

Web前端开发-作业6

1. 运行说明

1. 安装 Node.js: 终端输入 `npm install` ;
2. 输入 `node server.js` , 启动服务端;
3. 在 WebStorm 或浏览器中打开 `login.html` 以进入登录页面。

2. 实验功能和设计

使用 `opencv.js` 实现二级页面的卡通化, 主要包括:

1. 彩色图片转成灰度图;
2. 对灰度图进行高斯模糊;
3. 图像二值化;
4. 再次对二值化图像进行模糊;
5. 再次进行二值化;
6. 图像开运算。

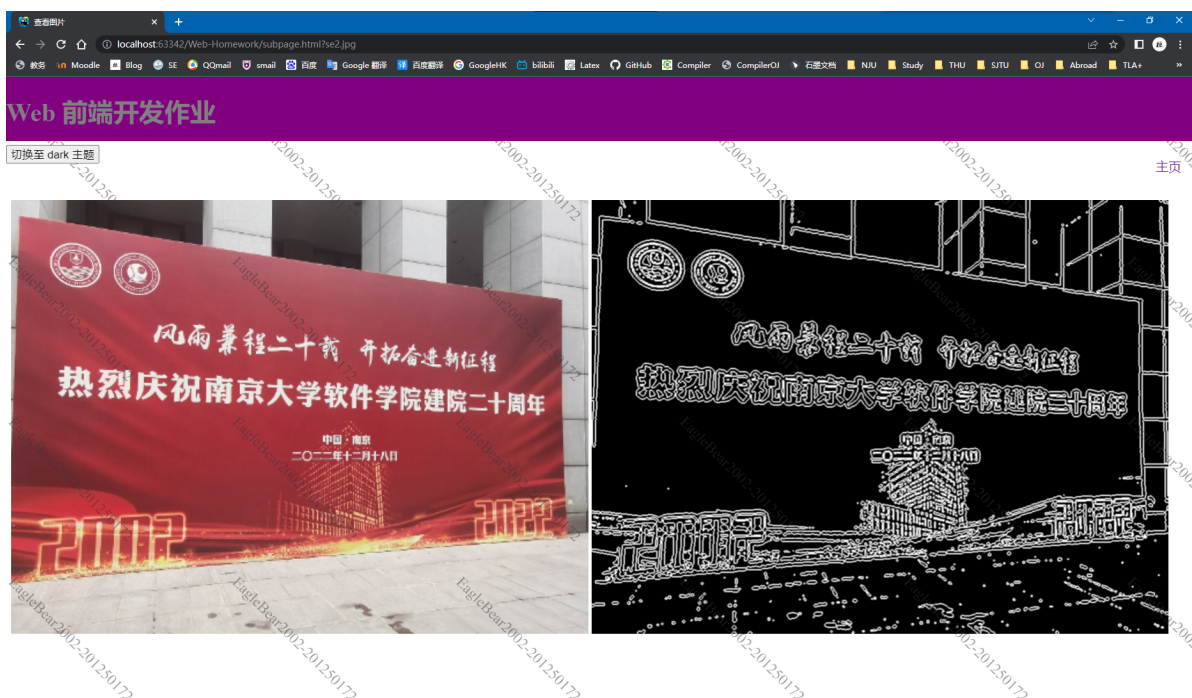
```
1  cv['onRuntimeInitialized'] = () => {
2      console.log("onRuntimeInitialized");
3      let canvasOutput = document.getElementById('canvasOutput');
4      let imgElement = document.getElementById('image1');
5      document.getElementById("image1").src = document.getElementById("image1").src;
6      imgElement.onload = function () {
7          let img_origin = cv.imread(imgElement);
8          let img_target = new cv.Mat();
9          img(img_origin, img_target);
10         img_origin.delete();
11         img_target.delete();
12     }
13 };
14
15 function img(img_origin, img_target) {
16     console.log("img");
17     let img_gray = cvtColor(img_origin);
18     let ksize1 = new cv.Size(5, 5);
19     let img_blurred1 = GaussianBlur(img_gray, ksize1);
20     let img_threshold1 = adaptiveThreshold(img_blurred1);
21     let img_blurred2 = GaussianBlur(img_threshold1, ksize1);
22     let img_threshold2 = threshold(img_blurred2);
23     let img_opening = bitwise_not(img_threshold2);
24     let ksize2 = new cv.Size(3, 3);
25     let img_opening_blurred = GaussianBlur(img_opening, ksize2);
26     img_target = img_opening_blurred;
27     cv.imshow('canvasOutput', img_target);
28 }
29
30 function cvtColor(img_origin) {
31     console.log("cvtColor");
32     let img_gray = new cv.Mat();
33     cv.cvtColor(img_origin, img_gray, cv.COLOR_RGBA2GRAY, 0);
34     return img_gray;
35 }
```

```

36
37 function GaussianBlur(img_origin, ksize) {
38     console.log("GaussianBlur");
39     let img_blurred = new cv.Mat();
40     cv.GaussianBlur(img_origin, img_blurred, ksize, 0);
41     return img_blurred;
42 }
43
44 function adaptiveThreshold(img_origin) {
45     console.log("adaptiveThreshold");
46     let img_threshold = new cv.Mat();
47     cv.adaptiveThreshold(img_origin, img_threshold, 255, cv.ADAPTIVE_THRESH_GAUSSIAN_C,
48 cv.THRESH_BINARY, 5, 2);
49     return img_threshold;
50 }
51
52 function threshold(img_origin) {
53     console.log("threshold");
54     let img_threshold = new cv.Mat();
55     cv.threshold(img_origin, img_threshold, 200, 255, cv.THRESH_BINARY);
56     return img_threshold;
57 }
58
59 function bitwise_not(img_origin) {
60     console.log("bitwise_not");
61     let img_opening = new cv.Mat();
62     let M = new cv.Mat();
63     let ksize = new cv.Size(3, 3);
64     M = cv.getStructuringElement(cv.MORPH_CROSS, ksize);
65     cv.morphologyEx(img_origin, img_opening, cv.MORPH_GRADIENT, M);
66     return img_opening;
67 }
68
69 Module.onRuntimeInitialized();

```

3. 实现效果



4. 参考文献

1. [OpenCV.js实现乔丹动图素描效果图文教程](#)