

摘要：初步开发三叶青图像识别 web 网页

1. 实验准备

训练好的模型文件、类别标签文件、python 代码

2. 实验步骤

2.1. 编写接口代码

通过编写 flask 框架中的路由装饰器,来实现前后端的交互

```
# sort probability
index_pre.sort(key=lambda x: x[1], reverse=True)
text = [template.format( *args: k, v) for k, v in index_pre]
return_info = {"result": text}
except Exception as e:
    return_info = {"result": [str(e)]}
return return_info

xuenai0wy3
@app.route( rule: "/predict", methods=["POST"])
@torch.no_grad()
def predict():
    image = request.files["file"]
    img_bytes = image.read()
    info = get_prediction(image_bytes=img_bytes)
    return jsonify(info)

xuenai0wy3
@app.route( rule: "/", methods=["GET", "POST"])
def root():
    return render_template("up.html")
```

2.2. 编写前端代码

使用 html、css、js 编写的前端代码，这里只是在网页上的展示，后续前端页面再修改。

```
<body>
<!--<h3>请选择图片文件: PNG/JPG/JPEG/SVG/GIF</h3-->
<div style="...">
  <div style="...">
    <a href="javascript:;" class="file">选择文件
      <input type="file" name="file" id="file0"><br>
    </a>
    <img src="" id="img0" style="...">
  </div>
  <div style="...">
    <input type="button" id="b0" onclick="test()" value="预测">
    <pre id="out" style="..."></pre>
  </div>
</div>

<script type="text/javascript">
  $("#file0").change(function(){
    var objUrl = getObjectURL(this.files[0]); //获取文件信息
    console.log("objUrl = "+objUrl);
    if (objUrl) {
      $("#img0").attr("src", objUrl);
    }
  });

  function test() {
```

3. 实验结果

图像在线识别功能暂且实现，后续还要添加数据库等，再对 web 进行进一步的开发。

