

Object Detection with Machine Learning

This script replicates what was covered in the video "Object Detection with Machine Learning". Additionally, at the end are some useful metrics for evaluating detection results. You will later use these metrics for the final detection project.

Loading the Ground Truth Data

The ground truth for the railroad signs dataset is stored in a ".mat" file in the course files. To access the ground truth data, you'll need to load it and then update the image file locations to match your system. To do this, run the following section of code.

```
load RRsignsGroundTruth.mat
```

Overwrite the image file locations stored in the ground truth variable to match your system.

```
testPath = overwriteGTruthLocations(gTruthTest,gTruthTrain);
```

All the file paths are resolved in this groundTruth object.

All the file paths are resolved in this groundTruth object.

Training the ACF Object Detector

Convert the ground truth variable into a table that the function `trainACFObjectDetector` can use.

```
objectTrainingData = objectDetectorTrainingData(gTruthTrain)
```

objectTrainingData = 90x2 table

	imageFilename	RRXingSign
1	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[357,379,316...]
2	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[627,317,309...]
3	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[629,207,343...]
4	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[367,150,303...]
5	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[408,83,307,...]
6	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[541,105,333...]
7	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[395,282,390...]
8	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[215,328,305...]
9	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[321,308,343...]
10	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[154,219,337...]
11	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[696,282,347...]
12	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[313,290,360...]
13	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...'	[371,340,353...]

	imageFilename	RRXingSign
14	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[409,198,448...
15	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[667,435,380...
16	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[497,106,579...
17	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[206,172,626...
18	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[466,393,518...
19	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[503,294,314...
20	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[331,408,393...
21	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[573,103,448...
22	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[624,244,314...
23	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[592,149,205...
24	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[474,168,372...
25	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[411,317,297...
26	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[353,114,399...
27	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[67,316,583,...
28	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[209,262,622...
29	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[137,110,470...
30	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[344,224,564...
31	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[216,239,633...
32	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[156,184,664...
33	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[183,195,495...
34	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[322,310,288...
35	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[366,238,294...
36	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[356,262,372...
37	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[370,261,466...
38	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[187,276,437...
39	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[381,207,454...
40	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[397,276,475...
41	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[558,407,492...
42	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[315,297,516...
43	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[237,248,368...
44	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[131,182,612...
45	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[434,260,314...
46	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[255,191,331...

	imageFilename	RRXingSign
47	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[328,243,346...
48	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[381,340,290...
49	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[357,259,455...
50	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[284,174,555...
51	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[76,126,497,...
52	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[179,138,541...
53	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[399,183,396...
54	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[169,163,496...
55	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[205,239,549...
56	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[126,220,539...
57	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[445,278,244...
58	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[584,303,357...
59	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[423,361,570...
60	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[389,54,695,...
61	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[49,221,390,...
62	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[409,333,403...
63	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[558,197,499...
64	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[326,266,391...
65	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[403,348,371...
66	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[427,219,481...
67	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[283,303,616...
68	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[293,97,636,...
69	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[528,294,239...
70	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[648,120,264...
71	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[443,114,377...
72	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[385,198,378...
73	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[306,195,262...
74	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[453,181,375...
75	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[537,204,332...
76	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[384,261,396...
77	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[159,250,367...
78	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[549,229,322...
79	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[582,232,295...

	imageFilename	RRXingSign
80	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[183,138,331...
81	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[421,276,326...
82	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[184,320,270...
83	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[574,169,398...
84	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[322,196,244...
85	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[520,233,268...
86	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[459,118,384...
87	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[495,211,400...
88	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[404,147,358...
89	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[482,142,307...
90	'E:\Computer Vision\Computer Vision for Engineering and Science\Data\Math...	[344,383,368...

Train the detector. This step may take a few minutes to run.

```
acfDetector = trainACFObjectDetector(objectTrainingData)
```

```
ACF Object Detector Training
The training will take 4 stages. The model size is 206x205.
Sample positive examples(~100% Completed)
Compute approximation coefficients...Completed.
Compute aggregated channel features...Completed.

-----
Stage 1:
Sample negative examples(~100% Completed)
Compute aggregated channel features...Completed.
Train classifier with 90 positive examples and 450 negative examples...Completed.
The trained classifier has 20 weak learners.

-----
Stage 2:
Sample negative examples(~100% Completed)
Found 279 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 90 positive examples and 450 negative examples...Completed.
The trained classifier has 19 weak learners.

-----
Stage 3:
Sample negative examples(~100% Completed)
Found 450 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 90 positive examples and 450 negative examples...Completed.
The trained classifier has 19 weak learners.

-----
Stage 4:
Sample negative examples(~100% Completed)
Found 450 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 90 positive examples and 450 negative examples...Completed.
The trained classifier has 25 weak learners.

-----
ACF object detector training is completed. Elapsed time is 41.1952 seconds.
acfDetector =
```

```
acfObjectDetector with properties:
```

```
    ModelName: 'RRXingSign'  
ObjectTrainingSize: [206 205]  
NumWeakLearners: 25
```

Detect Using the Trained Detector

Run this section to apply the detector on the test images. First, load the test images as an image datastore.

```
imdsTest = imageDatastore(testPath)
```

```
imdsTest =  
  ImageDatastore with properties:  
  
    Files: {  
      ' ... \Data\MathWorks Images\Railroad Signs\Test\RRXingSign_001.JPG';  
      ' ... \Data\MathWorks Images\Railroad Signs\Test\RRXingSign_002.JPG';  
      ' ... \Data\MathWorks Images\Railroad Signs\Test\RRXingSign_003.JPG'  
      ... and 7 more  
    }  
    Folders: {  
      ' ... \Data\MathWorks Images\Railroad Signs\Test'  
    }  
    AlternateFileSystemRoots: {}  
    ReadSize: 1  
    Labels: {}  
    SupportedOutputFormats: ["png" "jpg" "jpeg" "tif" "tiff"]  
    DefaultOutputFormat: "png"  
    ReadFcn: @readDatastoreImage
```

Run the detector.

```
bboxes = detect(acfDetector,imdsTest)
```

```
bboxes = 10x2 table
```

	Boxes	Scores
1	[420,148,375...	90.4952
2	[208,170,375...	105.7780
3	[]	[]
4	[636,285,315...	105.7780
5	[811,544,244...	99.5406
6	[273,186,315...	83.8488
7	[372,118,315...	88.4189
8	[427,346,410...	88.9790
9	[673,211,244...	57.2164
10	[]	[]

Insert the ground truth and detection results over each image. The ground truth bounding boxes are added as green rectangles and the detected bounding boxes as a yellow rectangle with a detection score. If a yellow rectangle is missing, that means the detector missed a sign.

```
for idx = 1:height(imdsTest.Files)
    % Load the image
    img = readimage(imdsTest,idx);

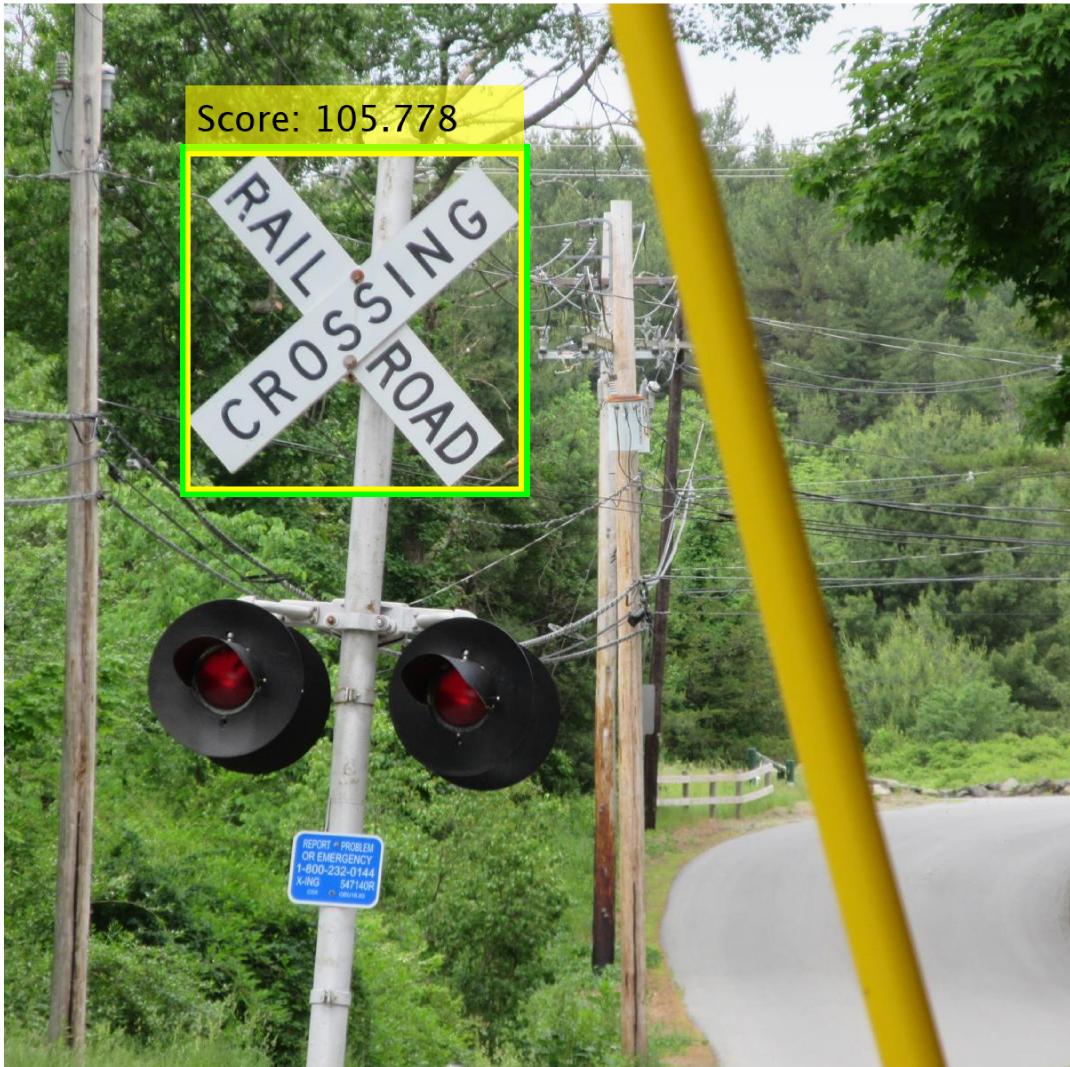
    % Insert the ground truth bounding box
    img = insertShape(img,"Rectangle",gTruthTest.LabelData.RailRoadXingSign{idx},...
        "Color","green","LineWidth",10);

    % Insert the detected bounding box
    img = insertObjectAnnotation(img,"Rectangle",bboxes.Boxes{idx},...
        "Score: " + bboxes.Scores{idx}, "FontSize",40, "LineWidth",6);

    % Display the image
    figure
    imshow(img)
end
```

Score: 90.4952







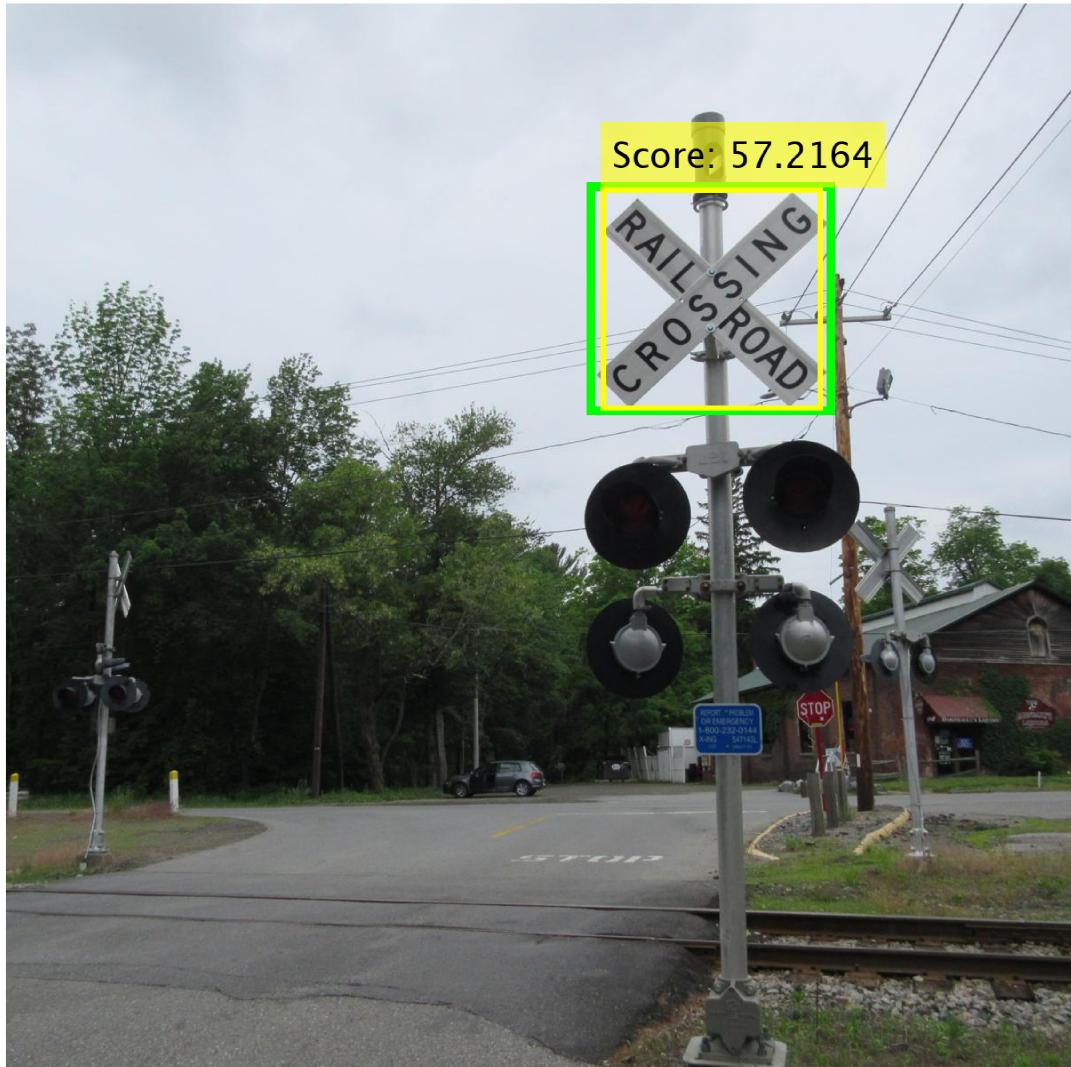














Evaluate the Results

Intersection over Union

A standard method for quantifying the accuracy of a single bounding box is to calculate its overlap ratio with the ground truth, also known as the Intersection over Union (IoU).

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

For example, here is the IoU value for one of the images.

```
idx = 4;  
img = readimage(imdsTest, idx);  
gTruthBbox = gTruthTest.LabelData.RailRoadXingSign{idx};  
detectionBbox = bboxes.Boxes{idx};
```

```
img = insertShape(img, "filled-rectangle", gTruthBbox, "Color", "green");
img = insertShape(img, "filled-rectangle", detectionBbox, "Color", "red");
imshow(img)
```



```
overlapRatio = bboxOverlapRatio(gTruthBbox, detectionBbox)
```

```
overlapRatio = 0.8415
```

A value of 0 means no overlap, and a value of 1 means perfect overlap.

Miss Rate and Precision

To measure the average performance of a detector over a collection of images, use the detection [miss rate](#) and [precision](#). Both metrics count a detection as "correct" if its IoU is above a threshold value (0.5 by default).

Miss Rate: Quantifies how many ground truth objects received no correct detections. This metric is log-averaged over all detections and scales from:

- 0 - if **all** the ground truth objects were detected.
- 1 - if **none** of the ground truth objects were detected.

```
evaluateDetectionMissRate(bboxes,gTruthTest.LabelData)
```

```
ans = 0.2000
```

Precision: Quantifies how many detections were correct out of all detections. Note that if a ground truth object receives multiple detections, then only one of them is marked "correct". This metric is log-averaged over all detections and scales from:

- 0 - if **none** of the detections were correct.
- 1 - if **all** the detections were correct.

```
evaluateDetectionPrecision(bboxes,gTruthTest.LabelData)
```

```
ans = 0.8000
```

Further Practice

In this example, the default trained detector missed the signs in 2 of the 10 test images, resulting in a miss rate of 0.2. See if you can adjust the options available for training the [trainACFObjectDetector](#) or [detect](#) functions to achieve a lower miss rate without also reducing the precision.

```
% Put your code here
```

```
acfDetector1= trainACFObjectDetector(objectTrainingData, "NumStages",8,"NegativeSamplesFactor",1)
```

```
ACF Object Detector Training
The training will take 8 stages. The model size is 206x205.
Sample positive examples(~100% Completed)
Compute approximation coefficients...Completed.
Compute aggregated channel features...Completed.
-----
Stage 1:
Sample negative examples(~100% Completed)
Compute aggregated channel features...Completed.
Train classifier with 90 positive examples and 180 negative examples...Completed.
The trained classifier has 19 weak learners.
-----
Stage 2:
Sample negative examples(~100% Completed)
Found 180 new negative examples for training.
Compute aggregated channel features...Completed.
Train classifier with 90 positive examples and 180 negative examples...Completed.
The trained classifier has 19 weak learners.
-----
Stage 3:
Sample negative examples(~100% Completed)
Found 180 new negative examples for training.
```

```
Compute aggregated channel features...Completed.  
Train classifier with 90 positive examples and 180 negative examples...Completed.  
The trained classifier has 19 weak learners.
```

Stage 4:

```
Sample negative examples(~100% Completed)  
Found 180 new negative examples for training.  
Compute aggregated channel features...Completed.  
Train classifier with 90 positive examples and 180 negative examples...Completed.  
The trained classifier has 19 weak learners.
```

Stage 5:

```
Sample negative examples(~100% Completed)  
Found 180 new negative examples for training.  
Compute aggregated channel features...Completed.  
Train classifier with 90 positive examples and 180 negative examples...Completed.  
The trained classifier has 21 weak learners.
```

Stage 6:

```
Sample negative examples(~100% Completed)  
Found 73 new negative examples for training.  
Compute aggregated channel features...Completed.  
Train classifier with 90 positive examples and 180 negative examples...Completed.  
The trained classifier has 23 weak learners.
```

Stage 7:

```
Sample negative examples(~100% Completed)  
Found 6 new negative examples for training.  
Compute aggregated channel features...Completed.  
Train classifier with 90 positive examples and 180 negative examples...Completed.  
The trained classifier has 24 weak learners.
```

Stage 8:

```
Sample negative examples(~100% Completed)  
Found 3 new negative examples for training.  
Compute aggregated channel features...Completed.  
Train classifier with 90 positive examples and 180 negative examples...Completed.  
The trained classifier has 27 weak learners.
```

ACF object detector training is completed. Elapsed time is 93.9239 seconds.

```
imdsTest = imageDatastore(testPath);  
bboxes = detect(acfDetector1,imdsTest,"MinSize",[210,210]);  
evaluateDetectionMissRate(bboxes,gTruthTest.LabelData)
```

```
ans = 0.1000
```

```
evaluateDetectionPrecision(bboxes,gTruthTest.LabelData)
```

```
ans = 0.9000
```

Helper Function

This function overwrites the image file locations stored in the ground truth variable to match your system.

```
function testPath = overwriteGTruthLocations(gTruthTest,gTruthTrain)
% Find the location of the image files
testPath = fileparts(which("RRXingSign_001.JPG"));
if isempty(testPath)
    error("The course files are not on the MATLAB search path." + ...
        "Follow the instructions when downloading the course files to add them to your path.")
end

% Overwrite locations for test images
currentPath = string(fileparts(gTruthTest.DataSource.Source{1}));
changeFilePaths(gTruthTest,[currentPath, testPath]);

% Overwrite locations for training images
trainPath = erase(testPath,'Test') + "Train";
currentPath = string(fileparts(gTruthTrain.DataSource.Source{1}));
changeFilePaths(gTruthTrain,[currentPath, trainPath]);
end
```

Copyright 2022 The MathWorks, Inc.