|  |  |
| --- | --- |
| **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  Пономарев А.Д.  Дроздов А.Д.  Проверил:  Доцент кафедры УиЗИ, к.т.н., с.н.с  Сафронов Ф.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 4](#_Toc182837281)

[4.3. Сети Петри 4](#_Toc182837282)

[5. ВЫВОД ПО РАБОТЕ 5](#_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** |
| **7** | *CMYK* | *Canvas* | Суммарная / Все каналы на одном | Прямой (оригинал) | Инверсный (справа-налево) | **5** |

1. Web-страница
   1. Код ImageAnalyzer.vue

<style>

body {

font-family: Arial, sans-serif;

}

.card {

margin: 15px 5px;

}

canvas {

border: solid 1px #999;

}

p {

margin: 5px 0 3px;

}

</style>

<template>

<div class="image-analyzer">

<div class="card">

<p>Выберите изображение из файла</p>

<input type="file" @change="handleFileChange">

<br>

<img :src="imageSrc" @load="getImageData" v-if="imageSrc">

</div>

<div class="card">

<p>Гистограмма</p>

<div>

<input type="radio" value="value" v-model="histogramMode" @change="drawHistogram">

<label>Значение</label>

<input type="radio" value="color" v-model="histogramMode" @change="drawHistogram">

<label>Цвет</label>

</div>

<canvas ref="histogramCanvas" width="256" height="150"></canvas>

</div>

</div>

</template>

<script>

export default {

data() {

return {

imageSrc: null,

imageData: null,

histogramMode: 'value',

histogramData: {

cyan: new Array(256).fill(0),

magenta: new Array(256).fill(0),

yellow: new Array(256).fill(0),

black: new Array(256).fill(0),

},

};

},

methods: {

handleFileChange(event) {

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

if (file) {

this.imageSrc = URL.createObjectURL(file);

}

},

getImageData(event) {

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

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

canvas.width = event.target.width;

canvas.height = event.target.height;

ctx.drawImage(event.target, 0, 0);

this.imageData = ctx.getImageData(0, 0, canvas.width, canvas.height).data;

this.calculateHistogram();

this.drawHistogram();

},

calculateHistogram() {

this.histogramData.cyan.fill(0);

this.histogramData.magenta.fill(0);

this.histogramData.yellow.fill(0);

this.histogramData.black.fill(0);

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

const r = this.imageData[i];

const g = this.imageData[i + 1];

const b = this.imageData[i + 2];

const c = 255 - r;

const m = 255 - g;

const y = 255 - b;

const k = Math.min(c, m, y);

this.histogramData.cyan[c]++;

this.histogramData.magenta[m]++;

this.histogramData.yellow[y]++;

this.histogramData.black[k]++;

}

},

drawHistogram() {

const canvas = this.$refs.histogramCanvas;

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

ctx.clearRect(0, 0, canvas.width, canvas.height);

const maxValue = Math.max(...this.histogramData.cyan, ...this.histogramData.magenta, ...this.histogramData.yellow, ...this.histogramData.black);

const drawChannel = (channel, color) => {

ctx.fillStyle = color;

for (let i = 0; i < 256; i++) {

const value = this.histogramData[channel][i];

const height = (value / maxValue) \* canvas.height;

ctx.fillRect(i, canvas.height - height, 1, height);

}

};

if (this.histogramMode === 'value') {

drawChannel('cyan', 'cyan');

drawChannel('magenta', 'magenta');

drawChannel('yellow', 'yellow');

drawChannel('black', 'black');

} else {

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

for (let i = 0; i < 256; i++) {

combined[i] = this.histogramData.cyan[i] + this.histogramData.magenta[i] + this.histogramData.yellow[i] + this.histogramData.black[i];

}

ctx.fillStyle = 'gray';

for (let i = 0; i < 256; i++) {

const value = combined[i];

const height = (value / maxValue) \* canvas.height;

ctx.fillRect(i, canvas.height - height, 1, height);

}

}

},

},

};

</script>

* 1. Код App.vue

<template>

<div id="app">

<ImageAnalyzer />

</div>

</template>

<script>

import ImageAnalyzer from './components/ImageAnalyzer.vue';

export default {

name: 'App',

components: {

ImageAnalyzer

}

}

</script>

<style>

body {

font-family: Arial, sans-serif;

}

</style>

* 1. Результат отображения в браузере

|  |
| --- |
|  |
| 1. – Результат запуска |
|  |
| 1. – Начальная страница |
|  |
| 1. – Результат отображения гистограммы значения |
|  |
| 1. – Результат отображения гистограммы для цвета |

* 1. Сети Петри

|  |
| --- |
|  |
| 1. – Сети Петри |

1. вывод по работе

Мы получили навыки по созданию локальных одностраничных web-приложение под управлением фреймворка Vue.js на языке JavaScript.