



安阳工学院  
ANYANG INSTITUTE OF TECHNOLOGY

---

## 训练 OpenCV 分类器

基于深度学习的自动驾驶小车的研究与实现

2017 年 5 月 28 日

# 目录

第一章 在 Windows 下训练分类器.....	1
1.1 准备工具.....	1
1.2 使用工具.....	2
1.2.1 制作正样本图片.....	2
1.2.2 制作负样本图片.....	3
1.2.3 创建样本.....	3
1.2.4 训练.....	4
1.2.5 最后生成模型文件.....	4
第二章 在 Linux 下训练分类器 .....	5
2.1 准备工作.....	5
2.2 步骤.....	5
2.2.1 准备要检测的目标.....	5
2.2.2 准备负样本（背景） .....	5
2.2.3 生成正样本数据.....	6
2.2.4 生成样本（create sample） .....	6
2.2.5 训练.....	6
2.2.6 使用生成的级联分类器.....	7

## 第一章 在 Windows 下训练分类器

本章参考了 YouTube 的教程, 地址为: <https://www.youtube.com/watch?v=Dg-4MoABv4I>

### 1.1 准备工具

工具放在了 GitHub 上面, 地址为: <https://github.com/rh01/haar-classifier-training>。首先克隆到本地 Windows 下。

```
git clone https://github.com/rh01/haar-classifier-training.git
```

具体包含以下目录:



图 1-1 工具的相关目录

首先 **cascades** 翻译过来的意思为级联, 也就是 OpenCV 里面的级联分类器。感兴趣的同学可以参考这篇官方的教程: [http://docs.opencv.org/trunk/dc/d88/tutorial\\_traincascade.html](http://docs.opencv.org/trunk/dc/d88/tutorial_traincascade.html), 其中 **cascades/** 目录下存放着每次迭代产生的参数信息。

**negative/**目录存放着所有的负样本图片, 主要是背景信息。**positive/rawdata/**目录下存放着你要检测和识别的目标图片 (或包含目标的图片)。

比如现在假设一个场景, 我要检测出道路上含有 STOP 标志牌, 并进行标识和检测, 也就是说给我一张图片, 然后在图片上找到感兴趣的区域即 STOP 部分, 接下来标识出来。所以在训练的时候, 需要有正样本和负样本, 正样本就是 STOP 部分, 而负样本就是背景 如下图所示:



图 1-2 negative/199.jpg



图 1-3 positive/rawdata/2.bmp

**vector** 目录下存放着生成的样本文件。

## 1.2 使用工具

### 1.2.1 制作正样本图片

正样本图片的格式为 **bmp** 格式的，存放位置为 **positive/rawdata/**。如下图所示。



图 1-4 **positive/rawdata/**

由于正样本标签包含非感兴趣区域，因此需要使用 **objectmaker.exe** 来选取感兴趣区域。双击 **Objectmaker.exe**，会弹出图片，然后利用鼠标选取目标，并按下空格键进行保存。选取的感兴趣区域的坐标和长宽等参数信息。然后按下回车键，进行下一张图片的选取和保存。依次这样做，直至结束。



图 1-5 鼠标选取 STOP 标志区域



图 1-6 按下空格后，会打印出目标区域的具体参数信息

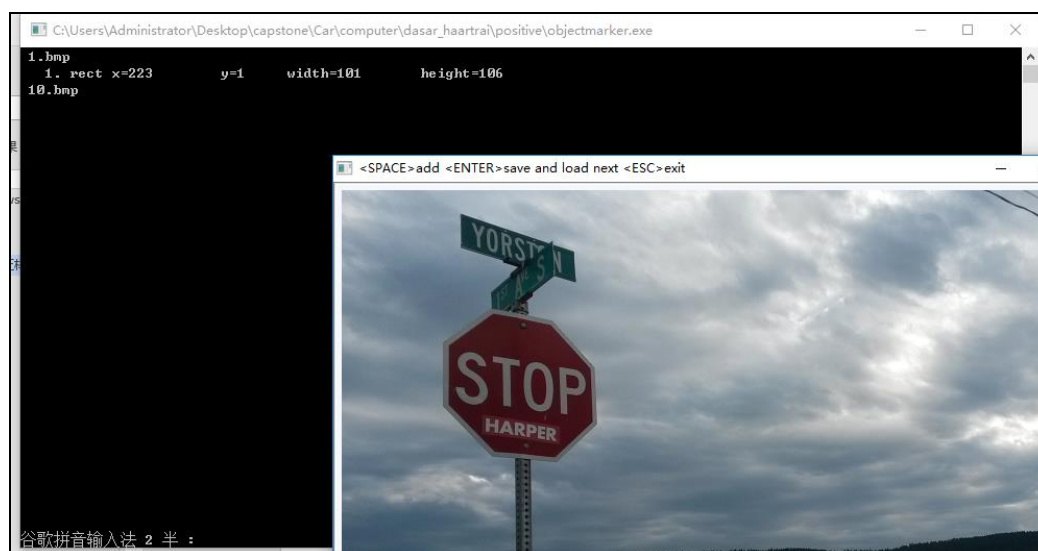


图 1-7 按下回车后，会显示下一张图片

最后，全部选取完成后，会将这些信息保存在 **positive/info.txt**

### 1.2.2 制作负样本图片

默认情况下，会有 200 张负样本图片（背景），格式不限，也可以增加图片数目。

双击 **negative/create\_list.bat** 这个批处理脚本，就会生成 **negative/bg.txt**，上面记录着所有负样本的文件名。

### 1.2.3 创建样本

双击根目录下的 **samples\_creation.bat**，会生成 positive vector 的文件，存放在 **vector**

目录下。

需要注意的是 **sample\_creation.bat** 需要定制。比如 **01 samples\_creation.bat** 文件内容是：

```
createsamples.exe -info positive/info.txt -vec vector/vector.vec -num 304 -w 24 -h 24
```

-num 304 表示负样本的个数，-w 24 -h 24 表示在检测的时候，检测 24\*24 大小的目标区域。

#### 1.2.4 训练

打开并编辑根目录下的脚本文件 **haar-Training.bat**, 改变 positive 和 negative 图片的值。并运行它。

```
haartraining.exe -data cascades -vec vector/vector.vec -bg negative/bg.txt -npos 29 -nneg 304 -nstages 15  
-mem 2048 -mode ALL -w 24 -h 24  
  
rem -nonsym
```

#### 1.2.5 最后生成模型文件

双击 **convert.bat** 脚本，会生成级联分类器模型 **myharr.xml**

## 第二章 在 Linux 下训练分类器

本章参考文章：

<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>

### 2.1 准备工作

1) opencv 环境

2) 目录结构

```
pi@raspberrypi:~ $ tree data/ -d
data/
├── data
├── info
├── neg
└── uglies

4 directories
```

图 2-1 目录结构

**data/data** 文件夹下存放着级联分类器文件，**data/info** 文件夹存放着正样本，**data/neg** 文件夹存放着负样本图片。

### 2.2 步骤

#### 2.2.1 准备要检测的目标

在这个例子中，我要做的是检测和识别出这个手表，如图 2-2 所示。



图 2-2 watch5050.jpg

#### 2.2.2 准备负样本（背景）



图 2-3 neg/1.jpg

默认情况下负样本的尺寸比正样本的尺寸大。正样本是  $50 \times 50$ ，而负样本是  $100 \times 100$  的尺寸。

### 2.2.3 生成正样本数据

由于只有一张目标图片，因此 OpenCV 提供了相应的工具，生成与负样本相同数量的正样本。

```
opencv_createsamples -img watch5050.jpg -bg bg.txt -info info/info.lst -pngoutput info -maxxangle 0.5  
-maxyangle 0.5 -maxzangle 0.5 -num 1950
```

-img watch5050.jpg 这是目标图片

-bg bg.txt 负样本图片的信息

-num 1950 负样本的个数（生成的正样本数与负样本图片的数目相同）

生成的正样本如下图：



图 2-4 info/0003\_0026\_0010\_0068\_0068.jpg

仔细看的话，会发现有目标图像的痕迹，这就是生成的正样本图片。

### 2.2.4 生成样本（create sample）

```
opencv_createsamples -info info/info.lst -num 1950 -w 20 -h 20 -vec positives.vec
```

-num 1950

-w 20

-h 20

与第一章相同。

### 2.2.5 训练

```
opencv_traincascade -data data -vec positives.vec -bg bg.txt -numPos 1800 -numNeg 900 -numStages 10  
-w 20 -h 20
```

-data data 表示生成的级联分类器的文件存放的目录

-numPos 1800 正样本的数目



-numNeg 900 负样本的数目

-numStages 10 训练的迭代次数

e	cascade.xml	2017/5/23 18:08	XML 文件	323 KB
e	params.xml	2017/5/23 15:07	XML 文件	1 KB
e	stage0.xml	2017/5/23 15:07	XML 文件	3 KB
e	stage1.xml	2017/5/23 15:10	XML 文件	4 KB
e	stage2.xml	2017/5/23 15:12	XML 文件	3 KB
e	stage3.xml	2017/5/23 15:14	XML 文件	4 KB
e	stage4.xml	2017/5/23 15:16	XML 文件	3 KB
e	stage5.xml	2017/5/23 15:18	XML 文件	4 KB
e	stage6.xml	2017/5/23 15:20	XML 文件	3 KB
e	stage7.xml	2017/5/23 15:23	XML 文件	4 KB
e	stage8.xml	2017/5/23 15:26	XML 文件	4 KB
e	stage9.xml	2017/5/23 15:28	XML 文件	3 KB
e	stage10.xml	2017/5/23 16:24	XML 文件	4 KB
e	stage11.xml	2017/5/23 16:27	XML 文件	4 KB
e	stage12.xml	2017/5/23 16:29	XML 文件	4 KB

图 2-5 生成的级联分类器，红色的代表最终的模分类器

## 2.2.6 使用生成的级联分类器

```
import numpy as np

import cv2

#this is the cascade we just made. Call what you want
watch_cascade = cv2.CascadeClassifier('cascade.xml')

img = cv2.imread("opencv-object-recognition.png")

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# add this

# image, reject levels level weights.
watches = watch_cascade.detectMultiScale(gray, 100, 100)

# add this

for (x,y,w,h) in watches:
```

```
cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)
```

```
cv2.imshow('img',img)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

效果图如下：



图 2-6 检测到手表