

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

设计文档



项目题目: 楼道杂物自动检测系统

学生姓名: 顾婧、鲍奕凡、仇雨恬、王蕾颖、陈予涵

目录

1 系统架构2

2 vibe 背景建模.....2

3 camshift 移动目标检测.....3

4 遗留物提取3

5 CNN 卷积神经网络分类4

 5.1 图片数据集的收集4

 5.2 自训 CNN 卷积神经网络分类器4

6 基于 yolo3 的特定目标检测5

 6.1 yolo3 原理简介5

 6.2 yolo3 模型的训练5

7 数据库搭建6

 7.1 数据库简介6

 7.2 数据格式6

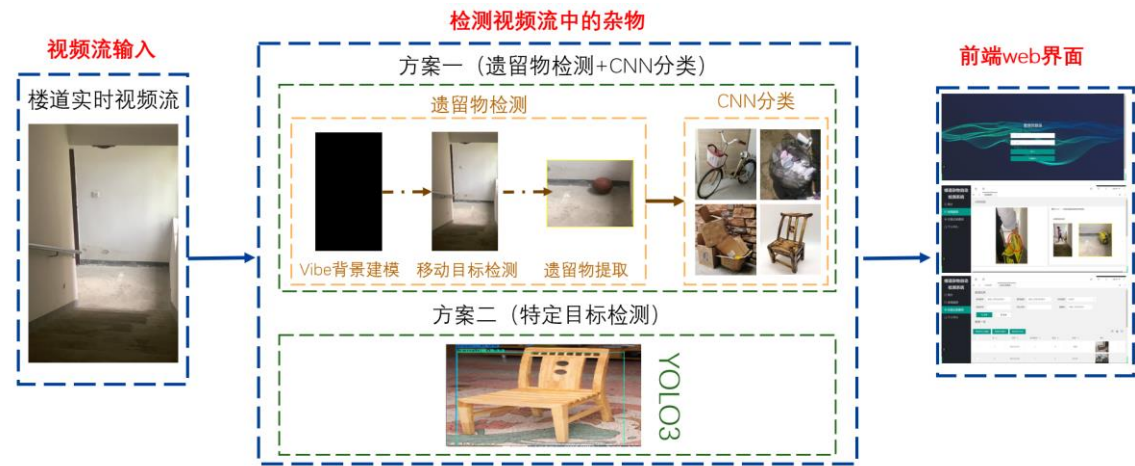
8 系统搭建7

 8.1 Django 原理简介.....7

 8.2 本项目中的应用8

1 系统架构

本项目的系统设计主要包含三部分内容，框架示意图如下图所示：



系统整体框架示意图

①视频流输入：获取楼道的实时视频流；

②检测楼道视频流中的杂物：本部分对前一部分获取的楼道实时视频流进行处理，提出了两个不同的杂物检测方案。方案一为遗留物提取与 CNN 卷积神经网络分类结合，遗留物提取的部分运用了 vibe 背景建模、移动目标检测与遗留物提取。方案二为使用 yolo3 进行特定目标检测。

③前端 web 界面：将处理好的结果在 web 界面上进行显示。

2 vibe 背景建模

遗留物检测是目前处理智能视频流所使用的关键性技术,其主要目的就是原视频信息中的背景建模后将后续遗留物体与原背景分离开,以提取到清晰完整的遗留物目标。本项目最终选取了其中精度最高的视觉背景提取法(VIBE 算法)作为本项目遗留物检测中的场景建模算法。

本项目通过 vibe 算法进行背景建模,其基本逻辑步骤为初始化视频流初背景、检测前景和判断是否更新背景模型三部分。其逻辑示意图如下图所示。



vibe 背景建模逻辑示意图

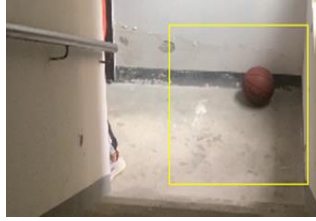
3 camshift 移动目标检测

进入每一帧画面时,基于先前 Vibe 背景建模获得的运动前景,可对应检测出当前画面中运动物体的位置。用小矩形框遍历画面,将前景所占区域达到阈值的部分标记为运动区域,这样就将运动物体划分为了 n 个部分。随后,在这 n 个矩形框里分别基于 vibe 提供的前景掩膜进行过滤,提取色调信息,获得反向概率图,并以之初始化搜索窗口,完成追踪的初始化。

Camshift 算法是一个自动迭代的过程。画面更新时,每一个搜索窗口的中心仍然处在上一帧的位置。该算法根据最初所提供的颜色信息,向周围搜索,寻找当前所追踪物体出现概率最大的区域,将搜索窗口的中心移动到该位置,同时自动调整搜索框的大小。对每一帧画面进行上述过程,就实现了运动目标的追踪。

4 遗留物提取

遗留物的特征是从运动状态进入静止状态,并长时间停留在画面中。基于此特征,对 camshift 追踪的 n 个搜索框进行考察。若其中的 k 个搜索框在运动一段时间后,到达画面边缘,且无法搜索到之前跟踪的物体,我们认为这 k 个搜索框所对应的运动物体已经离开画面。将这 k 个搜索框销毁,考察剩余的 $n-k$ 个搜索框。若搜索框长时间处在画面中且不在运动,判断其追踪目标已经成为遗留物。此时,根据这些窗口框选出遗留物,以备进一步检验;同时,记录这些遗留物从进入画面开始所经过的时间,以便于查找遗留物主。随后销毁这 $n-k$ 个窗口。

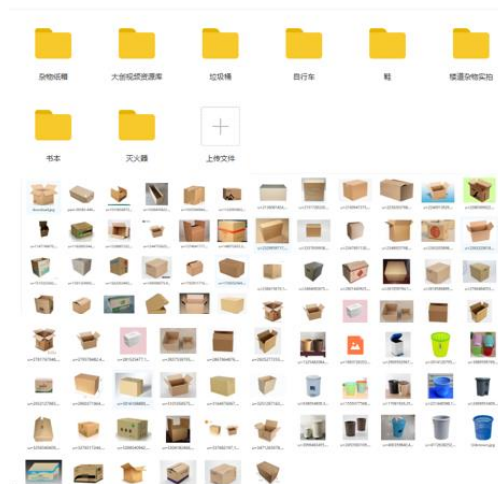


遗留物判断示例

5 CNN 卷积神经网络分类

5.1 图片数据集的收集

经过调研后，我们将常见的楼道杂物主要分为了“杂物纸箱”、“垃圾桶与垃圾袋”、“自行车”、“鞋”、“图书”、“灭火器”等七大类，随后结合项目特点，通过线下实拍楼道和线上寻找相关图片素材的方法双轨并行收集图片，得到了共计 3159 张可用图片数据。



部分图片数据集

5.2 自训 CNN 卷积神经网络分类器

为了辨别出 camshift 差帧法中判断为遗留物写入储存器的杂物所属种类，本项目采用 tensorflow 库实现 CNN 卷积神经网络作为机器学习系统训练分类器，目前已经成功实现能分类所有五个类别垃圾的五分类器。使用 tensorflow 训练自己的分类器的主要思路如下：

①准备图片数据集，针对每种分类准备三个不同的图片集，分别为训练集、测试集和预测集。其中训练集的图片用于训练分类模型，测试集图片用于测试该模型的精确度，训练好模型后，用预测集中的图片预测该图片所在分类。

②创建图像数据生成器，用于调整数据集中的图片的长、宽、面积等参数，从而生成用于训练用的标准格式化图片数据。

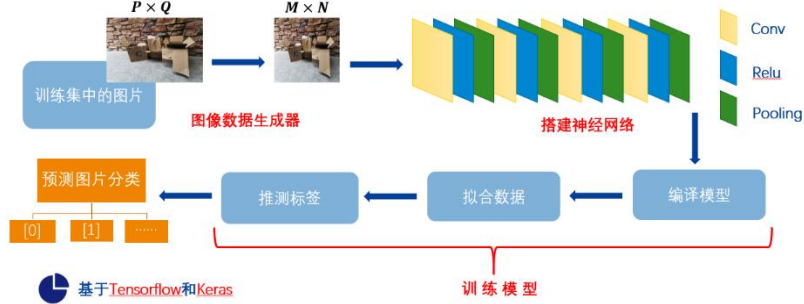
③定义神经网络，一个典型的卷积神经网络通常包含以下几种层，分别为卷积层 (Convolutional layer)、线性整流层 (Rectified Linear Units layer)、池化层 (Pooling layer) 和全连接层 (Fully-Connected layer)。在本项目的 CNN 深度卷积算法中设计了 4 层卷积层，4 层线性整流层，4 层池化层，每一层在进入密集层之前都有最大池化，各层之间的关系如下图所示。同时通过丢弃一点神经元来提高神经网络效率。由于是五分类器，因此在输出

层定义五个神经元，从而起到将待预测图片分为 5 类的效果。

④编译神经网络并拟合数据，并获取该模型的精确度。

⑤输入预测集中用于预测的图片，检验输出的标签是否符合类别。

该部分逻辑结构示意图如下图所示：

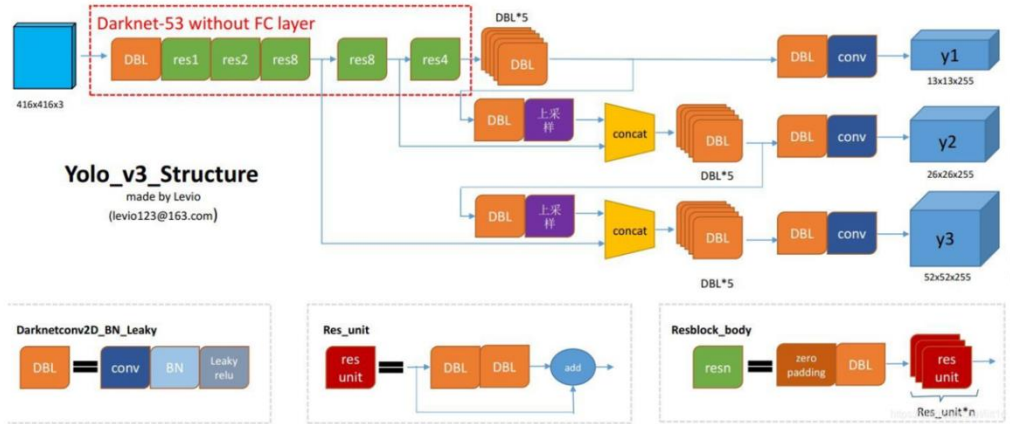


CNN 卷积神经网络分类逻辑示意图

6 基于 yolo3 的特定目标检测

6.1 yolo3 原理简介

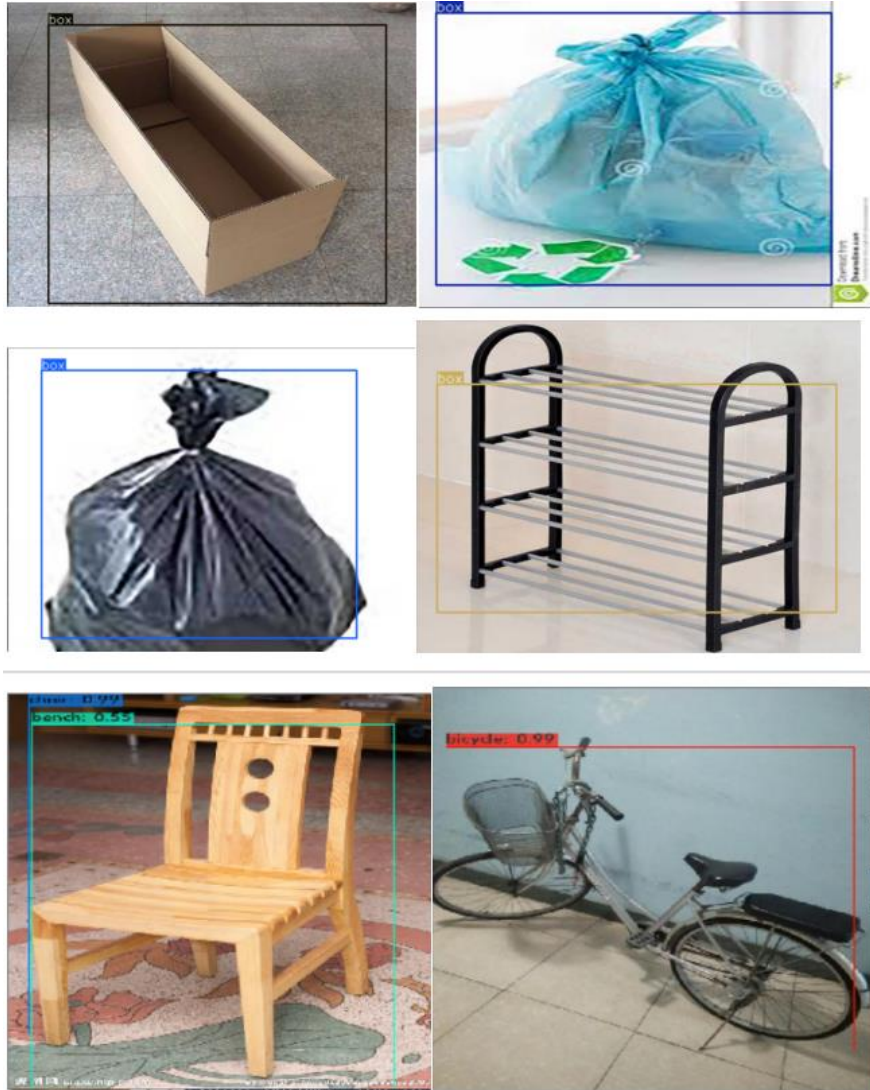
YOLO（You Only Look Once）系列算法的关键思想是利用整张图作为网络的输入，直接在输出层回归 bounding box 的位置及其类别，而非首先划分区域的传统两步式做法。这样可以使得检测效率得到提升。YOLOv3 使用卷积神经网络进行特定目标检测，由主干网络 Darknet-53 以及 YOLO 检测层构成。Darknet-53 有 5 个不同尺度和深度的残差模块，每个残差模块借鉴 Resnet 结构，由一对连续的 3×3 、 1×1 卷积层和跳层连接组成，可以克服梯度消失以及精度下降问题，增强了特征表达能力。



YOLOv3 基本流程图

6.2 yolo3 模型的训练

训练集处理包括重新调整图片和特定目标的标记。本项目使用 Photoshop 批量处理图像，将像素调整为 640×480 大小。随后手动标记每张图片中特定目标的位置，生成标记文件。完成上述准备后，即可按 YOLOv3 的要求开始模型的训练。



YOLOv3 特定目标检测示例

7 数据库搭建

7.1 数据库简介

MySQL 是最流行的关系型数据库管理系统，在 WEB 应用方面 MySQL 是最好的关系数据库管理系统应用软件之一。在本项目中，我们采用了 MySQL 作为数据库的实现方式，并在 Python 中用 Pymysql 插件与数据库相连接。

7.2 数据格式

数据库中主要存储两张表格：一张存储账号信息，另一张存储楼道监控视频的相关信息，如楼号、楼层、时间与是否已处理等。

用户注册后，把用户名和密码的散列值存入 users 表格。

上传图片/视频后，把图片处理结果对应信息存入 records 表格，其中 rphoto 为上传的图片/视频的截图，rcat 为从当前图片识别的杂物类别，若图片识别出多条类别杂物，则每一类

杂物记为一条数据。

对象

myapp_users @project (project) - 表myapp_records @project (project) - ...

开始事务

文本

筛选

排序

导入

导出

id	rdate	rbuilding	rfloor	rcat	rphoto	isDelete	isHandle
1	2022-04-13	1	1	纸箱	../static/people/video1-3.png	0	1
2	2022-04-22	2	2	自行车	../static/people/test-bicycle.jpg	0	1
3	2022-04-25	2	5	纸箱	../static/people/test-box1.jpg	0	0
4	2022-04-27	1	2	家具	../static/people/test-furniture2.jpg	0	0
5	2022-05-04	1	5	自行车	../static/people/test-bicycle2.jpg	0	0
6	2022-05-04	2	1	纸箱	../static/people/test-box2.jpg	0	1
7	2022-05-12	1	3	家具	../static/people/test-furniture.jpg	0	0
8	2022-06-03	1	5	纸箱	../static/people/video1-3.png	0	0
9	2022-06-03	1	4	纸箱	/static/people/video1-3.png	0	1
10	2022-06-06	1	1	cardbor	/static/upimg-2022060600534093799	0	0
11	2022-06-06	1	1	paper	/static/upimg-2022060600534093799	0	0
12	2022-06-06	1	1	cardbor	/static/upimg-2022060617404747839	0	0

对象		myapp_users @project (project) - 表	
开始事务		文本	筛选
		排序	导入
		导出	
username	passwd		
admin	e10adc3949ba59abbe56e057f20f883e		
admin1	e10adc3949ba59abbe56e057f20f883e		
user01	b75705d7e35e7014521a46b532236ec3		

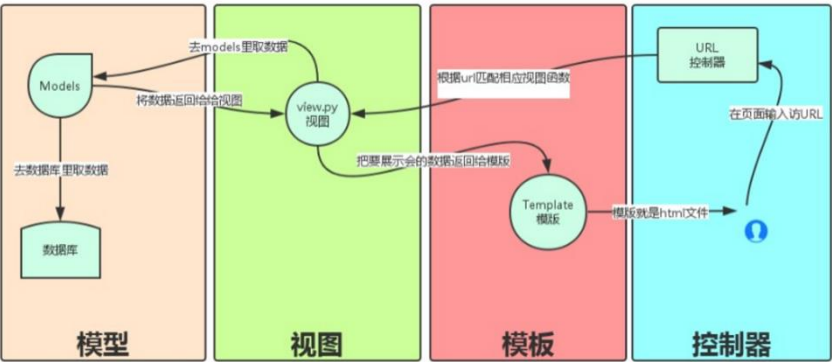
数据库表格示意图

8 系统搭建

8.1 Django 原理简介

本项目的定位为 PC 端提供给物业管理员使用的楼道杂物检测系统, 主要使用了 Django 作为应用框架搭建前端界面和后端数据集。

Django 是一个 MVC 框架。在 Django 中, 控制器接受用户输入的部分由框架自行处理, 所以 Django 里更关注的是模型 (Model)、模板(Template)和视图 (Views), 称为 MTV 模式。



Django 框架示意图

- ①M 代表模型 (Model)，即数据存取层。该层处理与数据相关的所有事务：如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等；
- ②T 代表模板(Template)，即表现层。该层处理与表现相关的决定：如何在页面或其他类型文档中进行显示；
- ③V 代表视图 (View)，即业务逻辑层。该层包含存取模型及调取恰当模板的相关逻辑，可视为模型与模板之间的桥梁。

8.2 本项目中的应用

在本项目中整个系统的前端采用 layui 框架，前后端连接采用 Django 框架编写。前端通过 URL 传参到后端。后端对前端输入的视频流等进行一系列视频流处理后，将处理后的结果返回前端显示，并输入数据库备份，其逻辑结构示意图如下图所示。



前后端搭建结构示意图