

Contents

- [detect the image from the web cam images](#)
- [Initiate the detectors for detecting the objects](#)
- [detect the object from the input images](#)
- [cleaning the detector](#)
- [stop the video](#)

```
% Name: testing.m
% type: object recognition
% description: open the file in mat lab and run the script it will start
% taking 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 Testing
% =====

% clearing
clc
close all
clear all
% adding paths of xml files and input images
addpath('inputSet');
addpath('XML Files');

% initilising the webcam
%vid = videoinput('winvideo',1,'YUY2_320x240');
% set the frame rate of video from webcam
% set(vid,'FramePerTrigger',Inf);
% set the camera color formate
%set(vid,'ReturnedColorspace','rgb');
% set the frame interval
%vid.FrameGrabInterval = 5;
% start the webcam
%start(vid);
% preview(vid);
```

detect the image from the web cam images

counter

```
i = 0;
% main loop
% while( i <3)
%     % taking a snapshot of from the video and storeing it in the image
%     % variable
%     image = getsnapshot(vid);
%     % converting the image in to gray scale
```

```

%     image = rgb2gray(image);
%     % figure;
%     % display the grayscale image to wthe window
%     % imshow(image);
%
%     %% Extract HOG Features
%     [hog16, vis16] = extractHOGFeatures(image,'CellSize',[16 16]);
%     figure;
%     imshow(image);
%     hold on;
%     plot(vis16);
%     % init the bounding box around the object
%     bbox = step(detector1,image);
%     %% Mark the location on the image using a bounding box
%     J = insertShape(image,'rectangle',bbox);
%     imshow(J)
%
%
%     % increase the counter
%     i = i + 1;
%
% end

```

Initiate the detectors for detecting the objects

init the dilog box for the input of the detector

```

prompt = {'Enter the detector name:'};
dlg_title = 'Name';
num_lines = 1;
defaultans = {'laptop'};
detectorType = inputdlg(prompt,dlg_title,num_lines,defaultans);
% checking the detector type send by user for the initiation
if strcmp(detectorType,'laptop')
    Detector = vision.CascadeObjectDetector('laptopdetector_9_25.xml');
elseif strcmp(detectorType,'bike')
    Detector = vision.CascadeObjectDetector('bikedetector_9_25.xml');
elseif strcmp(detectorType,'human')
    Detector = vision.CascadeObjectDetector('humandetector_9_25.xml');
end

```

deltect the object from the input images

init the dilog box for the input of the detector

```

prompt = {'Enter the image name(.jpg):'};
dlg_title = 'Name';
num_lines = 1;
defaultans = {'laptop (126)'};
% taking the image name from the user
imageName = inputdlg(prompt,dlg_title,num_lines,defaultans);
% concating the extention
loadImage = strcat(imageName,'.jpg');
% checking the image send by user for the detection

```

```
s = char(strcat('InputSet\',loadImage));
% load the input image from input set
img = imread(s);
% detect the image from the image using detector
bbox = step(Detector,img);
% inset the detected boundary in to the image
J = insertShape(img,'rectangle',bbox);
% display the detected region over the input image
imshow(J)
```



cleaning the detector

```
release(Detector);
```

stop the video

```
%stop(vid)
%flushdata(vid);
clear all
```