6．オブジェクトの検出と判別

6.1 果実の検出と熟度の判別

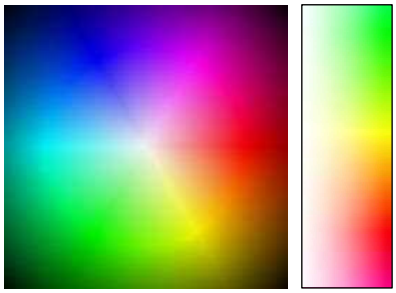
6.2 数字や文字の検出と認識

6.3 揺れる車載カメラ映像の安定化

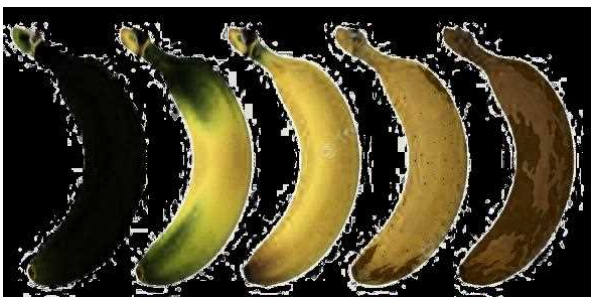
6.4 深層学習によるオブジェクト検出

6.1 果実の検出と熟度の判別　(1/4)

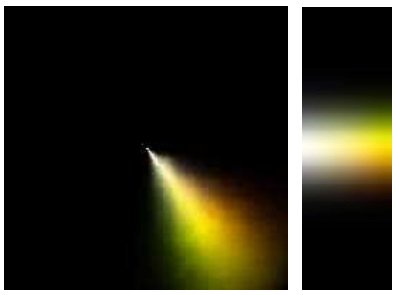
画像出典：freepik.com



色相（h=-53°　付近をガウス窓で切り出した）

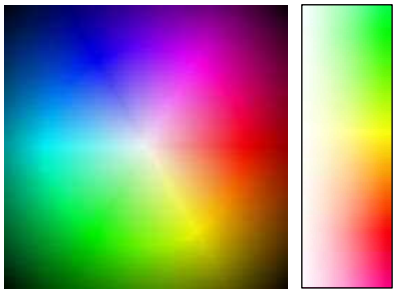


不要な「白」
も抽出された



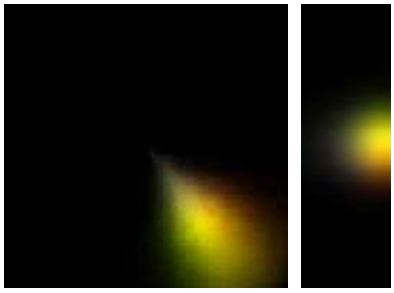
6.1 果実の検出と熟度の判別　(2/4)

画像出典：freepik.com



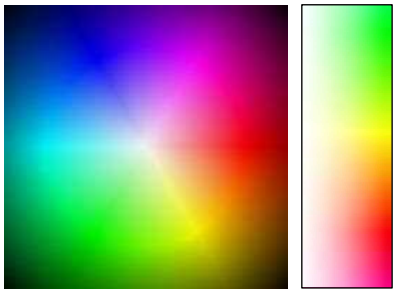
色相（h=-53°付近）かつ彩度（s=1.0 付近）
をガウス窓で切り出した

不要な「白」
は削除された



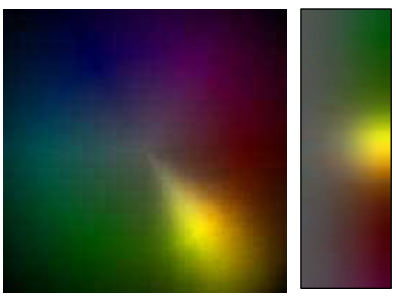
6.1 果実の検出と熟度の判別　(3/4)

画像出典：freepik.com



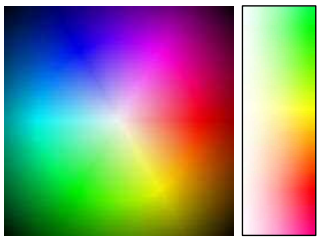
色相（h=-53°付近）かつ彩度（s=1.0 付近）
を抽出して、元画像との相加平均を出力した

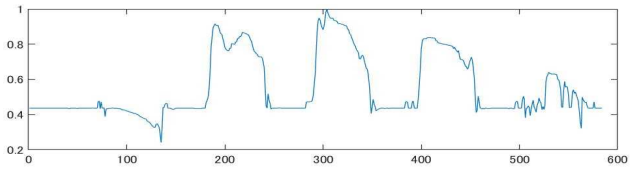
元の画像の
熟度を強調

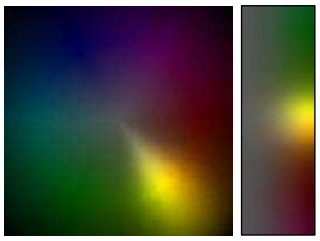


6.1 果実の検出と熟度の判別　(4/4)

画像出典：freepik.com

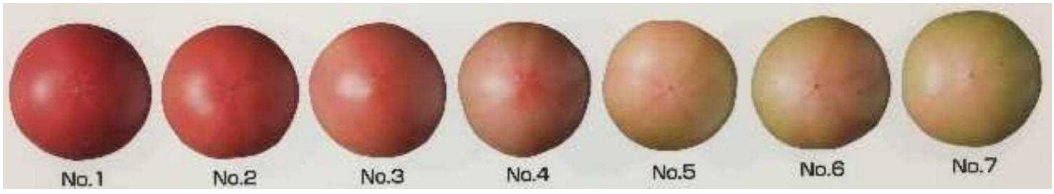


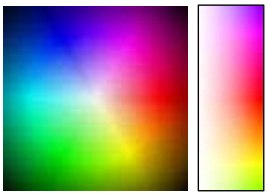




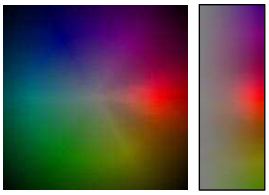
バナナの熟度を
定量評価できた

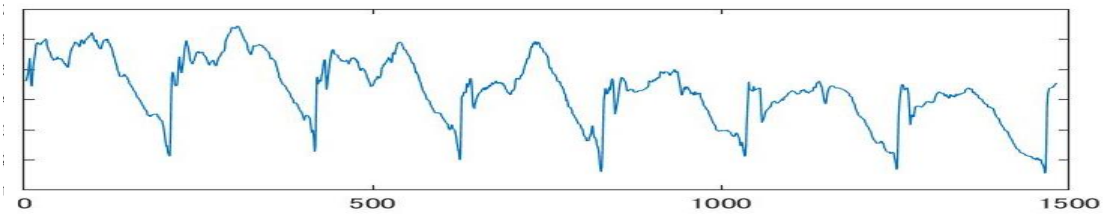
トマトの熟度を定量評価



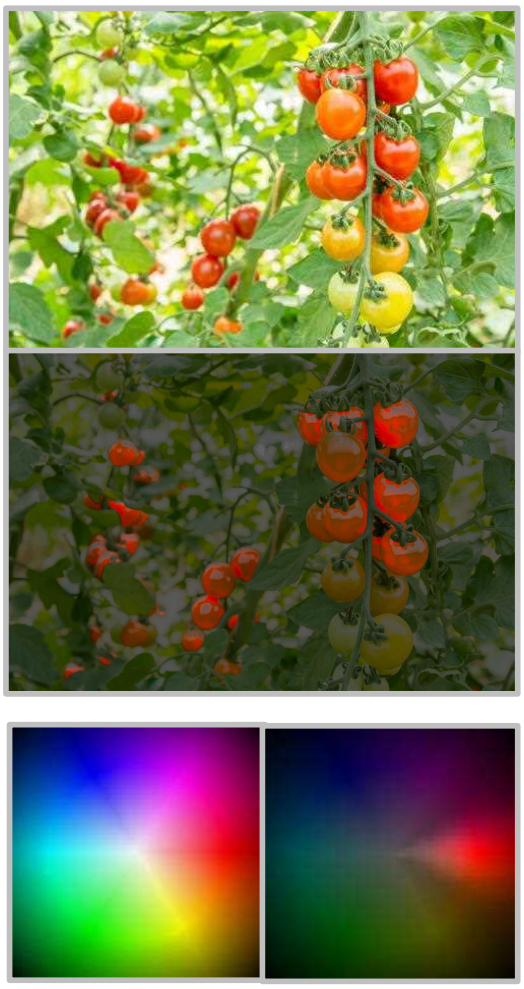


画像出典：千葉県トマト・カラーチャート

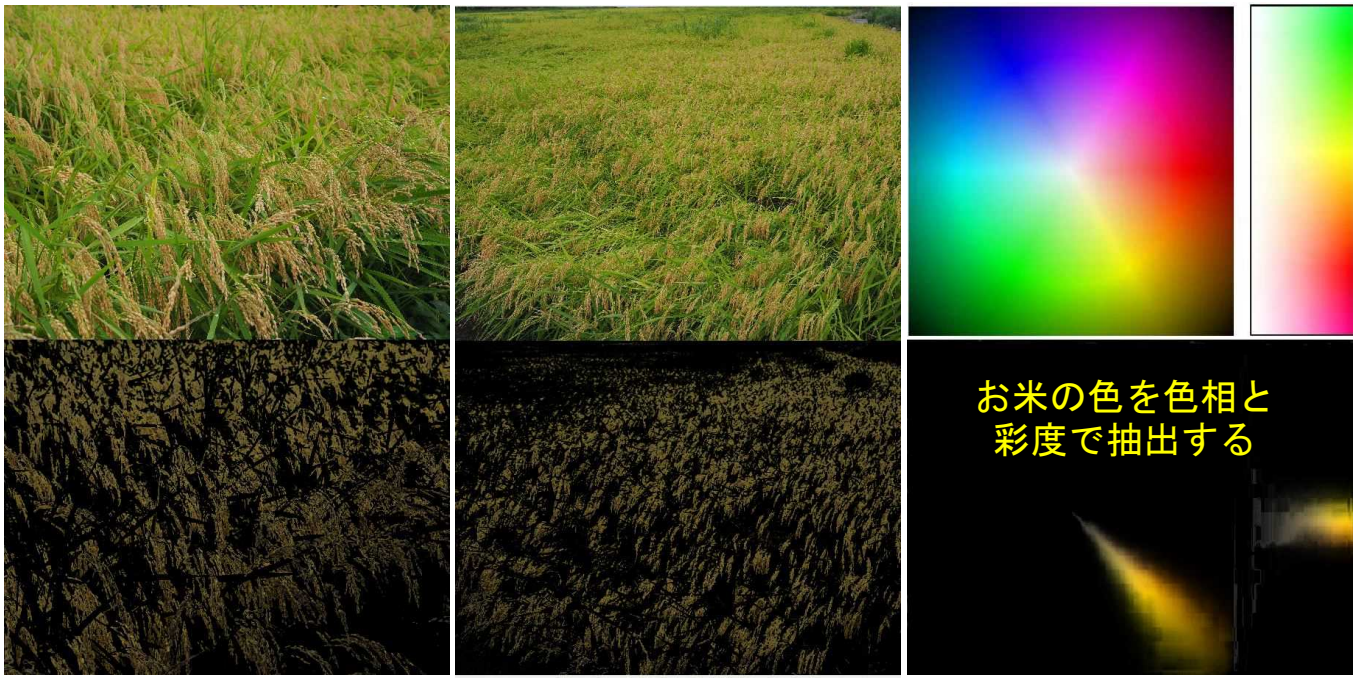




色相（h=0° 付近）かつ彩度（s=1.0 付近）
を抽出して、元画像との相加平均を出力



稲の育成を色でチェック



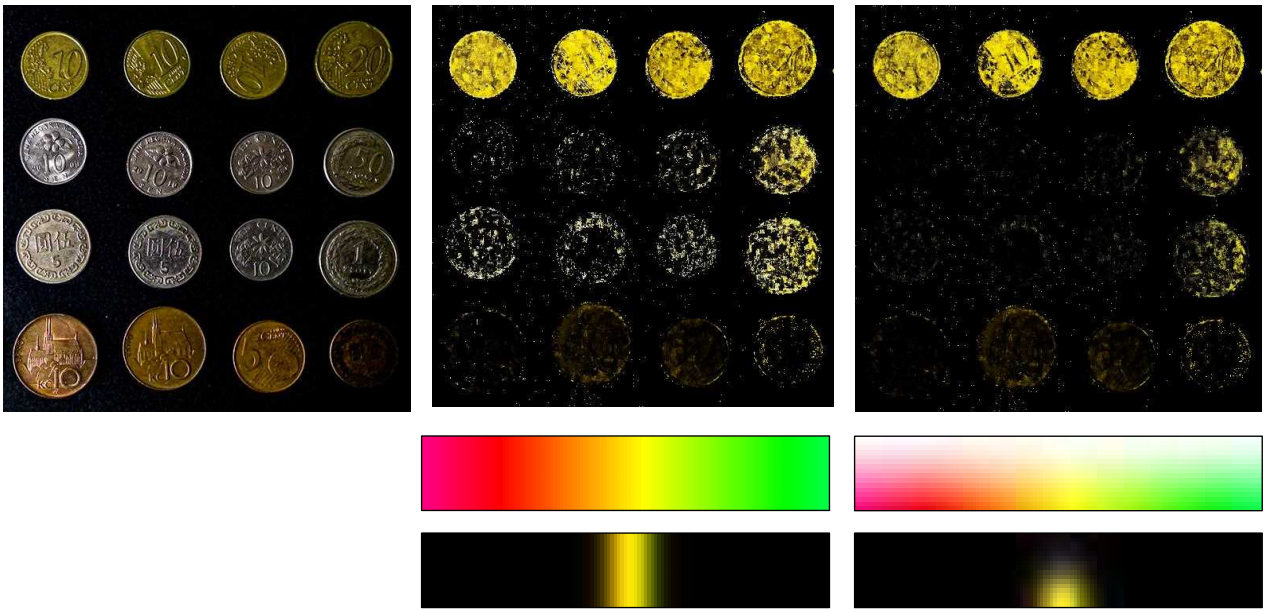
マゼンダ色の領域をガウス窓で抽出して明度を強調

特定の色の貨幣を抽出して計量する (1/2)

元の画像
（諸外国の硬貨）

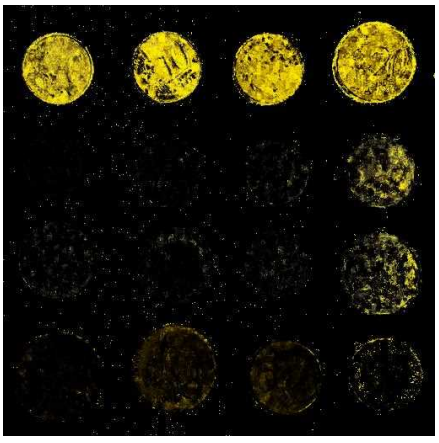
特定の色相を抽出
（彩度は無視）

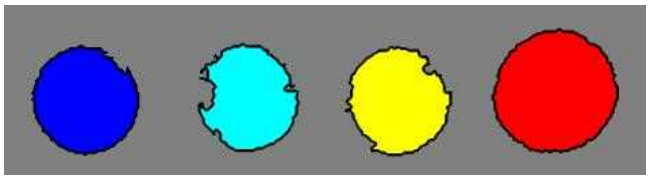
特定の色相かつ
高い彩度を抽出



```
clear all;

I1=imread('貨幣.jpg');
hsv=rgb2hsv(I1);
h=hsv(:,:,1);% 色相
s=hsv(:,:,2);% 彩度
v=hsv(:,:,3);% 明度

SgH=0.02; Heu=0.15;% 特定の色相を抽出
m=h-Heu; m(m>0.5)=m(m>0.5)-1;
m=m.^2; m=exp(-m/2/SgH/SgH);
mH=m;

SgS=0.30; Sat=1.00;% 特定の彩度を抽出
m=(s-Sat).^2; m=exp(-m/2/SgS/SgS);
mS=m;

hsv(:,:,3)=mH.*mS;
I2=hsv2rgb(hsv);

montage({I1,I2});
imwrite(I2,'o2.jpg');
```

特定の色の貨幣を抽出して計量する (2/2)





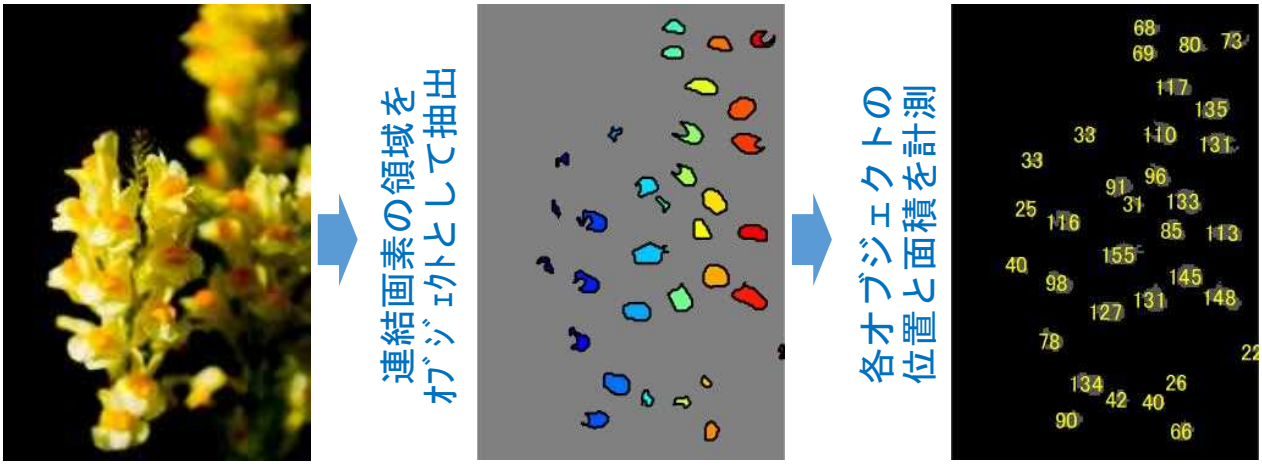
```
bwareaopen(bw,500);
% 500ピクセル未満のオブジェクトを削除)
imfill(bw,'holes');
% 穴を塗りつぶす
```

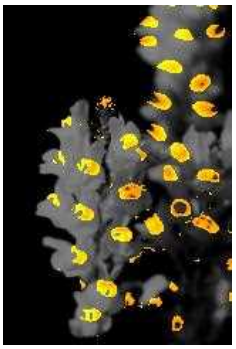
半径を表示した　→


```
bwboundaries(bw,'noholes');
% 境界線を検出する
```

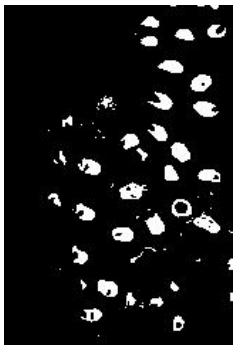
```
regionprops(L,'Area','Centroid','MajorAxisLength','MinorAxisLength');
% 面積、重心、長軸、短軸を検出する
```

```
bw=imbinarize(rgb2gray(I2));
bw=bwareaopen(bw,500);% 500pxl未満を削除
bw=imfill(bw,'holes');% 穴を塗りつぶす

figure(1);
[B,L] = bwboundaries(bw,'noholes');% 境界線を検出する
rgb=label2rgb(L,@jet,[.5 .5 .5]);  % 連続領域のラベル行列
imshow(rgb); hold on
for k = 1:length(B)
  boundary = B{k}; % 境界画素の位置
  plot(boundary(:,2),boundary(:,1),'w','LineWidth',1)
end


figure(2);
imshow(bw/3); hold on;
stats =
regionprops(L,'Area','Centroid','MajorAxisLength','MinorAxisLength');

for k = 1:length(B)
  cent = stats(k).Centroid;% 重心
  area = stats(k).Area;    % 面積
  D1   = stats(k).MajorAxisLength;
  D2   = stats(k).MinorAxisLength;
  radii=(D1+D2)/4;         % 半径

  string= sprintf('%3d',round(radii));
  text(cent(1,1)-10,cent(1,2),string,'Color','c','FontSize',12)
  hold on; viscircles(cent,radii);
end
```



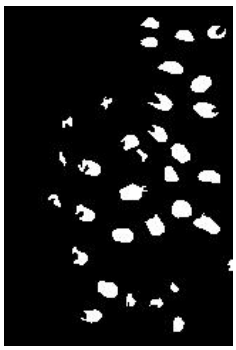


特定の
色相を抽出

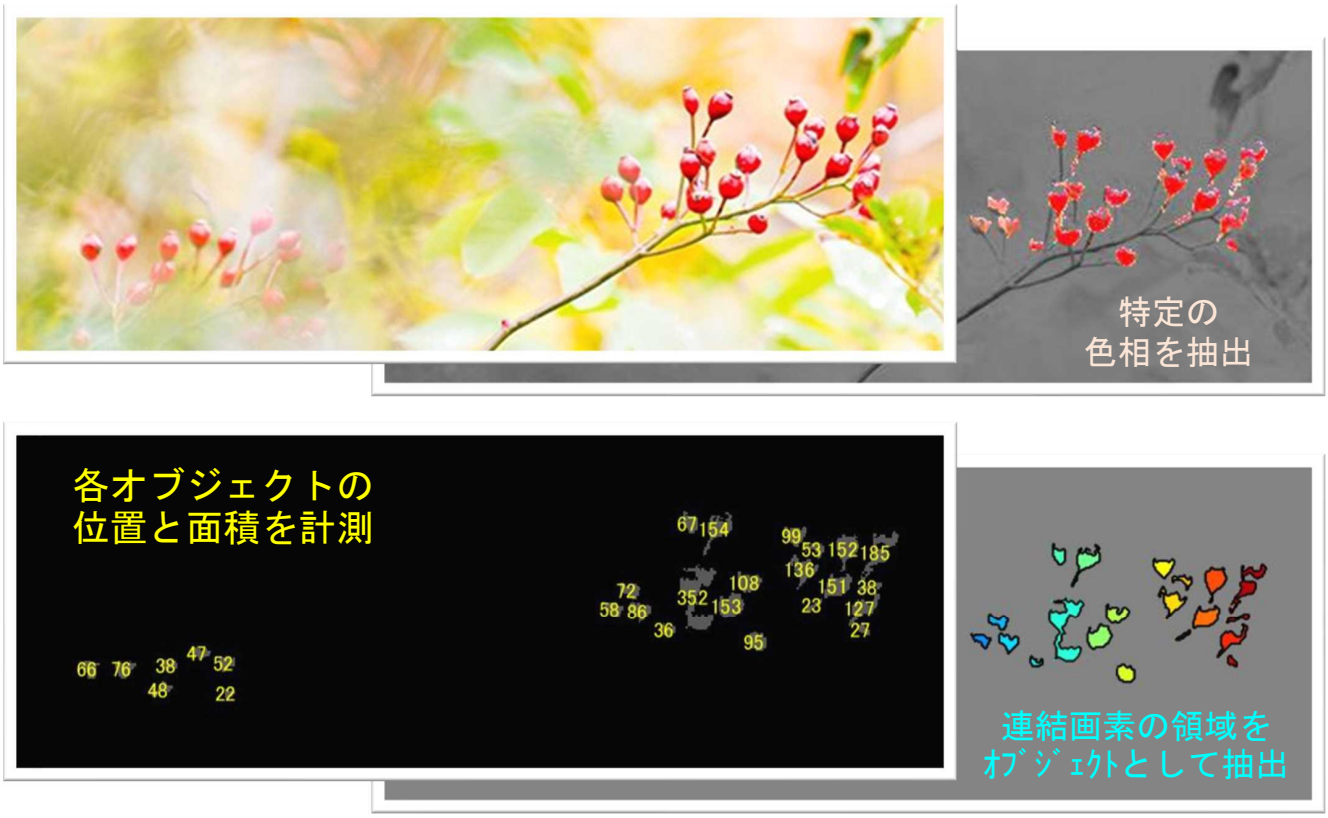


imbinarize(x);
画像を二値化

bwareaopen(bw,20)
20ピクセル未満の
オブジェクトを削除



imfill(bw,'holes')
穴を塗りつぶす



6．オブジェクトの検出と判別

6.1 果実の検出と熟度の判別

6.2 数字や文字の検出と認識

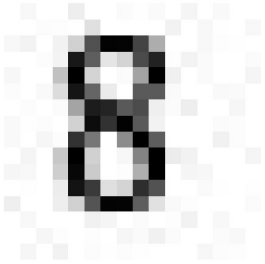
6.3 揺れる車載カメラ映像の安定化

6.4 深層学習によるオブジェクト検出

光学式文字認識 (OCR)

手書きの数字（0～9）を機械で判別する方法

HOGの特徴
ベクトルを計算

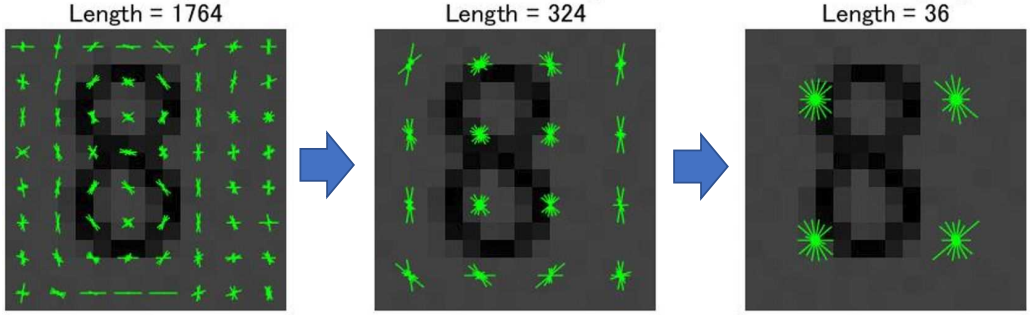


ベクトルが大きすぎ　　**丁度良い**　　**特徴を捉えていない**

CellSize = [2 2]
Length = 1764

CellSize = [4 4]
Length = 324

CellSize = [8 8]
Length = 36



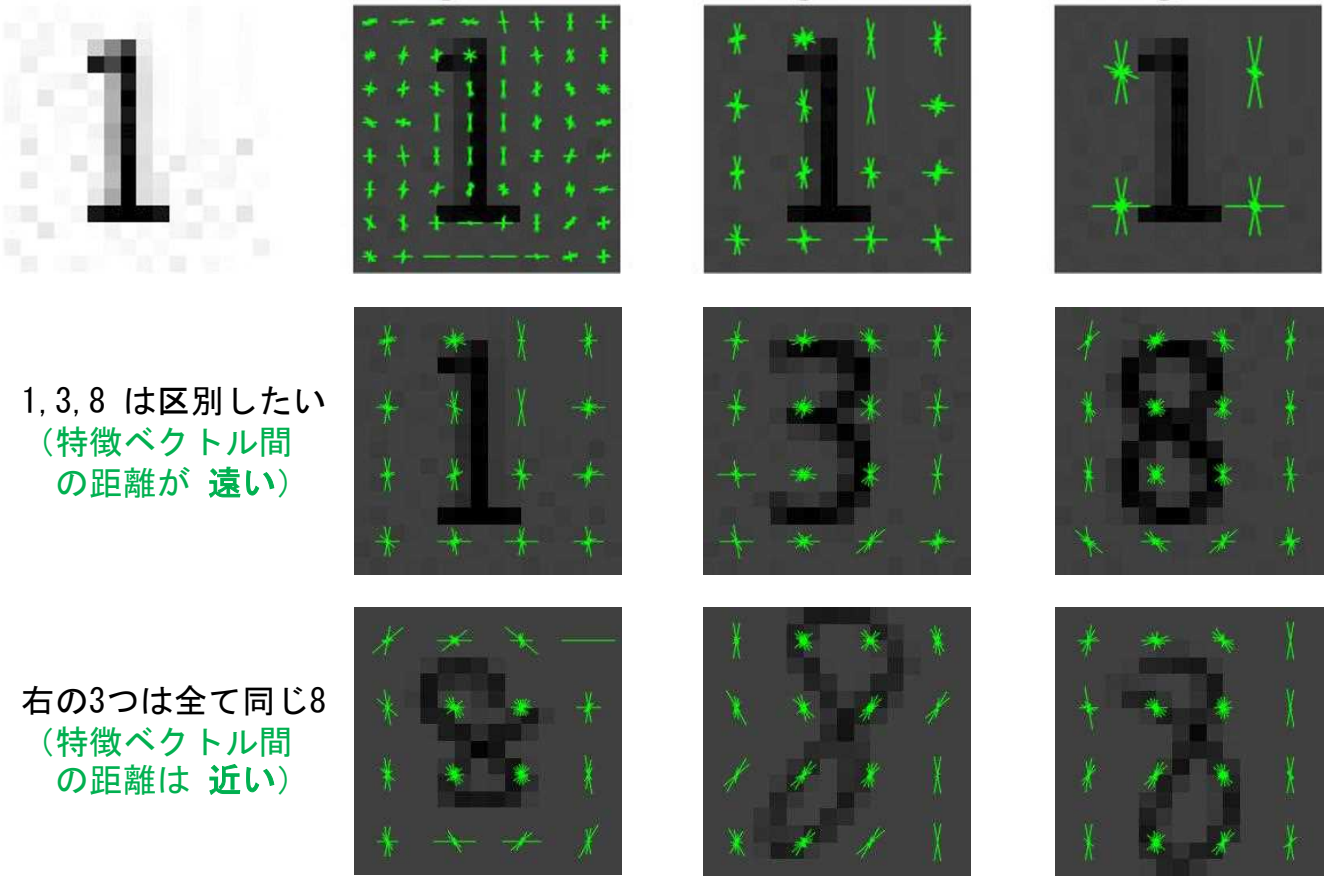
https://jp.mathworks.com/help/vision/ug/digit-classification-using-hog-features.html

---

CellSize = [2 2]
Length = 1764

CellSize = [4 4]
Length = 324

CellSize = [8 8]
Length = 36



1, 3, 8 は区別したい
（特徴ベクトル間
の距離が **遠い**）

右の3つは全て同じ8
（特徴ベクトル間
の距離は **近い**）

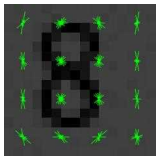
サポートベクターマシン (SVM) で判別する

学習セット

0～9（10種類）の数字それぞれ
につき 101 枚の画像を使って
SVMのパラメータを決定する

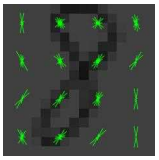
テスト セット

数字ごとに 12 枚のイメージを
判別してSVMの性能を評価する



SVMを学習させてﾊﾟﾗﾒｰﾀを決定

教師信号；
この特徴ベクトルは
「８」である



学習済みのSVMで判別する

機械の出力；
この文字は「８」
であると判別された

機械による判別の精度を評価する

判別の結果

入力した数字	0	1	2	3	4	5	6	7	8	9
0	0.25	0.00	0.08	0.00	0.00	0.00	0.58	0.00	0.08	0.00
1	0.00	0.75	0.00	0.00	0.08	0.00	0.00	0.08	0.08	0.00
2	0.00	0.00	0.67	0.17	0.00	0.00	0.08	0.00	0.00	0.08
3	0.00	0.00	0.00	0.58	0.00	0.00	0.33	0.00	0.00	0.08
4	0.00	0.08	0.00	0.17	0.75	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.33	0.58	0.00	0.08	0.00
6	0.00	0.00	0.00	0.00	0.25	0.00	0.67	0.00	0.08	0.00
7	0.00	0.08	0.08	0.33	0.00	0.00	0.17	0.25	0.00	0.08
8	0.00	0.00	0.00	0.08	0.00	0.00	0.00	0.08	0.67	0.17
9	0.00	0.08	0.00	0.25	0.17	0.00	0.08	0.00	0.00	0.42

・数字 0 が誤って 6 と分類されることが多い
・数字 9 が誤って 3 と分類されている
・何千ものデータセット（MNIST, SVHN など）で学習すると
性能が向上する（今回のデータセットは 1,010 個）

6．オブジェクトの検出と判別

6.1 果実の検出と熟度の判別

6.2 数字や文字の検出と認識

6.3 揺れるカメラ映像の安定化

6.4 深層学習によるオブジェクト検出

揺れるカメラ映像を安定化する

揺れる車中から撮影した
不安定な映像

特徴点のマッチングを適用して
映像を安定化する

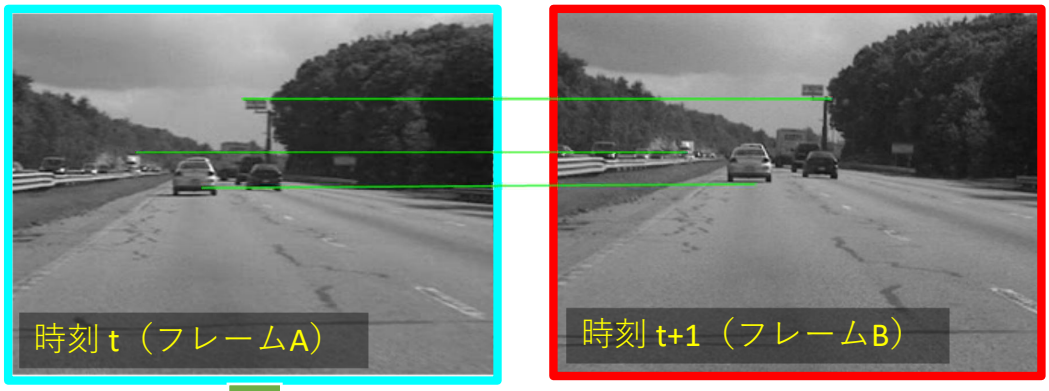
MATLAB: Computer Vision Toolbox

- コーナー検出は detect **FAST** Features を使用
- 特徴ベクトルは FREAK (Fast Retina Keypoint) 記述子
- ベクトル間の対応づけはハミング距離を使用

https://jp.mathworks.com/help/vision/ug/video-stabilization-using-point-feature-matching.html

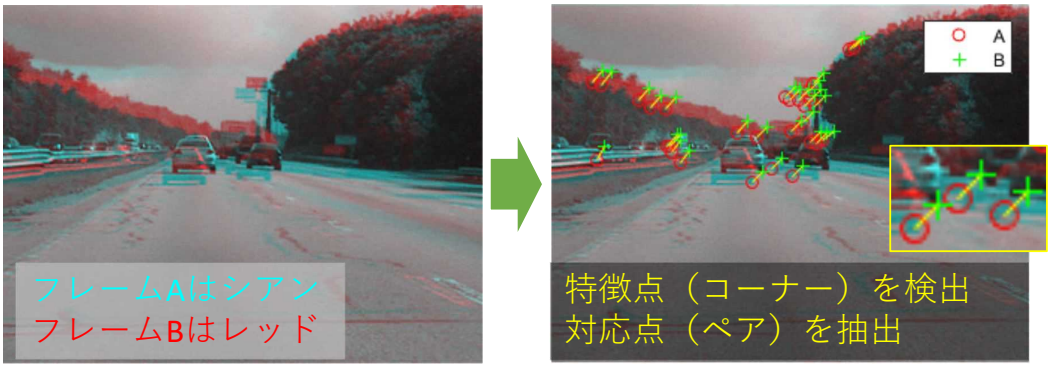
特徴点の位置の違いを検出

揺れの原因は
2つのフレーム
の違い

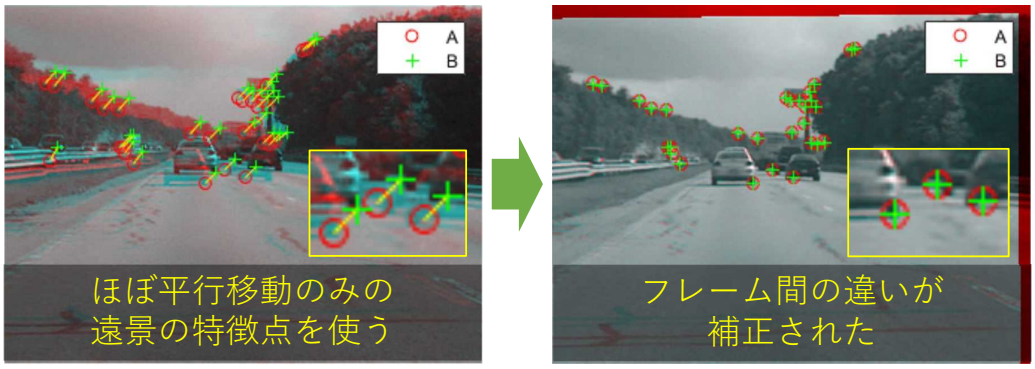


両者の違いを
色で表現

両者の違い
（位置と歪み）
を補正したい



違い（位置と歪み）を補正



M-estimator SAmple Consensus (MSAC)
アルゴリズムにより、フレーム間の
アフィン変換をロバストに推定する

アフィン変換で
位置と歪みを補正

$$\begin{bmatrix} a_1 & a_3 & 0 \\ a_2 & a_4 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

6つのパラメータ

estimateGeometricTransform2D(..,'affine');

$$\begin{bmatrix} \alpha\cos\theta & -\alpha\cos\theta & 0 \\ \alpha\sin\theta & \alpha\cos\theta & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

4つのパラメータ
に当てはめる

平行移動
$t_x = 9.165$
$t_y = -6.95$

$\alpha = 0.9966$
$\theta = 0.0078$

揺れるカメラ映像を安定化する

手ブレの大きな
不安定な映像

特徴点のマッチングを適用して
映像を安定化した

MATLAB: Computer Vision Toolbox

- コーナー検出は detect **FAST** Features を使用
- 特徴ベクトルは FREAK (Fast Retina Keypoint) 記述子
- ベクトル間の対応づけはハミング距離を使用

https://jp.mathworks.com/help/vision/ug/video-stabilization-using-point-feature-matching.html

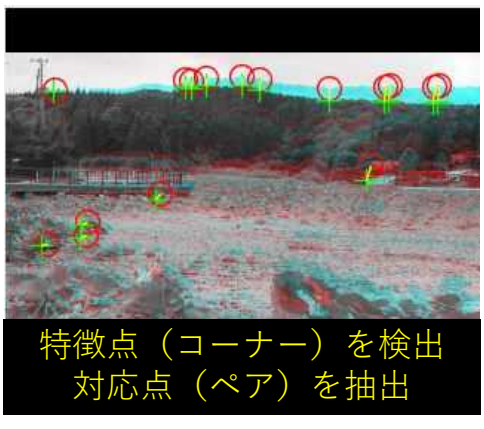
---

特徴点の位置の違いを検出

揺れの原因は
2つのフレーム
の違い

両者の違いを
色で表現

両者の違い
（位置と歪み）
を補正したい



違い（位置と歪み）を補正

対応する特徴点が揃うような
アフィン変換を求める

フレーム間の違いが減って
特徴点の位置が揃った

M-estimator SAmple Consensus (MSAC)
アルゴリズムにより、フレーム間の
アフィン変換をロバストに推定する

アフィン変換で
位置と歪みを補正

$$\begin{bmatrix} a_1 & a_3 & 0 \\ a_2 & a_4 & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

6つのパラメータ

estimateGeometricTransform2D(..,'affine');

$$\begin{bmatrix} \alpha\cos\theta & -\alpha\cos\theta & 0 \\ \alpha\sin\theta & \alpha\cos\theta & 0 \\ t_x & t_y & 1 \end{bmatrix}$$

4つのパラメータ
に当てはめる

平行移動
$t_x = 0.239$
$t_y = -2.11$

$\alpha = 1.001$
$\theta = 2.24°$

MATLAB (Computer Vision Toolbox) の例

オブジェクトの追跡

assignDetectionsToTracks	複数のオブジェクトの追跡のための検出のトラックへの割り当て
configureKalmanFilter	オブジェクトの追跡のためのカルマン フィルターの作成
vision.KalmanFilter	測定値、状態、および状態推定誤差の共分散の修正
vision.HistogramBasedTracker	Histogram-based object tracking
vision.PointTracker	Kanade-Lucas-Tomasi (KLT) アルゴリズムを使用したビデオ内の点の追跡
vision.BlockMatcher	イメージ間またはビデオ フレーム間の動きの推定
vision.TemplateMatcher	イメージ内でのテンプレートの検出

動き推定

opticalFlow	オプティカル フロー行列を格納するオブジェクト
opticalFlowFarneback	Farneback 法を使用してオプティカル フローを推定するオブジェクト
opticalFlowHS	Horn-Schunck 法を使用してオプティカル フローを推定するオブジェクト
opticalFlowLK	Lucas-Kanade 法を使用してオプティカル フローを推定するオブジェクト
opticalFlowLKDoG	Object for estimating optical flow using Lucas-Kanade derivative of Gaussian
vision.BlockMatcher	イメージ間またはビデオ フレーム間の動きの推定
vision.TemplateMatcher	イメージ内でのテンプレートの検出

https://jp.mathworks.com/help/vision/referencelist.html?type=function&s_tid=CRUX_topnav

6．オブジェクトの検出と判別

6.1 果実の検出と熟度の判別

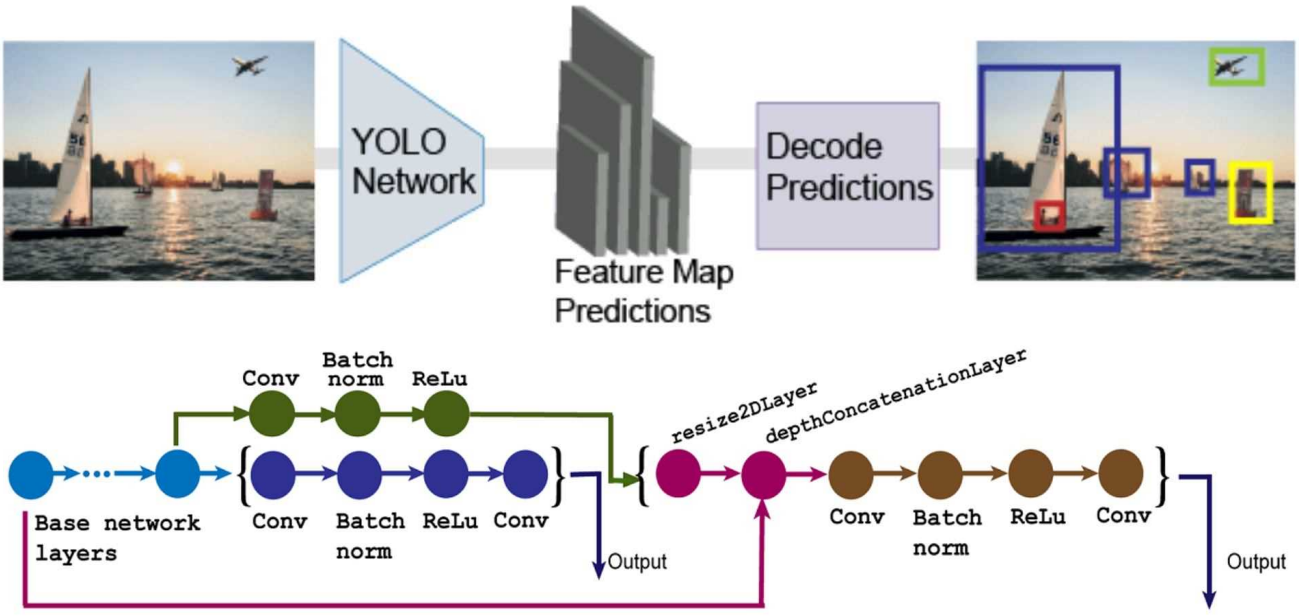
6.2 数字や文字の検出と認識

6.3 揺れる車載カメラ映像の安定化

6.4 深層学習によるオブジェクト検出

何ができる？
MATLABの例

MATLAB
深層学習（YOLO）を使用したオブジェクトの検出



You Only Look Once (**YOLO**) を実装する関数 yolov3ObjectDetector が MATLAB (Computer Vision Toolbox) に用意されている

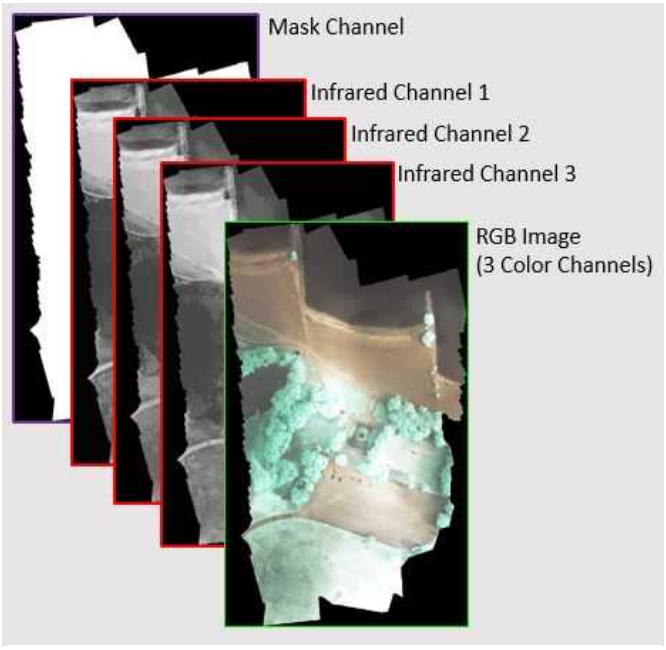
https://jp.mathworks.com/help/deeplearning/ug/object-detection-using-yolo-v2.html

MATLAB
深層学習（Deeplab v3+）によるセグメンテーション

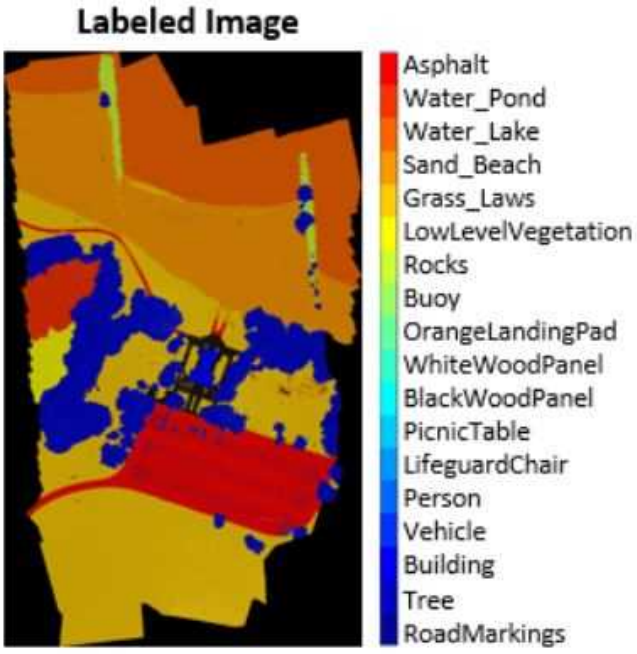
https://jp.mathworks.com/help/vision/ug/semantic-segmentation-using-deep-learning.html

---

深層学習（U-Net）を使用したマルチスペクトル画像のセマンティック セグメンテーション



7 チャネル（カラー 3 ＋近赤外 3 ＋マスク 1）



https://jp.mathworks.com/help/vision/ug/multispectral-semantic-segmentation-using-deep-learning.html

深層学習（SegNet）を使用した
セマンティック セグメンテーション

VGG16 などの事前学習済みネットワークを読み込み、
SegNetLayers コマンドを使用して、
ピクセルレベルのラベル付けを実行できる

https://jp.mathworks.com/solutions/image-video-processing/semantic-segmentation.html

---

MATLAB (Computer Vision Toolbox) の例

オブジェクト検出器の学習

trainRCNNObjectDetector	R-CNN 深層学習オブジェクト検出器の学習
trainFastRCNNObjectDetector	Fast R-CNN 深層学習オブジェクト検出器の学習
trainFasterRCNNObjectDetector	Faster R-CNN 深層学習オブジェクト検出器の学習
trainSSDObjectDetector	Train an SSD deep learning object detector
trainYOLOv2ObjectDetector	Train YOLO v2 object detector

イメージ カテゴリの分類と画像検索

trainImageCategoryClassifier	イメージ カテゴリ分類器の学習
bagOfFeatures	bag of visual words オブジェクト
imageCategoryClassifier	Predict image category
invertedImageIndex	Search index that maps visual words to images
evaluateImageRetrieval	Evaluate image search results
indexImages	Create image search index
retrieveImages	イメージ セットでの類似イメージの検索

https://jp.mathworks.com/help/vision/referencelist.html?type=function&s_tid=CRUX_topnav

国立大学法人
長岡技術科学大学
Nagaoka University of Technology

電気電子情報工学専攻　岩橋政宏
iwahashi@vos.nagaokaut.ac.jp

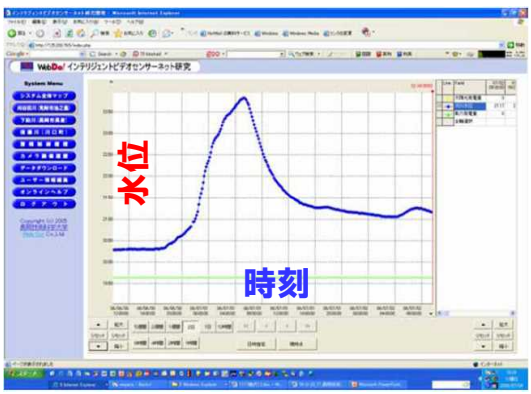
6．オブジェクトの検出と判別

6.0 画像による河川水位の検出

・時間方向の平滑化フィルタ　と
・　横方向の　微分フィルタ　と
・　二値化の
組み合わせによるシンプルな方法

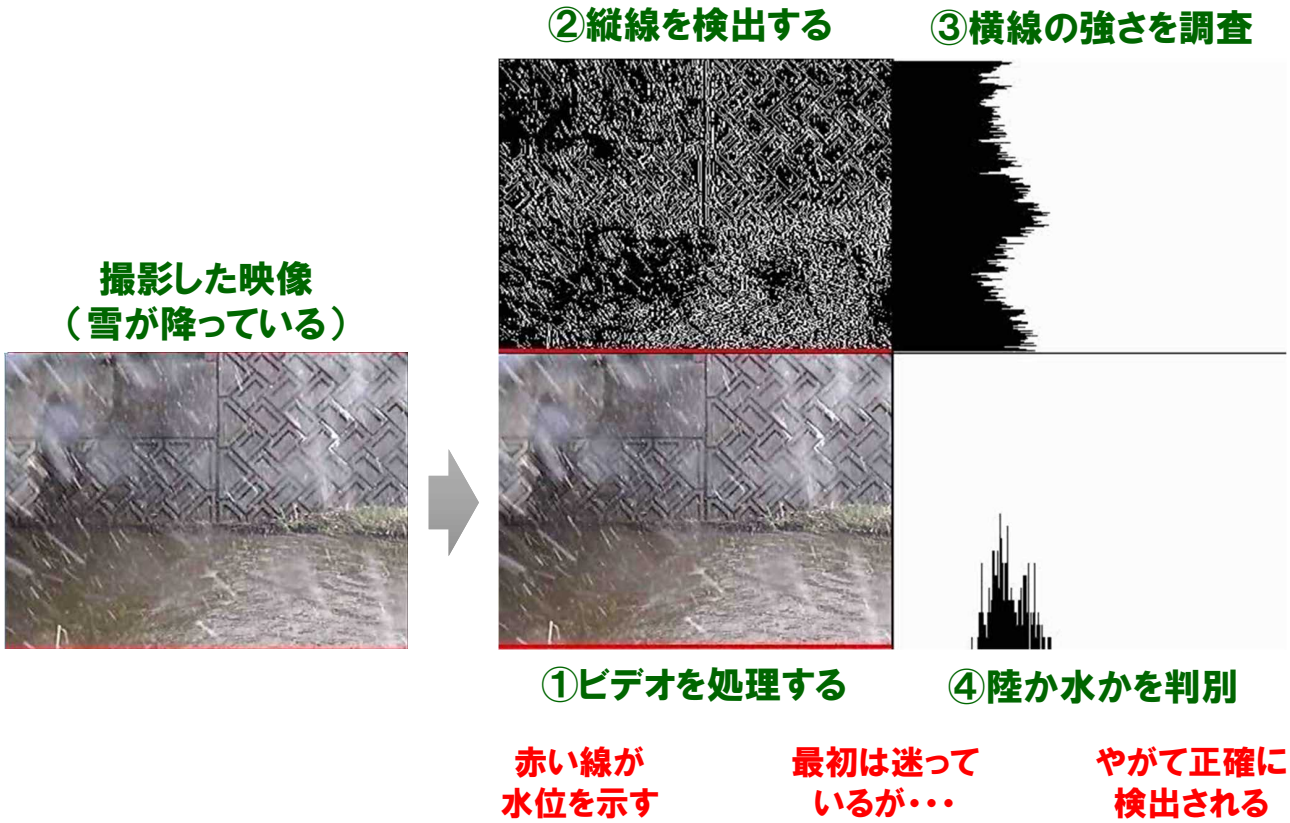
「水位」を 自動で 検出したい

夜も昼もビデオ監視

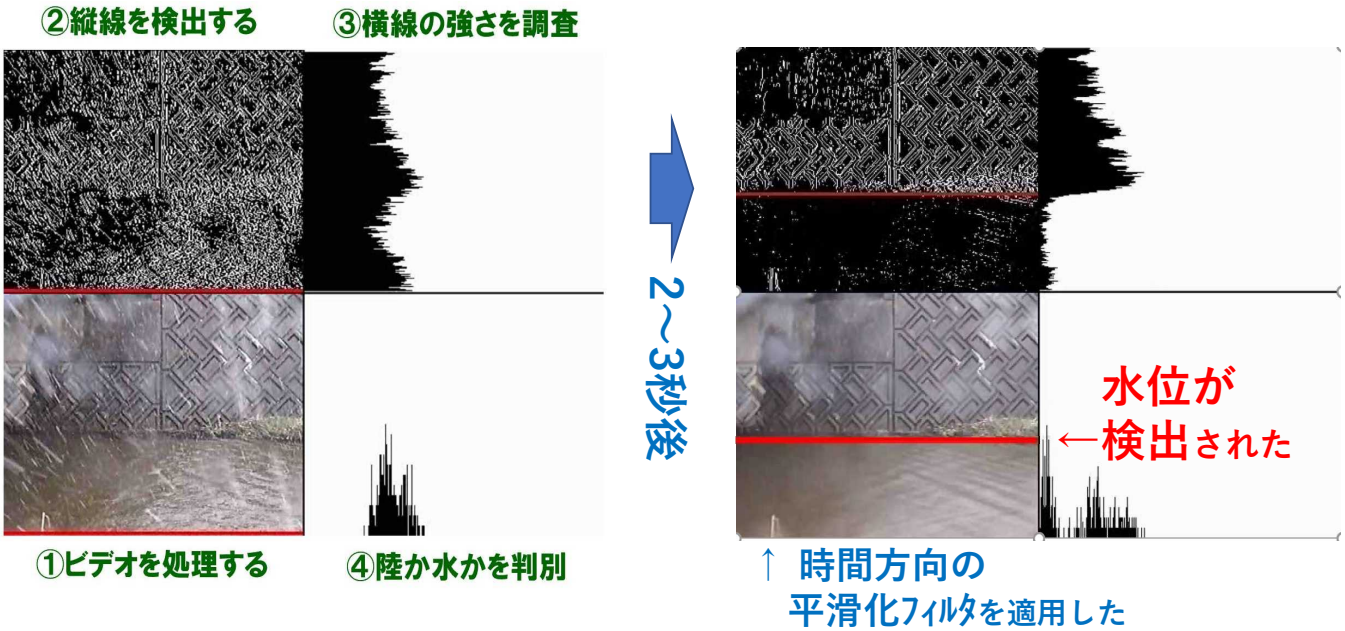


スマートセンサ

雪が降っても 水位を 検出できる

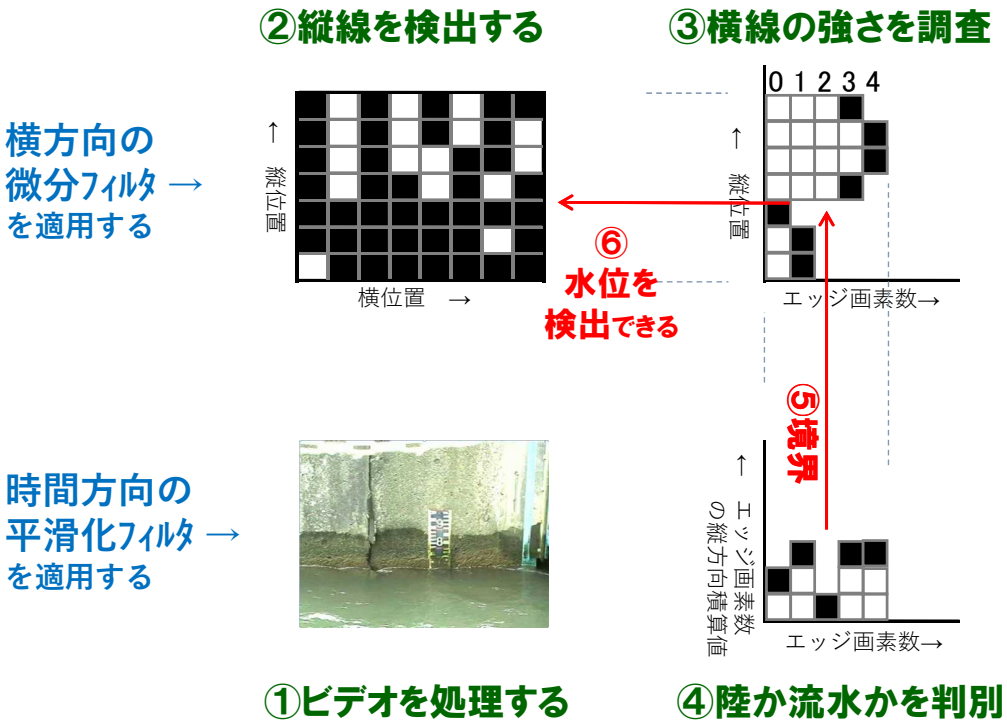


降雪時でも 水位を 検出できる



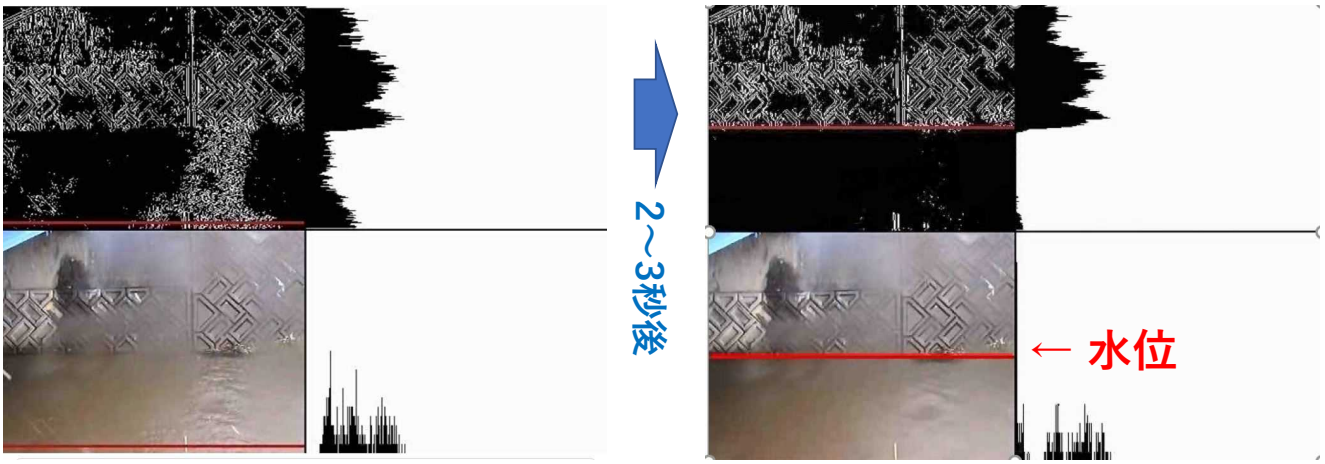
M. IWAHASHI, S. UDOMSIRI, Y. IMAI, Shogo MURAMATSU, “Water Level Detection for Functionally Layered Video Coding”, IEEE International Conference on Image Processing (ICIP), MP-P3.11, vol.II, pp.321-324, Sept. 2007.

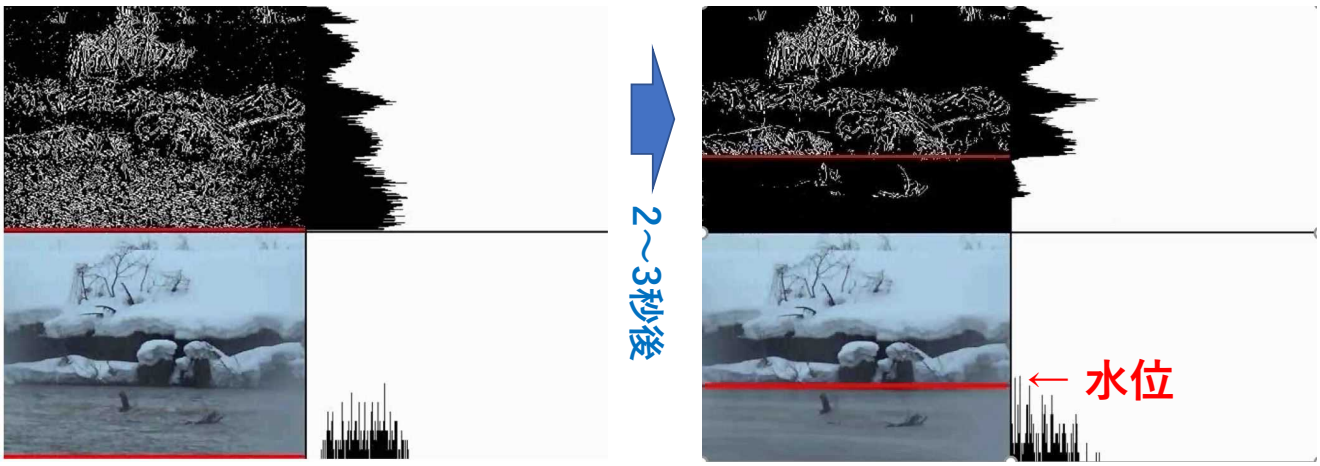
水位検出のアルゴリズム



岩橋、齋藤 「流水領域検出システム、流水領域検出方法、及びプログラム」：特許登録第04910139号, 登録日：2012-01-27,

---





今後の話題

深層学習

・活性化関数、損失関数、誤差逆伝搬

・畳み込みニューラルネットワーク（CNN）

・変分オートエンコーダ（VAE）、敵対的生成ネットワーク（GAN）

機械学習

・サポートベクターマシン、ランダムフォレスト、アダブースト

今後の話題

２次統計量による特徴の抽出と判別

（主成分分析、判別分析、正準相関分析）

回帰分析とスパース推定、3D再構成

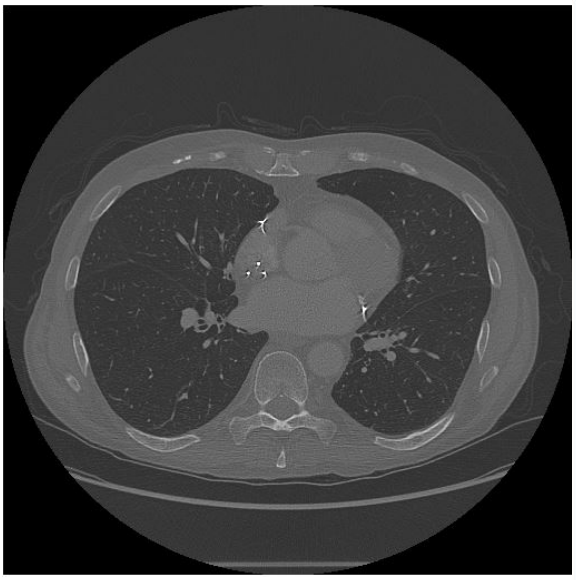
（線形回帰、リッジ正則化、LASSO、ガウス過程回帰）

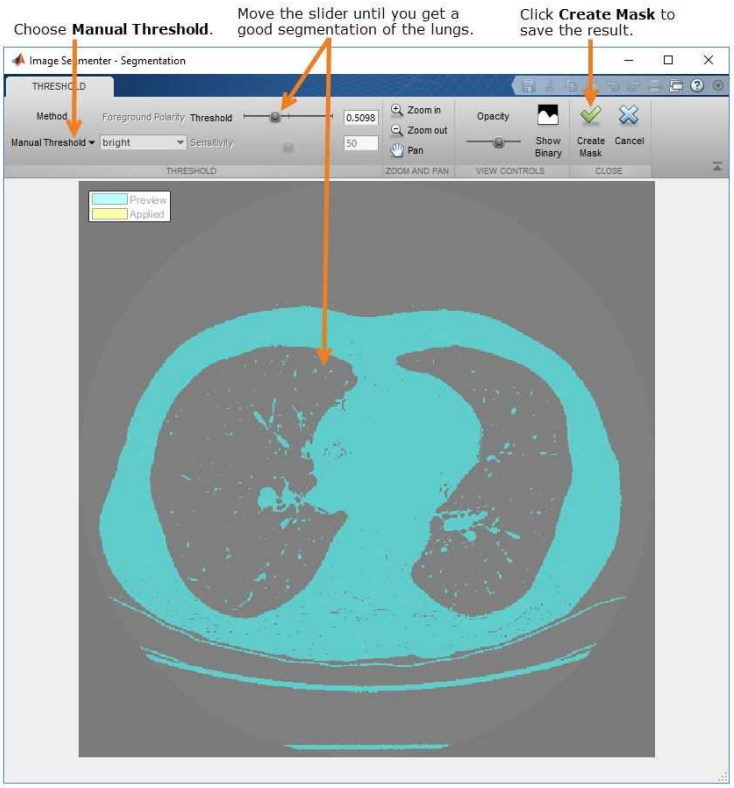
画像のセグメンテーション、クラスタリング、検索

（ﾍﾞｲｽﾞ則、混合ｶﾞｳｽ、ﾛｼﾞｽﾃｨｯｸ判別、SVM, K近傍法）

MATLAB / Snakes

動的輪郭 (snakes) を使用して 3 次元セグメンテーションを実行する

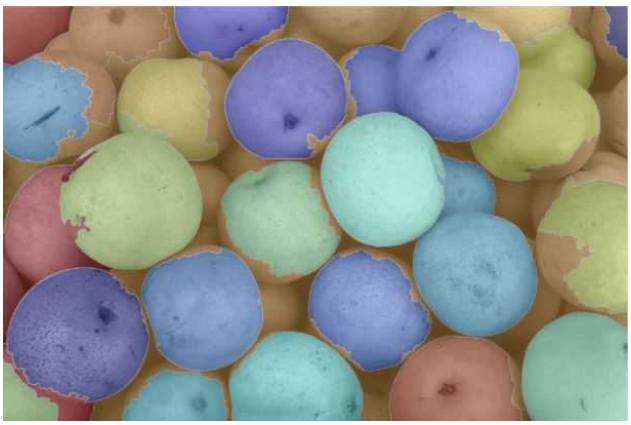


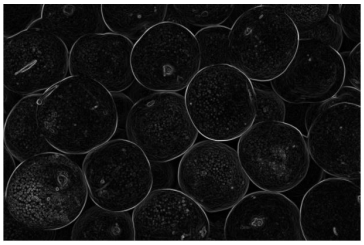


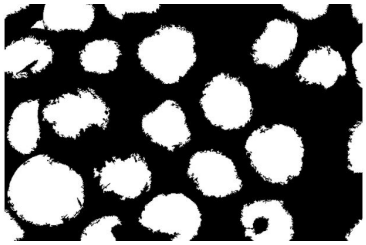
https://jp.mathworks.com/help/images/segment-lungs-from-3-d-chest-mri-data.html

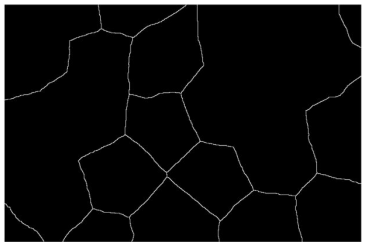
MATLAB / Watershed

watershed セグメンテーションを使用してイメージ内で接触しているオブジェクトを分離する方法





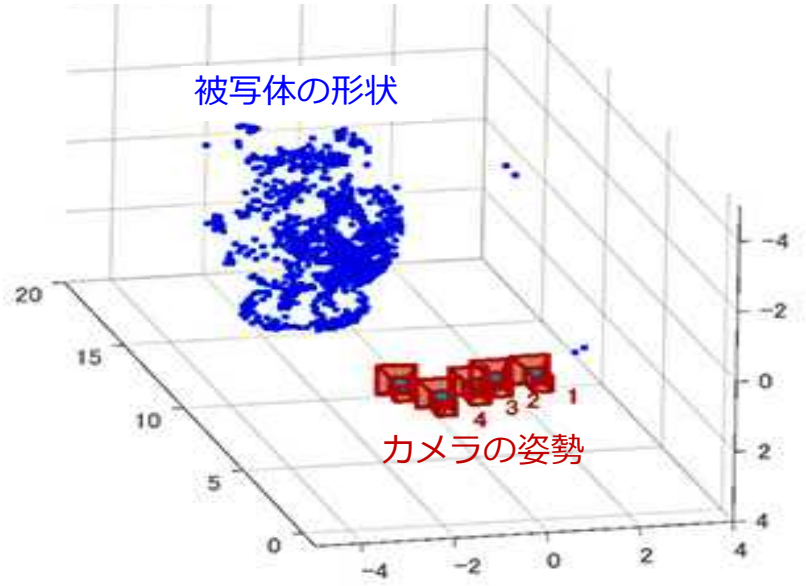




https://jp.mathworks.com/help/images/marker-controlled-watershed-segmentation.html

MATLAB / Structure from Motion (SfM)

視点を動かしながら撮影されたビデオ映像から
カメラの姿勢と被写体の３D形状を推定する

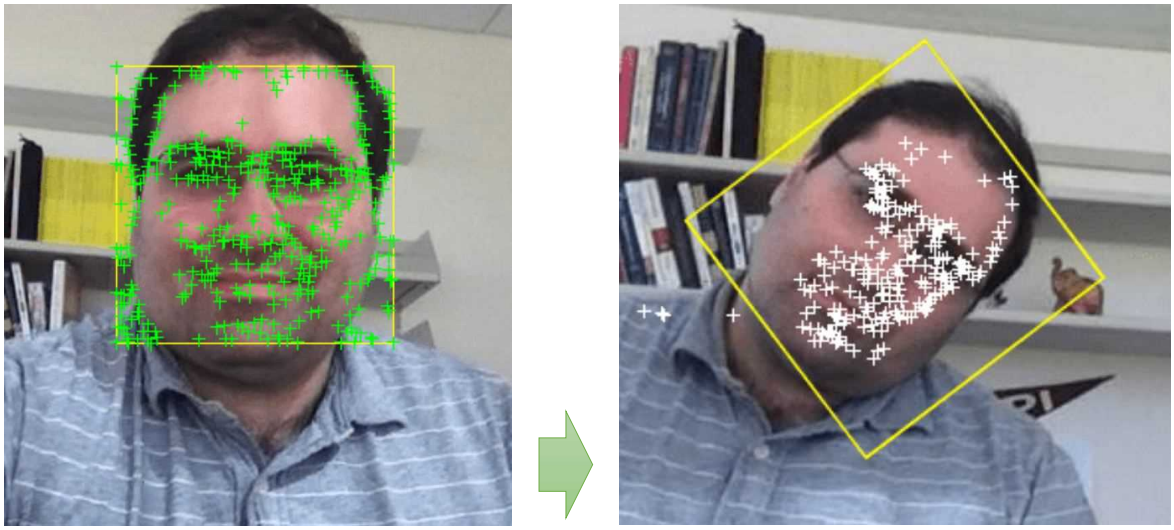


・detect**SURF**Features により特徴点と特徴ベクトルを計算
・matchFeatures により特徴ベクトル間の対応を探索

https://jp.mathworks.com/help/vision/ug/structure-from-motion-from-multiple-views.html

MATLAB / 顔の検出と追跡

顔のパーツを局所的な特徴として検出し、その位置を
追跡し続ける。顔を傾けたりズームしても追跡できる。



・Viola-Jones のｱﾙｺﾞﾘｽﾞﾑで顔のﾊﾟｰﾂ（目や鼻など）を検出
・パーツの判別には学習済みの分類モデルを使用
・Kanade-Lucas-Tomasi (KLT) アルゴリズムで特徴点を追跡

https://jp.mathworks.com/help/vision/ug/face-detection-and-tracking-using-the-klt-algorithm.html

今後の話題

ソフトウェア実装（MATLABプログラム）

ハードウェア実装（MATLABとの連携、Arduino、Jetson）

並列処理による高速化（GPU、CUDA）

応用例（スマートセンサ、遠隔監視で異常通知）