

## Lab 5

### CSC 5930/9010 - Computer Vision

---

**Description:** In this lab you will practice the Viola Jones detector in Matlab. Because this method is so popular, it has made its way into the standard libraries of Matlab.

**Step 1** - Pick your favorite photograph that contains a face. Use the built in Viola Jones face detector to find faces in the image.

Create a detector object.

```
faceDetector = vision.CascadeObjectDetector;
```

Read input image.

```
I = imread('visionteam.jpg');
```

Detect faces.

```
bboxes = step(faceDetector, I);
```

Annotate detected faces.

```
IFaces = insertObjectAnnotation(I, 'rectangle', bboxes, 'Face');  
figure, imshow(IFaces), title('Detected faces');
```

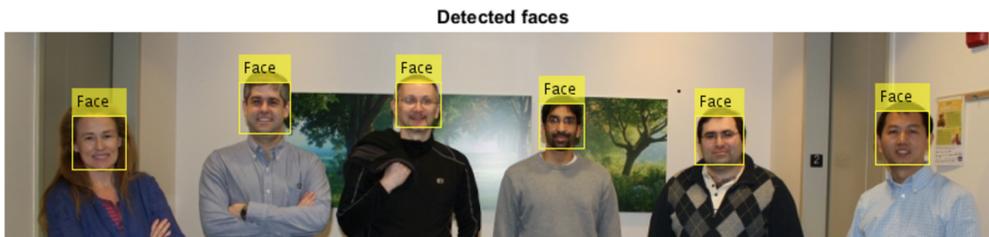


Figure 1: Instructions to run the built in VJ face detector.

**Step 2** - We are going to use the cascade trainer in matlab to build our own detector. You will be using the `trainCascadeObjectDetector` as shown below.

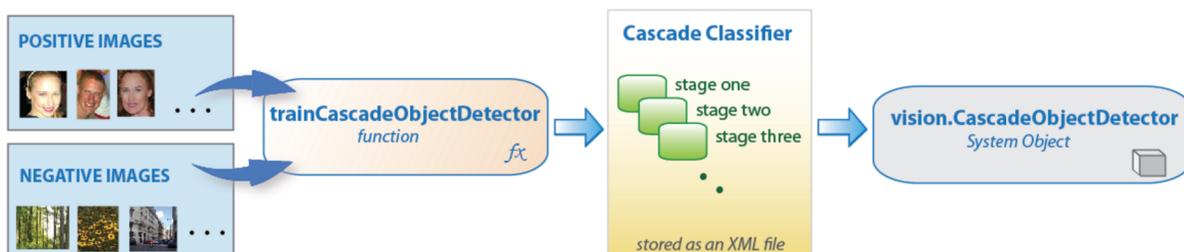


Figure 2: Instructions to run the cascade trainer in matlab.

The cascade classifier consists of stages, where each stage is an ensemble of weak learners. The weak learners are simple classifiers called decision stumps. Each stage is trained using a technique

called boosting. Boosting provides the ability to train a highly accurate classifier by taking a weighted average of the decisions made by the weak learners.

Each stage of the classifier labels the region defined by the current location of the sliding window as either positive or negative. Positive indicates that an object was found and negative indicates no objects were found. If the label is negative, the classification of this region is complete, and the detector slides the window to the next location. If the label is positive, the classifier passes the region to the next stage. The detector reports an object found at the current window location when the final stage classifies the region as positive.

```
Load positive samples.

load('stopSignsAndCars.mat');

Select the bounding boxes for stop signs from the table.

positiveInstances = stopSignsAndCars(:,1:2);

Add the image directory to the MATLAB path.

imDir = fullfile(matlabroot,'toolbox','vision','visiondata',...
    'stopSignImages');
addpath(imDir);

Specify the folder for negative images.

negativeFolder = fullfile(matlabroot,'toolbox','vision','visiondata',...
    'nonStopSigns');

Create an imageDatastore object containing negative images.

negativeImages = imageDatastore(negativeFolder);

Train a cascade object detector called 'stopSignDetector.xml' using HOG features. NOTE: The command can take several minutes to run.

trainCascadeObjectDetector('stopSignDetector.xml',positiveInstances, ...
    negativeFolder,'FalseAlarmRate',0.1,'NumCascadeStages',5);
```

Figure 3: Instructions to run the cascade trainer in matlab.

**Note about Negative Samples** - Negative samples are not specified explicitly. Instead, the `trainCascadeObjectDetector` function automatically generates negative samples from user-supplied negative images that do not contain objects of interest. Before training each new stage, the function runs the detector consisting of the stages already trained on the negative images. Any objects detected from these image are false positives, which are used as negative samples. In this way, each new stage of the cascade is trained to correct mistakes made by previous stages.

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');
```

Detect a stop sign.

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

Insert bounding box rectangles and return the marked image.

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

Display the detected stop sign.

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



Figure 4: Instructions to test the cascade trainer in matlab.

**Train your own object detector** Once you have successfully implemented the cascade classifier on the stop sign test, now its your turn to train a new classifier. Use the “trainingImageLabeler” command to provide positive examples of your own object. Export your ROIs and use them instead of the stopSignsAndCars table from above.

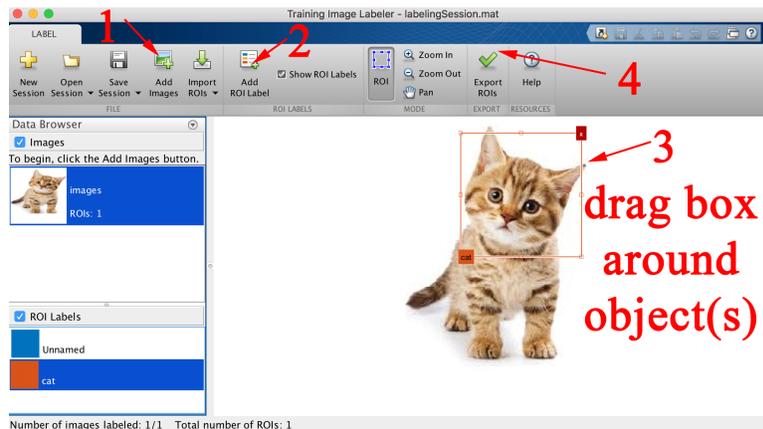


Figure 5: Instructions to create your own cascade calssifier.