|  |  |
| --- | --- |
| **Российский университет транспорта (МИИТ)**  **Институт транспортной техники и систем управления**  **Кафедра «Управление и защита информации»** | |
| **Задание №6**  **по теме «Анализ изображения при использовании инструментов фреймворка*Vue.js*»**  **по дисциплине «Web-программирование»** | |
|  | Выполнил:  Студент группы ТКИ-542  Шуриков Д. А.  Кожак И. А.  Проверил:  Доцент кафедры УиЗИ, к.т.н.  Сафронов А.И. |
| Москва 2024 | |

Оглавление

[1. ЦЕЛЬ РАБОТЫ 3](#_Toc182837276)

[2. ФОРМУЛИРОВАКА ЗАДАЧИ 3](#_Toc182837277)

[3. ТАБЛИЦА СООТВЕТСВИЯ ПЕРЕМЕННЫХ 3](#_Toc182837278)

[4. WEB-СТРАНИЦА 4](#_Toc182837279)

[4.1. Код web-приложения 4](#_Toc182837280)

[4.2. Результат отображения в браузере 11](#_Toc182837281)

[4.3. Сети Петри 16](#_Toc182837282)

[5. ВЫВОД ПО РАБОТЕ 17](#_Toc182837283)

1. Цель работы

Провести анализ изображения, применить навыки создания простых локальных одностраничных web-приложений под управлением Фреймворка Vue.js на языке JavaScript.

1. ФОРМУЛИРОВАКА ЗАДАЧИ

Средствами *SVG* или *Canvas* под управлением фреймворка *Vue.js* построить программное, браузерное *web*-обеспечение, реализующее анализ подгруженного изображения (*\*.jpeg*) по варианту.

1. таблица соответсвия переменных

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Цветовая модель** | **Объект графики на web-странице** | **Реализация управления гистограммой** | **Режим вывода загруженного изображения** | **Режим построения гистограммы** | **Вариант** |
| **16** | *CMYK* | *SVG* | Суммарная / Каждый канал – своя опция | Цветовая инверсия | Прямой (слева-направо) | **16** |

1. Web-страница
   1. Код ImageAnalyzer.vue

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Task 6</title>

<script src="js/vue-2-6-14.js"></script>

</head>

<body>

<div id="app">

<div style="display: flex; flex-direction: column; align-items: center; justify-content: center; text-align: center; gap: 20px">

<h1>Task 6</h1>

<div style="border: 1px solid black; padding: 10px;">

<h3>Выберите изображение из файла</h3>

<input type="file" @change="onFileChange" accept="image/jpeg" />

</div>

<div v-show="imageSrc">

<table style="max-width: 600px;">

<tr>

<th>Оригинал</th>

<th>Оригинал инверсия</th>

<th>CMYK</th>

<th>CMYK инверсия</th>

</tr>

<tr>

<td>

<img :src="imageSrc" alt="Uploaded Image" style="max-width: 300px;" />

</td>

<td>

<svg style="width: 300px; height: 300px"><image id="rgb-negative-svg-image" x="0" y="0" width="100%" height="100%"/></svg>

</td>

<td>

<svg style="width: 300px; height: 300px"><image id="cmyk-svg-image" x="0" y="0" width="100%" height="100%"/></svg>

</td>

<td>

<svg style="width: 300px; height: 300px"><image id="cmyk-negative-svg-image" x="0" y="0" width="100%" height="100%"/></svg>

</td>

</tr>

</table>

<div style="display: flex; flex-direction: column; align-items: center; justify-content: center; text-align: center; gap: 20px">

<h4>Гистограмма</h4>

<div>

<label>

<input type="radio" value="sum" v-model="mode" /> Суммарная

</label>

<label>

<input type="radio" value="cyan" v-model="mode" /> Cyan

</label>

<label>

<input type="radio" value="magenta" v-model="mode" /> Magenta

</label>

<label>

<input type="radio" value="yellow" v-model="mode" /> Yellow

</label>

</div>

<svg width="255" height="180" style="background: #f8f8f8; zoom: 1.5">

<text x="0" y="170" font-family="Arial" font-size="12" fill="black">0</text>

<text x="255" y="170" font-family="Arial" font-size="12" fill="black" text-anchor="end">255</text>

<text x="127" y="175" font-family="Arial" font-size="12" fill="black" text-anchor="middle">Интенсивность</text>

<text x="127" y="15" font-family="Arial" font-size="12" fill="black" text-anchor="middle">Количество пикселей</text>

<line v-if="mode === 'cyan'"

v-for="(value, index) in histogramData.c"

:key="'c' + index"

:x1="index"

y1="150"

:x2="index"

:y2="150 - value"

stroke="cyan"

:stroke-width="1"

/>

<line v-if="mode === 'magenta'"

v-for="(value, index) in histogramData.m"

:key="'m' + index"

:x1="index"

y1="150"

:x2="index"

:y2="150 - value"

stroke="magenta"

:stroke-width="1"

/>

<line v-if="mode === 'yellow'"

v-for="(value, index) in histogramData.y"

:key="'y' + index"

:x1="index"

y1="150"

:x2="index"

:y2="150 - value"

stroke="yellow"

:stroke-width="1"

/>

<line v-if="mode === 'sum'"

v-for="(value, index) in histogramData.k"

:key="'k' + index"

:x1="index"

y1="150"

:x2="index"

:y2="150 - value"

stroke="black"

:stroke-width="1"

/>

</svg>

</div>

</div>

</div>

</div>

<script>

function rgbToCmyk(r, g, b) {

let c = (r / 255); // 1 - (r / 255)

let m = (g / 255); // 1 - (g / 255)

let y = (b / 255); // 1 - (b / 255)

let k = 1 - Math.max(c, m, y); // Math.min(c, Math.min(m, y))

if (k === 1) return [0, 0, 0, 1];

c = (1 - c - k) / (1 - k); // (c - k) / (1 - k)

m = (1 - m - k) / (1 - k); // (m - k) / (1 - k)

y = (1 - y - k) / (1 - k); // (y - k) / (1 - k)

return [c, m, y, k];

}

new Vue({

el: '#app',

data() {

return {

imageSrc: null,

histogramData: { c: [], m: [], y: [], k: [] },

mode: 'sum'

};

},

methods: {

onFileChange(e) {

const file = e.target.files[0];

if (!file) return;

const image = new Image();

image.onload = () => {

this.imageSrc = image.src;

this.analyzeImage(image);

};

image.src = URL.createObjectURL(file);

},

getCanvasAndContextForImage(image, dataSize) {

const canvas = document.createElement('canvas');

const context = canvas.getContext('2d');

canvas.width = image.width;

canvas.height = image.height;

let data = undefined;

if (dataSize) data = new Uint8ClampedArray(dataSize);

return { canvas: canvas, context: context, data: data }

},

loadDataToSvg(plot, svgImageId) {

const imageData = new ImageData(plot.data, plot.canvas.width, plot.canvas.height);

plot.context.putImageData(imageData, 0, 0);

document.getElementById(svgImageId).setAttribute('href', plot.canvas.toDataURL());

},

analyzeImage(image) {

const rgb = this.getCanvasAndContextForImage(image);

rgb.context.drawImage(image, 0, 0, image.width, image.height);

const { data } = rgb.context.getImageData(0, 0, image.width, image.height);

const rgbNegative = this.getCanvasAndContextForImage(image, data.length);

const cmyk = this.getCanvasAndContextForImage(image, data.length);

const cmykNegative = this.getCanvasAndContextForImage(image, data.length);

const histogramC = new Array(256).fill(0);

const histogramM = new Array(256).fill(0);

const histogramY = new Array(256).fill(0);

const histogramK = new Array(256).fill(0);

for (let i = 0; i < data.length; i += 4) {

const [c, m, y, k] = rgbToCmyk(data[i], data[i + 1], data[i + 2]);

histogramC[Math.floor(c \* 255)]++;

histogramM[Math.floor(m \* 255)]++;

histogramY[Math.floor(y \* 255)]++;

histogramK[Math.floor(k \* 255)]++;

rgbNegative.data[i] = 255 - data[i];

rgbNegative.data[i + 1] = 255 - data[i + 1];

rgbNegative.data[i + 2] = 255 - data[i + 2];

rgbNegative.data[i + 3] = data[i + 3];

cmyk.data[i] = c \* 255;

cmyk.data[i + 1] = m \* 255;

cmyk.data[i + 2] = y \* 255;

cmyk.data[i + 3] = 255; // Alpha

cmykNegative.data[i] = (1 - c) \* 255;

cmykNegative.data[i + 1] = (1 - m) \* 255;

cmykNegative.data[i + 2] = (1 - y) \* 255;

cmykNegative.data[i + 3] = 255; // Alpha

}

const maxK = Math.max(...histogramK);

this.histogramData = {

c: histogramC.map(v => 150 \* v / maxK),

m: histogramM.map(v => 150 \* v / maxK),

y: histogramY.map(v => 150 \* v / maxK),

k: histogramK.map(v => 150 \* v / maxK),

};

console.log(this.histogramData.y)

this.loadDataToSvg(rgbNegative, 'rgb-negative-svg-image');

this.loadDataToSvg(cmyk, 'cmyk-svg-image');

this.loadDataToSvg(cmykNegative, 'cmyk-negative-svg-image');

}

}

});

</script>

</body>

</html>

* 1. Результат отображения в браузере

|  |
| --- |
|  |
| 1. – Результат запуска |
|  |
| 1. – Начальная страница. |
|  |
| 1. – Результат отображения гистограммы суммы. |
|  |
| 1. – Результат отображения гистограммы для цвета Cyan.      1. – Результат отображения гистограммы для цвета Magenta.      1. – Результат отображения гистограммы для цвета Yellow. |

* 1. Сети Петри

|  |
| --- |
| Изображение выглядит как текст, снимок экрана, круг  Автоматически созданное описание |
| 1. – Сети Петри. |

1. вывод по работе

В ходе нашей работы мы приобрели ценные практические навыки по созданию локальных одностраничных веб-приложений, используя современный и популярный фреймворк Vue.js. Этот инструмент позволяет эффективно управлять пользовательским интерфейсом и обеспечивает высокую производительность благодаря своей реактивности и компонентному подходу. Мы освоили ключевые аспекты работы с Vue.js, включая настройку проекта, создание и управление компонентами, работу с состоянием приложения, маршрутизацию, а также взаимодействие с данными через JavaScript. Эти знания открывают перед нами широкие возможности для разработки удобных, функциональных и масштабируемых пользовательских интерфейсов, которые соответствуют современным стандартам веб-разработки.