**北京航空航天大学软件学院**

工程实践结题报告

课题名称： 基于深度学习的行人检测系统的设计与实现

姓 名： 王冠 陈林 华旎 夏涛

组 别： 15

学 号： ZF1821238 ZF1821103

ZF1821128 ZF1821308

专业方向： 人工智能研究方向

所属院系： 北京航空航天大学软件学院

指导老师：

**实践起止时间：2019年3月 至 2019年7月**

摘要

行人检测技术由于应用的广泛性使其在计算机视觉领域成为一个重要分支，对视频监控、车辆辅助驾驶、智能机器人等多个领域提供了重要的技术支持。比如视频监控中关注的主要是行人的运动轨迹和行为，要实现对行人跟踪和行为分析首先要做的就是检测出视频中的行人。然而现实生活中由于行人在衣着上的多样性、形态变化的多样性、所处背景的多样性、光照强弱的多样性、行人之间相互遮挡使得行人检测存在着诸多需要解决的问题。

本文对基于深度学习的目标检测做了介绍，并选用其中的Faster R-CNN算法进行行人检测，首先对行人检测算法进行设计，对图像中的行人目标生成区域建议并进行类别匹配。即先采用卷积神经网络对图片进行特征提取，针对传统的选择性搜索算法在生成区域建议时经常导致建议窗口数目过多，导致训练耗时，占用空间大等问题，本文直接通过RPN（区域建议网络）生成建议窗口送入后续网络进行类别判断，将区域建议的生成也统一到深度学习框架中，实现在GPU中完成全部计算。从而大大减小产生的窗口数目，进一步提高区域建议的生成速度与准确率。训练Faster R-CNN训练模型，通过反向传播和随机梯度下降的方法对RPN进行端到端的训练，在卷积神经网络的基础上额外增加了两个卷积层，实现对目标边界和分数的准确预测。然后以联合训练的方式将训练后的RPN和目标检测网络进行二次交替训练，实现RPN与目标检测联合网络的建立和卷积层共享。

行人检测系统主要应用于检测自然场景中的行人以及行人数目，实现行人目标的实时检测与行人数目统计。通过对比实验和测试结果进一步表明，本行人检测系统能够达到较高的行人检测率，可以满足实际的系统需求。

关键词：深度学习、行人检测、计算机视觉、人工智能、目标检测

**Abstract**

Pedestrian detection technology has become an important branch in the field of computer vision due to its wide application, providing important technical support for video surveillance, vehicle-assisted driving, intelligent robots and other fields. For example, video surveillance focuses on pedestrian trajectories and behaviors. To achieve pedestrian tracking and behavior analysis, the first thing to do is to detect pedestrians in the video. However, in real life, due to the diversity of pedestrians' clothing, the diversity of morphological changes, the diversity of backgrounds, the diversity of illumination, and the mutual occlusion of pedestrians, there are many problems that need to be solved in pedestrian detection.

In this paper, the target detection based on deep learning is introduced, and the Faster R-CNN algorithm is used for pedestrian detection. Firstly, the pedestrian detection algorithm is designed to generate regional recommendations and class matching for pedestrian targets in the image. Firstly, the feature extraction is carried out by using the convolutional neural network. The traditional selective search algorithm often leads to the number of suggestion windows when the regional recommendation is generated, which leads to the problem of time-consuming training and large space. This paper directly passes the RPN (region). It is recommended that the network generate a suggestion window and send it to the subsequent network for category judgment. The generation of the regional proposal is also unified into the deep learning framework to complete all calculations in the GPU. Thereby, the number of generated windows is greatly reduced, and the generation speed and accuracy of the regional recommendations are further improved. The Faster R-CNN training model is trained to train the RPN end-to-end through backpropagation and stochastic gradient descent. Two convolutional layers are added to the convolutional neural network to achieve the target boundary and fraction. Accurate forecasting. Then, the trained RPN and the target detection network are trained alternately in a joint training manner to realize the establishment of the joint network of RPN and target detection and the convolutional layer sharing.

Pedestrian detection system is mainly used to detect the number of pedestrians and pedestrians in natural scenes, real-time detection of pedestrian targets and statistics of pedestrians. The comparison experiment and test results further show that the pedestrian detection system can achieve a high pedestrian detection rate and can meet the actual system requirements.

**Keywords**: deep learning, pedestrian detection, computer vision, artificial intelligence, target detection

**目录**

[摘要 I](#_Toc31193)

[Abstract II](#_Toc12261)

[第一章 绪论 1](#_Toc27032)

[1.1 项目背景及意义 1](#_Toc2282)

[1.2 国内外发展现状 2](#_Toc26084)

[1.2.1 国外发展现状 2](#_Toc32147)

[1.2.2 国内发展现状 3](#_Toc19551)

[第二章 行人检测技术相关理论综述 5](#_Toc31069)

[2.1深度学习框架介绍 5](#_Toc21869)

[2.1.1深度学习 5](#_Toc19089)

[2.2.1 Caffe框架 5](#_Toc4029)

[2.2行人检测方法综述 8](#_Toc22269)

[2.3基于深度学习的目标检测方法综述 9](#_Toc18962)

[2.3.1 R-CNN 10](#_Toc30216)

[2.3.2 SPP-NET 12](#_Toc890)

[2.3.3 Fast R-CNN 14](#_Toc1106)

[2.3.4 Faster R-CNN 16](#_Toc11948)

[2.4 Open CV技术简介 18](#_Toc28585)

[2.5特征提取 18](#_Toc22659)

[2.5.1卷积神经网络特征提取 19](#_Toc5811)

[2.5.2 VGG-16特征提取 20](#_Toc15711)

[第三章 行人检测算法的设计与实现 22](#_Toc856)

[3.1行人检测算法的设计 22](#_Toc10854)

[3.1.1算法总体设计 22](#_Toc5504)

[3.1.2算法详细设计 23](#_Toc4563)

[3.1.3此算法主要设计原理 24](#_Toc2214)

[3.2Faster R-CNN网络训练模型的训练 34](#_Toc12984)

[3.2.1训练文件架构 35](#_Toc21261)

[3.2.2训练环境 35](#_Toc30689)

[3.2.3主要训练过程 36](#_Toc5020)

[3.3行人检测算法的实现 36](#_Toc3767)

[第四章 行人检测系统的测试与分析 40](#_Toc11184)

[4.1数据集 41](#_Toc32341)

[4.1.1数据集的介绍 41](#_Toc19557)

[4.1.2数据集的结构 41](#_Toc2094)

[4.2系统测试实验 42](#_Toc15129)

[4.2.1复杂环境下的测试 42](#_Toc9970)

[4.3实验结果分析 42](#_Toc28487)

[第五章 演示系统设计 44](#_Toc26009)

[5.1系统架构设计 45](#_Toc31714)

[5.2 web展示系统 45](#_Toc22455)

[5.2.1 web展示页面设计 46](#_Toc2264)

[5.2.2 web展示系统使用框架介绍 46](#_Toc1418)

[5.3模型层 47](#_Toc1383)

[5.4 数据存储层 48](#_Toc29025)

[第六章 个人工作总结 49](#_Toc17006)

[6.1王冠个人总结 49](#_Toc18763)

[6.2陈林个人总结 50](#_Toc23512)

[6.3华旎个人总结 51](#_Toc12081)

[6.4夏涛个人总结 52](#_Toc21207)

[参考文献 53](#_Toc4805)

## 第一章 绪论

### 项目背景及意义

行人检测( Pedestrian Detection)是计算机视觉中的经典问题，利用计算机视觉技术判断图像或者视频序列中是否存在行人并给予精确定位。行人检测是行人跟踪，行为分析，步态分析，行人身份识别等研究的基础和前提，一个好的行人检测算法能够为后者提供优良的支持和保障。

行人检测系统的研究起始于二十世纪九十年代中期，是目标检测的一种。从最开始到2002年，研究者们借鉴、引入了一些图像处理、模式识别领域的成熟方法，侧重研究了行人的可用特征、简单分类算法。自2005 年以来，行人检测技术的训练库趋于大规模化、检测精度趋于实用化、检测速度趋于实时化。随着高校、研究所以及汽车厂商的研究持续深入，行人检测技术得到了飞速的发展。

行人检测在计算机应用领域有着非常广泛的应用，比如车辆辅助驾驶系统、智能视频监控、机器人、航拍图像、人机交互系统、运动分析等。近年来智能车辆领域研究中备受关注的前沿方向，目前一些汽车生产厂商、 大学和研究机构相继开始了行人检测技术的研究  。比如，欧洲戴姆勒、德国大众等就发起了旨在研究行人保护方案的 PROTECTOR 项目 , 并且已经取得了初步的成效。2004-2005年的SAVE-U项目实现的目标是减少行人和车辆碰撞造成的伤亡数量和事故等级, 并在危险状况下驾驶员警告和车辆自动减速试验车辆。作为无人驾驶的先驱技术之一，行人检测系统在近几年也已成为研发热点，它通常整合到碰撞预防系统当中，利用雷达摄像头和感应器来检测行人，并及时减速刹车从而减少事故伤害。沃尔沃、丰田等车企已率先推出先进的行人检测系统，而福特也推出了先进的行人检测系统，能够识别路上的行人并进行动态分析，预测他们是否会闯入驾驶路线中。除了传统汽车公司外，很多互联网公司也在研发行人检测系统，以期实现智能汽车。谷歌最新的行人检测系统只靠摄像机影像来掌握行人动向，但是优化了速度问题 。此外，行人检测在智能交通等领域也越来越受到重视。

然而现实生活中由于行人在衣着上的多样性、形态变化的多样性、所处背景的多样性、光照强弱的多样性、行人之间相互遮挡使得行人检测存在着诸多需要解决的问题。

如何设计一个与其他类别之间差异大，行人之间的类内差异小并且不受光照等多样性因素影响的特征，以及如何训练一个区分力强的分类器是行人检测领域的研究重点。近几年来，深度学习在大规模图像分类方面取得的了重大突破，表明深度学可以从多媒体内容中提取具有很强表达能力的特征，从而高效地解决语义鸿沟的问题。

综上所述，建立起一套具有高度准确性，实时性，健壮性的行人检测系统对改变人们生产生活方式有着深远的意义和影响，其背后也蕴藏着巨大的商业价值。

### 国内外发展现状

随着人工智能技术的不断发展，行人检测技术在国内外多个领域中都有所应用，相关产业己经逐渐发展壮大起来。

#### 国外发展现状

近几年随着相关行人检测技术的不断完善与发展，越来越多的领域开始将这一技术应用到实际中，特斯拉作为一款高端电动汽车，最大的优势在于其搭载了一套比较成熟的汽车辅助驾驶系统，包括自动跟车系统，行人规避系统，车道偏移报警系统，前车碰撞预警系统等，将目标检测真正应用了智能驾驶领域中，同时谷歌也表示将开始研究如何能够让自动驾驶汽车识别行人及其手势并进一步理解行人的动作，由于无人驾驶技术的兴起，美国国家公路交通安全管理局也开始正式将自动驾驶汽车进行分类。而欧洲方面也开始修改《维也纳道路交通公约》和《日内瓦道路交通公约》来适应无人车技术的发展。

就在前不久，加州大学圣地亚哥分校电气工程教授NunoVasconcelos宣布，他的团队研宄的新算法，可以以每秒2-4帧的速度对行人的变化来进行检测，能够有效的避免因为行人突然停止移动而导致紧急刹车等突发意外事故的发生。这套系统关键在于其可以对更加极端复杂的道路情况进行辨认与识别，同时与行业内其他类似的系统相比，其算法的出错率只有后者一半，不但拥有出色的计算能力，还能够实现对远处的行人移动进行预先判断。研宄小组还计划通过一系列的升级与改进让系统变的更加完美：即不仅应用于车辆上面，还能扩大适用对象，进一步应用于机器人、安全摄像头等设备上面。“在此之前，没有算法能够在行人检测的准确性和速度上做到优化和平衡，我们通过新算法能得到更好的实时、准确的行人检测结果。”Vasconcelos说道。

但是尽管行人检测技术开始越来越多的投入到智能无人驾驶汽车等各个行业中，但是其带来的问题与事故仍然不容乐观。2018年3月，美国亚利桑那州一女子在深夜不慎被一辆无人驾驶汽车撞倒后身亡，事件发生时被撞女性正在人行横道外的地方横穿马路，尽管车里还配有一名安全观察员，但是仍然没有阻止事故的发生。事件发生后，该汽车公司暂停了在坦佩、匹兹堡、旧金山和多伦多等城市进行的自动驾驶汽车测试。而包括特斯拉在内的自动驾驶事故也层出不穷，这些都暴露出目前行人检测技术的不足，在惨痛的事故背后也不禁引起了人们对自动驾驶这一技术的质疑，但这同时也是促成该领域技术成熟并不断完善的最大动力。如何在更艰苦的光线天气条件下识别行人，或者识别出被部分障碍物遮挡的行人，降低误判率等问题也成为了接下来要解决的一系列技术难题。

#### 国内发展现状

在行人检测领域，国内相关行业也不堪落后，2018年1月8日，全球权威机器视觉算法排行榜KITTI发布，国内著名互联网巨头阿里巴巴集团旗下的iDST夺得行人检测单项冠军。与此同时，在知名的行人再识别数据集Marketl501中，iDST的首位命中率也提升至96.17%，位居世界第一。阿里巴巴iDST团队在行人检测技术方面，提出了基于目标尺寸分级的级联网络，并充分发挥感兴趣区域的上下文信息，提升网络特征提取的能力，以解决行人检测问题中存在的目标尺寸浮动大、遮挡、形变且定位不准等问题；与此同时，在目标定位方面采用交叉熵正则约束来优化边框定位准确度。而在行人再识别方面，团队不仅利用最新的深度学习技术提取行人的全局特征，还提出了超分辨率模块和深度注意力网络来获得头部、躯干、四肢、携带物等局部细节特征，并提出了融合粗粒度全局特征和细粒度局部特征的新方法，进一步提高了跨摄像头场景下行人表征的一致性和行人再识别的准确性。

行人检测与行人识别这两项技术有着丰富的应用场景，包括景区商场的人流预测、人群个性化分析、行人交通安全、无人驾驶、寻找丢失老人儿童等应用等等。目前上述技术己经全部集成到阿里云ET城市大脑当中，并已落地。

同时，国内最大的搜索引擎百度集团也在开始与汽车厂商合作制造无人驾驶汽车，该项目的主导研发单位为百度研究院，以“百度大脑”作为技术核心，能够实现车况行人识别厘米级定位，在17年7月5日举行的百度AI开发者大会，百度CEO李彦宏更是现场展示了其乘坐无人驾驶汽车前往会场的视频。至今日，百度无人驾驶Apollo已经开发到3.5版本。

除了民间科技领域对行人检测系统有着广泛的应用，政府部门也开始将一系列相关技术引入到安全监管中来，天网系统就是一个典型的行人检测系统，中国己经建成世界上最大的视频监控网，视频镜头超过2000万个，整个大工程叫做“中国天网”，不仅在此基础上可以实现行人的检测，还可以进一步实现对行人的人脸进行检测与识别，准确识别出行人的性别，衣着，身份等相关信息。同时还可以和本地的数据库进行对比，不但采用了高科技摄像头，而且受到光照，角度等因素的影响非常小，真正实现对街道，社区等公共场所的24小时无人智能化监控，有效发现治安隐患，使得抓捕犯罪，安全防控工作水平得到进一步的提高。

如今，随着人工智能领域的不断发展不断完善，行人检测技术将进一步改变人们的生产和生活方式，成为人类科技发展中不可缺少的重要一环。

## 

## 第二章 行人检测技术相关理论综述

### 2.1深度学习框架介绍

#### 2.1.1深度学习

深度学习是一种实现机器学习的技术。其本身并不是一种独立的学习方法，也会用到有监督和无监督的学习方法来训练深度神经网络。但由于近几年该领域发展迅猛，一些特有的学习手段相继被提出（如残差网络），因此越来越多的人将其单独看作一种学习的方法。

最初的深度学习是利用深度神经网络来解决特征表达的一种学习过程。深度神经网络本身并不是一个全新的概念，可大致理解为包含多个隐含层的神经网络结构。为了提高深层神经网络的训练效果，人们对神经元的连接方法和激活函数等方面做出相应的调整。其实有不少想法早年间也曾有过，但由于当时训练数据量不足、计算能力落后，因此最终的效果不尽如人意。深度学习，作为目前最热的机器学习方法，但并不意味着是机器学习的终点。起码目前存在以下问题：

1、深度学习模型需要大量的训练数据，才能展现出神奇的效果，但现实生活中往往会遇到小样本问题，此时深度学习方法无法入手，传统的机器学习方法就可以处理；

2、有些领域，采用传统的简单的机器学习方法，可以很好地解决了，没必要非得用复杂的深度学习方法；

3、深度学习的思想，来源于人脑的启发，但绝不是人脑的模拟。

神经网络一般包括训练、测试两大阶段。训练就是把训练数据和神经网络模型（AlexNet、RNN等神经网络，训练框架Caffe等）用CPU或GPU提炼出模型参数的过程。测试就是把测试数据用训练好的模型（神经网络模型+模型参数）运行后查看结果。而Caffe，keras，tensorflow就是把训练过程所涉及的环节数据统一抽象，形成可使用框架。

常见的深度学习框架有 TensorFlow 、Caffe、Theano、Keras、PyTorch、MXNet等，这些深度学习框架被应用于计算机视觉、语音识别、自然语言处理与生物信息学等领域，并获取了极好的效果。本次设计将采用Caffe框架进行模型训练。

#### 2.2.1 Caffe框架

1、概念

Caffe的全称是Convolutional Architecture for Fast Feature Embedding，它是一个清晰而高效的深度学习框架，也是一个被广泛使用的开源深度学习框架，在Tensorflow出现之前一直是深度学习领域Github star最多的项目。主要优势为：上手容易，网络结构都是以配置文件形式定义，不需要用代码设计网络，训练速度快，组件模块化，可以方便的拓展到新的模型和学习任务上。但是Caffe最开始设计时的目标只针对于图像，没有考虑文本、语音或者时间序列的数据，因此Caffe对卷积神经网络的支持非常好，但是对于时间序列RNN，LSTM等支持的不是特别充分。Caffe工程的models文件夹中常用的网络模型比较多，比如Lenet、AlexNet、ZFNet、VGGNet、GoogleNet、ResNet等。

2、Caffe的模块结构

Caffe的核心模块有三个，分别是Blobs、Layers和Nets。Blobs用来进行数据存储、数据交互和处理，通过Blobs，统一制定了数据内存的接口。Layers是神经网络的核心，定义了许多层级结构，它将Blobs视为输入输出。Nets是一系列Layers的集合，并且这些层结构通过连接形成一个网图。下面进行详细介绍：

（1）Blobs本质是一个N维向量，用来存储数据信息，这些数据信息包括图片、深度网络进行前向传输时的数据和反向求梯度过程时的梯度数据等。对于图像数据来说，Blobs通常是一个4维向量，其格式为(Number，Channel，Height，Width)，其中Channel表示图像的通道数，若图像是单通道的灰度图，则Channel=1；若是3通道的RGB图像，则Channel=3。Height和Width分别表示图像的高度和宽度。至于Number则表示图像批块(Batch)，批处理可以使神经网络有更大的吞吐量。

（2）Layers是神经网络的核心，Caffe设计实现了许多层结构，包括卷积、池化、损失等层结构，利用这些层结构可以实现绝大部分的神经网络模型。Layers将下层的数据输出作为输入，进而通过内部运算输出。Layers层的定义和使用一般需要三个步骤：a)建立层，包括建立连接关系和初始化其中一些变量参数；b)前向传输过程，给定输入并计算出相应的输出；c)反向传播过程，进行反向梯度的计算，并把梯度保存在层结构中。

（3）Nets是由层Layers组成的，定义了输入、输出、网络各层，并将各层连接成一个有向无环图（DAG），由此定义了一个网络。一个典型的网络应该有数据输入，并且以一个代价函数作为输出，针对不同的任务，例如分类和重构，应选择不同的代价函数。

3、安装方式

Caffe在不同的操作系统中搭建方法不同，Caffe官网给出了Caffe开发环境搭建的软硬件要求和搭建步骤。但是由于Caffe依赖于许多第三方的工具包，因此Caffe的搭建过程十分复杂，Caffe在64位ubuntu14.04操作系统上的搭建步骤，共六个步骤。

（1）CUDA安装。随着硬件生产技术的发展，GPU的性能和功能都大幅度提升，在计算性能上已经超越了通用的CPU，如此强大的芯片如果只做为显卡就太浪费了。目前，大多数应用和设备都采用CPU和GPU协同工作的模式，充分利用GPU来进行高性能计算。CUDA（Compute Unified Device Architecture）是英伟达公司推出的运算平台，其提供了CUDA指令集架构（ISA）以及GPU内部的并行计算引擎，通过CUDA驱动GPU来进行并行计算。

（2）Open BLAS库安装。Open BLAS是一个基于Goto BLAS2 1.13 BSD版本优化的BLAS库，Caffe需要Open BLAS来提供简易的矩阵和向量运算操作。

（3）Open CV开源视觉库安装。Open CV提供了图像数据操作、基本图像处理和目标识别等功能。

（4）Anaconda Python安装。Anaconda Python是Python科学技术包的合集，其包含的科学计算工具包有：numpy、sicpy、matplotlib、spyder等，提供完全免费的企业级的大规模数据处理、预测分析和科学计算工具。

（5）安装其他依赖项。Caffe还依赖其他工具包，包括leveldb、snappy、hdfs5和gflags等。

（6）编译Caffe。官网下载Caffe工具包并解压，进入Caffe根目录，首先复制一份Makefile.config，并修改其参数。

1. 使用Caffe搭建神经网络流程
2. 数据格式处理。将数据处理成Caffe支持格式，具体包括LEVELDB、LMDB、内存数据、hdfs数据、图像数据、Windows数据、dummy等。
3. 编写网络结构文件。定义网络结构，如当前网络包括哪几层，每层作用是什么，使用Caffe过程中最麻烦的一个操作步骤，具体编写格式可参考Caffe框架自带自动识别手写体样例：caffe/examples/mnist/lenet\_train\_test.prototxt。
4. 编写网络求解文件。定义了网络模型训练过程中需要设置的参数，比如学习率、权重衰减系数、迭代次数、使用GPU还是CPU等，一般命名方式为xx\_solver.prototxt。可参考：caffe/examples/mnist/lenet\_solver.prototxt。
5. 训练。基于命令行的训练，如：caffe train -solver examples/mnist/lenet\_solver.prototxt
6. 测试。caffe test -model examples/mnist/lenet\_train\_test.prototxt -weights examples/mnist/lenet\_iter\_10000.caffemodel -gpu 0 -iterations

在上述流程中，步骤2是核心操作，也是caffe使用最复杂的地方，keras则对该部分做了更高层的抽象，让使用者能够快速编写出自己想要实现的模型。

### 2.2传统行人检测方法综述

传统行人检测方法有三种，分别为基于全局特征的方法、基于人体部位的方法和基于立体视觉的方法。

1、基于全局特征的方法

该类方法是目前较为主流的行人检测方法，主要采用边缘特征、 形状特征、 统计特征或者变换特征等图像的各类静态特征来描述行人，其中代表性的特征包括Haar小波特征、HOG 特征、Edgelet特征、Shapelet特征和轮廓模板特征等。

（1）基于Haar小波特征的方法

PapageorgIoU和Poggio最早提出Haar小波的概念，Viola等引进了积分图的概念，加快了Harr特征的提取速度，并将该方法应用于行人检测，结合人体的运动和外观模式构建行人检测系统，取得了较好的检测效果，为行人检测技术的发展奠定了基础。

（2）基于HOG特征的方法

Dalal和Triggs在2005年提出梯度方向直方图( Histogram of Oriented Gradients，HOG)的概念，并将其用于行人检测，在 MIT行人数据库上获得近乎 100% 的检测成功率；在包含视角、 光照和背景等变化的 INRIA 行人数据库上，也取得了大约 90% 的检测成功率。HOG是目前使用最为广泛的行人特征描述子。

（3）基于edgelet特征的方法

B．Wu等人提出了“小边” ( Edgelet) 特征的概念，即一些短的直线或者曲线片段，并将其应用于复杂场景的单幅图像的行人检测，在 CAVIAR 数据库上取得了大约92%的检测率。

（4）基于Shapelet特征的方法

针对上述Edgelet特征存在的缺点，Sabzmeydani在2007 年提出了一种可以利用机器学习的方法自动得到的特征，即Shapelet特征。该算法首先从训练样本提取图片不同方向的梯度信息，然后利用AdaBoost算法进行训练，从而得到Shapelet特征。

（5）基于轮廓模板的方法

该方法是指利用图像中目标物体的边缘轮廓、纹理和灰度等信息构建模板，通过模板匹配的方法检测目标。Gavrila等较早提出了基于人体边缘轮廓的模板识别方法用以检测行人。

（6）基于运动特征的方法

近年来一些学者尝试将目标的运动信息加入到行人检测系统中，并与其他静态特征相结合用于检测行人。其中较具代表性的算法包括：1) Viola等人针对摄像机静止的情况提出在不同图像上计算Haar－like特征，然后将运动信息与图像的灰度信息相结合构建行人检测系统。2) Dalal等针对摄像机运动的情况，提出将基于外观的梯度描述子和基于运动的差分光流描述子相结合来构建行人检测器，但该方法只对单个窗口的检测比较有效，对于整幅图像检测效果则很差。

2、基于人体部位的方法

该类方法的基本思想是把人体分成几个组成部分，然后对图像中每部分分别检测，最后将检测结果按照一定的约束关系进行整合，最终判断是否存在行人。

3、基于立体视觉的方法

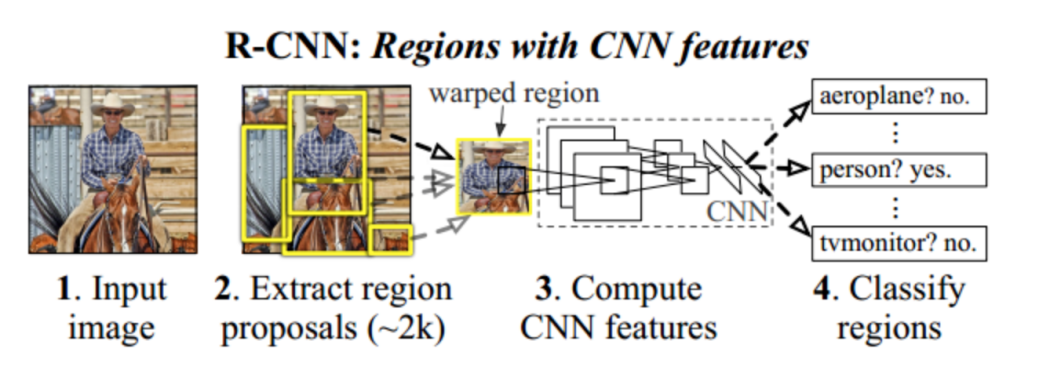
该类方法是指通过2个或2个以上的摄像机进行图像采集，然后分析图像中目标的三维信息以识别出行人。

### 2.3基于深度学习的目标检测方法综述

行人检测技术是目标检测的一种，主流的目标检测算法主要是基于深度学习模型，其可以分成两大类：one -stage检测算法；two-stage检测算法。one-stage检测算法，其不需要region proposal阶段，直接产生物体的类别概率和位置坐标值，经过单次检测即可直接得到最终的检测结果，因此有着更快的检测速度，比较典型的算法如YOLO，SSD，Retina-Net；two-stage算法因其对图片的两阶段处理得名，先由算法生成一系列作为样本的候选框，再通过卷积神经网络进行样本分类，也称为基于区域（Region-based）的方法，比较典型的算法如R-CNN系列。

1. CNN这个领域目前研究非常活跃，先后出现了R-CNN、SPP-net、Fast R-CNN 、Faster R-CNN、R-FCN、Mask R-CNN等研究。Ross Girshick作为这个领域的开山鼻祖总是神一样的存在，R-CNN、Fast R-CNN、Faster R-CNN、Mask R-CNN都和他有关。这些创新的工作其实很多时候是把一些传统视觉领域的方法和深度学习结合起来了，比如选择性搜索（Selective Search)和图像金字塔（Pyramid）等。本文使用的是R-CNN系列中的Faster R-CNN方法进行行人检测，以下将对此系列目标检测方法进行介绍。

#### 2.3.1 R-CNN



**图1 R-CNN结构图**

1. CNN（Region with CNN feature）是卷积神经网络应用于目标检测问题的一个里程碑的飞跃，将CNN与Region Proposal方法进行结合。其中，CNN具有良好的特征提取和分类性能，采用Region Proposal方法实现目标检测问题。

算法可以分为三步，分别为：

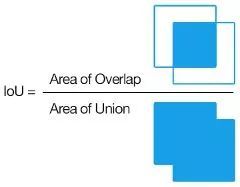
（1）候选区域选择。 Region Proposa是一种传统的区域提取方法，基于启发式的区域提取方法，用的方法是Selective Search算法，查看现有的小区域，合并两个最有可能的区域，重复此步骤，直到图像合并为一个区域，最后输出候选区域。然后将根据建议提取的目标图像标准化，作为CNN的标准输入可以看作窗口通过滑动获得潜在的目标图像，在RCNN中一般Candidate选项为1k~2k个即可，即可理解为将图片划分成1k~2k个网格，之后再对网格进行特征提取或卷积操作，这根据RCNN类算法下的分支来决定。然后基于就建议提取的目标图像将其标准化为CNN的标准输入。

（2）CNN特征提取。标准卷积神经网络根据输入执行诸如卷积或池化的操作以获得固定维度输出，也就是说，在特征提取之后，特征映射被卷积和汇集以获得输出。

（3）分类与边界回归。实际上有两个子步骤，一个是对前一步的输出向量进行分类（分类器需要根据特征进行训练）；第二种是通过边界回归框回归（缩写为bbox）获得精确的区域信息。其目的是准确定位和合并完成分类的预期目标，并避免多重检测。在分类器的选择中有支持向量机SVM，Softmax等等；边界回归有bbox回归，多任务损失函数边框回归等 。

R-CNN算法的出现为目标检测做出了两大贡献，一是CNN可用于基于区域的定位和分割物体，影响了之后几乎所有two-stage方法；二是监督训练样本数紧缺时，在额外的数据上预训练的模型经过fine-tuning可以取得很好的效果，这种用分类任务（Imagenet）中训练好的模型作为基网络，在检测问题上fine-tuning的做法也在之后的工作中一直沿用。传统的计算机视觉方法常用精心设计的手工特征(如SIFT, HOG)描述图像，而深度学习的方法则倡导习得特征，从图像分类任务的经验来看，CNN网络自动习得的特征取得的效果已经超出了手工设计的特征。R-CNN在局部区域应用卷积网络，以发挥卷积网络学习高质量特征的能力。

另外，R-CNN中的两个做法值得注意。一是对IoU的计算，如图2所示，IoU计算了两个区域之交的面积跟它们之并的比，描述了两个区域的重合程度。在输入CNN前，需要根据Ground Truth对提出的Region Proposal进行IoU标记。文章中特别提到，IoU阈值的选择对结果影响显著，一个是用来识别正样本的阈值，如跟ground truth的IoU大于0.5，另一个用来标记负样本即背景的阈值，如IoU小于0.1，而介于两者之间的则为难例（Hard Negatives），若标为正类，则包含了过多的背景信息，反之又包含了要检测物体的特征，因而这些Proposal便被忽略掉。



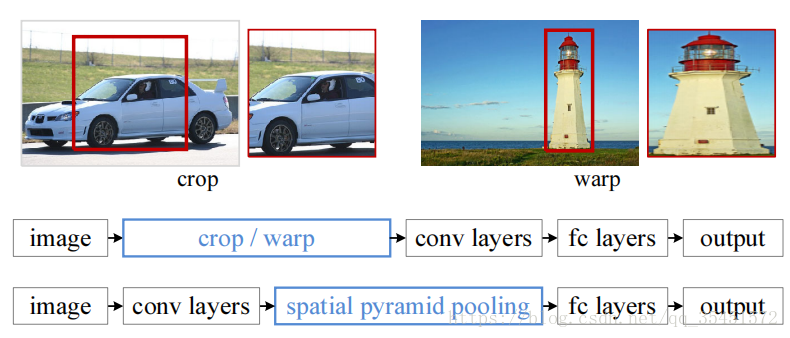
**图2 IoU**

另一值得注意的做法是位置坐标的回归（Bounding-Box Regression），这一过程是Region Proposal向Ground Truth调整，实现时加入了log/exp变换来使损失保持在合理的量级上，可以看做一种标准化（Normalization）操作。

当然，R-CNN算法同样存在着缺点，在R-CNN刚刚被发明出来的2014年，RCNN在目标检测与行人检测上取得了巨大的成就，然而效率低下，花费时间长等一系列的问题的产生，还是导致了R-CNN的运用并没有取得大范围的应用，其最大的问题有三：首先需要事先提取多个候选区域对应的图像，这一行为会占用大量的磁盘空间；另外针对传统的CNN来说，输入的map需要时固定尺寸的，而归一化过程中对图片产生的形变会导致图片大小改变，这对CNN的特征提取有致命的坏处；而且每个region proposal都需要进入CNN网络计算，会导致过多次的重复的相同的特征提取，导致大大的计算浪费。在这之后，随之而来的Fast R-CNN逐渐进入了人们的眼帘。

#### 2.3.2 SPP-NET

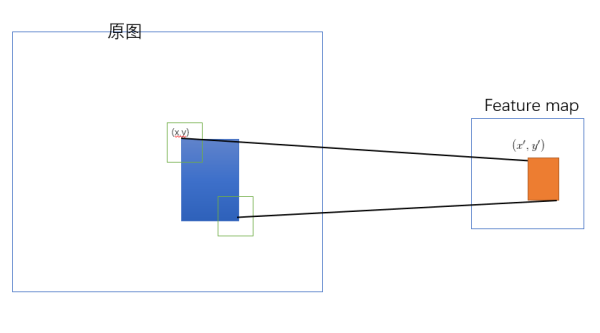
SPP-NET是MSRA何恺明等人提出的，其主要思想是去掉了原始图像上的crop/warp等操作，换成了在卷积特征上的空间金字塔池化层（Spatial Pyramid Pooling，SPP），如图3所示。为何要引入SPP层 ，主要原因是CNN的全连接层要求输入图片是大小一致的，而实际中的输入图片往往大小不一，如果直接缩放到同一尺寸，很可能有的物体会充满整个图片，而有的物体可能只能占到图片的一角。传统的解决方案是进行不同位置的裁剪，但是这些裁剪技术都可能会导致一些问题出现，比如图3中的crop会导致物体不全，warp导致物体被拉伸后形变严重，SPP就是为了解决这种问题的。在R-CNN中，每个候选框先resize到统一大小，然后分别作为CNN的输入，这样是很低效的。所以SPP-NET根据这个缺点做了优化,SPP对整图提取固定维度的特征，再把图片均分成4份，每份提取相同维度的特征，再把图片均分为16份，以此类推。可以看出，无论图片大小如何，提取出来的维度数据都是一致的，这样就可以统一送至全连接层了。在这个过程中，只对原图进行一次卷积得到整张图的feature map，然后找到每个候选框在feature map上的映射patch，将此patch作为每个候选框的卷积特征输入到SPP layer和之后的层。节省了大量的计算时间，比R-CNN有一百倍左右的提速。



**图3 传统crop/warp结构和空间金字塔池化网络的对比**

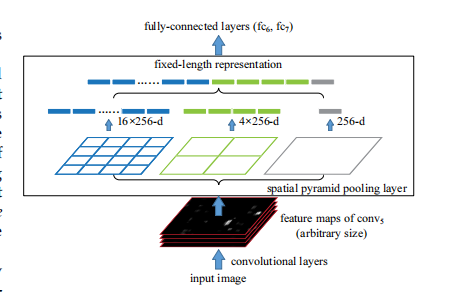
**ROI映射到feature map**

ROI(region of interest)即上文提到的候选框，这个想法的具体内容是，通过Selective Search选择出来的2000个候选框输入CNN计算特征速度太慢，为什么不把整张图直接输入CNN，得到一整张图的特征，然后根据各个候选框的位置从那个一整张图的特征中crop对应的特征，再输入SPP进行运算呢，后来经过这般改进后，准确度没有受影响，而速度提高了好多好多，如图4所示



**图4 ROI映射到feature map**

SPP-NET的网络结构如图5所示，实质是最后一层卷积层后加了一个SPP层，将维度不一的卷积特征转换为维度一致的全连接输入。



**图5 SPP-NET网络结构**

SPP-NET做目标检测的主要步骤为：

（1）区域提名：用Selective Search从原图中生成2000个左右的候选窗口；

（2）区域大小缩放：SPP-net不再做区域大小归一化，而是缩放到min(w, h)=s，即统一长宽的最短边长度，s选自{480,576,688,864,1200}中的一个，选择的标准是使得缩放后的候选框大小与224×224最接近；

（3）特征提取：利用SPP-net网络结构提取特征；这一步就是和R-CNN最大的区别了，同样是用卷积神经网络进行特征提取，但是SPP-Net用的是金字塔池化。这一步骤的具体操作如下：把整张待检测的图片，输入CNN中，进行一次性特征提取，得到feature maps，然后在feature maps中找到各个候选框的区域，再对各个候选框采用金字塔空间池化，提取出固定长度的特征向量。而R-CNN输入的是每个候选框，然后在进入CNN，因为SPP-Net只需要一次对整张图片进行特征提取，速度是大大地快啊。江湖传说可一个提高100倍的速度，因为R-CNN就相当于遍历一个CNN两千次，而SPP-Net只需要遍历1次。

（4）分类与回归：类似R-CNN，利用SVM基于上面的特征训练分类器模型，用边框回归来微调候选框的位置。

**SPP-NET缺点**：SPP已有一定的速度提升，它在ConvNet的最后一个卷积层才提取proposal，但是依然有不足之处。

（1）和RCNN一样，训练过程仍然是隔离的，提取候选框 | 计算CNN特征| SVM分类 | Bounding Box回归独立训练，大量的中间结果需要转存，无法整体训练参数；

（2）SPP-NET在无法同时Tuning在SPP-Layer两边的卷积层和全连接层，很大程度上限制了深度CNN的效果；

（3）在整个过程中，Proposal Region仍然很耗时。

**SPP-NET创新点**：

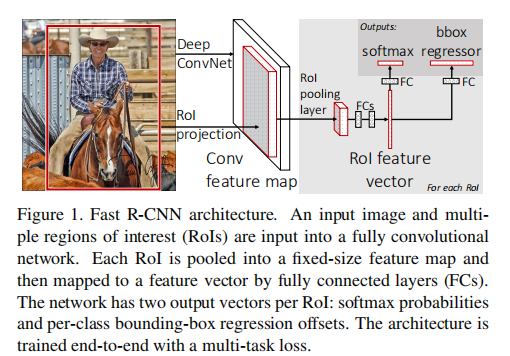
利用空间金字塔池化结构；对整张图片只进行了一次特征提取，加快运算速度。

#### 2.3.3 Fast R-CNN

即使使用了selective search等预处理步骤来提取潜在的bounding box作为输入，但是R-CNN仍会有严重的速度瓶颈，原因也很明显，R-CNN 需要非常多的候选区域以提升准确度，但其实有很多区域是彼此重叠的，因此 R-CNN 的训练和推断速度非常慢。如果我们有 2000 个候选区域，且每一个都需要独立地馈送到 CNN 中，那么对于不同的 ROI，我们需要重复提取 2000 次特征。Fast R-CNN正是为了解决这个问题诞生的。

R-CNN的进阶版Fast R-CNN就是在R-CNN的基础上采纳了SPPNET方法，对R-CNN作了改进，使得性能进一步提高。

Fast R-CNN借鉴了SPPNET的做法，在全图上进行卷积，然后采用ROI-pooling得到定长的特征向量，例如不管窗口大小是多少，转换成7x7这么大。Fast R-CNN还引入了一个重要的策略，在对窗口进行分类的同时，还会对物体的边框进行回归，使得检测框更加准确。前面我们说候选窗口会有非常高的召回率，但是可能框的位置不是很准，例如一个人体框可能是缺胳膊缺腿，那么通过回归就能够对检测框进行校准，在初始的位置上求精。Fast R-CNN把分类和回归放在一起来做，采用了多任务协同学习的方式。

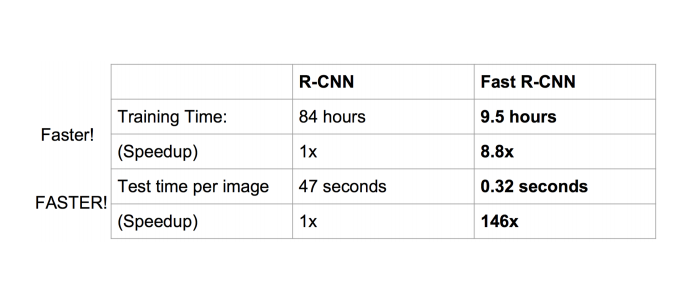


**图6 Fast R-CNN结构图**

Fast R-CNN的主要步骤如下：

1. 特征提取：以整张图片为输入利用CNN得到图片的特征层；
2. 区域提名：通过Selective Search等方法从原始图片提取区域候选框，并把这些候选框一一投影到最后的特征层；
3. 区域归一化：针对特征层上的每个区域候选框进行RoI Pooling操作，得到固定大小的特征表示；
4. 分类与回归：然后再通过两个全连接层，分别用softmax多分类做目标识别，用回归模型进行边框位置与大小微调。

所以容易看见，Fast R-CNN相对于R-CNN的提速原因就在于：不像R-CNN把每个候选区域给深度网络提特征，而是整张图提一次特征，再把候选框映射到conv5上，而SPP只需要计算一次特征，剩下的只需要在conv5层上操作就可以了。在性能上提升也是相当明显的,如图7所示

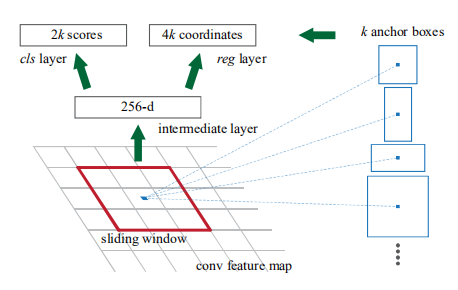


**图7 R-CNN与Fast R-CNN性能对比**

#### 2.3.4 Faster R-CNN

Fast R-CNN使用Selective Search来进行区域提名，速度依然不够快。为了解决Fast R-CNN算法缺陷，使得算法实现two stage的全网络结构，2015年微软研究院的任少庆、何恺明以及Ross B Girshick等人又提出了Faster R-CNN算法。设计辅助生成样本的RPN（Region Proposal Networks）网络，将算法结构分为两个部分，先由RPN网络判断候选框是否为目标，再经分类定位的多任务损失判断目标类型，整个网络流程都能共享卷积神经网络提取的的特征信息，节约计算成本，且解决Fast R-CNN算法生成正负样本候选框速度慢的问题，同时避免候选框提取过多导致算法准确率下降。

Faster R-CNN则直接利用RPN（Region Proposal Networks）网络来计算候选框。RPN以一张任意大小的图片为输入，输出一批矩形区域提名，每个区域对应一个目标分数和位置信息。Faster R-CNN中的RPN结构如图8所示。

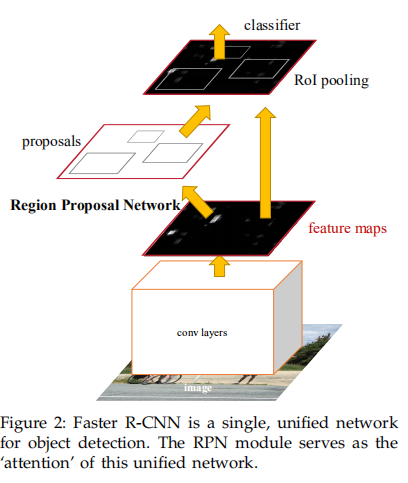


**图8 Region Proposal Network(RPN)**

Faster R-CNN的主要步骤如下：

1. 特征提取：同Fast R-CNN，以整张图片为输入，利用CNN得到图片的特征层；
2. 区域提名：在最终的卷积特征层上利用k个不同的矩形框（Anchor Box）进行提名，k一般取9；
3. 分类与回归：对每个Anchor Box对应的区域进行object/non-object二分类，并用k个回归模型（各自对应不同的Anchor Box）微调候选框位置与大小，最后进行目标分类。

总之，Faster R-CNN抛弃了Selective Search，引入了RPN网络，使得区域提名、分类、回归一起共用卷积特征，从而得到了进一步的加速。但是，Faster R-CNN需要对两万个Anchor Box先判断是否是目标（目标判定），然后再进行目标识别，分成了两步。



**图9 Faster R-CNN**

**算法流程：**

1. 把整张图片送入CNN，进行特征提取；
2. 在最后一层卷积feature map上生成region proposal（通过RPN），每张图片大约300个建议窗口；
3. 通过RoI pooling层（其实是单层的SPP layer）使得每个建议窗口生成固定大小的feature map；
4. 继续经过两个全连接层（FC）得到特征向量。特征向量经由各自的FC层，得到两个输出向量。第一个是分类，使用softmax，第二个是每一类的bounding box回归。利用SoftMax Loss和Smooth L1 Loss对分类概率和边框回归（Bounding Box Regression）联合训练。

**创新点：**

（1）采用RPN(Region Proposal Network)代替选择性搜索(Selective Search)，利用GPU进行计算大幅度缩减提取region proposal的速度。

（2）产生建议窗口的CNN和目标检测的CNN共享。

### 2.4 Open CV技术简介

OpenCV是一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效，由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。

OpenCV用C++语言编写，它的主要接口也是C++语言，但是依然保留了大量的C语言接口。该库也有大量的Python、Java and MATLAB/OCTAVE（版本2.5）的接口。这些语言的API接口函数可以通过在线文档获得。如今也提供对于C#、Ch、Ruby、GO的支持。

所有新的开发和算法都是用C++接口。一个使用CUDA的GPU接口也于2010年9月开始实现。

OpenCV致力于真实世界的实时应用，通过优化的C代码的编写对其执行速度带来了可观的提升，并且可以通过购买Intel的IPP高性能多媒体函数库（Integrated Performance Primitives）得到更快的处理速度。

### 2.5特征提取

本次用到的行人检测算法Faster R-CNN第一步要使用在图片分类任务（例如ImageNet）上预训练好的卷积神经网络VGG-16，使用该网络得到中间层特征的输出。虽然原始的Faster R-CNN使用的是在ImageNet上预训练的ZF和VGG，但是在这之后又出现了很多不同的网络，并且不同网络之间的参数数量变化特别大。例如，MobileNet，以速度优先的一个小型的高效框架，大约有330万个参数，而ResNet-152（152层），曾经的ImageNet图片分类竞赛优胜者，大约有6000万个参数。还有的网络结构如DenseNet，可以在提高准确度的同时缩减参数数量。

本节将首先介绍基于卷积神经网络的图像特征提取，然后以VGG-16网络为例来阐述Faster R-CNN的特征输出。

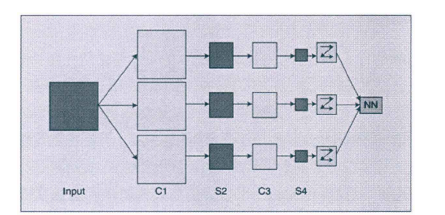
#### 2.5.1卷积神经网络特征提取

卷积神经网络（CNN）是局部连接网络。相对于全连接网络其最大的特点就是局部连接性和权值共享性。因为对一副图像中的某个像素p来说，一般离像素p越近的像素对其影响也就越大（局部连接性）；另外，根据自然图像的统计特性，某个区域的权值也可以用于另一个区域（权值共享性）。这里的权值共享说白了就是卷积核共享，对于一个卷积核将其与给定的图像做卷积就可以提取一种图像的特征，不同的卷积核可以提取不同的图像特征。概况的讲，卷积层的计算方法就是根据公式（2.1）

 （2.1）

其中“σ”表示激活函数；“imgMat”表示灰度图像矩阵；“W”表示卷积“。”表示卷积操作；“b”表示偏置值。

如图10所示，卷积神经网络是一个多层的神经网络，每层由多个二维平面组成，而每个平面由多个独立神经元组成。输入图像通过和三个可训练的滤波器和可加偏置进行卷积，卷积后在C1层产生三个特征映射图，然后特征映射图中每组的四个像素再进行求和，加权值，加偏置，通过一个Sigmoid函数得到三个S2层的特征映射图。这些映射图再进过滤波得到C3层。这个层级结构再和S2—样产生S4。最终，这些像素值被光栅化，并连接成一个向量输入到传统的神经网络，得到输出。一般地，C层为特征提取层，每个神经元的输入与前一层的局部感受野相连，并提取该局部的特征，一旦该局部特征被提取后，它与其他特征间的关系也随之确定下来。



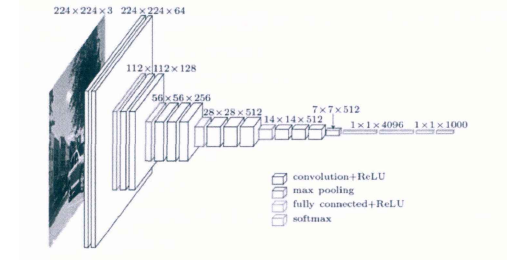
**图10 CNN架构图**

S层则是特征映射层，网络的每个计算层由多个特征映射组成，每个特征映射为一个平面，平面上所有神经元的权值相等。特征映射结构采用影响函数核小的Sigmoid函数作为卷积网络的激活函数，使得特征映射具有位移不变性。

此外，由于一个映射面上的神经元共享权值，因而减少了网络自由参数的个数，降低了网络参数选择的复杂度。卷积神经网络中的每一个特征提取层（C-层）都紧跟着一个用来求局部平均与二次提取的计算层（S-层），这种特有的两次特征提取结构使网络在识别时对输入样本有较高的畸变容忍能力。

#### 2.5.2 VGG-16特征提取

首先详细介绍一下VGG-16的网络结构。如图11所示，当使用VGG进行特征提取任务时，其输入是224\*224\*3的张量（表示一个224\*224像素大小的RGB图片）。唯一的预处理是从每个像素中减去在训练集上计算的RGB均值。图像通过一堆卷积（conv）层，使用感受野很小的滤波器：3\*3（这是捕获左/右，上/下，中心概念的最小尺寸）。在其中一种配置中，同时还使用了1\*1卷积滤波器，可以看作输入通道的线性变换（后面是非线性）。卷积步长固定为1个像素；卷积层输入的空间填充要满足卷积之后保留空间分辨率，即3\*3卷积层的填充为1个像素。空间池化由五个最大池化层进行，这些层在一些卷积层之后（不是所有的卷积层之后都是最大池化）。在2\*2像素窗口上进行最大池化，步长为2。在一堆卷积层（在不同架构中具有不同深度）之后是三个全连接（FC）层：前两个每个都有4096个通道，第三个执行1000维ILSVRC分类，因此包含1000个通道（一个通道对应一个类别）。最后一层是soft-max层。所有网络中全连接层的配置是相同的。



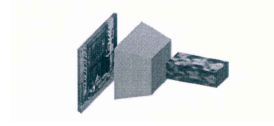
**图11 VGG网络结构图**

在分类任务中输入图片的尺寸是固定的，因为网络最后一部分的全连接层需要固定长度的输入。在接入全连接层前，要将最后一层卷积的输出展开成一维张量。

因为要使用卷积网络中间层的输出，所以输入图片的尺寸不再有限制。至少在这个模块中不再是问题，因为只有卷积层参与计算。最后Faster R-CNN使用conv4/con5\_1这一层作为卷积网络的输出。

每一层卷积网络都在前一层的信息基础上提取更抽象的特征。第一层学习到简单的边缘，第二层寻找目标边缘的模式，以激活后续卷积网络中更加复杂的形状。最终得到一个在空间维度上比原始图片小很多，但表征更深的卷积特征图。特征图的长和宽会随着卷积层间的池化而缩小，深度会随着卷积层滤波器的数量而增加。

如图12所示，卷积特征图将图片的所有信息编码到深度的维度上，同时保留着原始图片上目标物体的相对位置信息。例如，如果图片左上角有一个红色矩形，经过卷积层的激活，那么红色矩形的位置信息仍然保留在卷积特征图的左上角。



**图12 卷积特征图的生成**

## 第三章 行人检测算法的设计与实现

### 3.1行人检测算法的设计

主要论述行人检测算法的整体设计和详细设计原理，对行人检测中的整个目标提取过程进行具体介绍。

#### 3.1.1算法总体设计

本次基于深度学习的行人检测系统采用的检测方法是Faster R-CNN。如图13所示整体流程，对于每一张输入的图片，首先会将其输入到CNN网络中进行特征提取来生成卷积特征图。然后由区域建议网络（RPN）采用生成的卷积特征图在图像上生成区域建议，同时输出目标性得分（只关心建议出来的是不是目标，不关心其目标的类别；用来对不好的预测结果进行过滤，为下一步做准备），然后进行边框回归(调整锚点以更好的与目标进行拟合)。这里使用RPN的原因在于其只使用卷积层的网络，非常简单，所以生成预测的速度特别的快，同时和基础的网络相比，也会加快训练的时间，这也是Faster R-CNN和前几代目标检测算法的改进之处。

在前面的区域建议网络生成区域建议之后，剩下的工作就是分别对这些边框建立类别并将其进行类别匹配，把这些边框进一步分配到想要的类别之中。传统方法是将其中每个建议一个一个地裁剪出来，使用预先训练的网络，将提取出来的特征输到图像分类器中，但是对于运行所有建议来说无疑会导致运算速率降低，大大浪费运算的时间，降低运算效率。为了对这种方法进行改进，Faster R-CNN引入了兴趣区域池化。

在兴趣区域池化之后就到了Faster R-CNN算法工作流程的最后一步，即通过Faster R-CNN网络进行类别匹配，它的作用在于为之前生成的区域目标类别建立分数。将每个建议分配到一个类中并生成背景类(背景类的意义是将不好的区域建议进行删除)。同时根据预测的类别进一步调整建议的边框。

Fast R-CNN和RPN计算目标的方法在本质上来说几乎是一样的，但是其计算的目标对象是不一样的。对于R-CNN来说，将采用之前生成的建议以及真实的边框情况，经过计算得出其之间的IoU值，简而言之，就是计算出来最终建议出来的目标对象是不是其对应的类别。而RPN计算得出得分数则用来衡量建议出来的是不是目标对象，并不关心其类别。

IoU值以0.5分作为其分界点。对于真实边框与建议，当其分数大于0.5分时，将会把这些边框分别分配给相应的类别。分数在0.1~0.5之间的则会被标为背景。



**图13 Faster R-CNN总体流程图**

当边框与类别进行匹配之后，就是Faster R-CNN的最后一个职能了——边框回归。顾名思义，就是计算目标和真实边框之间的偏移量，最后将精校好的目标进行输出。至此，Faster R-CNN算法的总体架构基本就介绍完毕了。

#### 3.1.2算法详细设计

对于一张输入到网络的P\*Q的图像，先将其缩放至固定大小M\*N(长边不超过1000，短边不超过600)，然后将缩放后的图像输入至Conv Layer层中进行特征提取，在Conv Layer中一共包含了13个cony层+13个relu层+4个pooling层。最后一个特征输出层为conv5.3，特征数(channels)为512。其中，13个relu层为激活函数，不改变图片大小；pooling层会让输出图片是输入图片的1/2。经过Conv layers层处理后，图片大小变成(M/16)\*(N/16)，即：60\*40(1000/16≈60，600/16≈-40)；Feature Map则是60\*40\*512-d(注：VGGl6是512-d，ZF是256-d)，表示特征图的大小为60\*40，数量为512。

然后RPN网络在特征图conv5-3上执行3\*3卷积操作，后接一个512维的全连接层，主要负责生成锚并对锚进行分类和回归，然后输入到建议层，该层主要负责综合计算并获取最终的目标建议。步骤主要包括再次生成锚并对所有锚进行回归和排序，取排序前6000的锚将其映射回原图以判断其是否超出边界，并采用非极大值抑制算法精校候选框的位置，之后再次进行得分排序，取得分前300的候选框为目标建议。

在得到目标建议之后，将目标建议输入RoI池，RoI池则提取目标建议特征进行池化操作以计算特征，然后送入后续网络做分类和回归。

虽然RPN网络和Faster R-CNN网络中均有分类和回归，但两者有所不同，RPN中分类是判断conv5-3中对应的锚属于目标和背景的概率(score)，并通过回归获取锚的偏移和缩放尺度，根据目标得分值筛选用于后续检测识别的目标建议；Faster R-CNN是对RPN网络提取的目标建议做分类识别，并通过回归参数调整得到目标的精确位置。

其主要步骤如下：

1. Conv layers作为一种can网络目标检测的方法，Faster R-CNN首先使用一组基础conv+relu+pooling层提取图片的特征。该特征被共享用于后续的RPN层和全连接层。
2. RPN层是Faster R-CNN最大的亮点，RPN网络用于生成目标区域建议，该层通过softmax判断锚属于前景(目标)或者背景(非目标)，再利用边框回归修正锚获得精确的建议。
3. RoI池收集输入的特征和建议，综合这些信息提取建议特征，送入后续的全连接层判定目标类别。
4. 利用建议特征计算建议类别，同时再次通过边框回归获得检验框的最终精确地位置。

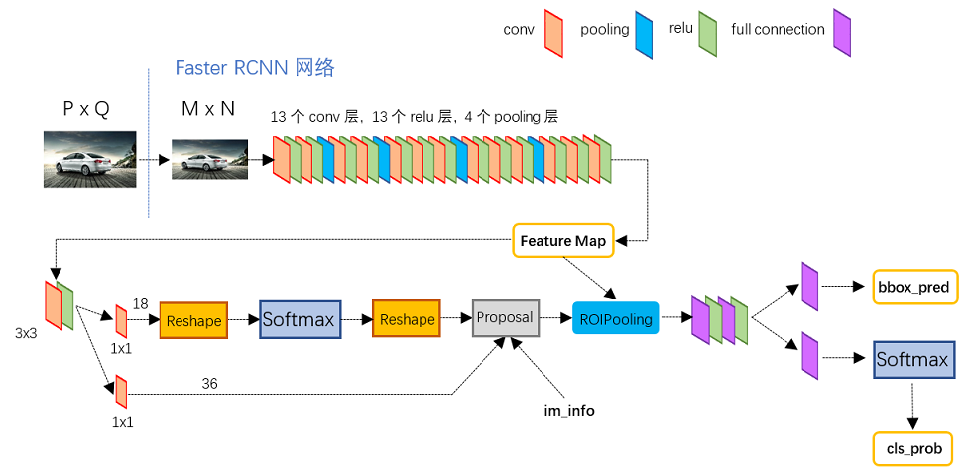
整个Faster R-CNN算法详细流程如图14所示。

#### 3.1.3此算法主要设计原理

Faster R-CNN的网络结构图如图15所示。



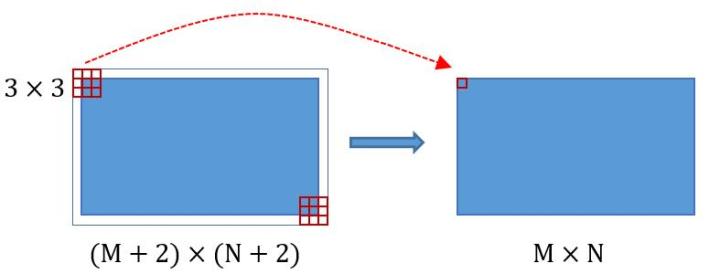
**图14 Faster R-CNN算法详细流程图**



**图15 Faster R-CNN的网络结构图**

1、Conv layers特征提取原理

Convlayers包含了cony，pooling，relu三种层。Conv layers部分共有13个conv层，13个relu层，4个pooling层。其中非常重要的在于，在Conv layers中：所有的conv层都是：kernel\_size=3，pad=l，stride=1；所有的pooling层都是：kernel size=2，stride=2；如此重要的原因在于在Faster RCNN在Conv layers中对所有的卷积都做了扩边处理(pad=l，即填充一圈0)，导致原图变为(M+2)\*(N+2)大小，再做3x3卷积后输出M\*N。正是这种设置，导致Conv layers中的conv层不改变输入和输出矩阵大小，如图16所示。



**图16 Conv layers原理**

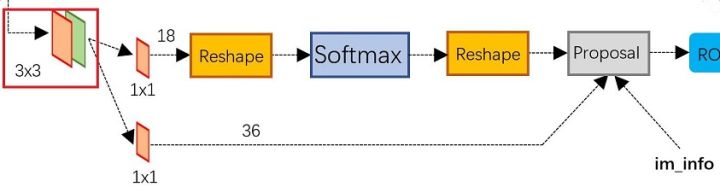
同理，Conv layers中的pooling层kernel-size=2，stride=2。这样每个经过pooling层的M\*N矩阵，都会变为(M/2)\*(N/2)大小。综上所述，在整个Conv layers中，conv和relu层不改变输入输出大小，只有pooling层使输出长宽都变为输入的1/2。这样使得一个M\*N大小的矩阵经过Conv layers固定变为(M/1 6)\*(N/1 6)。这样Conv layers生成的特征中都可以和原图对应起来。

2、区域建议网络(RPN)生成原理

经典的检测方法生成检测框都非常耗时，如OpenCV adaboost使用滑动窗口+图像金字塔生成检测框；或如RCNN使用SS(Selective Search)方法生成检测框。而Faster R-CNN则抛弃了传统的滑动窗口和SS方法，直接使用RPN生成检测框，这也是Faster R-CNN的巨大优势，能极大提升检测框的生成速度。

RPN网络实际分为2条线，一条通过softmax分类锚获得前景(即目标)和背景(检测目标是否是背景)，另外一条用于计算对于锚的边框回归偏移量，以获得精确的建议。RPN具体结构网络如图17所示。

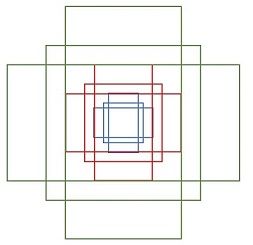
最后的建议层则负责综合前景锚和边框回归偏移量获取建议，同时剔除太小和超出边界的建议。其实整个网络到了建议层这里，就完成了相当于目标定位的功能。关于这一步的具体设计流程如下：



**图17 RPN网络结构**

1. 锚的生成(anchors)

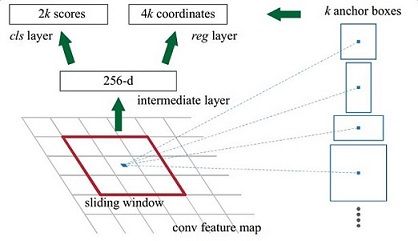
锚，实际上就是一组由rpn/generate\_anchors.py生成的矩形，主要代码结构是[x1，y1, x2，y2]。其中括号内的4个值代表矩形左上和右下角点坐标。如图18，一共生成了9个矩形，总共有3种形状，长宽比为大约为：宽：高=[1：1，1：2，2：1]三种，实际上通过锚就引入了检测中常用到的多尺度方法。



**图18 锚示意图**

上面的锚的尺度问题，其实是根据检测图像设置的。先把任意大小的输入图像reshape成800\*600。再回头来看锚的大小，锚中长宽1：2中最大为352x704，长宽2：1中最大736x384，基本是覆盖了800x600的各个尺度和形状。

如图19生成的9个锚的主要用途在于遍历Conv layers计算获得的特征，为每一个点都配备这9种锚作为初始的检测框。虽然这样导致最后获得的检测框很不准确，但是后面还有2次边框回归可以修正检测框位置。



**图19锚用途示意图**

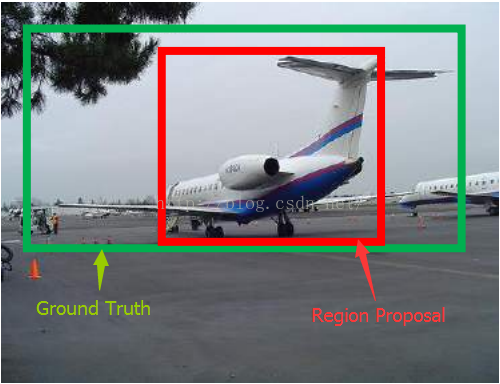
（2）softmax判定前景与背景

一副M\*N大小的矩阵送入Faster RCNN网络后，到RPN网络后变为(M/16)\*(N/16)，设W=M/16，H=N/16。在进入重塑层与softmax之前，先做了l\*l卷积，经过该卷积的输出图像为W\*H\*l8大小，这也就刚好对应了特征图每一个点都有9个锚，同时每个锚又有可能是前景和背景，所有这些信息都保存W\*H\*(9\*2)大小的矩阵。这样做的原因在于后面接softmax分类获得前景锚，也就相当于初步提取了检测目标候选区域box(一般认为目标在前景锚中)。softmax前后都接一个重塑层的原因则其实只是为了便于softmax分类，至于具体原因这就要从caffe的实现形式说起了。在caffe基本数据结构blob中以如下形式保存数据：blob=[batch\_size，channel，height，width] 。

对应至上面的保存bg/fg anchors的矩阵，其在caffe blob中的存储形式为[1, 2\*9, H, W]。而在softmax分类时需要进行fg/bg二分类，所以reshape layer会将其变为[1, 2, 9\*H, W]大小，即单独“腾空”出来一个维度以便softmax分类，之后再reshape回复原状。综上所述，RPN网络中利用锚和softmax初步提取出了前景锚作为候选区域。

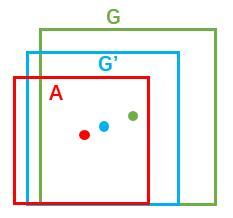
（3）边框回归原理(bounding box regression)

在候选框生成之后，接下来就是边框回归，先简要介绍以下边框回归的具体原理，如图20所示，最终目的是对飞机进行目标提取。其中绿色框为飞机的Ground Truth(GT)，也就是理想中的目标提取候选框。但是红色为实际提取的前景锚，所以即便红色的框被分类器识别为飞机，但是由于红色的框定位不准，那么这张图仍然相当于没有正确的检测出飞机。所以希望采用一种方法对红色的框进行微调，使得前景锚和GT更加接近。以此来进一步对边框进行校正与调整，以此来输出正确的标记结果，这时候就用到了边框回归。接下来将对边框回归的原理进行具体介绍，包括边框的平移，缩放以及线性回归，最后得出函数的优化目标以精校候选框。



**图20 边框回归原理**

对于窗口一般使用四维向量(x，y，w，h)表示，分别表示窗El的中心点坐标和宽高。对于图21，红色的框A代表原始的ForegroundAnchors，绿色的框G代表目标的GT，我们的目标是寻找一种关系，使得输入原始的anchorA经过映射得到一个跟真实窗口G更接近的回归窗口G’，即：给定A=(Ax，Ay,Aw,Ah)，寻找一种映射f，使得f(Ax，Ay,Aw,Ah)=(G’x，G’y，G’w，G’h)，其中(G’x，G’y，G’w，G’h)≈(Gx，Gy,Gw,Gh)。



**图21窗口示意图**

将图21的A变为G’的过程如下：

先平移，见公式（3.1）：

 ， （3.1）

再缩放，见公式（3.2）：

， （3.2）

上面4个公式发现，需要学习的是，，，这四个变换。当输入的anchor与GT相差较小时，可以认为这种变换是一种线性变换， 那么就可以用线性回归来建模对窗口进行微调(注意，只有当anchors和GT比较接近时，才能使用线性回归模型，否则就是复杂的非线性问题了)。平移量与尺度因子见公式（3.3）：

，

， （3.3）

接下来的问题就是如何通过线性回归获得，，，了。线性回归就是给定输入的特征向量X，学习一组参数W，使得经过线性回归后的值跟真实值Y(即GT)非常接近，即Y=WX。对于该问题，输入X是一张经过num 0utput=l的lxl卷积获得的特征图，定义为φ；同时还有训练传入的GT，即。输出是，，，四个变换。那么目标函数可以表示为公式（3.4）：

 (3.4)

其中是对应锚的特征图组成的特征向量，W是需要学习的参数，d(A)是得到的预测值(\*表示X，Y，W，h，也就是每一个变换对应一个上述目标函数)。为了让预测值与真实值最小，得到损失函数见公式（3.5）：

 （3.5）

函数优化目标见公式（3.6）：

 （3.6）

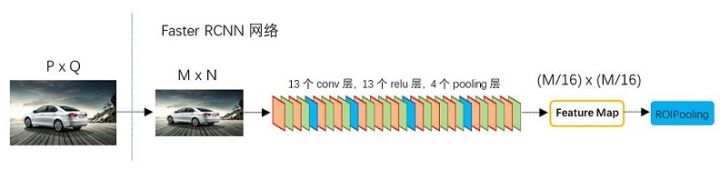
在得出函数的优化目标后，经过l\*l卷积输出后的图像为W\*H\*36，而在caffe Blob存储为[1，36，H，W]，所以这里相当于特征图每个点都有9个锚，每个锚又有4个用于回归的变换量。

（4）建议层(Proposal layer)原理

建议层负责综合所有变换量和前景锚，计算出精准的建议，送入后续RoI池。

建议层有3个输入：fg/bg锚分类器结果rpn\_cls\_probreshape，对应的bbox reg的变换量rpn\_bbox\_pred，以及im\_info；另外还有参数feat\_stride=16.

这里的im\_info。对于一副任意大小P\*Q图像，传入Faster RCNN前首先reshape到固定M\*N，im\_info=[M，N，scalefactor]保存了此次缩放的所有信息。然后经过Conv Layers，经过4次poolmg变为W\*H=(M/16)\*(N/16)大小，其中feature—stride=16则保存了该信息。所有这些数值都是为了将proposal映射回原图而设置的，如图22所示。



**图22特征处理**

Proposal Layer forward(caffe layer的前传函数)按照以下顺序依次处理：

①再次生成锚，并对所有的锚做边框位置回归(注意这里的锚生成顺序和之前是即完全一致的)。

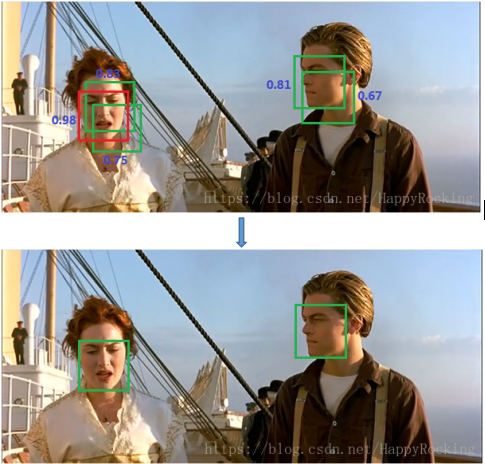
②按照输入的前景softmax scores由大到小排序锚，提取前pre\_nms\_topN(e.g.6000)个锚。即提取修正位置后的前景锚。

③利用feat\_stride和im\_info将锚映射回原图，判断前景锚是否大范围超过边界，剔除严重超出边界前景锚。

④进行nms(nonmaximum suppression，非极大值抑制)，这里将对非极大值抑制的具体原理进行阐述。

非极大值抑制顾名思义就是抑制不是极大值的元素，搜索局部的极大值。这个局部代表的是一个邻域，邻域有两个参数可变，一是邻域的维数，二是邻域的大小。这里不讨论通用的NMS算法，而是用于在目标检测中用于提取分数最高的窗口的。在行人检测中，滑动窗口从提取特征开始，再通过分类器进行分类识别以后，每个窗口都会得到一个分数。但是这种滑动窗口会导致大量的窗口之间彼此重合，包含或者大面积的交叉。这时候就需要用到非极大值抑制来选取那些邻域里分数最高的，即将行人概率最大的窗口选出来。并且对那些分数值较低的窗口进行抑制。

如图23所示，应用非极大值抑制的主要目的在于消除多余的检测框以便确定人体所在的最佳位置。



**图23非极大值抑制**

主要原理是：

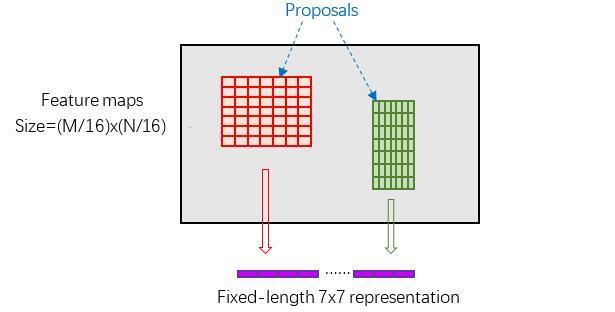
先假设有6个矩形框，根据分类器类别分类概率做排序，从小到大分别属于车辆的概率分别为A、B、C、D、E、F。从最大概率矩形框F开始，分别判断A～E与F的重叠度IOU是否大于某个设定的阈值；假设B、D与F的重叠度超过阈值，那么就扔掉B、D；并标记第一个矩形框F同时将其保留。从剩下的矩形框A、C、E中，选择概率最大的E，然后判断E与A、C的重叠度，重叠度大于一定的阈值，那么就扔掉；并标记E是保留下来的第二个矩形框。就这样一直重复，找到所有被保留下来的矩形框。

主要方法是通过py\_cpu\_nms类来实现的。在这个类中先依次取出边界框左上角和右下角的坐标以及分类器得分(置信度)，然后开始分别计算每个框的面积，将得分进行排序，然后每次取得分最大的框，依次进行相交面积计算，最后对相交面积小于阙值的框进行保留。在本算法里设定IoU为0.7的阈值，即仅保留覆盖率不超过0.7的局部最大分数的box(粗筛)。最后留下大约2000个锚，然后再取前N个box(比如300个)给Fast R-CNN。Fast R-CNN将输出300个判定类别及其box，对类别分数采用阈值为0.3的非极大值抑制(精筛)，并仅取分数大于某个分数的目标结果(比如，只取分数60分以上的结果)。

⑤再次按照nms后的前景softmax scores由大到小排序前景锚，提取前post\_nms\_topN(e.g.300)结果作为建议输出。

⑥之后输出proposal=[xl，yl，x2，y2]，注意，由于在第三步中将锚映射回原图判断是否超出边界，所以这里输出的建议是对应M\*N输入图像尺度的，这点在后续网络中有用。至此，RPN网络结构就介绍到这里了，总结起来就是：生成锚→softmax分类器提取前景锚→边框回归前景锚→建议层生成建议。

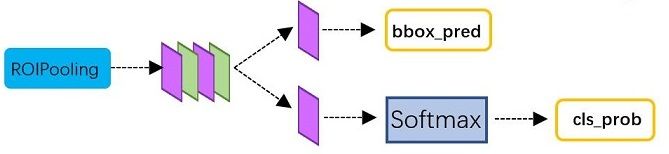
（5）RoI池化原理

RoI池负责收集建议，并计算出建议特征，送入后续网络。从图25中可以看到RoI池有2个输入：原始的特征图和RPN输出的proposal boxes(大小各不相同)RoI Pooling layer forward过程：在之前有明确提到：proposal=[x1，y1，x2，y2]是对应M\*N尺度的，所以首先使用spatial 参数将其映射回大小的特征图．scale M\*N尺度；之后将每个建议水平和竖直都分为7份，对每一份都进行max pooling处理。这样处理后，即使大小不同的建议，输出结果都是7\*7大小，实现了fixed\_1ength output，如图24所示。

**图24 proposal示意图**

（6）分类(Classification)原理

分类部分利用已经获得的建议特征图，通过full connect层与softmax计算每个建议具体属于那个类别(如人，车，电视等)，输出概率向量；同时再次利用边框回归获得每个建议的位置偏移量，用于回归更加精确的目标检测框。分类部分网络结构如图25所示。从RoI池获取到7\*7=49大小的建议特征图后，送入后续网络进行训练。



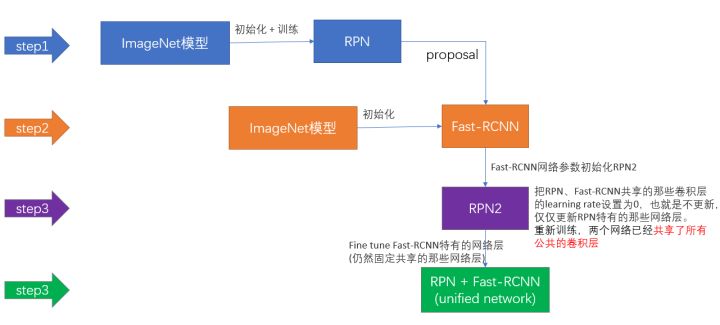
**图25 Classification部分网络结构图**

### 3.2Faster R-CNN网络训练模型的训练

Faster R-CNN的训练，是在已经训练好的model（如VGG\_CNN\_M\_1024，VGG，ZF）的基础上继续进行训练。实际中训练过程分为6个步骤：

1. 在已经训练好的model上，训练RPN网络，对应stage1\_rpn\_train.pt。
2. 利用步骤1中训练好的RPN网络，收集proposals，对应rpn\_test.pt。
3. 第一次训练Fast R-CNN网络，对应stage1\_fast\_rcnn\_train.pt。
4. 第二训练RPN网络，对应stage2\_rpn\_train.pt。
5. 再次利用步骤4中训练好的RPN网络，收集proposals，对应rpn\_test.pt。
6. 第二次训练Fast R-CNN网络，对应stage2\_fast\_rcnn\_train.pt。

可以看到训练过程类似于一种“迭代”的过程，循环2次。整体训练过程如图26所示：



**图26 整体训练流程**

#### 3.2.1训练文件架构

工程目录下有以下几个文件夹：

(1)caffe-fast-rcnn：caffe框架的存放目录；

(2)data：用来存放pretrained模型，比如imagenet上的，以及读取文件的cache缓存；

(3)experiments：用于存放配置文件以及运行的log文件，另外这个目录下有scripts可以用end2end或者alt\_Opt两种方式训练；

(4)Lib：用来存放一些python接口文件，有以下子文件夹：

1. datasets：主要负责数据库读取
2. Fast-rcnn：主要存放的是python的训练和测试脚本，以及训练的配置文件conffig.py
3. nms:非极大值抑制，用于合并候选框
4. roi\_data\_layer：ROI处理操作
5. rpn：rpn的核心代码，定义了建议和锚的生成方法。
6. Models：里面存放了三个模型文件，小型网络的ZF，大型网络VGGl6，中型网络VGG CNN M 1024,本文使用VGGl6，其中使用到了端到端的approximatejoint training方法，开启CuDNN，只需要3G的显存。

(6)Output：这里存放的是训练完成后的输出目录，默认会在faster rciln end2end文件夹下.

(7)Tools：里面存放的是训练和测试的Python文件。

#### 3.2.2训练环境

1、硬件配置

电脑配置：GPU，内存16G，硬盘空间：200G

电脑系统：Linux

2、软件配置

训练框架：Caffe，Pycaffe

3、对py-faster-rcnn进行配置与安装

Faster R-CNN的算法框架是开源的，主要分为python版本和matlab版本两种。本文将主要论述pyhton版本的Faster R-CNN算法的相关训练与使用。算法框架的安装很简单，在刚开始首先对Faster R-CNN进行克隆，然后对Cpython模块进行编译，最后对里面的Caffe和pycaffe进行编译。

4、py-faster-rcnn部署测试

为了对py-faster-rcnn是否已经成功安装进行测试，笔者采用了官方已经训练好的数据集PASCALVOC2007来进行测试。将faster\_rcnn\_models．tgz文件下载后，自动解压成ZF\_faster\_rcnn\_final．caffemodel和VGGl6\_faster\_rcnn\_final．caffemodel两个模型。其中，前者是在ZF网络模型下进行训练的，后者是在VGGl6模型下进行训练。

#### 3.2.3主要训练过程

首先是对RPN网络的训练。

先读取RBG提供的预训练好的model（VGG），开始迭代训练。其训练主要网络结构stage1\_rpn\_train.pt如图27所示：



**图27 stage1\_rpn\_train.pt网络结构**

Conv Layers提取feature maps。整个网络使用的Loss见公式（3.7）：

 （3.7）

上述公式中，i表示anchors index，表示foreground softmax probability 代表对应的GT predict概率，（即当第i个anchor与GT间 IoU>0.7,认为是该anchor是foreground，；反之IoU<0.3时，认为是该anchor是background，；至于那些0.3<IoU<0.7的anchor则不参与训练）；t代表predict bounding box，代表对应foreground anchor对应的GT box.整个loss分为两个部分：

cls loss，即rpn\_cls\_loss层计算的softmax loss，用于分类anchors为forground与background的网络训练；

reg loss，即rpn\_loss\_bbox层计算的soomth L1 loss，用于bounding box regression网络训练。注意在该loss中乘了 ，相当于只关心foreground anchors的回归（其实在回归中也完全没必要去关心background）。

由于在实际过程中，和差距过大，用参数平衡二者（如，时设置），使总的网络Loss计算过程中能够均匀考虑2种Loss。这里比较重要是Lreg使用的soomth L1 loss，见公式（3.8）：

 （3.8）

可以发现:

1.在RPN训练阶段，rpn-data（python AnchorTargetLayer）层会按照和test阶段Proposal层完全一样的方式生成Anchors用于训练；

2.对于rpn\_loss\_cls，输入的rpn\_cls\_scors\_reshape和rpn\_labels分别对应与，参数隐含在与的caffe blob的大小中；

3.对于rpn\_loss\_bbox，输入的rpn\_bbox\_pred和rpn\_bbox\_targets分别对应与，rpn\_bbox\_inside\_weigths对应，rpn\_bbox\_outside\_weigths未用到（从soomth\_L1\_Loss layer代码中可以看到），而同样隐含在caffe blob中；

这样，公式与代码就完全对应。在训练和检测阶段生成和存储anchors的顺序完全一样，这样训练结果才能被用于检测。

其次是对通过训练好的RPN网络收集proposals。

在该步骤中，利用之前的RPN网络，获取proposal rois，同时获取foreground softmax probability，如图28rpn\_test.pt结构，然后将获取的信息保存在python pickle文件中。该网络本质上和检测中的RPN网络一样，没有什么区别。



**图 28 rpn\_test.pt**

最后是对Faster R-CNN网络的训练。

读取之前保存的pickle文件，获取proposals与foreground probability。从data层输入网络。然后：

1）将提取的proposals作为rois传入网络，如图30蓝框；

2）计算bbox\_inside\_weights+bbox\_outside\_weights，作用与RPN一样，传入soomth\_L1\_loss layer，如图30绿框；

这样就可以训练最后的识别softmax与最终的bounding box regression了，如图29所示。



**图29 stage1\_fast\_rcnn\_train.pt**

### 3.3行人检测算法的实现

算法的整个实现过程包括模型的离线训练、在线接收用户数据、数据的存储、web页面的展示等。

首先在系统中预先对基于Faster R-CNN算法进行数据集的输入和离线训练。算法的整个流程是基于 VGG16 模型的faster\_rcnn 网络结构。网络首先将一副任意大小P\*Q的图像，首先缩放至固定大小M\*N，然后将M\*N图像送入网络；而 Conv layers 中包含了13个conv层+13个relu层+4个 pooling 层。经过这一系列的操作，图像的大小变为的原图的 (M/16)\*(N/16)，也就是 Feature Map 的大小。接下来是RPN网络首先对输入的(M/16)\*(N/16)的 Feature Map 做一个3\*3卷积操作，然后分成两条路：一条通过softmax对anchors分类获得foreground和background，另一条计算 bounding box 的预测值。这样 Proposal 的输入就有6个，分别是bbox的四个预测值（x,y,w,h），以及anchor的类别(foreground和background)，用上面的参数，就可以进行bounding box 的修正；在RPN的最后部分Proposal Layer输入三个参数：anchors分类器结果，bounding box 的回归结果，以及im\_info，总和输入获得获取proposal，同时剔除太小和超出边界的propocals，这里完成了目标定位的功能；其后将原始的featrue map和RPN输出的proposal boxes输入到RoI Pooling层，计算出proposal feature maps；后面是classification部分，这部分利用已经获得的proposal featuer map，通过full connect层与softmax计算每个proposal具体属于哪个类别，输出cls\_prob概率向量，同时再次利用Bounding box regression获得每个proposal的位置偏移量bbox\_pred，用于回归更加精确的目标检测框。这样完成了对图片的目标检测。

因此整个系统就是将训练数据和用户上传的待检测数据上传到数据存储层，将训练的数据在模型层进行对模型训练，进而预测，将用户输入的数据放在训练好的模型里进行训练，得到检测结果在web展示系统中进行检测结果的展示。

## 第四章 行人检测系统的测试与分析

### 4.1数据集

本节重点介绍模型训练所用到的数据集，对其特点进行详细的介绍。

#### 4.1.1数据集的介绍

训练模型采用的训练数据集是caltech数据集。该数据集是由加州理工提供的，包括训练集和测试集，其中数据是seq格式的，标签是vbb（video bounding box）格式的，该格式标签主要是数据集中的行人的bounding box。官网上提供了toolbox对数据进行读写操作，toolbox基于MATLAB。由于我们训练时主要是采用图片格式的数据，所以需要将.seq和.vbb格式的数据转换为.jpg和.xml格式的数据。 该数据库是眼下规模较大的行人数据库。采用车载摄像头进行拍摄，拍摄时长约10个小时左右，视频的分辨率为640x480，30帧/秒。标注了约250,000帧（约137分钟），350000个矩形框，2300个行人。另外还对矩形框之间的时间相应关系及其遮挡的情况进行了标注。

数据集分为set00 - set10，其中set00 - set05是训练集，set06 - set10是测试集。性能评估方法主要有以下三种：（1）用外部数据进行训练，然后用set06 - set10进行测试。（2）6-fold交叉验证，选择其中的五个进行用来做训练，另外一个用来做测试，调整参数，最后给出训练集上的性能。（3）用set00 - set05进行训练，set06 - set10进行测试。因为测试集的标注信息没有公开，需要提交给Pitor Dollar，结果提交方法为每30帧做一次测试，将结果保存在txt文档中（文件的命名格式为I00029.txt， I00059.txt），每一个txt文件中的每一行表示检测到的一个行人，格式为“[left，top，width，hight，score]”。假设没有检测到行人，则txt文档为空。该数据集还提供了对应的Matlab工具包，包含视频标注信息的读取、画ROC曲线图和非极大值抑制等工具。

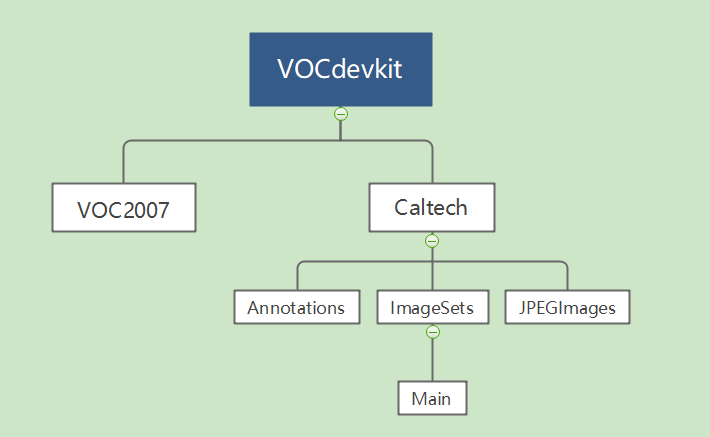
#### 4.1.2数据集的结构

由于Faster R-CNN的原始测试实验是PASCAL VOC2007数据集上进行的，所以Caltech数据集最后应该转换成相应的格式，这样能能够使用该数据集进行训练。

将Caltech数据集转换成VOC格式之后的文件结构图如图30所示。接下来对各文件夹的子结构进行详细的介绍。

（1）Annotations

该文件夹用来存放标注图片的XML格式的数据，每张图片都有一个XML格式的数据与之对应。



**图30文件结构图**

（2）ImageSets

ImageSets文件夹下有三个子文件夹，Layout、Main和Segmentation文件夹，其中主要是main文件夹，main文件夹下有四个文件，分别为train.txt、val.txt、test.txt、trainval.txt文件，每个文件中写着供训练、验证和测试所用文件名的集合。

（3）JPEGImages

存放着所有的jpg格式的输入图片。

### 4.2系统测试实验

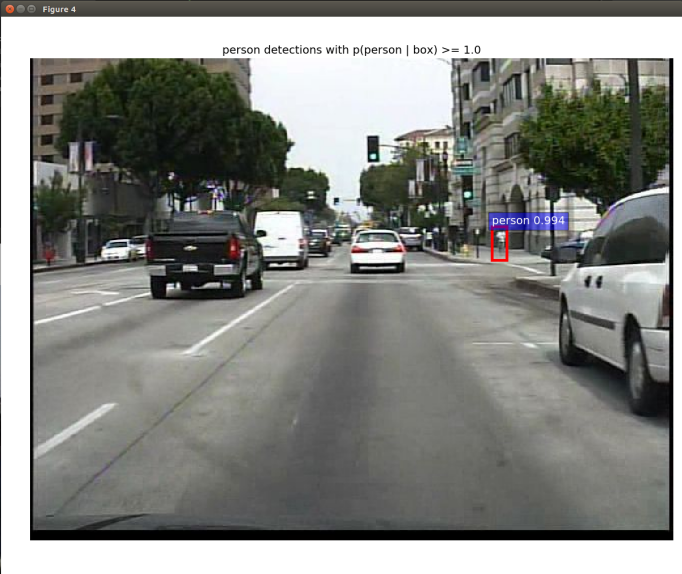
#### 4.2.1复杂环境下的测试

由于caltech数据集是在复杂环境下拍摄的，所以选取了若干张复杂背景下的图片进行测试，测试结果如图31、图32所示。

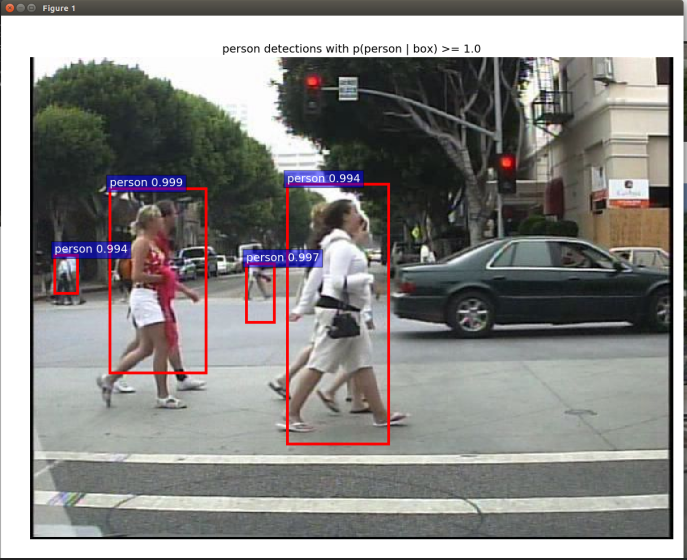
### 4.3实验结果分析

通过对该系统的测试结果进行分析，可以看出该系统在简单场景下能够取得不错的准确率，在复杂场景下，准确率有一定程度的下降，当行人数量较多时，该系统不能够准确的区分不同的行人。

造成这种情况主要有以下几种原因：



**图31 测试结果图**



**图32测试结果图**

（1）背景干扰

行人检测数据集大部分是在大街上拍摄的，存在各种各样的干扰物，如汽车、自行车、手提包、背包等等都有可能造成误检，一定程度上会影响行人检测的准确率。

（2）光照因素

光照的强弱也会影响行人检测的正确率，比如光照在早上和傍晚的时候较为柔和，中午的时候较为强烈，一定程度上影响了成像效果，所以会带来一定程度上的误检率。另一方面，当行人在树下或者有遮挡物的下面行走时，光照造成的阴影打在行人身上也会造成一定的误检。

（3）拍摄角度

由于拍摄角度的原因，行人在照片中有大有小，而且不同的拍摄角度会造成一定的识别难度，比如侧面拍摄就会比正面拍摄较难识别，所以行人目标在图像中所占比例的不确定性给行人检测带来了一定的困难。

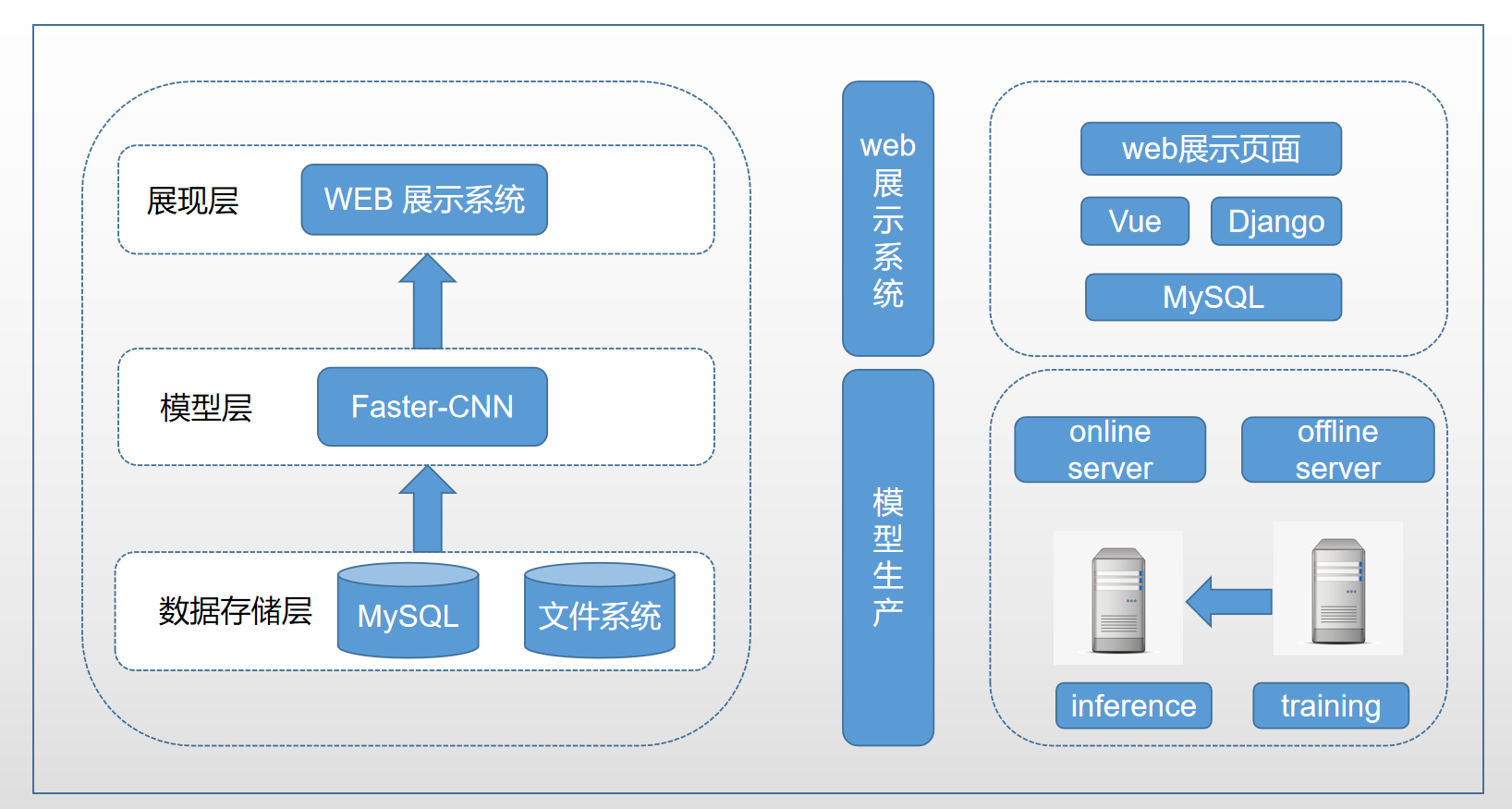
（4）遮挡物

日常生活中，摄像头并不是能够拍摄到整个人的大小，尤其在复杂的环境中中，有各种各样的遮挡物会遮挡行人的一部分身体，这样就会导致系统不能够准确提取行人的特征，从而造成检测的准确率下降。

综上所述，对该系统的不足进行了明确的分析，指明了该系统的不足，最终确定该系统的应用。

## 第五章 演示系统设计

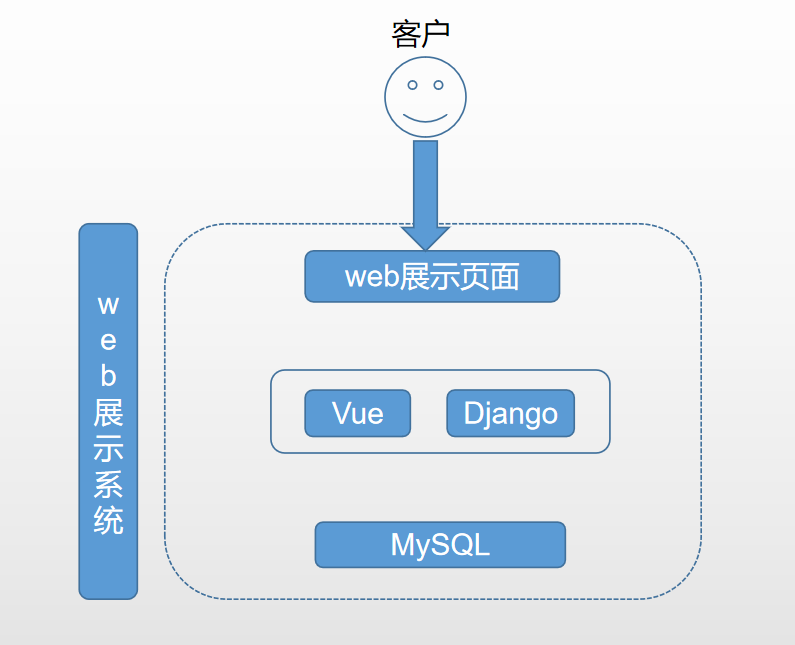
### 5.1系统架构设计



**图33系统总体架构图**

演示系统主要用于展示和介绍项目成果，调用训练好的模型展示目标检测的效果，收集用户反馈信息。演示系统总共包含三部分，数据存储层、模型层和展现层。

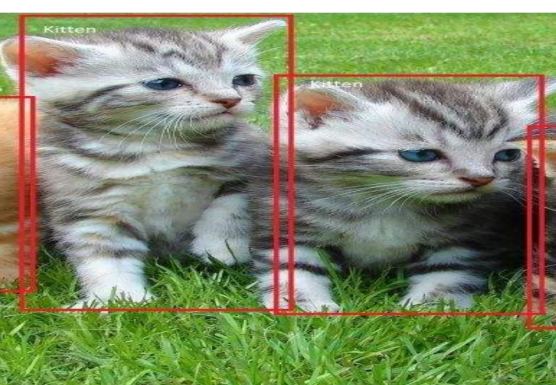
### 5.2 web展示系统

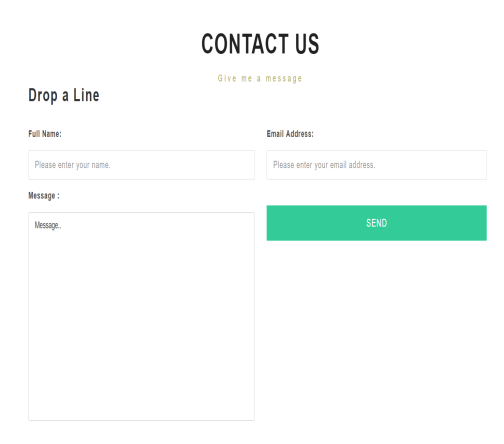
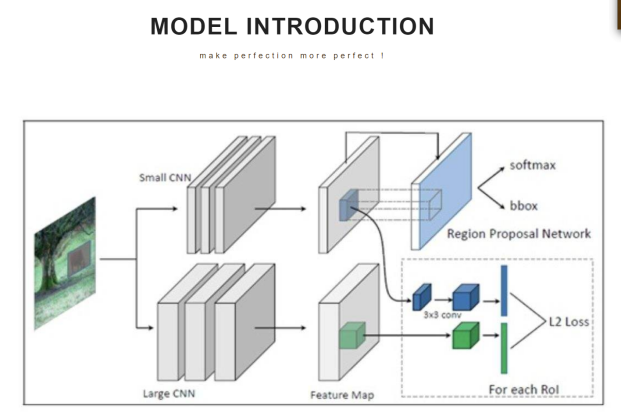


**图34 web展示系统架构**

#### 5.2.1 web展示页面设计

Web展示系统主要分为三部分：web展示页面、框架层、数据存储层。web展示页面分为四个部分：我们的团队、我们的模型、目标检测展示、联系我们。

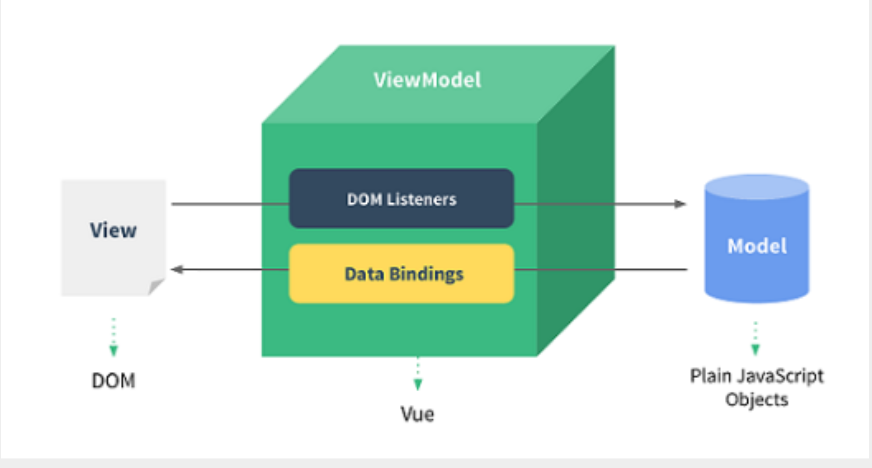
 

**图35系统页面展示**

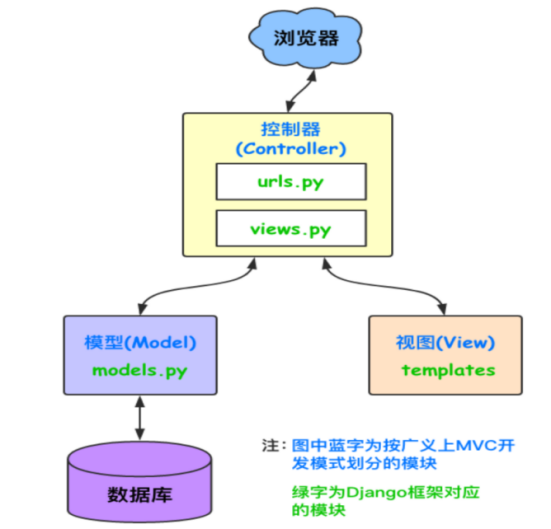
#### 5.2.2 web展示系统使用框架介绍

Web展示系统主要使用了Vue和Django两个主流的开发框架。Vue.js是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，它不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与单文件组件和 Vue 生态系统支持的库结合使用时，Vue 也完全能够为复杂的单页应用程序提供驱动。此外，Vue采用数据驱动页面的方式，是一种MVVM模式，其基本架构如图36所示。

Django是一个开放源代码的Web应用框架，由Python写成。采用了MVT的框架模式，即模型M，视图V和模版T。它最初是被开发来用于管理劳伦斯出版集团旗下的一些以新闻内容为主的网站的，即是CMS（内容管理系统）软件。并于2005年7月在BSD许可证下发布。这套框架是以比利时的吉普赛爵士吉他手Django Reinhardt来命名的，其基本架构如图37所示。

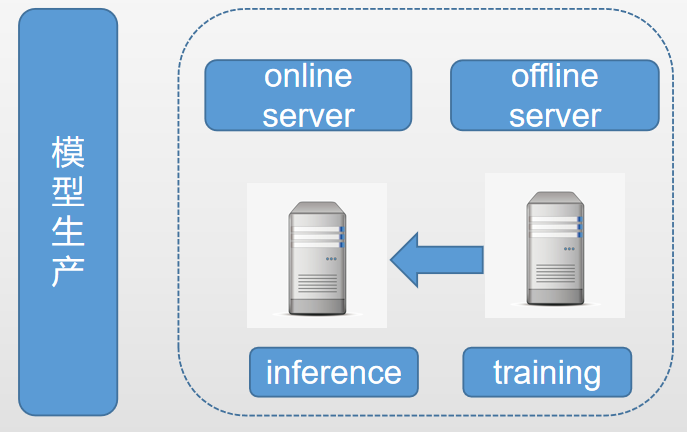


**图36 Vue框架**



**图37 Django框架**

### 5.3模型层



**图38模型层架构**

模型层主要负责对模型进行训练，进行预测。模型层架构如图38所示，关于Faster R-CNN模型的相关问题，前面已经进行了详细描述，此处不再赘述。我们采用离线服务器对模型进行训练，在线服务器接收用户上传的图片调用训练好的模型进行目标检测。

### 5.4 数据存储层

数据存储主要使用Mysql和文件系统，所有的图片数据直接存储到文件系统中，包括训练数据和用户上传的待检测的数据，Mysql数据库主要用于存储图片的路径信息和用户反馈信息等相关数据。

## 第六章 个人工作总结

### 6.1王冠个人总结

经过一段时间的忙碌准备，终于在规定时间内完成了二级工程实践的大作业，这段和队友一起合作努力的日子很有意义，也感谢所有给予我帮助的人。在完成大作业的过程中，遇到了很多的困难和挑战，也收获了很多，对自己所学知识进行了很好的检验，下面我将从几个方面来总结我的工作。

我在这次大作业中主要负责模型的设计与训练，以及编写自己负责部分的论文。由于自己的能力有限，不能够设计出一个全新的神经网络模型去解决问题，所以就直接参考github上开源的Faster R-CNN源代码，代码是基于caffe框架，使用python进行编写。我首先是阅读代码，理解网络的结构原理，并总结了该网络中所用到的技巧，比如nms等等，在阅读代码的过程中，有些python库不是很熟练，例如numpy、scipy、matplotlib、cv2、os等等，又重新进行了学习，代码中提供了基于coco和Pascal数据集预训练的VGG和ZF网络，需要翻墙下载，在训练自己的数据集时就可以在这些预训练好的模型的基础上进行训练，节省时间。阅读代码大约花费了一两周的时间，这部分没有遇到很大的麻烦，而且提升了自己的python知识。

阅读完代码之后下一步就是在自己电脑上搭建环境。需要做的就是安装各种库，例如OpenCV、caffe等等，在编译caffe的时候遇到了很多问题，比如缺失一些依赖库，安装的库的版本不对，以及修改Makefile文件等等，通过在网上查询博客也都一一解决，编译完caffe基本上就完成了百分之八十的工作，下一步就是下载预训练模型，最后成功跑起来了代码提供的demo。

下一步就是准备自己的数据集，我们做的系统的目的是检测行人，网上有很多关于行人的数据集，使用比较多的就是caltech数据集，所以我们也是采用的这个数据集，该数据集是基于matlab进行标注的，数据集是一段一段的视频，所以应该转换成voc的数据格式，即将视频转换成jpg格式的图像，标注文件转换成xml的文件，github上有转换这部分的脚本，不过都不能满足自己的需求，所以我又基于网上的脚本自己进行了改写，最终成功的把数据集转换成voc格式。

最后是进行训练以及测试，用于训练的显卡是1070，能够满足自己的需求，大概训练了7个小时左右，每迭代10000次保存一次模型，在训练的过程中没有遇到什么问题，训练完成后使用自己的图片进行了测试，行人检测的准确率还是很高的，能够达到百分之99左右，但是候选框的位置不是很精确，不能够将行人全部的框出，而且存在很多的重合框，尝试了几种方法进行改进，效果不是很好，最后调了一下超参数以及nms的过滤阈值，效果有一点提升。

总的来说，在完成这次大作业的过程中认识到了自己知识上的欠缺以及下一步应该努力的方向，在一定程度上也培养了自己解决问题的能力，最后感谢老师的悉心指导。

### 6.2陈林个人总结

首先在确定实践项目之前，我们对于目标检测（object detection）涉及到的知识了解也是有限的，所以需要恶补相关的内容。目标检测就是给定的图片中精确找到物体所在位置，并标注出物体的类别；目标检测发展历程是从传统的目标检测方法（多尺度形变部件模型DPM）再到 以R-CNN为代表的结合region proposal和CNN分类的目标检测框架（R-CNN, SPP-NET, Fast R-CNN, Faster R-CNN, R-FCN），再到现在以YOLO为代表的将目标检测转换为回归问题的端到端（End-to-End）的目标检测框架(YOLO, SSD)。比如对该领域的发展有详细的了解，才能知道算法的发展和改进过程，这样才能更好的实现我们的目标检测算法，并且作出适当的改进。

经过查阅资料才知道了传统目标检测的方法一般分为三个阶段：首先利用不同尺寸的滑动窗口框住图中的某一部分作为候选区域，然后在候选区域提取相关的特征（Harr特征、HOG特征等），然后在利用常用得分类器（SVM等）进行识别分类；但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，并且手工设计的特征对于多样性的变化并没有很好的鲁棒性。

下一阶段便是基于R-CNN一系列方法；R-CNN主要流程就是先输入图像，用selective search(选择性搜索)算法在图像中提取2000个左右的region proposal（侯选框），并把所有region proposal warp（缩放）成固定大小（原文采用227×227），然后将归一化后的region proposal输入CNN网络，提取特征 ，最后对于每个region proposal提取到的CNN特征，再用SVM分类来做识别，用线性回归来微调边框位置与大小，其中每个类别单独训练一个边框回归（bounding-box regression）器；SPP-NET的主要思想是去掉了原始图像上的crop/warp等操作，换成了在卷积特征上的空间金字塔池化层（Spatial Pyramid Pooling，SPP），SPP只对整幅图像做一次CNN特征提取，那么原图一个region proposal可以对应到featuremap（特征图）一个window区域，只需要将这些不同大小window的特征映射到同样的维度，将其作为全连接的输入，就能保证只对图像提取一次卷积层特征，大大加快了检测速度。Fast R-CNN对R-CNN的基础上又做了一些改进，第一点它与SPP类似，它只对整幅图像做一次CNN特征提取，在后面加了一个类似于SPP的ROI pooling layer，其实也就是下采样。不过因为不是固定尺寸输入，因此每次的pooling网格大小需要动态调整，从而实现区域归一化；第二点是用softmax替代SVM分类，同时利用Multi-task Loss（多任务损失函数）将边框回归和分类一起进行。Faster-RCNN用RPN获得的proposal替代Fast R-CNN中selective search获取的proposal，这是因为因为RPN生成ROI时效率更高。

最后了解的便是YOLO为代表的单阶段检测算法；彻底端到端（End-to-End）的目标检测方法，就是不需要中间的region proposal在找目标，直接回归便完成了位置和类别的判定，但是也存在检测精度不高的问题，但是YOLO的速度快，泛化性能好；SSD与YOLO不同的是，YOLO预测某个位置使用的是全图的特征，SSD预测某个位置使用的是这个位置周围的特征。使用Faster R-CNN类似的anchor机制，SSD与RPN也很类似，SSD中的dafault bounding box类似于RPN中的anchor，但是，SSD在不同的特征层中考虑不同的尺度，RPN在一个特征层考虑不同的尺度。

通过对目标检测系统性的学习，让我对该领域有了不仅仅是感性上的认识，也从理性上明白了他们的区别与联系，同时结合他们一起训练模型，调参数，改数据集，让我在学习到了许多神经网络相关的知识，我也希望有了一定的理论基础后，可以站在前人的肩膀上，能对现有的目标检测算法能做出比较好的改进办法，也感谢这么多天来老师同学们的帮助。

### 6.3华旎个人总结

本次工程实践我们选择了行人检测的题目，这个题目也是当前研究的一个热点问题，在智能医疗，安全防范，无人驾驶等人工智能领域有着重要的应用价值，当前随着深度学习技术的发展目标检测算法已经相对于传统的方法在检测准确度上有了大幅度的提升，但依然存在着大量的问题。在本次课程设计过程中通过查阅相关的资料使我们对于目标检测算法有了基本的了解，为我们后续的研究打下了基础。

本次课程设计我主要负责报告的撰写，从前期的资料收集，文献综述，到最后的报告整合，格式整理。在这个过程中也了解了目标检测的一系列发展过程以及成果，传统的目标检测算法及策略已经难以满足目标检测中数据处理的效率、性能、速度和智能化等各个方面要求，基于深度学习的方法具有强大的视觉目标检测能力，成为了当前目标检测的主流算法。了解学习了传统目标检测的发展以及存在问题；然后着重介绍了4种典型的基于深度学习目标检测算法，对算法的概况、步骤、优缺点等进行详细的介绍和分析；最后对深度学习目标检测算法进行总结和展望。在目标检测的基础上对行人检测算法的发展进行了学习与了解，在完成报告的过程中也对模型训练以及检测有所了解，对整个项目的完成也有了整体的把握。

最后，向和我一起完成本次课程设计的小组成员们，为我们的课程设计做出辛勤指导的老师们致以诚挚的谢意！

### 6.4夏涛个人总结

本次课程设计我们选择了目标检测的题目，这个题目也是当前研究的一个热点问题，在智能医疗，安全防范，无人驾驶等人工智能领域有着重要的应用价值，当前随着深度学习技术的发展目标检测算法已经相对于传统的方法在检测准确度上有了大幅度的提升，但依然存在着大量的问题。在本次课程设计过程中通过查阅相关的资料使我们对于目标检测算法有了基本的了解，为我们后续的研究打下了基础。

本次课程设计我主要负责了演示系统的设计和实现，这是我第一次从前端到后端完整实现的一个系统。前端采用了当前主流的设计框架Vue.js，后端采用Django框架，从系统的原型设计开始逐步完成整个系统的编码工作。Vue.js框架由于是第一次接触，遇到了大量的问题，从CSS样式调节到前后端传参以及js动画和Vue路由设置等方面都遇到了困难，经过自己的努力最终一一得到解决，踉踉跄跄的完成了整个系统的编码。当前的演示系统主要目标还是用于展示我们此次课程设计的成果，在系统设计中还存在着许多不规范的地方，也没有考虑系统性能、高并发等方面的问题，因此当前的系统只能还用于演示，离真正意义上的线上的生产系统还有很长一段距离。本次课程设计我的主要目标是训练自己实现系统的能力。因此没有在模型方面投入过多的精力，通过本次课程设计也让我意识到了必须进一步加强自身的理论知识水平，提升自己实现模型的能力。

最后，向和我一起完成本次课程设计的小组成员们，为我们的课程设计做出辛勤指导的老师们致以诚挚的谢意！

## 参考文献

[1] 张春凤,宋加涛,王万良.行人检测技术研究综述[J].电视技术,2014,38(3):157-162.

[2] 张春凤.基于多特征的行人检测技术研究[D].浙江工业大学,2013.

[3] 李斌,史忠科.基于计算机视觉的行人检测技术的发展[J].计算机工程与设计,2005, 26(10):2565-2568.

[4] 冯偲.基于机器视觉的行人检测技术研究[D].长春工业大学,2017.

[S] 常玲玲.基于深度学习的行人检测问题研究[D].宁夏大学,2017.

[6] 陈文明.行人检测技术简述[J].广东蚕业,2017(11).

[7] 王斌.基于深度学习的行人检测[D].北京交通大学,2015.

[8] 袁德东,基于深度学习的特定场景下的行人检测方法研究[D].厦门大学,厦门大学, 2015.

[9] 祝浩.基于多特征融合行人检测系统设计与实现[D].电子科技大学,2016.

[10] 张宁,基于粒子滤波的校园行人检测追踪系统设计与实现[D].山东科技大学,2014.

[11] 董观利,宋春林.基于视频的矿井行人越界检测系统[J].工矿自动化,2017,43(2):29-34.

[12] 王焱.基于随机梯度提升决策树的行人检测算法设计与实现[D].浙江大学,2017.

[13] 谢林江,季桂树,彭清,等.改进的卷积神经网络在行人检测中的应用,2017年10月20日[J].计算机科学与探索,2017.

[14] 王梦来,李想,陈奇,等.基于CNN 的监控视频事件检测[J].自动化学报,2016, 42(6):892-903.

[15] Girshick R.Fast R-CNN[J].Computer Science,2015.

[16] Ren S,He K,Girshick R,et al.Faster R-CNN:towards real-time object detection with region proposal networks[C]. MIT Press,2015:91-99.

[17] Rose M.Management Information Base for Network Management of TCPIP-based internets: MIB-II[J].Rfc,1991.

[18] INetTCPIP.Internetworking with TCP/IP[M].Tsinghua University Pres,1998.

[19] Chua LO,Roska T.CNN paradigm[J].IEEE Transactions on Circuits&Systems I Fundamental Theory&Applications,1993,40(3):147-156.

[20] Tuzel O,Porikli F,Meer P.Pedestrian Detection via Classification on Riemannian Manifolds[J].IEEE Trans Pattern Anal Mach Intell,2008,30(10):1713-1727.

[21] Li J, He Q, Yang L, et al. Pedestrian Detection and Counting in Crowded Scenes[C],GreenIntelligent Transportation Systems.2018.

[22] Bradski G.The Opencv Library[J].Doctor Dobbs Journal,2000,25(11):384-386.

[23] Bradski G R,Kaehler A.Learning OpenCV[J].Oreilly Media,2009.

[24] Altschul S F,Madden T L,Schaffer A A,et al.A new generation of protein database search programs[J].Dissertations&Theses-Gradworks,1997,13(5):148.

[25] Lin T Y,Maire M,Belongie S,et al.Microsoft COCO:Common Objects in Context[J].2014, 8693:740-755.

[26] Chen X,Fang H,Lin T Y,et al.Microsoft COCO Captions:Data Collection and Evaluation Server[J].Computer Science,2015.

[27] Veit A,Matera T,Neumann L,et al.COCO-Text:Dataset and Benchmark for Text Detection and Recognition in Natural Images[J].2016.

[28] Grande V,Kruglov A,Foglini F,et al.The CoCoNet solution for management and access heterogenous marine datasets and metadata[C],IMDIS 2016 International Conference on Marine Data and Information Systems.2016.