

Contents

- Load positive images and bounding boxes of bikes,laptops,AC,pen,table
- set the positive instances
- Step 2: Specify folder with negative images
- Step 3: Train the detector
- start the trainging of cascade object detectors for 5 stages (bikes)
- start the trainging of cascade object detectors for 5 stages (Laptops)
- start the trainging of cascade object detectors for 5 stages (AC)
- start the trainging of cascade object detectors for 5 stages (pen)
- start the trainging of cascade object detectors for 5 stages (table)

```
% Name: training.m
% type: object recognition
% description: open the file in mat lab asnd run the script it will start
% takimng the images from the web cam and extract the object from the image
% s of the video and display the object name in a pop up window.
% contact-us: https://learn-kevin.blogspot.com
% Email :      sidd5sci@gmail.com
% =====
%                               Object Recognition Training
% =====

% clearing
clc
close all
clear all
% adding paths of xml files and input images
addpath('database');
addpath('database/bike');
addpath('database/laptop');
addpath('XML Files');
```

Load positive images and bounding boxes of bikes,laptops,AC,pen,table

```
% positiveInstances = data;
% load laptopPositive.mat
% load acPositive.mat
% load penPositive.mat
% load tablePositive.mat
%whos -file BikePositive.mat
laptop = load('laptopPositive.mat');
bike = load('bikePositive.mat');
%disp('Contents of workspace after loading file:')
%whos
```

set the positive instances

```
positiveInstances1 = struct2table(bike.Bike);
```

```
positiveInstances2 = laptop.Laptop;
%disp(positiveInstances1);
%disp(laptop.Laptop)
```

Step 2: Specify folder with negative images

```
negativeFolder = [pwd '\database\nonbike'];
negativeImages = imageDatastore(negativeFolder);
```

Step 3: Train the detector

```
prompt = {'Enter the number training stages:', 'FAR'};
dlg_title = 'Stages';
num_lines = 1;
defaultans = {'9', '0.25'};
labelType = inputdlg(prompt, dlg_title, num_lines, defaultans);
% setting the number of stages and the FAR
NumStages = str2num(labelType{1:1});
FAR = str2num(labelType{2:2});
```

start the training of cascade object detectors for 5 stages (bikes)

```
trainCascadeObjectDetector('bikedetector_5_5.xml', Bike, negativeFolder, 'NumCascadeStages', NumStages, 'FalseAlarmRate', FAR);
```

```
disp("Training bike:")
trainCascadeObjectDetector('bikedetector_9_25.xml', positiveInstances1, negativeFolder, 'FalseAlarmRate', 0.1, 'NumCascadeStages', 5);
```

Training bike:

Specified outputXMLFileName bikedetector_9_25.xml
already exists. What would you like to do?

- [1] Delete existing xml file and start training from the beginning
- [2] Exit this function without making any changes

Enter 1/2 [2]

Error using input
Cannot call INPUT from EVALC.

Error in trainCascadeObjectDetector>shouldExitBecauseOutputXMLFileExists (line 743)
reply = input('', 's');

Error in trainCascadeObjectDetector (line 168)
if shouldExitBecauseOutputXMLFileExists(parser.Results.outputXMLFileName)

```
Error in training (line 58)
trainCascadeObjectDetector('bikedetector_9_25.xml',positiveInstances1,negativeFolder,'FalseAlarmRate',0.1,'NumCascadeStages',5);
```

start the training of cascade object detectors for 5 stages (Laptops)

```
disp("Training laptop:")
trainCascadeObjectDetector('laptopdetector_9_25.xml', positiveInstances2, negativeFolder, 'NumCascadeStages', NumStages, 'FalseAlarmRate', FAR);
```

start the training of cascade object detectors for 5 stages (AC)

```
disp("Training AC:")
%trainCascadeObjectDetector('ACdetector_5_5.xml', positiveInstances, negativeFolder, 'NumCascadeStages', NumStages, 'FalseAlarmRate', FAR);
```

start the training of cascade object detectors for 5 stages (pen)

```
disp("Training pen:")
%trainCascadeObjectDetector('pendetector_5_5.xml',positiveInstances, negativeFolder,'NumCascadeStages', NumStages, 'FalseAlarmRate', FAR);
```

start the training of cascade object detectors for 5 stages (table)

```
disp("Training table:")
%trainCascadeObjectDetector('tabledetector_5_5.xml', data, negativeFolder, 'NumCascadeStages', NumStages, 'FalseAlarmRate', FAR);
```