

# セマンティック・セグメンテーションの基礎

MathWorks Japan

アプリケーション エンジニアリング部 テクニカルコンピューティング

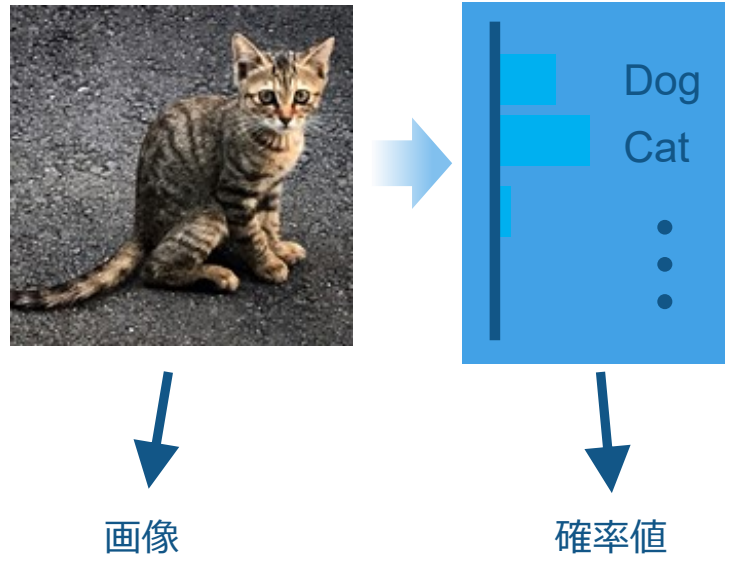
太田 英司

# 画像分野におけるディープラーニングの展開

R2017a  
R2017b

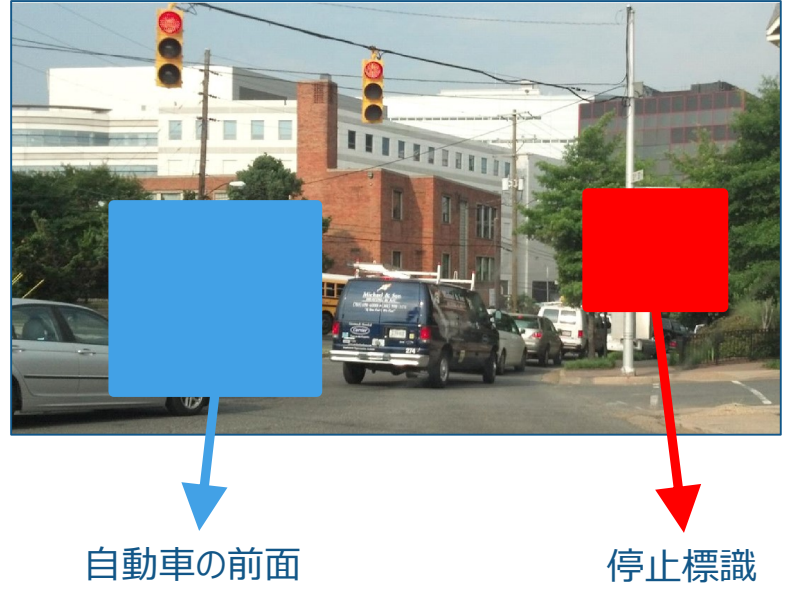
## 物体認識 (画像全体)

CNN (Convolutional Neural Network)



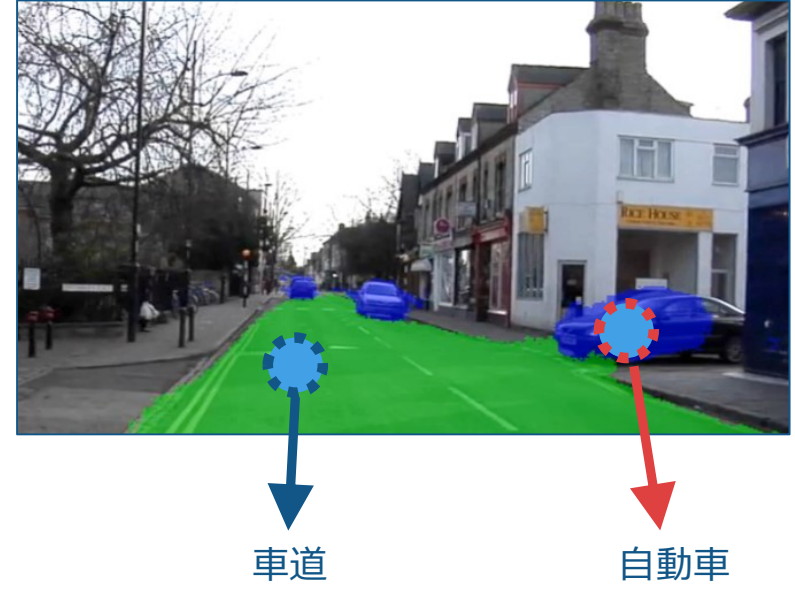
## 物体の検出と認識

R-CNN / Fast R-CNN / Faster R-CNN



## 物体認識 (ピクセル単位)

FCN / SegNet



# Semantic Segmentation

畳み込みニューラルネットによるセグメンテーション

R2017b

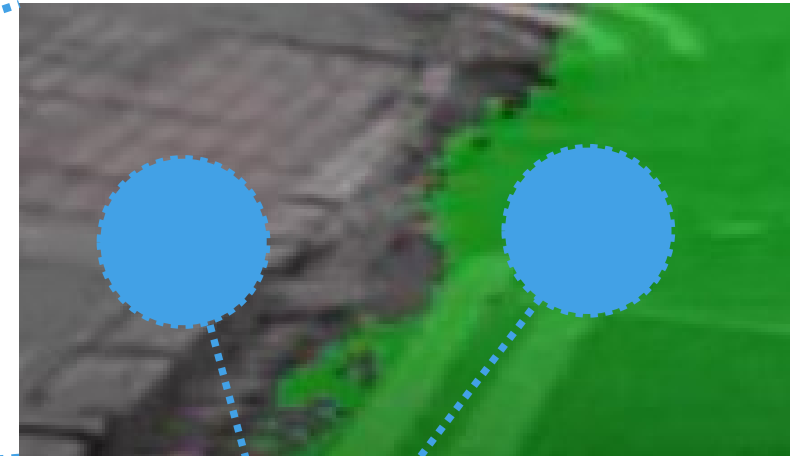
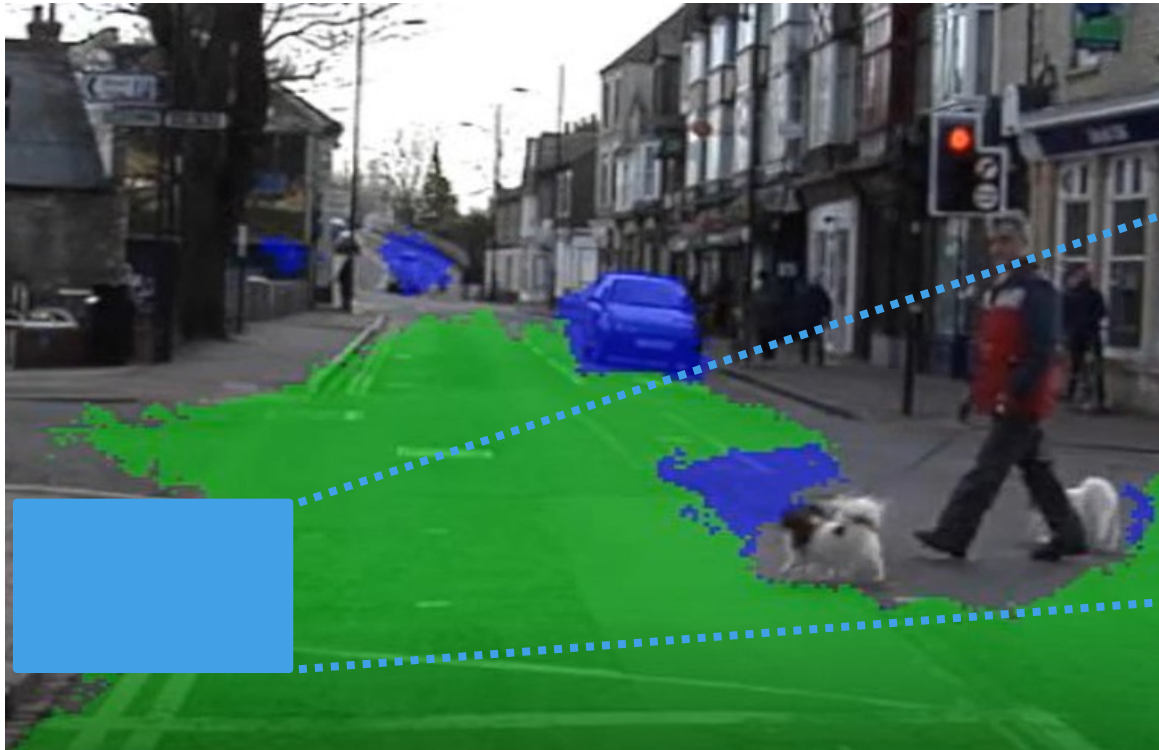




# Semantic Segmentation とは？

R2017b

各ピクセルをその意味（周辺のピクセルの情報）に基づいて、カテゴリ分類する手法

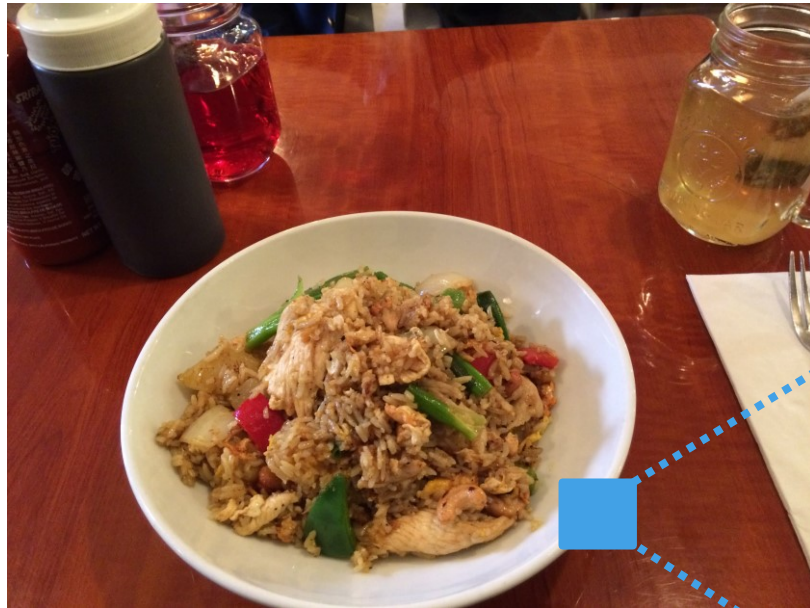


ちゃんと歩道と車道を区別できている！  
色だけを見ているわけではない

# Semantic Segmentation とは？

R2017b

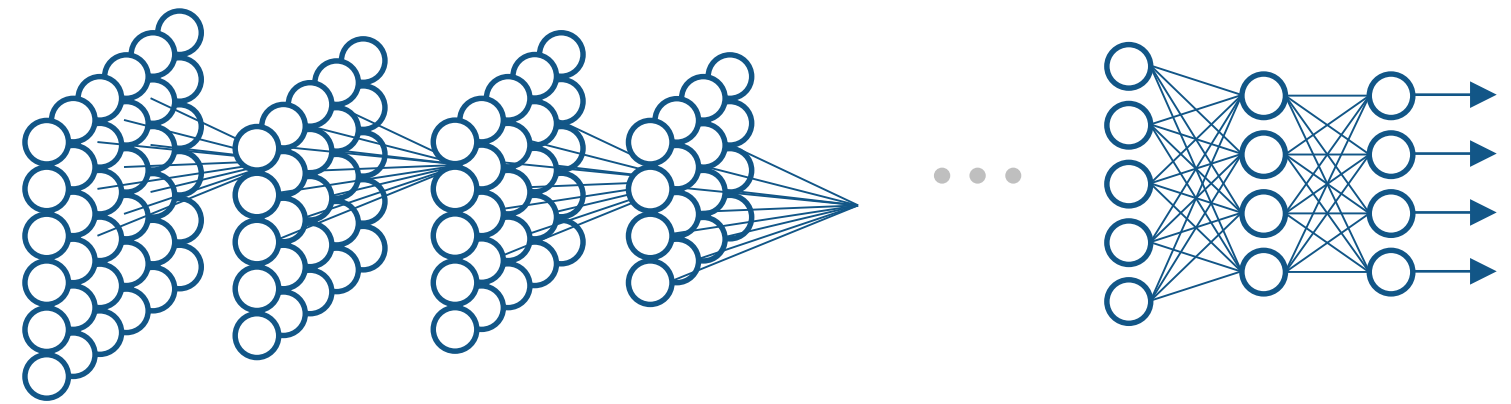
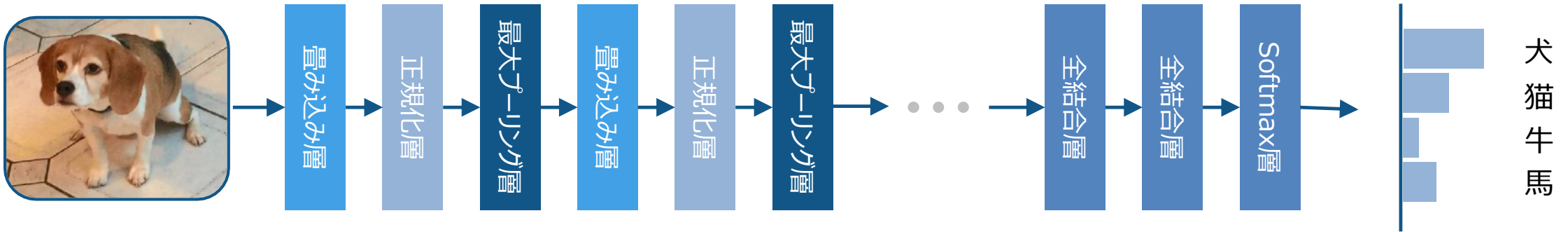
各ピクセルをその意味（周辺のピクセルの情報）に基づいて、カテゴリ分類する手法



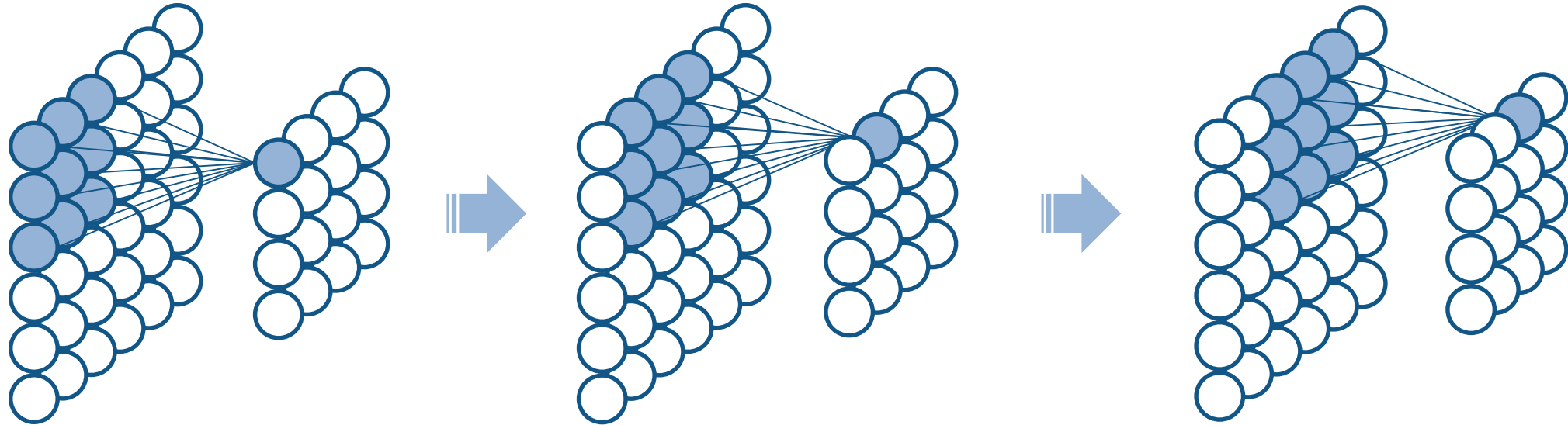
皿	皿	皿	皿	皿	皿	机	机	机
皿	皿	皿	皿	皿	皿	机	机	机
皿	皿	皿	皿	皿	机	机	机	机
皿	皿	皿	皿	皿	机	机	机	机
皿	皿	皿	皿	皿	机	机	机	机
皿	皿	皿	皿	机	机	机	机	机
皿	皿	皿	皿	机	机	机	机	机

畳み込みニューラルネットとは？

# 畳み込みニューラルネットとは？



# Convolution Layer (畳み込み層) / Pooling Layers (プーリング層)



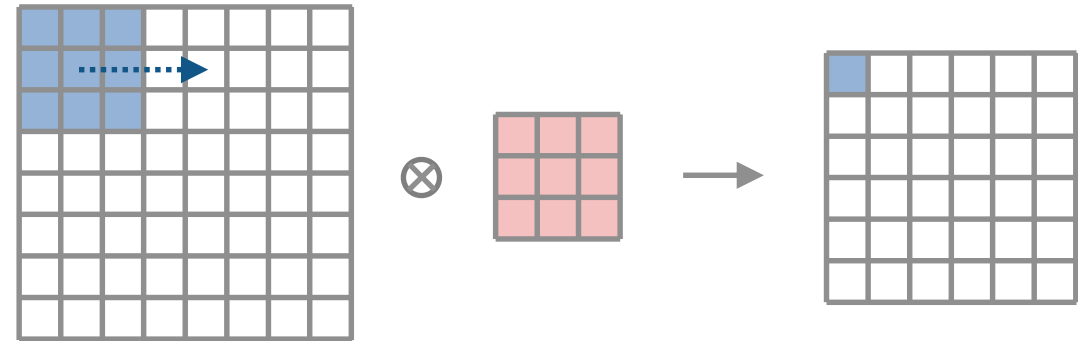
層と層の間を一部のみ連結して、ウェイトを共有すると、ニューラルネットで畳み込みが表現できる



# Convolution Layer (畳み込み層) / Pooling Layer (プーリング層)

## Convolution Layer (畳み込み層)

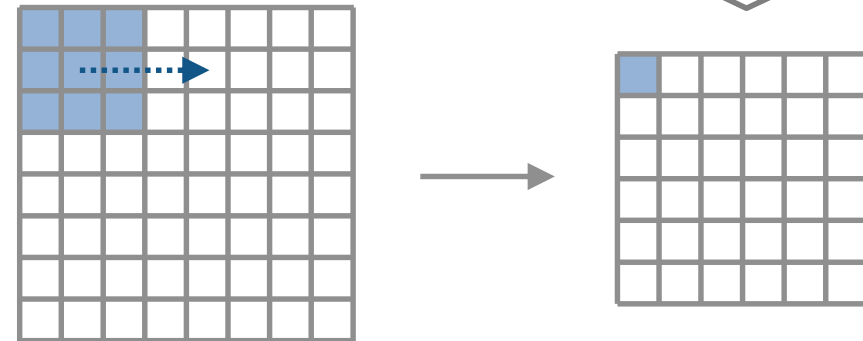
- 画像のフィルタ処理に相当する処理
- 特徴抽出器としての役割
- スライドと呼ばれる間引きを行うこともある



最大値を出力する場合 : Max Pooling  
 平均値を出力する場合 : Average Pooling

## Pooling Layer (プーリング層)

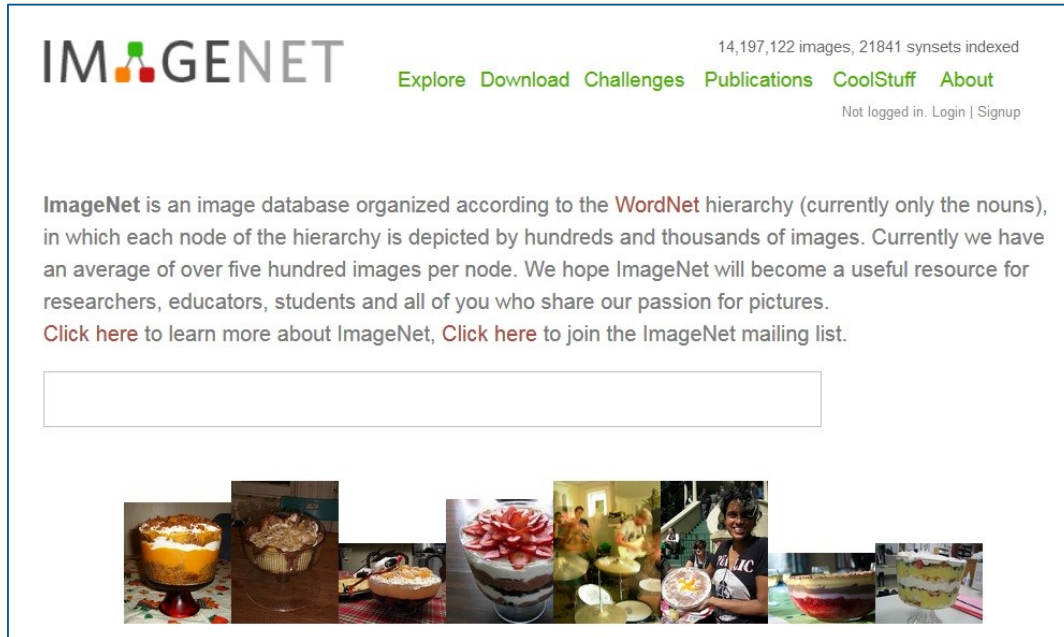
- 領域内の最大値または平均値を出力
- 平行移動等に対するロバスト性に関係
- スライドと呼ばれる間引きを行うこともある



# 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



14,197,122 images, 21841 synsets indexed

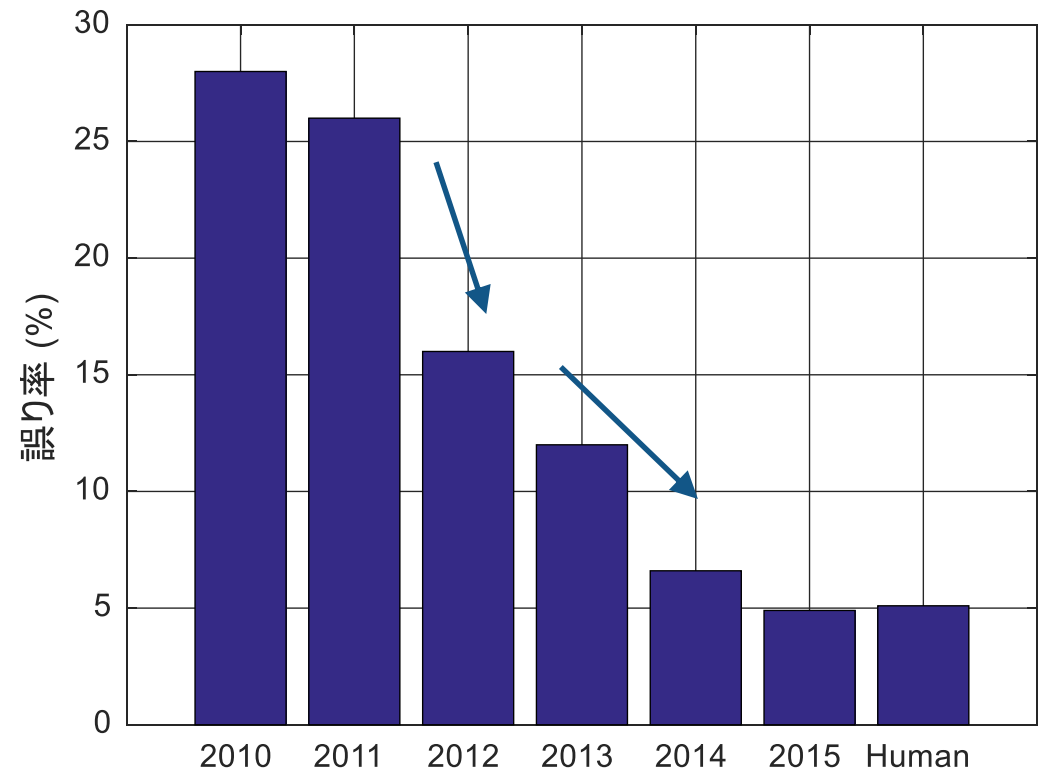
Explore Download Challenges Publications CoolStuff About

Not logged in. Login | Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

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



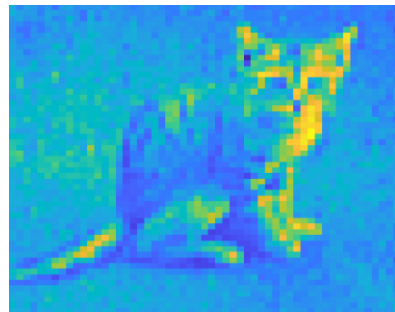
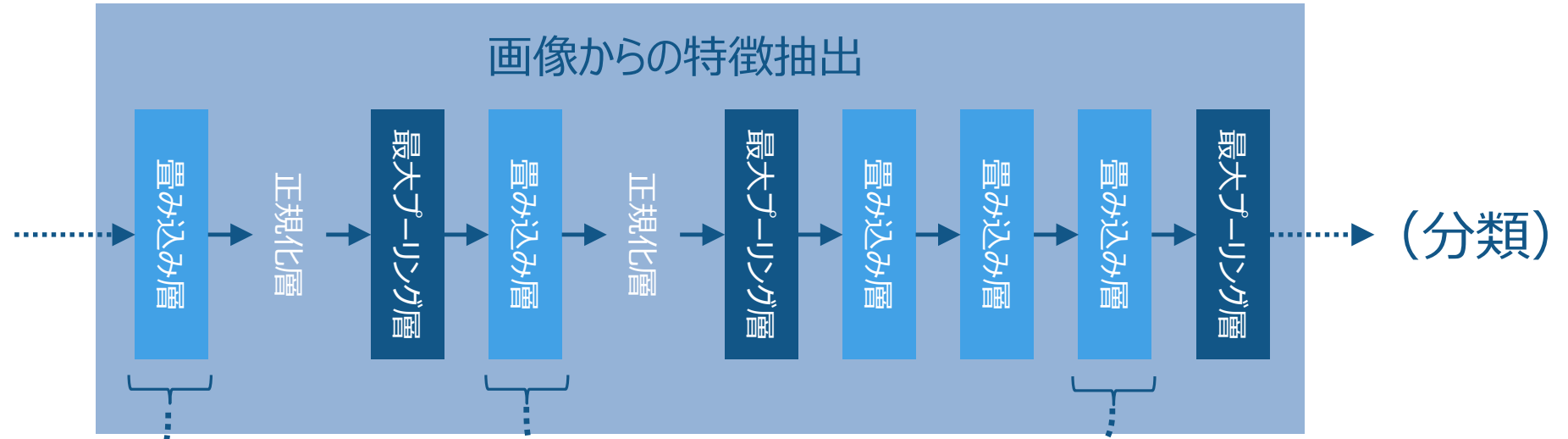
**CNNの登場によって10%以上の性能向上(2012)  
GoogLeNet, VGG等の深いCNNが登場(2014)**

# 畳み込みニューラルネットとは？

畳み込みにより、エッジ等の空間的な情報を取り出すと同時に、ダウンサンプリングを行っている

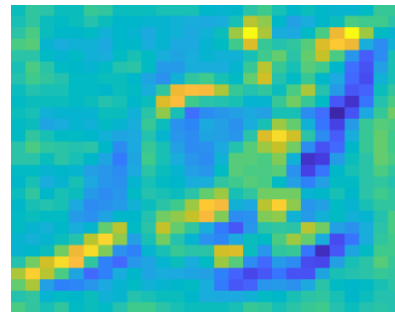


227 x 227

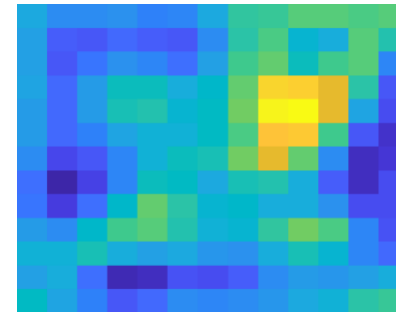


55 x 55

ピクセル単位の情報からエッジなどの空間情報を取り出している



27 x 27

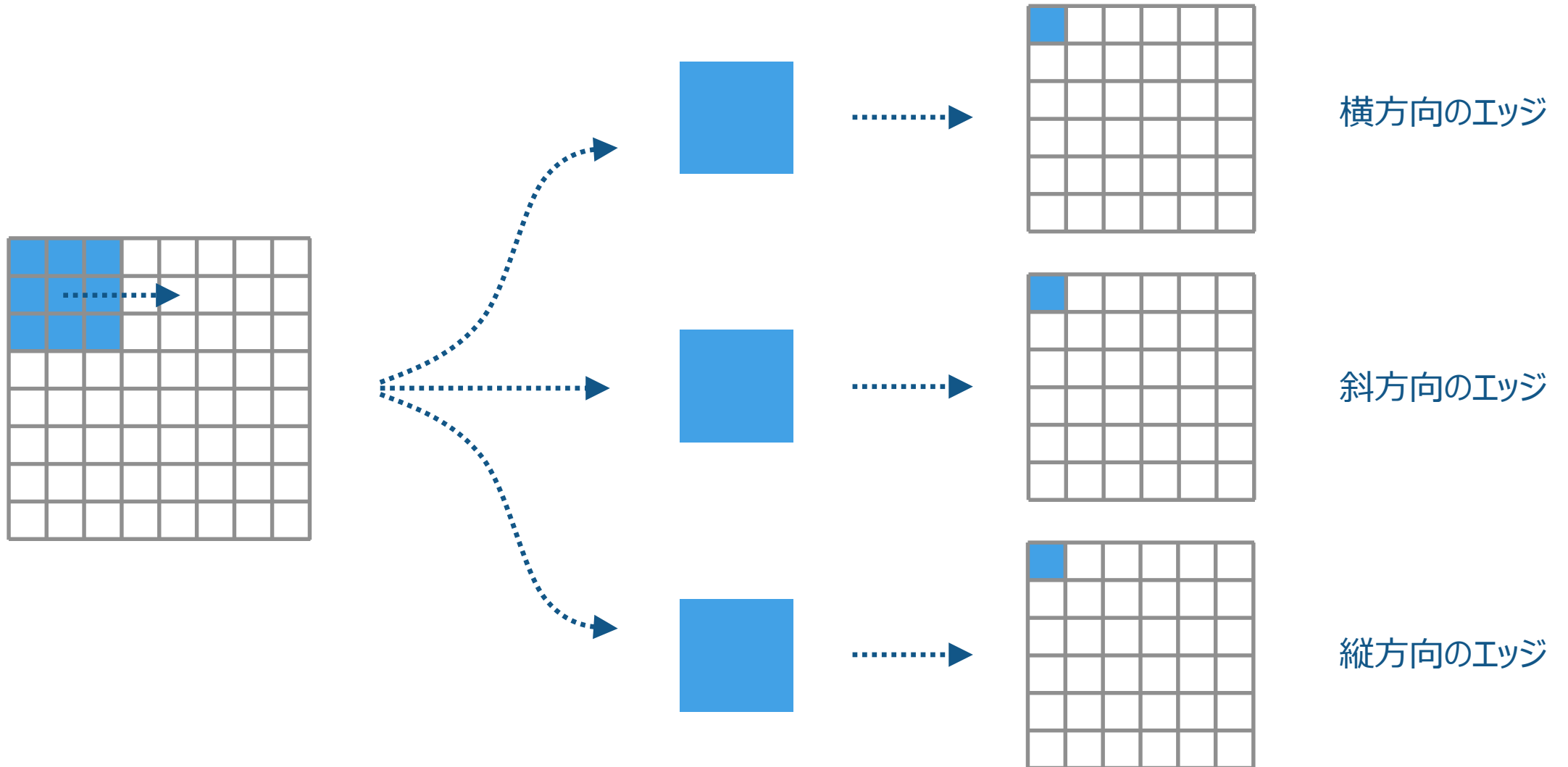


13 x 13

ダウンサンプルの反復により位置に関する情報が粗くなってしまう

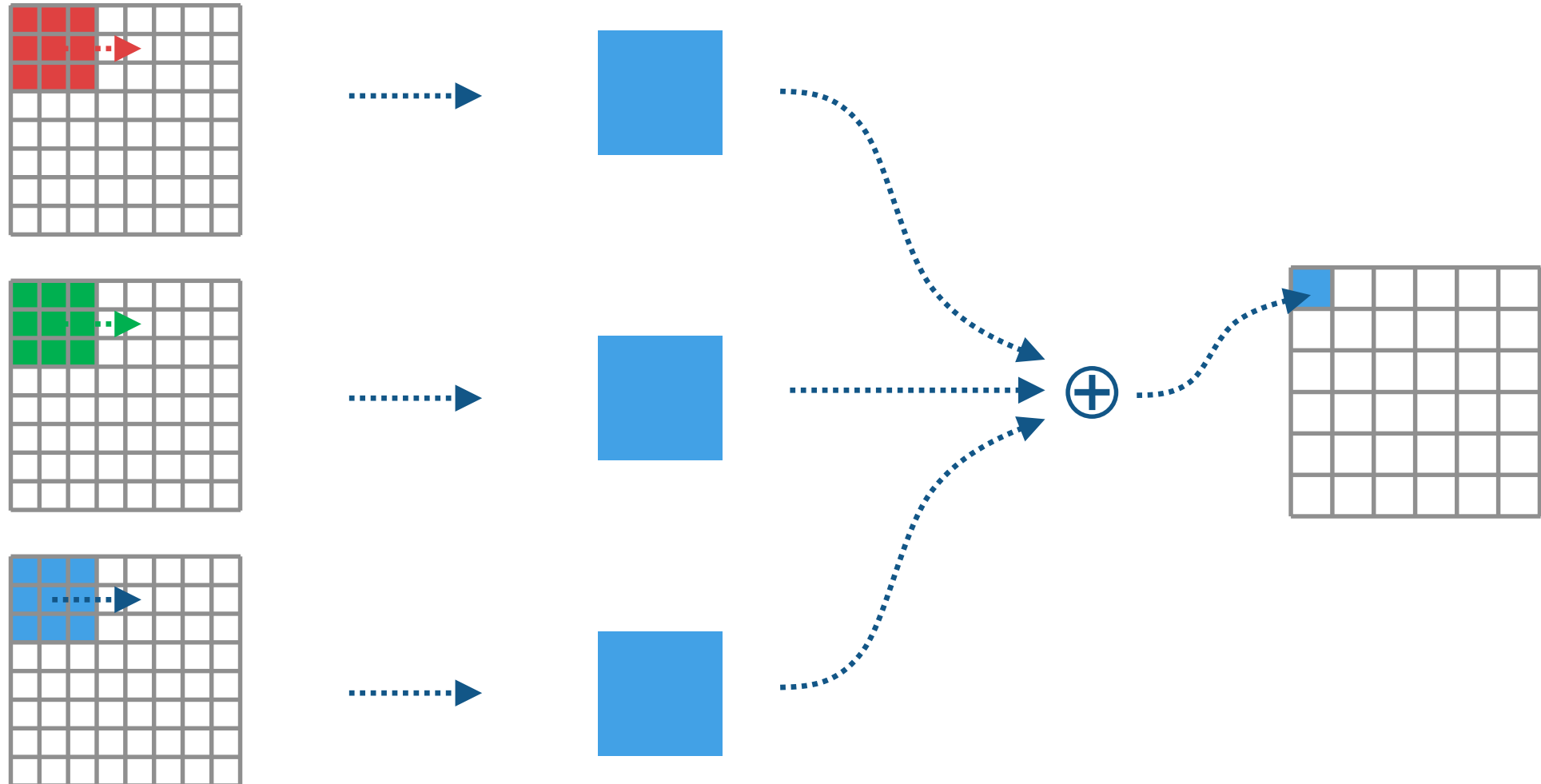
# Convolution Layer (畳み込み層)

複数のフィルタにより、画像の空間方向のさまざまなパターンを抽出することができる



# Convolution Layer (畳み込み層)

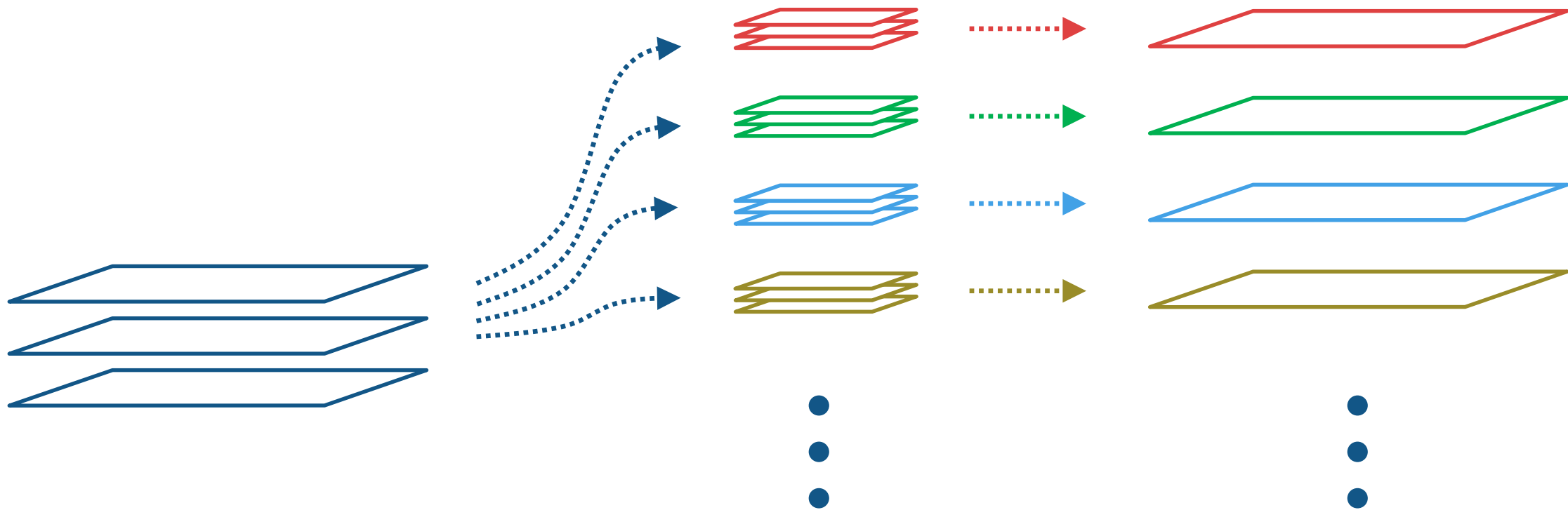
空間方向だけでなく、チャンネル方向のパターンも抽出することができる





# Convolution Layer (畳み込み層)

AlexNet の 1 段目の畳み込みでは 3 ch の特徴マップから 96 ch の特徴マップを生成している

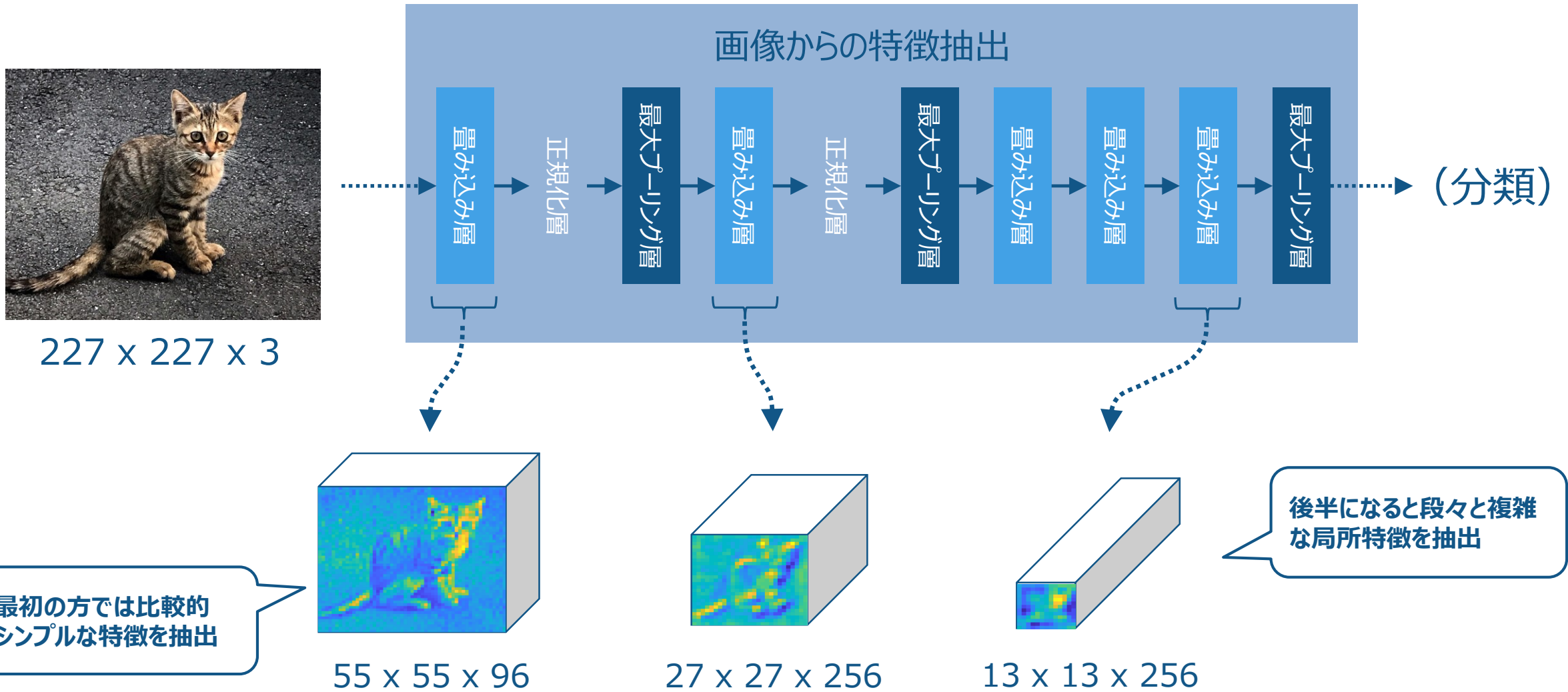


入力特徴マップ  
(チャンネル数: 3)

出力特徴マップ  
(チャンネル数: 96)

# 畳み込みニューラルネットとは？

空間方向の情報が削減されるのと同時にチャンネル数が増え、複雑な特徴を保持できるようになっていく

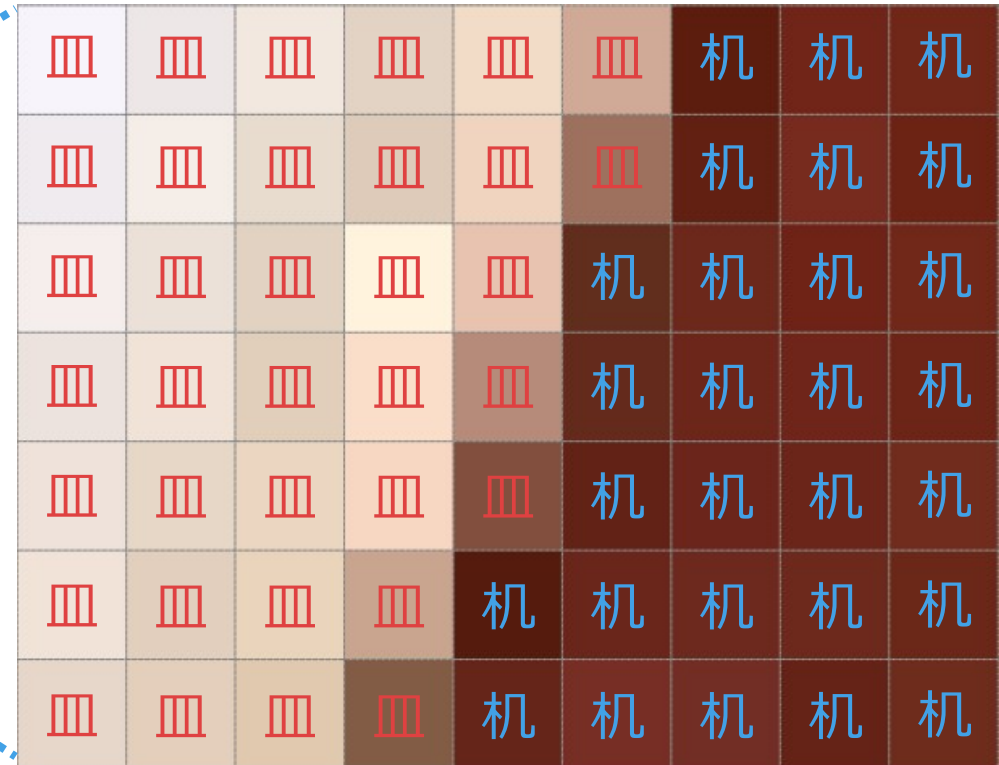
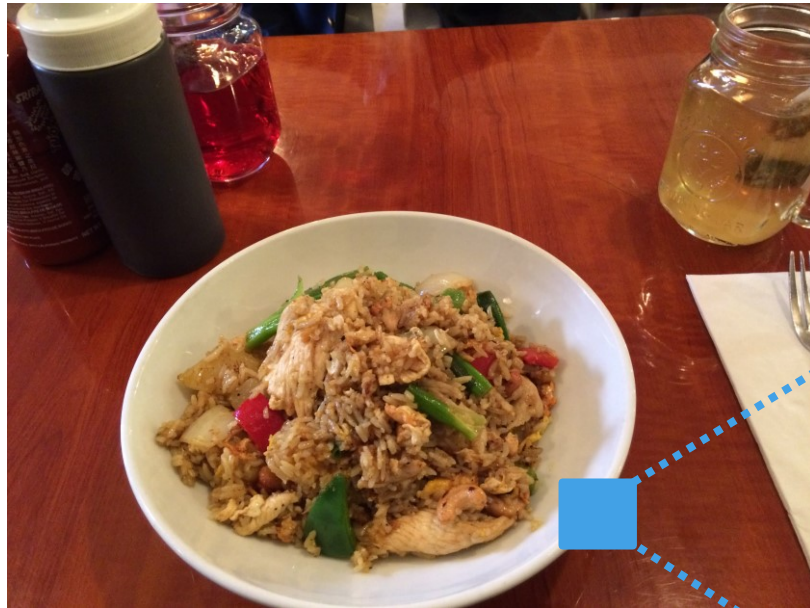


# Semantic Segmentation とは？

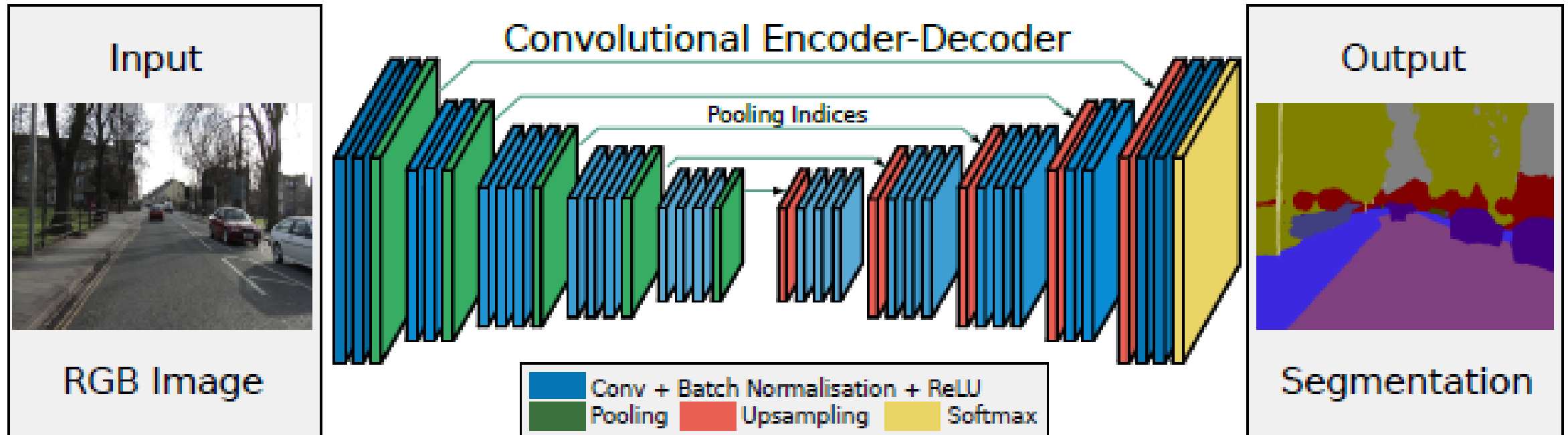
# Semantic Segmentation とは？

R2017b

各ピクセルをその意味（周辺のピクセルの情報）に基づいて、カテゴリ分類する手法



# SegNet (Semantic Segmentation)



Badrinarayanan, V., A. Kendall, and R. Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv. Preprint arXiv: 1511.0051, 2015.



# 逆プーリング (Unpooling)

2	5	3	
4		4	8
3		5	4
5	6	3	

Max Pooling Indices

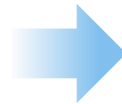


0	0	0	
0		0	0
0		0	0
0	0	0	

最大プーリング  
(Max Pooling)



8	9
7	6



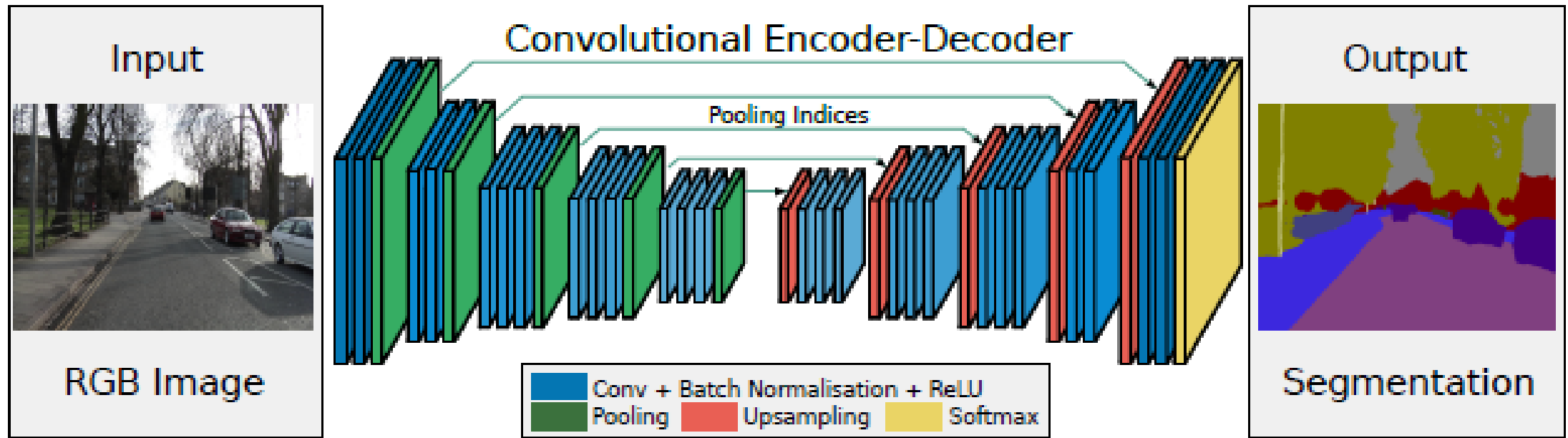
8	9
7	6

逆プーリング  
(Unpooling)



# SegNet (Semantic Segmentation)

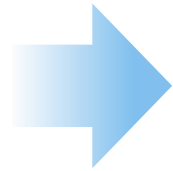
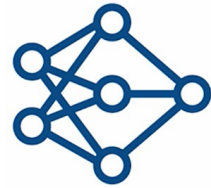
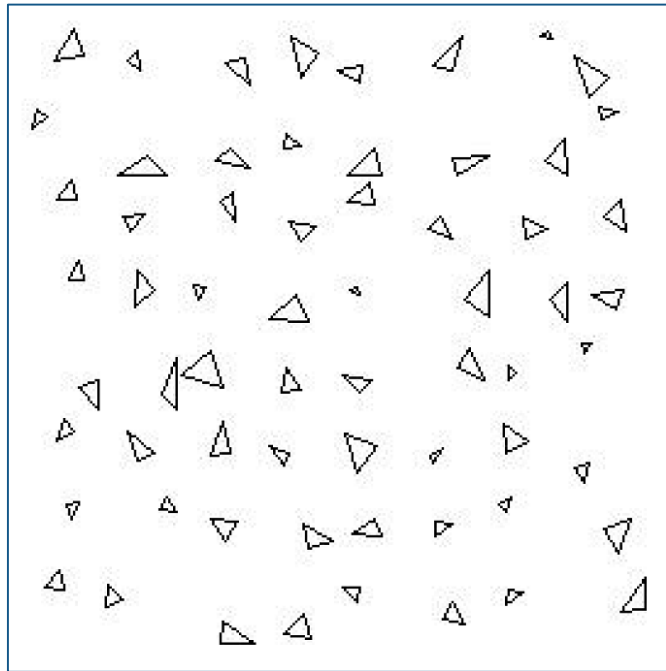
Max Pooling時のIndexを転送して位置に関する情報を補充している



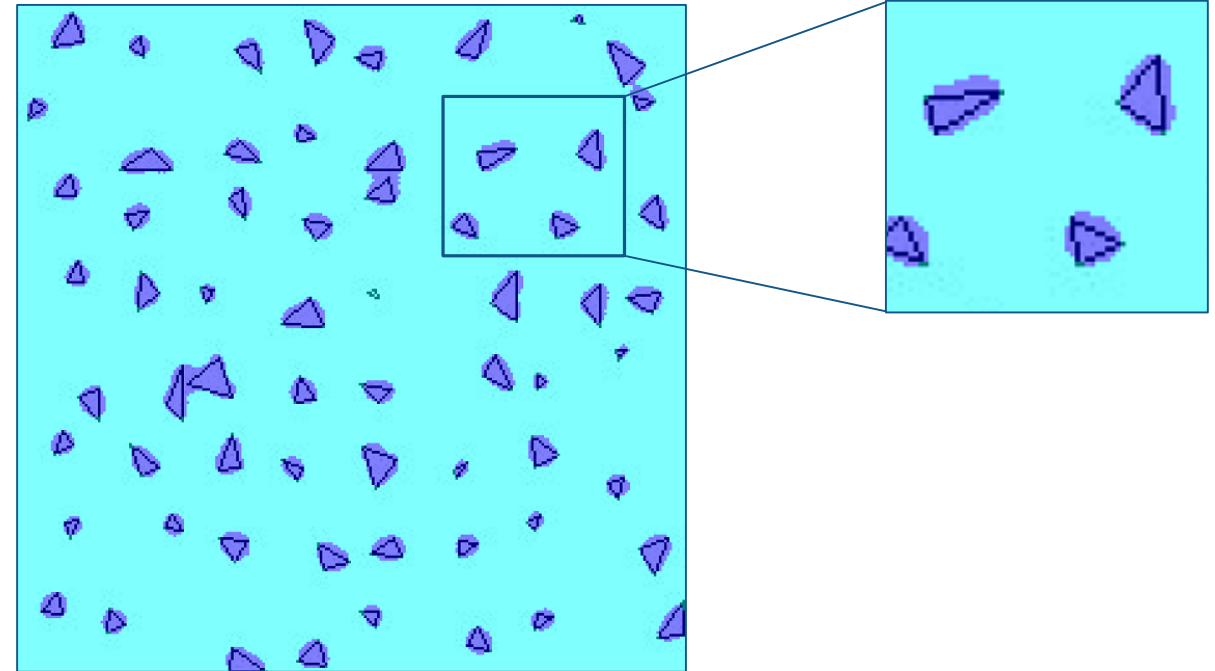
Badrinarayanan, V., A. Kendall, and R. Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." arXiv. Preprint arXiv: 1511.0051, 2015.

# 【例題】 三角形で囲まれた領域の識別

セマンティック・セグメンテーションによるピクセルの分類



Semantic Segmentation

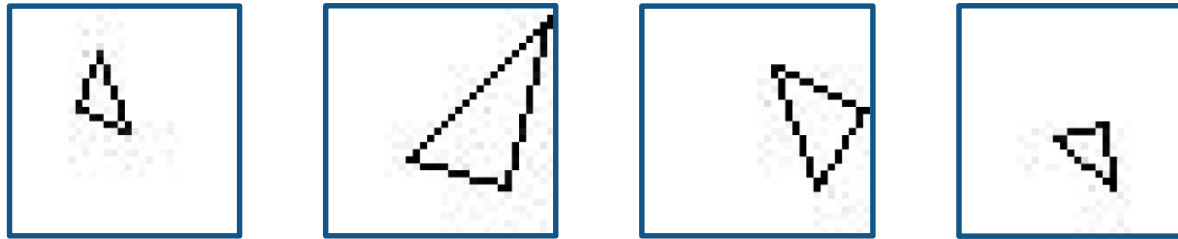


内側のピクセルと外側のピクセル  
をちゃんと識別できている

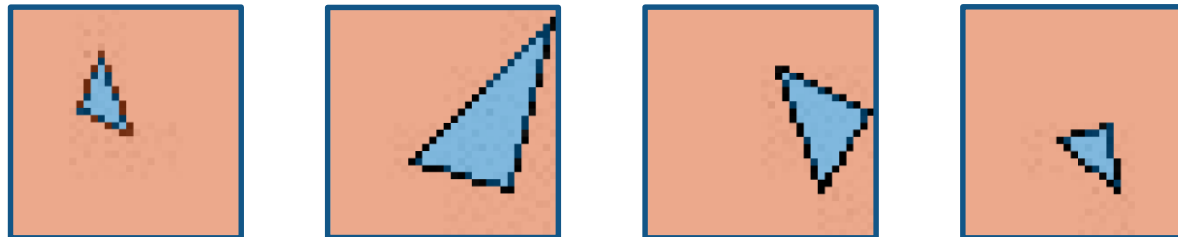
# 【例題】 三角形で囲まれた領域の識別

画像を入力として、ピクセル毎のラベルを出力するネットワークを学習させたい

画像 (入力)  
32 x 32 pixel



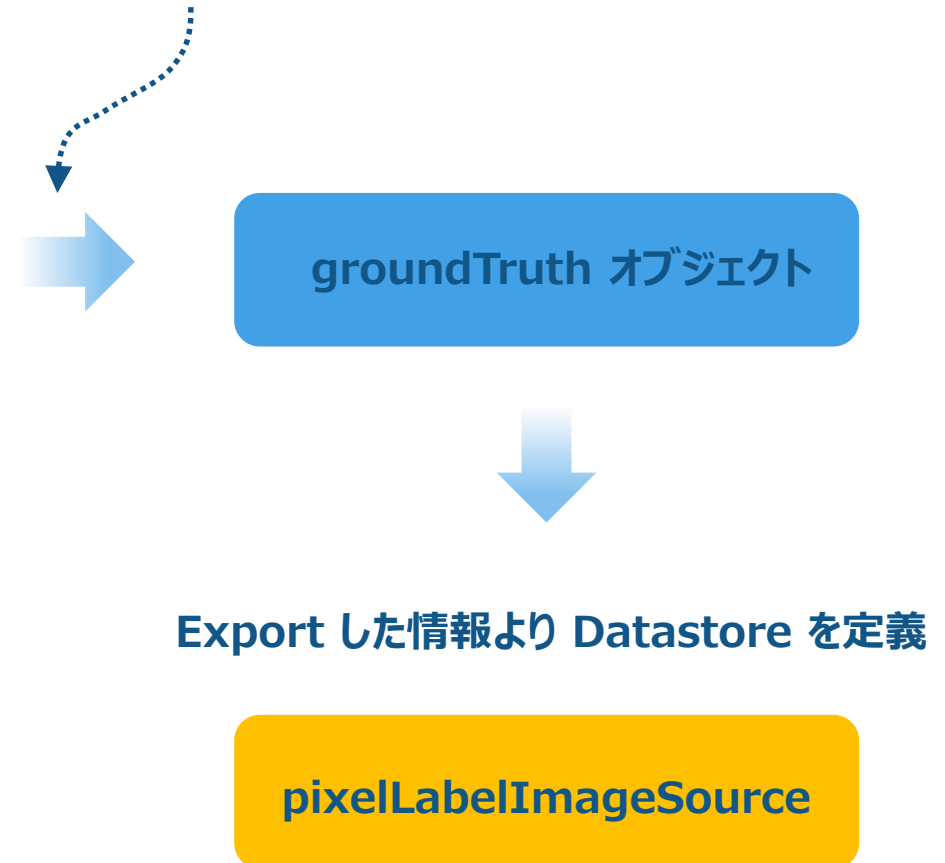
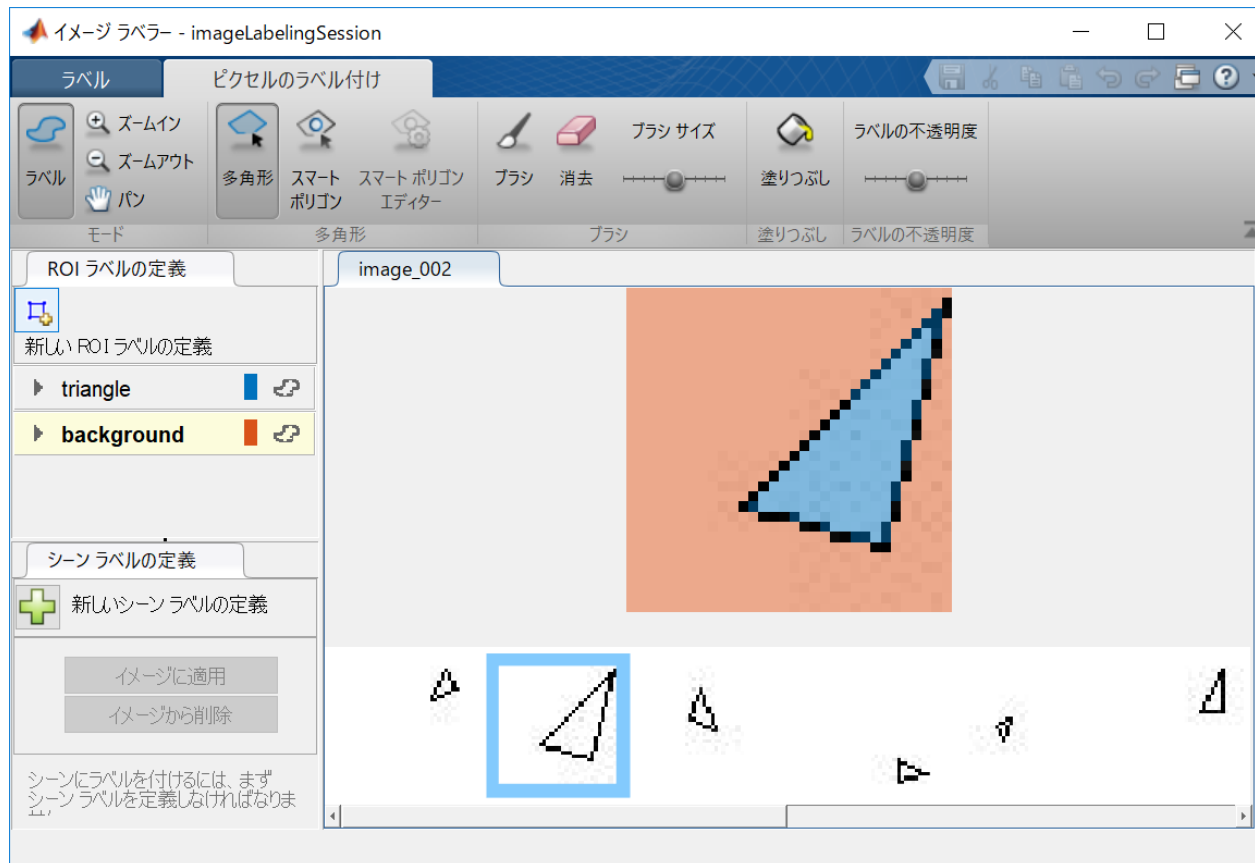
ラベル (出力)  
32 x 32 pixel



# 学習データの準備：ラベルが予め準備されていない場合

イメージ ラベラー（Image Labeler） により、ラベル情報を定義

画像とラベルに関する情報を Export





# 学習と推論 : SegNet の場合

## Step1) SegNet のレイヤーを定義

`lgraph` = `segnetLayers(imageSize, numClasses, encoderDepth)`

Encoder の深さを選択

または

VGG16 または VGG19 を選択

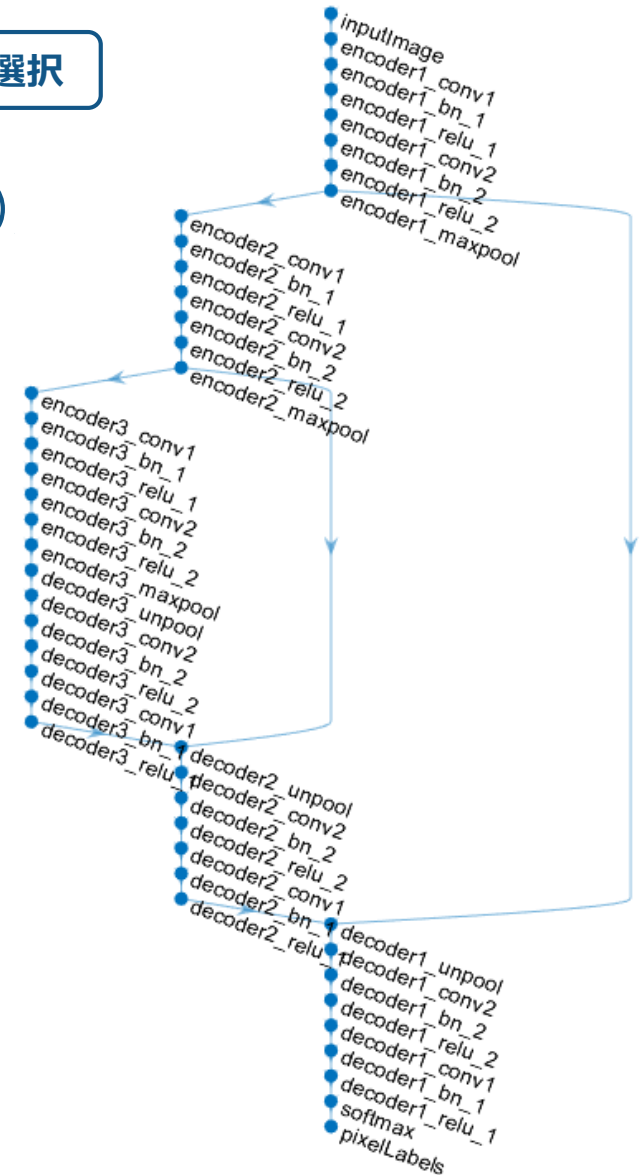
`lgraph` = `segnetLayers(imageSize, numClasses, model);`

## Step2) データ源を指定して、学習を実行する

`net` = `trainNetwork(datasource, lgraph, options);`

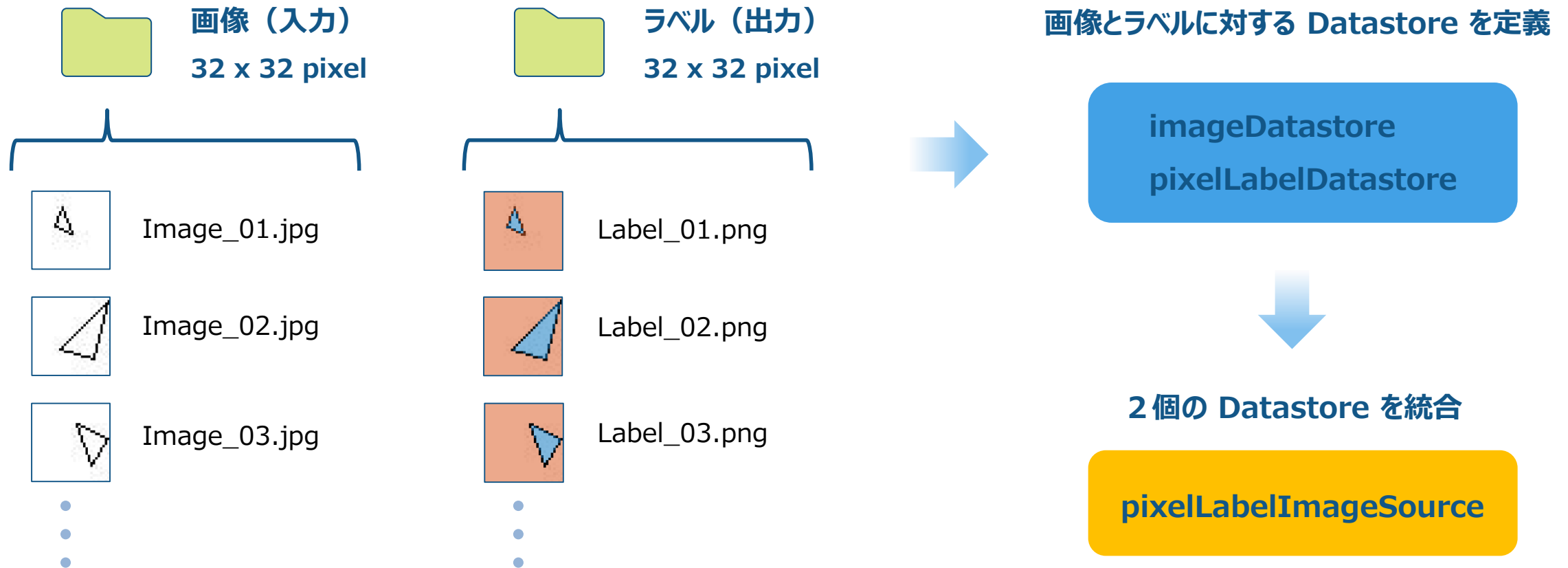
## Step3) 学習させたネットワークでセグメンテーションを行う

`C` = `semanticseg(I, net);`



# 学習データの定義：ラベルが予め準備されている場合

imageDatastore と pixelLabelDatastore を経由して、pixelLabelImageSource を定義する



## 学習データの定義：ラベルが予め準備されている場合

Step1) ラベルの ID番号とラベルのマッピングを決めておく

```
classNames = ["sky" "grass" "building" "sidewalk"];
```

```
pixelLabelID = [1 2 3 4];
```

Step2) 画像とピクセルラベルの組を定義する

```
imds = imageDatastore(imageDir);
```

```
pxds = pixelLabelDatastore(labelDir, classNames, pixelLabelID);
```

building sky



3	3	1	1	1	1	1
3	3	1	1	1	1	1
3	3	3	1	1	1	1
3	3	3	1	1	1	1
3	3	3	1	1	1	1
3	3	3	3	1	1	1
3	3	3	3	3	1	1

Step3) 画像とピクセルラベルの組から学習データを定義する

```
datasource = pixelLabelImageSource(imds, pxds);
```

## 推論結果の評価

テストデータが imageDatastore でポイントされている場合、推論結果の評価は以下の手順で行います

### Step1) 画像データに対して、推論を実行する

```
pxdsResults = semanticseg(imds, net, 'WriteLocation', resultDir);
```

Datastore に対して、推論を実行した場合結果も Datastore となることに注意！

### Step2) 真のラベルデータに対して、Datastore を定義する

```
pxdsTruth = pixelLabelDatastore(labelDir, classNames, pixelLabelID);
```

### Step3) 画像とピクセルラベルの組から学習データを定義する

```
metrics = evaluateSemanticSegmentation(pxdsResults, pxdsTruth);
```

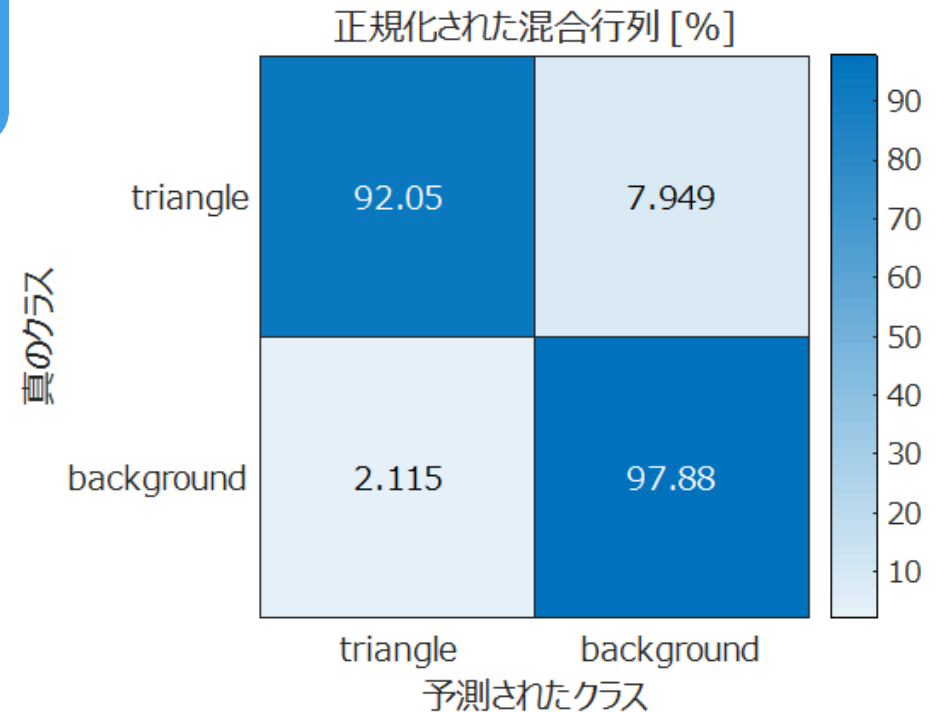
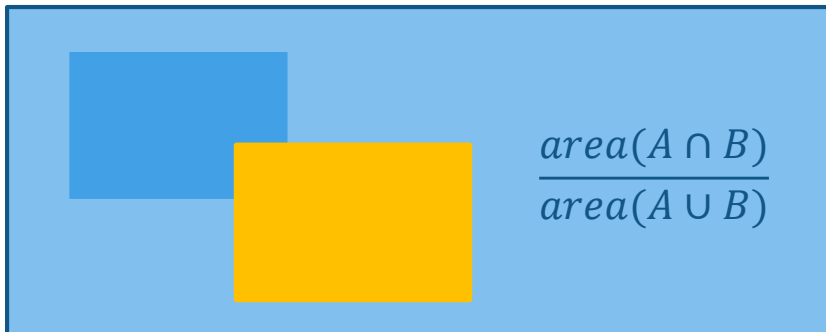
# 推論結果の評価

```
metrics = evaluateSemanticSegmentation(pxdsResults, pxdsTruth);
```

**ConfusionMatrix** : 混合行列  
**NormalizedConfusionMatrix** : 正規化された混合行列

**DataSetMetrics** : データセット全体のメトリック  
**ImageMetrics** : 画像ごとのメトリック  
**ClassMetrics** : クラスのメトリック

IoU (Intersection over Union)



# 推論結果の評価

## データセット全体のメトリック (DataSetMetrics)

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.98	0.95	0.81	0.96	0.59

## 画像ごとのメトリック (ImageMetrics)

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.99	0.88	0.83	0.99	0.77
0.98	0.96	0.79	0.96	0.56
0.97	0.96	0.85	0.94	0.55
0.97	0.96	0.78	0.95	0.52

## クラスごとのメトリック (ClassMetrics)

	Accuracy	IoU	MeanBFScore
triangle	0.92	0.64	0.31
background	0.98	0.98	0.87

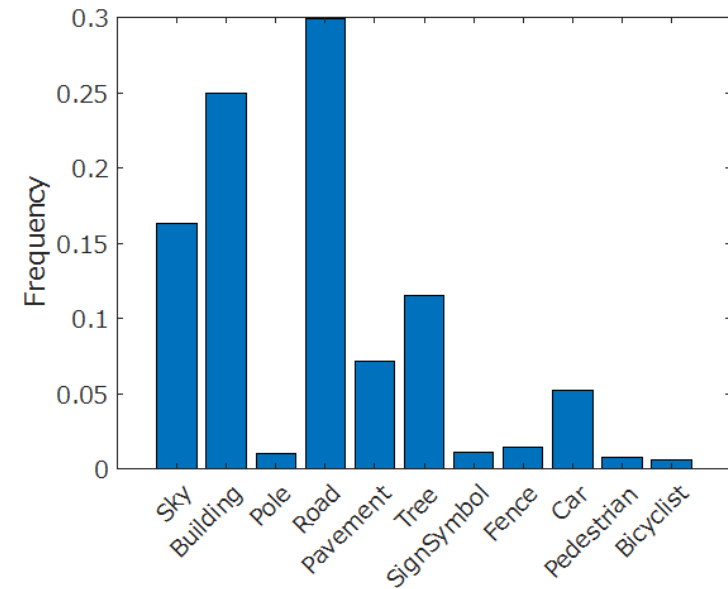
# 【例題】CamVid データセットへの SegNet の適用

運転中の自動車から取得した画像（CamVid データセット）に対して、SegNet を適用した例題がある

<https://www.mathworks.com/help/releases/R2018a/vision/examples/semantic-segmentation-using-deep-learning.html>



クラス毎にピクセルの出現頻度が異なるので学習時にウェイトを付けてやるのがポイント！

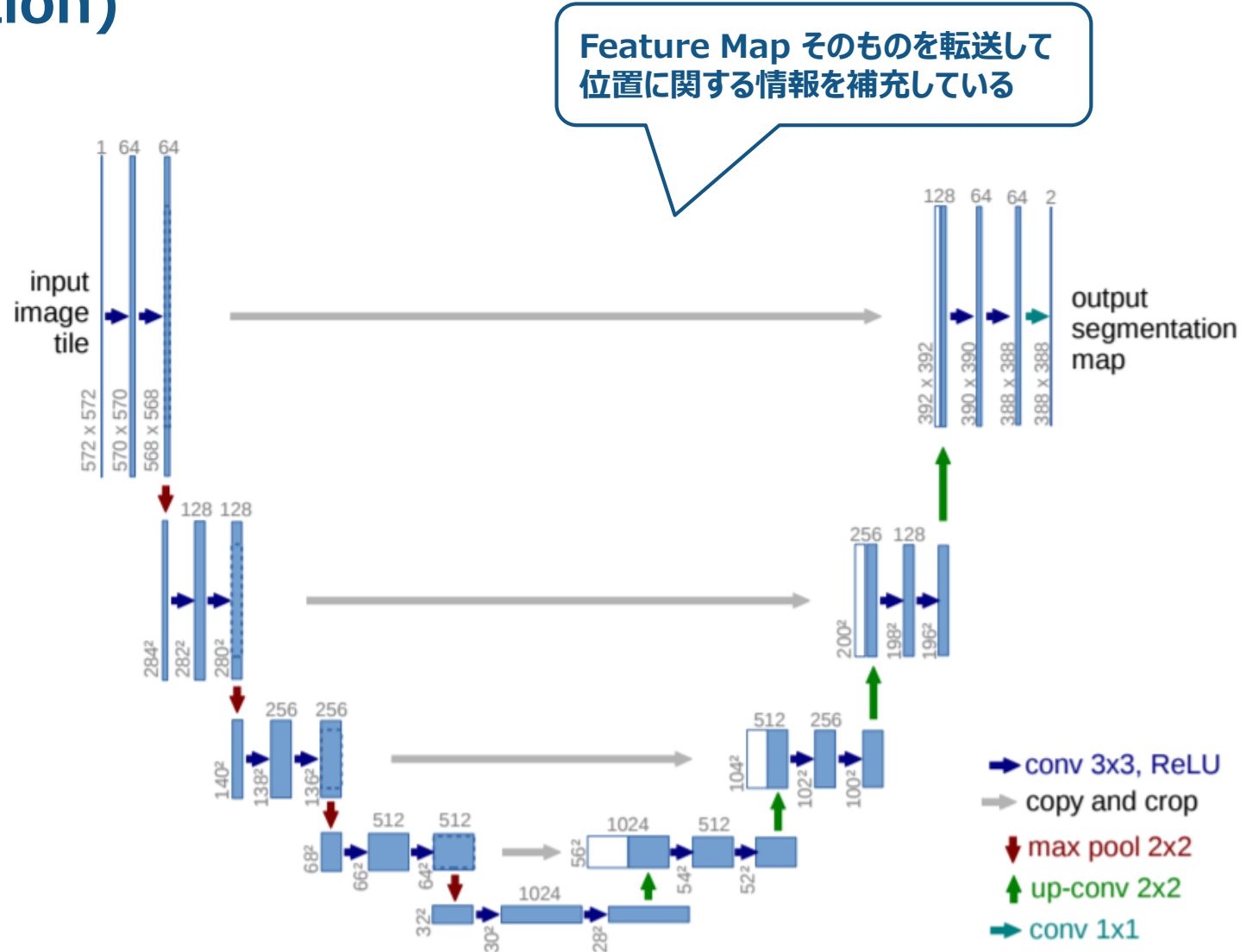
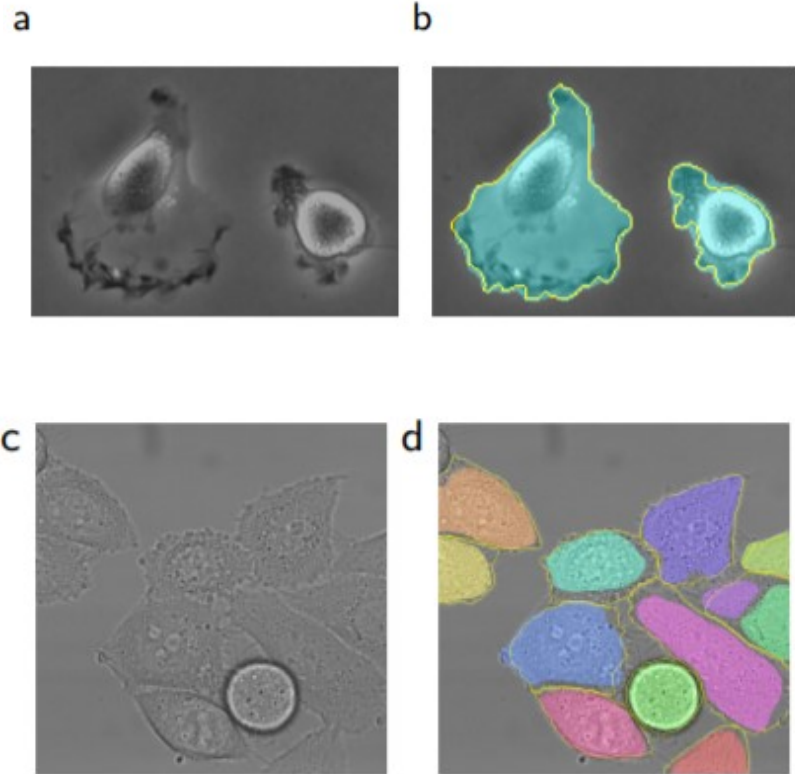


<http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/>



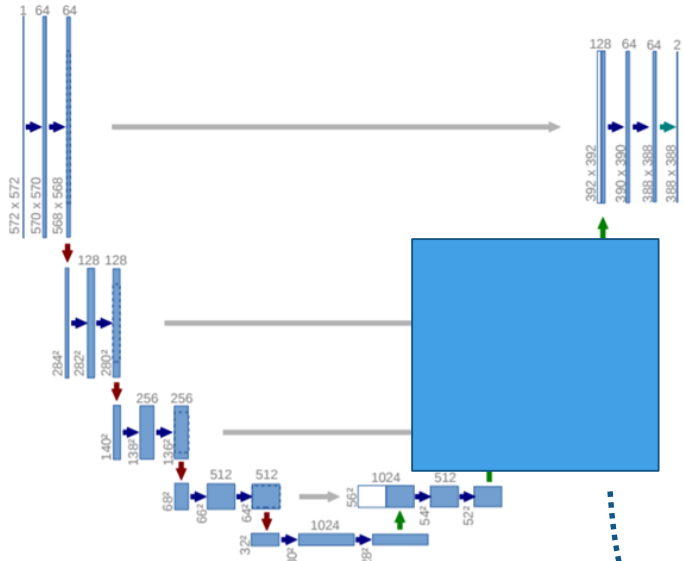
# U-Net とは？

# U-Net (Semantic Segmentation)

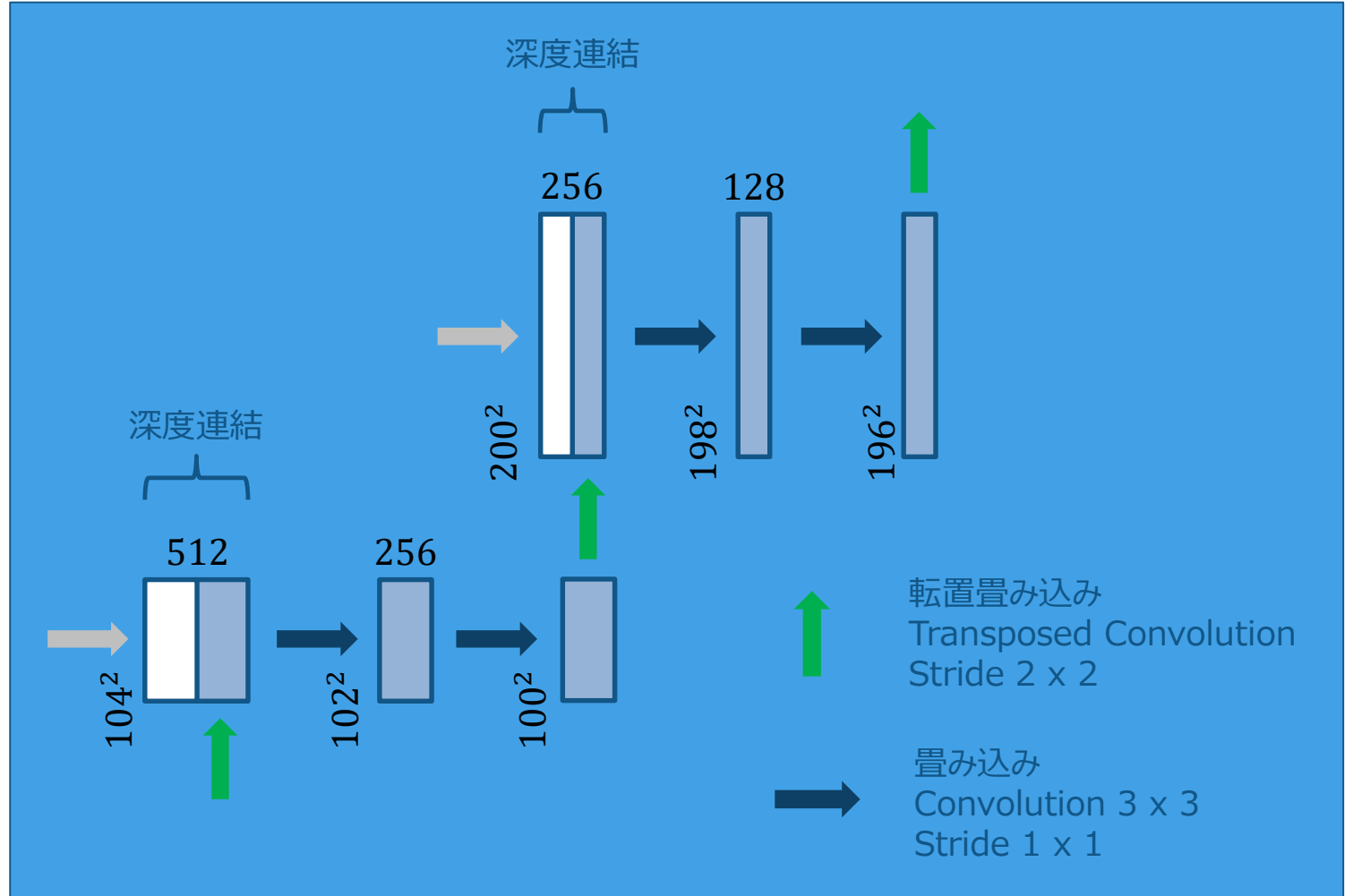


O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation" in MICCAI, pp. 234–241, Springer, 2015.

# U-Net (Semantic Segmentation)



拡大!



O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation" in MICCAI, pp. 234–241, Springer, 2015.

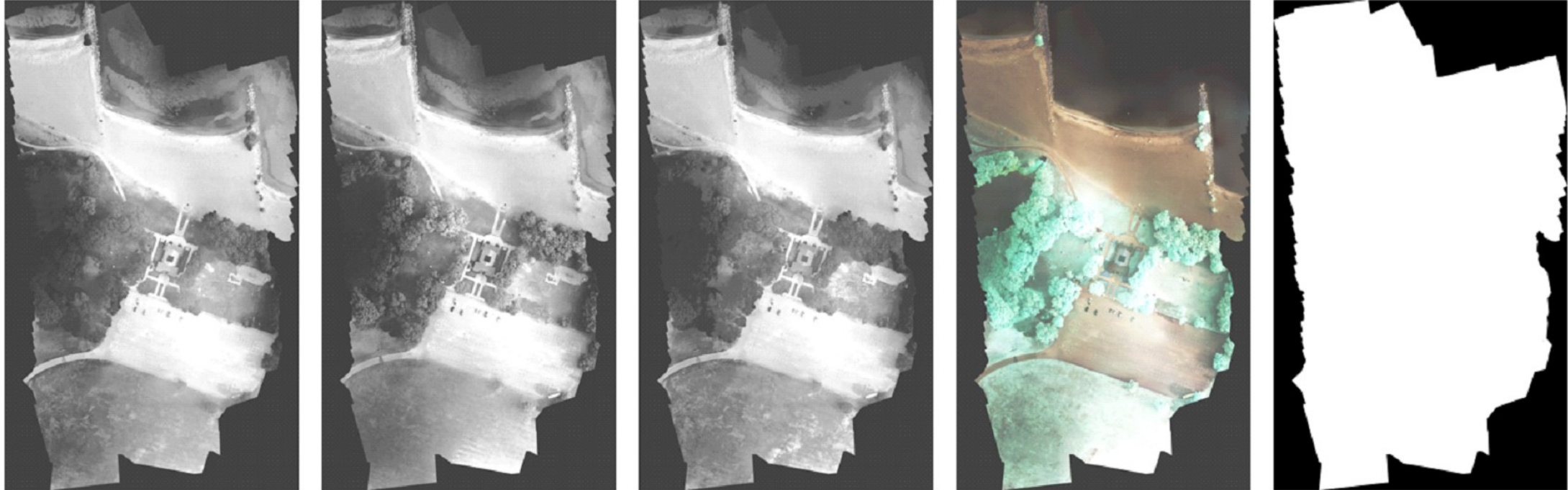
## 【例題】マルチスペクトル画像への U-Net の適用

マルチスペクトル画像に対して、U-Net を適用した例題がある

赤外線チャンネル  
CH1 ~ CH3

RGB 画像  
CH4 ~ CH6

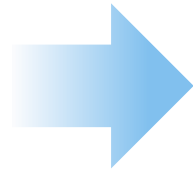
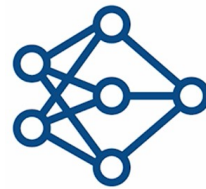
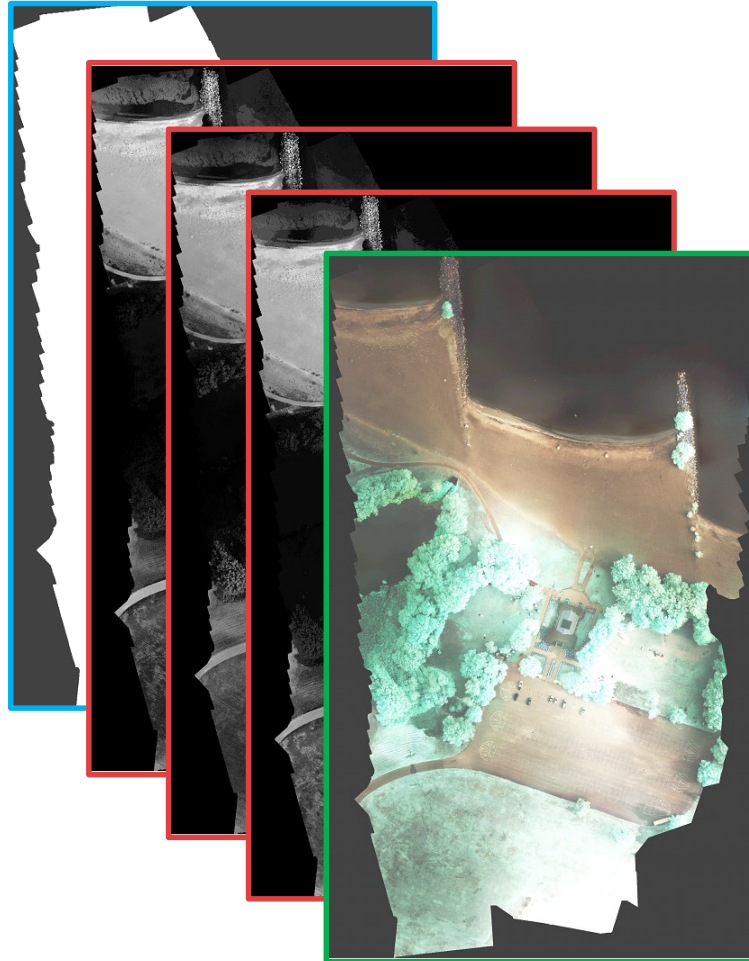
マスク画像  
CH7



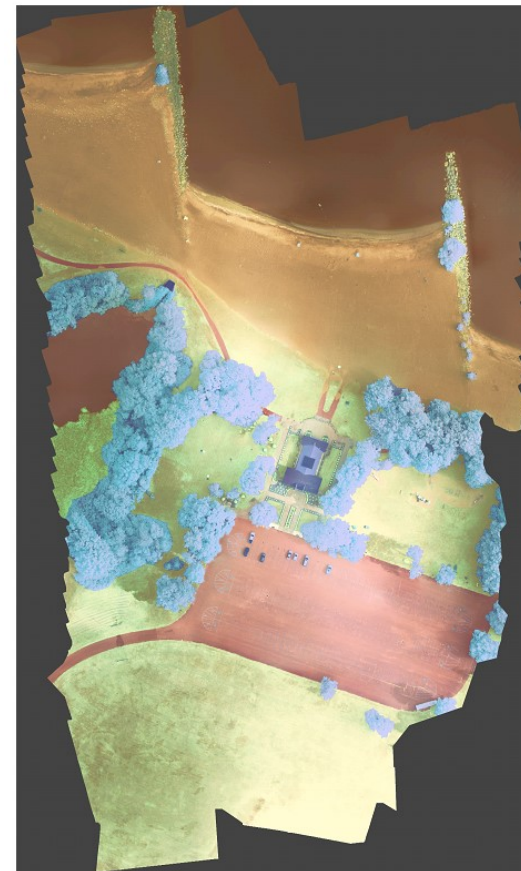
<https://www.mathworks.com/help/releases/R2018a/images/multispectral-semantic-segmentation-using-deep-learning.html>

# 【例題】マルチスペクトル画像への U-Net の適用

マルチスペクトル画像から、被植率 (Vegetation Cover) を算出したい



**Semantic Segmentation**

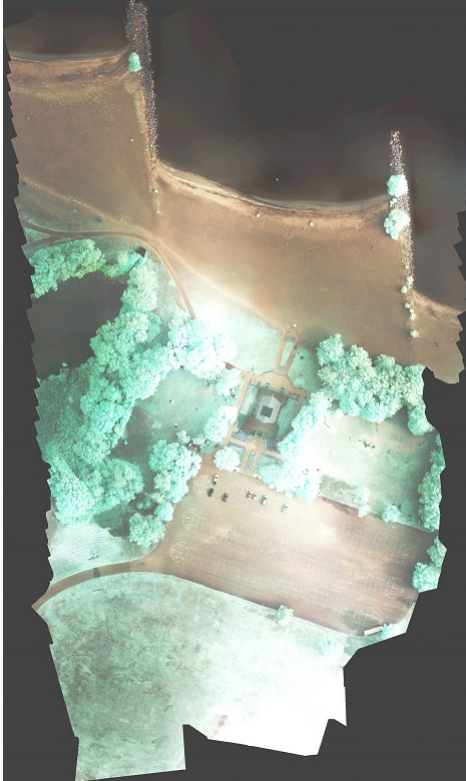


- Asphalt
- Water\_Pond
- Water\_Lake
- Sand\_Beach
- 
- Rocks
- Buoy
- OrangeLandingPad
- WhiteWoodPanel
- BlackWoodPanel
- PicnicTable
- LifeguardChair
- Person
- Vehicle
- Building
- 
- RoadMarkings



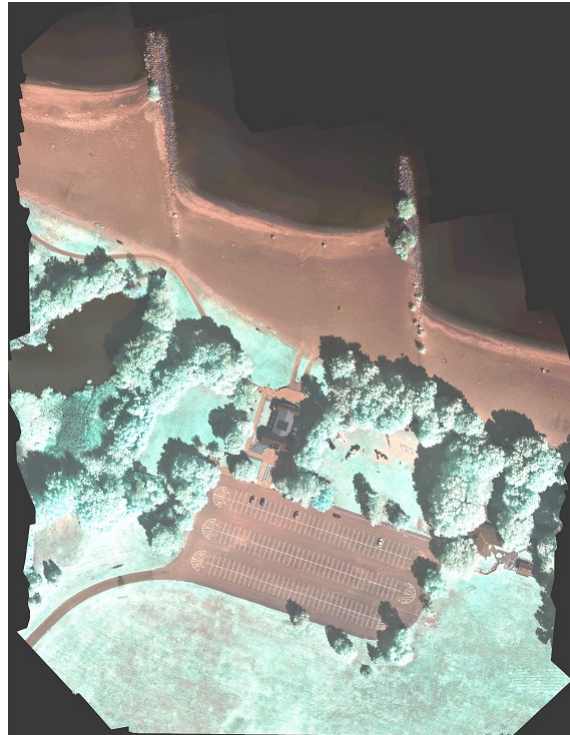
## 【例題】マルチスペクトル画像への U-Net の適用

学習用・検証用・テスト用のそれぞれに、サイズの異なる画像が 1 枚ずつ存在する



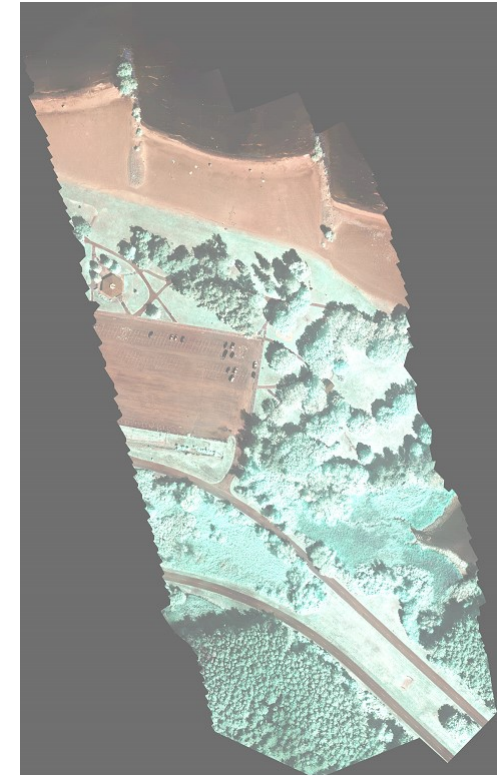
学習用画像

9393 x 5642 x 7



検証用画像

8833 x 6918 x 7

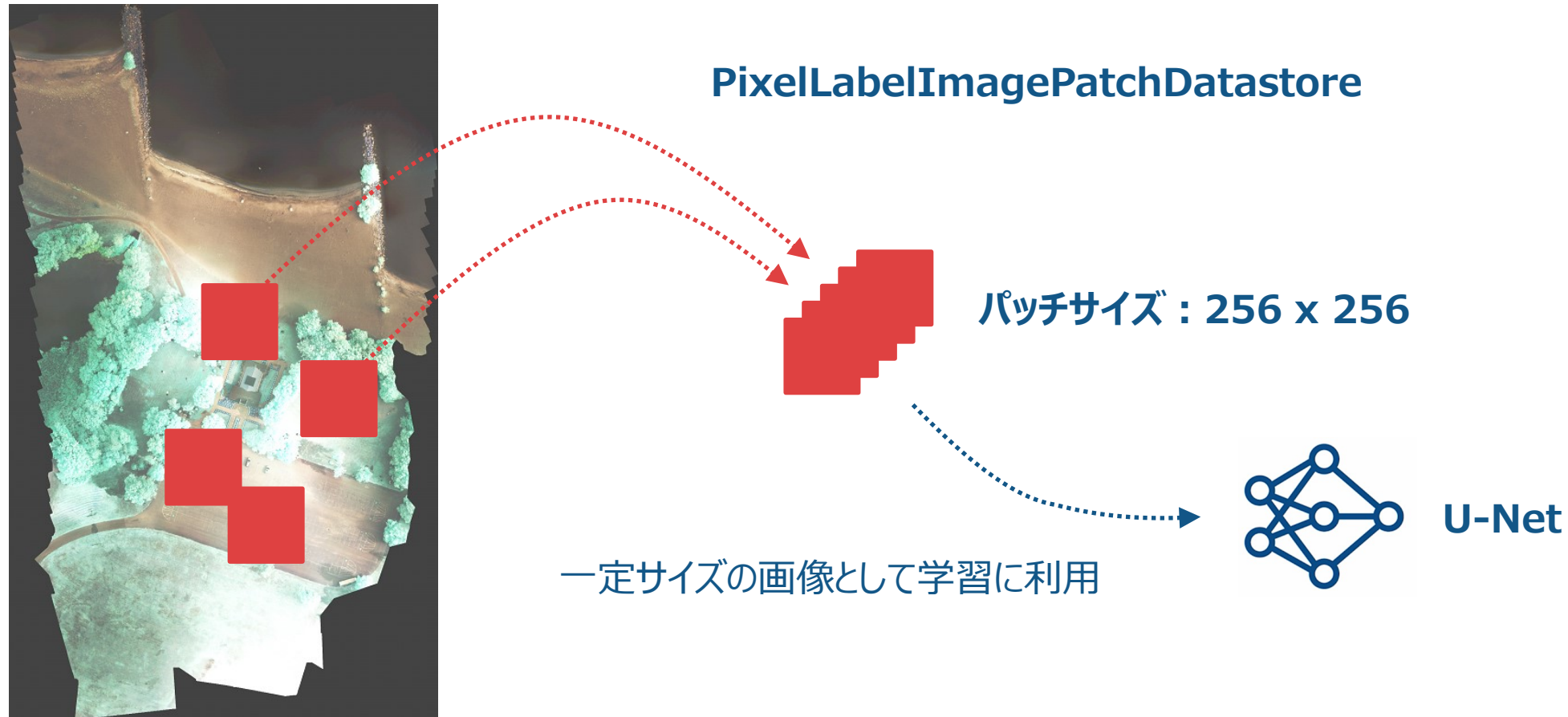


テスト用画像

12446 x 7654 x 7

## 【例題】マルチスペクトル画像への U-Net の適用（学習時の工夫）

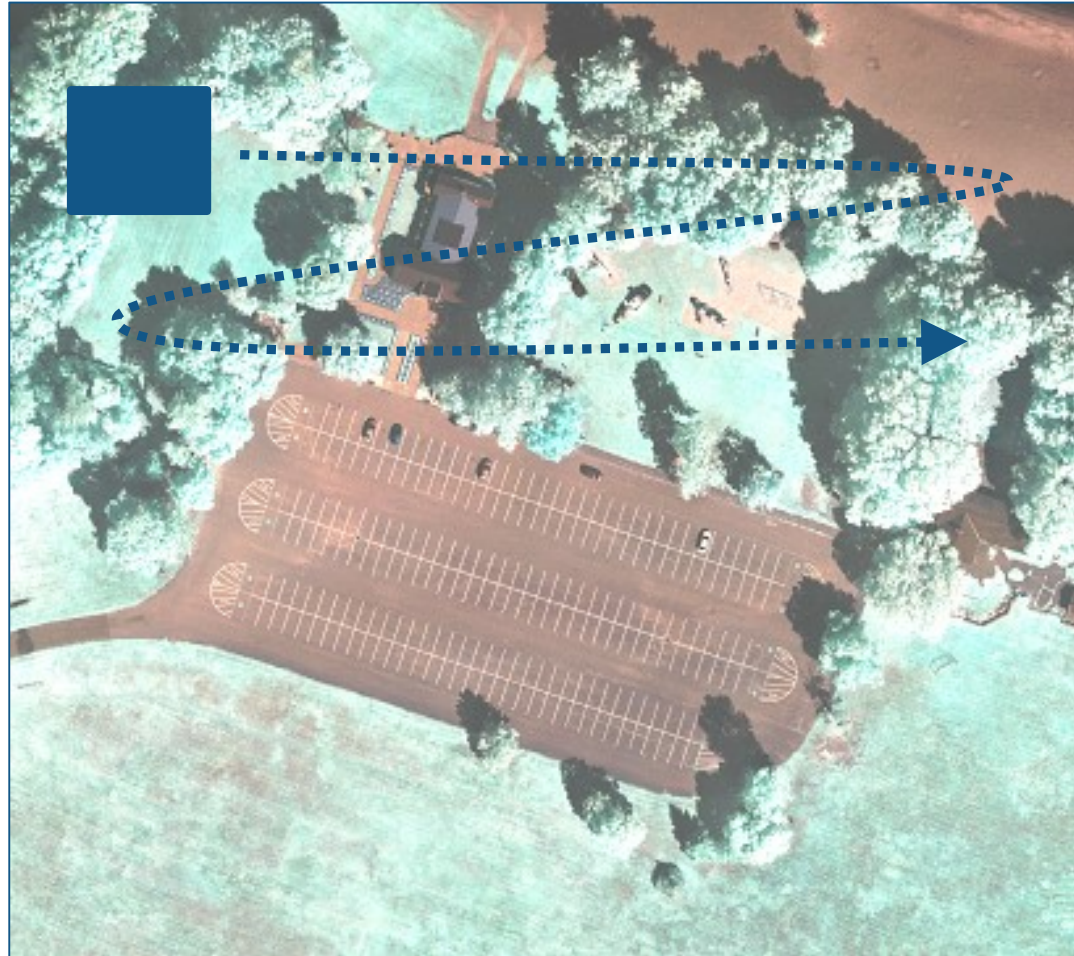
サイズや形の違う画像から、一定サイズの画像を切り出して使うカスタムの Datastore が利用されている





## 【例題】マルチスペクトル画像への U-Net の適用（推論時の工夫）

サイズの大きな画像から小さなパッチ領域を滑らせつつ推論を実行している（関数：segmentImage）



## 【例題】マルチスペクトル画像への U-Net の適用（カスタム関数・クラスのリスト）

本例題で利用されているカスタム関数・カスタムクラスのリスト

- **PixelLabelImageDatastore（データ読み込み）**
  - サイズや形の違う画像から一定サイズのパッチを切り出して読み出す
- **segmentImage（推論）**
  - サイズの大きい画像を小さなパッチに分けて推論を実行する
  - 使用メモリを削減する効果がある
- **createUnet（ネットワーク構築）**
  - U-Net に似たカスタムのネットワークを構築する
- **matReader（データ読み込み）**
  - imageDatastore のカスタム化に利用する
  - MATファイルに格納されたデータの読み込みに利用する

# 画像系ディープラーニングのための構成

# 深層学習に必要な Toolbox と Hardware

MATLAB

← R2018a以降のMATLABを推奨

Neural Network Toolbox™

← 必須

Parallel Computing Toolbox™

← GPUを使う場合 ..... →

Statistics and Machine Learning Toolbox™

← 一部の例題等で使う可能性

Image Processing Toolbox™

Computer Vision System Toolbox™

← R-CNN, Fast R-CNN, Faster R-CNN  
Semantic Segmentation 等で必須



NVIDIA® のチップを搭載したGPU  
(Compute Capability 3.0以降)



© 2018 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.