Matlab在图像领域的应用：分类问题

报告人：鄂得俊

导师：李公平(教授)

专业：核技术及其应用

2019年5月6号

大纲

1. Matlab图像处理基础知识
2. 图像处理应用：分类问题

大纲

1. Matlab图像处理基础知识

Matlab图像处理基础知识

Matlab中图像基本操作

读、写和查询

Matlab中图像基本操作

读、写和查询

```
imfinfo('ch1_images\\cameraman.tif')

I1 = imread('ch1_images\\cameraman.tif');  %Read in
the TIF format cameraman image
imwrite(I1,'figures\\cameraman.jpg','jpg');   %Write
the resulting array I1 to disk as a JPEG image

imfinfo('figures\\cameraman.jpg')
```

```
          FileSize: 10717
            Format: 'jpg'
     FormatVersion: ''
             Width: 256
            Height: 256
          BitDepth: 8
         ColorType: 'grayscale'
   FormatSignature: ''
  NumberOfSamples: 1
     CodingMethod: 'Huffman'
    CodingProcess: 'Sequential'
           Comment: {}
```

Matlab图像处理基础知识

1 Matlab中图像基本操作

图像显示

1 Matlab中图像基本操作

图像显示

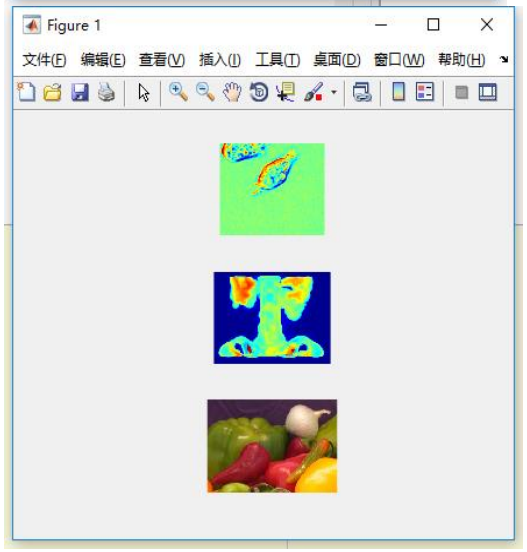
```
A = imread('ch1_images\\cameraman.tif');
 imshow(A);
 imagesc(A);
 axis image;  % Correct aspect ratio of displayed image
 axis off;   % Turn off the axis labelling
 colormap(gray);  % Display intensity image in grey scale
```

1 Matlab中图像基本操作

图像显示

```
A = imread('ch1_images\\cameraman.tif');
imshow(A);
imagesc(A);
axis image;  % Correct aspect ratio of displayed image
axis off;   % Turn off the axis labelling
colormap(gray);  % Display intensity image in grey scale
```


```
% subplot
B = imread('ch1_images\\cell.tif');
C = imread('ch1_images\\spine.tif');
D = imread('ch1_images\\onion.png');
subplot(3,1,1);imagesc(B); axis image;
axis off;colormap(gray);
subplot(3,1,2);imagesc(C); axis image;
axis off;colormap(jet);
subplot(3,1,3);imshow(D);
```



Matlab图像处理基础知识

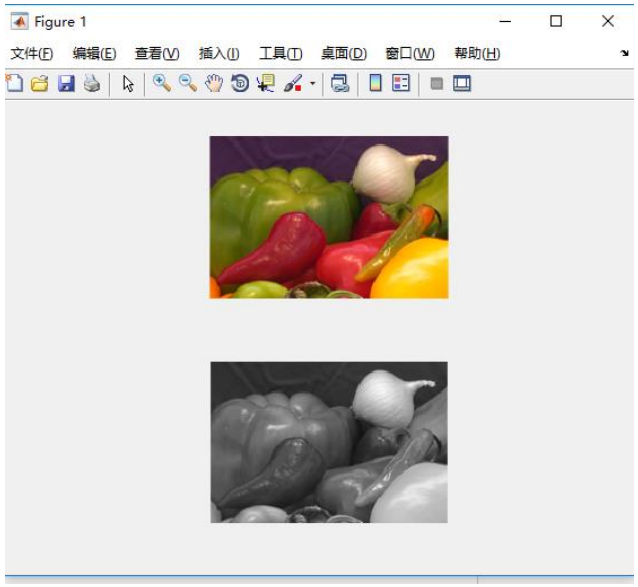
1 Matlab中图像基本操作

图像类型转化

1 Matlab中图像基本操作

图像类型转化

```
% Converting image types
D = imread('ch1_images\\onion.png');
Dgray = rgb2gray(D);
subplot(2,1,1);imshow(D); axis image;
subplot(2,1,2);imshow(Dgray);
```



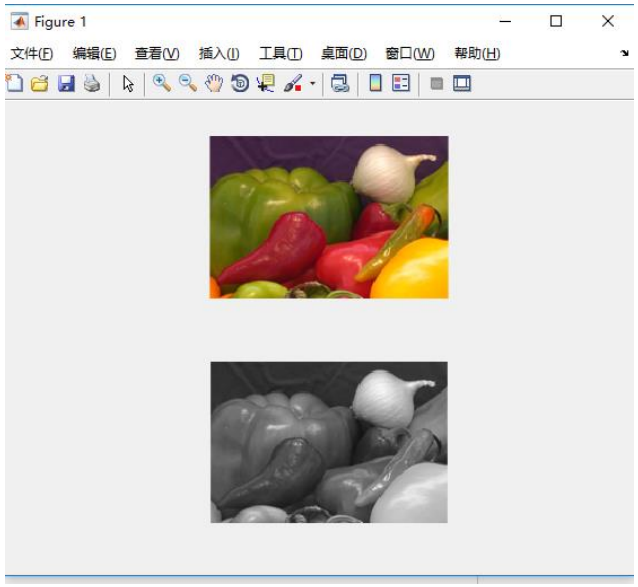
Matlab图像处理基础知识

1 Matlab中图像基本操作

图像类型转化

- 二进制图
- 索引图（伪彩色）
- 灰度图
- RGB图（真彩色）

```
% Converting image types
D = imread('ch1_images\\onion.png');
Dgray = rgb2gray(D);
subplot(2,1,1);imshow(D); axis image;
subplot(2,1,2);imshow(Dgray);
```



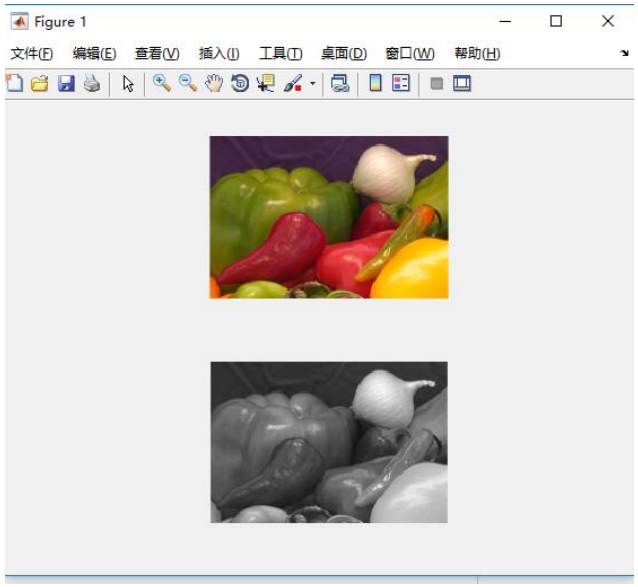
Matlab图像处理基础知识

1 Matlab中图像基本操作

图像类型转化

- 二进制图
- 索引图（伪彩色）
- 灰度图
- RGB图（真彩色）

```
% Converting image types
D = imread('ch1_images\\onion.png');
Dgray = rgb2gray(D);
subplot(2,1,1);imshow(D); axis image;
subplot(2,1,2);imshow(Dgray);
```



转化类型	command
灰度图转换为索引图	[X,map] = gray2ind(I,n)
索引图转换为灰度图	I = ind2gray(X,map)
RGB图转换为灰度图	I = rgb2gray(RGB)
RGB图转换为索引图	[X,map] = rgb2ind(RGB)
索引图转换为RGB图	RGB = ind2rgb(X,map)
阈值法从灰度图产生索引图	X = grayslice(I)
将矩阵转换为灰度图像	I = mat2gray(X,[Xmin Xmax])

大纲

1. Matlab图像处理基础知识
2. 图像处理应用：分类问题

图像处理应用：分类问题

Industrial Inspection—soft drink bottles

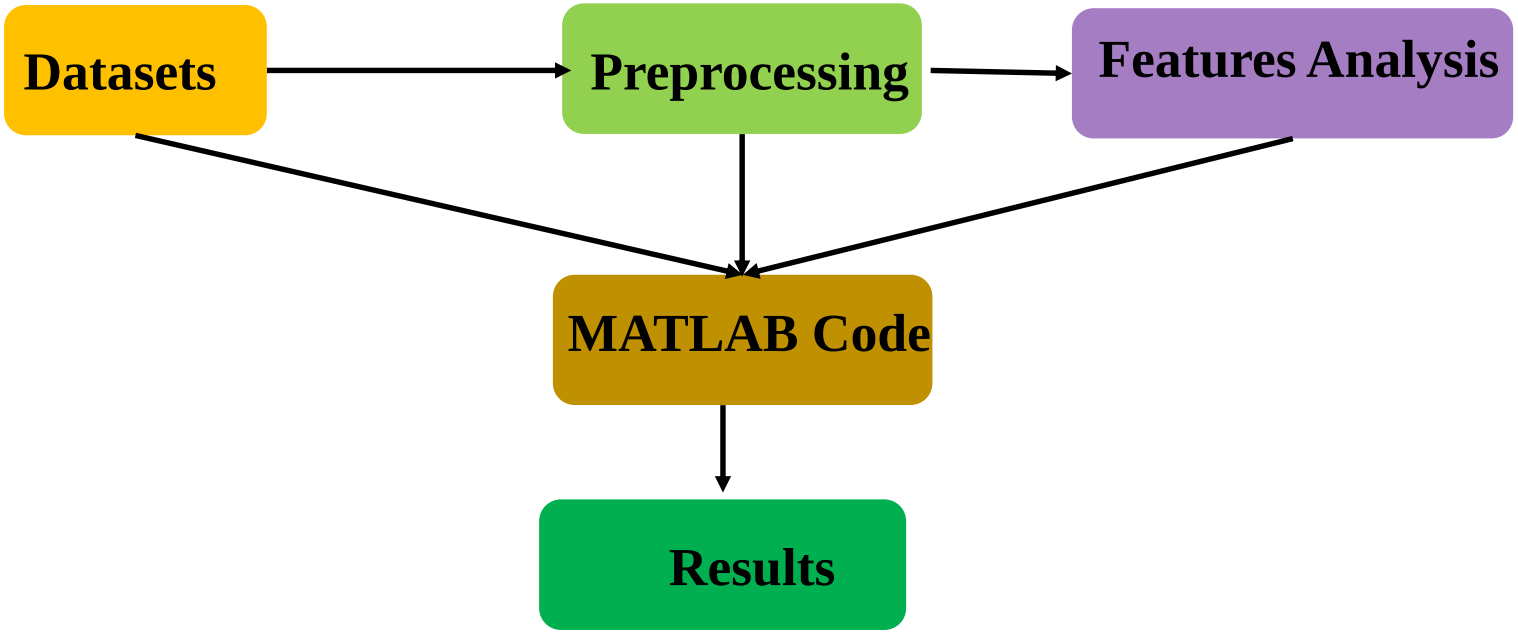
Datasets:http://www.fundipbook.com/

图像处理应用：分类问题

Industrial Inspection—soft drink bottles

Datasets:http://www.fundipbook.com/

Solution



图像处理应用：分类问题

Datasets（141）:http://www.fundipbook.com/

Data View

Normal

Bottle missing

Bottle label not printed

Bottle label not straight

Bottle Overfilled

Bottle deformed

Bottle label missing

Bottle Underfilled

Bottle cap missing

图像处理应用：分类问题

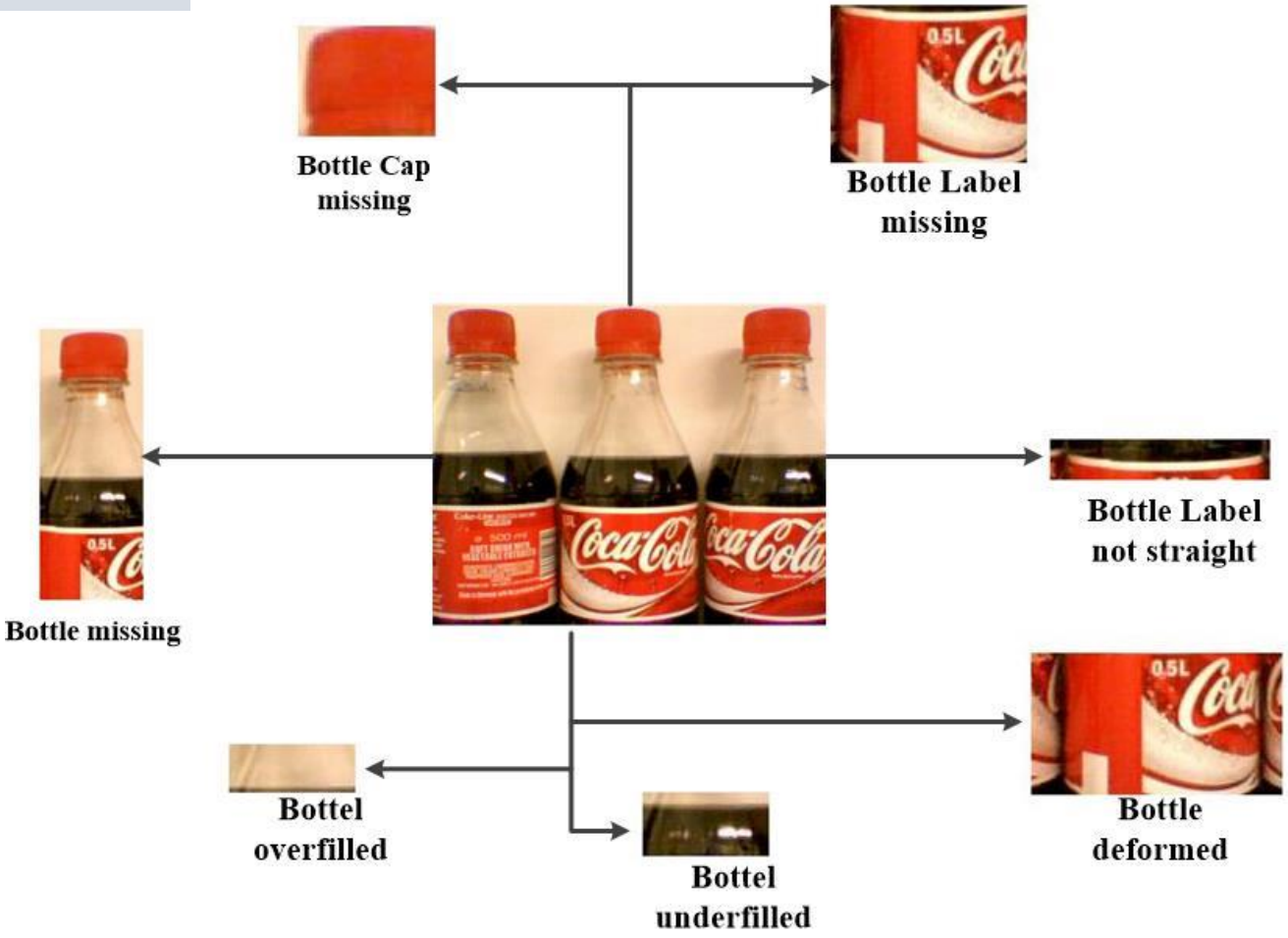
Preprocessing

Features Analysis

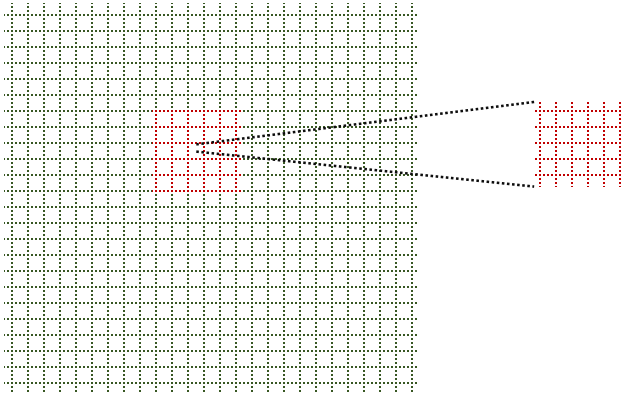
图像处理应用：分类问题

Preprocessing

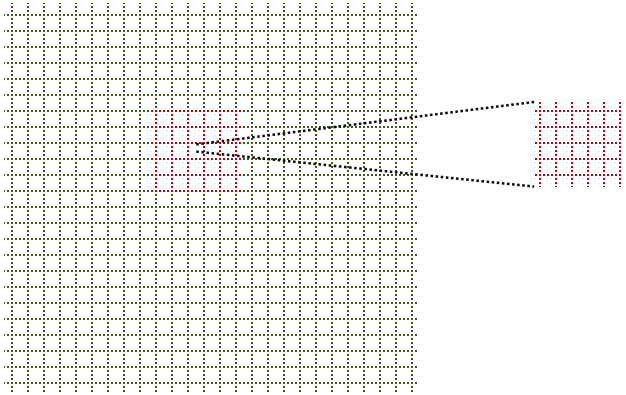
Features Analysis



图像处理应用：分类问题

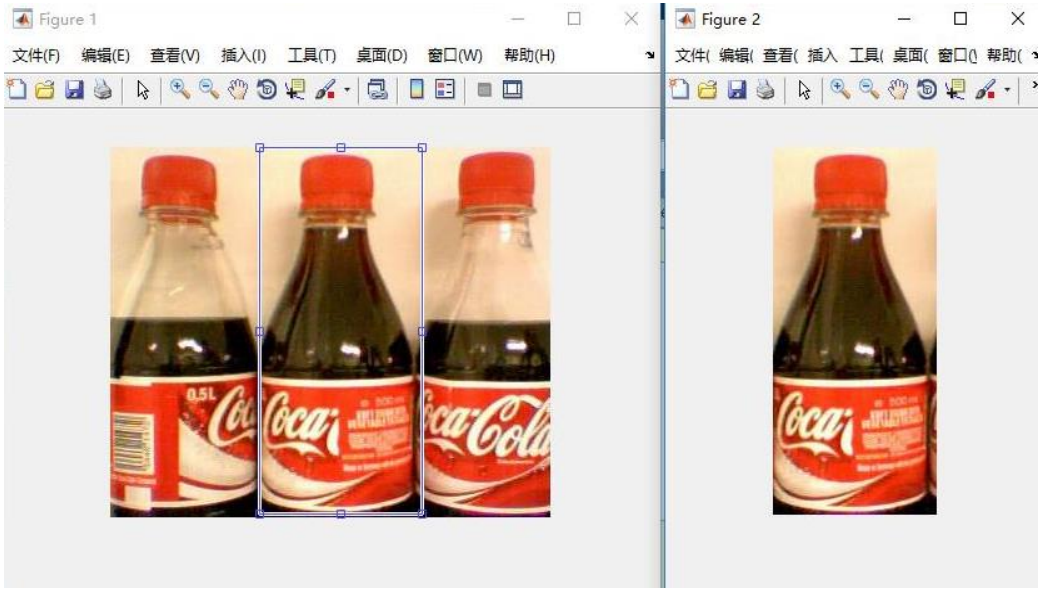


图像处理应用：分类问题

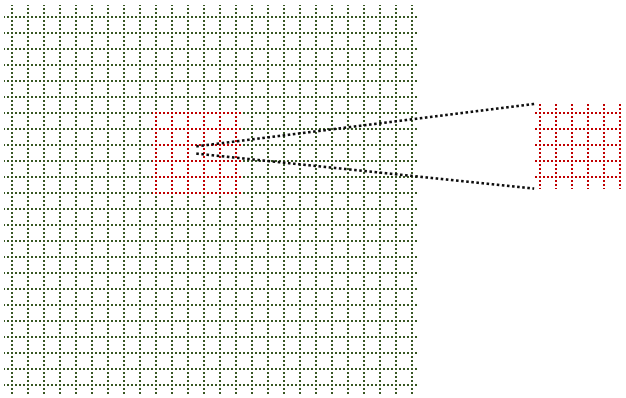


```
image = imread('image001.jpg');
[m,n,z]=size(image);
figure(1)
% image = rgb2gray(image);
imshow(image)

h = imrect;
pos = getPosition(h);
imCp = imcrop(image,pos);
figure(2)
imshow(imCp);
```

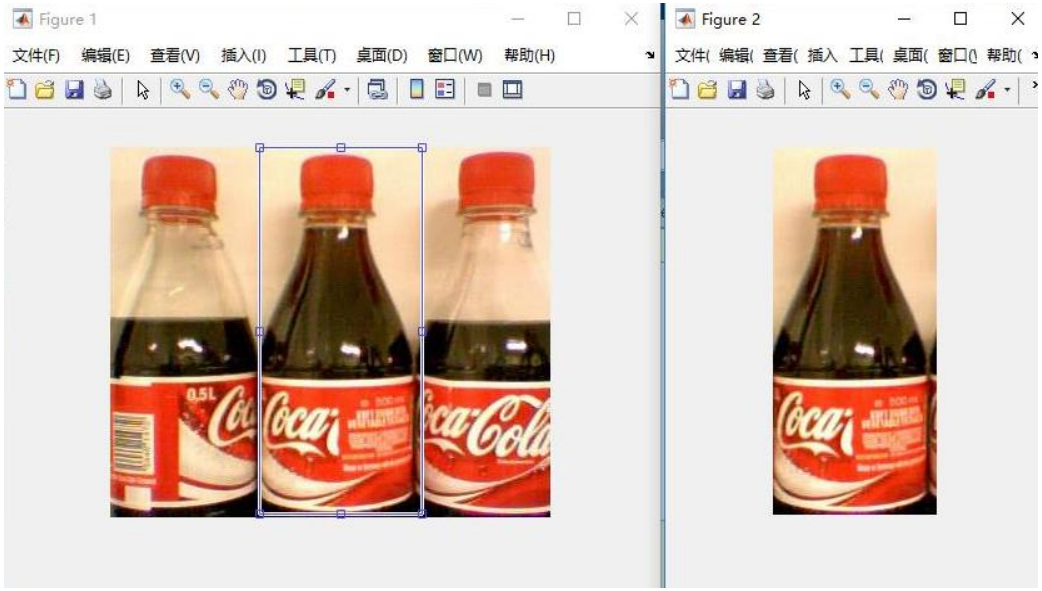


图像处理应用：分类问题



```
image = imread('image001.jpg');
[m,n,z]=size(image);
figure(1)
% image = rgb2gray(image);
imshow(image)

h = imrect;
pos = getPosition(h);
imCp = imcrop(image,pos);
figure(2)
imshow(imCp);
```



pos	xmin	ymin	width	height
value	120	1	130	285

图像处理应用：分类问题

pos	x1	y1	x2	y2
Bottle missing	135	1	225	250
Bottle cap missing	150	5	200	45
Bottle Overfilled	140	110	220	140
Bottle Underfilled	140	130	220	170
Bottle label not printed	110	180	240	280
Bottle label missing	110	180	240	280
Bottle label not straight	110	170	250	195
Bottle deformed	100	190	260	280

图像处理应用：分类问题

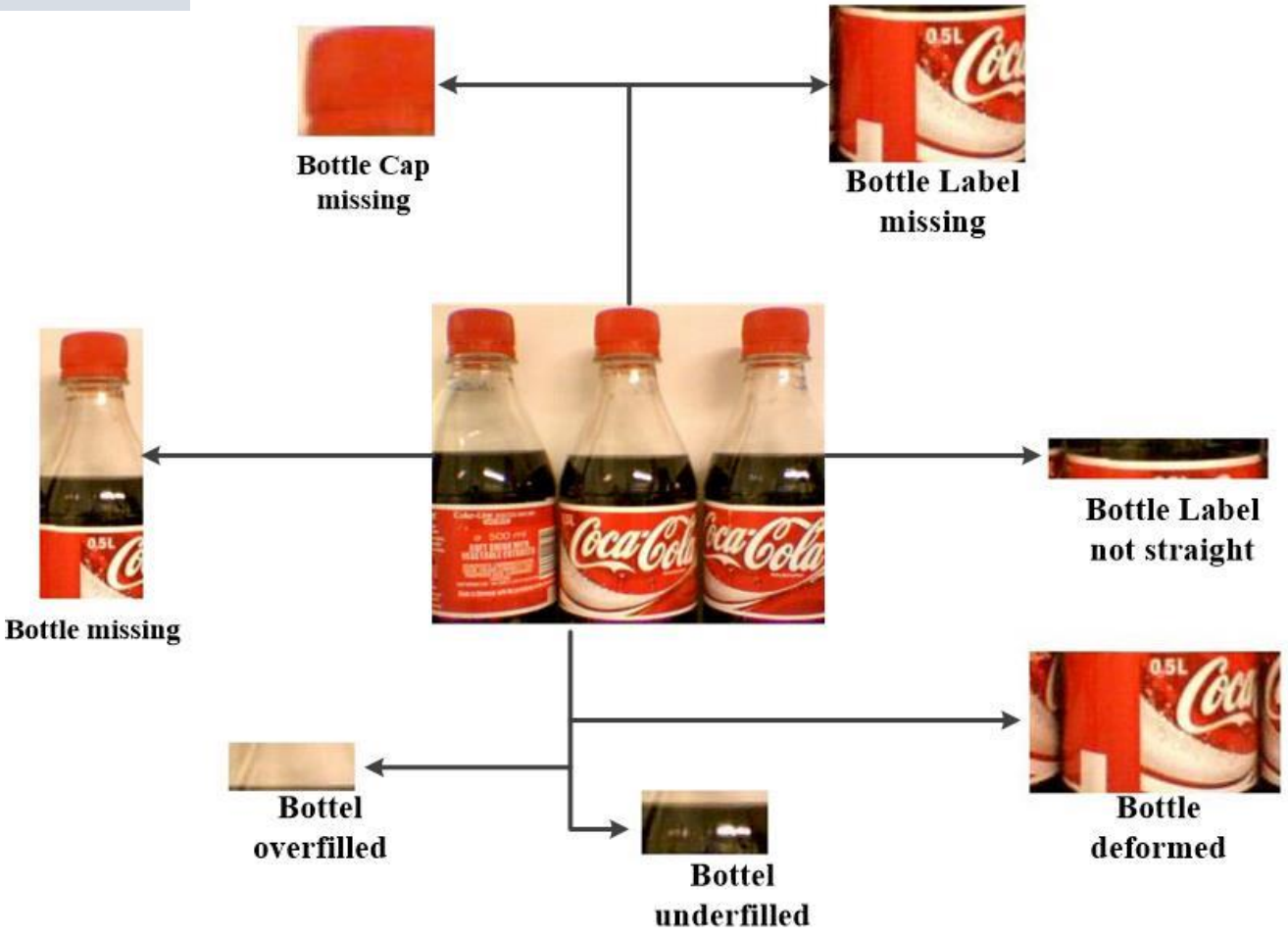
pos	x1	y1	x2	y2
Bottle missing	135	1	225	250
Bottle cap missing	150	5	200	45
Bottle Overfilled	140	110	220	140
Bottle Underfilled	140	130	220	170
Bottle label not printed	110	180	240	280
Bottle label missing	110	180	240	280
Bottle label not straight	110	170	250	195
Bottle deformed	100	190	260	280

```
y1 = 1;
x1 = 135;
y2 = 250;
x2225;
imageOut_origin = image_origin(y1:y2, x1:x2, :);
imshow(imageOut_origin);
```

图像处理应用：分类问题

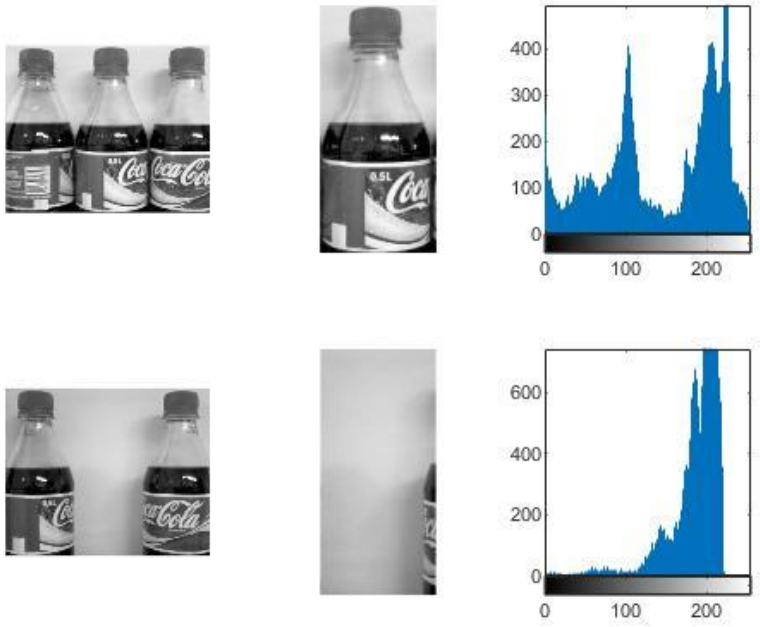
Preprocessing

Features Analysis



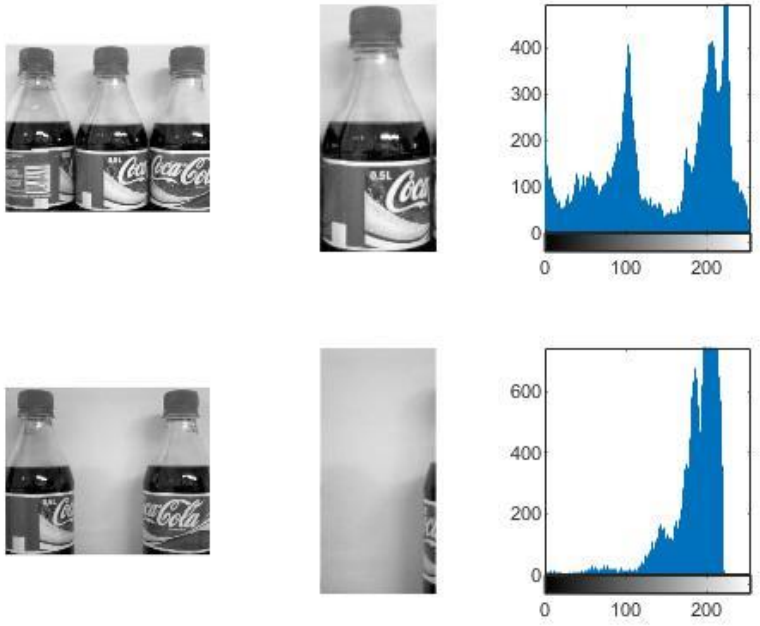
图像处理应用：分类问题

Pixel distributions:histograms



图像处理应用：分类问题

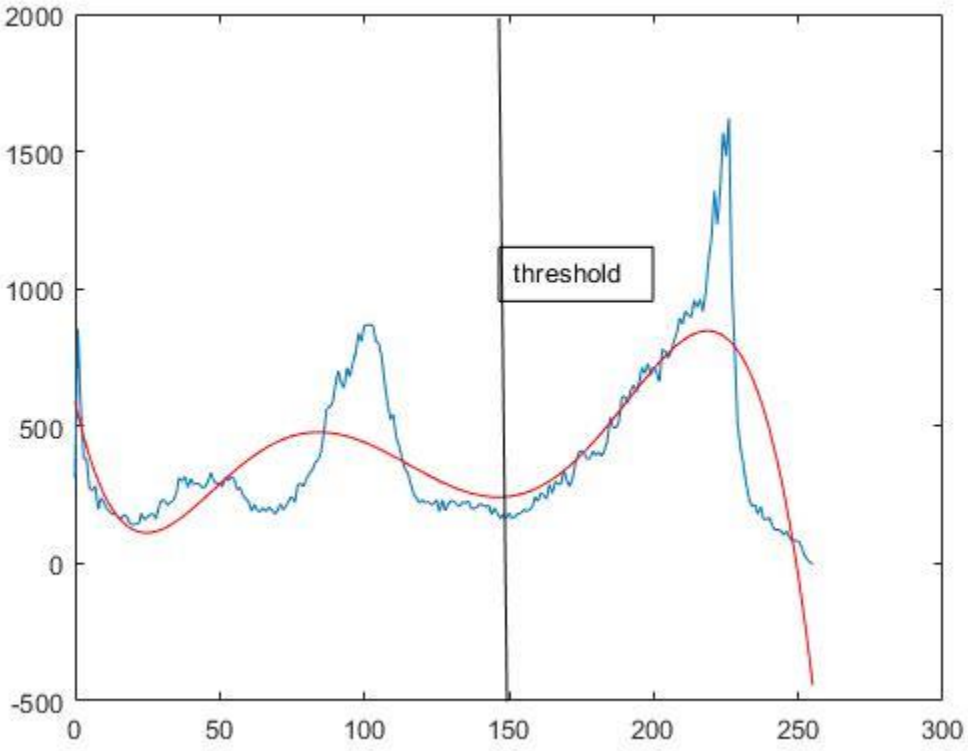
Pixel distributions:histograms




```
%% histogram
image_origin = rgb2gray(imread('image005.jpg'));
image_new = rgb2gray(imread('image018.jpg'));
y1 =3 ;
x1 = 119 ;
y2 = 287;
x2= 252 ;
imageOut_origin = image_origin(y1:y2, x1:x2, :);
imageOut_new = image_new(y1:y2, x1:x2, :);
%blackPercentage = 100 * (sum(imageOut_origin(:) == 0) /
numel(imageOut_origin(:)))
roiBinary_origin = imbinarize(imageOut_origin,
double(150/256));
roiBinary_new = imbinarize(imageOut_new,
double(150/256));
blackPercentage = 100 * (sum(imageOut_new(:) == 0) /
numel(imageOut_new(:)))
subplot(2,3,1),imshow(image_origin);
subplot(2,3,2),imshow(imageOut_origin);
subplot(2,3,3),imhist(imageOut_origin);
subplot(2,3,4),imshow(image_new);
subplot(2,3,5),imshow(imageOut_new);
subplot(2,3,6),imhist(imageOut_new);
```

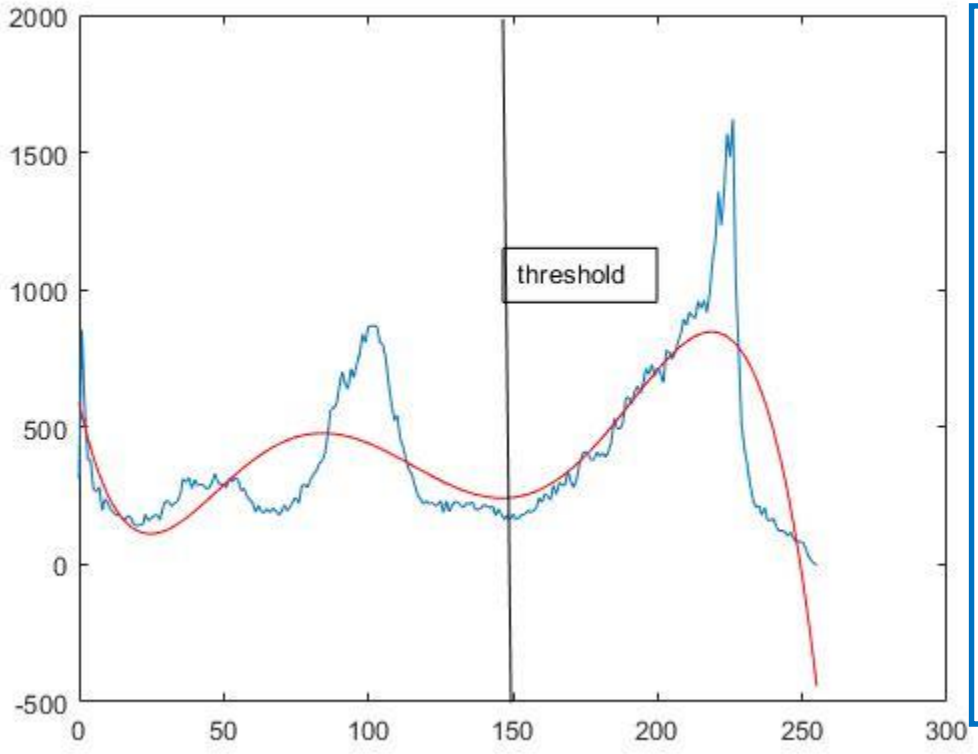
图像处理应用：分类问题

Histogram based thresholding method



图像处理应用：分类问题

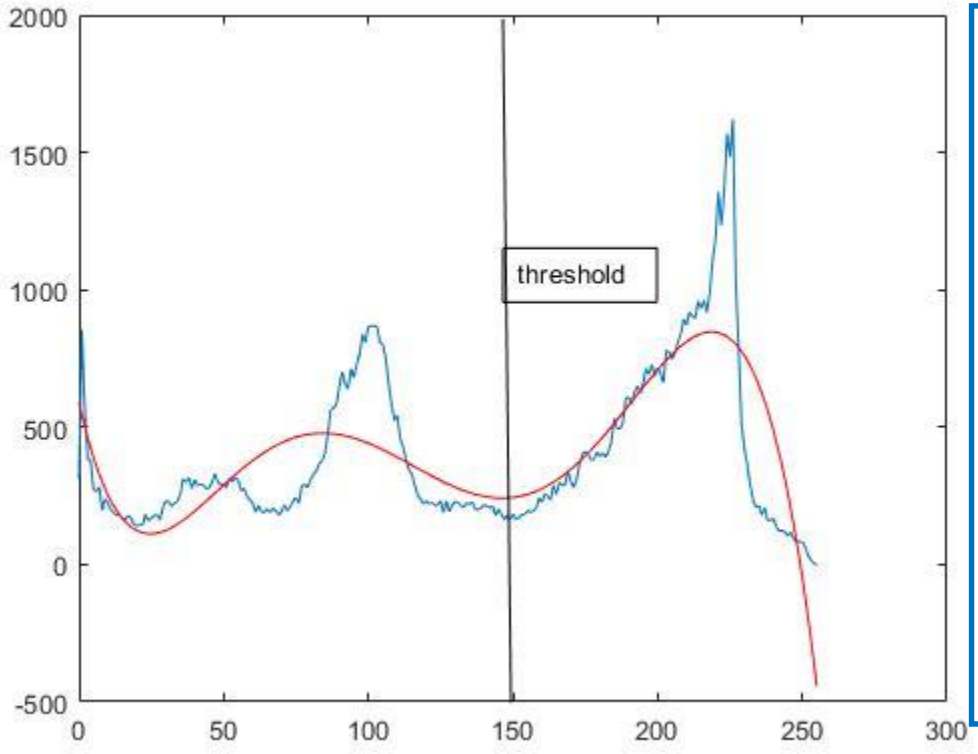
Histogram based thresholding method




```
%% Intensity thresholding
image_origin = rgb2gray(imread('image005.jpg'));
image_new = rgb2gray(imread('image018.jpg'));
y1 =3 ;
x1 = 119 ;
y2 = 287;
x2= 252 ;
imageOut_origin = image_origin(y1:y2, x1:x2, :);
imageOut_new = image_new(y1:y2, x1:x2, :);
[counts_old,X_old]=imhist(image_origin);
[counts_new,X_new]=imhist(image_new);
P = polyfit(X_old,counts_old,6); Y = polyval(P,X_old);
figure;
plot(X_old,counts_old);
hold on,plot(X_old,Y,'r');
```

图像处理应用：分类问题

Histogram based thresholding method




```
%% Intensity thresholding
image_origin = rgb2gray(imread('image005.jpg'));
image_new = rgb2gray(imread('image018.jpg'));
y1 =3 ;
x1 = 119 ;
y2 = 287;
x2= 252 ;
imageOut_origin = image_origin(y1:y2, x1:x2, :);
imageOut_new = image_new(y1:y2, x1:x2, :);
[counts_old,X_old]=imhist(image_origin);
[counts_new,X_new]=imhist(image_new);
P = polyfit(X_old,counts_old,6); Y = polyval(P,X_old);
figure;
plot(X_old,counts_old);
hold on,plot(X_old,Y,'r');
```

1. **Balanced histogram thresholding**
2. Ostu’s method
3. **Iterative Selection Threshold Method**

图像处理应用：分类问题

Ostu's method

Algorithm

1.Compute histogram and probabilities of each intensity level

2.Set up initial $\omega_i(0)$and $\mu_i(0)$

3.Set through all possible thresholds $t = 1.....$ maximum intensity

1.Update ω_i and μ_i

2.Compute $\sigma_b^2(t)$

4.Desired threshold corresponds to the maximum $\sigma_b^2(t)$

图像处理应用：分类问题

Ostu's method

Algorithm

1.Compute histogram and probabilities of each intensity level

2.Set up initial $\omega_i(0)$and $\mu_i(0)$

3.Set through all possible thresholds $t = 1.....$ maximum intensity

1.Update ω_i and μ_i

2.Compute $\sigma_b^2(t)$

4.Desired threshold corresponds to the maximum $\sigma_b^2(t)$

```
function level = otsu_new(histogramCounts)
total = sum(histogramCounts); % '''total''' is the number of pixels in the given image.
%% OTSU automatic thresholding
top = 256;
sumB = 0;
wB = 0;
maximum = 0.0;
sum1 = dot(0:top-1, histogramCounts);
for ii = 1:top
   wF = total - wB;
   if wB > 0 && wF > 0
      mF = (sum1 - sumB) / wF;
      val = wB * wF * ((sumB / wB) - mF) * ((sumB / wB) - mF);
      if ( val >= maximum )
         level = ii;
         maximum = val;
      end
   end
   wB = wB + histogramCounts(ii);
   sumB = sumB + (ii-1) * histogramCounts(ii);
end
end
```

图像处理应用：分类问题

Ostu's method

Algorithm

1.Compute histogram and probabilities of each intensity level

2.Set up initial $\omega_i(0)$and $\mu_i(0)$

3.Set through all possible thresholds $t = 1.....$ maximum intensity

1.Update ω_i and μ_i

2.Compute $\sigma_b^2(t)$

4.Desired threshold corresponds to the maximum $\sigma_b^2(t)$

```
[counts_old,X_old]=imhist(image_origin);
level = ostu_new(counts_old);
```

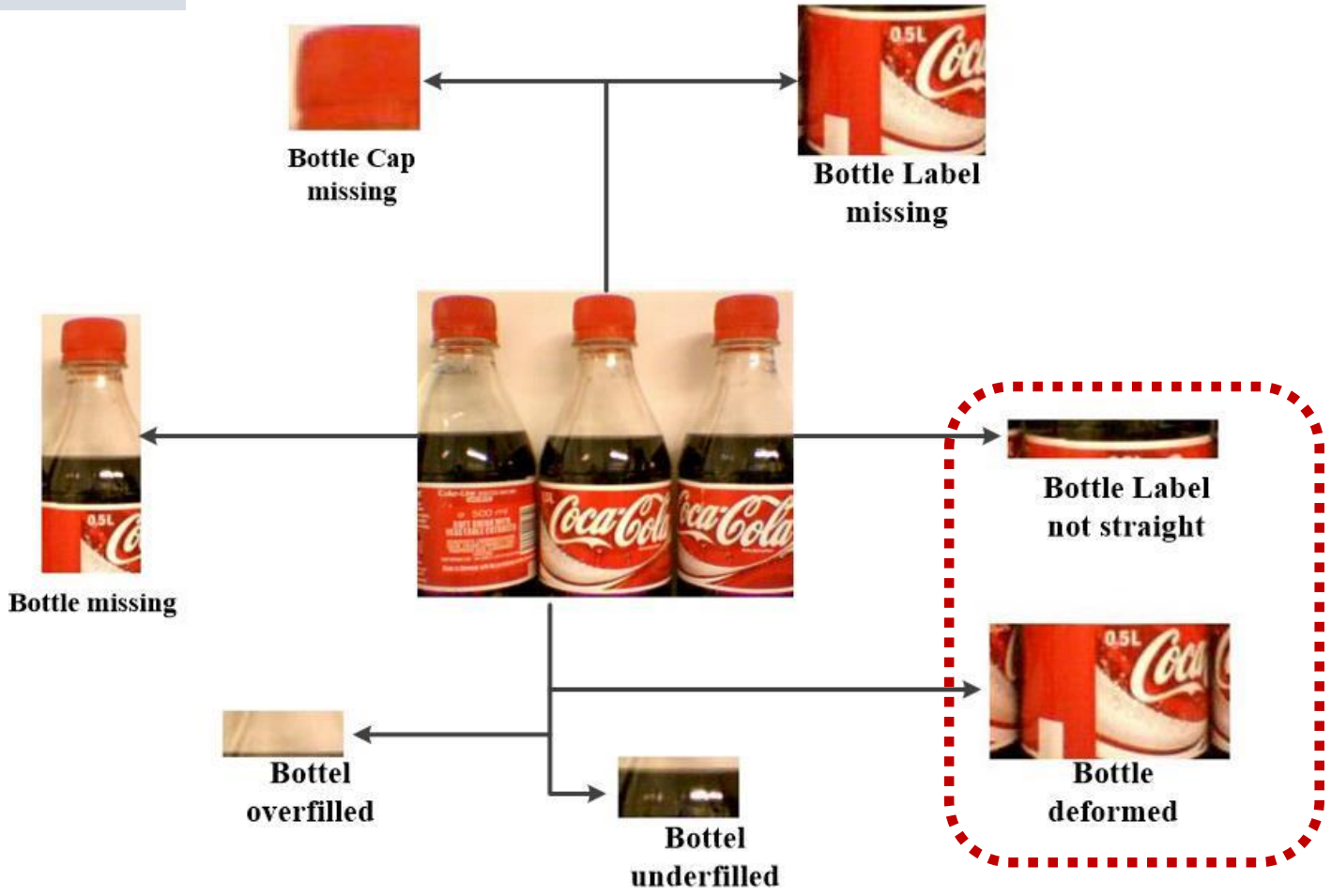
level=140

```
function level = otsu_new(histogramCounts)
total = sum(histogramCounts); % '''total''' is the number of pixels in the given image.
%% OTSU automatic thresholding
top = 256;
sumB = 0;
wB = 0;
maximum = 0.0;
sum1 = dot(0:top-1, histogramCounts);
for ii = 1:top
   wF = total - wB;
   if wB > 0 && wF > 0
      mF = (sum1 - sumB) / wF;
      val = wB * wF * ((sumB / wB) - mF) * ((sumB / wB) - mF);
      if ( val >= maximum )
         level = ii;
         maximum = val;
      end
   end
   wB = wB + histogramCounts(ii);
   sumB = sumB + (ii-1) * histogramCounts(ii);
end
end
```

图像处理应用：分类问题

Preprocessing

Features Analysis



图像处理应用：分类问题

Bottle deformed

图像处理应用：分类问题

Bottle deformed

图像分割

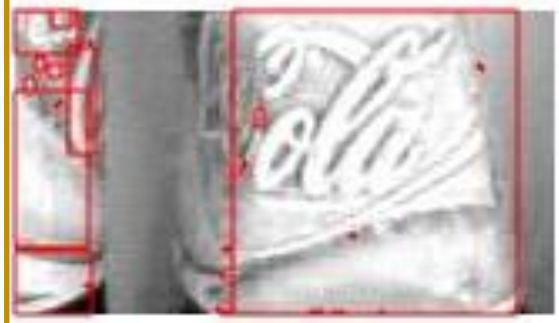
```
%----------------------------------
image_origin = imread('image005.jpg');
image_new = imread('image040.jpg');
image_origin = image_origin(:, :, 1);
image_origin = imadjust(image_origin);
image_new = image_new(:, :, 1);
image_new = imadjust(image_new);
y1 =190 ;
x1 = 100 ;
y2 = 280;
x2= 260 ;
imageOut_origin = image_origin(y1:y2, x1:x2, :);
imageOut_new = image_new(y1:y2, x1:x2, :);
maskR = imbinarize(imageOut_origin, double(200/256));
subplot(1,2,1),imshow(imageOut_origin);
subplot(1,2,2),imshow(imageOut_new);
 cc = bwconncomp(maskR, 4);
 measurements = regionprops(cc, 'area','BoundingBox');
 areas = [measurements.Area];
 rects = cat(1,measurements.BoundingBox);
 % ÏÔÊ¾ËùÓÐÁ¬Í¨ÇøÓò
 figure(2)
 imshow(imageOut_new);
 size(rects,1)
 for i=1:size(rects,1)
    rectangle('position',rects(i,:),'EdgeColor','r')
 end
 % ÏÔÊ¾×î´óÁ¬Í¨ÇøÓò
 [~,max_id] = max(areas);
 max_rect = rects(max_id,:);
 figure(3)
 imshow(imageOut_new);
 rectangle('position',max_rect,'EdgeColor','b');
 %measurements(3).BoundingBox
 length(measurements)
```

图像处理应用：分类问题

Bottle deformed

Origin image

Deformed image



图像处理应用：分类问题

Bottle label not straight

图像处理应用：分类问题

Bottle label not straight

边缘检测

图像分割

```
%% Edge detections of objects
% Carry out edge detection on the ROI
image_origin = imread('image005.jpg');
image_new = imread('image006.jpg');
image_origin = rgb2gray(image_origin);
image_origin = imadjust(image_origin);
image_new = rgb2gray(image_new);
image_new = imadjust(image_new);
y1 =170 ;
x1 = 110 ;
y2 = 195;
x2= 250 ;
imageOut_origin = image_origin(y1:y2, x1:x2, :);
imageOut_new = image_new(y1:y2, x1:x2, :);
[bw, t] = edge(imageOut_origin, 'Sobel');
roiEdge = edge(imageOut_origin, t*0.75);

% Find connected components and get info 'measurements' about each one
cc = bwconncomp(roiEdge);
measurements = regionprops(cc,'area', 'BoundingBox');
areas = [measurements.Area];
maxWidth = 0; maxHeight = 0;
subplot(1,2,1),imshow(imageOut_origin);
subplot(1,2,2),imshow(imageOut_new);
length(measurements)
rects = cat(1,measurements.BoundingBox);
 % ÏÔÊ¾ËùÓÐÁ¬Í¨ÇøÓò
 figure(2)
 imshow(imageOut_origin);
 size(rects,1)
 for i=1:size(rects,1)
    rectangle('position',rects(i,:),'EdgeColor','r')
 end

 [~,max_id] = max(areas);
 max_rect = rects(max_id,:);
 figure(3)
```

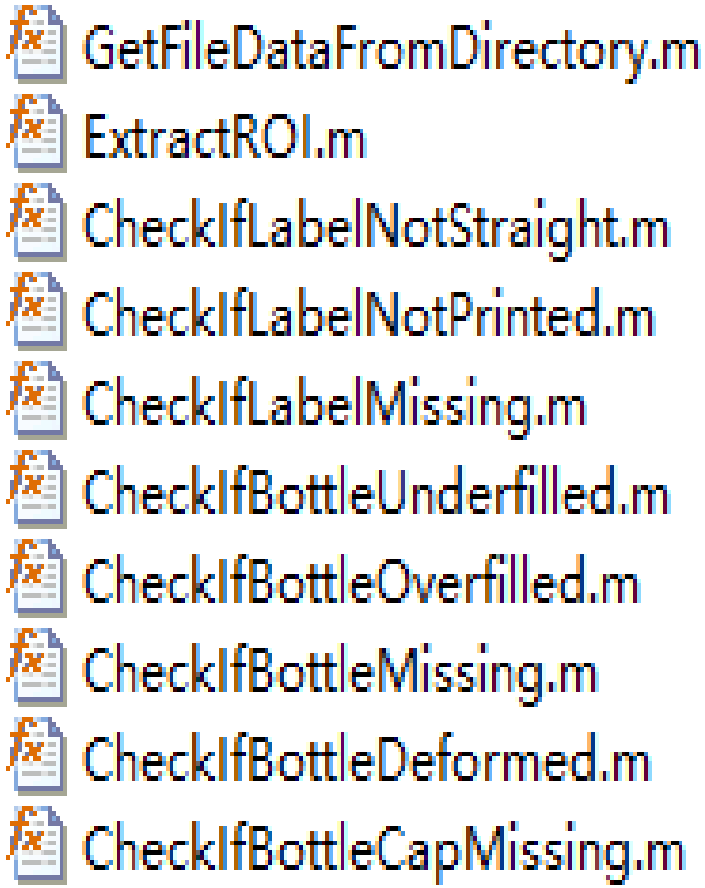
图像处理应用：分类问题

Bottle label not straight

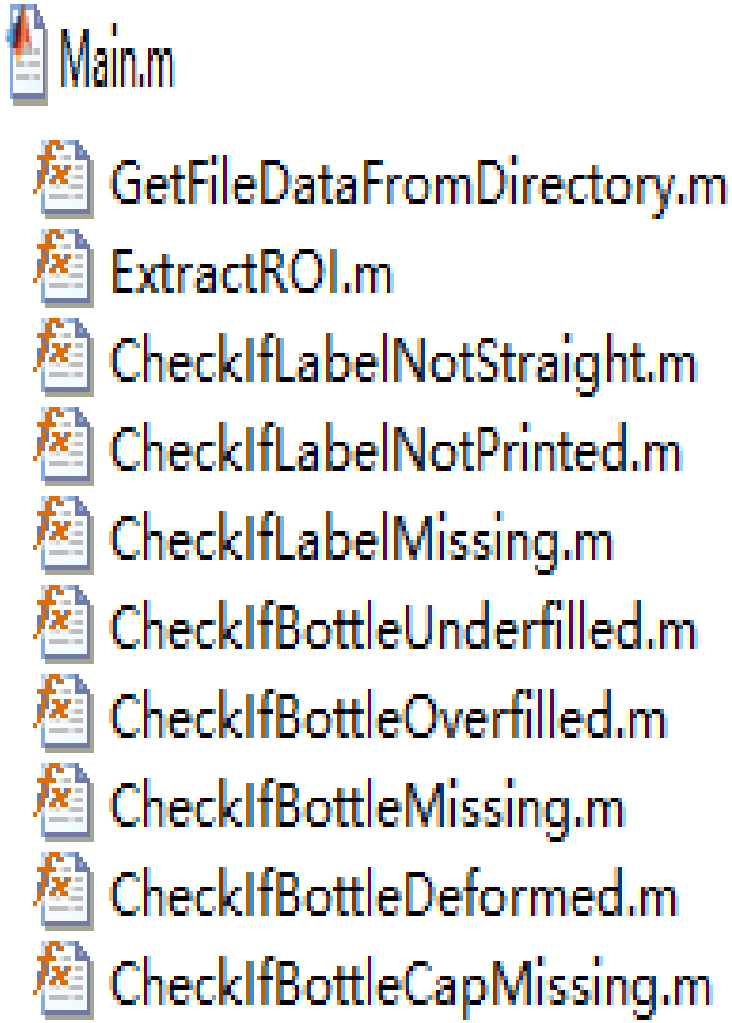
Origin image		
Not straight Line		

图像处理应用：分类问题


Main.m


GetFileDataFromDirectory.m
ExtractROI.m
CheckIfLabelNotStraight.m
CheckIfLabelNotPrinted.m
CheckIfLabelMissing.m
CheckIfBottleUnderfilled.m
CheckIfBottleOverfilled.m
CheckIfBottleMissing.m
CheckIfBottleDeformed.m
CheckIfBottleCapMissing.m

图像处理应用：分类问题



伪代码

```
load Datasets

for i=1:length(Datasets)
        do CheckBottleMissing
        if BottleMissing
                record
        else
                do bottleCapMissing
        if bottleCapMissing
                record
        end
        .....
```

图像处理应用：分类问题

Fault Type	Images	Faults Detected	Classification%
Bottle Cap Missing	10	10	100%
Bottle Deformed	10	9	90%
Bottle missing	11	11	100%
Bottle Overfilled	10	10	100%
Bottle Underfilled	10	10	100%
Label Missing	10	10	100%
Label Not Printed	10	10	100%
Label Not Straight	10	10	100%
Multiple Faults	10	9	90%

Fault Type	Images	Faults Detected	Classification%
All	141	139	98.58%

Thank You!