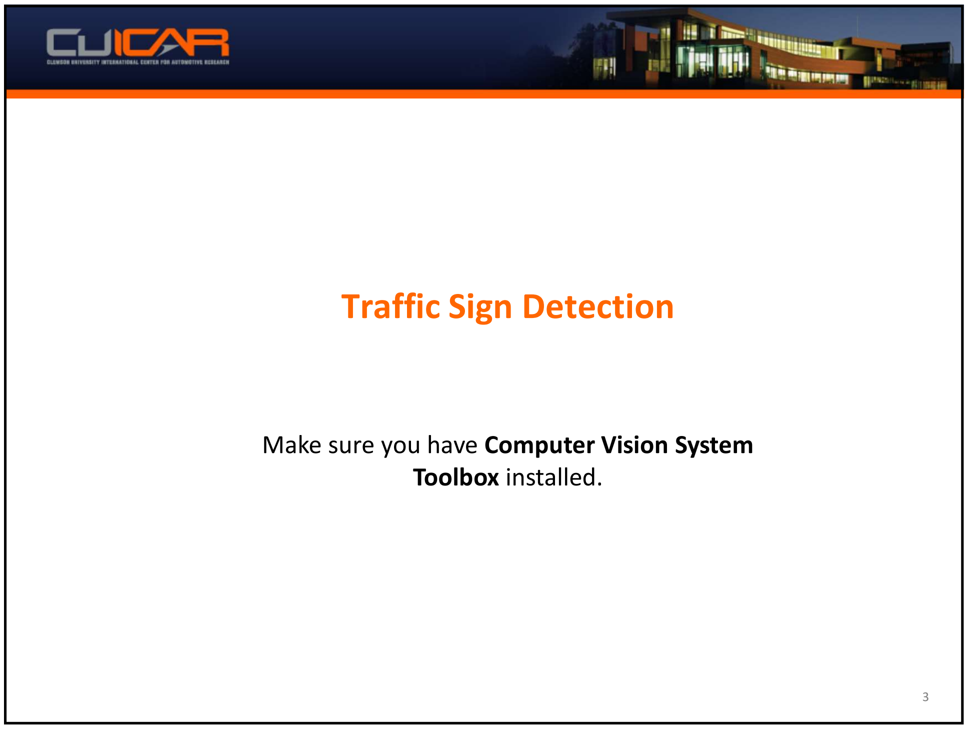
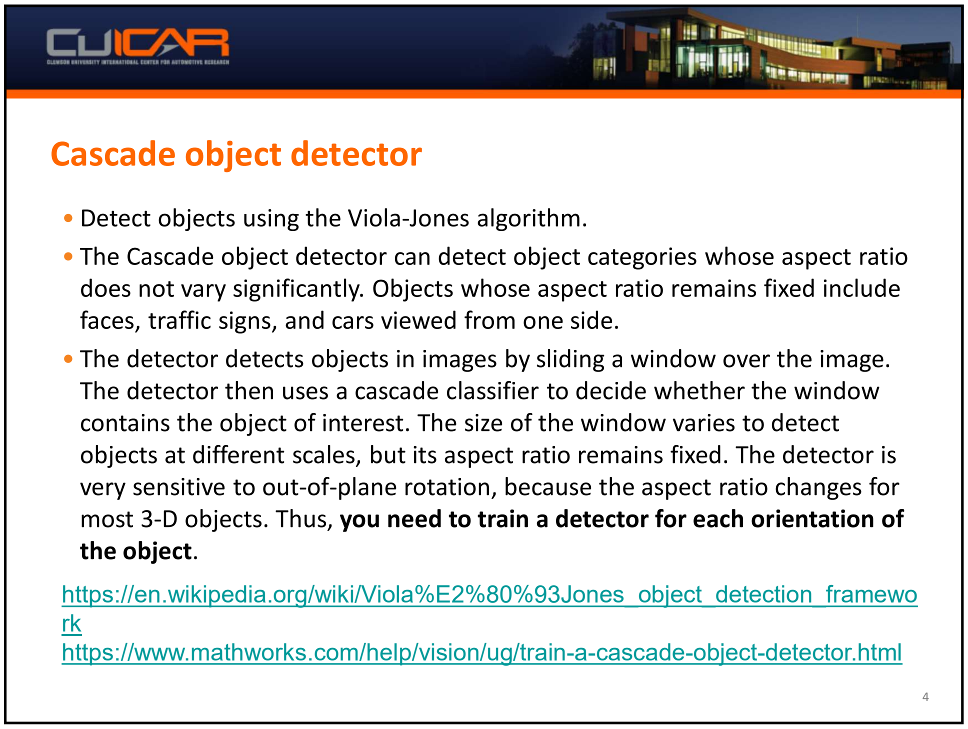


Traffic Sign Detection

Make sure you have **Computer Vision System Toolbox** installed.

3



Cascade object detector

- Detect objects using the Viola-Jones algorithm.
- The Cascade object detector can detect object categories whose aspect ratio does not vary significantly. Objects whose aspect ratio remains fixed include faces, traffic signs, and cars viewed from one side.
- The detector detects objects in images by sliding a window over the image. The detector then uses a cascade classifier to decide whether the window contains the object of interest. The size of the window varies to detect objects at different scales, but its aspect ratio remains fixed. The detector is very sensitive to out-of-plane rotation, because the aspect ratio changes for most 3-D objects. Thus, **you need to train a detector for each orientation of the object**.

https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework
https://www.mathworks.com/help/vision/ug/train-a-cascade-object-detector.html

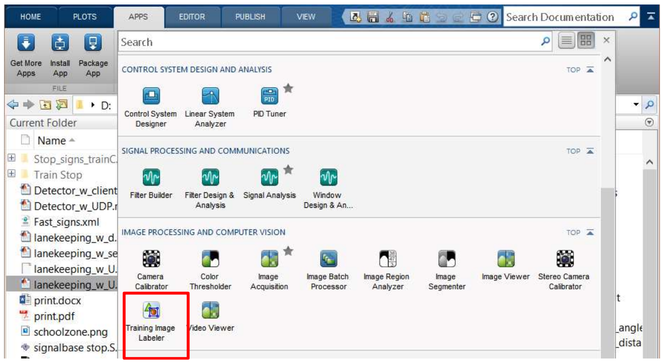
4

Train a Cascade object detector

- Cascade classifier training requires a set of **positive samples** and a set of **negative images**.
- You must provide a set of **positive images with regions of interest** specified to be used as positive samples. You can use the Training Image Labeler to label objects of interest with bounding boxes. The Training Image Labeler outputs a table to use for positive samples.
- You also must provide a set of negative images from which the function generates negative samples automatically.
- To achieve acceptable detector accuracy, set the number of stages, feature type, and other function parameters.
- A Cascade object detector can only detect one type of object. In our case we want to detect two different traffic signs, **we'll need to train two Cascade object detector separately.**

Generate training data with Training Image Labeler

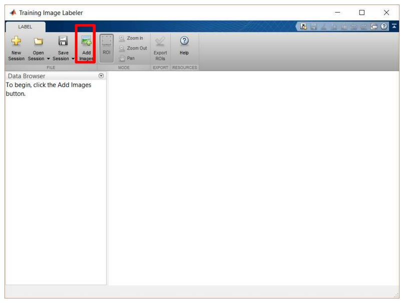
- Type `trainingImageLabeler` on the MATLAB command line or select it from the MATLAB desktop Apps tab.



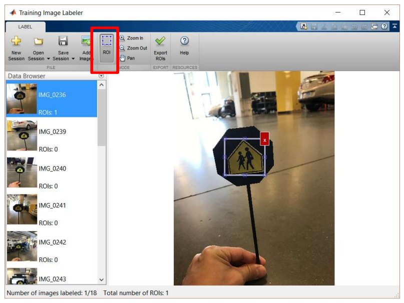
https://www.mathworks.com/help/vision/ref/trainingimagelabeler-app.html

Generate training data with Training Image Labeler

- Click ‘Add Images’ to add your **positive training images.**
- Manually select region of interest (ROI) from every image.



To begin, click the Add Images button.



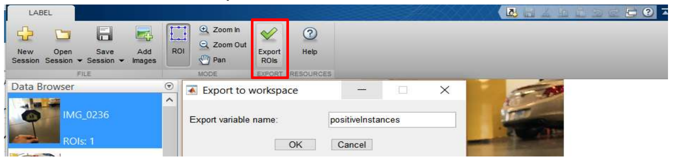
Number of images labeled: 1/18 Total number of ROIs: 1

https://www.mathworks.com/help/vision/ref/trainingimagelabeler-app.html

7

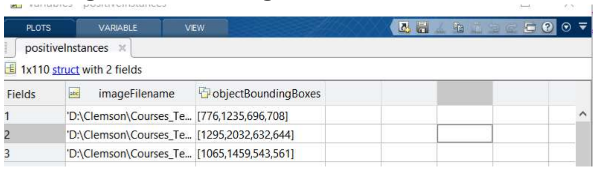
Generate training data with Training Image Labeler

- After finish selecting ROIs, click ‘Export ROIs’ and to export positive training data information into workspace.



Export variable name: positiveInstances

- Notice that the positveInstances variable is a struct that includes two fields: ‘imageFilename’ and ‘objectBoundingBoxes’. In the next steps the data will be read from the ‘imageFilename’ . Make sure **you do not change the name or the path of the images after lebeling.**



1x110 struct with 2 fields

Fields	imageFilename	objectBoundingBoxes
1	'D:\Clemson\Courses_Te...	[776,1235,696,708]
2	'D:\Clemson\Courses_Te...	[1295,2032,632,644]
3	'D:\Clemson\Courses_Te...	[1065,1459,543,561]

8

Train a Cascade object detector

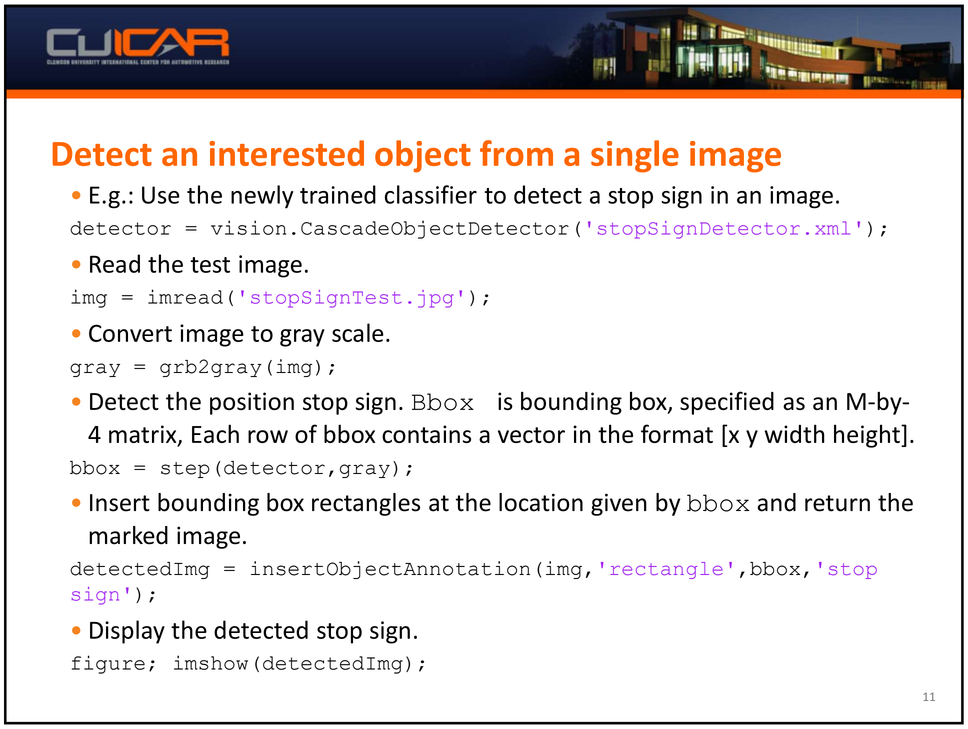
```
load('Your file');
negativeFolder = 'Your Folder';
trainCascadeObjectDetector('Your name.xml',
positiveInstances, negativeFolder, 'FalseAlarmRate',
0.3, 'NumCascadeStages', 10);
```

- `Your file` is the file that saves the ‘`positiveInstances`’ variable.
- `Your Folder` is the path that saves all the negative images. Make sure none of the images in this folder contains positive features.
- `'Your name.xml'` is the file that the trained detector will be saved in.
- `'NumCascadeStages'` is the number of cascade stages to train. A higher number of stages may need more training data and longer training time, but will result in a more accurate detector. Default value is 20.
- `'FalseAlarmRate'` is the acceptable false alarm rate at each stage, the false alarm rate is the fraction of negative training samples incorrectly classified as positive samples. Lower values for FalseAlarmRate can achieve fewer false detections but can result in longer training and detection times.

https://www.mathworks.com/help/vision/ref/traincascadeobjectdetector.html

Some hints for training a Cascade object detector

- The training function uses histograms of oriented gradients (HOG) to train the model, which means the RGB images will be converted into grayscale images before training. The color doesn’t mater, but the gradient of the grayscale image matters.
- When you are training a stop-sign detector, you should include negative images that contain school zone sign and vice versa.
- Trade-off between fewer stages with a lower false positive rate per stage or more stages with a higher false positive rate per stage: generally, it is better to have a greater number of simple stages with higher positive rate per stage.
- The amount of images needed: to train a decent traffic sign detector, at least 100 positive images are needed. The amount of negative images should be more than positive images.
- The time needed to train a detector: if you set false alarm rate to 0.3 and layers to 10, then it will take 30 mins or more to finish one training session.



Detect an interested object from a single image

- E.g.: Use the newly trained classifier to detect a stop sign in an image.

```
detector = vision.CascadeObjectDetector('stopSignDetector.xml');
```

- Read the test image.

```
img = imread('stopSignTest.jpg');
```

- Convert image to gray scale.

```
gray = grb2gray(img);
```

- Detect the position stop sign. `Bbox` is bounding box, specified as an M-by-4 matrix, Each row of bbox contains a vector in the format [x y width height].

```
bbox = step(detector,gray);
```

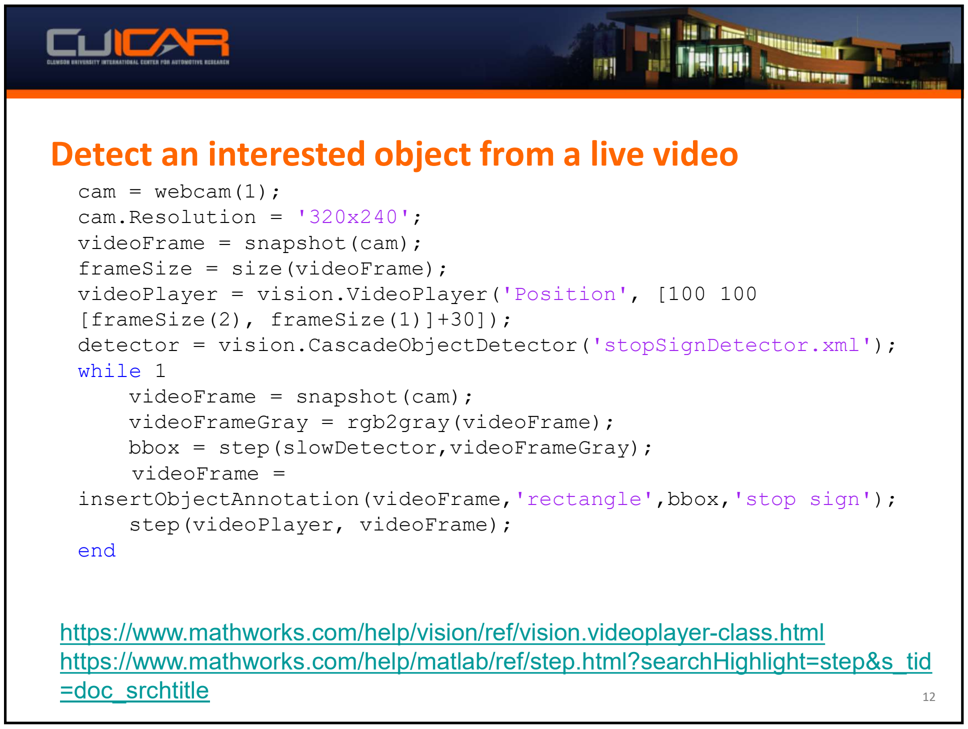
- Insert bounding box rectangles at the location given by `bbox` and return the marked image.

```
detectedImg = insertObjectAnnotation(img,'rectangle',bbox,'stop
sign');
```

- Display the detected stop sign.

```
figure; imshow(detectedImg);
```

11



Detect an interested object from a live video

```
cam = webcam(1);
cam.Resolution = '320x240';
videoFrame = snapshot(cam);
frameSize = size(videoFrame);
videoPlayer = vision.VideoPlayer('Position', [100 100
[frameSize(2), frameSize(1)]+30]);
detector = vision.CascadeObjectDetector('stopSignDetector.xml');
while 1
    videoFrame = snapshot(cam);
    videoFrameGray = rgb2gray(videoFrame);
    bbox = step(slowDetector,videoFrameGray);
     videoFrame =
insertObjectAnnotation(videoFrame,'rectangle',bbox,'stop sign');
    step(videoPlayer, videoFrame);
end
```

https://www.mathworks.com/help/vision/ref/vision.videoplayer-class.html
https://www.mathworks.com/help/matlab/ref/step.html?searchHighlight=step&s_tid=doc_srchtitle

12

Run two detectors at the same time

- In project 2, two different signs need to be detected, so two detectors need to be trained and run at the same time.
- In the test the 2 signs won’t show up at the same time. However, due to the imperfections in the trained model, two detectors may report a detected object at the same time.
- Unfortunately, `vision.CascadeObjectDetector` in Matlab doesn’t return confidence score. To get confidence score, support vector machine needs to be used, which could be a bit difficult for the project.
- You need to design a ‘vote’ algorithm that can find the ‘true’ detection result from the raw detection result.

https://www.mathworks.com/help/stats/fitcsvm.html

13

Run two detectors at the same time

```
points = detectMinEigenFeatures(videoFrameGray, 'ROI',
bbox(1, :));
xyPoints = points.Location;
numPts = size(xyPoints,1);
```

- `detectMinEigenFeatures` detects corners using minimum eigenvalue algorithm and return `cornerPoints` object.
- From observation, if a traffic sign is wrongly detected, then most of the times it will have fewer corners within the bounding box than the ‘true’ sign.
- By comparing the amount of corners within the two bounding boxes, the ‘true’ detection result can be possibly found.

14