

# ディープラーニングによる画像認識の基礎と実践ワークフロー

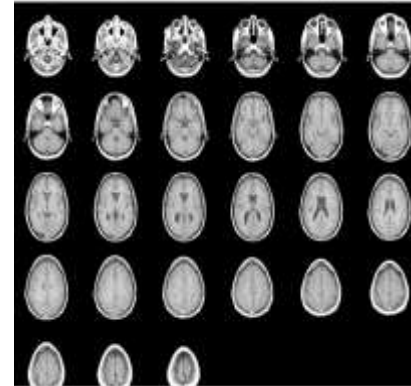
MathWorks Japan  
アプリケーションエンジニアリング部  
アプリケーションエンジニア  
福本 拓司

# 一般的におこなわれる目視による評価

製造ライン



医用データ

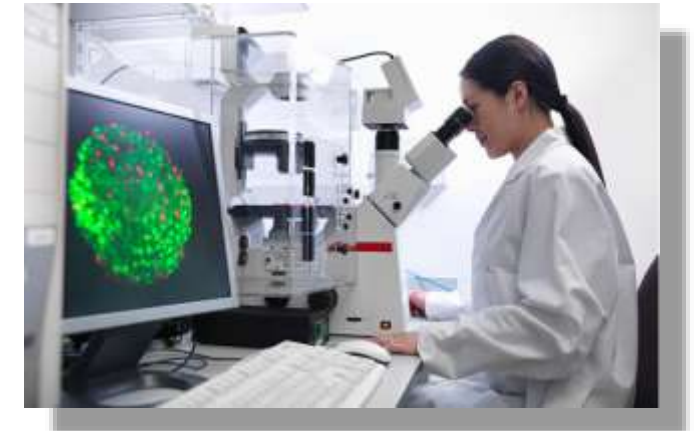
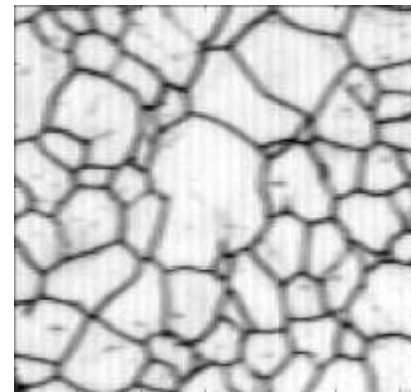


現場での目視

作業現場・インフラ



研究データ

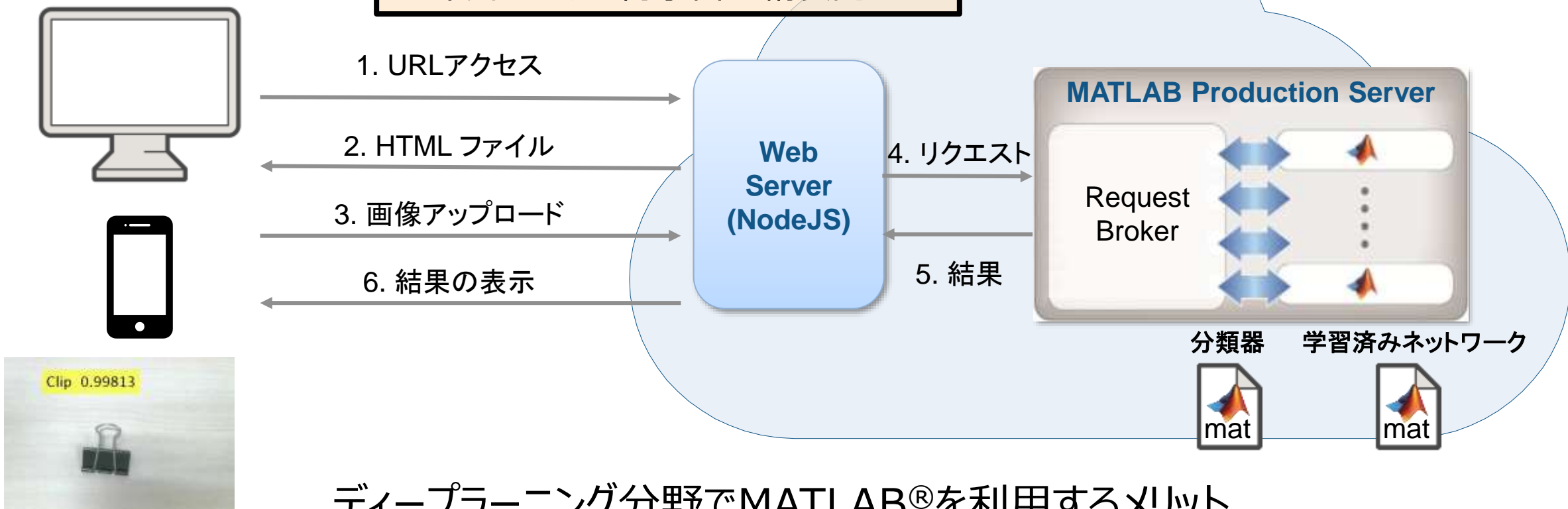


大量画像の収集  
専門家によるチェック

# スマートフォンで撮影した映像をその場で評価

MATLAB® Production Server™

- ・遠隔地でスマートフォンで判定
- ・活用しながら画像収集の継続、ネットワークの再学習⇒精度向上

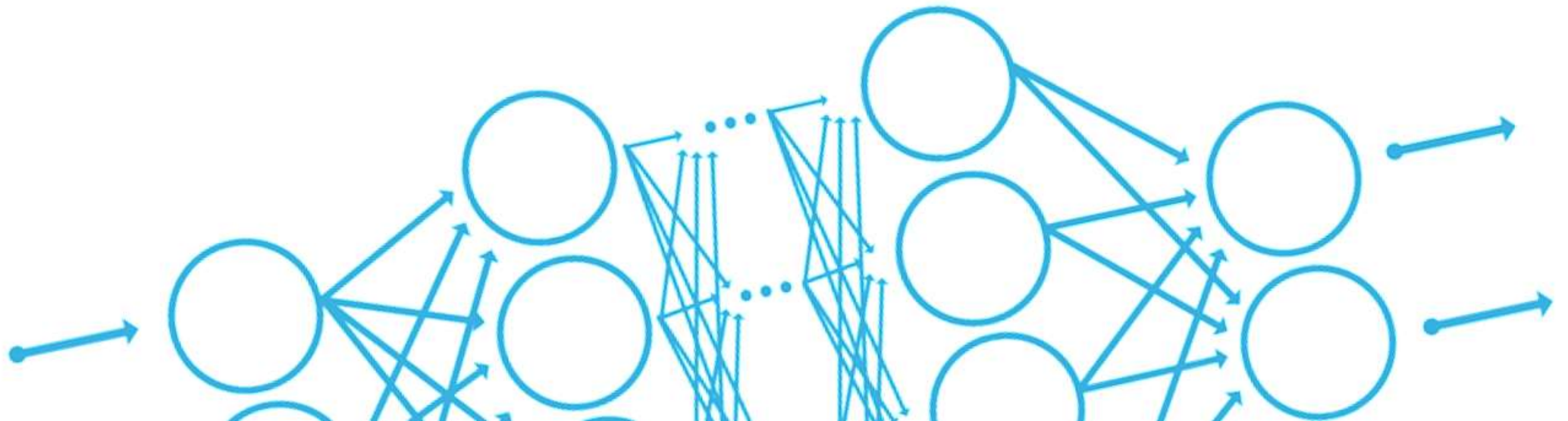


ディープラーニング分野でMATLAB®を利用するメリット

- ・画像があれば簡単にはじめられるフレームワーク
- ・作ったネットワークをすぐにシステムに統合できる

# Agenda

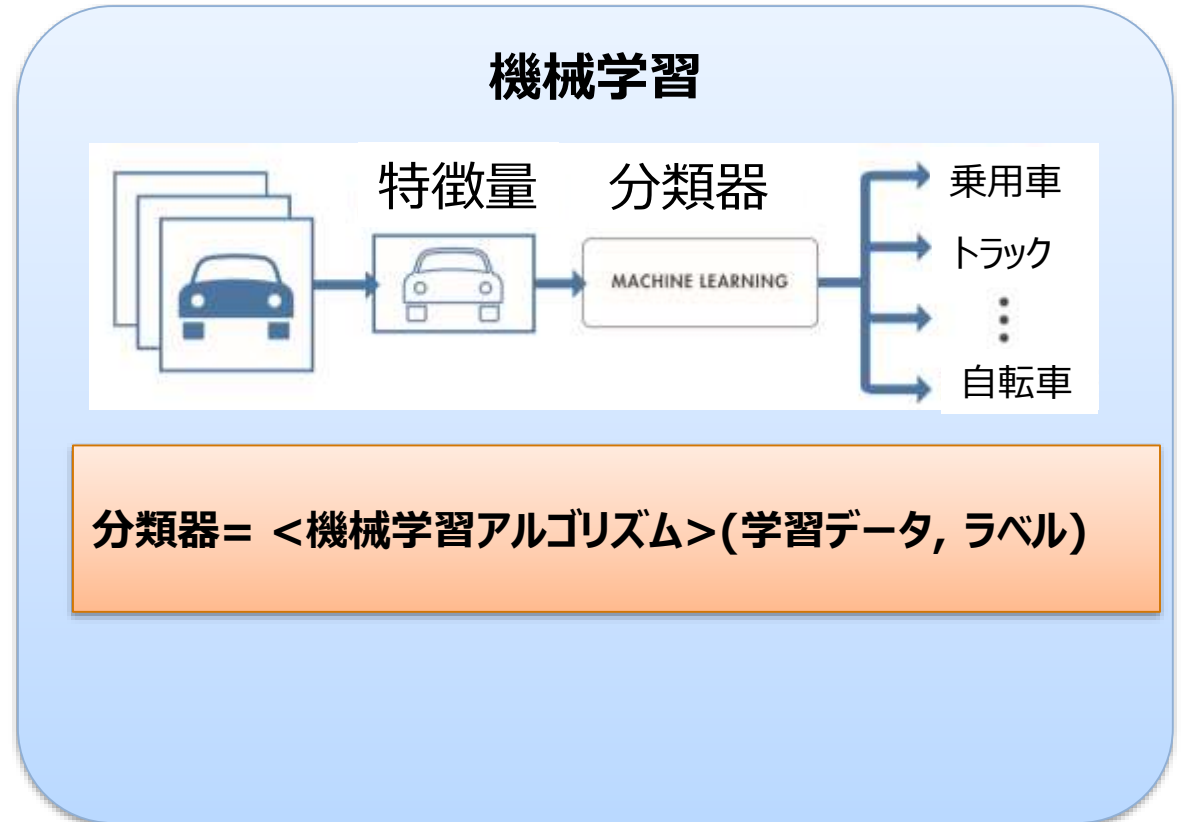
- これからはじめるディープラーニング活用ワークフロー
- ディープラーニングの異常検知への適用
- 学びを助けるリソース



# 機械学習を利用すべき場面とは？



数値で条件を指定し切り分け  
 明確な切り分けが可能の場合に○



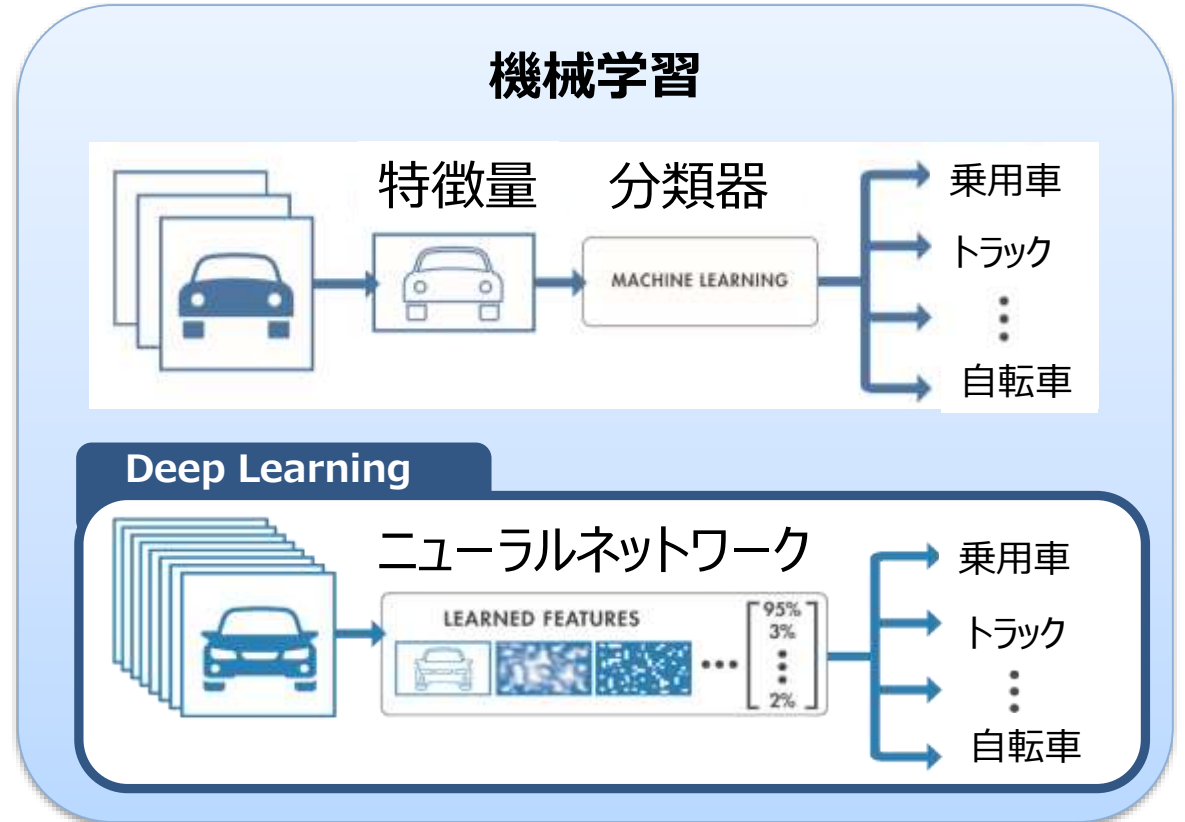
画像データを使って分類器を学習  
 うまくくと柔軟な切り分けができる

# 機械学習を利用すべき場面とは？



数値で条件を指定し切り分け

明確な切り分けが可能の場合に○



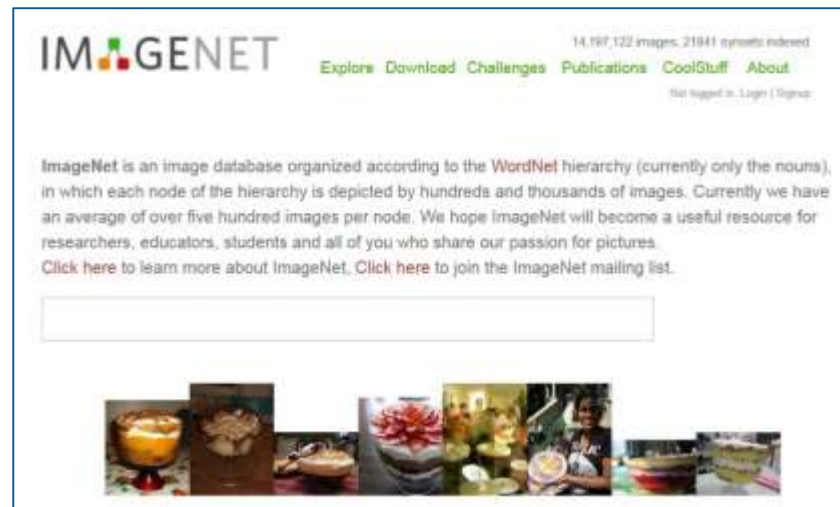
画像データを使って分類器を学習

うまくくと柔軟な切り分けができる

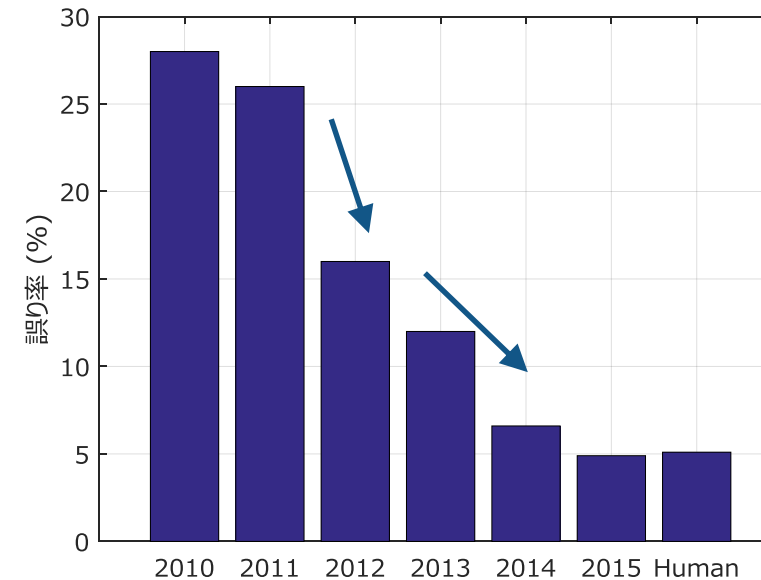
# ILSVRC (ImageNet Large Scale Visual Recognition Challenge)

## ImageNet とは？

- 画像認識の研究のための大規模な画像データベース
- 1000のカテゴリを持ち、カテゴリ毎に1000枚の画像



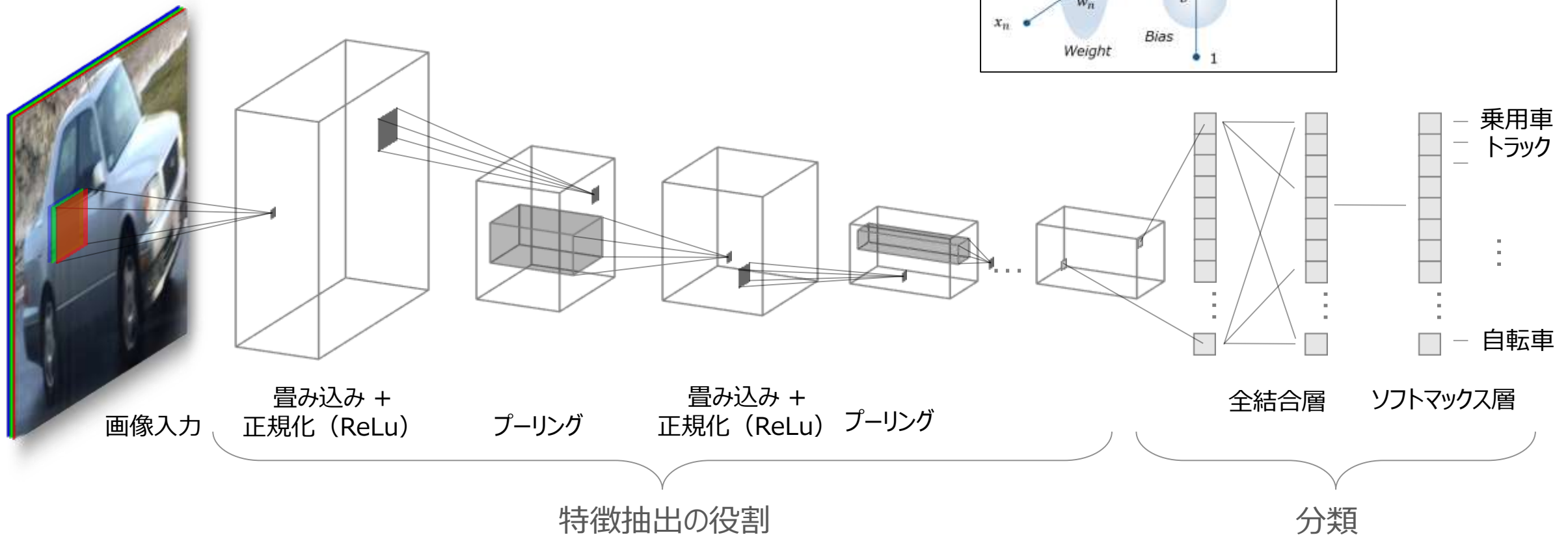
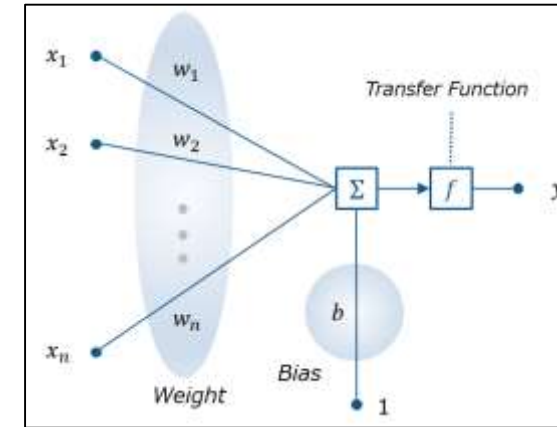
<http://www.image-net.org/>



**CNNの登場によって10%以上の性能向上(2012)**  
**GoogLeNet, VGG等の深いCNNが登場(2014)**

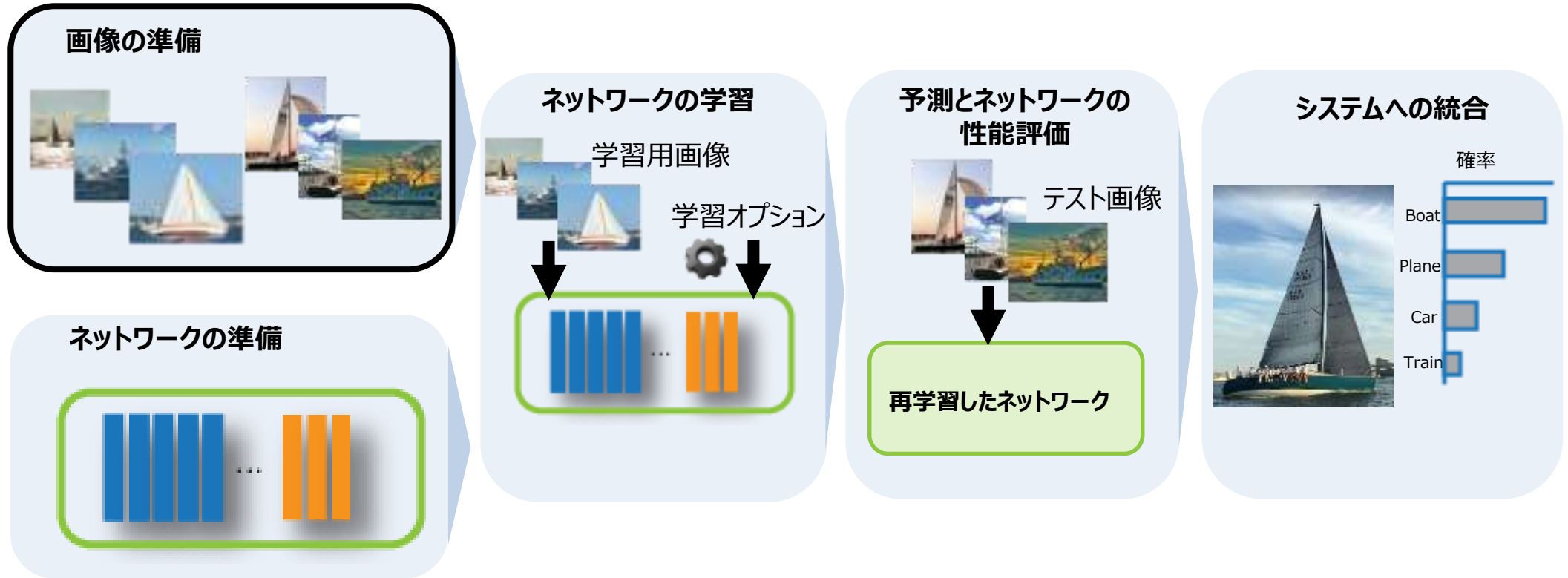
# 畳み込みニューラルネットワーク(Convolutional Neural Network)

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





# MATLABによるディープラーニングワークフロー

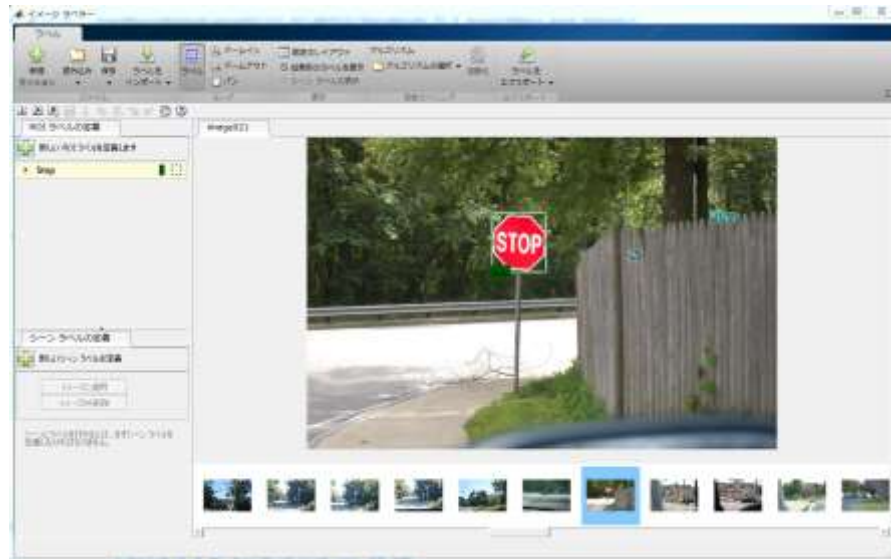


# 画像のラベル付けをサポート Image Labeler

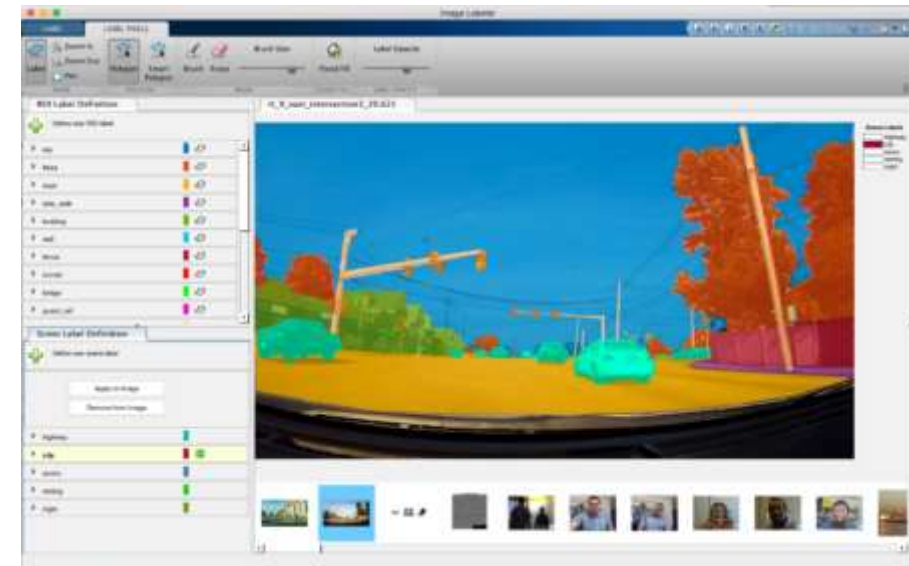
R2017b

- シーン、オブジェクト、ピクセルに対してのラベリング
- 独自のアルゴリズムによる自動化

## ROIのラベリング



## ピクセルのラベリング



# 効率よい学習と精度向上のためのデータ拡張

- 大量の画像セットにメモリ効率の良いアクセス

- imageDatastore

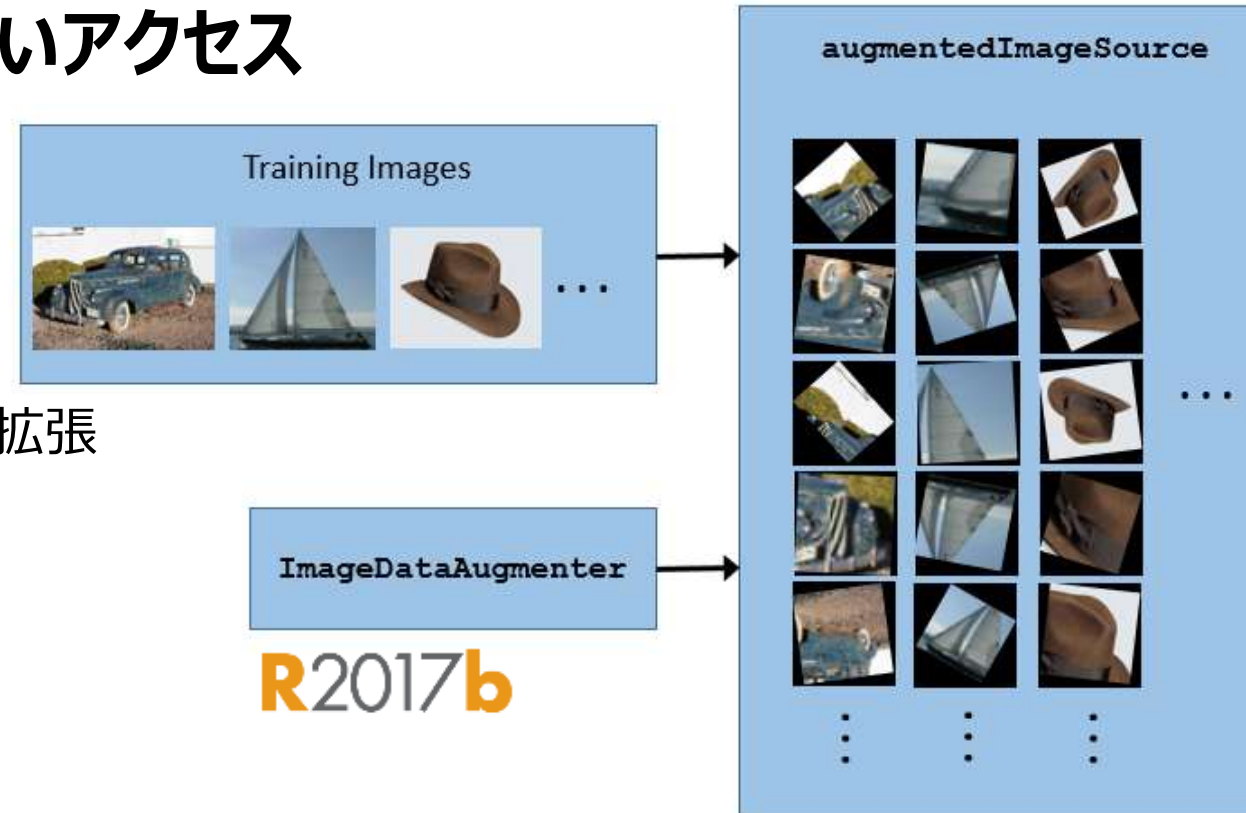
- データ拡張

- 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

拡張された画像セット



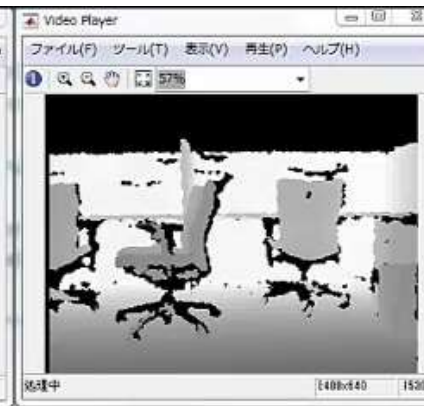
# 各種カメラデバイスからのデータ取り込み

## Image Acquisition Toolbox

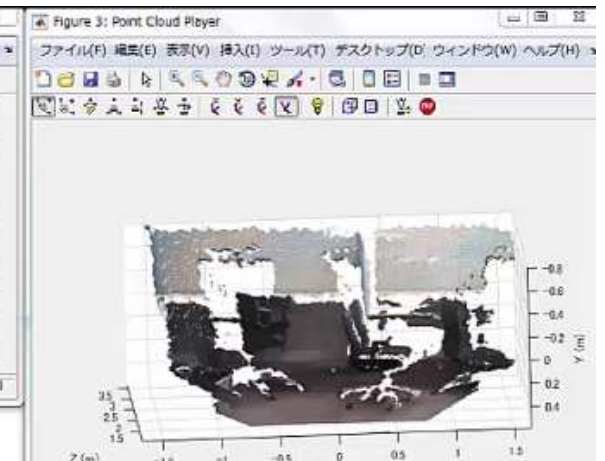
- 業界標準のHWからの動画像取込み機能を提供
  - フレームグラバ
    - Analog
    - Camera Link
  - DCAM 互換 FireWire (IIDC 1394)
  - GigE Vision
  - USB3 Vision
  - IPカメラ
- Microsoft Kinect



RGB

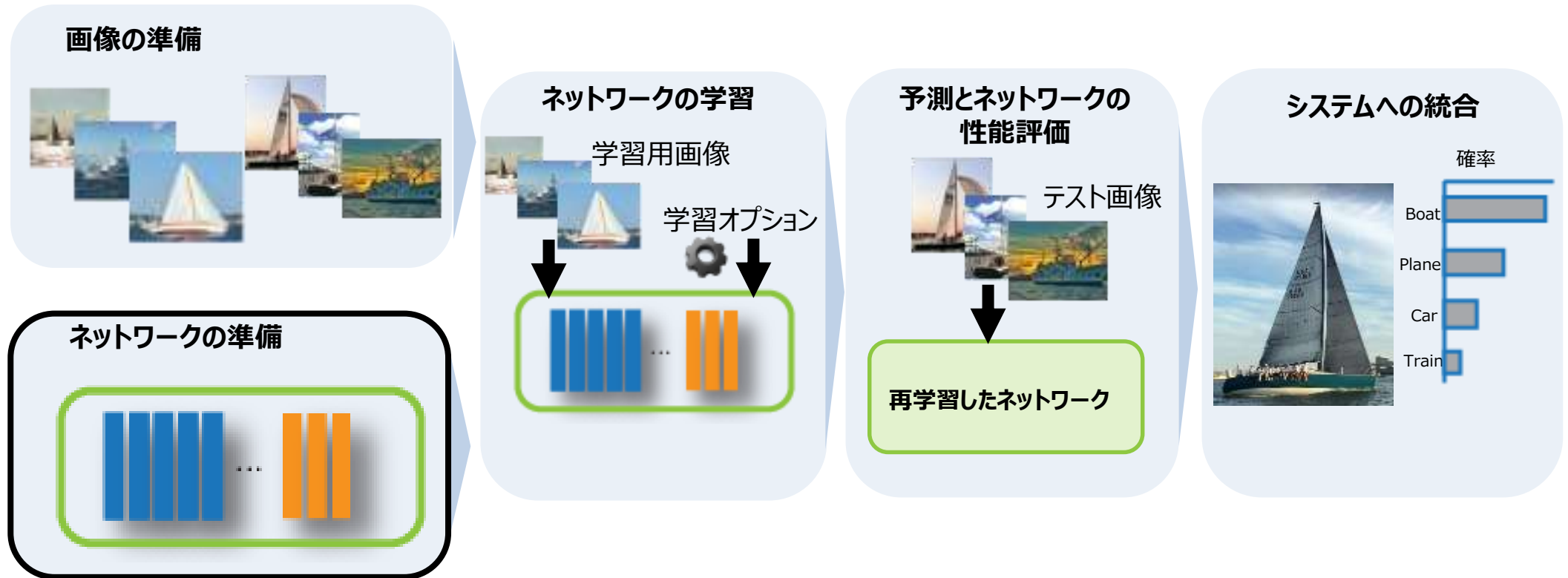


Depth



ポイントクラウド

# MATLABによるディープラーニングワークフロー



# 畳み込みニューラルネットの構築と学習

28×28 ピクセルの画像（数字）を認識させる例題でのネットワーク構築の例



```
layers = [ ...  
    imageInputLayer([28 28 1], 'Normalization', 'none');  
    convolution2dLayer(5, 20);  
    reluLayer();  
    maxPooling2dLayer(2, 'Stride', 2);  
    fullyConnectedLayer(10);  
    softmaxLayer();  
    classificationLayer()];
```

畳み込み層・プーリング層・正規化層  
などの層を積み上げて定義

```
opts = trainingOptions('sgdm', 'MaxEpochs', 50);  
net = trainNetwork(XTrain, TTrain, layers, opts);
```

学習率や最大反復数などを定義して  
学習の関数を呼び出す

**GPU有無を自動で判定。あればGPU,なければCPUで学習。**

<http://www.mathworks.com/help/releases/R2017b/nnet/ref/trainnetwork.html>

# ネットワークをスクラッチで作るハードル

精度が高いネットワークはどのような学習をしているのか

・100万枚の画像セット

## AlexNet

■ NVIDIA® GeForce® GTX 580 2機 による 5~6日間の学習

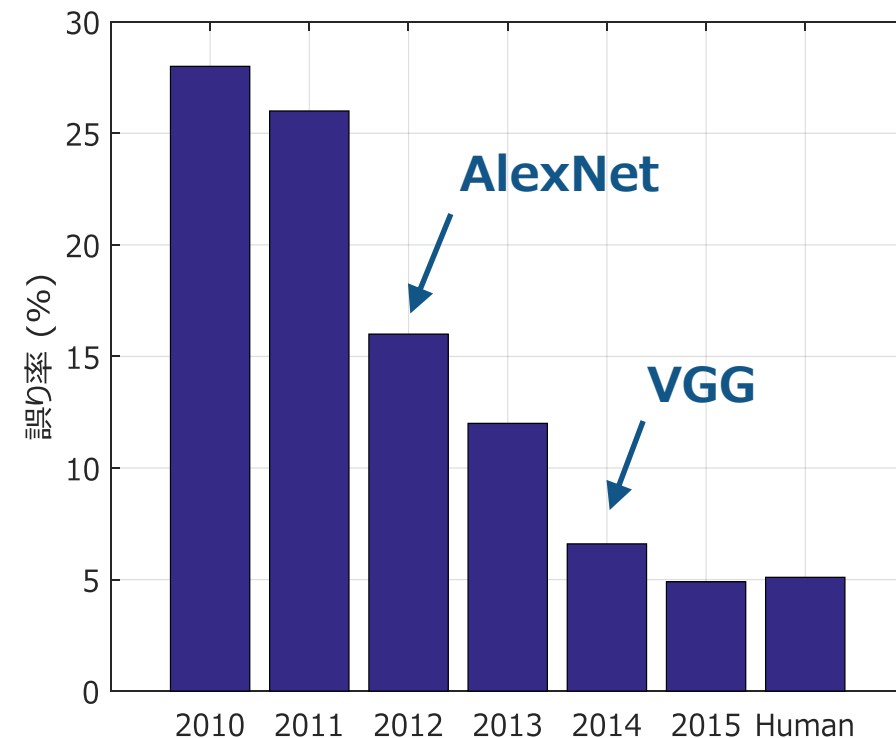
## VGG Net

■ NVIDIA® GeForce® TITAN Black 4機 による 2~3週間の学習

ネットワークをスクラッチで作る際のハードル

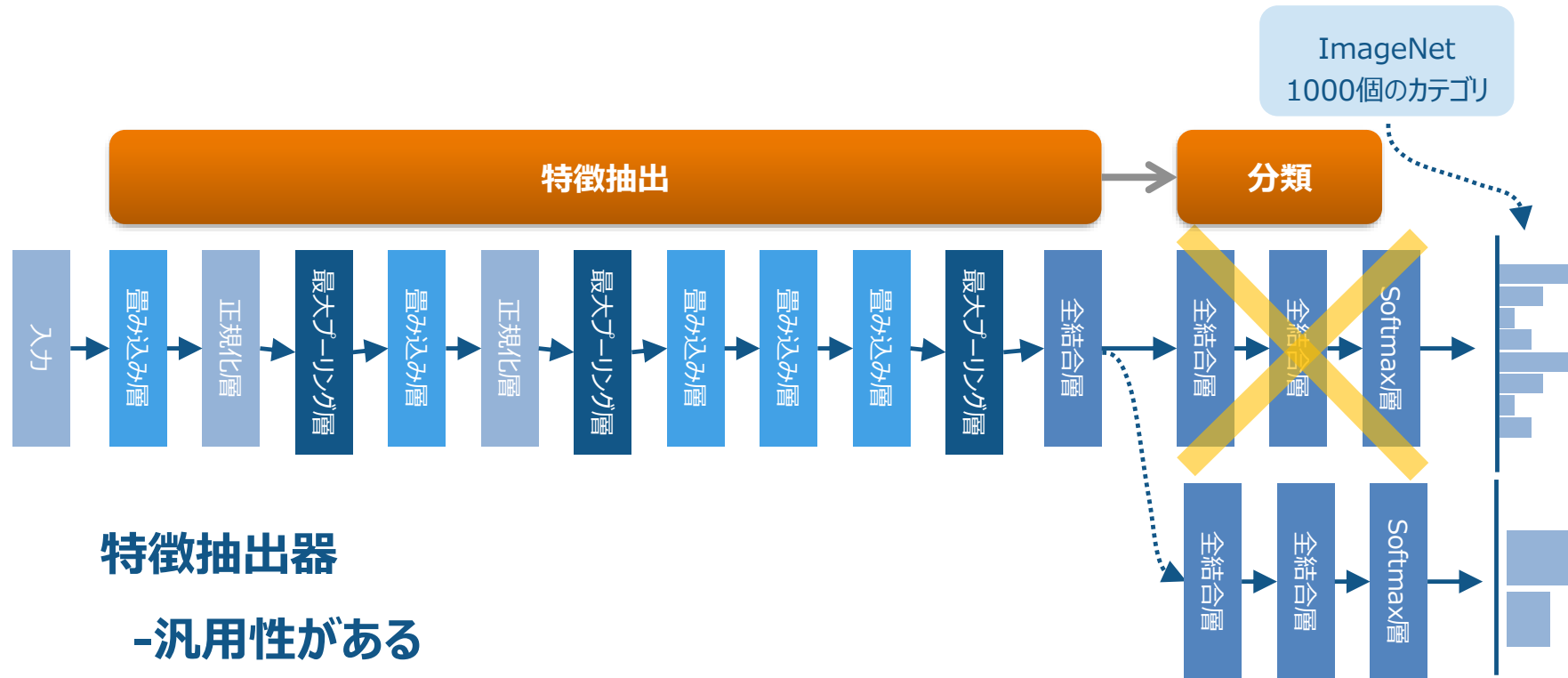
- ・ネットワーク構築の知識
- ・大量の画像セット
- ・膨大な計算コスト

ILSVRC 2010 - 2015



Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks" In NIPS, pp.1106-1114, 2012  
 K. Simonyan, A. Zisserman "Very Deep Convolutional Networks for Large-Scale Image Recognition" arXiv technical report, 2014

# CNN と転移学習



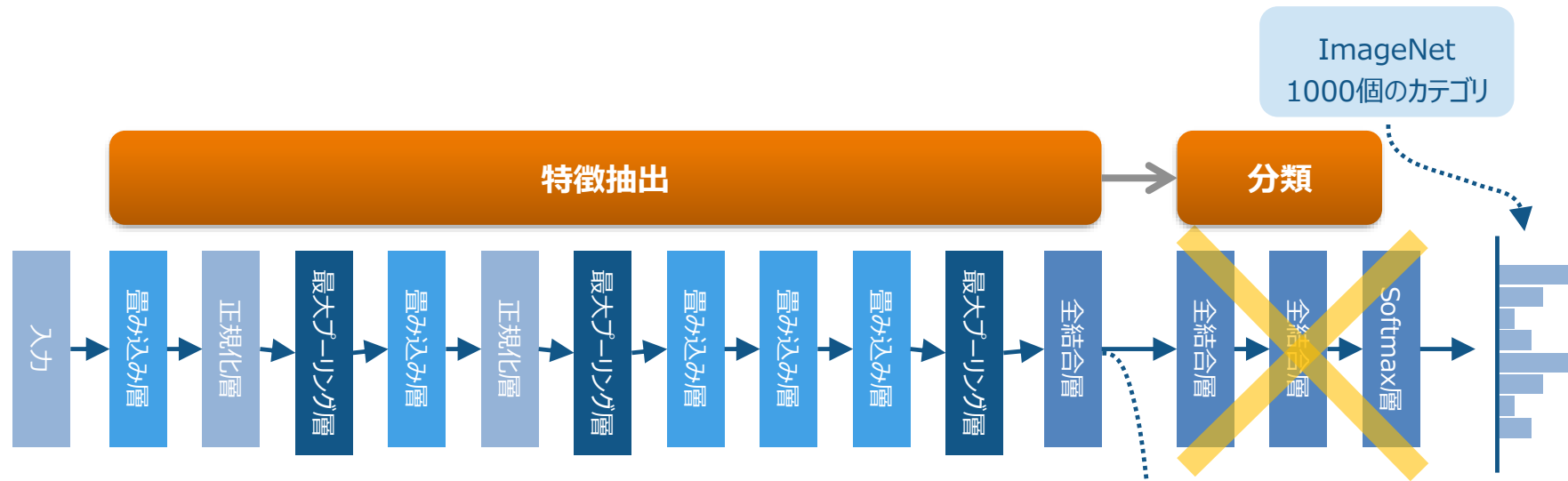
## 特徴抽出器

- 汎用性がある
- 多層だと学習させるために画像が大量に必要

優れた既存ネットワークの特徴抽出器 + 独自分類器



# CNN と転移学習



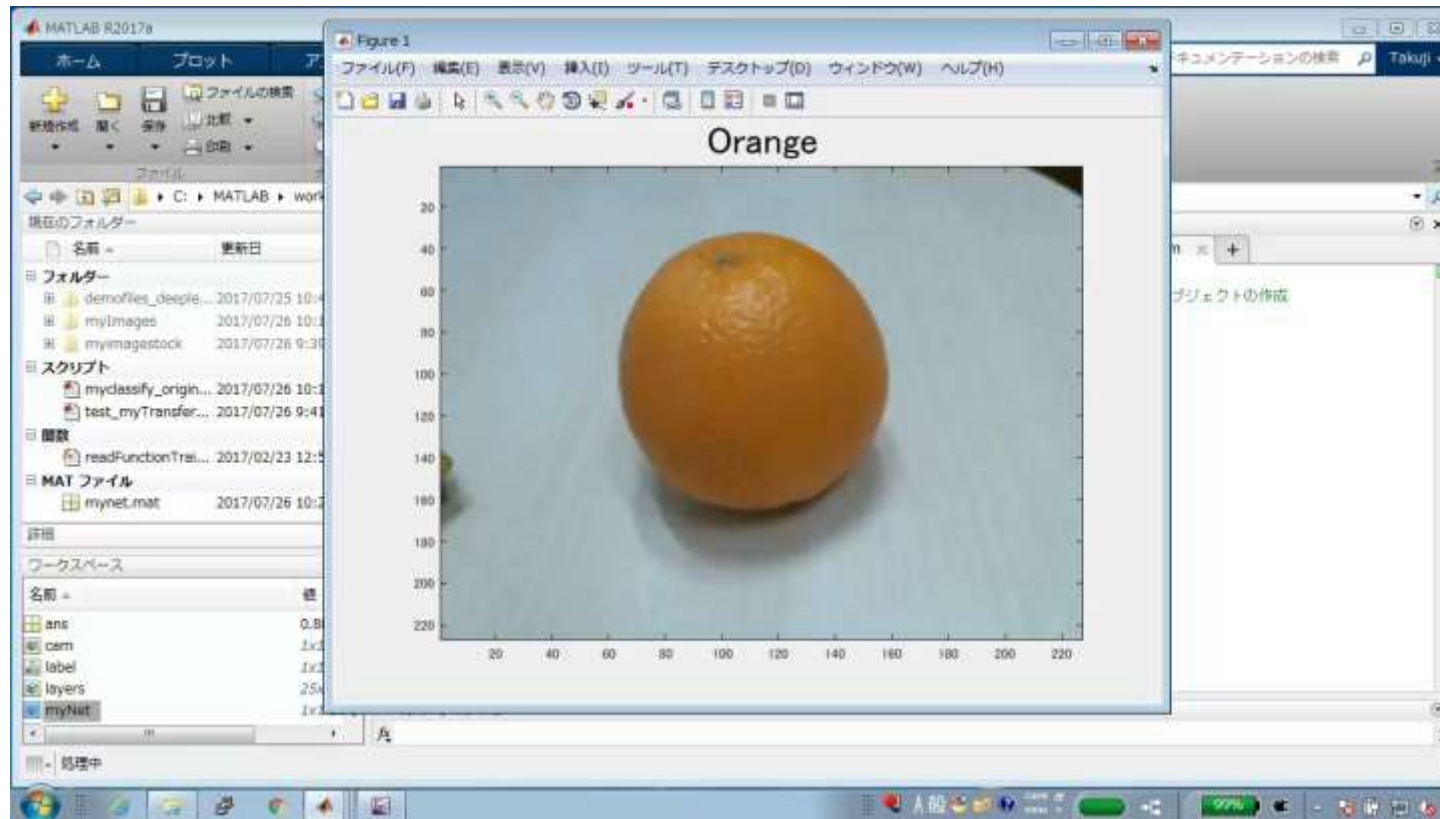
## 特徴抽出器

- 汎用性がある
- 多層だと学習させるために画像が大量に必要

優れた既存ネットワークの特徴抽出器 + 独自分類器

# ディープラーニングによる物体認識

ディープラーニング：10行でできる転移学習 ～画像分類タスクに挑戦～



学習した種類：

- オレンジ
- みかん
- グレープフルーツ(ルビー)
- グレープフルーツ(ホワイト)
- レモン

学習画像数：各 20 枚

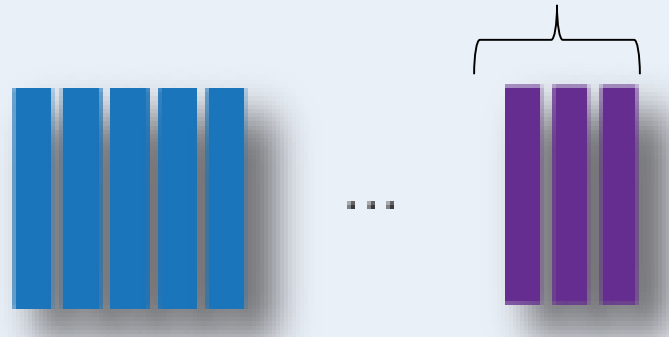
- 要件を満たすPC&MATLAB環境
  - 学習済みAlexnet
  - 画像セット
- で10行のコーディングで始められます

<https://www.youtube.com/watch?v=XMcHiMIT8iE>

# 転移学習ワークフロー

## 学習済みネットワーク読み込み

特定のタスクのために  
学習された分類器



画像数 100万枚以上  
クラス 1000以上

# 転移学習ワークフロー

## 学習済みネットワーク読み込み

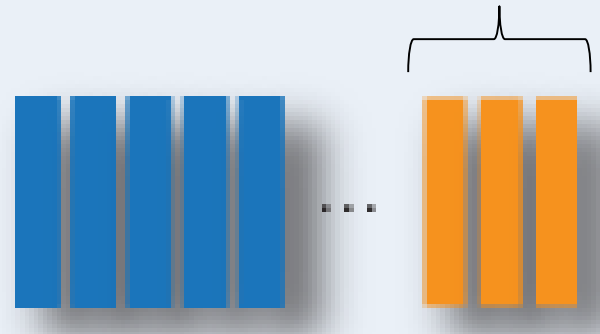
特定のタスクのために  
学習された分類器



画像数 100万枚以上  
クラス 1000以上

## 最後段の層を いくつか置き換え

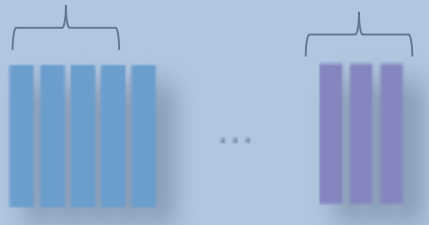
新しいデータを  
学習するための新しい層



# 転移学習ワークフロー

## 学習済みネットワーク読み込み

特定のタスクのために  
学習された分類器



画像数 100万枚以上  
クラス 1000以上

## 最後段の層を いくつか置

新しいデー  
タのための新

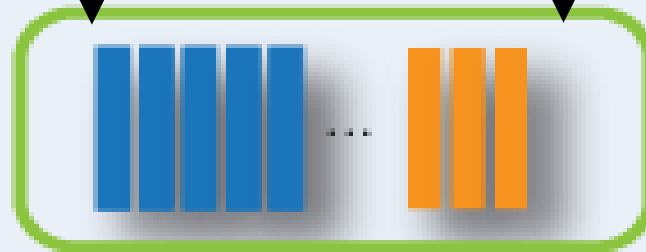


## ネットワークの学習



学習用画像

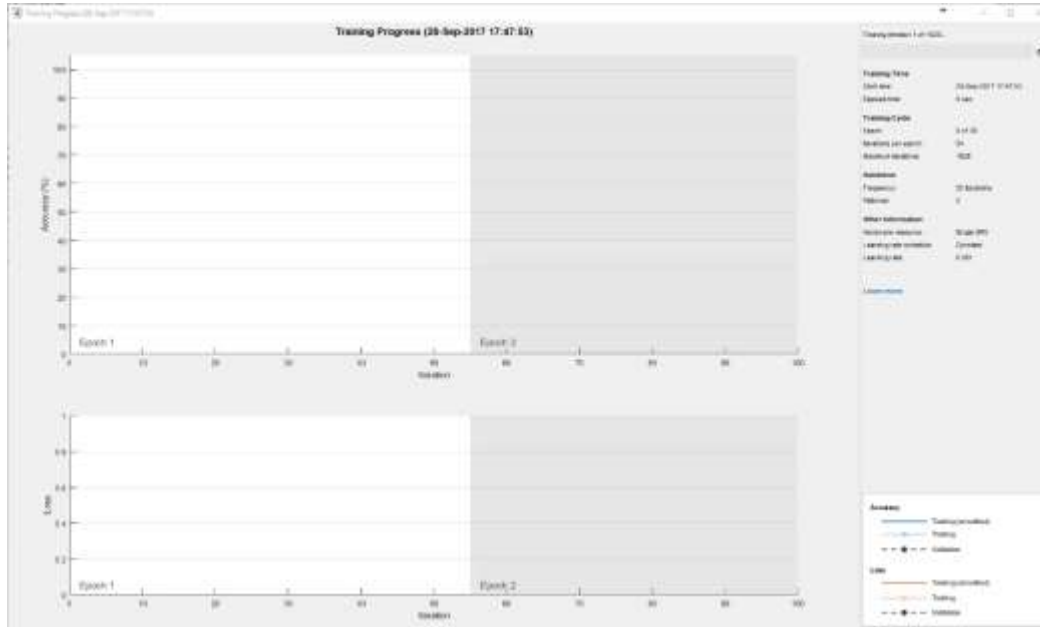
学習オプション



画像 数100枚  
クラス 数10カテゴリ

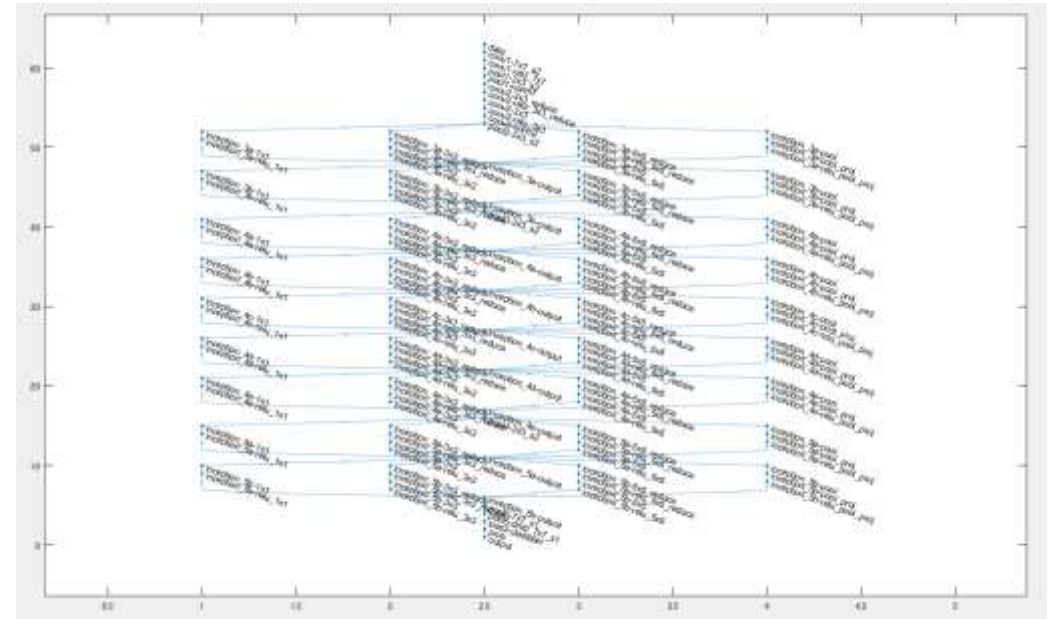
# 豊富な可視化機能

## 学習進捗の可視化と学習停止



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

## ネットワークの可視化



```
lgraph = layerGraph(net);
plot(lgraph)
```

# 転移学習ワークフロー

## 学習済みネットワーク読み込み

特定のタスクのために  
学習された分類器



画像数 100万枚以上  
クラス 1000以上

## 最後段の層を いくつか置

新しいデー  
タのための新

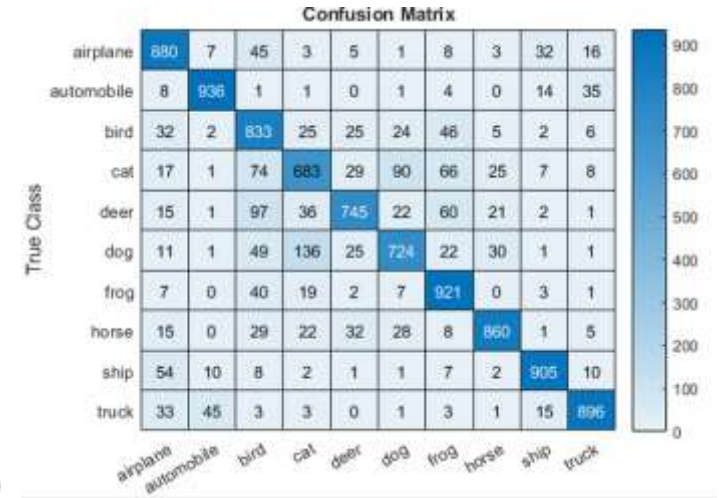


## 予測とネットワークの 性能評価



再学習したネットワーク

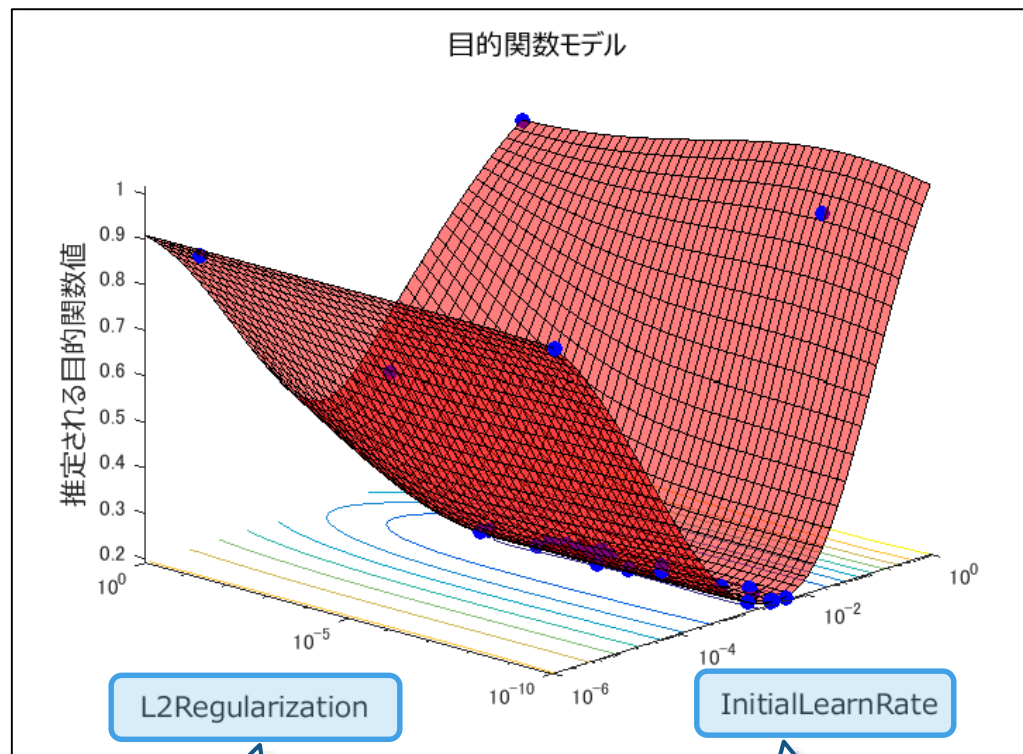
## 混合行列/ヒートマップ表示



```
[cmat,classNames] =
confusionmat(testLabels,predictedLabels);
h = heatmap(classNames,classNames,cmat);
```

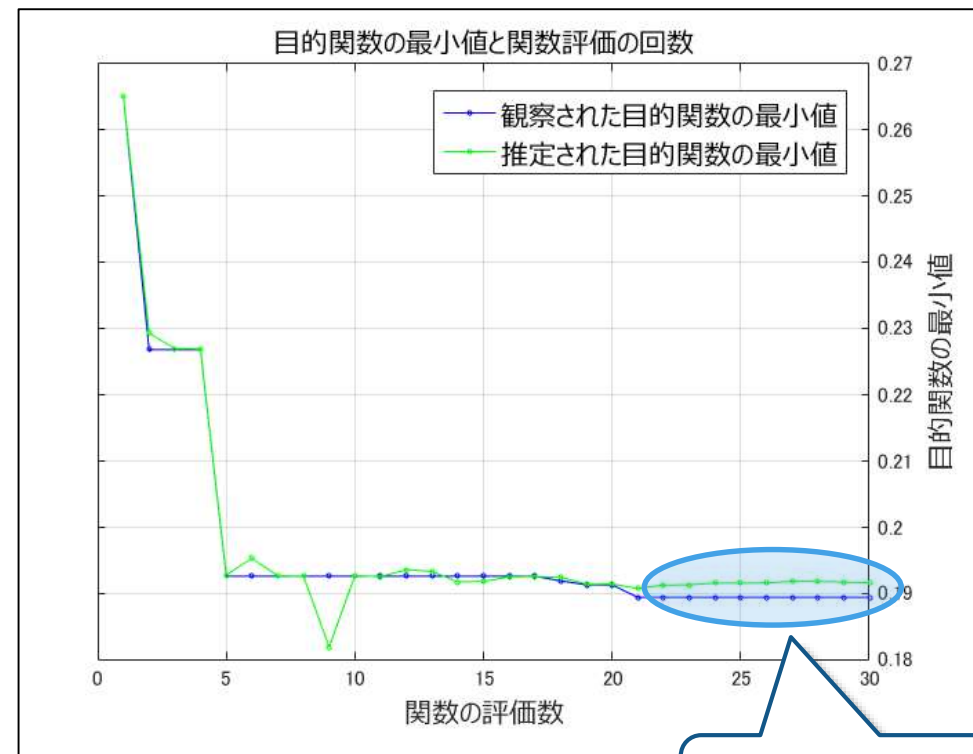
# パラメータの自動最適化（ベイズ最適化）

以下は、CNN におけるパラメータ調整にベイズ最適化を用いた例



L2正則化の係数

学習率の初期値



最適なパラメータへ収束



# 転移学習のための学習済みモデル読み込み

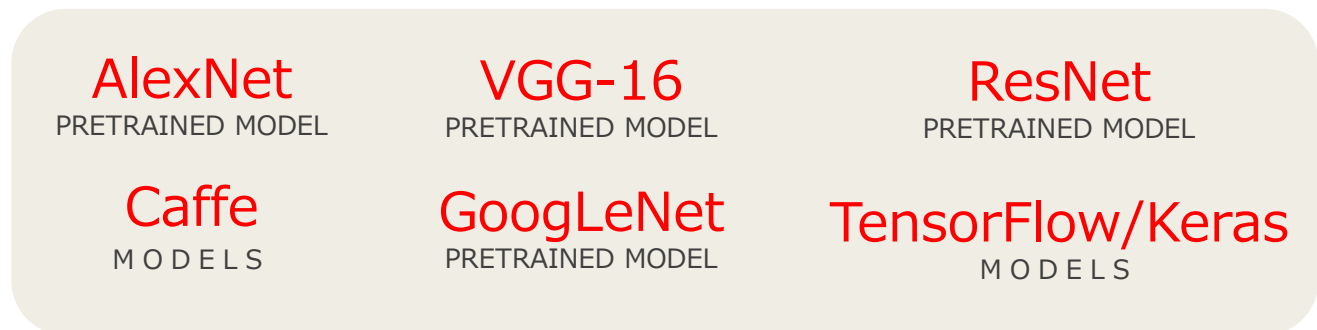
## 学習済みネットワーク\*

- AlexNet
- VGG-16
- VGG-19
- GoogLeNet **New**
- Resnet50 **New**
- InceptionV3 **New**

\* 一行でモデル読み込み

## 他のフレームワークのモデル読み込み

- Caffe Model Importer
- TensorFlow/Keras Model **New** Importer



転移学習 + 最適化で多くのことを試すことができます。  
どの程度タスクが難しいのか知ることが重要です。

# 転移学習ワークフロー

## 学習済みネットワーク読み込み

特定のタスクのために  
学習された分類器



画像数 100万枚以上  
クラス 1000以上

## 最後段の層を いくつか置

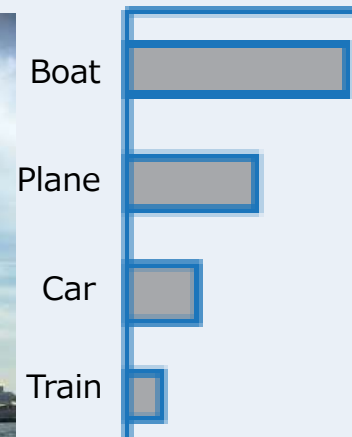
新しいデー  
タのための新



## システムへの統合



確率



## 予測とネットワークの 性能評価



テスト画像

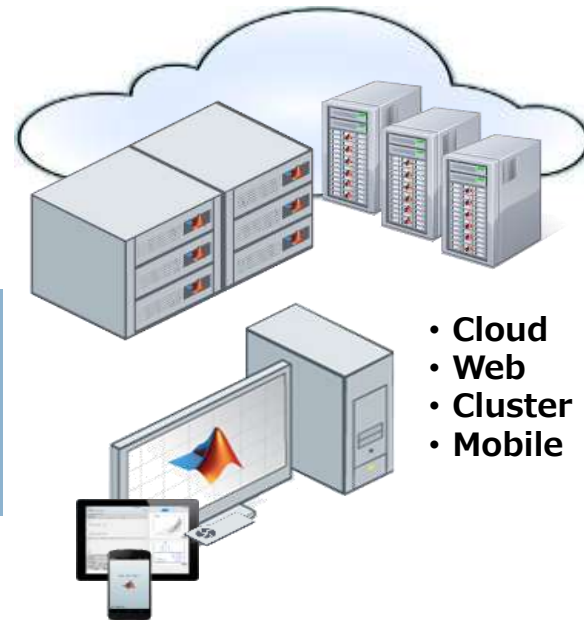
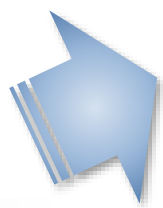
学習したネットワーク

# MATLABによるアプリケーション配布

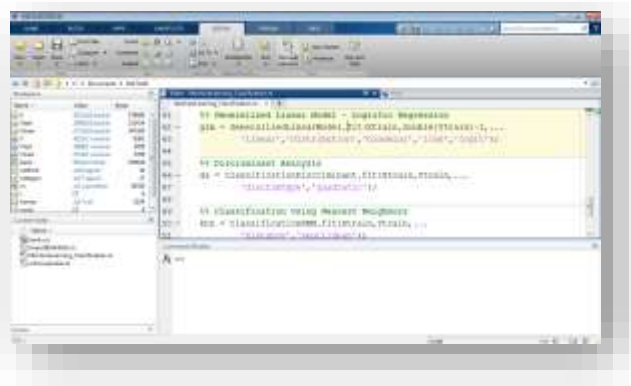


最適なアルゴリズムの探求

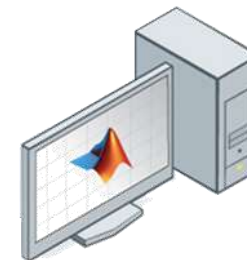
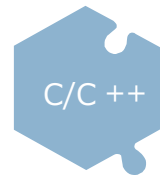
遠隔地での判断、  
スマホ、タブレットの利用



- Cloud
- Web
- Cluster
- Mobile



MATLABライセンスのない  
PCへの配布



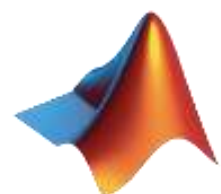
# GPU Coder™

New in **R2017b**

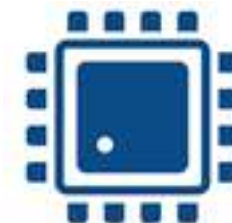
- プラグマによる関数解析とカーネル生成
  - CUDAの文法を知らなくても利用できる
- デザインパタンの利用も可能
  - より確実かつ効率の良いカーネル生成
- GPU Coder専用GUIを使ったコード生成
  - 初めてでも使いやすいGUI



製造ラインなど  
リアルタイム性が  
必要なケースに有効



MATLAB



NVIDIA GPU

MATLABコードからCUDA Cを生成します

# 転移学習ワークフロー

## 学習済みネットワーク読み込み

特定のタスクのために  
学習された分類器



画像数 100万枚以上  
クラス 1000以上

## 最後段の層を いくつか置き換え

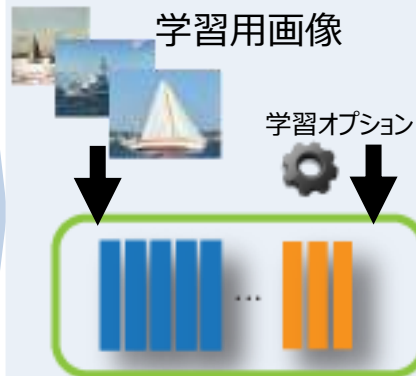
新しいデータを  
学習するための新しい層



## ネットワークの学習

学習用画像

学習オプション



画像数100枚  
クラス 数10カテゴリ

## 予測とネットワークの 性能評価

テスト画像



再学習したネットワーク

## システムへの統合



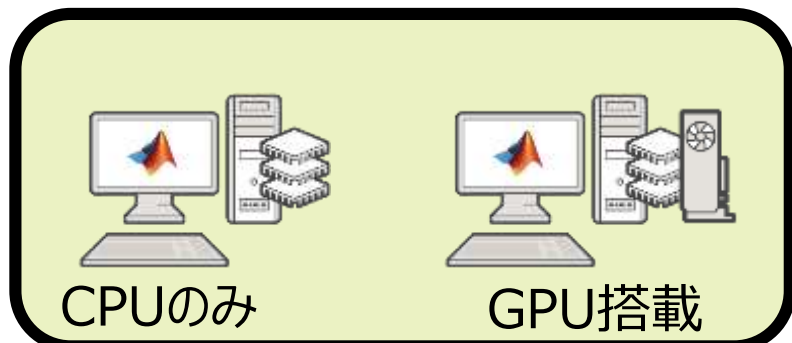
MATLABは学習済みネットワークからシステムへの統合までサポート  
画像セットがあればすぐに始められます。

# multi-GPU、クラウド利用による学習の高速化

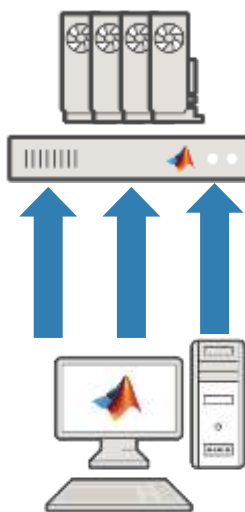
```
opts = trainingOptions('sgdm', ...  
    'MaxEpochs', 100, ...  
    'MiniBatchSize', 250, ...  
    'InitialLearnRate', 0.00005, ...
```

```
'ExecutionEnvironment', 'auto' );
```

GPUの有無を自動で認識、  
あればGPU、なければCPUで学習



複数GPU



サーバー/クラウド



More GPUs

# multi-GPU、クラウド利用による学習の高速化

```
opts = trainingOptions('sgdm', ...  
    'MaxEpochs', 100, ...  
    'MiniBatchSize', 250, ...  
    'InitialLearnRate', 0.00005, ...
```

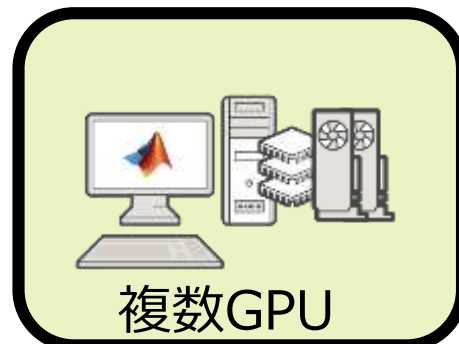
```
'ExecutionEnvironment', 'multi-gpu' );
```



CPUのみ



GPU搭載



複数GPU



サーバー/クラウド



More GPUs

# multi-GPU、クラウド利用による学習の高速化

```
opts = trainingOptions('sgdm', ...  
    'MaxEpochs', 100, ...  
    'MiniBatchSize', 250, ...  
    'InitialLearnRate', 0.00005, ...
```

```
'ExecutionEnvironment', 'parallel' );
```

•わずかな書き換えでスケールアップ&高速化が可能



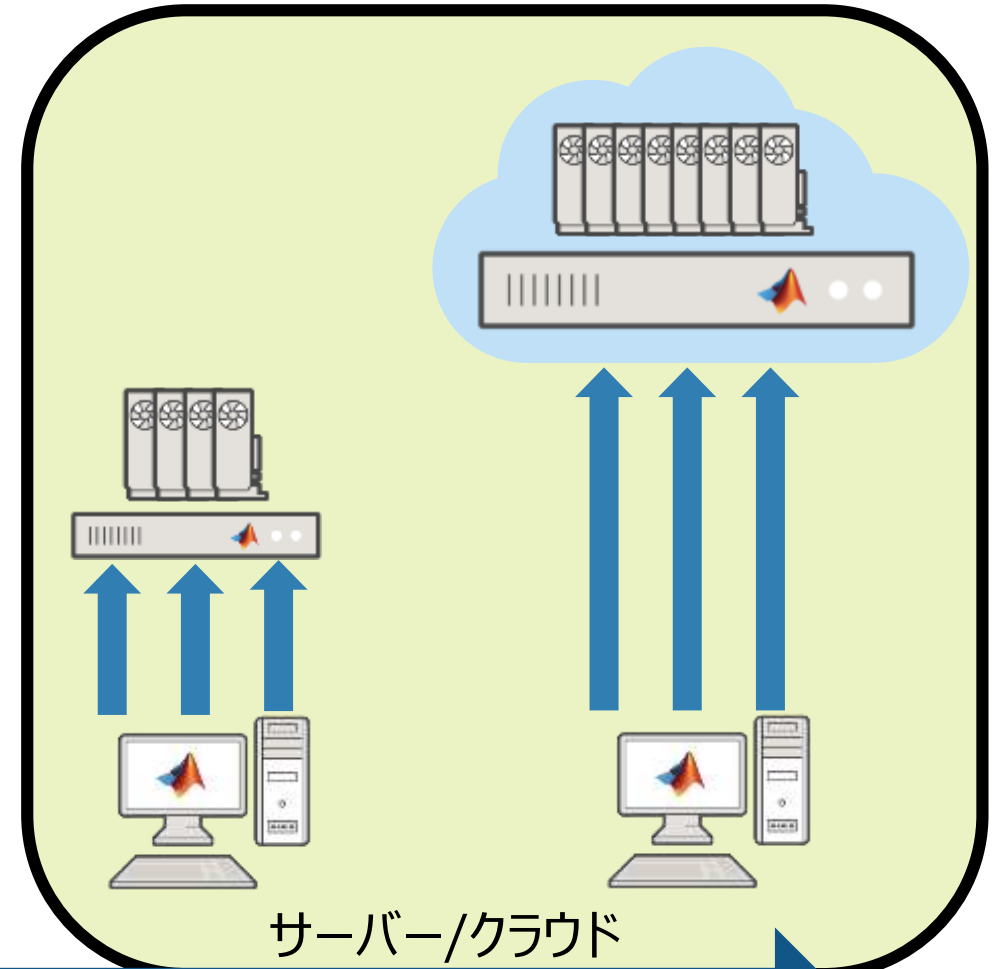
CPUのみ



GPU搭載



複数GPU



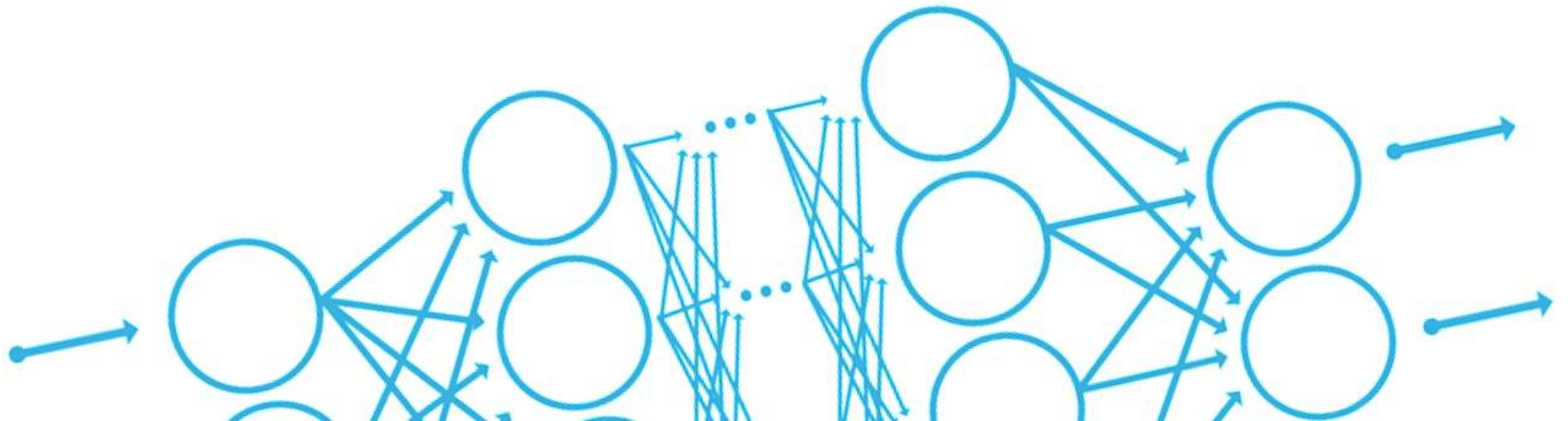
サーバー/クラウド

More GPUs



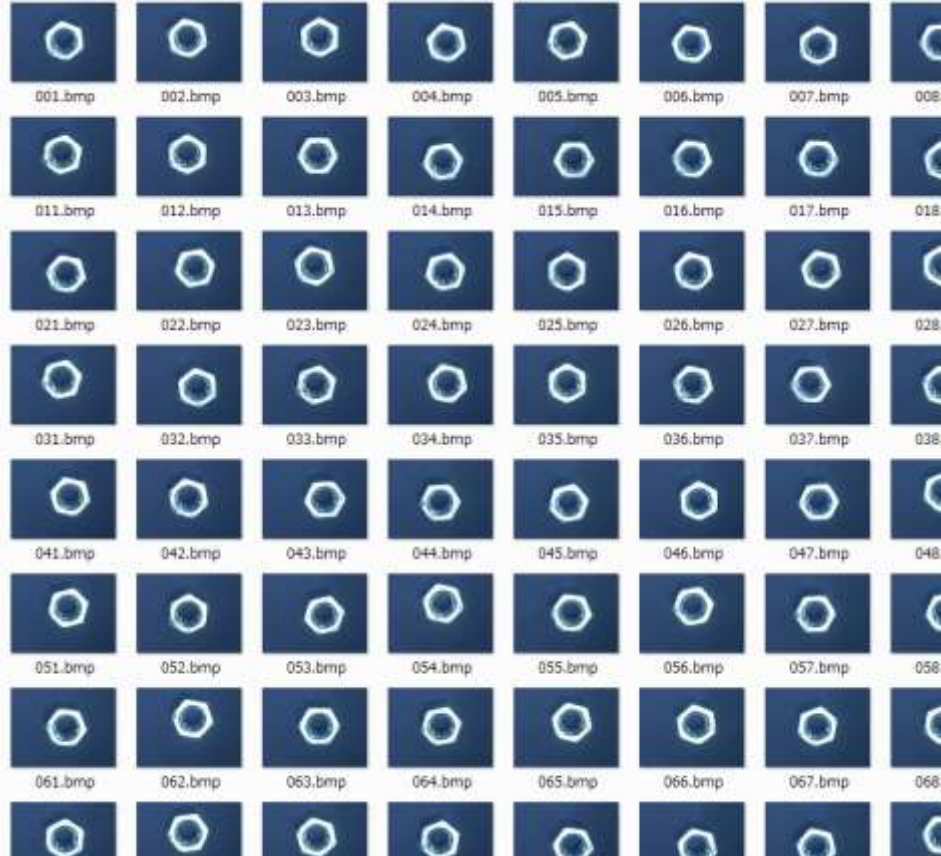
# Agenda

- これからはじめるディープラーニング活用ワークフロー
- ディープラーニングの異常検知への適用
- 学びを助けるリソース



## 【例題】特徴量を使った異常検出

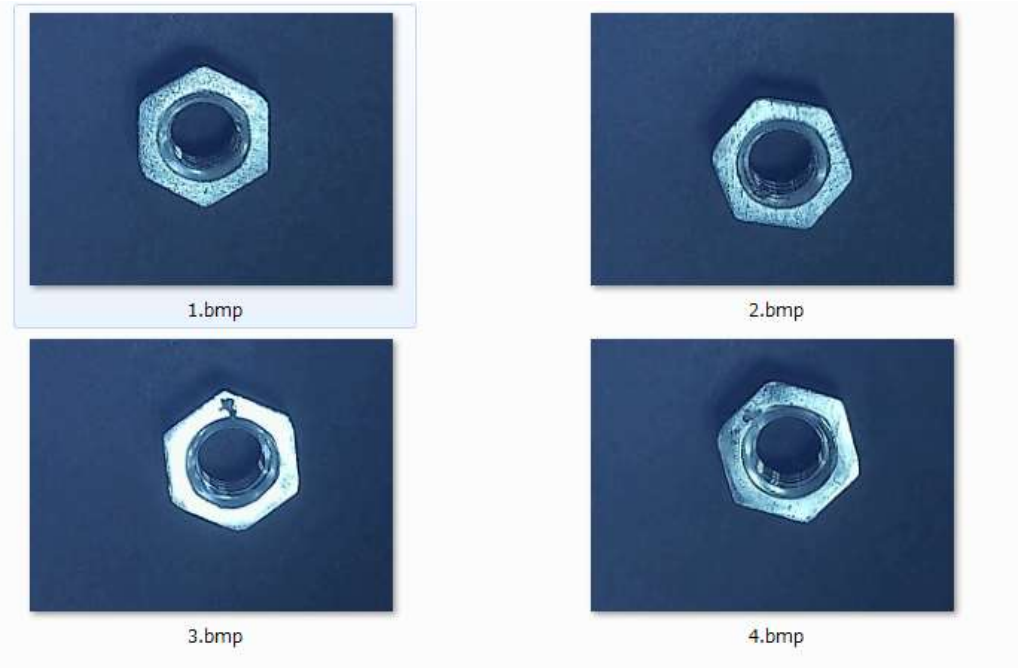
100個のナット



一般的に  
正常なモノに対して異常なモノが少ない

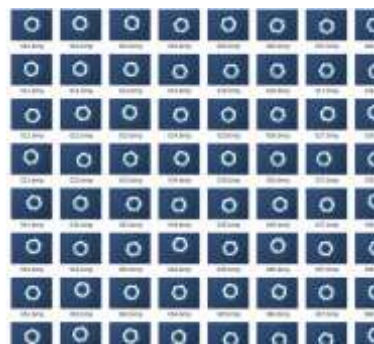
デモ

異常



100個の中に紛れ込んでいる4個の異常を見つけられるか

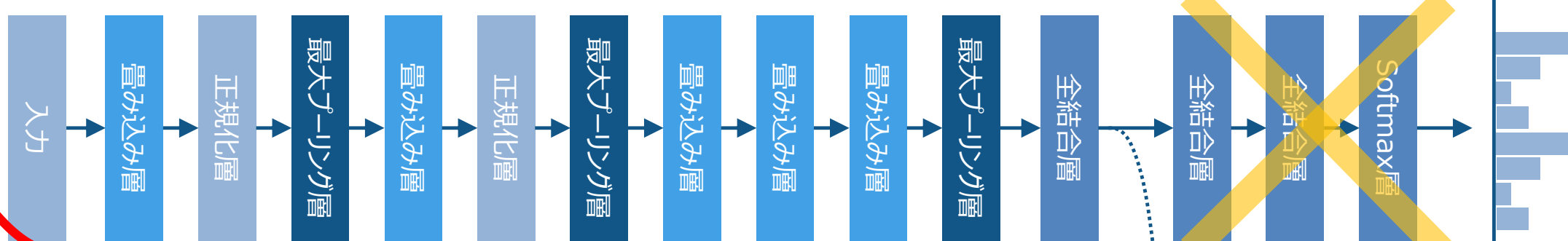
# 転移学習でナットを見つける分類器を学習



正常/異常のラベル  
はつけない

ImageNet  
1000個のカテゴリ

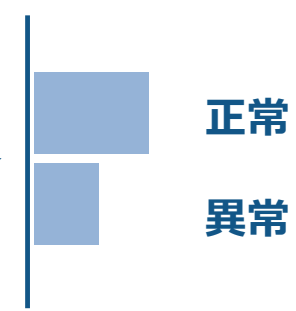
学習済みのAlexnetを特徴抽出器として利用



activations

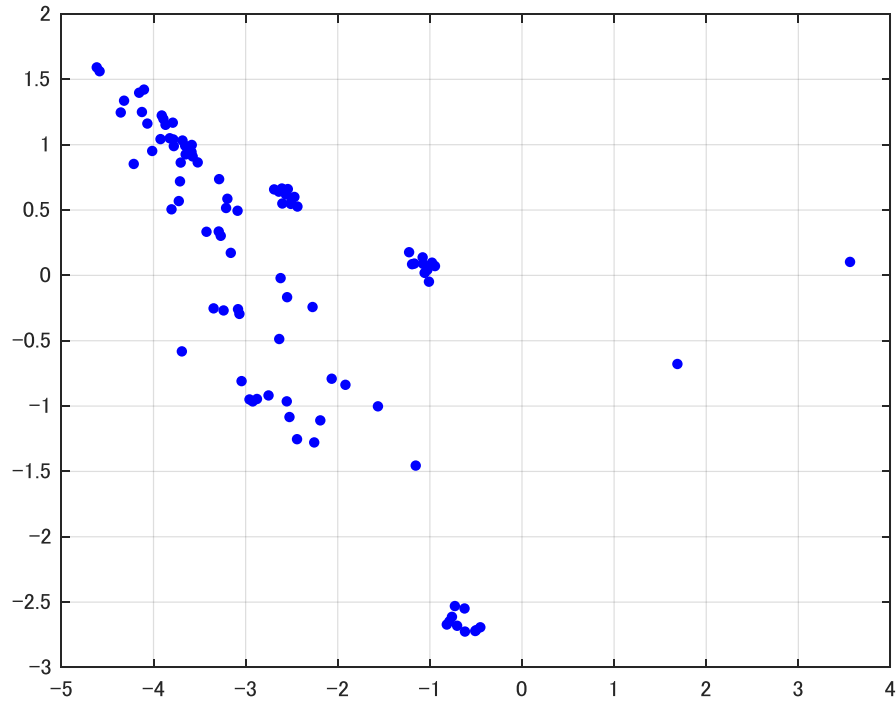
4096次元の  
特徴量

分類器の学習 :  
fitcsvm

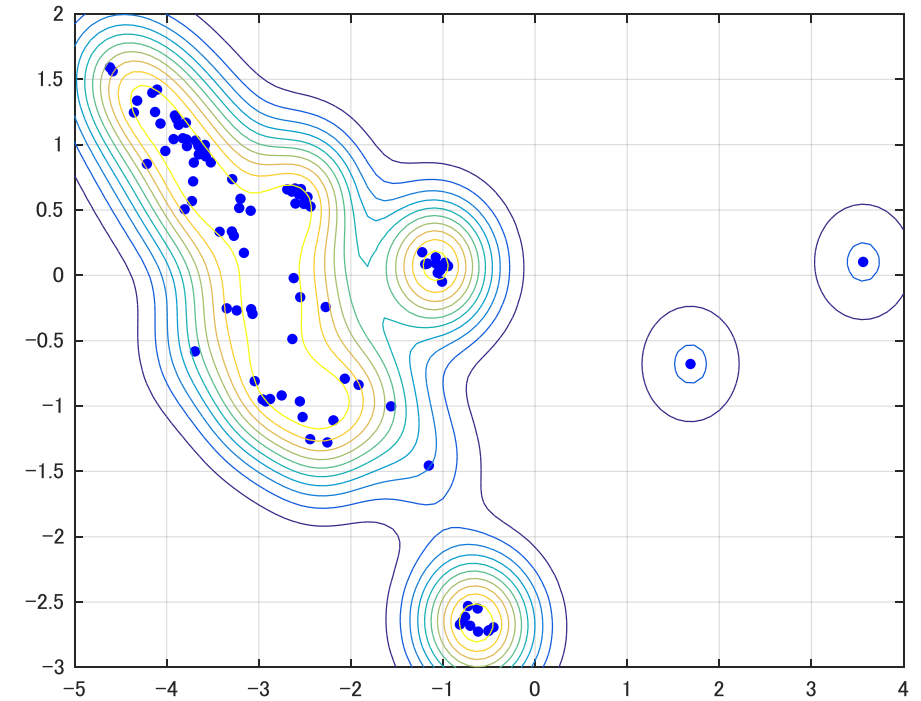


# 1-Class SVM とは？

カーネル法を使った正常度スコアの推定アルゴリズム



異常データを含むデータの例



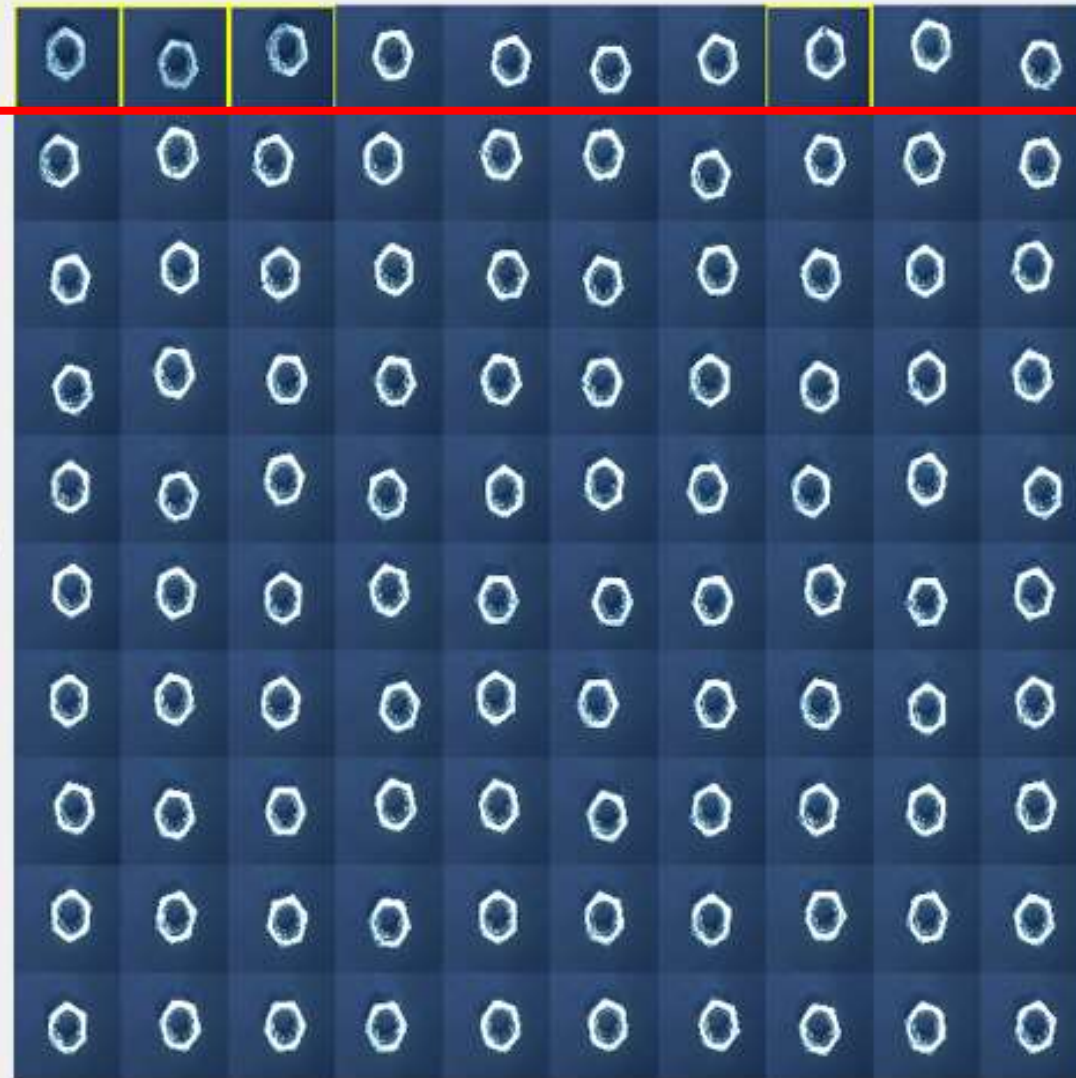
スコアの値が高い方が正常度が高い



1-Class SVM

## 【例題】特徴量を使った異常検出

異常度：高

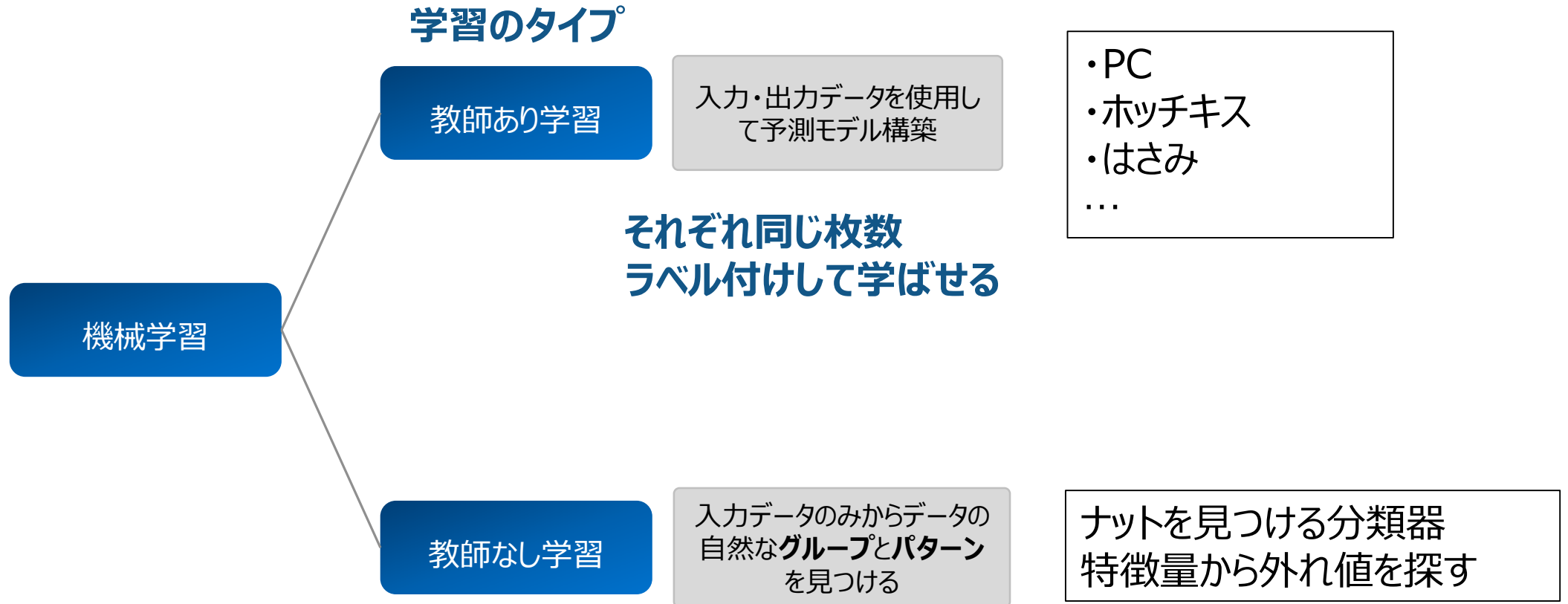


ディープラーニングの判定  
で異常度が高いものだけ  
を人が判断

⇒人が行う検査が1割  
大幅な工数削減。

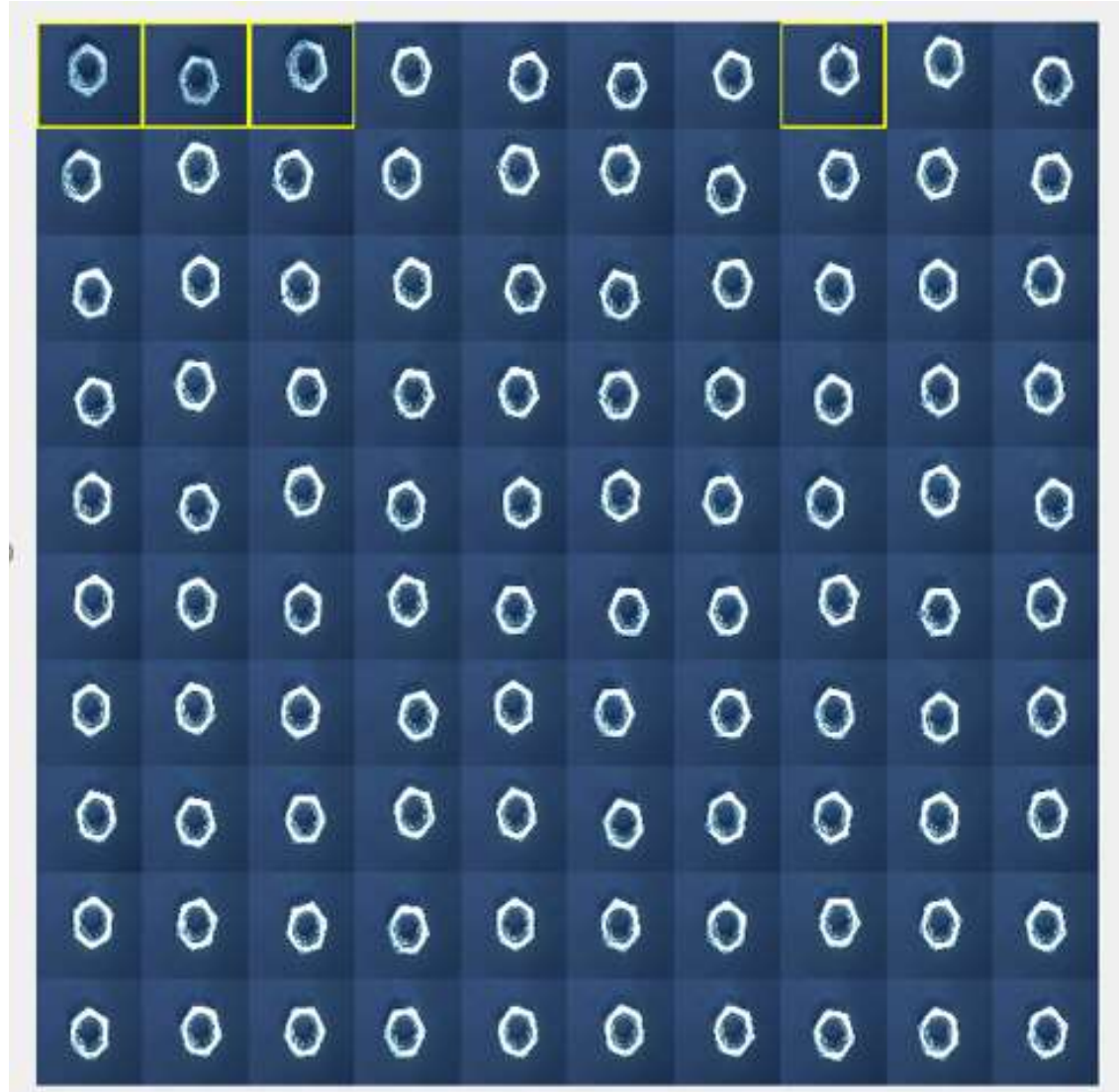
異常度：低

# 取り扱った二つの学習タイプ



# やはりもっと精度をあげたい

異常度：高



異常度：低

## 精度を上げるためには何を変更しますか？

- 今ある画像データを回転等で拡張する
- 学習枚数を増やす
- 分類器(SVM)のパラメータを変える
- 特徴抽出のネットワークを変える(vgg, GoogleNet...)
- スクラッチでネットワークを作る



## 正常(高スコア) 画像の加工と評価

行列を反転させる関数 :  $B = \text{fliplr}(A)$

正常元画像



9.30 (1位)

3.53 (75位)

左右反転 ⇒ スコア悪化

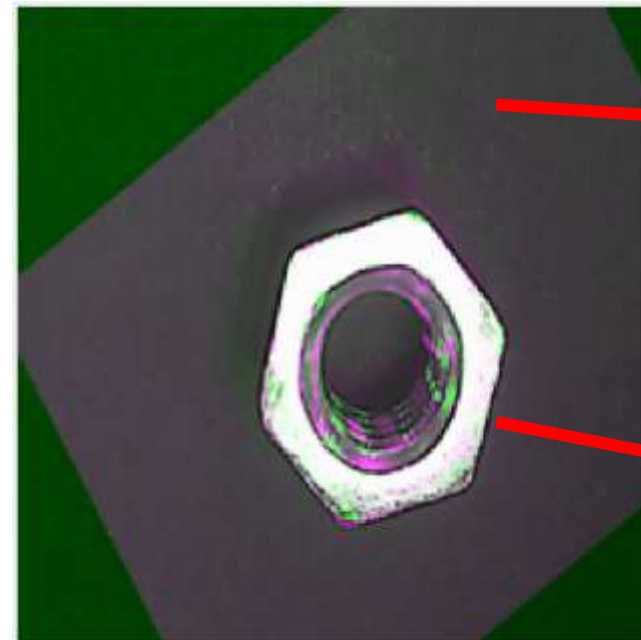
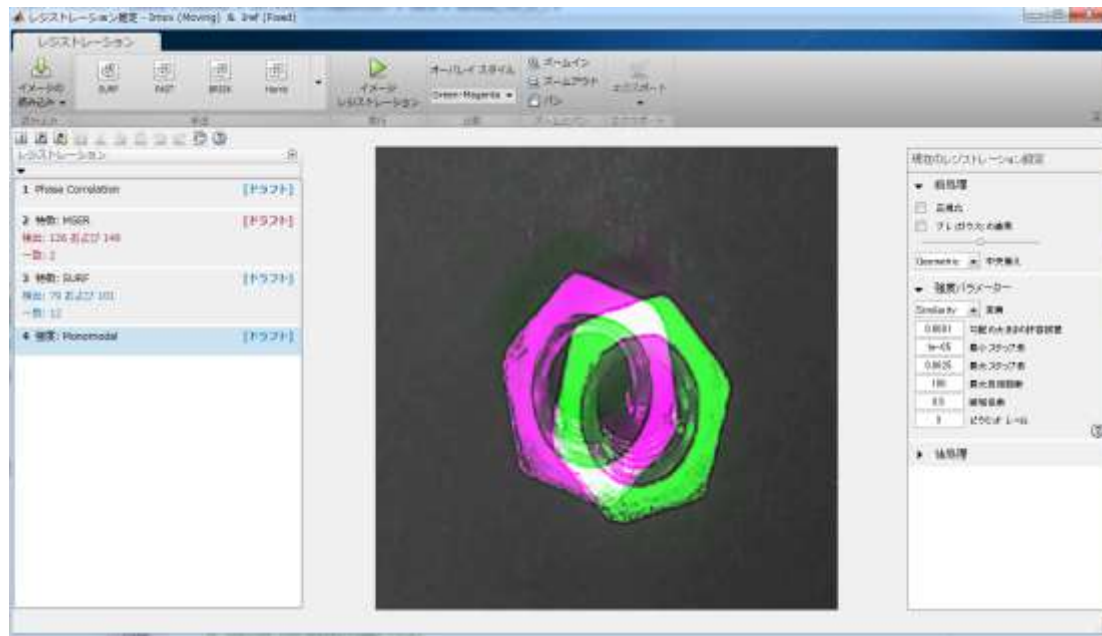
位置関係や輝度グラデーションなどもスコアに大きく影響がありそう

## 解析結果を元に前処理をおこなう

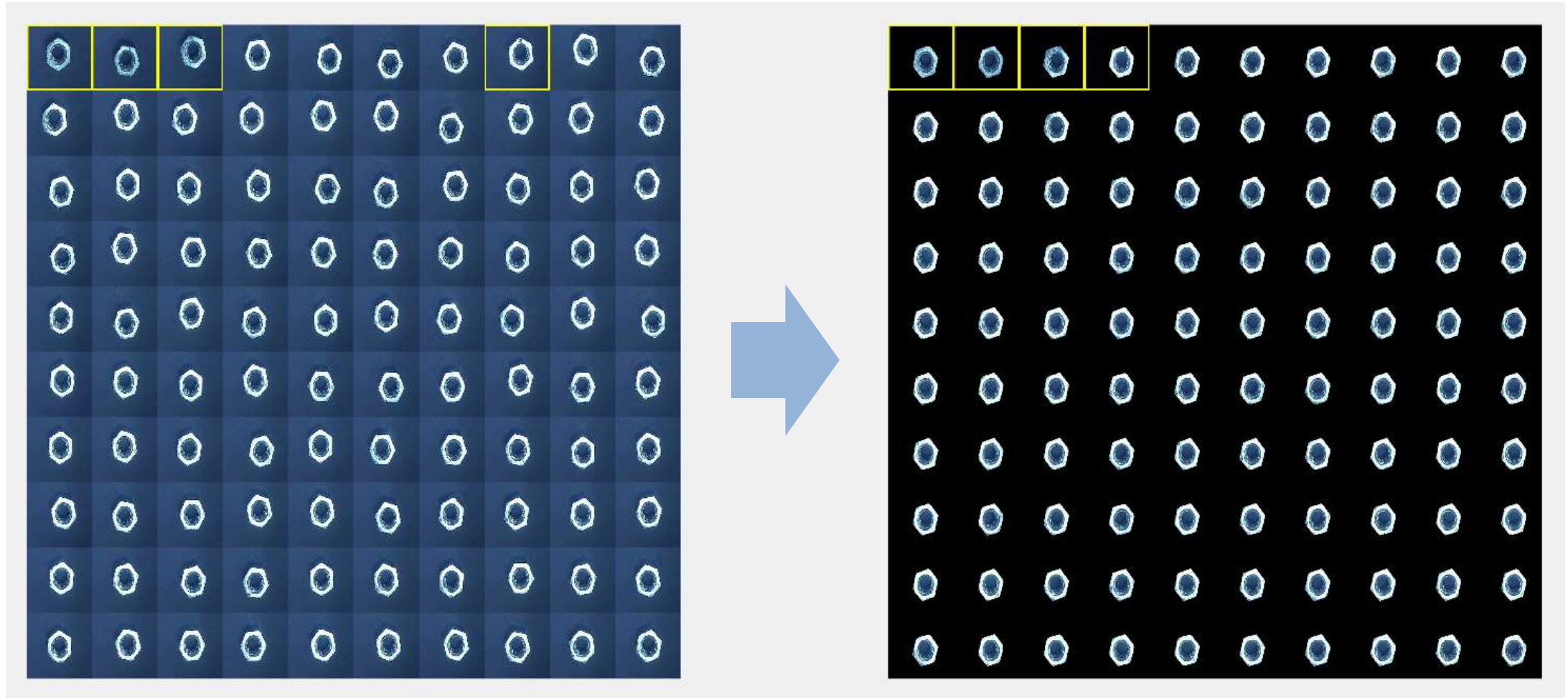
表面のキズに注目したい。

解析からナットの位置や背景の輝度分布などもスコアに影響している。

⇒輝度のレジストレーションと幾何学変換により、位置合わせ。  
ナット以外のところはマスク。

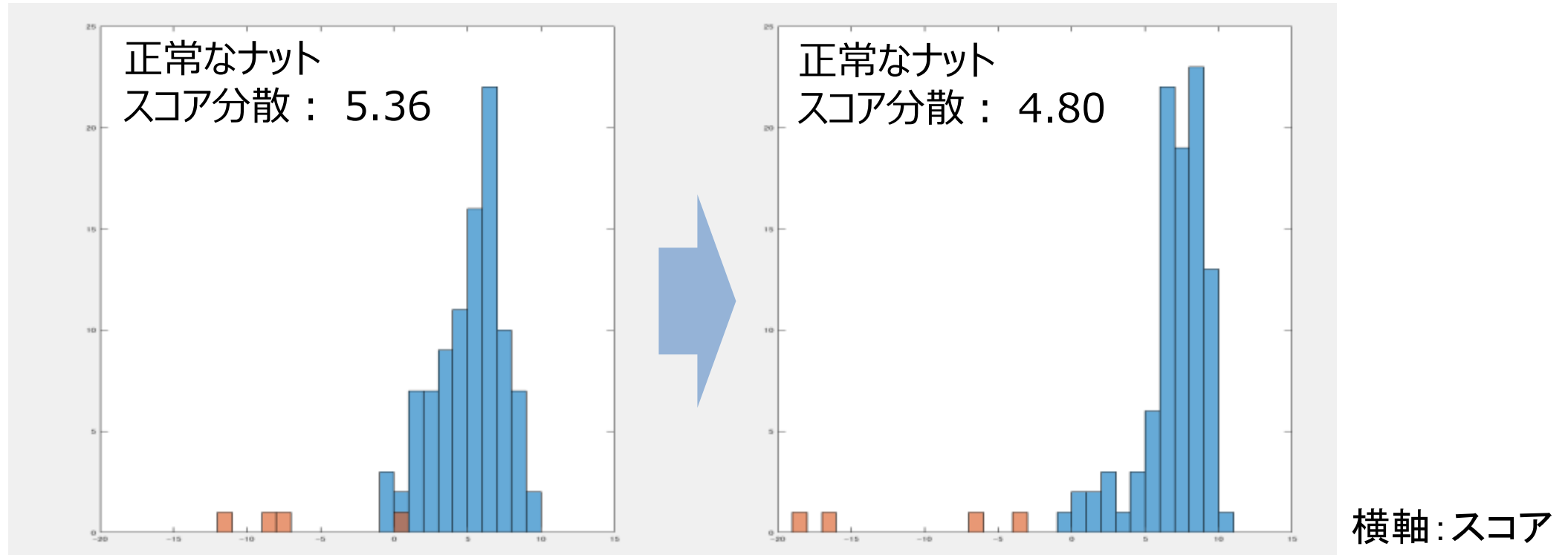


# 画像前処理と結果



# 画像前処理によるスコア分布の変化

縦軸：頻度



- 正常/異常のラインが明確に
- 正常なナットのスコア分散が減少

効率的な精度向上には多角的な解析とアプローチが必要

# まとめ

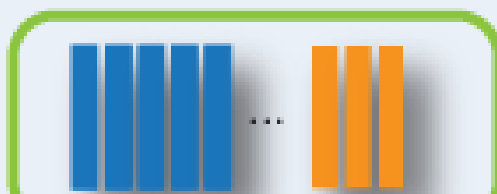
## 画像の準備をサポート

- ・カメラ&画像への効率的なアクセス
- ・ラベリング&画像拡張機能

### 画像の準備



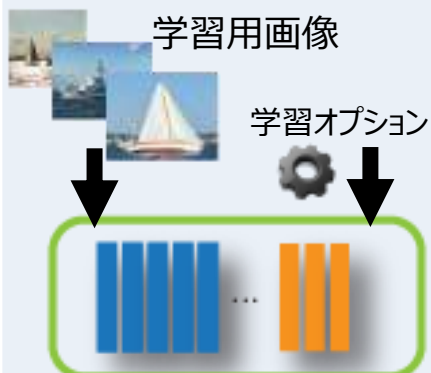
### ネットワークの準備



ディープラーニング分野でMATLABを利用するメリット

- ・画像があれば簡単にはじめられるフレームワーク
- ・作ったネットワークをすぐにシステムに統合できる

### 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



## 転移学習を強力にサポート

- ・多数の学習済みモデルに対応
- ・今後も拡張予定

## 作業効率・精度を高める機能をサポート

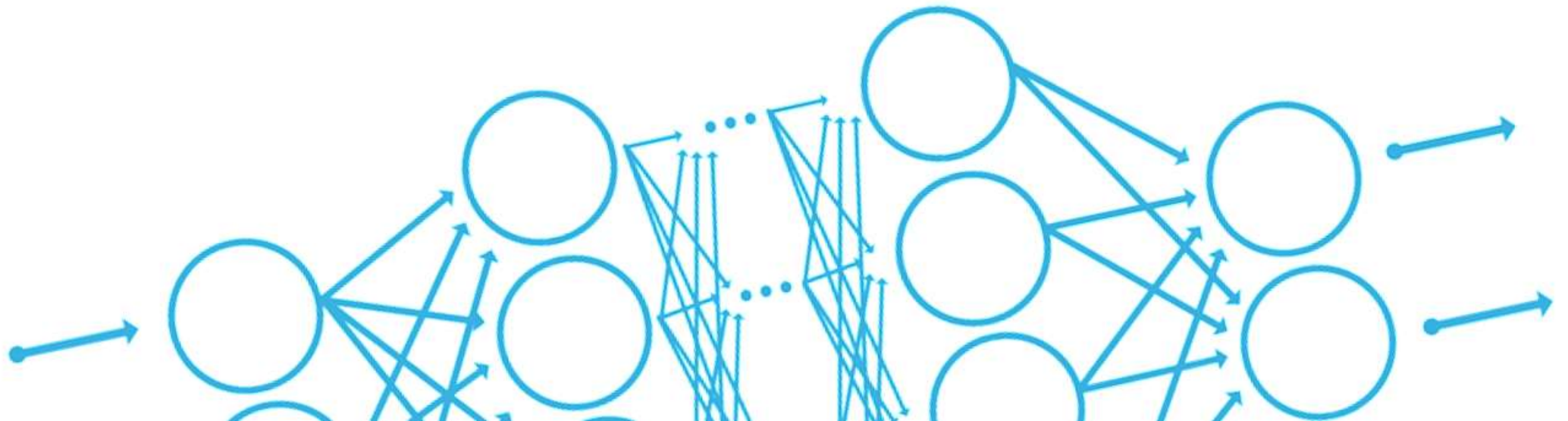
- ・豊富な前処理用関数群
- ・ベイズ最適化
- ・学習過程・ネットワーク・結果の可視化
- ・複数GPU/クラウドによる学習高速化

## すばやい実用化をサポート

- ・PC
  - ・組み込みGPU
  - ・クラウド
- 上への統合が可能

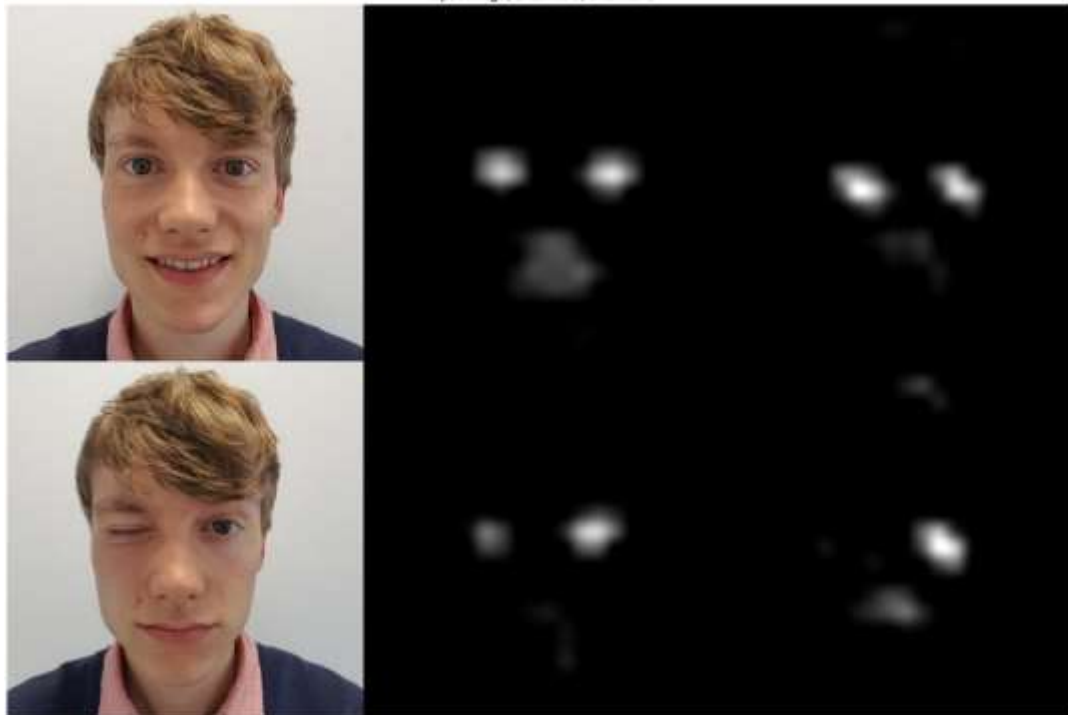
# Agenda

- これからはじめるディープラーニング活用ワークフロー
- ディープラーニングの異常検知への適用
- **学びを助けるリソース**



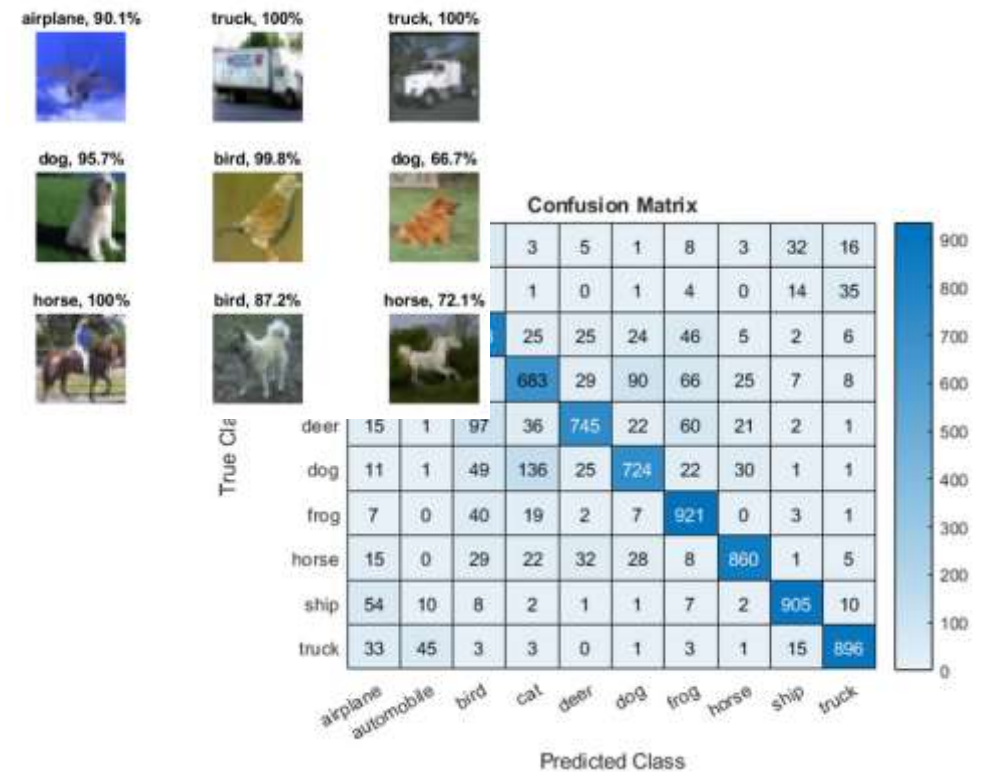
# 精度向上を手助けするサンプルプログラム

たたみ込みニューラル ネットワークの活性化の可視化



<https://www.mathworks.com/help/releases/R2017b/nnet/examples/visualize-activations-of-a-convolutional-neural-network.html>

ベイズ最適化を用いたディープラーニング



<https://www.mathworks.com/help/releases/R2017b/nnet/examples/deep-learning-using-bayesian-optimization.html>

# 実践的なディープラーニング物体認識を自分のペースで

MATLAB Academy

<https://matlabacademy.mathworks.com/jp>

## あなたの自己学習形式コース

The screenshot shows two course cards. Each card has a blue '開始' (Start) button on the left. The first card is for 'MATLAB 入門 (0%)' with a gear icon and '無制限のアクセス' (Unlimited access) below it. The second card is for 'ディープラーニング入門 (0%)' with a gear icon and '無制限のアクセス' (Unlimited access) below it. Progress bars are shown below the course titles, all at 0%.

WEBブラウザ上で  
MATLABディープラーニングの操作体験

無料！！

ディープラーニングによる画像認識を  
2時間で体感いただけます。





© 2017 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.