

人脸检测算法

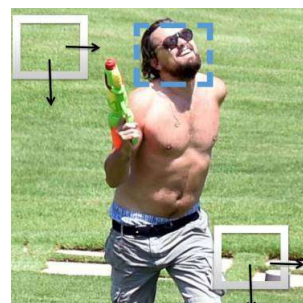
1

人脸（物体）检测

物体检测是计算机视觉中的经典问题之一，其任务是用框去标出图像中物体的位置，并给出物体的类别

检测算法里面通常包含三个部分，

1. 检测窗口的选择；
2. 图像特征的设计；
3. 分类器的设计



2

人脸检测方法

基于规则/知识方法

- 人脸模式的变化满足一定的规律，所以可以归纳描述人脸特征的规则，如灰度分布、比例关系、纹理信息等

基于模板的方法

- 固定模板法，可变形模板法

基于不变特征的方法，如彩色信息

- 人脸的肤色在彩色空间中的分布相对比较集中，所以可用来检测和跟踪人脸。

基于外观学习的方法---目前的主流方法

- 将人脸检测视为区分“非人脸样本”与“人脸样本”的模式识别问题，通过对人脸样本集和非人脸样本集的学习产生分类器

3

3

基于肤色特征的检测

在很多人脸检测和手的跟踪应用中，人的肤色信息已被证明是一种非常有效的特征

有很多颜色空间可用来表征肤色，包括RGB, YUV或者YCrCb（亮度+两个色差）、HSV（色调、饱和度、亮度）等

有许多建立肤色模型的方法

- 最简单的方法是用Cr, Cb值定义肤色区域，选定阈值[Cr1,Cr2]和[Cb1,Cb2]
- 一个像素点被归为肤色点，如果它的(Cr, Cb)值落入下面的范围：
 $Cr1 \leq Cr \leq Cr2$ and $Cb1 \leq Cb \leq Cb2$

4

4

多种肤色模型

J.L.Crowley and F. Berard, "Multi-modal tracking of faces for video communications"

- 将肤色区域的RGB颜色归一化，用其中的(r,g)值的颜色直方图h(r,g) 获取肤色变量的阈值

M. H. Yang, N. Ahuja, "Detecting human faces in color images"

- 认为人脸肤色区域的颜色值呈高斯分布，用高斯分布的均值和方差确定肤色变量的阈值

T. S. Jebara and A. Pentland, "Parameterized structure from motion from 3D adaptive feedback tracking of faces"

- 认为不同的种族和国家的人的肤色分布不同，在颜色直方图上形成多个聚类，可用高斯混合模型来表示

5

5

颜色空间

(r g b) 'x' 'y'

RGB到 "rg" 空间

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$$

RGB到YUV(YCrCb)空间，再转化到 "FI" 空间

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \quad F = \tan^{-1}(|V|/|U|), \quad I = (0.596, -0.274, -0.322) \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

6

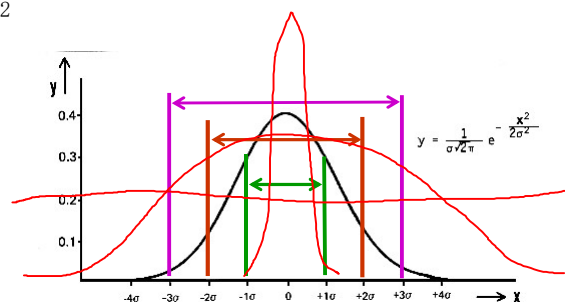
6

高斯肤色模型

一元正态分布肤色模型 (以F颜色特征为例)

$$F \sim N(\mu, \sigma)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N F_i, \quad \sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (F_i - \mu)^2$$



7

7

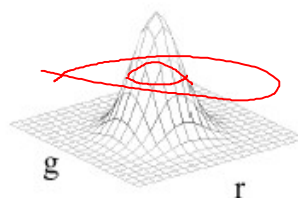
高斯肤色模型

二元正态分布肤色模型 (以rg颜色特征为例)

$$N(\mu, \Sigma), \text{ where } \mu = (\bar{r}, \bar{g})$$

$$\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i, \quad \bar{g} = \frac{1}{N} \sum_{i=1}^N g_i$$

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix}$$



8

8

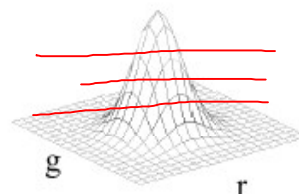
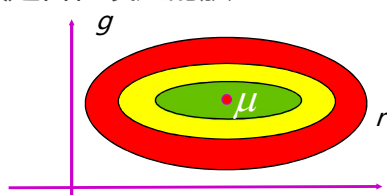
多人脸训练肤色模型

训练肤色模型

- 手工标注部分人脸(肤色区域)人脸图像
- 统计方法得到 μ, Σ
- 设置合适的阈值截断

测试阶段

- 逐像素判断其是否在设定的肤色特征范围内



9

9

提取肤色区域

对检测到的肤色区域进行分析

- 接近椭圆形
- 有部分非肤色区域(五官、头发)



10

10

肤色模型的缺点

肤色模型难以适应各种环境光照变化

- 对于背景和前景的**光照变化，肤色通常不稳定**
- 单纯的肤色信息对于人脸检测通常是不充分的

如何适应在不同光照下的人脸跟踪

- 如果环境光照有变化，原有的肤色模型可能不再适用，如何建立一个自适应的肤色模型需要进一步的研究

11

11

其他特征检测算法——关键点检测

关键点：

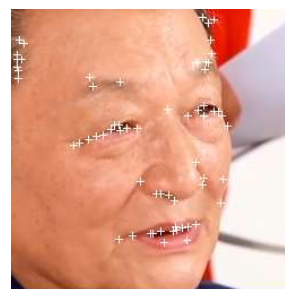
An interest point is a point in an image which has a well-defined position and can be robustly detected. ——维基百科

与周边任意点都有较大差异

- 比如眼睛周围、鼻孔周围，嘴唇周围

具有良好的数学性质

- 稳定性、不变性
- 通常用于目标物体的跟踪



12

关键特征点检测：“一点”数学

关键点的数学性质？（这里的“点”可以是图片上的一个小区域）与周边有较大差异，其中差异就是平方误差和S（维基百科）

$$S(x, y) = \sum_u \sum_v w(u, v) (I(u + x, v + y) - I(u, v))^2$$

$$I(u + x, v + y) \approx I(u, v) + I_x(u, v)x + I_y(u, v)y \quad \text{泰勒展开}$$

$$S(x, y) \approx \sum_u \sum_v w(u, v) (I_x(u, v)x + I_y(u, v)y)^2 \approx \begin{pmatrix} x & y \end{pmatrix} A \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\text{其中 } A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x(u, v)^2 & I_x(u, v)I_y(u, v) \\ I_x(u, v)I_y(u, v) & I_y(u, v)^2 \end{bmatrix}$$

13

关键特征点检测：“一点”数学

S越大，A的两个特征值 λ_1, λ_2 就应该越大，所以变成求解矩阵的特征值问题

- 等价于 $\min(\lambda_1, \lambda_2)$ 要“大”
- 遍历所有区域的A矩阵，求得 $\min(\lambda_1, \lambda_2)$ 大的点，就是关键点

matlab实现：**detectMinEigenFeatures**

14

关键点检测算法及其matlab实现

```
img = imread('../orl_faces/s1/1.pgm');
points = detectMinEigenFeatures(img);
imshow(img);hold on;
plot(points.Location(:,1), points.Location(:,2),'r+' );
```

其他检测算法：

detectFASTFeatures

detectSURFFeatures

detectHarrisFeatures



15

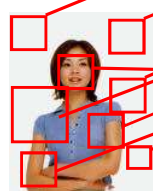
基于机器学习的方法

ANN

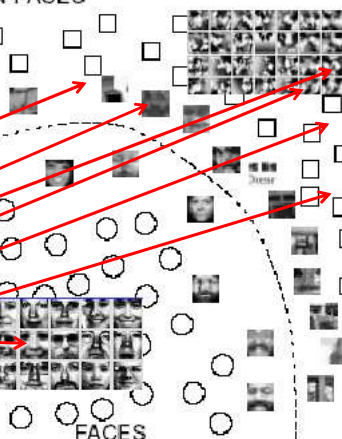
SVM

Naïve Bayes Classifier

AdaBoost



NON-FACES



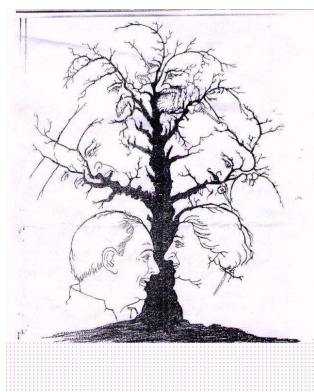
遍历所有可能的“矩形窗口”，判断每个小窗口是否人脸？

(请问，一张 $n \times n$ 的图片有多少矩形？)

16

16

基于AdaBoost 的快速人脸检测



17

17

AdaBoost算法简介

在2001年的ICCV上，当时在Compaq的研究员Paul Viola和Michael J.Jones发表的文章介绍一个实时人脸检测系统

- 图像大小为384x288时，其速度是平均每秒15帧
- 第一个准实时的(准正面)人脸检测系统
- 可以较容易的扩展到多姿态人脸检测

系统在技术上的三个贡献：

- 1.用简单的Haar-like矩形特征作特征，可快速计算
- 2.基于AdaBoost的分类器设计
- 3.采用了Cascade(分级分类器)技术提高检测速度

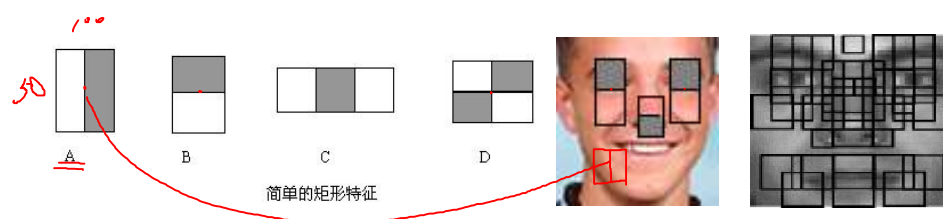
18

18

人脸的特征表示方法

矩形特征 (Haar-like特征)

- 矩形特征的值是所有白色矩形中点的亮度值的和减去所有灰色矩形中点的亮度值的和，所得到的差
- 有4种类型的矩形特征

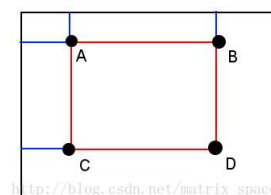
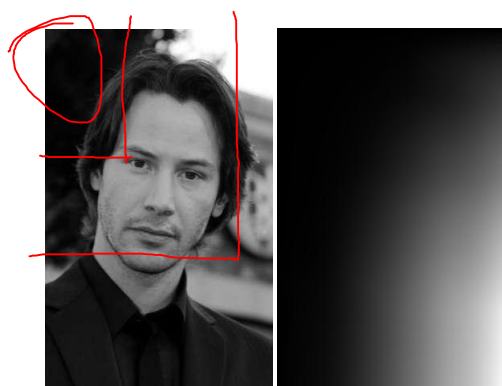


19

19

积分图像

$I = \text{intergralImage}(img);$



$$S_{abcd} = I(D) - I(B) - I(C) + I(A)$$

20

积分图像与Haar特征

一个矩形可以由四个点来表示，二邻接矩形需要六个点表示，三邻接矩形需要八个点，而四邻接矩形需要九个点

根据卷积定理，我们知道一个 $W \times H$ 的图像与 $m \times n$ 的filter 做卷积，新生成的图像大小为 $(W-m+1) \times (H-n+1)$ ，新图像的每一个像素其实就是原图一个 $m \times n$ 的local patch与 $m \times n$ 的filter 的乘积和。新图像有多少个像素，就对应着原图多少个 $m \times n$ 的矩形。

一个 24×24 的图像最终会产生162336个矩形特征，这个维度远远高于图像本身的维度

21

如何应对极多特征

1. 特征选择--通过分类器训练中添加显式或者隐式的特征选择机制来实现：

Boosting--不同特征维度上的弱分类器

SVM--L1正则化或L2正则化

2. 分类器结构设计

分类器级联--应对极不平衡的正负类分类问题

层次化分类器--应对极多类数据分类问题

22

AdaBoost分类器

AdaBoost分类器

- Adaboost学习算法可以提高简单分类算法的性能
- 通过对一些弱分类器的组合来形成一个强分类器

功能

- 将分类性能不好的弱分类器提升为分类性能好的强分类器的学习算法

思想

- 学习一系列分类器，在这个序列中每一个分类器对它前一个分类器导致的错误分类样例给予更大的重视

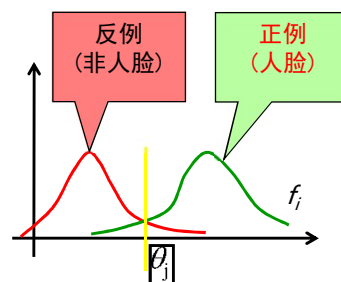
23

23

AdaBoost用于人脸检测

训练弱分类器

$$h_j(x) = \begin{cases} 1, & \text{if } f_j(x) \geq \theta_j \\ 0, & \text{otherwise} \end{cases}$$



其中， h 表示弱分类器的值， θ_j 表示弱学习算法寻找出的阈值， $f_j(x)$ 表示特征值， x 表示一个Harr-like特征。

24

24

AdaBoost: 三个臭皮匠 > 诸葛亮

强分类器

$$f(x) = \sum \alpha_i h_i(x_i) \begin{cases} 1 & \text{if } f(x) \geq 0.5 \\ 0 & \text{elsewise} \end{cases}$$

0.5 → < 0.1)

25

Adaboost训练过程

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:

1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

so that w_t is a probability distribution.

2. For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
3. Choose the classifier, h_t , with the lowest error ϵ_t .

4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

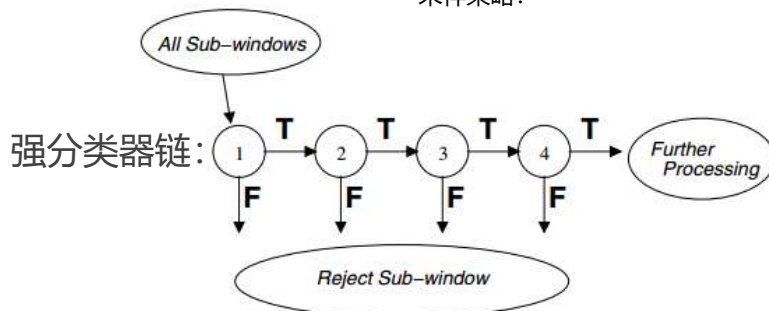
十几个特征!!!

26

Cascade-一种经典的物体检测架构

解决正反例样本极度不均衡的问题

采样策略：



27

第一个强分类器选取的特征

下一个分类器处理的数据：

1. 设定一个较低的阈值，使得所有的正例人脸为+
2. 该阈值使得40%的反例被误判成人脸(较低标准)，
选取仍被误判为人脸的反例样本作为-

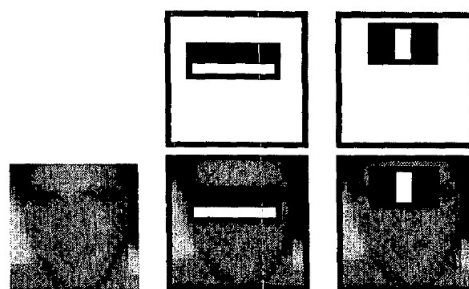


Figure 4: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

28

基于深度学习的目标（人脸）检测算法

Viola and Jone所提方法开创了视觉目标检测的新时代，但仍存在如下缺陷：

1. 未能解决如何产生候选区域的问题
2. 仅能检测正脸
3. Haar 特征仍属于“底层”视觉特征

29

目标检测流程和技术总览

1. 区域选择

这一步是为了对目标进行定位。传统方法是采用穷举策略。由于目标可能在图片上的任意位置，而且大小不定，因此使用滑动窗口的策略对整幅图像进行遍历，而且需要设置不同的长宽。这种策略虽然可以检测到所有可能出现的位置，但是时间复杂度太高，产生的冗余窗口太多，严重影响后续特征的提取和分类速度的性能。

2. 特征提取

提取特征的好坏会直接影响到分类的准确性，但又由于目标的形态多样性，提取一个鲁棒的特征并不是一个简单的事。这个阶段常用的特征有SIFT（尺度不变特征变换，Scale-invariant feature transform）和HOG（方向梯度直方图特征，Histogram of Oriented Gradient）等。

3. 分类器

主要有SVM，Adaboost等

综上所述，传统目标检测存在两个主要问题：一个是基于滑动窗口的区域选择策略没有针对性，时间复杂度高，窗口冗余；而是手工设计的特征对于多样性没有很好的鲁棒性。

30

区域选择

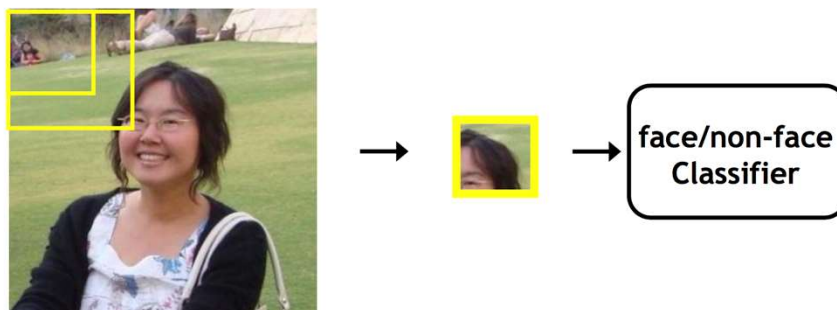
针对滑动窗口问题，region proposal（候选区域）是预先找出图中目标可能出现的位置，这可以保证在选取较少窗口（几千个甚至几百个）的情况下保持较高的召回率。并且获取的候选窗口要比滑动窗口的质量更高（滑动窗口固定长宽比）。

- 比较常用的region proposal算法有selective search和edge boxes。针对特征选取，卷积神经网络(convolution neural network: CNN) 的特征比传统手工特征效果更好。
- 2014年，RBG（Ross B. Girshick）使用Region proposal+CNN（RCNN）代替传统目标检测使用的滑动窗口+手工设计特征，设计了RCNN框架，使得目标检测取得巨大突破，并开启了基于深度学习的目标检测热潮。

31

区域选择

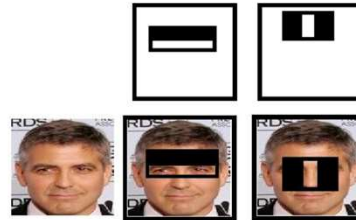
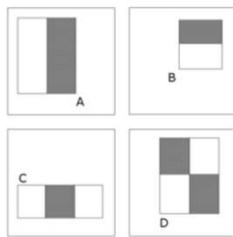
- Scans the detector at multiple **locations** and **scales**



32

特征提取

Haar-feature

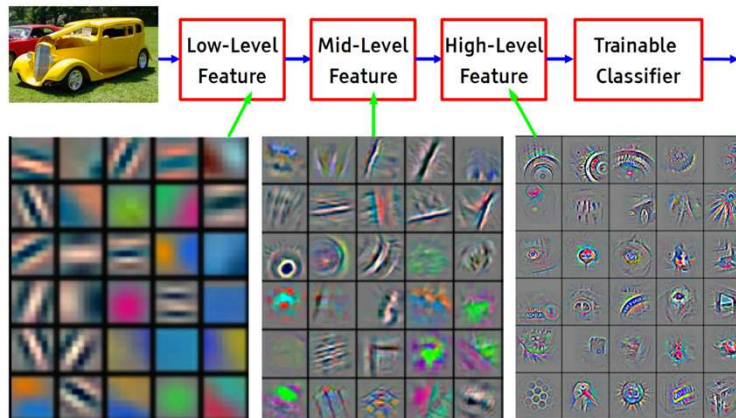


33

CNN提取特征

Deep Learning = Learning Hierarchical Representations

It's deep if it has more than one stage of non-linear feature transformation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

34

RCNN

Region CNN(RCNN)可以说是利用深度学习进行目标检测的开山之作。作者Ross Girshick多次在PASCAL VOC的目标检测竞赛中折桂，2010年更带领团队获得终身成就奖，如今供职于Facebook旗下的FAIR。

http://web.cs.ucdavis.edu/~yjee/teaching/ecs289h-fall2014/CollinMcCarthy_RCNN.pdf

35

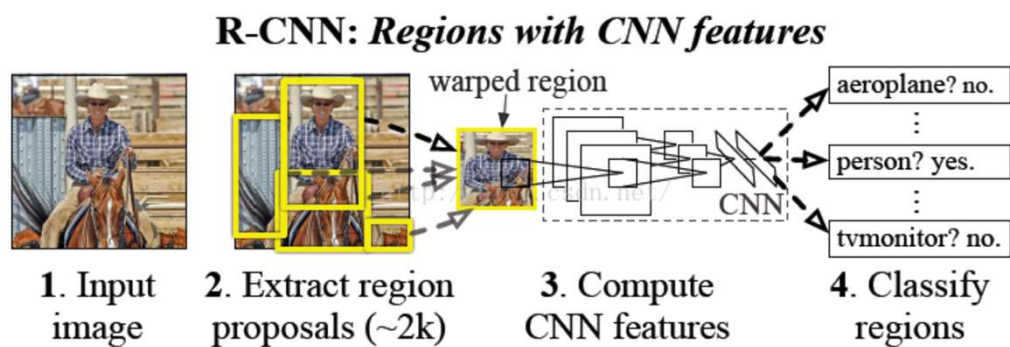
RCNN

RCNN算法分为4个步骤

- 一张图像生成1K~2K个**候选区域**
- 对每个候选区域，使用深度网络**提取特征**
- 特征送入每一类的SVM **分类器**，判别是否属于该类
- 使用回归器**精细修正**候选框位置

36

RCNN



37

Fast RCNN

FRCNN针对RCNN在训练时是multi-stage pipeline和训练的过程中很耗费时间空间的问题进行改进。它主要是将深度网络和后面的SVM分类两个阶段整合到一起，使用一个新的网络直接做分类和回归。

38

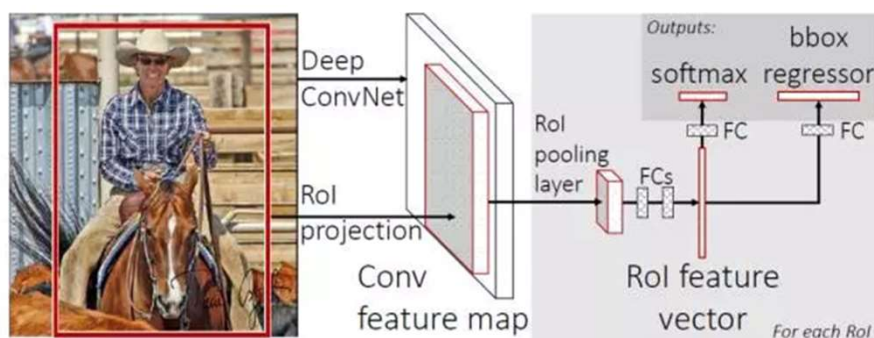
Fast RCNN

主要做以下改进:

1. 最后一个卷积层后加了一个ROI pooling layer。ROI pooling layer首先可以将image中的ROI定位到feature map, 然后是用一个单层的SPP layer将这个feature map patch池化为固定大小的feature之后再传入全连接层。
2. 损失函数使用了多任务损失函数(multi-task loss), 将边框回归直接加入到CNN网络中训练。

39

Fast RCNN结构



40

Faster RCNN

与Fast R-CNN的分类网络共享了卷积层

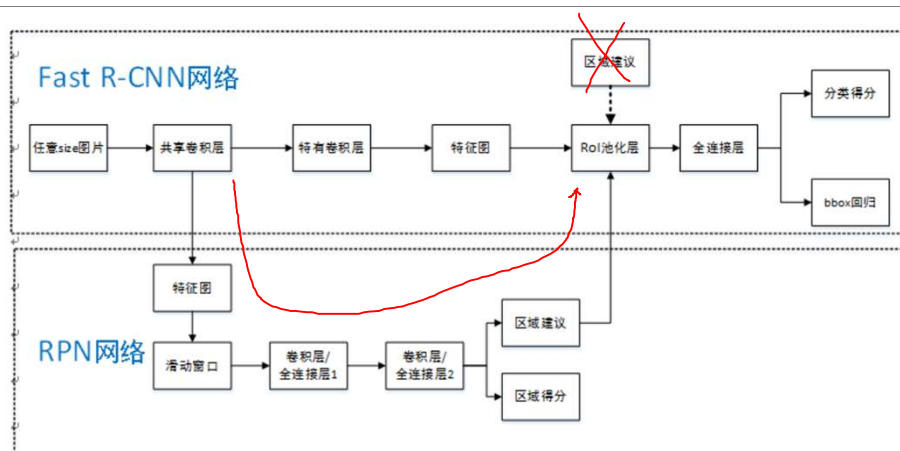
RPN替代了之前非常慢的Selective Search

所用的候选窗口的数目也比较少

Region Proposal Network (RPN) → object bounds and objectness score

41

Faster RCNN



<https://arxiv.org/pdf/1506.01497.pdf>

42

RCNN概览



43

实验结果-性能曲线, RCNN, fast RCNN, faster R-CNN

RCNN与fast RCNN训练速度比较

	R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours
	(Speedup)	1x
FASTER!	Test time per image	47 seconds
	(Speedup)	1x
Better!	mAP (VOC 2007)	66.0
		66.9

44

实验结果-性能曲线, RCNN, fast RCNN, faster R-CNN

RCNN与fast RCNN测试速度比较

Test-time speeds don't include region proposals

	R-CNN	Fast R-CNN
Test time per image	47 seconds	0.32 seconds
(Speedup)	1x	146x
Test time per image with Selective Search	50 seconds	2 seconds
(Speedup)	1x	25x

45

实验结果-性能曲线, RCNN, fast RCNN, faster R-CNN

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	0.2 seconds
(Speedup)	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

46

基于AdaBoost的视频人脸检测

- 1、建立人脸检测器
- 2、读取视频
- 3、用检测器检测视频每一帧上的人脸（并标出位置）
- 4、视频播放完毕，退出

47

快速搭建简单的人脸检测系统

创建一个基于AdaBoost的人脸检测器

```
faceDetector =  
vision.CascadeObjectDetector('FrontalFaceCART');
```

创建一个视频文件读取对象，接受avi、mp4文件和图片文件

```
videoFileReader = vision.VideoFileReader('visionface.avi'  
);
```

48

快速搭建简单的人脸检测系统

创建视频播放器

```
videoInfo = info(videoFileReader);  
videoPlayer = vision.VideoPlayer('Position',[300 100  
videoInfo.VideoSize+30]);
```

49

快速搭建简单的人脸检测系统

播放视频并检测

```
while ~isDone(videoFileReader)  
    videoFrame = step(videoFileReader);  
    %在每一帧上检测人脸，并且标记  
    bbox = step(faceDetector, videoFrame);  
    videoOut = insertObjectAnnotation(videoFrame,'rectangle',bbox,'Face');  
    step(videoPlayer, videoOut); %imshow(videoOut)  
end
```

50

快速搭建简单的人脸检测系统

完成后释放资源

```
release(videoFileReader);
```

```
release(videoPlayer);
```

51

测试一下这个系统

52

作业——orl_face数据批量读取

```
function [f_matrix,realclass] = ReadFace(npersons,flag)
%读取orl_face人脸库照片里的数据到矩阵
%输入:
%   npersons-需要读入的人数,每个人的前五幅图为训练样本,后五幅为验证样本
%   imgrow-图像的行像素为全局变量
%   imgcol-图像的列像素为全局变量
%   flag-标志,为0表示读入训练样本,为1表示读入测试样本
%输出:
%   f_matrix-所有读入图片保存在矩阵f_matrix中,f_matrix中的每个行向量
%           是一张图片,可以用img(:)把一个图片矩阵变成行向量。例如,
%           img = imread(face_file); f_matrix(i,:) = img(:)';
%   realclass-列向量,记录f_matrix中每一行是第几个人
%已知全局变量: imgrow=112; imgcol=92;
global imgrow;
global imgcol;
```

如果是彩色图,可以:
img = rgb2gray(img)
变成灰度图

53

作业-思考题

1. Haar特征属于哪种视觉特征 (颜色、纹理、形状)
2. Adaboost对训练样本做加权的目的是什么?
3. 在不考虑效率的前提下,如何提高Adaboost强分类器的效果?

54