演示系统设计

### 系统架构设计

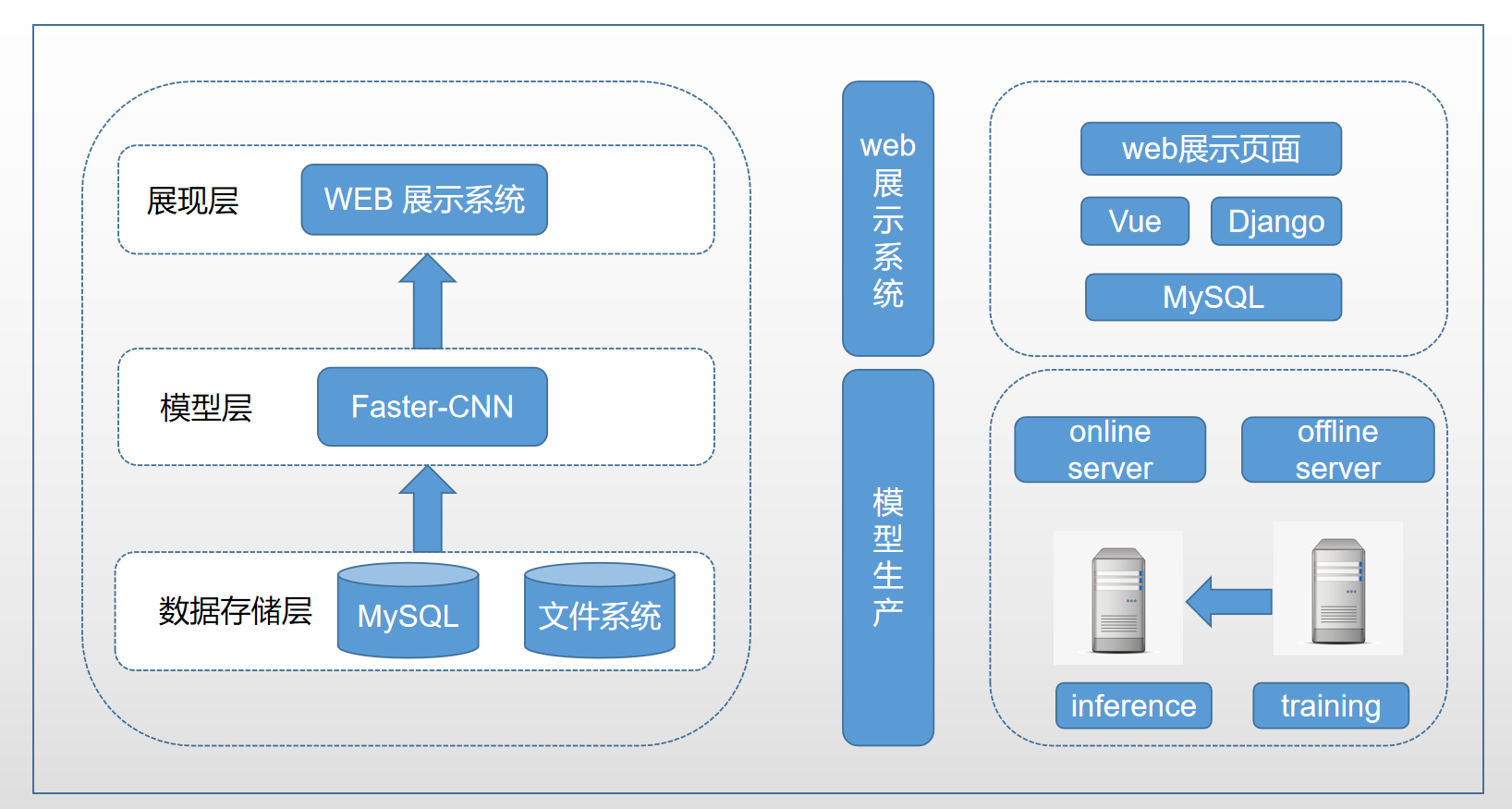


图1 系统总体架构图

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。

### web展示系统

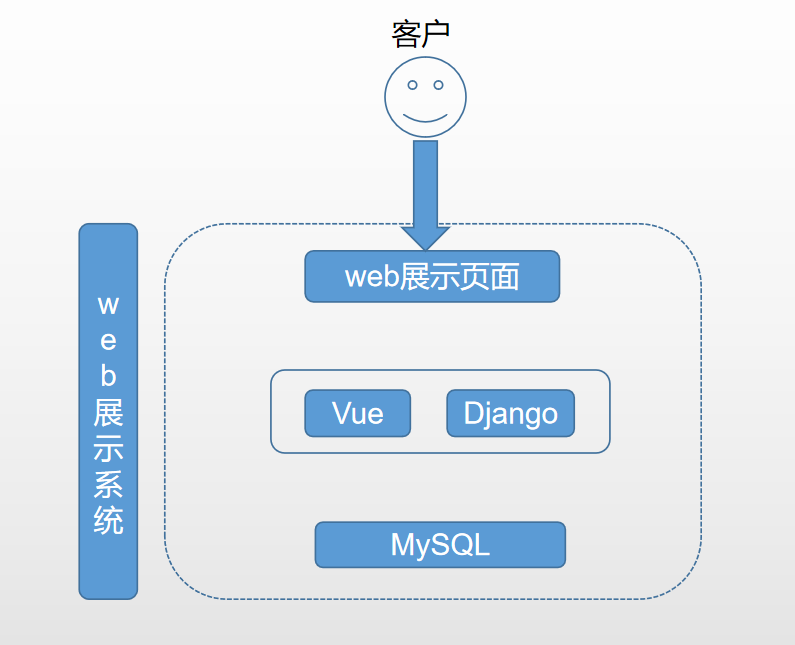
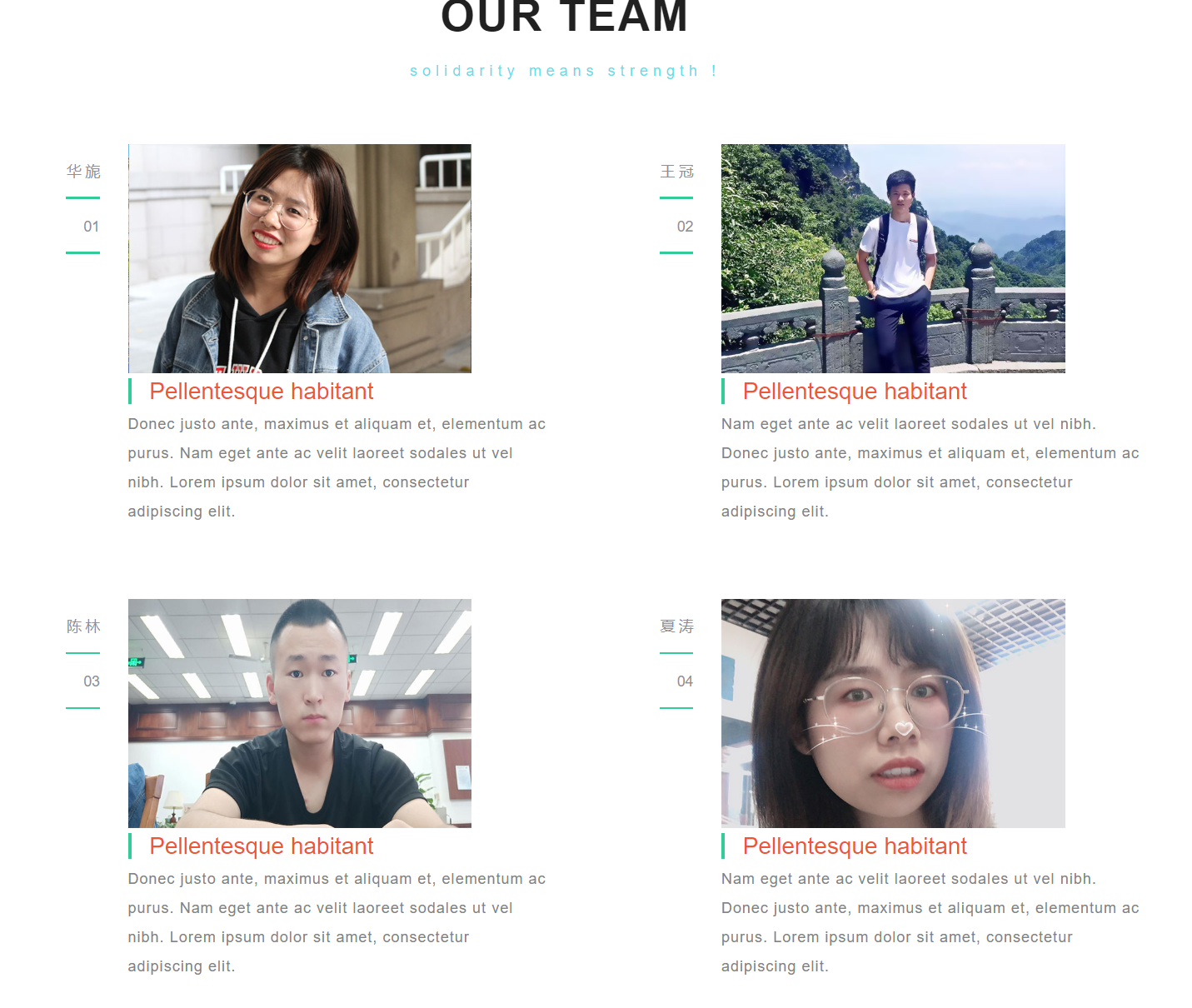


图2 web展示系统架构

#### 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。

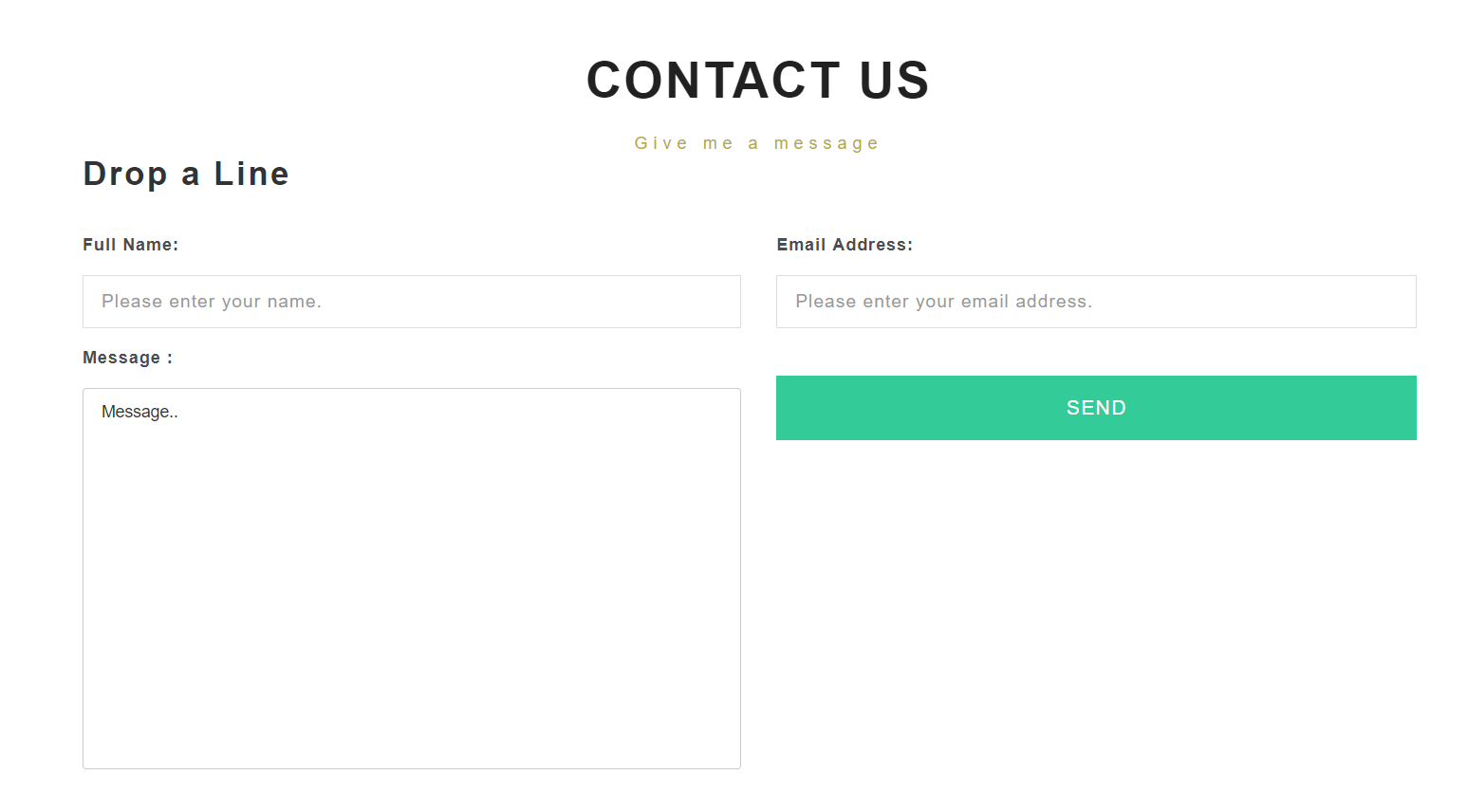
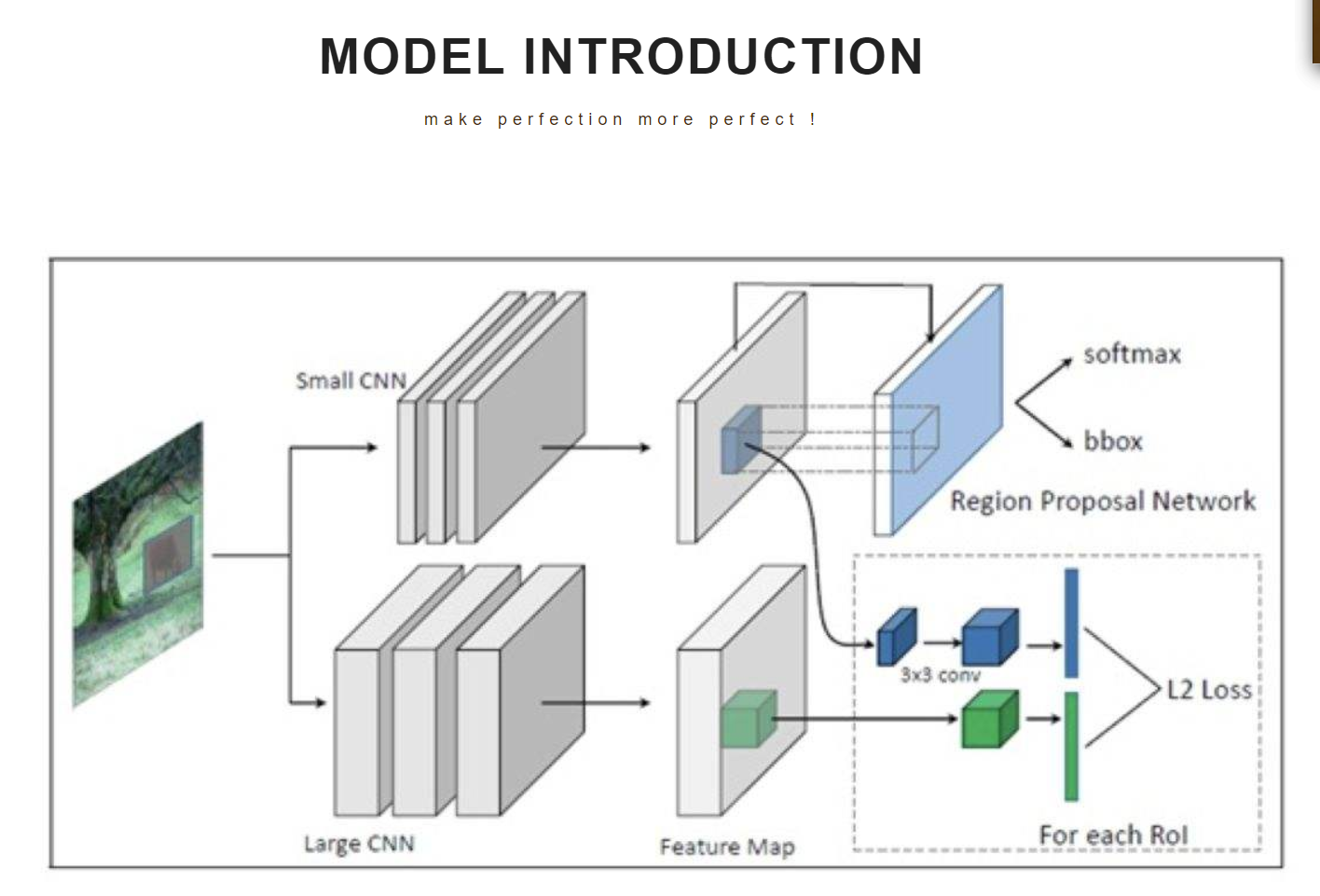
 

图3 系统页面展示

#### 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。

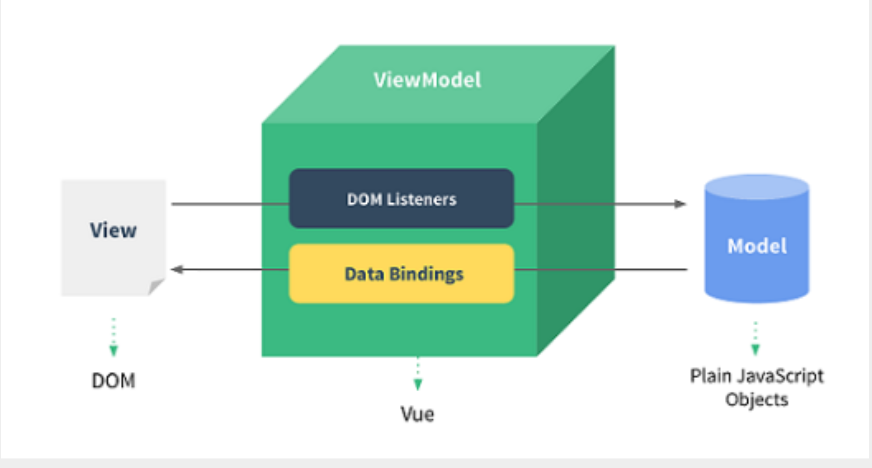


图4 Vue框架

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。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。

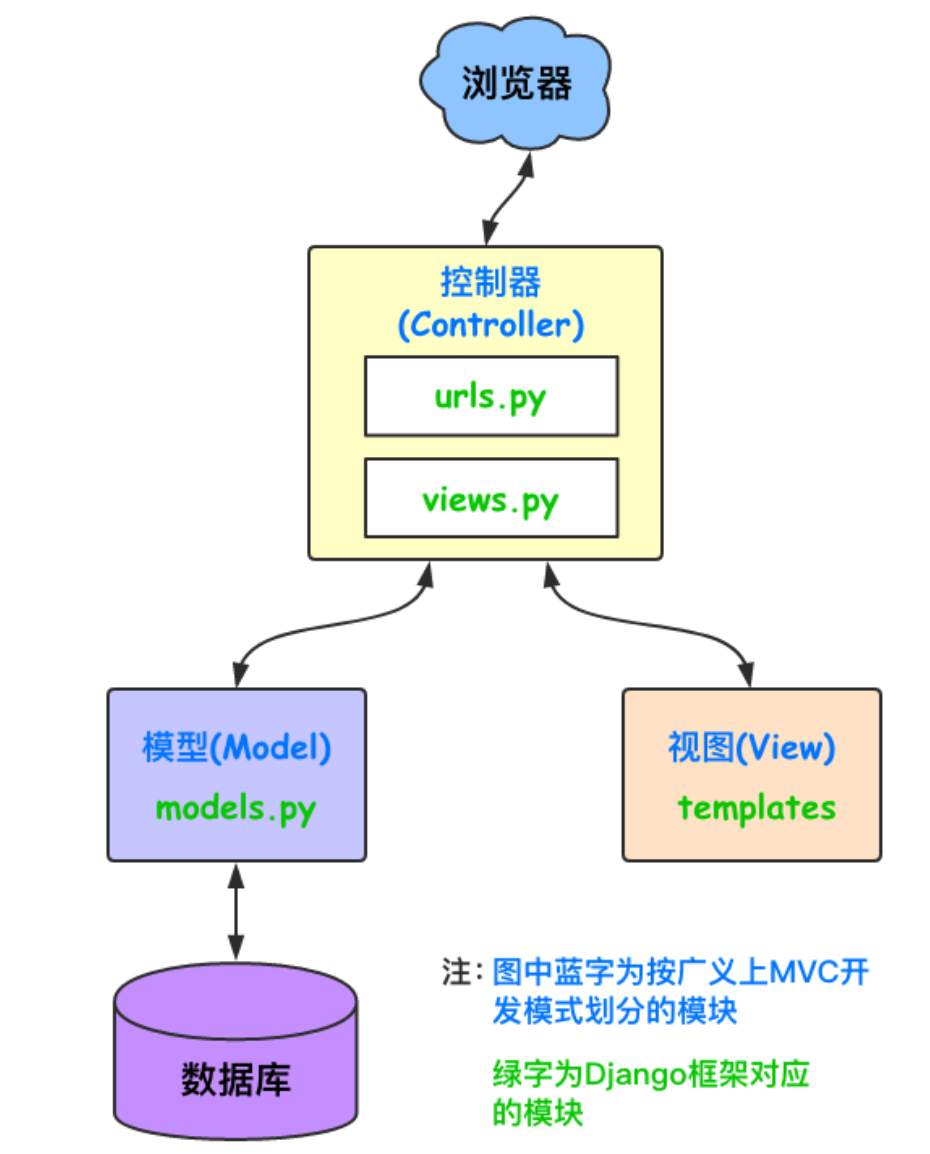


图5 Django框架

### 模型层

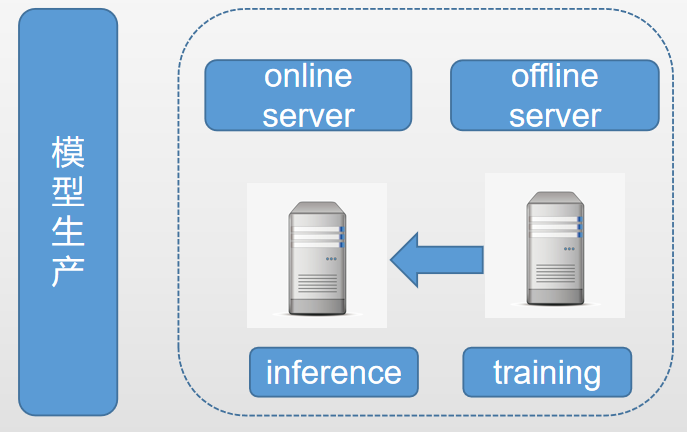


图6 模型层架构

模型层主要负责对模型进行训练，进行预测。关于faster-RCNN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。

### 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。

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，安防，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大致的了解，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第一个系统。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路由设置都遇到了大量的问题，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、高并发等问题，因此当前的系统只能用于演示，还不能算是真正意义上的线上的生产系统。本次课程设计我的主要目标是训练自己实现系统的能力。因此没有过多的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！