



医用画像のための画像処理および AI最新機能紹介

- AIで切り開く医用画像処理 2023 -

Misa Taguchi

Senior Application Engineer
MathWorks Japan



本日起り上げる例題

深層学習を使用した超音波からの乳房腫瘍のセグメンテーション

元の画像



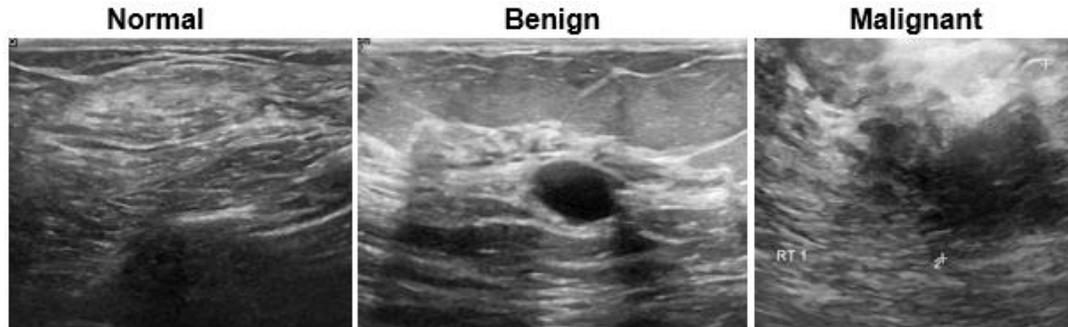
アジェンダ

- 医用画像解析のためのディープラーニングワークフロー
 - データアクセス・可視化
 - モデル作成
 - 新機能紹介
 - 事例紹介
- 深層学習を使用した超音波からの乳房腫瘍のセグメンテーション

医用画像解析のためのAIワークフロー

画像のディープラーニングでできること

分類

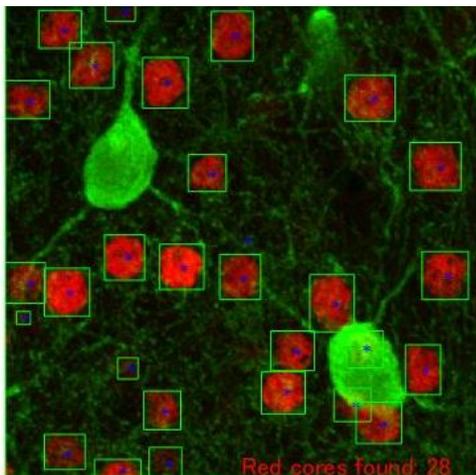


Al-Dhabyani, Walid, et al. "Dataset of breast ultrasound images." *Data in brief* 28 (2020): 104863.

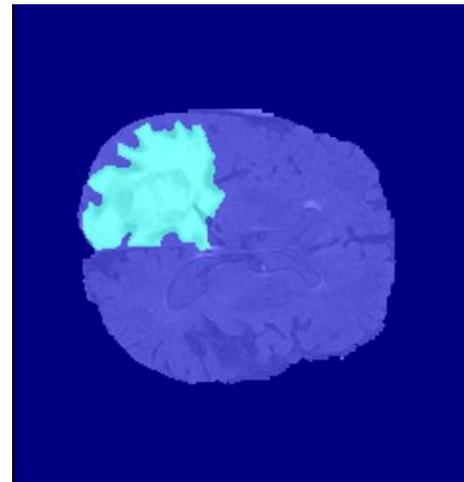
回帰



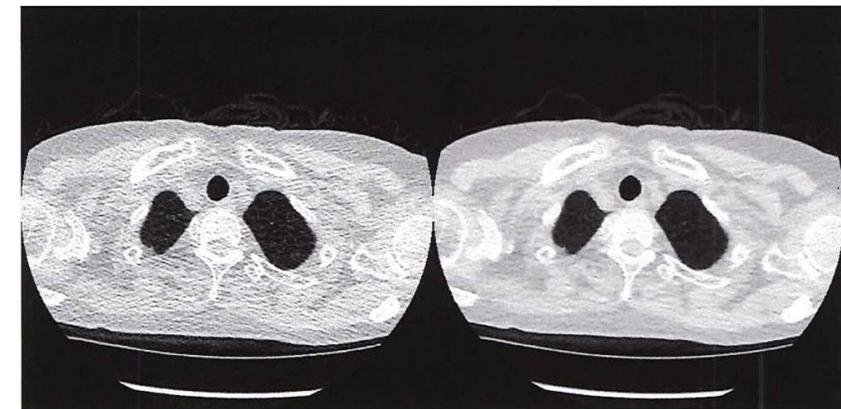
検出



セマンティック セグメンテーション



画像生成



医用画像のディープラーニングで想定されるワークフロー

データアクセス

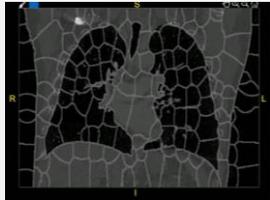
前処理

特徴抽出

モデルの学習

モデルの
チューニング

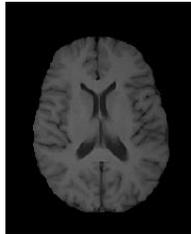
展開



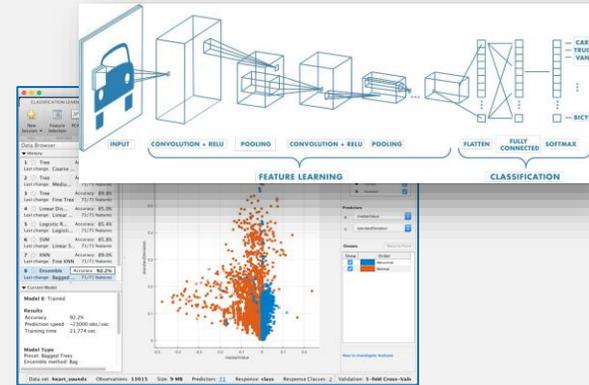
- 匿名化
- 必要な個所の切り出し
- アノテーション

Original MRI

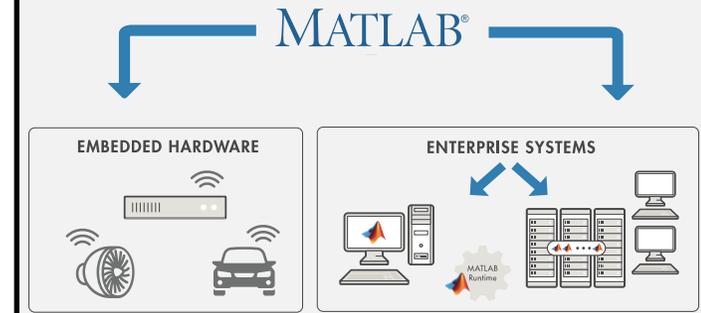
After Skull Stripping



- フィルタ
- リサイズ
- 正規化

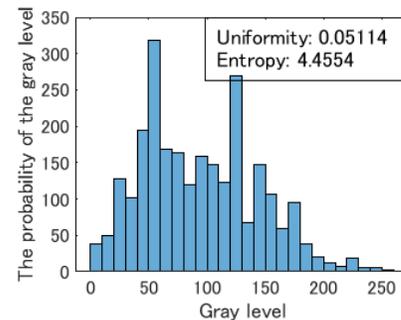
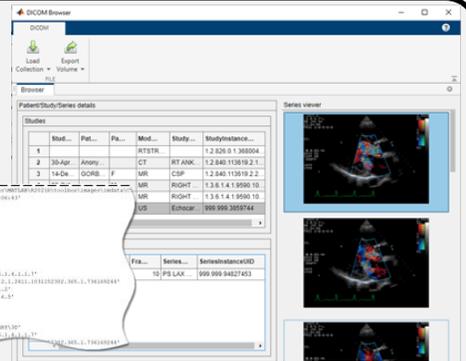


機械学習・ディープラーニング

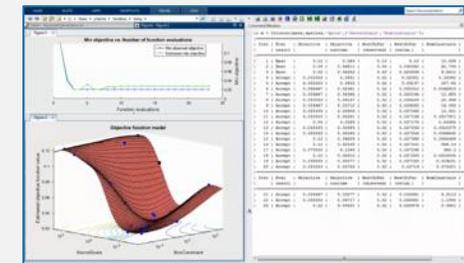


C/C++コード生成とシステムへの展開・ハードウェア実装

- データ読み込み
- 可視化
- データの確認



- テクスチャ
- フーリエ変換
- ディープニューラルネットワーク



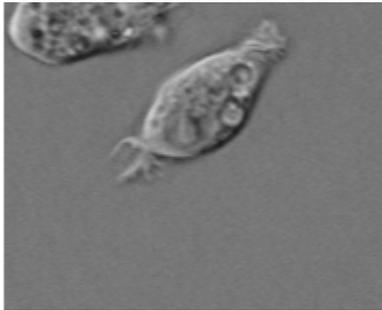
自動パラメータチューニング

医用画像処理のための

データアクセス

一般的なファイル形式の画像読み込み

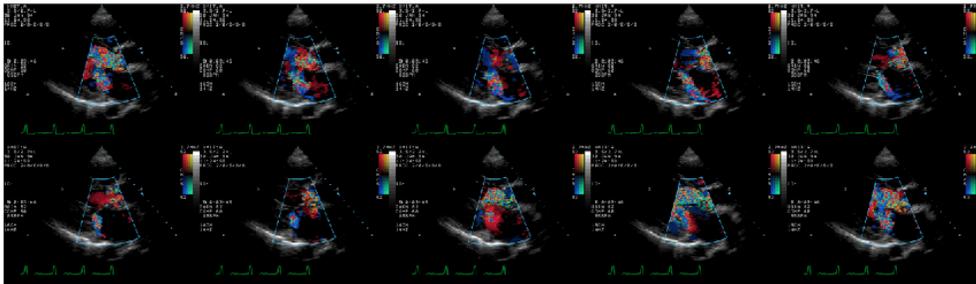
```
I = imread('xxx.jpg');
```



.tif
.bmp
など多数対応

DICOM画像読み込み

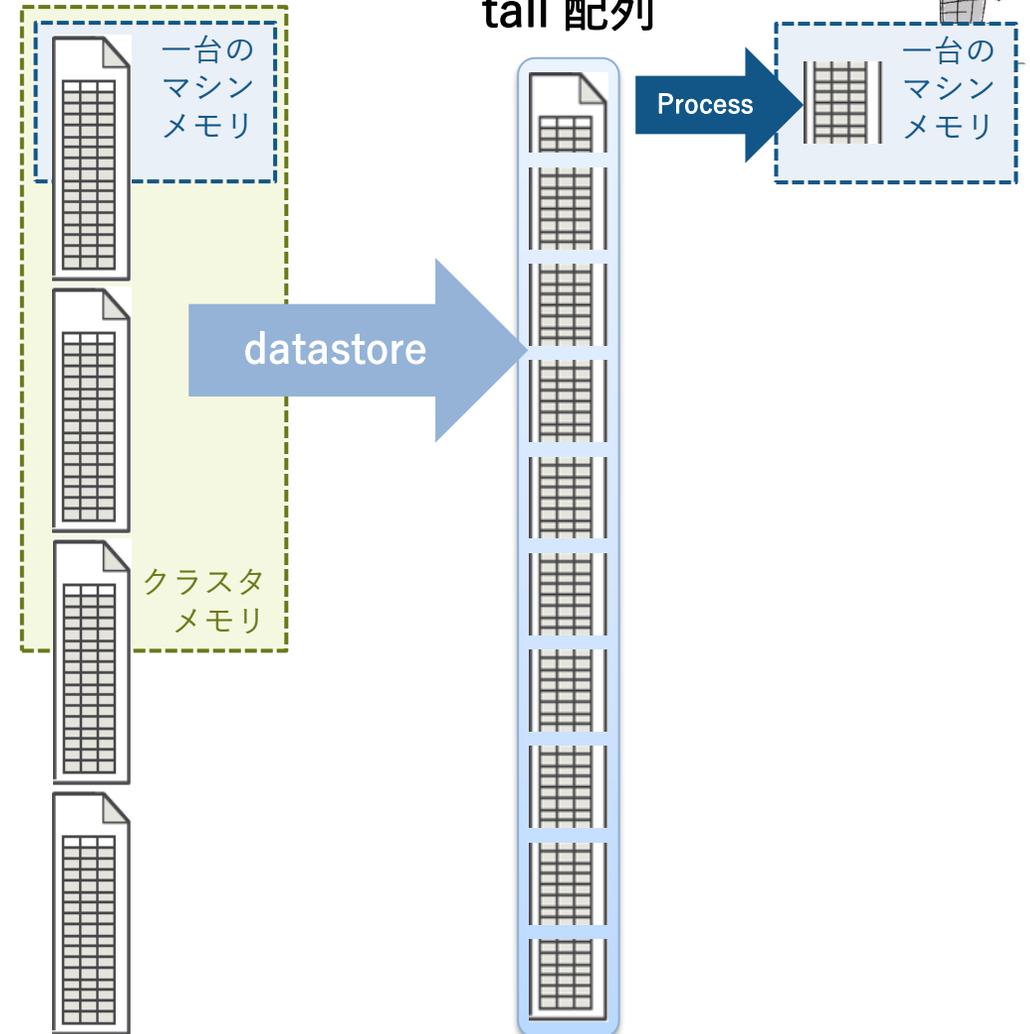
```
X = dicomread('xxx.dcm');
```



大規模な画像集合の読み込み

```
ds = imageDatastore();
```

tall 配列



医用画像処理のための

データアクセス

Mayo Analyze 7.5 ファイルの読み込み

```
info = analyze75info('xxx.hdr');  
X = analyze75read(info);
```

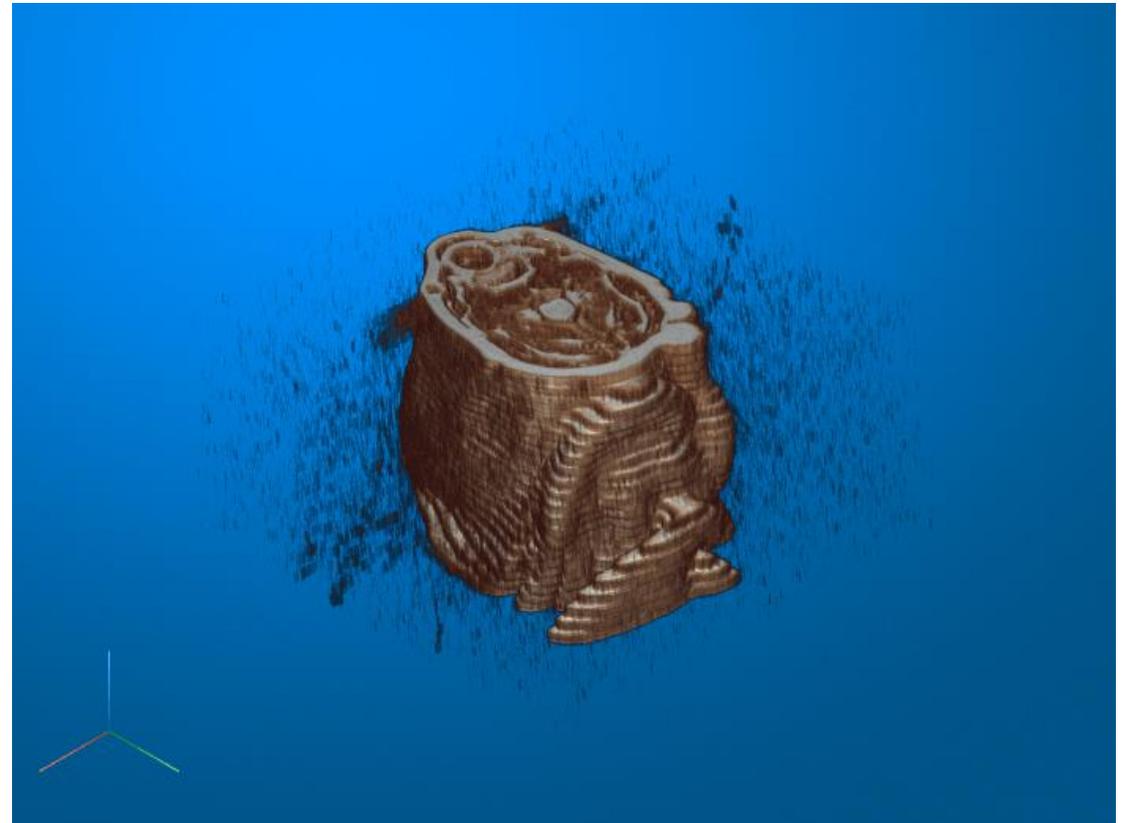
Interfile形式での画像の読み取り

```
I = interfileread(filename)
```



DICOMイメージセットから4Dボリューム作成

```
V = dicomreadVolume()
```



医用画像処理のための

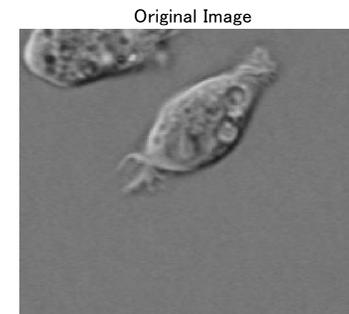
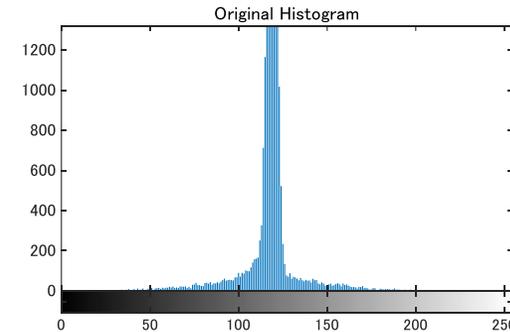
前処理

データの匿名化

`dicomanon(file_in, file_out)`

info2.PatientName	
フィールド	値
FamilyName	''
GivenName	''
MiddleName	''
NamePrefix	''
NameSuffix	''

コントラストの調整

`J = histeq(I)`

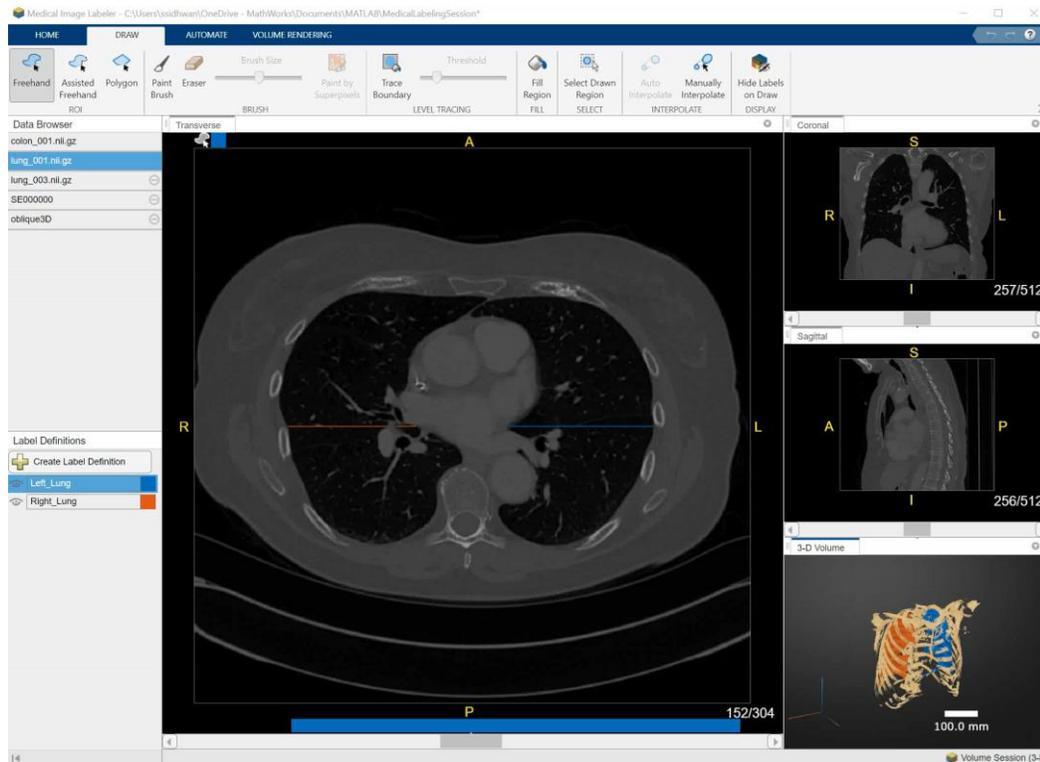
【参考】 MATLABによる画像の前処理機能

<https://jp.mathworks.com/help/images/image-enhancement-and-restoration.html>

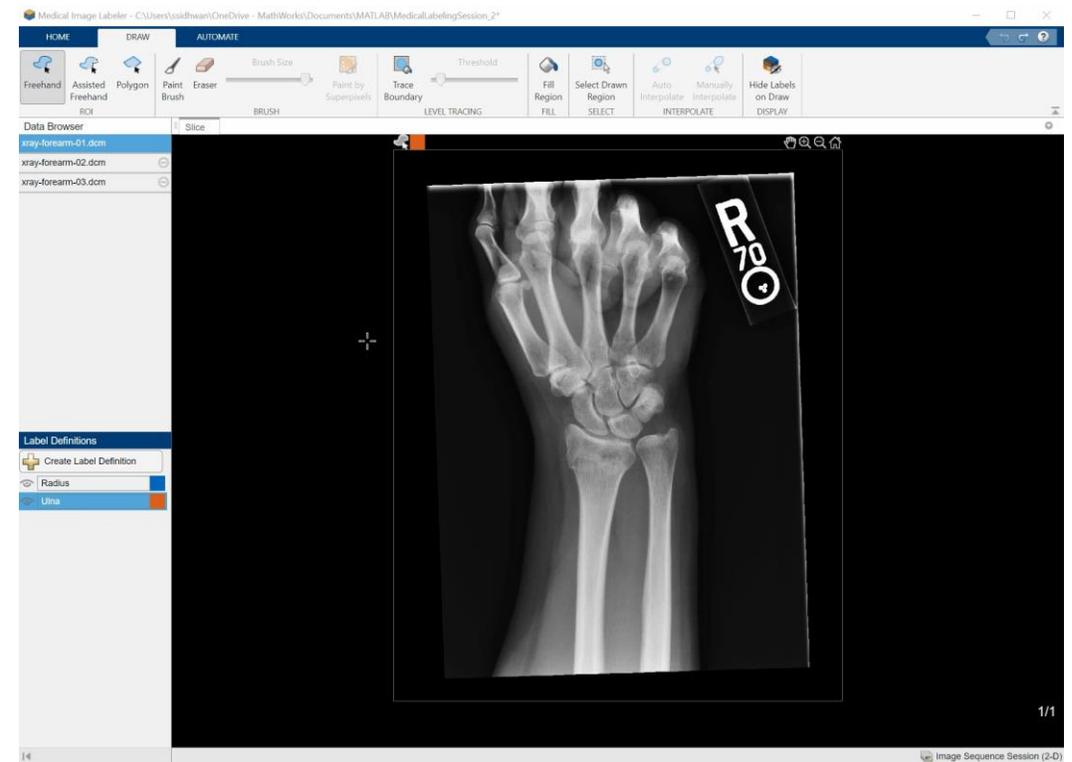
医用画像処理のための

前処理

医用画像ラベラーアプリによるアノテーション

>> `medicalImageLabeler`>> `imageLabeler`

Level Tracing

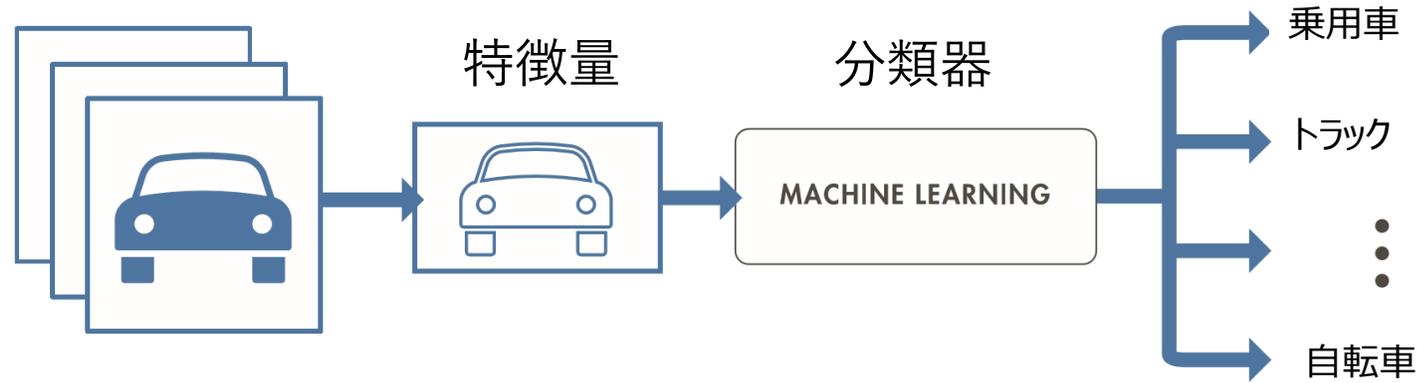


Paint by Superpixels

医用画像処理のための

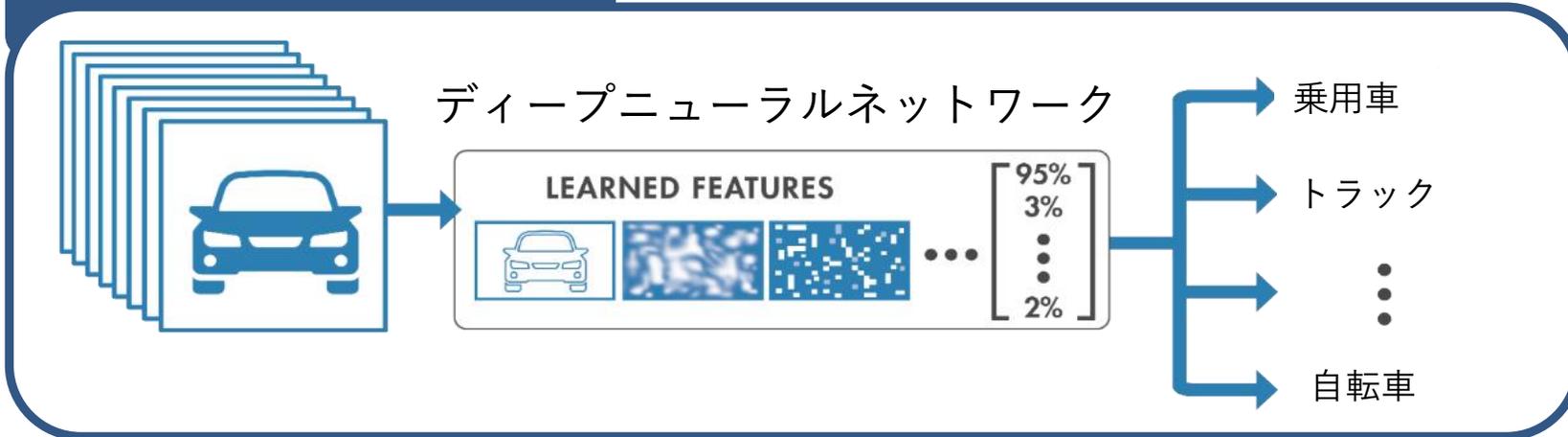


従来の機械学習



特徴量は人間が作成

Deep Learning



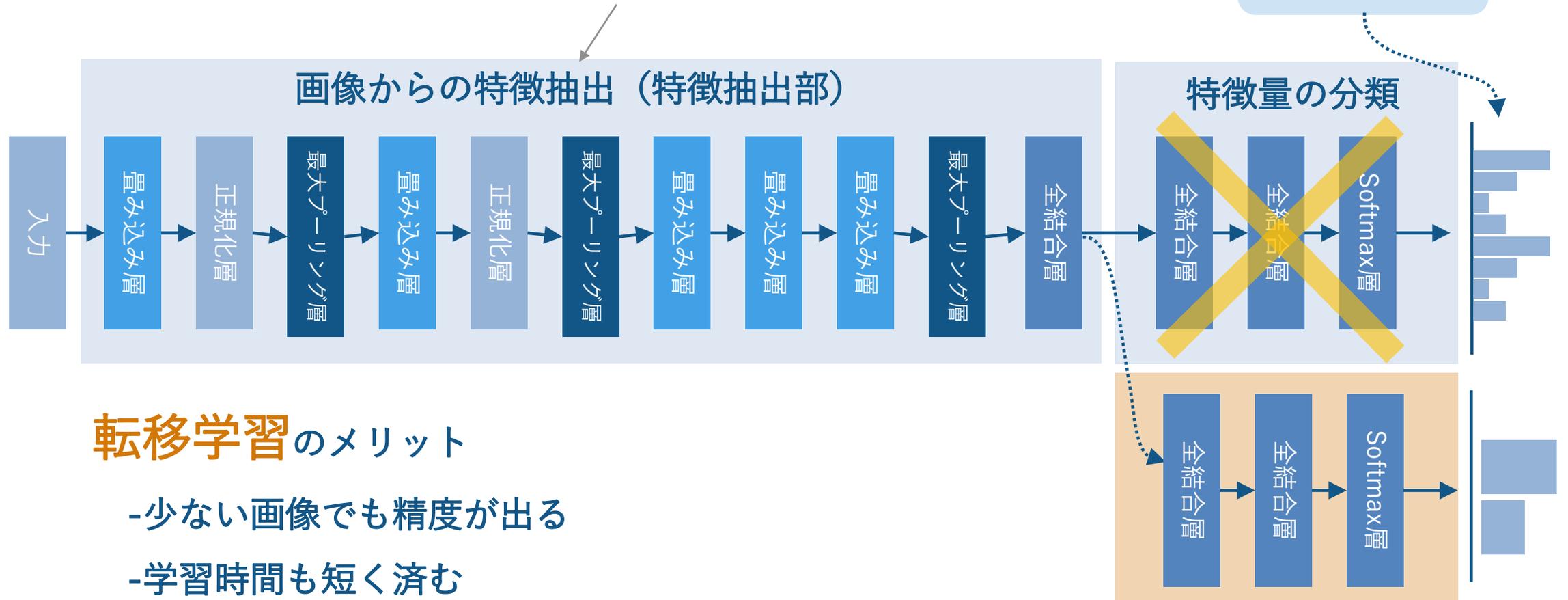
特徴量も学習

医用画像処理のための

特徴抽出

モデルの学習

事前に大量の画像で学習させた特徴抽出部を活用

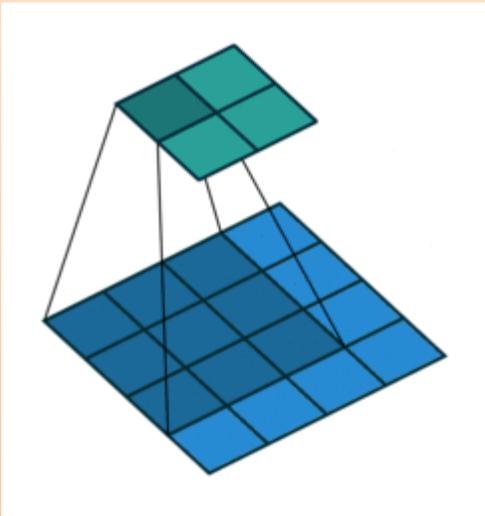


優れた既存ネットワークの特徴抽出器 + 独自分類器

Transformerモデルのサポート

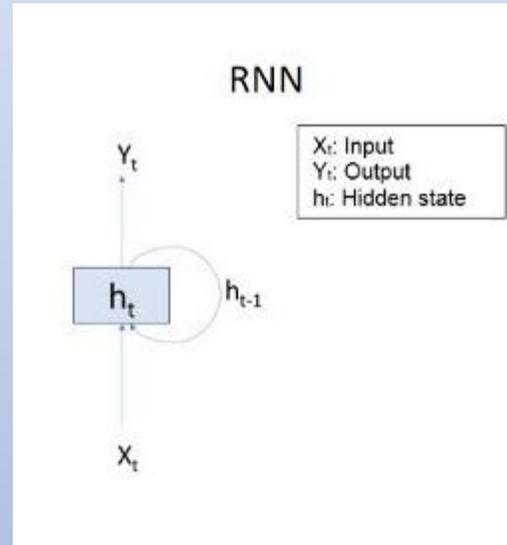
畳み込みニューラルネットワーク (CNN)

- **畳み込み**演算
- **画像**解析用途が主



リカレント型ニューラルネットワーク (RNN)

- **再帰型**演算
- **時系列データ**解析が主

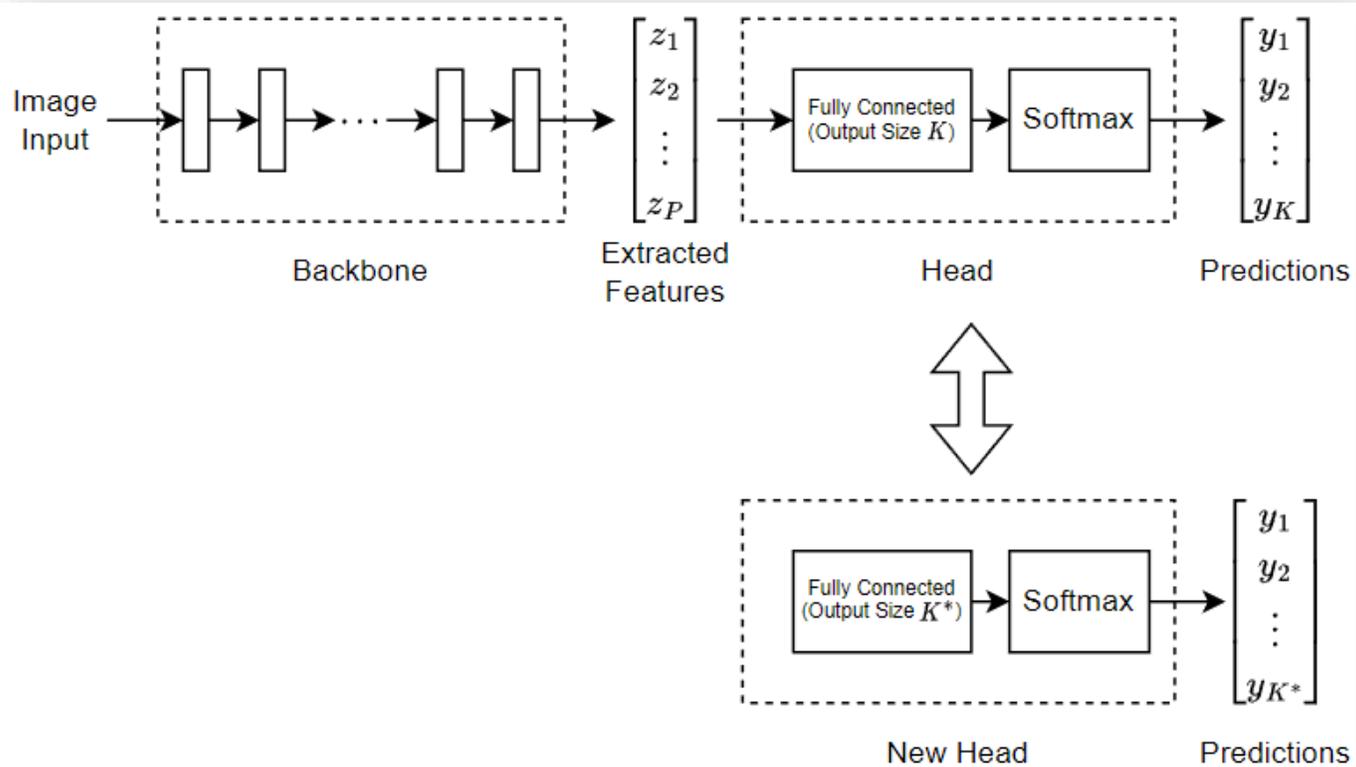


トランスフォーマー (Transformer)

- **マルチヘッドアテンション**層
- **様々なデータ**に適用可

- テキスト生成
- 文章要約
- 言語翻訳
- 画像分類
- 物体検出
- 音声認識
- 時系列予測 ...

Vision Transformer (ViT) モデルのサポート



Computer Vision Toolbox Model for Vision Transformer Network

作成者: MathWorks Computer Vision Toolbox Team **STAFF**

Implementation of several variants of the vision transformer (ViT) model.

概要 レビュー (2) ディスカッション (5)

The Vision Transformer (ViT) model is a pretrained transformer model for image classification. It is also used as a backbone for other computer vision tasks such as object detection. The support package consists of three variants of the ViT model:

- Base-16 model
- Small-16 model
- Tiny-16 model

Here, "base", "small" and "tiny" represent the model architecture and size, and 16 represents the patch size hyper-parameter. Each variant has been pretrained on ImageNet data set with input resolution of 384 and is stored as a .MAT file.

PatchCoreとは

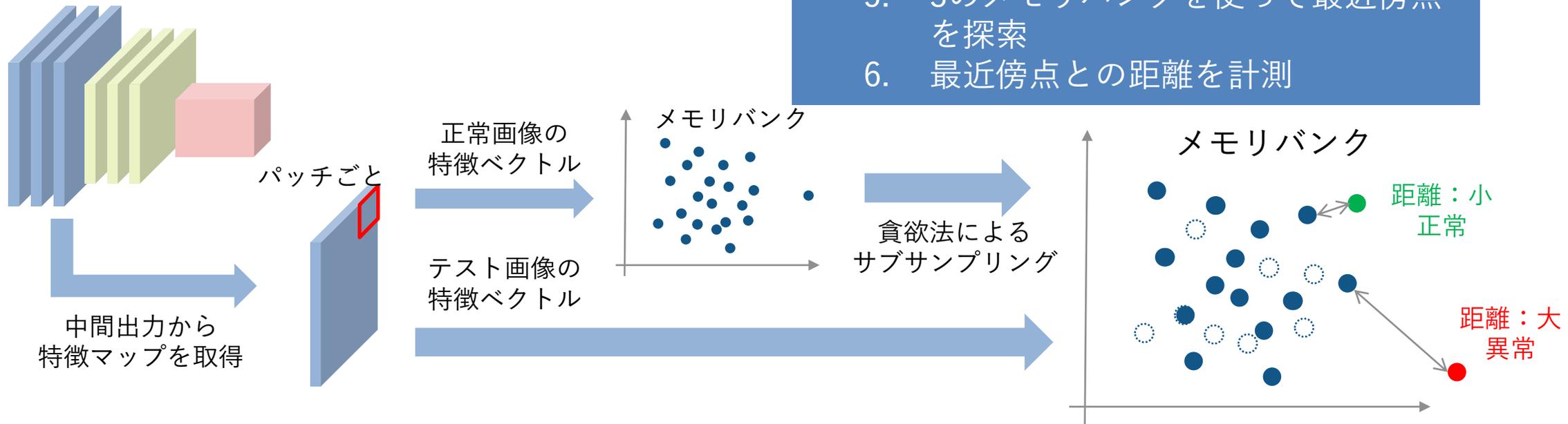
まず試してもらいたい！学習不要でお手軽！

- ニューラルネットの学習は不要
- 正常画像のみで学習完了
- MVTec ADデータセットでSOTA（最高精度）達成

手順概要

- 学習済みCNNで正常画像の特徴抽出
- メモリバンクとして分布を保持
- 貪欲法で軽量化
- 推論時、同様にCNNで特徴抽出
- 3のメモリバンクを使って最近傍点を探索
- 最近傍点との距離を計測

学習済みCNN



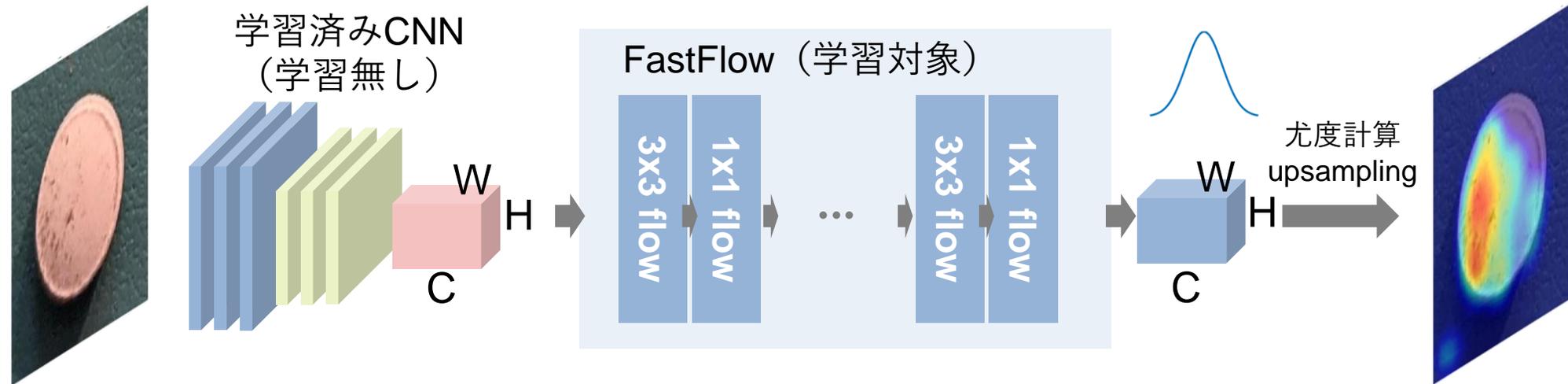
[Towards Total Recall in Industrial Anomaly Detection](#)

(Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, Peter Gehler, 2022)

FastFlowとは

PatchCoreでうまくいかないときに...

- 正常画像のみで学習完了
- 画像特徴を標準正規分布に変換する正規化フローを学習する。尤度を異常スコアとする
- MVTec ADデータセットでSOTA（最高精度）達成



[FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows](#)
(Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, Liwei Wu, 2021)

医用画像処理のための

特徴抽出

モデルの学習

ディープネットワークデザイナーによるネットワーク設計および学習

- 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

深層学習ネットワーク アーキテクチャの作成

次のプロ/ティを使って深層学習ネットワークの層を作成するためのスクリプト:

層の数: 144
結合数: 170

ワークスペース変数 Igraph 内の層を作成するためのスクリプトを実行します。
詳細についてはディープネットワークデザイナーから MATLAB コードを生成を参照してください。
MATLAB からの自動生成日: 06-Feb-2022 18:05

層グラフの作成

ネットワーク層を格納するために層グラフの変数を作成します。

```

1 Igraph = layerGraph();

2
3 tempLayers = [
4 imageInputLayer([224 224 3], "Name", "data")
5 convolution2dLayer([7 7], 64, "Name", "conv1-7x7_s2", "BiasLearnRateFactor", 2, "Padding", [3 3 3 3], "Stride", [2 2])
6 reluLayer("Name", "conv1-relu_7x7")
7 maxPooling2dLayer([3 3], "Name", "pool1-3x3_s2", "Padding", [0 1 0 1], "Stride", [2 2])
8 crossChannelNormalizationLayer(5, "Name", "pool1-norm1", "K", 1)
9 convolution2dLayer([1 1], 64, "Name", "conv2-3x3_reduce", "BiasLearnRateFactor", 2)
10 reluLayer("Name", "conv2-relu_3x3_reduce")
11 convolution2dLayer([3 3], 96, "Name", "conv2-3x3", "BiasLearnRateFactor", 2, "Padding", [1 1 1 1])
12 reluLayer("Name", "conv2-relu_3x3")
13 crossChannelNormalizationLayer(5, "Name", "conv2-norm2", "K", 1)
14 maxPooling2dLayer([3 3], "Name", "pool2-3x3_s2", "Padding", [0 1 0 1], "Stride", [2 2]);
15 Igraph = addLayers(Igraph, tempLayers);
16
17 tempLayers = [
18 convolution2dLayer([1 1], 16, "Name", "inception_3a-5x5_reduce", "BiasLearnRateFactor", 2)
19 reluLayer("Name", "inception_3a-relu_5x5_reduce")
20 convolution2dLayer([3 3], 32, "Name", "inception_3a-5x5", "BiasLearnRateFactor", 2, "Padding", [2 2 2 2])
21 reluLayer("Name", "inception_3a-relu_5x5")
22 Igraph = addLayers(Igraph, tempLayers);
23
24 Igraph = connectLayers(Igraph, "pool4-3x3_s2", "inception_3a-...");
25 Igraph = connectLayers(Igraph, "pool4-3x3_s2", "inception_3a-pool1");
26 Igraph = connectLayers(Igraph, "pool4-3x3_s2", "inception_3a-3x3_reduce");
27 Igraph = connectLayers(Igraph, "inception_3a-relu_5x5", "inception_3a-output/in3");
28 Igraph = connectLayers(Igraph, "inception_3a-relu_pool_proj", "inception_3a-output/in4");
29 Igraph = connectLayers(Igraph, "inception_3a-relu_3x3", "inception_3a-output/in2");
30 Igraph = connectLayers(Igraph, "inception_3a-relu_1x1", "inception_3a-output/in1");
31 Igraph = connectLayers(Igraph, "inception_3a-output", "inception_3b-pool");
32 Igraph = connectLayers(Igraph, "inception_3a-output", "inception_3b-1x1");
33 Igraph = connectLayers(Igraph, "inception_3a-output", "inception_3b-5x5_reduce");
34 Igraph = connectLayers(Igraph, "inception_3a-output", "inception_3b-3x3_reduce");
35 Igraph = connectLayers(Igraph, "inception_3b-relu_pool_proj", "inception_3b-output/in4");
36 Igraph = connectLayers(Igraph, "inception_3b-relu_1x1", "inception_3b-output/in1");
37 Igraph = connectLayers(Igraph, "inception_3b-relu_5x5", "inception_3b-output/in3");
38 Igraph = connectLayers(Igraph, "inception_3b-relu_3x3", "inception_3b-output/in2");
39
40 Igraph = plot(Igraph);

```

← 351行目

医用画像処理のための

特徴抽出

モデルの学習

trainNetwork: 標準的な学習方法

```
% 学習オプション設定
options = trainingOptions('adam', ...
    'MaxEpochs', 150, ...
    'InitialLearnRate', 0.005, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod', 125, ...
    'LearnRateDropFactor', 0.2, ...
    'Plots', 'training-progress');
```

```
% 学習
```

```
net = trainNetwork(XTrain, YTrain, layers, options);
```

データ ネットワーク オプション

コマンドウィンドウ

```
fx >> doc trainingOptions
```

ミニバッチ オプション

すべて折りたたむ

MaxEpochs — エポックの最大回数
30 (既定値) | 正の整数

学習に使用するエポックの最大回数。'MaxEpochs' と正の整数で構成されるコンマ区切りのペアとして指定します。

反復とは、勾配降下法アルゴリズムでミニバッチを使用して損失関数の最小化を目指して実行される 1 ステップのことです。1 エポックは、データセット全体を一巡することです。

例: 'MaxEpochs', 20

MATLAB ドキュメンテーション

医用画像処理のための

モデルの
チューニング

様々な初期条件下でネットワークを学習させる実験を行い、結果を比較:

- ハイパーパラメータ値の範囲をスイープ
- 異なるデータセットを使用した場合の結果を比較
- 様々なディープネットワークアーキテクチャをテスト

実験マネージャー アプリ

初期リリース

R2020a

学習の並列化

R2020b

ベイズ最適化

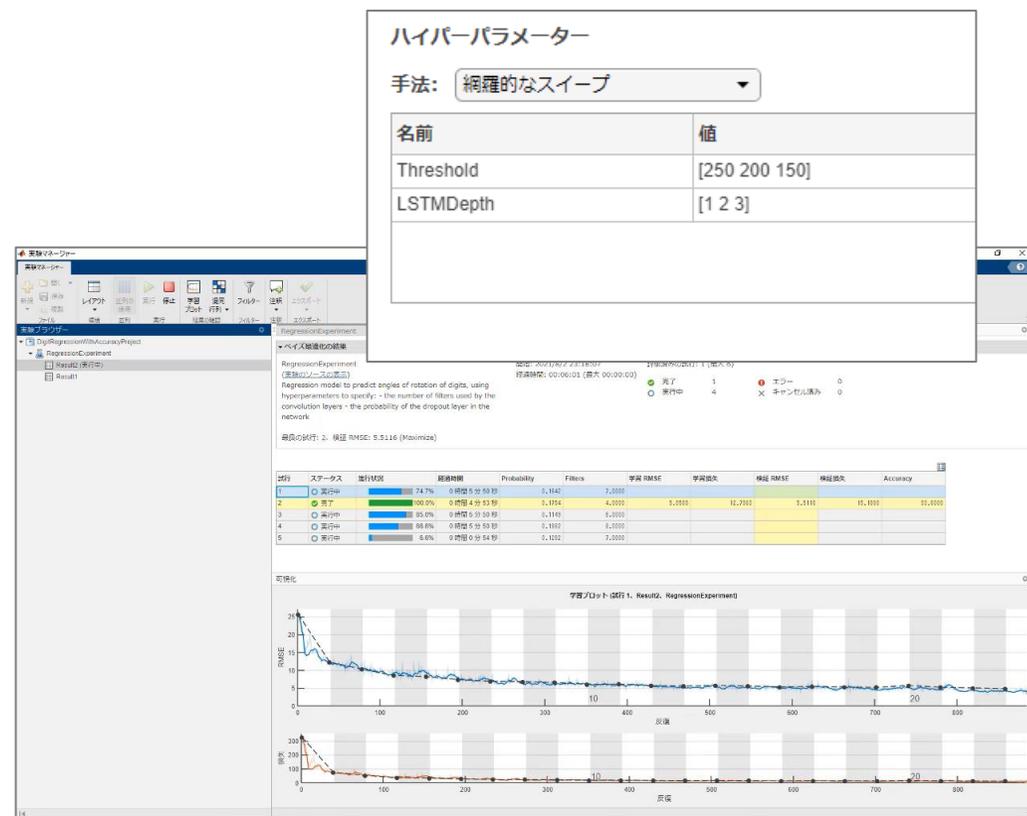
R2021b

カスタム学習の実験

R2021a

MATLAB online サポート

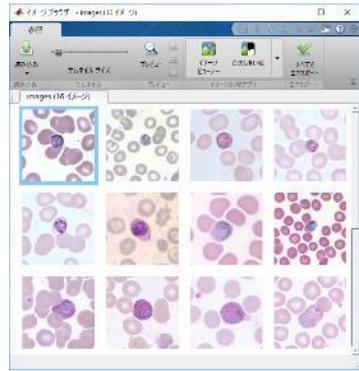
R2021b



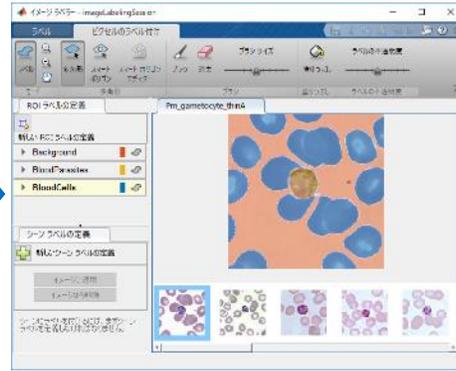
実験マネージャーアプリ
複数のディープラーニング実験を管理し、
結果やコードを分析・比較

事例：特定の領域の抽出(セグメンテーション)

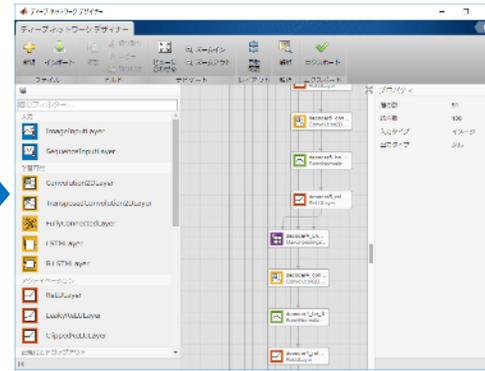
- 医用画像(血液検査画像)から特定のオブジェクトの抽出



画像読み込み
imageDatastore



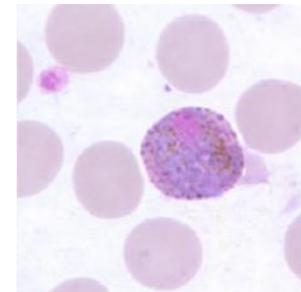
ラベリング
imageLabeler



ネットワークの定義
segnetLayers,
deepNetworkDesigner



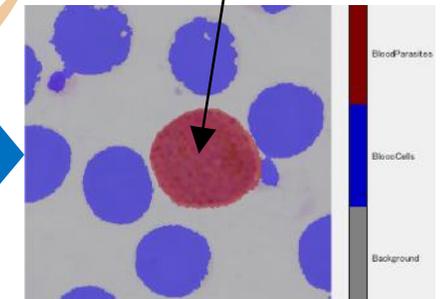
ネットワーク学習
trainingOptions,
trainNetwork



画像読み込み
imread



CNNによる
セグメンテーション
semanticseg



病原虫が寄生した
領域のみを抽出

- サンプルプログラム

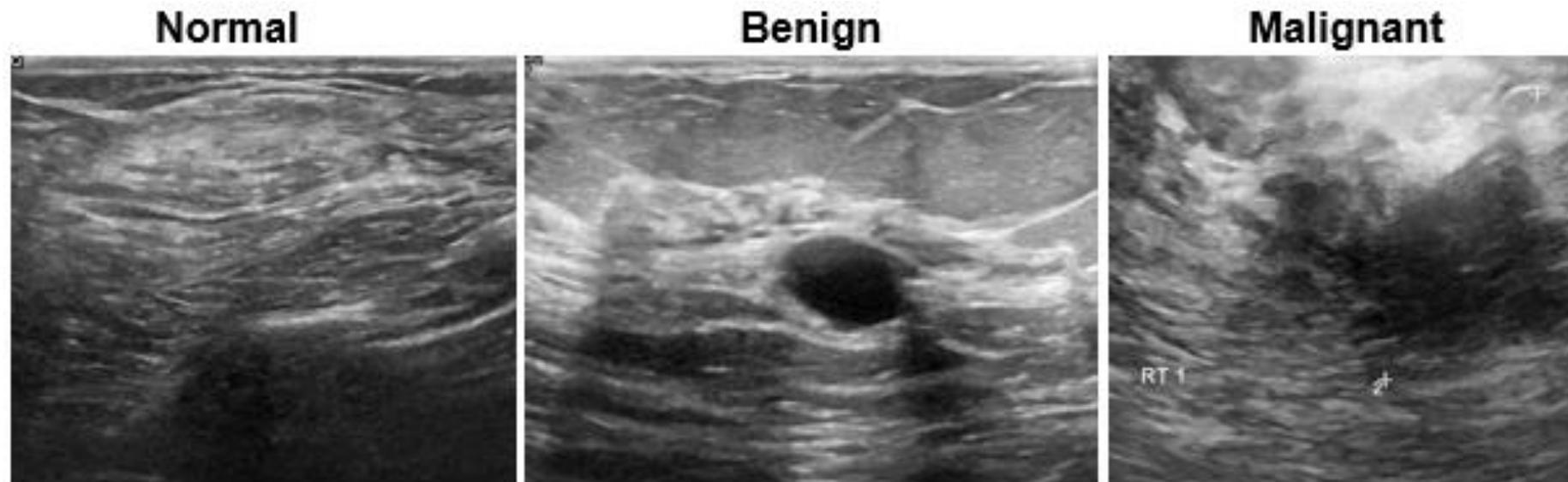
– [File Exchange : Medical Image Segmentation Using SegNet](#)

超音波からの
乳房腫瘍のセグメンテーション

乳房超音波画像 (BUSI) データセットを用いた解析デモ

データアクセス

- 正常なイメージ (Normal) : 133枚
- 良性腫瘍 (Benign) : 487枚
- 悪性腫瘍 (Malignant) : 210枚

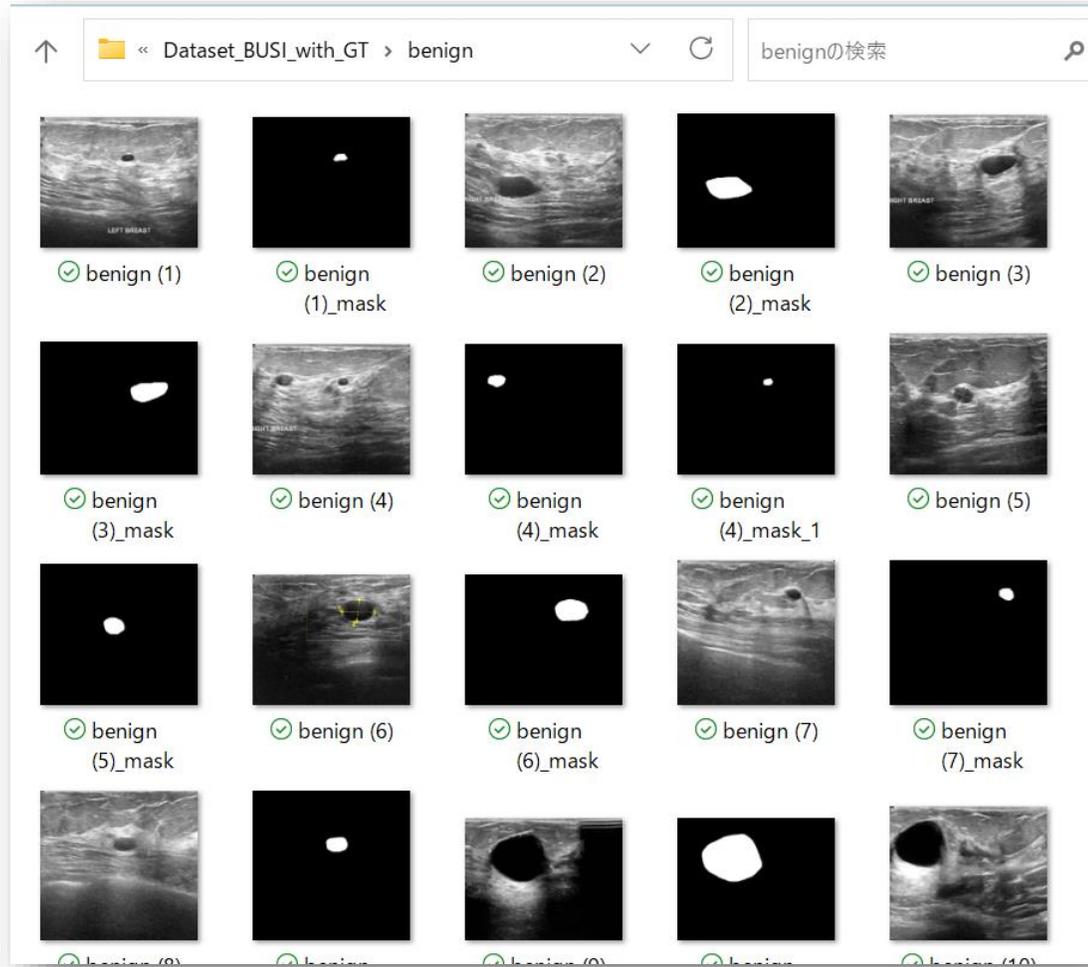


学習用画像の準備

データストアを用いた大規模な画像集合の読み込み

データアクセス

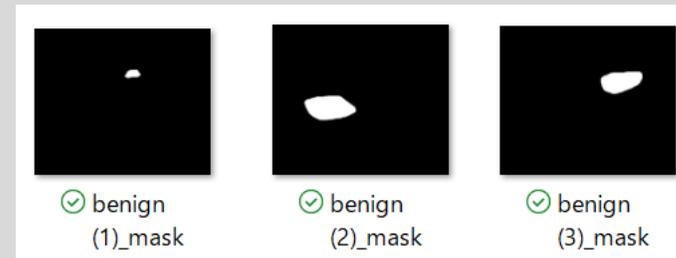
前処理



imageDatastore



pixelLabelDatastore



combine

学習用画像の準備

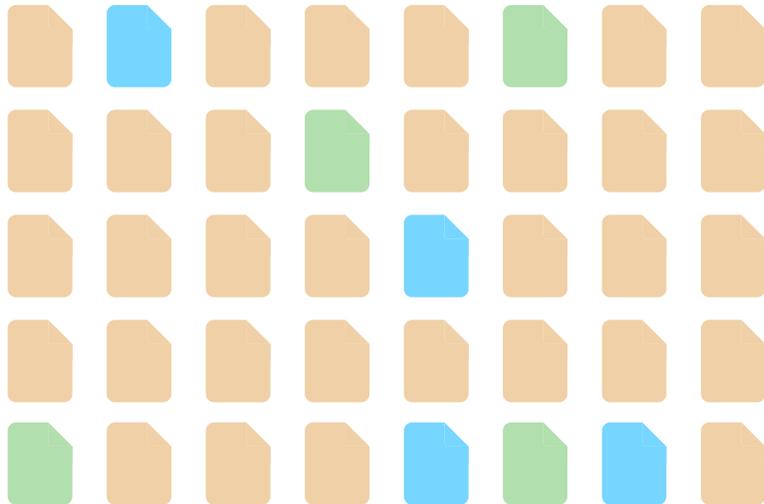
汎化性能向上および精度向上のための取り組み

データアクセス

前処理

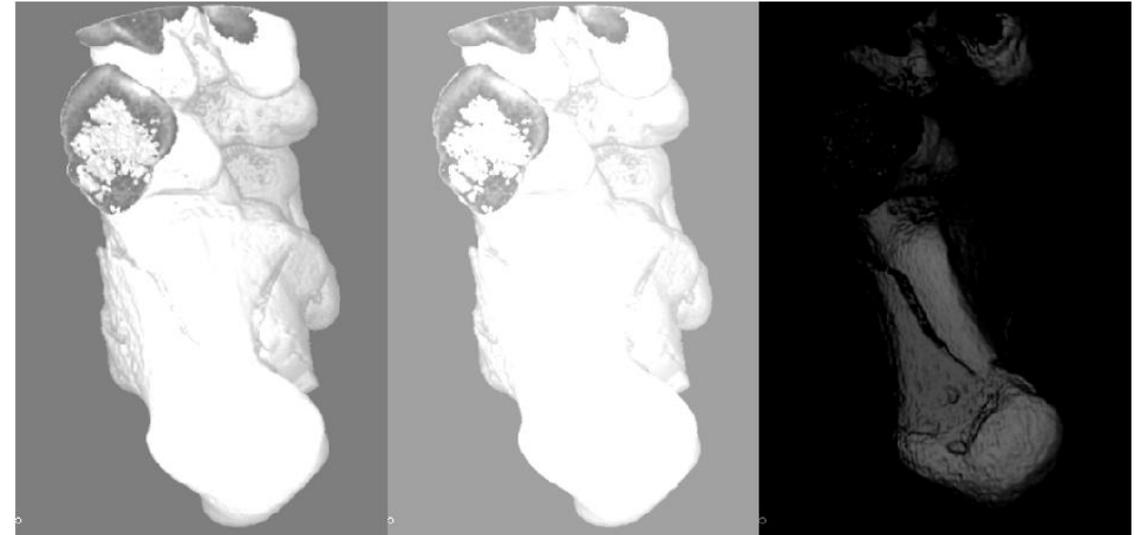
データストアの分割

- 学習用：80%
- 検証用：10%
- テスト用：10%



輝度・コントラスト・ガンマ補正をランダムに変化させグレースケール画像を**水増し**

`jitterIntensity`



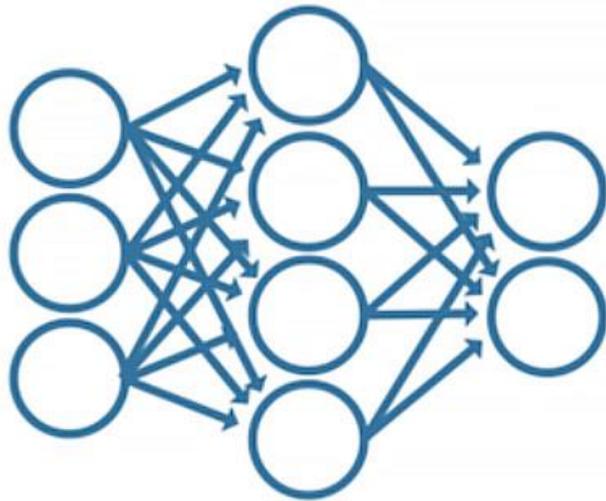
ディープニューラルネットワークの準備

学習済みネットワークを活用した高精度モデルの作成

特徴抽出

モデルの学習

ResNet-50ベースのDeepLab v3+



imageInputLayer

入力画像サイズの変更

Before: 256 x 256 x 3

After: 256 x 256

convolution2dLayer

新しい入力層のサイズに一致するように、最初の2次元畳み込み層を新しい2次元畳み込み層に置き換え

tverskyPixelClassificationLayer

小さな腫瘍領域をより適切にセグメント化し、より大きな背景領域の影響を軽減するため、カスタム Tversky ピクセル分類レイヤーを使用

BUSIデータセットを用いた深層学習

ローコードで実現するGPU設定および学習

特徴抽出

モデルの学習

```
% 学習オプション設定
```

```
options = trainingOptions("adam", ...  
    ExecutionEnvironment="gpu", ...  
    InitialLearnRate=1e-3, ...  
    ValidationData=tdsVal, ...  
    MaxEpochs=300, ...  
    MiniBatchSize=16, ...  
    VerboseFrequency=20, ...  
    Plots="training-progress");
```

```
% 学習
```

```
[trainedNet, info] = trainNetwork(tdsTrain, lgraph, options);
```

Key Takeaways

- 医用画像解析のためのディープラーニングワークフロー
- 深層学習を使用した超音波からの乳房腫瘍のセグメンテーション



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See 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.