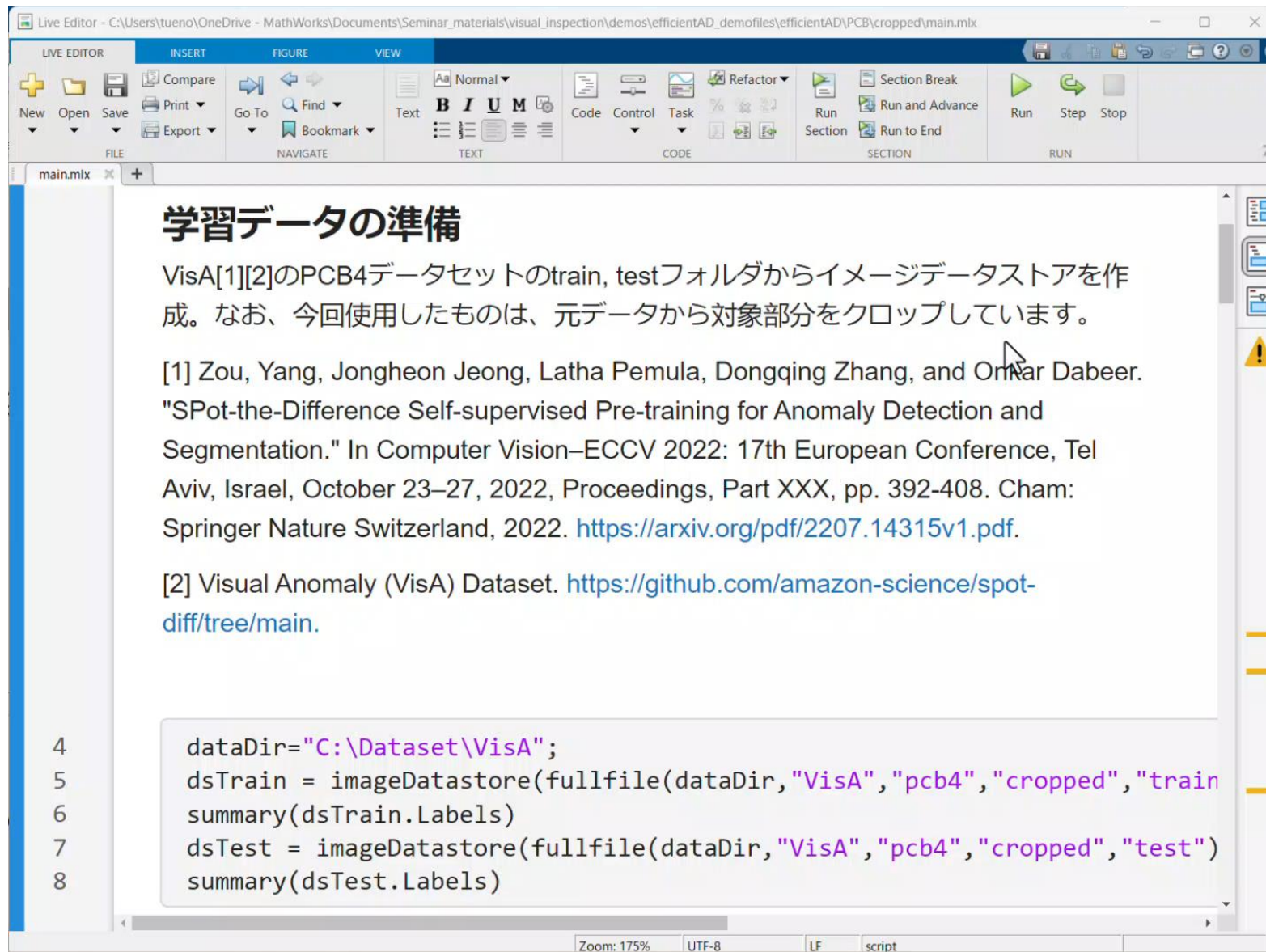
Predicted Class

	Anomaly	Normal
Normal	48	2
Anomaly	1	49

[Getting Started with Anomaly Detection Using Deep Learning](#)

学習用の
画像準備モデル
学習しきい値
決定

検証

展開・
実装

Live Editor - C:\Users\tueno\OneDrive - MathWorks\Documents\Seminar_materials\visual_inspection\demos\efficientAD_demo\files\efficientAD\PCB\cropped\main.mlx

FILE NAVIGATE TEXT CODE SECTION RUN

学習データの準備

VisA[1][2]のPCB4データセットのtrain, testフォルダからイメージデータストアを作成。なお、今回使用したものは、元データから対象部分をクロップしています。

[1] Zou, Yang, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Omar Dabeer. "SPot-the-Difference Self-supervised Pre-training for Anomaly Detection and Segmentation." In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX, pp. 392-408. Cham: Springer Nature Switzerland, 2022. <https://arxiv.org/pdf/2207.14315v1.pdf>.

[2] Visual Anomaly (VisA) Dataset. <https://github.com/amazon-science/spot-diff/tree/main>.

```
4 dataDir="C:\Dataset\VisA";
5 dsTrain = imageDatastore(fullfile(dataDir,"VisA","pcb4","cropped","train
6 summary(dsTrain.Labels)
7 dsTest = imageDatastore(fullfile(dataDir,"VisA","pcb4","cropped","test")
8 summary(dsTest.Labels)
```

Zoom: 175% UTF-8 LF script

学習用の
画像準備モデル
学習しきい値
決定

検証

展開・
実装

The screenshot displays the MATLAB Live Editor interface. The top toolbar includes tabs for LIVE EDITOR, INSERT, FIGURE, and VIEW. The LIVE EDITOR tab is active, showing a code editor with the following content:

```
20 dsTest = transform(dsTest,resizeFcn);
```

Below the code editor, there is a section titled "EfficientAD 異常検出器のネットワークアーキテクチャを定義" (Define EfficientAD Anomaly Detector Network Architecture). This section contains the following code:

```
21 % detector=patchCoreAnomalyDetector;
```

```
22 detector=efficientADAnomalyDetector;
```

Below this code, there is a section titled "検出器の学習" (Learning the Detector). This section contains the following text:

学習を行う場合、doTraining 変数を true に指定します。
trainEfficientADAnomalyDetector 関数によって上で定義した検出器を学習します。

falseの場合、事前に学習しておいた検出器をロードします。

```
23 doTraining = false;
```

```
24
```

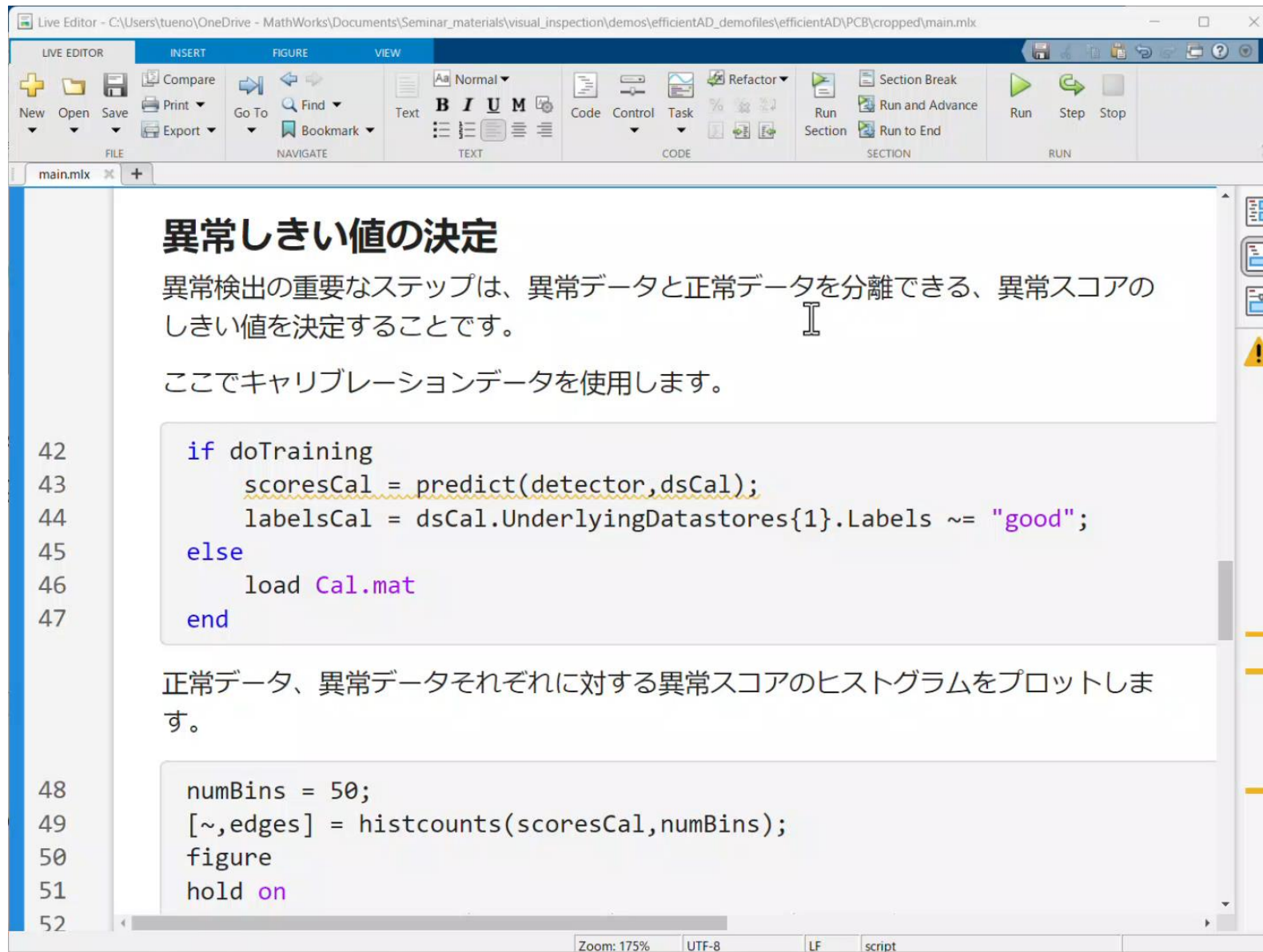
```
25 opts=trainingOptions('adam',...
```

```
26 'MaxEpochs' 30
```

The bottom status bar shows "Zoom: 175%", "UTF-8", "LF", and "script".

学習用の
画像準備モデル
学習しきい値
決定

検証

展開・
実装

Live Editor - C:\Users\tueno\OneDrive - MathWorks\Documents\Seminar_materials\visual_inspection\demos\efficientAD_demo\files\efficientAD\PCB\cropped\main.mlx

FILE NAVIGATE TEXT CODE SECTION RUN

異常しきい値の決定

異常検出の重要なステップは、異常データと正常データを分離できる、異常スコアのしきい値を決定することです。

ここでキャリブレーションデータを使用します。

```
42 if doTraining
43     scoresCal = predict(detector, dsCal);
44     labelsCal = dsCal.UnderlyingDatastores{1}.Labels ~= "good";
45 else
46     load Cal.mat
47 end
```

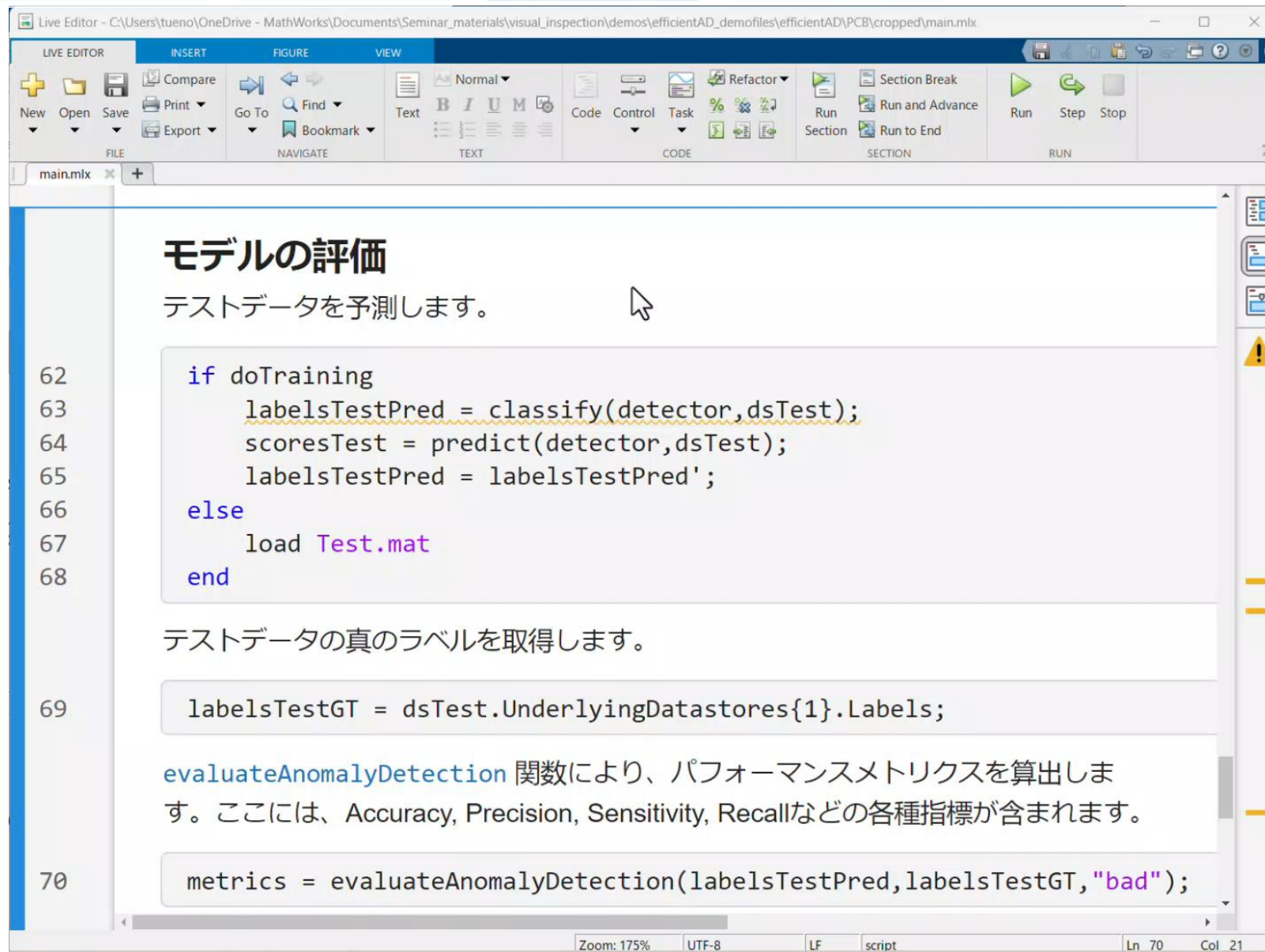
正常データ、異常データそれぞれに対する異常スコアのヒストグラムをプロットします。

```
48 numBins = 50;
49 [~,edges] = histcounts(scoresCal,numBins);
50 figure
51 hold on
52
```

Zoom: 175% UTF-8 LF script

学習用の
画像準備モデル
学習しきい値
決定

検証

展開・
実装

The screenshot shows the MATLAB Live Editor interface. The title bar indicates the file path: C:\Users\tueno\OneDrive - MathWorks\Documents\Seminar_materials\visual_inspection\demofiles\efficientAD_demo\files\efficientAD\PCB\cropped\main.mlx. The editor has tabs for LIVE EDITOR, INSERT, FIGURE, and VIEW. The LIVE EDITOR tab is active, showing a script with Japanese comments and MATLAB code. The script is titled 'モデルの評価' (Model Evaluation) and describes testing data prediction. The code includes an if-statement for training and a call to the evaluateAnomalyDetection function. The status bar at the bottom shows 'Zoom: 175%', 'UTF-8', 'LF', 'script', 'Ln 70', and 'Col 21'.

```
Live Editor - C:\Users\tueno\OneDrive - MathWorks\Documents\Seminar_materials\visual_inspection\demofiles\efficientAD_demo\files\efficientAD\PCB\cropped\main.mlx
```

モデルの評価

テストデータを予測します。

```
62 if doTraining
63     labelsTestPred = classify(detector,dsTest);
64     scoresTest = predict(detector,dsTest);
65     labelsTestPred = labelsTestPred';
66 else
67     load Test.mat
68 end
```

テストデータの真のラベルを取得します。

```
69 labelsTestGT = dsTest.UnderlyingDatastores{1}.Labels;
```

evaluateAnomalyDetection 関数により、パフォーマンスメトリクスを算出します。ここには、Accuracy, Precision, Sensitivity, Recallなどの各種指標が含まれます。

```
70 metrics = evaluateAnomalyDetection(labelsTestPred,labelsTestGT,"bad");
```

Zoom: 175% UTF-8 LF script Ln 70 Col 21

検証のためのGUI

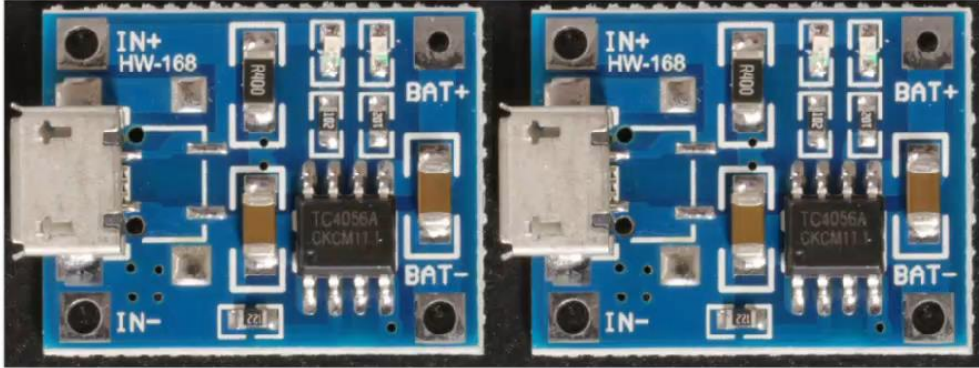
Live Editor - C:\Users\tueno\OneDrive - MathWorks\Documents\Seminar_materials\visual_inspection\demos\efficientAD_demo\files\efficientAD\PCB\cropped\main.mlx

LIVE EDITOR INSERT VIEW

New Open Save Compare Print Export Go To Find Bookmark Text Normal B I U M Code Control Task Refactor Run Section Run and Advance Run Step Stop Run to End

main.mlx x +

```
79 overlay_pos=anomalyMapOverlay(Itest,anomap,'MapRange',[0.1 0.5]);  
80 figure  
81 imshowpair(Itest,overlay_pos,'montage')
```



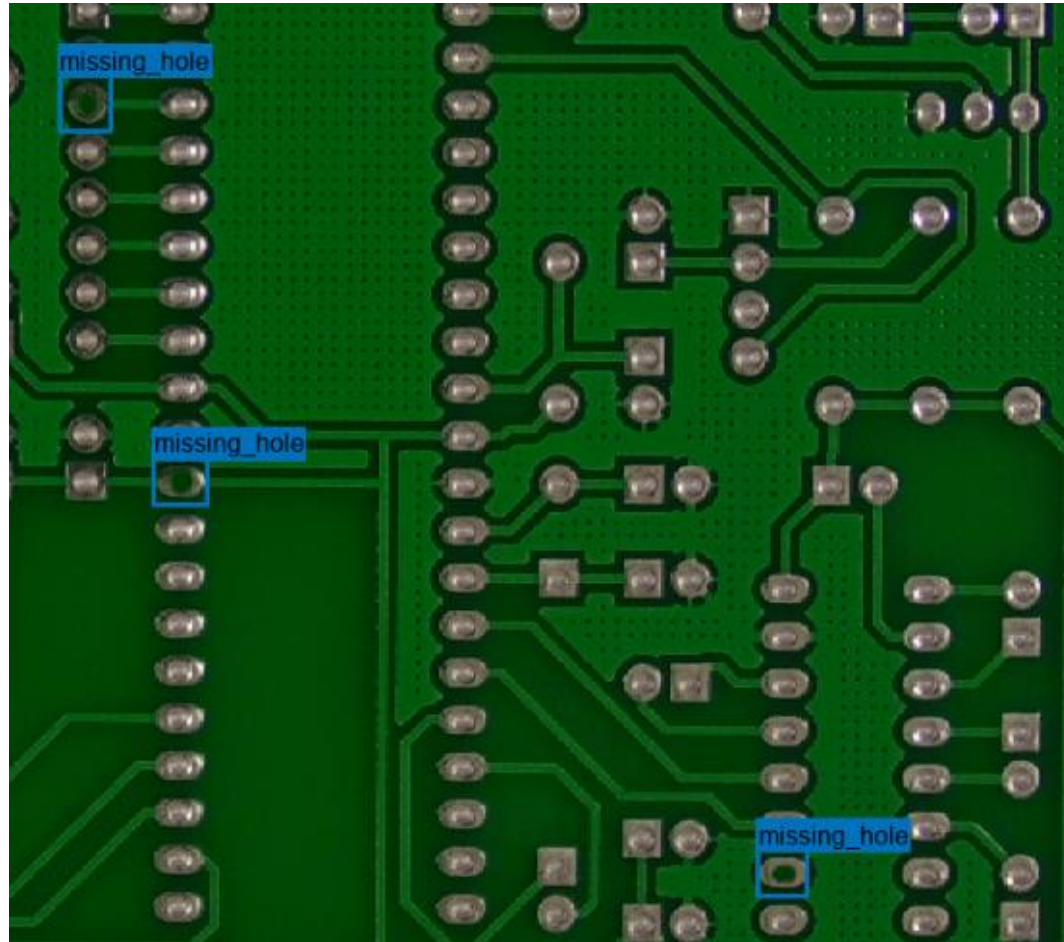
GUIでの確認

```
82 viewAnomalyDetectionResults(detector,dsTest,labelsTestGT,scoresTest,"bad")
```

Zoom: 175% UTF-8 LF script Ln 82 Col 18

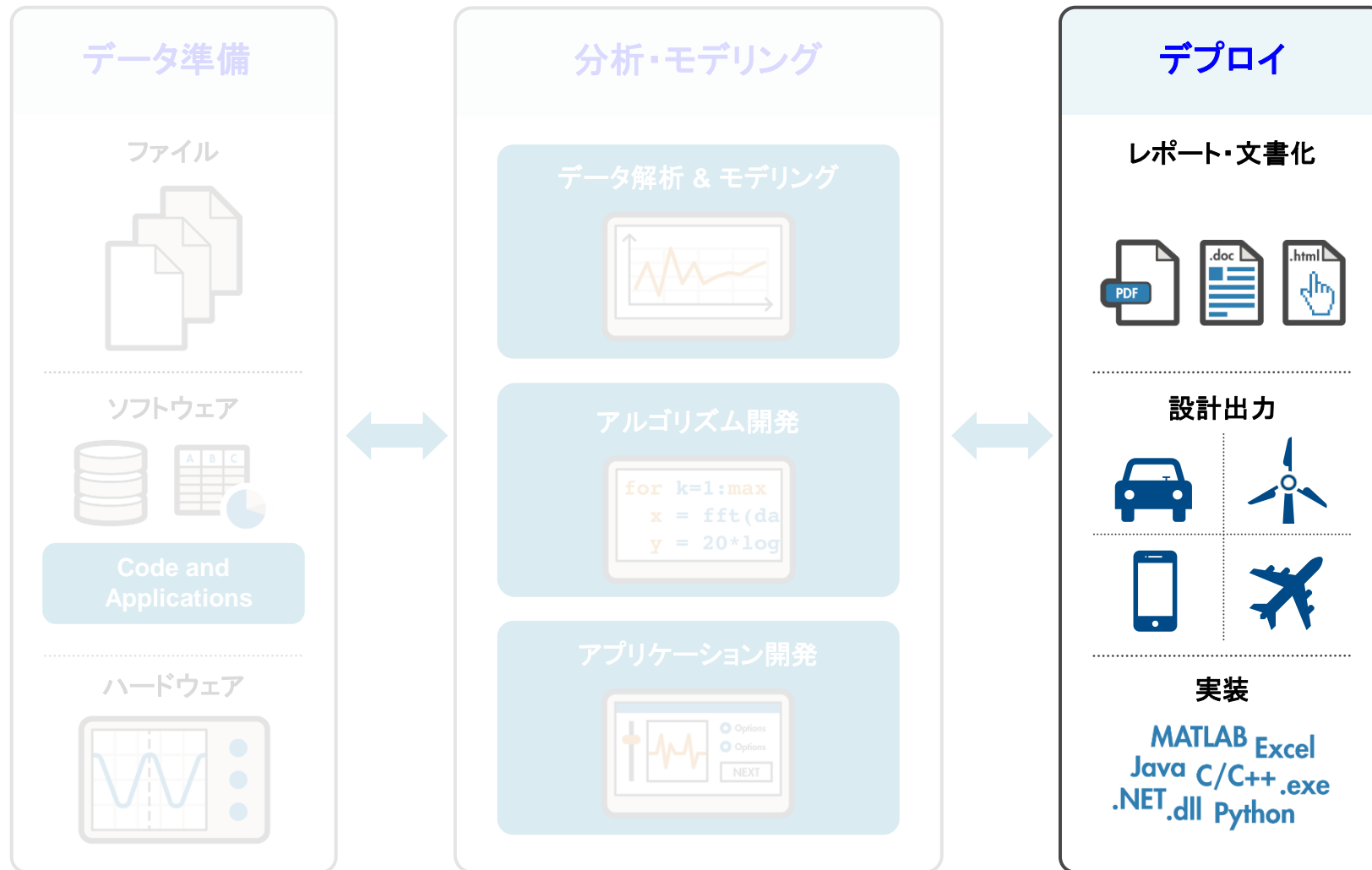
異常個所を直接検出も

YOLOX(教師あり学習物体検出器)による異常個所検出(小さな異常にも対応)



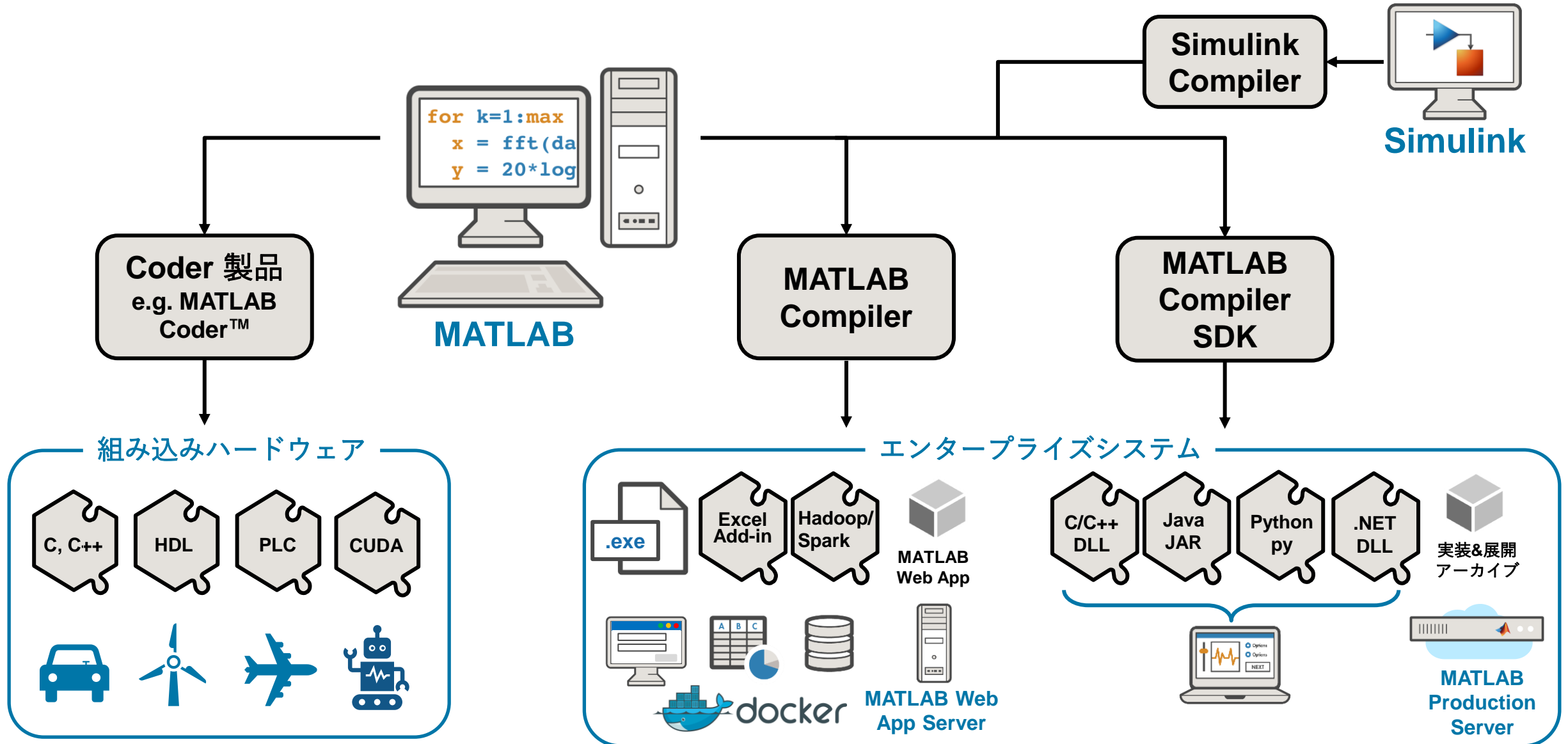
MATLABの技術計算ワークフロー

外観検査も同じワークフローで



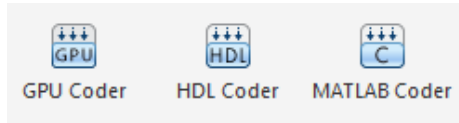
非MATLABユーザーへの成果物の展開・実装

Share



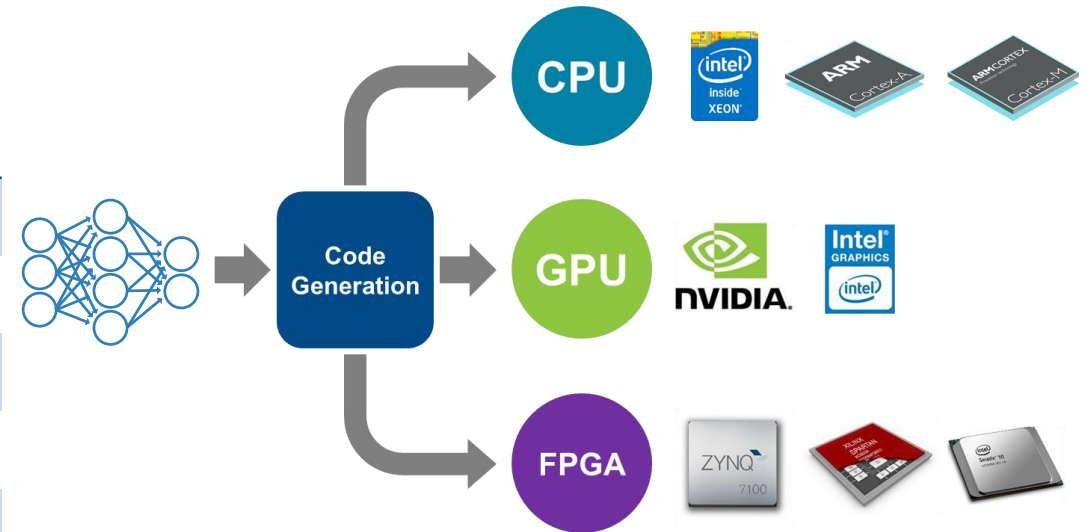
AI モデルを組み込み機器に実装する - 自動コード生成

学習したモデルを低コストの試作ボードでテストしたり、再コーディングせずに製品化



最適化されたネイティブコードを生成:

GPU – GPU Coder	R2017b
CPU – MATLAB Coder	R2018b
FPGA – Deep Learning HDL Coder	R2020b
標準化C/C++コード生成	R2021a
FPGA – HDL Coder	



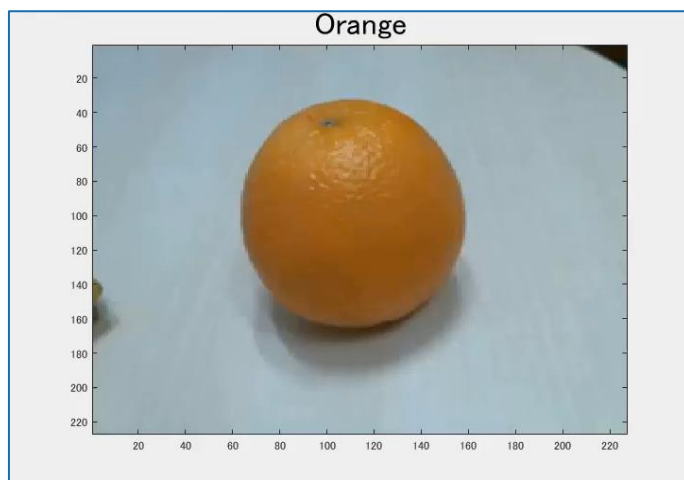
低コストのボードでモデルをテスト:

Raspberry Pi	R2019a
ARM Targets	



2クリックで *.exe作成 & 配布

- MATLABプログラムからの
 - スタンドオンアプリ作成
 - Webアプリ作成
 - Excelアドイン作成
 - 暗号化されたコード
 - 配布は無料
- “10行でできるディープラーニング”



エンジニア・研究者のための GUI 作成アプリ

- 外部へのアプリ配布の方法

- スタンドアロンアプリ
- Webアプリ

 共有

 実行

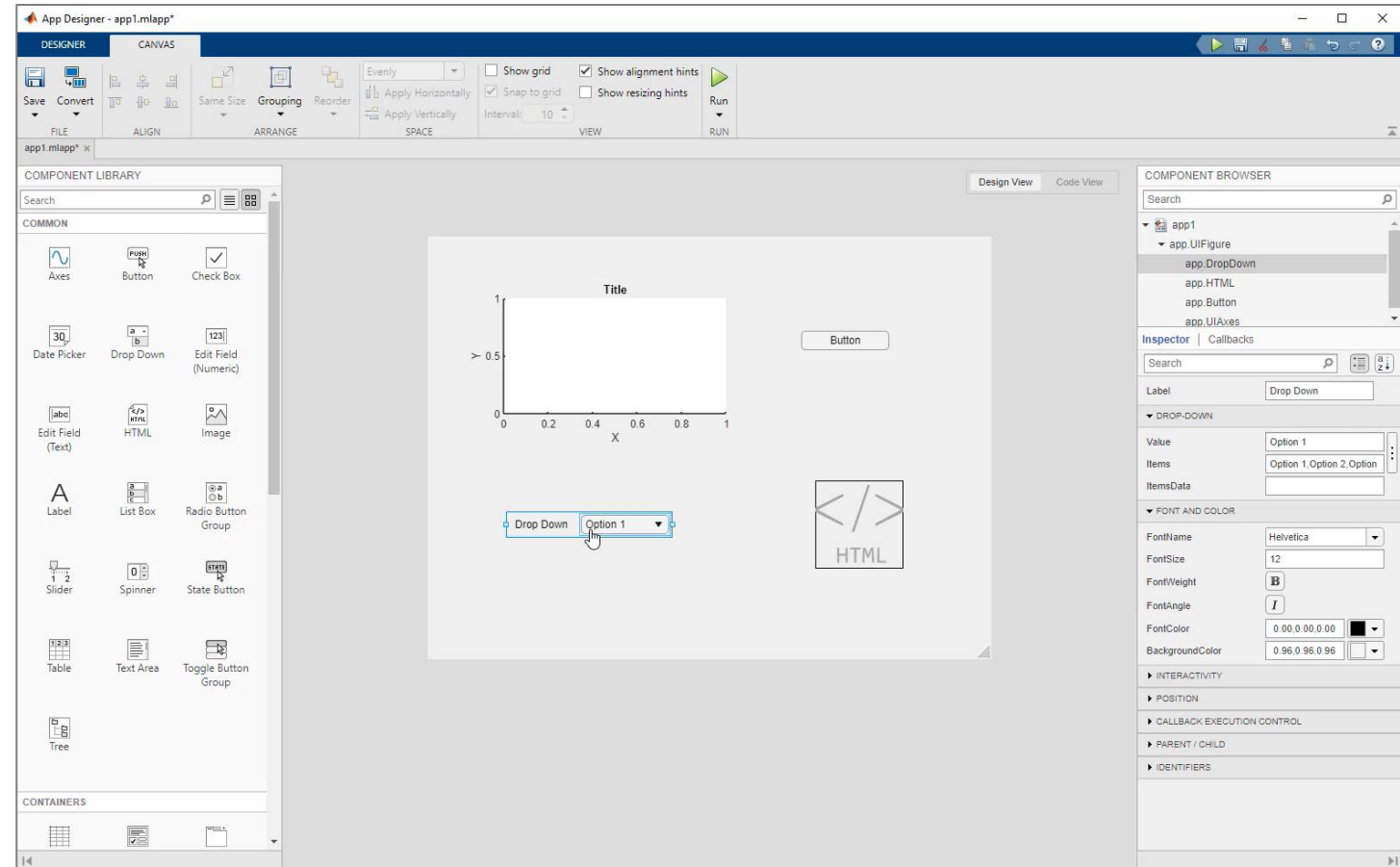
MATLAB アプリ
MATLAB ユーザーとアプリを共有するためのアプリ インストール ファイルを作成します

 **Web アプリ** **MATLAB Compiler™**
MATLAB Compiler を使用して配布用 Web アプリを作成します

 **スタンドアロンのデスクトップ アプリ** **MATLAB Compiler™**
MATLAB Compiler を使用してスタンドアロンのデスクトップ アプリケーションを作成します

- コンポーネント配置でアプリ作成

- コンポーネントライブラリ
- コールバック関数



App Designer

成果物をウェブアプリとして展開

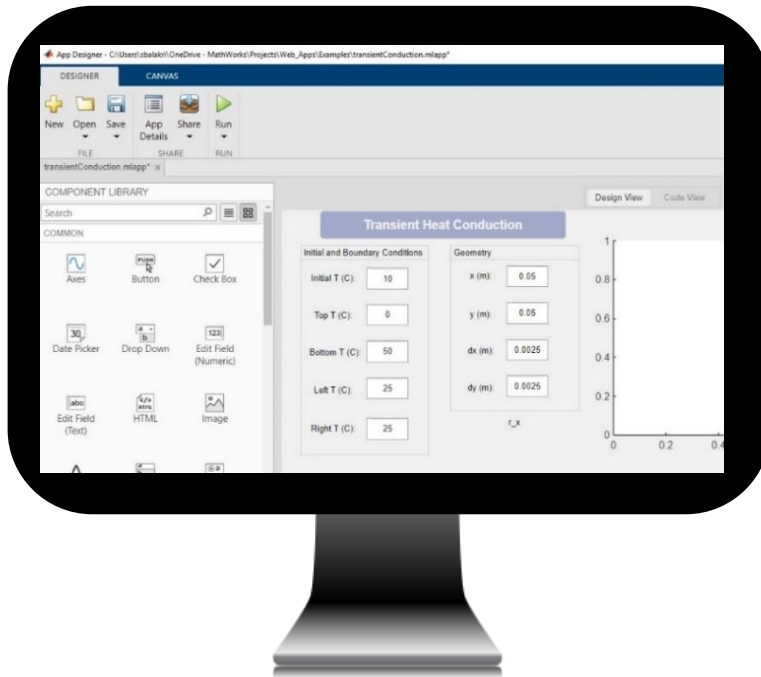
アプリを作成



ホストし、アプリを共有



アクセスし、アプリ実行



App Designer



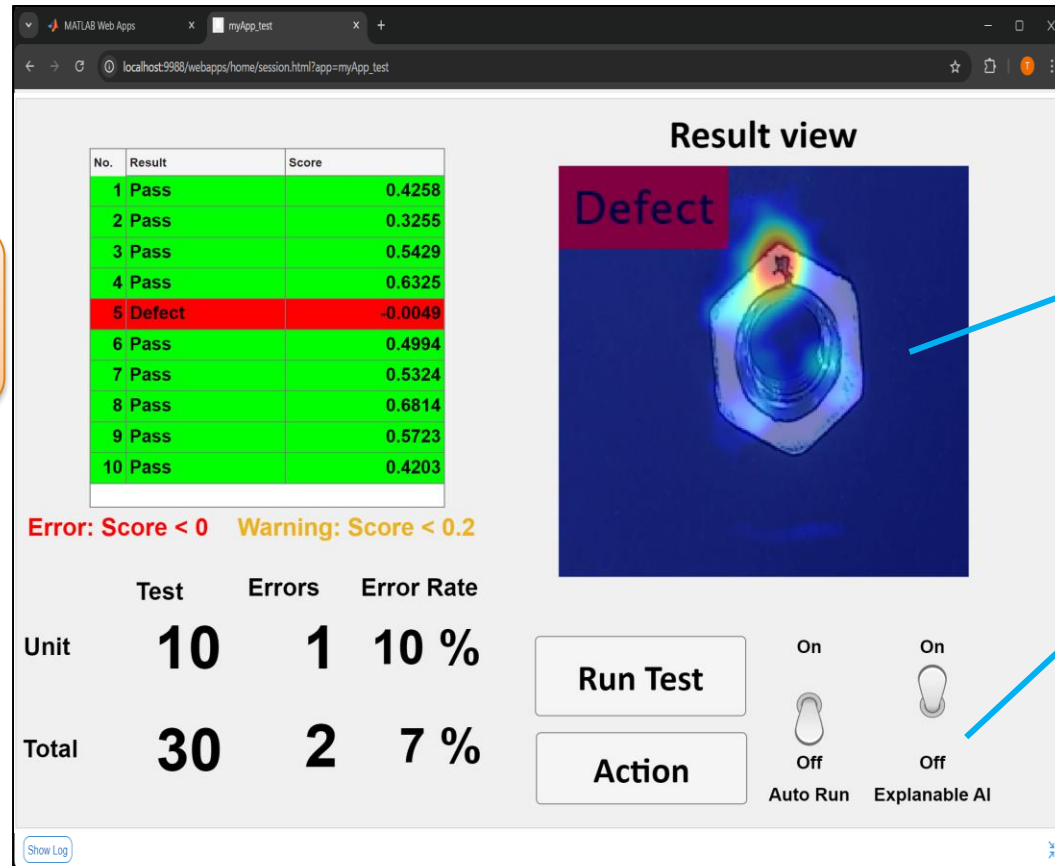
MATLAB Web App Server



作成したGUIをexeやWeb Appとして配布 human-in-the-loop workflow

結果によって
色分け

結果の
チェック



画像と異常値マップの
オーバーレイ表示

オーバーレイ表示
のオン/オフ

MATLAB Compiler™ / MATLAB Compiler SDK™

Standalone
Application

無償で配布

ドメインエキスパートの方が、
マウス操作だけでAIツールを利用可能に！

三井化学、TensorFlowとMATLABを使用してAI・自動化技術を現場展開

課題

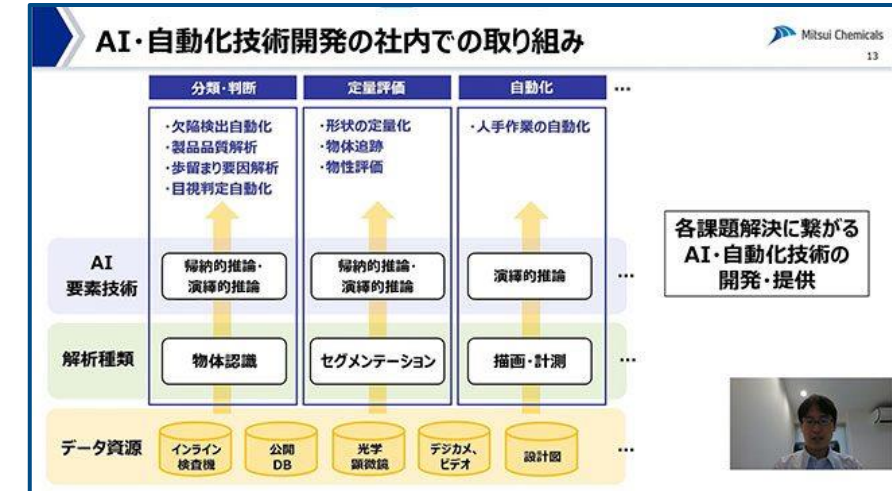
シート状製品の外観検査の自動化における開発モデルのユーザー利便性およびメンテナンス性

ソリューション

TensorFlow-Keras向けインポーターを使用してモデルをMATLABに取り込み、GUIアプリケーションとして配布して運用

結果

- 外観検査時間の8割削減
- 他のフレームワークで学習したモデルの有効活用
- 誰もが利用できるGUIアプリケーションとして展開



Python(TensorFlow-Keras)を使用したモデル開発と、MATLABを使用したモデルの効率的な現場実装。

“MATLABを活用することで、現場実装上の課題が解決され、工数を掛けることなく、確度の高い開発に繋がれるようになりました。”

- 前川 真太郎氏, 三井化学株式会社

Key Takeaways

- MATLABのAI開発ワークフロー
 - データアクセスからデプロイまで
- Automated Visual Inspection ライブラリ
 - 異常データが大量に手に入らない場合に使用できる手法も
- 様々なデプロイ方法
 - コード生成 -> 実機実装については次のパートで