**Django开发网页原理**

1. Django是一个开源的Web应用框架。采用了MVC的框架模式，即模式M(Model)，视图V(View)和控制器C()。

Django的核心

1. 用于创建模型的对象关系映射(ORM)容易使用的数据库API，同时也支持原始的SQL语句

2. 自动化管理界面Admin，不必花大量时间创建后台管理。

3. URL地址路由设计(采用正则表达式匹配)

4. 方便可重用的模板语言

**工作机制：**

1.用manage .py runserver 启动Django服务器时就载入了在同一目录下的settings .py。该文件包含了项目中的配置信息，如前面讲的URLConf等，其中最重要的配置就是ROOT\_URLCONF，它告诉Django哪个Python模块应该用作本站的URLConf，默认的是urls .py

2.当访问url的时候，Django会根据ROOT\_URLCONF的设置来装载URLConf。

3.然后按顺序逐个匹配URLConf里的URLpatterns。如果找到则会调用相关联的视图函数，并把HttpRequest对象作为第一个参数(通常是request)

4.最后该view函数负责返回一个HttpResponse对象。

开发过程：

**2）web应用的前后端**

1. 前端用bootstrap框架做css开发。节省了前端开发的工作强度。

2. 前端用原生JavaScript开发。用户通过点击图标”+”即可完成上传图片。

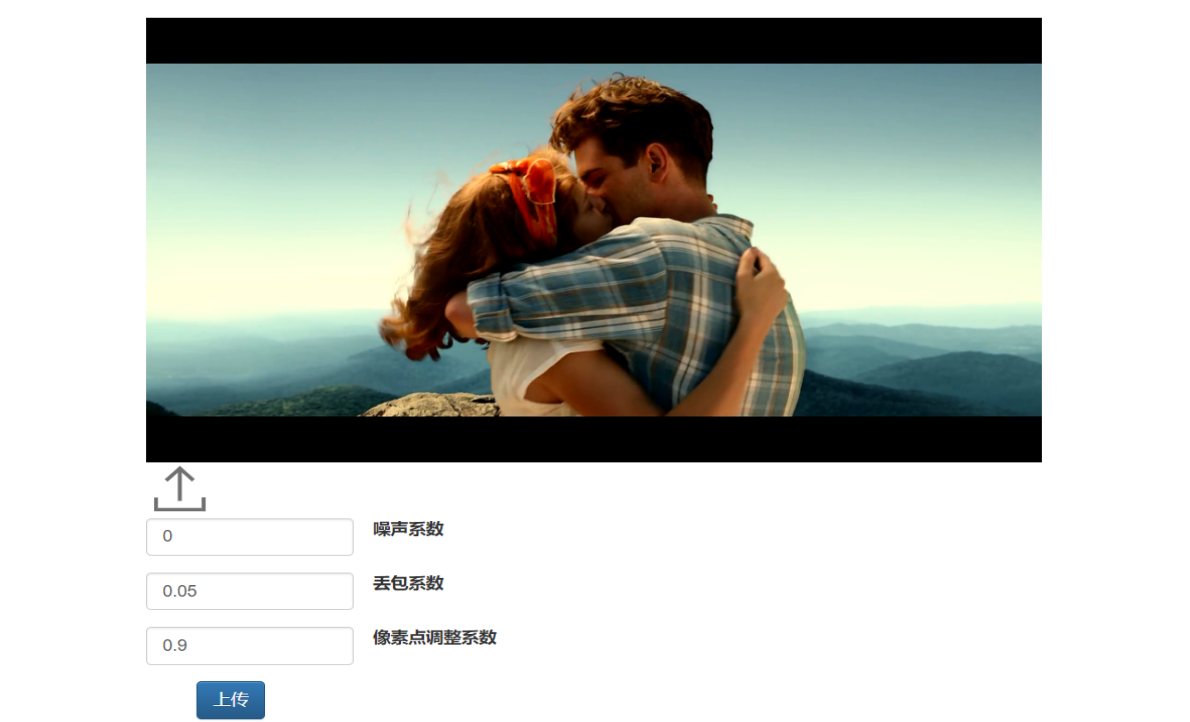
3. 上传图片后，Web前端JavaScript脚本将用户上传的文件通过Ajax发送到后台。Ajax实现异步请求避免页面重新刷新影响用户体验。

4. 前端的实现过程，首先，通过DOM操作把上传的文件获取到。然后应用AJAX的FormData序列化表格，使后台能够把文件信息解析出来。

后台解析到上传的图像文件。调用encode\_main()函数实现图像的加密。加密后的图像保存到/static/encode/images/.下，由于Django需要将图像文件存到静态文件，所以需要指定路径，在不同的机器上路径需要重新配置。返回保存的图片的路径，由前端负责解析图像文件，动态加载图像。

四、测试

4.3项目测试展示



4.1测试一张彩色图片，传递的噪声参数为0，丢包系数为0.05，像素点调正为0.9，得到的解密图片，PSNR为83



4.2测试一张灰度图，传递噪声系数为0.2，丢包系数为0.02，像素点调整系数为0.9

原图



解密后的图



PSNR为87

五、操作指南

5.1依赖环境

1. python3.7

Django2.1.2

Chrome浏览器

2.使用步骤

1. 桌面使用命令行路径切换到11\_ev/myblog/Script/

2. 执行命令./activate

3. 文件路径跳转到11\_env/myblog/

4. 执行python manage.py runserver启动本地服务器

5. 在浏览器输入localhost:8000/encode转到加密主页

6. 点击上传图片(支持主流的图片格式，彩色会先转换为bmp灰度图进行加密)

7. 经过1s后，页面弹出加密后的图像，点击可以查看大图，同时显示psnr的数值作为参考。

8. 需要重复测试再次点击上传图片即可，可以选择丢包系数等参数(设置了默认值)。

六、核心源码