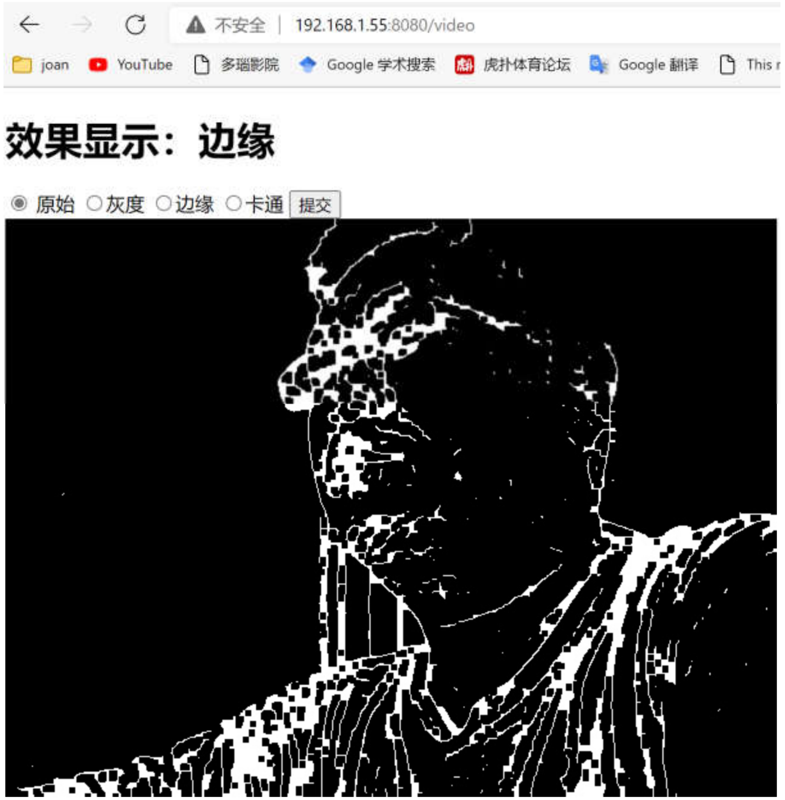
智能系统与控制



效果显示：边缘

● 原始 ○灰度 ○边缘 ○卡通 提交

树莓派网络控制：视频特效

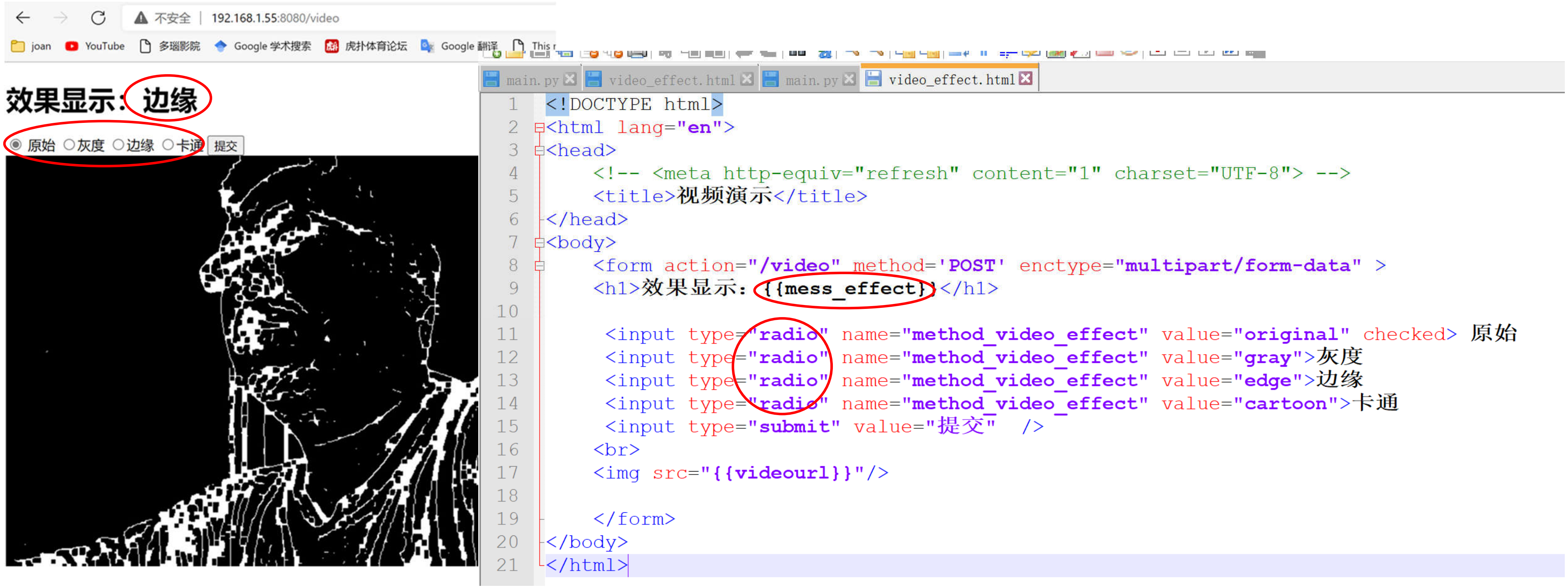
于泓

鲁东大学

信息与电气工程学院

2022.6.11

视频特效的实现



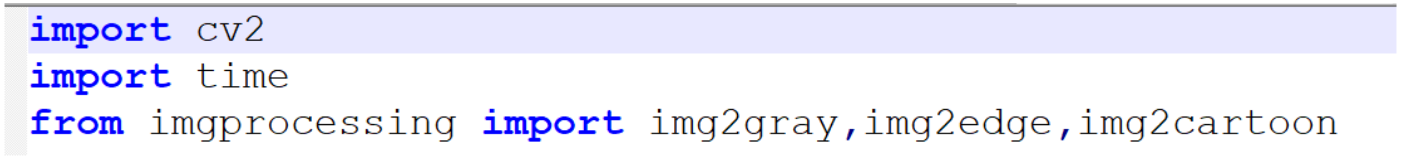
效果显示：边缘

● 原始 ○ 灰度 ○ 边缘 ○ 卡通 提交

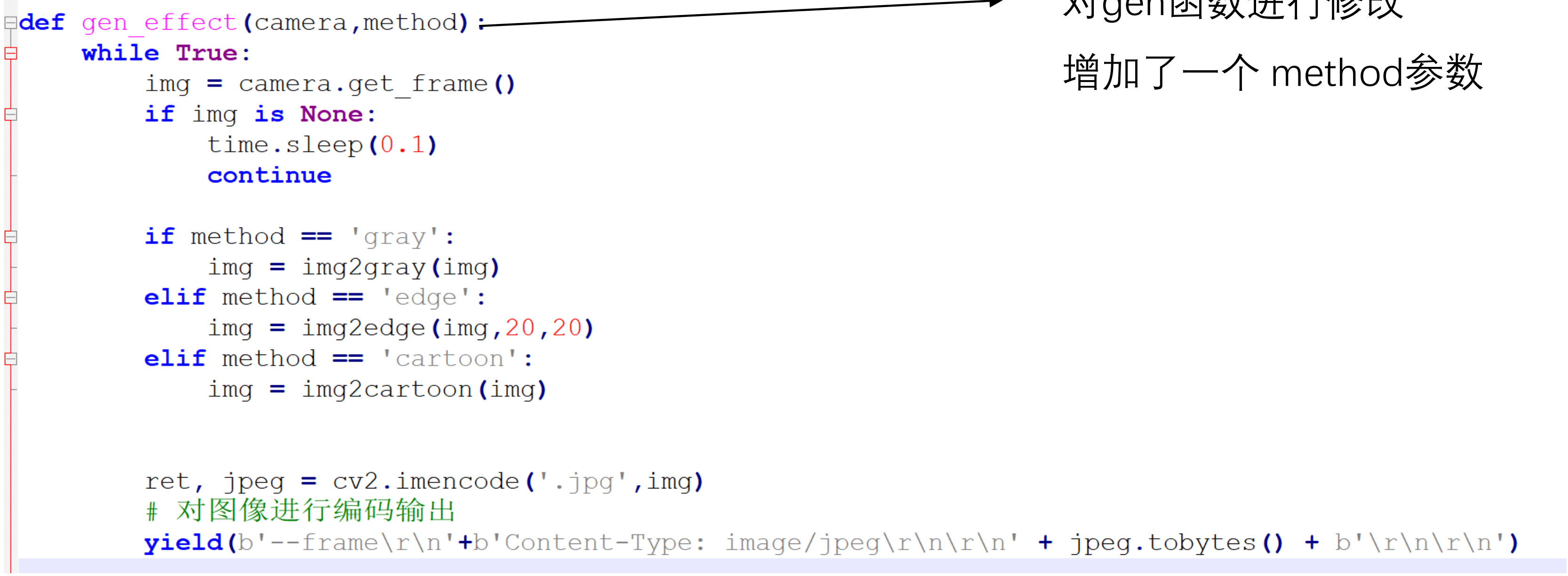
```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- <meta http-equiv="refresh" content="1" charset="UTF-8"> -->
    <title>视频演示</title>
</head>
<body>
    <form action="/video" method='POST' enctype="multipart/form-data" >
    <h1>效果显示：{{mess_effect}}</h1>

     <input type="radio" name="method_video_effect" value="original" checked> 原始
     <input type="radio" name="method_video_effect" value="gray">灰度
     <input type="radio" name="method_video_effect" value="edge">边缘
     <input type="radio" name="method_video_effect" value="cartoon">卡通
     <input type="submit" value="提交"  />
    <br>
    <img src="{{videourl}}"/>

    </form>
</body>
</html>
```



```
import cv2
import time
from imgprocessing import img2gray,img2edge,img2cartoon
```



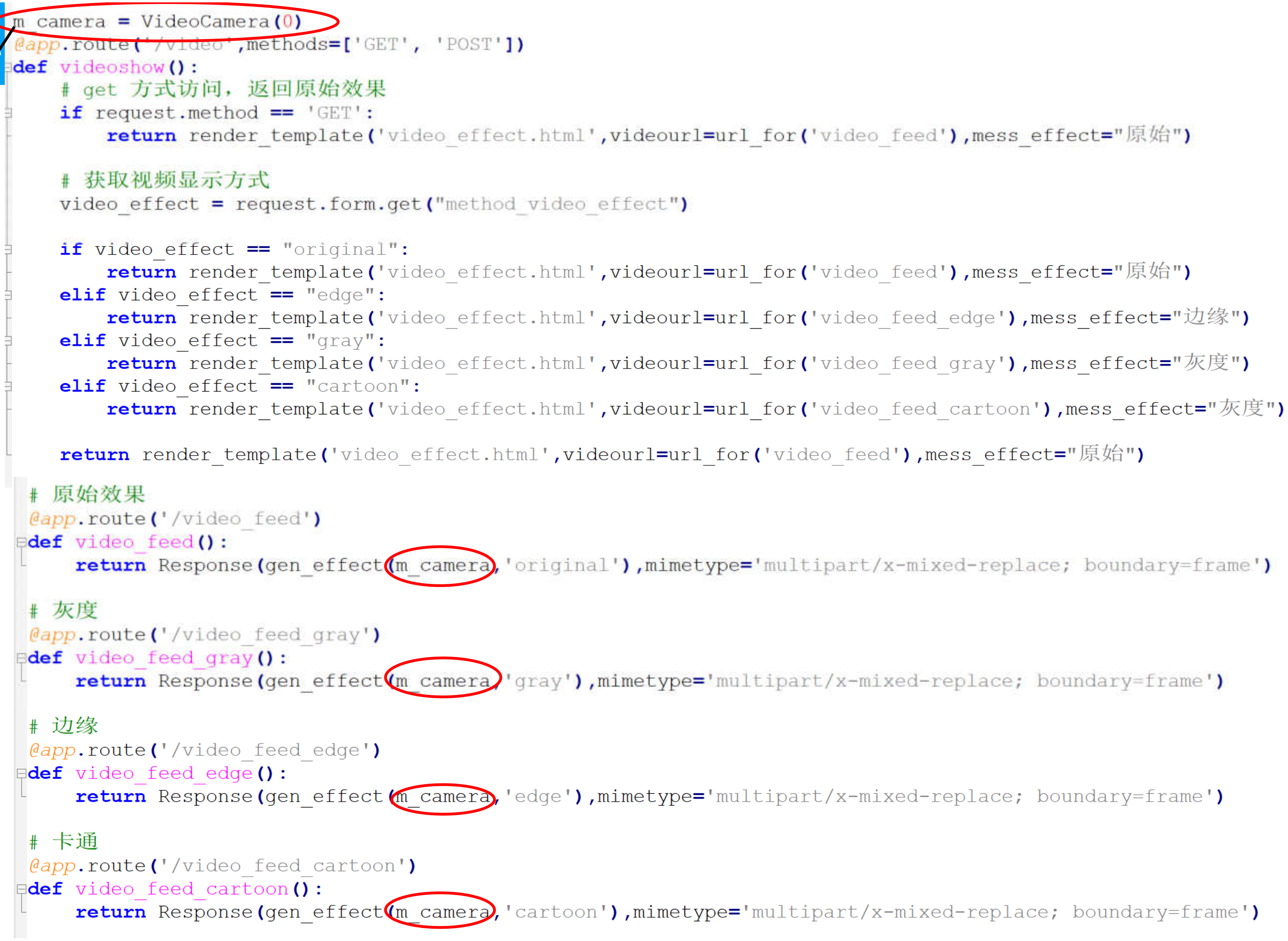
```
def gen_effect(camera,method):
    while True:
        img = camera.get_frame()
        if img is None:
            time.sleep(0.1)
            continue

        if method == 'gray':
            img = img2gray(img)
        elif method == 'edge':
            img = img2edge(img,20,20)
        elif method == 'cartoon':
            img = img2cartoon(img)


        ret, jpeg = cv2.imencode('.jpg',img)
        # 对图像进行编码输出
        yield(b'--frame\r\n'+b'Content-Type: image/jpeg\r\n\r\n' + jpeg.tobytes() + b'\r\n\r\n')
```

对gen函数进行修改

增加了一个 method参数



```
m_camera = VideoCamera(0)
@app.route('/video',methods=['GET', 'POST'])
def videoshow():
    # get 方式访问，返回原始效果
    if request.method == 'GET':
        return render_template('video_effect.html',videourl=url_for('video_feed'),mess_effect="原始")

    # 获取视频显示方式
    video_effect = request.form.get("method_video_effect")

    if video_effect == "original":
        return render_template('video_effect.html',videourl=url_for('video_feed'),mess_effect="原始")
    elif video_effect == "edge":
        return render_template('video_effect.html',videourl=url_for('video_feed_edge'),mess_effect="边缘")
    elif video_effect == "gray":
        return render_template('video_effect.html',videourl=url_for('video_feed_gray'),mess_effect="灰度")
    elif video_effect == "cartoon":
        return render_template('video_effect.html',videourl=url_for('video_feed_cartoon'),mess_effect="灰度")

    return render_template('video_effect.html',videourl=url_for('video_feed'),mess_effect="原始")

# 原始效果
@app.route('/video_feed')
def video_feed():
    return Response(gen_effect(m_camera,'original'),mimetype='multipart/x-mixed-replace; boundary=frame')

# 灰度
@app.route('/video_feed_gray')
def video_feed_gray():
    return Response(gen_effect(m_camera,'gray'),mimetype='multipart/x-mixed-replace; boundary=frame')

# 边缘
@app.route('/video_feed_edge')
def video_feed_edge():
    return Response(gen_effect(m_camera,'edge'),mimetype='multipart/x-mixed-replace; boundary=frame')

# 卡通
@app.route('/video_feed_cartoon')
def video_feed_cartoon():
    return Response(gen_effect(m_camera,'cartoon'),mimetype='multipart/x-mixed-replace; boundary=frame')
```

作为全局变量