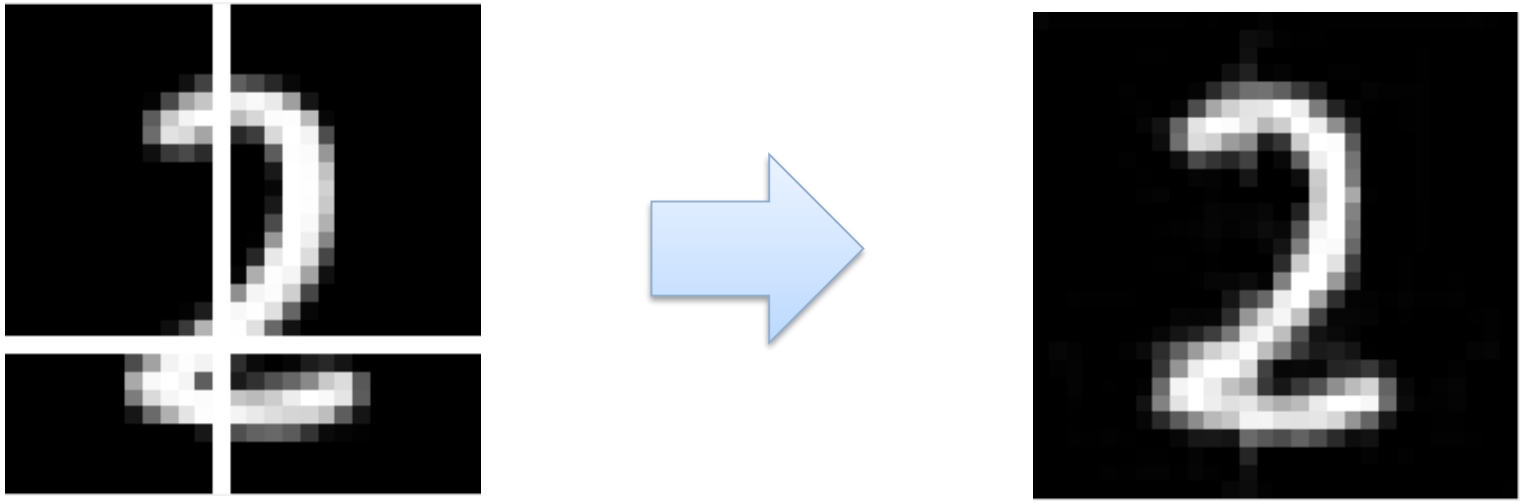
ディープラーニング：
画像補正の基礎から超解像変換の応用まで

MathWorks Japan
アプリケーションエンジニアリング部
木川田 亘

縦・横線ノイズに対応したノイズ除去ネットワークの構築

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox



ディープラーニングを活用して高機能な画像補正を実現

アジェンダ

- ディープラーニングによる画像補正の基礎
- ノイズ除去ネットワークの学習ワークフロー
- ディープラーニングによる画像補正の応用例
- まとめ

アジェンダ

- ディープラーニングによる画像補正の基礎
- ノイズ除去ネットワークの学習ワークフロー
- ディープラーニングによる画像補正の応用例
- まとめ

ディープラーニングを使用したイメージのノイズ除去

- ディープラーニングを使ってノイズを除去
  - 従来の画一的なフィルタではノイズ除去と同時に本来のエッジ成分がぼけるなどの弊害
  - ディープラーニングを活用することで画像の局所的な特徴を保存したノイズ除去を実現

ノイズあり画像

学習済み
ノイズ除去ネットワーク

ノイズ除去画像

ノイズ除去畳み込みニューラルネットワーク（DnCNN）レイヤー構造

Image Processing Toolbox
Neural Network Toolbox

- DnCNN[1]:ノイズ除去たたみ込みニューラル ネットワーク
- 20層の畳み込み層で構成
- １チャネル入力

```
>> layers = dnCNNLayers
layers =
  次の層をもつ 1x59 の Layer 配列:

     1   'InputLayer'              イメージの入力    50x50x1 イメージ
     2   'Conv1'                   たたみ込み      ストライド [1  1] およびパディング [1  1  1  1] の 64 3x3x1 たたみ込み
     3   'ReLU1'                   ReLU        ReLU
     4   'Conv2'                   たたみ込み      ストライド [1  1] およびパディング [1  1  1  1] の 64 3x3x64 たたみ込み
     5   'BNorm2'                  バッチ正規化     64 チャネルでのバッチ正規化
     6   'ReLU2'                   ReLU        ReLU
     :      :                         :           :
    55   'Conv19'                  たたみ込み      ストライド [1  1] およびパディング [1  1  1  1] の 64 3x3x64 たたみ込み
    56   'BNorm19'                 バッチ正規化     64 チャネルでのバッチ正規化
    57   'ReLU19'                  ReLU        ReLU
    58   'Conv20'                  たたみ込み      ストライド [1  1] およびパディング [1  1  1  1] の 1 3x3x64 たたみ込み
    59   'FinalRegressionLayer'    回帰出力       mean-squared-error
```

[1] Zhang, K., W. Zuo, Y. Chen, D. Meng, and L. Zhang. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising." IEEE Transactions on Image Processing. Vol. 26, Number 7, Feb. 2017, pp. 3142-3155.

DnCNNによるガウシアンノイズの除去

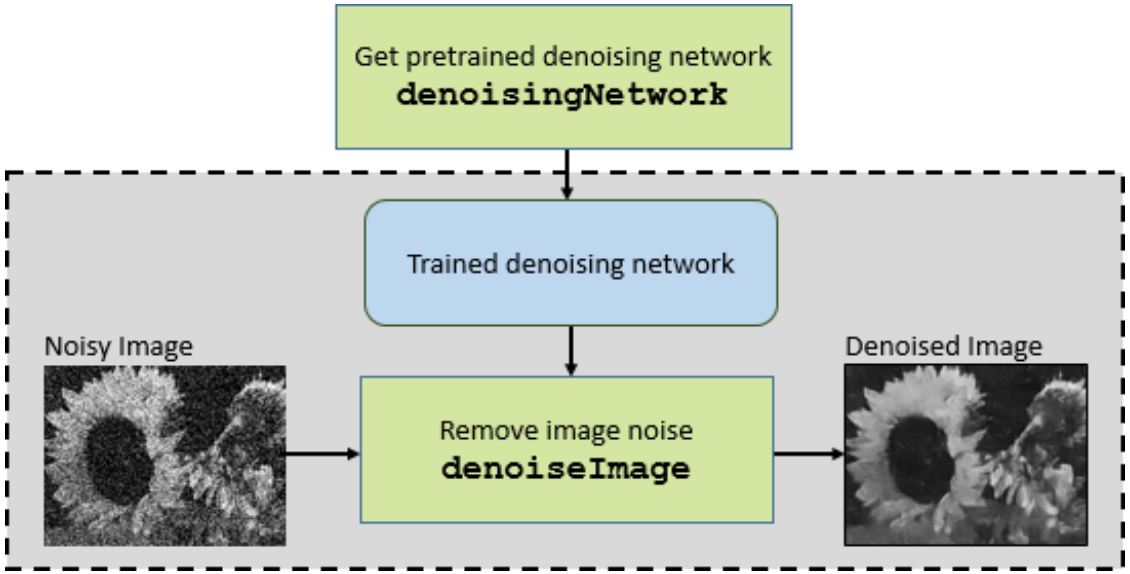
Image Processing Toolbox
Neural Network Toolbox

- ガウシアンノイズ除去のための学習済みネットワークを用意
  - ステップ１　ノイズ除去畳み込みニューラルネットワークの読み込み

```
net = denoisingNetwork('DnCNN');
```

  - ステップ２　ノイズ付加画像と使用するネットワークを指定して実行

```
I = denoiseImage(noisyI, net);
```



他のフィルタとの比較

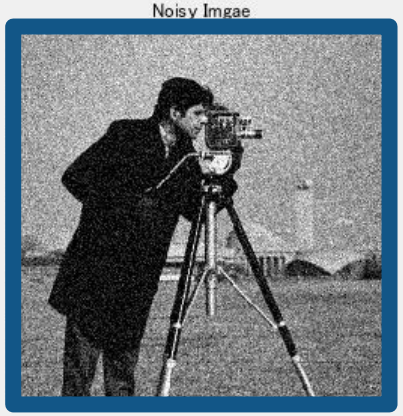
Image Processing Toolbox
Neural Network Toolbox

原画像

Original Image

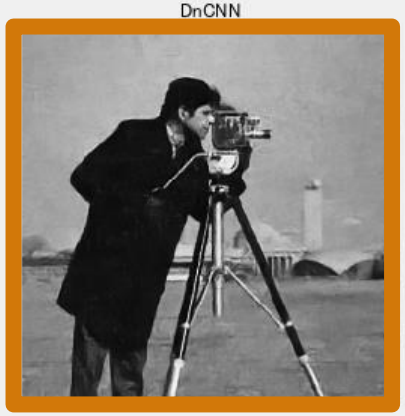
ノイズあり

Noisy Imgae



DnCNN使用時

DnCNN



ガウシアンフィルタ

Gaussianfilter
(sigma=2)

バイラテラルフィルタ

Bilateral Filtered

メディアンフィルタ

Median Filtered

ウィーナーフィルタ

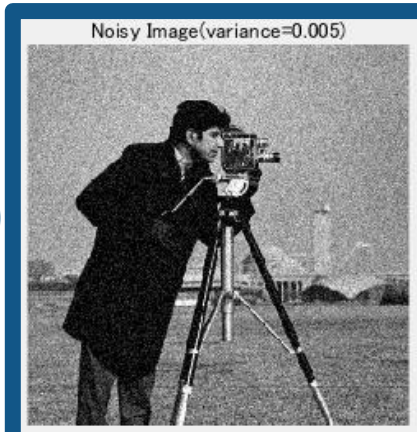
Adaptive Filtered

ガイド付きフィルタ

Guided Filtered

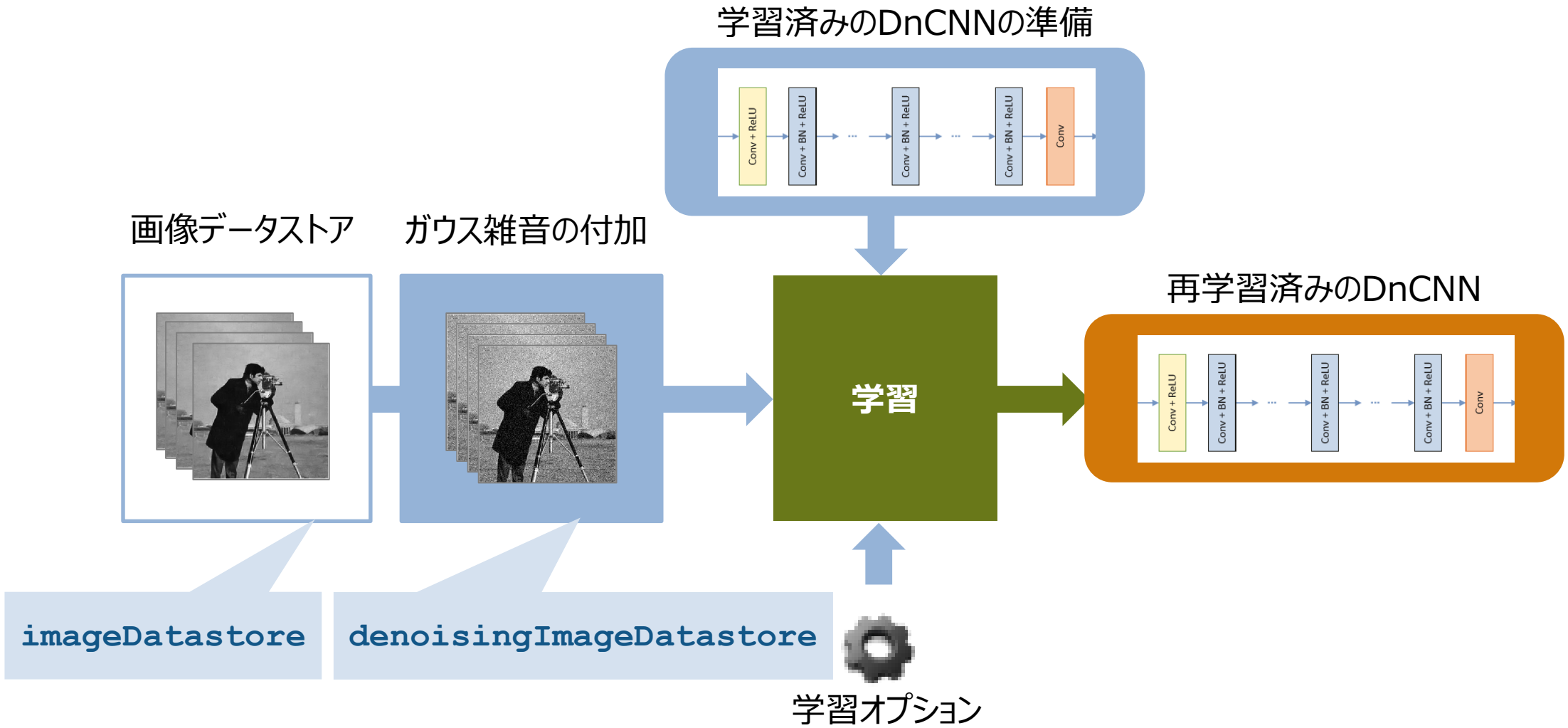
ガウシアンノイズの分散値を変えた場合の比較

Image Processing Toolbox
Neural Network Toolbox

	$\sigma^2=0.005$	$\sigma^2=0.01$	$\sigma^2=0.05$	$\sigma^2=0.1$
ノイズあり				
DnCNN 使用時				

再学習を行うことも可能

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox



処理結果の評価

- イメージのピーク S/N 比（PSNR）

```
PSNR = psnr(I, Iref);
```

- イメージの構造類似度（SSIM）

```
SSIM = ssim(I, Iref);
```

- Naturalness Image Quality Evaluator (NIQE) 非参照画質スコア

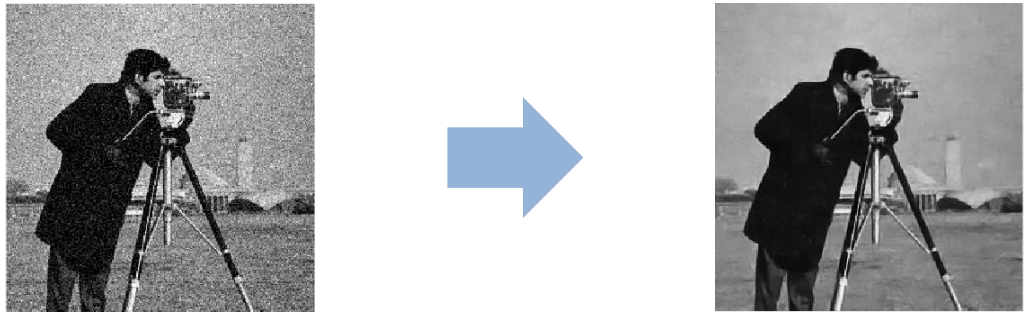
```
SSIM = niqe(I);
```

アジェンダ

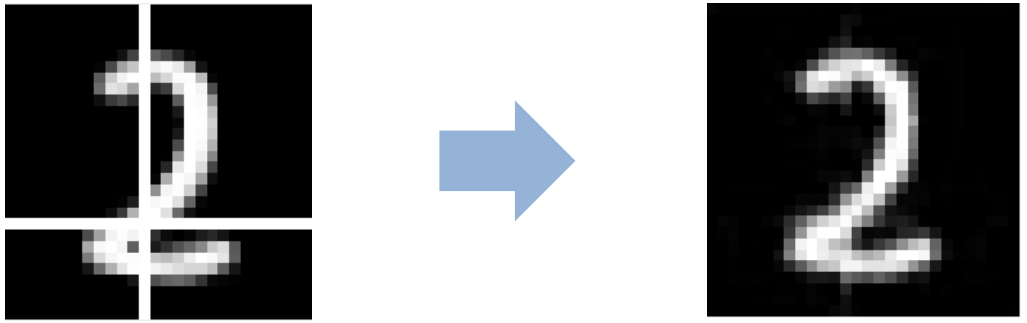
- ディープラーニングによる画像補正の基礎
- ノイズ除去ネットワークの学習ワークフロー
- ディープラーニングによる画像補正の応用例
- まとめ

独自のノイズに対して学習をさせたい

- DnCNNはガウシアンノイズのみ

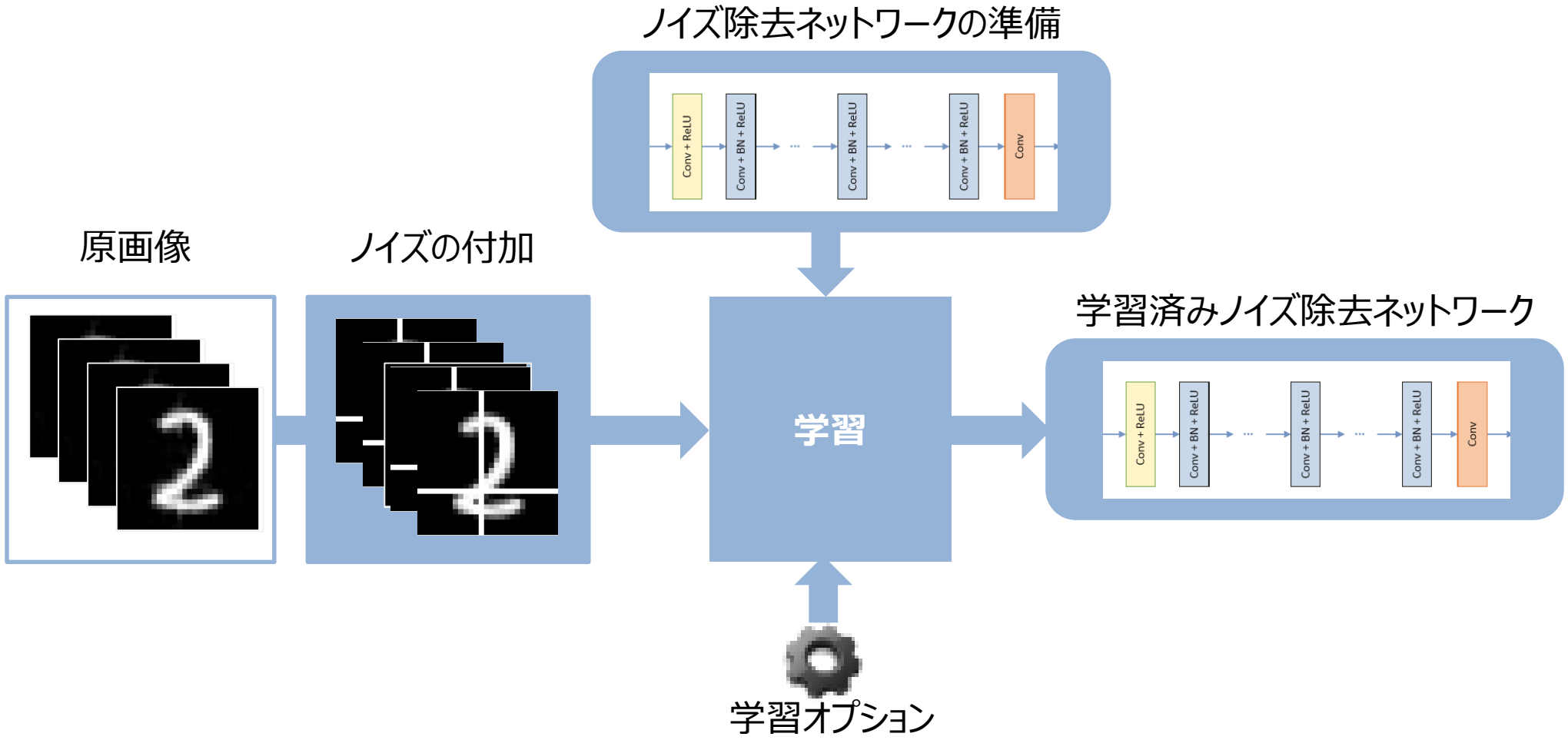


- 任意の特性を持ったノイズを除去するには？



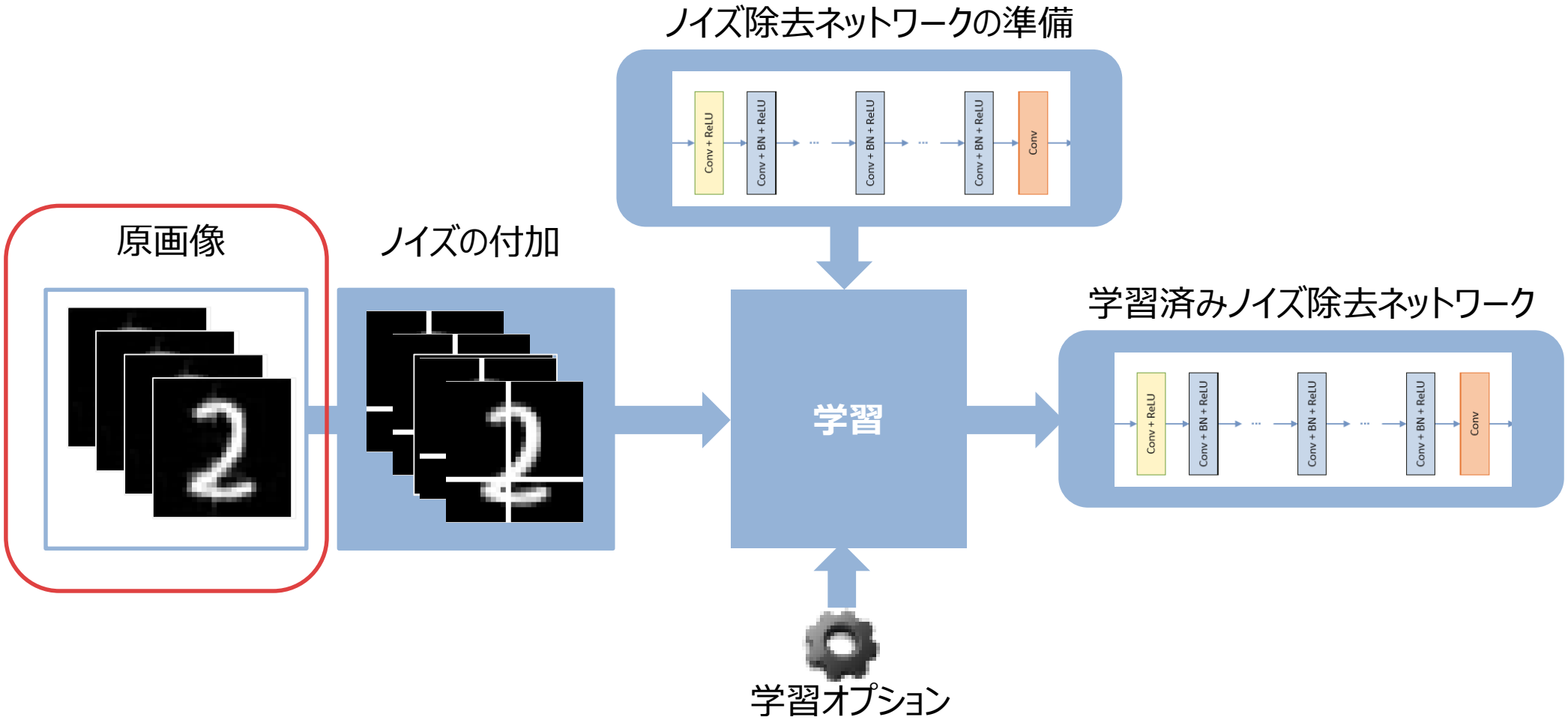
ノイズ除去ネットワークの学習ワークフロー

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox



ノイズ除去ネットワークの学習ワークフロー

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox



大量の画像セットへのアクセス

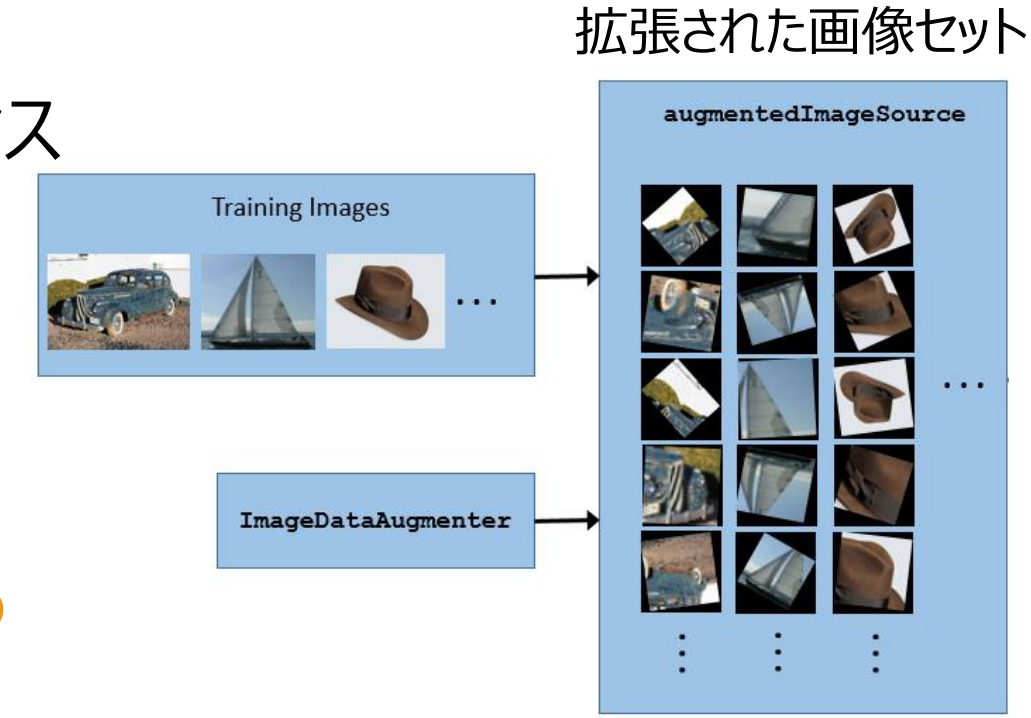
- 大量の画像セットにメモリ効率の良いアクセス
  - `imageDatastore`
- データ拡張
  - **imageDataAugmenter**による画像データの拡張
    - スケール/せん断/回転

    等の制限をかけながらデータを拡張

R2017b

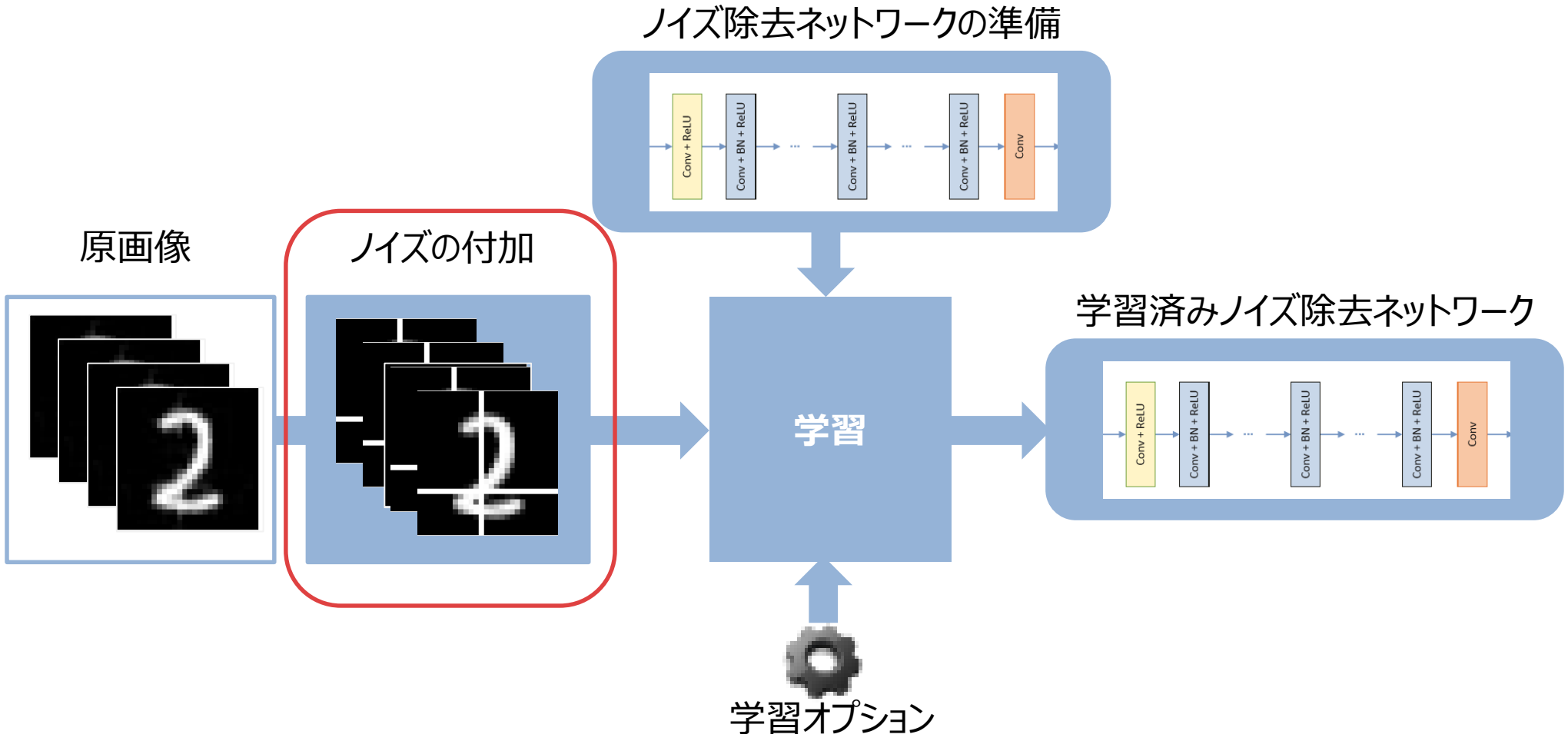
- ユーザー指定のデータ拡張/前処理
  - カスタムのミニバッチデータストア対応

R2018a



ノイズ除去ネットワークの学習ワークフロー

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox



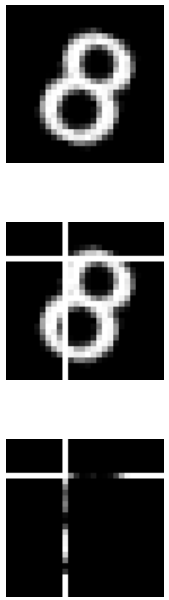
縦・横線ノイズ画像の生成（カスタムミニバッチデータストア）

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

```
classdef RowColNoiseImageDatastore < matlab.io.Datastore &...
        matlab.io.datastore.MiniBatchable
    methods
        function [miniBatchTable,info] = read(ds)
            patches = cell(ds.MiniBatchSize,1);
            noisyPatches = cell(ds.MiniBatchSize,1);
            noiseComponents = cell(ds.MiniBatchSize,1);
            for i = 1:ds.MiniBatchSize
                ds.CurrentFullImage = ...
                  im2double(ds.InputImageDatastore.readimage...
                  (ds.CurrentImageIndex)); % 画像読み出し
                patches{i} = ds.CurrentFullImage;
                randrow = randi(size(patches{i},1)); % ランダムに縦線位置を決定
                randcol = randi(size(patches{i},2)); % ランダムに横線位置を決定
                noisyPatch = patches{i}; % オリジナル画像取り出し
                noisyPatch(randrow,:) = 1; % 画像に縦線をいれる
                noisyPatch(:,randcol) = 1; % 画像に横線をいれる
                noisyPatches{i} = noisyPatch;
                noiseComponents{i} = noisyPatches{i} - patches{i};  %ノイズ画像
            end
            miniBatchTable = [table(noisyPatches) table(noiseComponents)];
        end
```

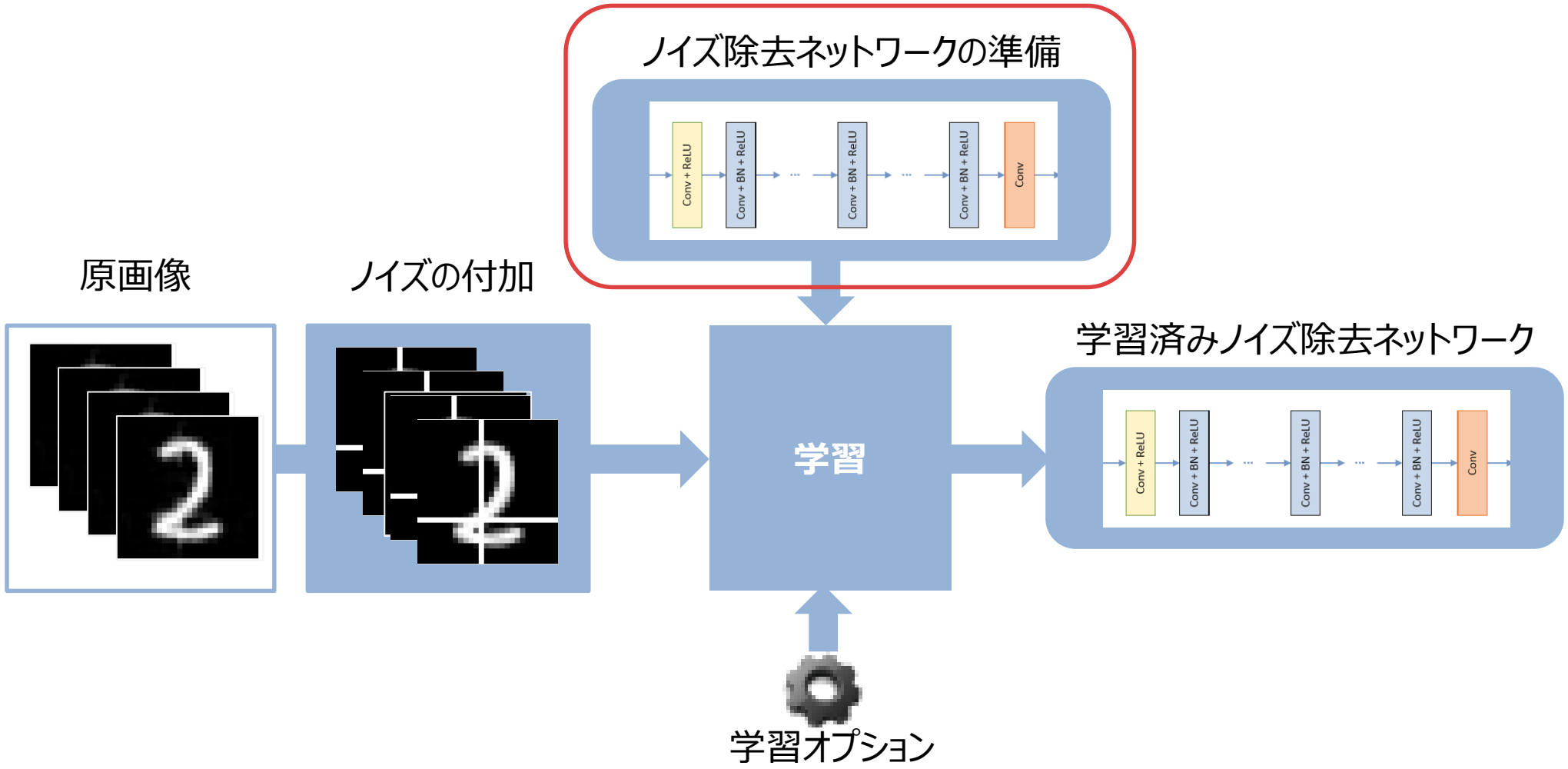
```
% カスタムミニバッチストアを呼び出し
trainSource = RowColNoiseImageDatastore(trainDigitData,...
    'MiniBatchSize',miniBatchSize,...
    'BatchesPerImage',1);

% 劣化画像と残差を生成
inputBatch = read(trainSource);
```

- MiniBatchDatastoreクラスを継承して独自のデータストアを定義可能
- カスタムデータフォーマットや前処理、水増しなど任意の処理が追加可能
- 事前に大量の水増し画像を生成しておく必要がない（学習実行時にデータ加工）



ノイズ除去ネットワークの学習ワークフロー

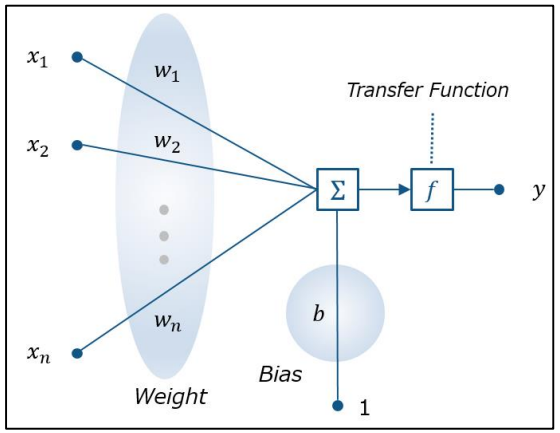
Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

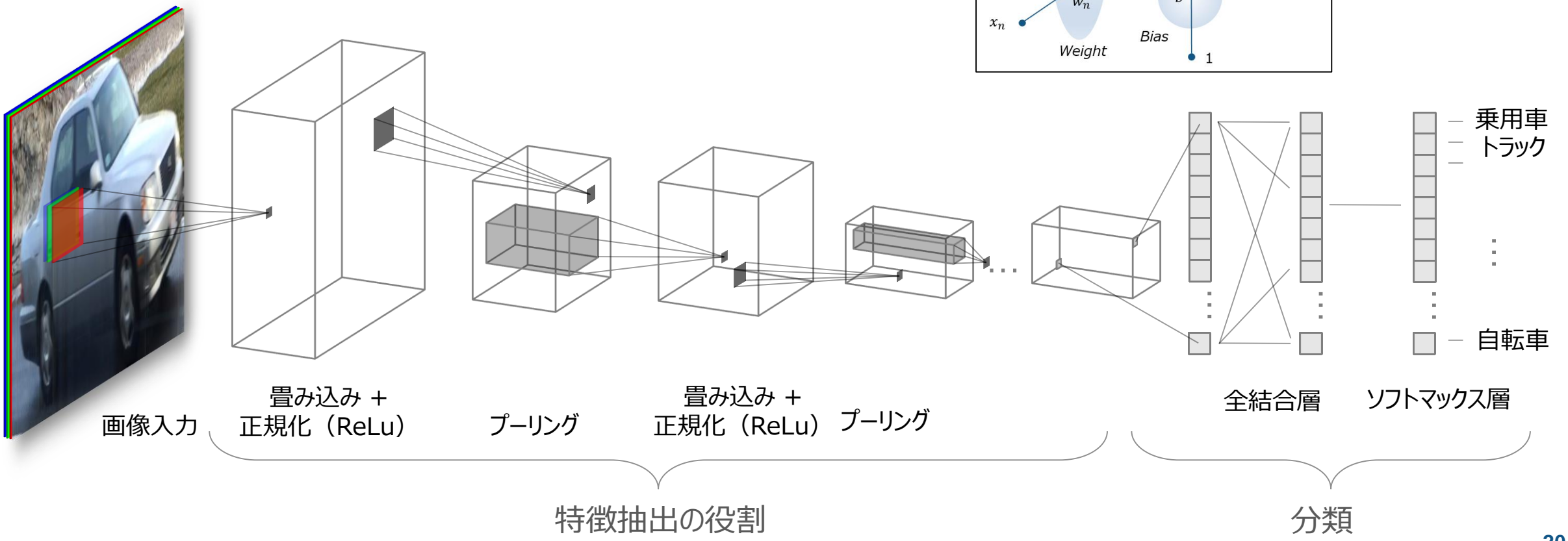


畳み込みニューラルネットワーク

畳み込み層・プーリング層・正規化層などを
積み重ねて作られた多層のニューラルネットワーク

分類のための畳み込みニューラルネットワーク(CNN)の例：





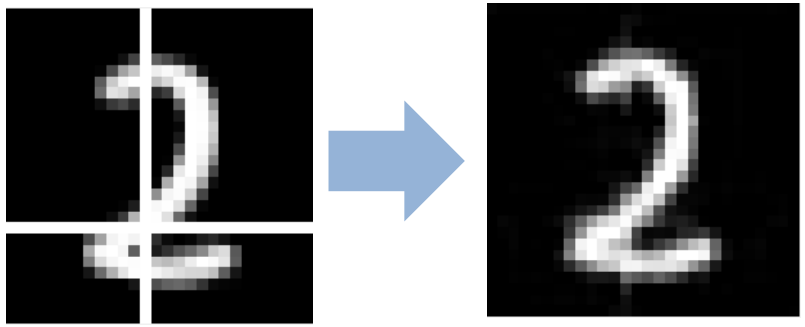
逆プーリング（Unpooling）

縦・横線ノイズ除去ネットワークの構築

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

ネットワークの定義

```
layers = [imageInputLayer([28 28 1], 'Normalization','none','Name','input');
 convolution2dLayer(3,32,'Padding',[1 1 1 1],'Name','conv1');
 reluLayer('Name','relu1');
 maxPooling2dLayer(2,'Stride',2,'Name','mpool1','HasUnpoolingOutputs',true);
 convolution2dLayer(3,32,'Padding',[1 1 1 1],'Name','conv2');
 reluLayer('Name','relu2');
 maxUnpooling2dLayer('Name','upool1');
 convolution2dLayer(3,32,'Padding',[1 1 1 1],'Name','conv3');
 reluLayer('Name','relu3');
 convolution2dLayer(3,1,'Padding',[1 1 1 1],'Name','conv4');
 regressionLayer('Name','routput');]
```



畳み込み層・プーリング層・逆プーリング層
などの層を積み上げて定義

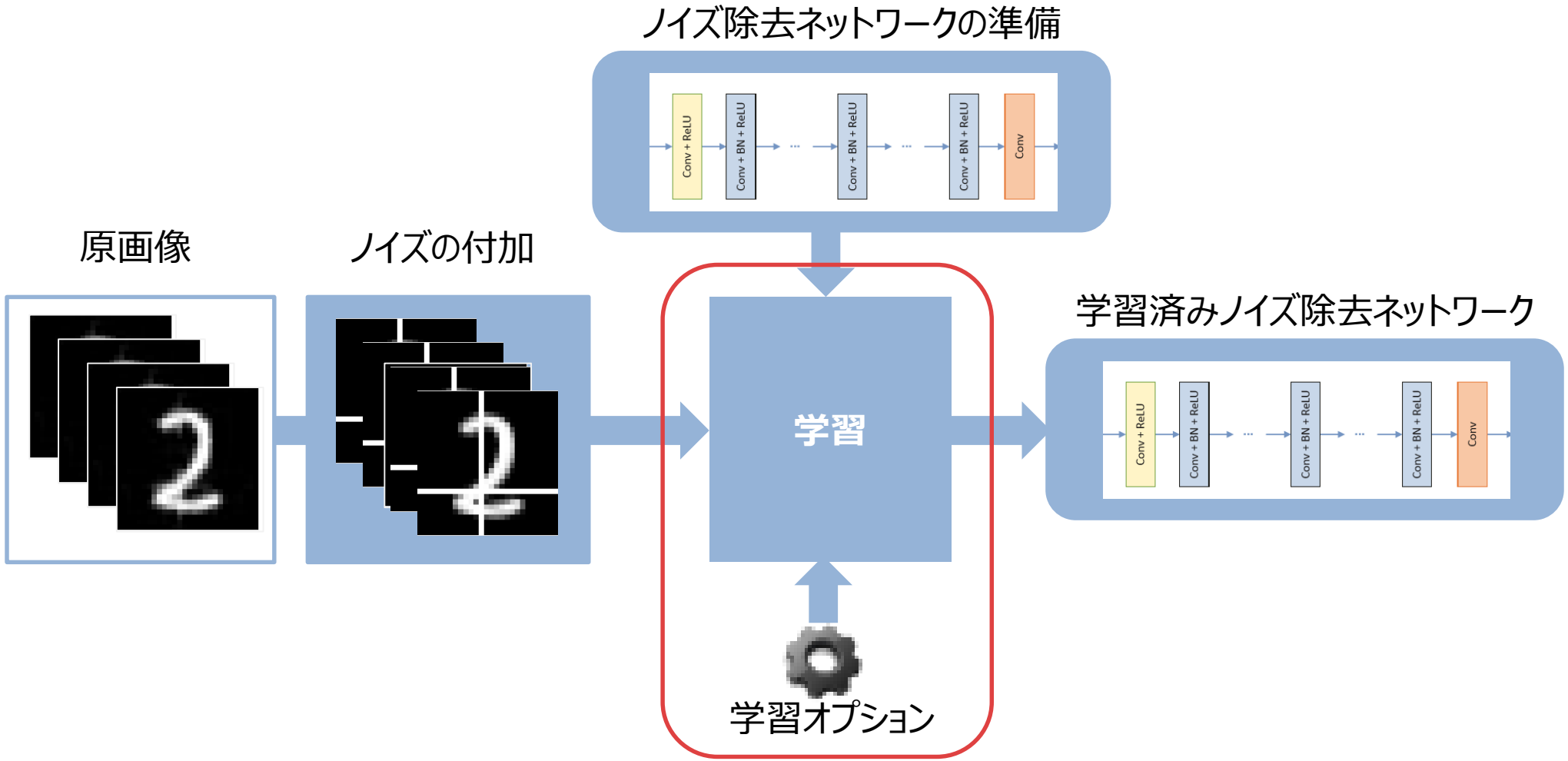
画像出力に対して、回帰層を使い学習させる

```
layers = layerGraph(layers);
layers = connectLayers(layers,'mpool1/indices','upool1/indices');
layers = connectLayers(layers,'mpool1/size','upool1/size');
```

最大プーリング層で選択した画素の
位置情報を最大逆プーリング層へ渡す

ノイズ除去ネットワークの学習ワークフロー

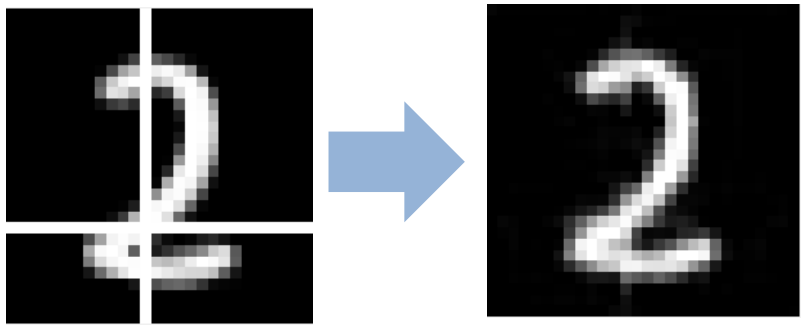
Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox



縦・横線ノイズ除去ネットワークの学習

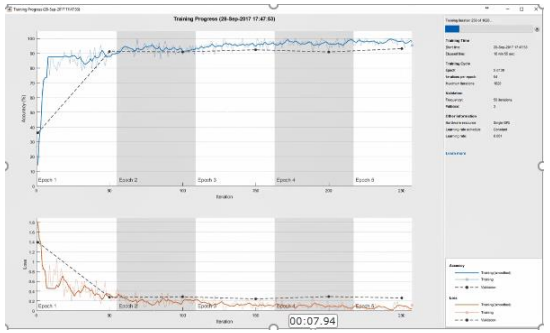
Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

28×28 ピクセルの画像（数字）のノイズ除去を行う例題でのネットワーク学習の例



オプションの設定と学習

```
options = trainingOptions('adam', 'MaxEpochs',15,...
                                  'Plots','training-progress');
net = trainNetwork(XTrain, XResponse, layers, options);
```



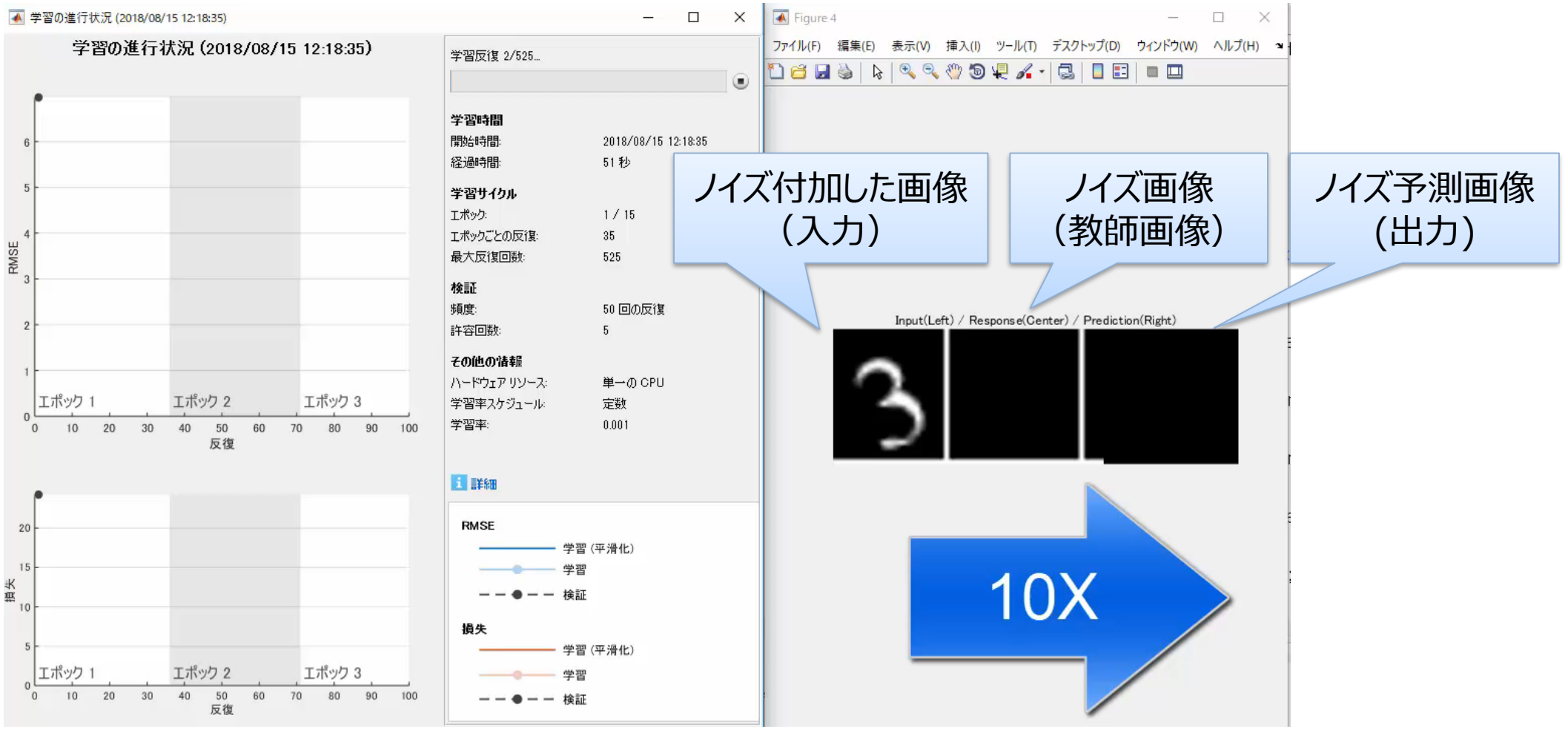
学習率や最大反復数などを定義して学習

'Plots'オプション指定で、学習過程をグラフで可視化

GPU有無を自動で判定。あればGPU,なければCPUで学習。

https://jp.mathworks.com/help/releases/R2018a/nnet/ref/trainnetwork.html

縦・横線ノイズ除去ネットワークの学習



ノイズ除去ネットワークの学習ワークフロー

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

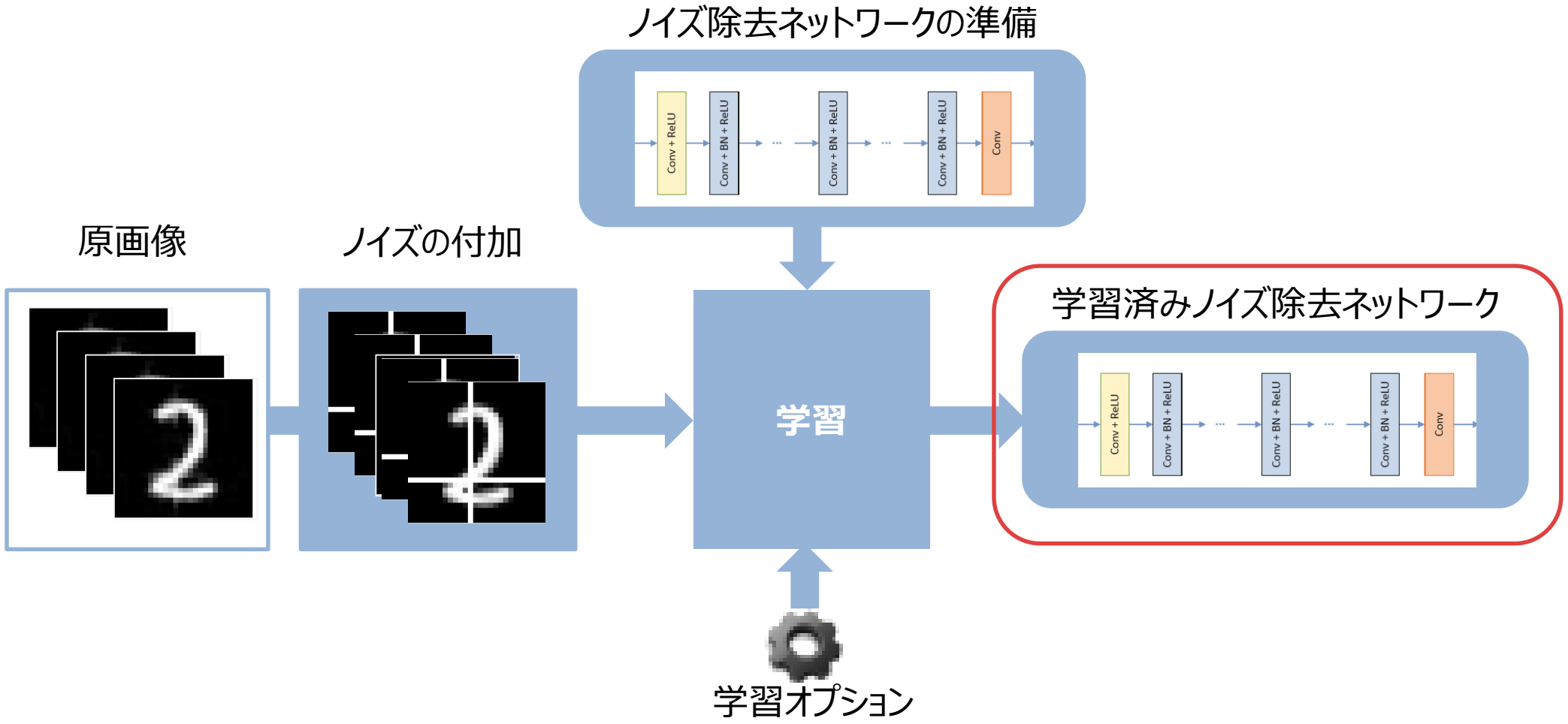


Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

縦・横線ノイズ除去ネットワークを使った予測

28×28 ピクセルの画像（数字）のノイズ除去を行う例題でのネットワーク学習の例

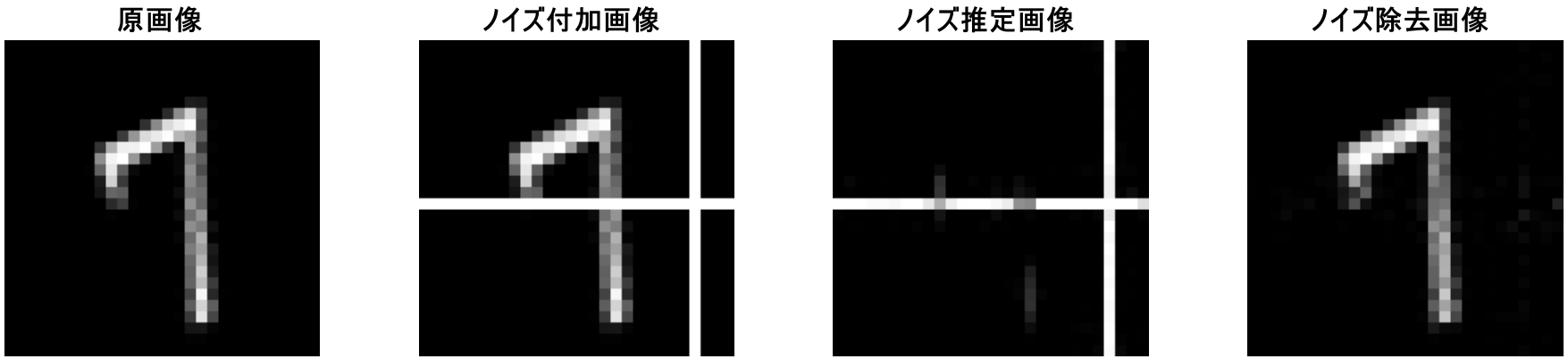
学習済みネットワークを使った予測

```
noisyImages = read(testSource);
predI = predict(net,noisyImages);
denoisedImage = noisyImages.noisyPatches{1} - predI(:,:,1,1);
```

テストデータセットからノイズ画像を読み出し、

predictメソッドでノイズ成分を予測

ノイズ成分を減算することでオリジナル画像を復元

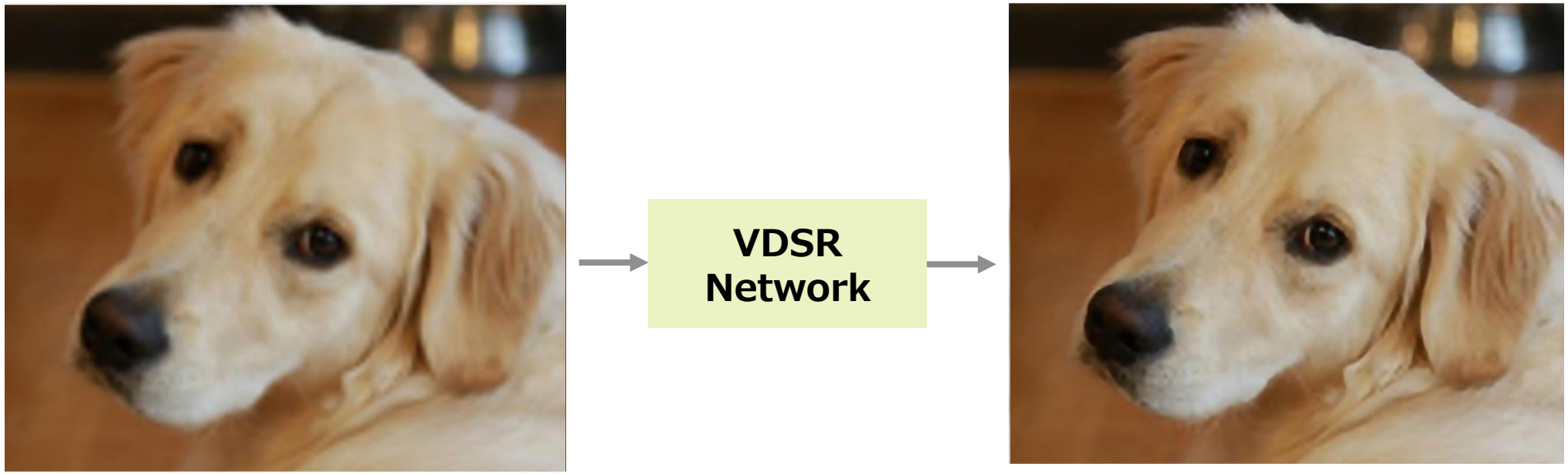


アジェンダ

- ディープラーニングによる画像補正の基礎
- ノイズ除去ネットワークの学習ワークフロー
- ディープラーニングによる画像補正の応用例
- まとめ

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

超解像変換
Very-Deep Super-Resolution (VDSR)[1] ネットワーク



画像品質比較

評価指標	双三次補間での拡大画像	拡大画像にVDSR適用
PSNR	38.47	**39.44**
SSIM	0.9861	**0.9878**

[1] Kim, J., J. K. Lee, and K. M. Lee. "Accurate Image Super-Resolution Using Very Deep Convolutional Networks." Proceedings of the IEEE® Conference on Computer Vision and Pattern Recognition. 2016, pp. 1646-1654.

JPEG圧縮のアーチファクト低減

Image Processing Toolbox
Neural Network Toolbox
Parallel Computing Toolbox

- JPEG圧縮は非可逆変換、アーチファクト(ブロッキングノイズ)発生
- DnCNNで学習し、アーチファクト低減

画像品質比較

評価指標	JPEG圧縮	JPEG圧縮 →DnCNN処理後
PSNR	28.802	**29.338**
SSIM	0.9490	**0.9544**

アジェンダ

- ディープラーニングによる画像補正の基礎
- ノイズ除去ネットワークの学習ワークフロー
- ディープラーニングによる画像補正の応用例
- まとめ

まとめ

- ディープラーニングによる画像補正の基礎
  - DnCNNを使ったガウス雑音の除去
  - denoisingNetwork/denoiseImageで簡単に実行
  - 従来のフィルタより、柔軟かつ高精能なノイズ除去ができる可能性
- ノイズ除去ネットワークの学習ワークフロー
  - 任意のノイズに対するノイズ除去
  - 新たなネットワークの構築・学習を効率的に実現できる仕組み
- ディープラーニングによる画像補正の応用例
  - 超解像変換(Very-Deep Super-Resolutionネットワーク)
  - JPEG圧縮のアーチファクト低減