|  |  |
| --- | --- |
| **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  Зинченко Б.А.  Круглов В.А.  Проверил:  Доцент кафедры УиЗИ, к.т.н., с.н.с  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 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** |
| **5** | *CMYK* | *SVG* | Суммарная / Все каналы на одном | Прямой (оригинал) | Инверсный (справа-налево) | **13** |

1. Web-страница
   1. Код ImageAnalyzer.vue

<template>

<div style=**"text-align: center;"**>

<h3>**Выберите изображение из файла**</h3>

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

<div v-if=**"imageSrc"**>

<img :src=**"imageSrc"** alt=**"Uploaded Image"** style=**"max-width: 100%; margin: 20px 0;"** />

<div>

<h4>**Гистограмма**</h4>

<label>

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

</label>

<label>

<input type=**"radio"** value=**"all"** v-model=**"mode"** /> **Все каналы**

</label>

<svg width=**"300"** height=**"150"** style=**"background: #f8f8f8;"**>

<line

v-for="(value, index) in displayedHistogram('c')"

:key="'c' + index"

:x1="300 - index"

y1="150"

:x2="300 - index"

:y2="150 - value"

stroke="cyan"

:stroke-width="1"

/>

<line

v-for="(value, index) in displayedHistogram('m')"

:key="'m' + index"

:x1="300 - index"

y1="150"

:x2="300 - index"

:y2="150 - value"

stroke="magenta"

:stroke-width="1"

/>

<line

v-for="(value, index) in displayedHistogram('y')"

:key="'y' + index"

:x1="300 - index"

y1="150"

:x2="300 - index"

:y2="150 - value"

stroke="yellow"

:stroke-width="1"

/>

<line

v-for="(value, index) in displayedHistogram('k')"

:key="'k' + index"

:x1="300 - index"

y1="150"

:x2="300 - index"

:y2="150 - value"

stroke="black"

:stroke-width="1"

/>

</svg>

</div>

</div>

</div>

</template>

<script>

// Функция для преобразования RGB в CMYK

**function** rgbToCmyk**(**r**,** g**,** b**)** **{**

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

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

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

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

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

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

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

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

**return** **[**c**,** m**,** y**,** k**];**

**}**

**export** **default** **{**

data**()** **{**

**return** **{**

imageSrc**:** **null,**

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

mode**:** 'sum'**,** // Default mode

**};**

**},**

computed**:** **{**

displayedHistogram**()** **{**

**return** **(**channel**)** **=>** **{**

**if** **(**this.mode **===** 'sum'**)** **{**

**return** this.histogramData.k**;**

**}**

**return** this.histogramData**[**channel**];**

**};**

**},**

**},**

methods**:** **{**

onFileChange**(**e**)** **{**

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

**if** **(**file**)** **{**

**const** reader **=** **new** FileReader**();**

reader.onload **=** **(**event**)** **=>** **{**

this.imageSrc **=** event.target.result**;**

this.$nextTick**(()** **=>** **{**

this.analyzeImage**();**

**});**

**};**

reader.readAsDataURL**(**file**);**

**}**

**},**

analyzeImage**()** **{**

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

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

**const** image **=** **new** Image**();**

image.src **=** this.imageSrc**;**

image.onload **=** **()** **=>** **{**

canvas.width **=** image.width**;**

canvas.height **=** image.height**;**

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

**const** imageData **=** context.getImageData**(**0**,** 0**,** image.width**,** image.height**);**

**const** **{** data **}** **=** imageData**;**

**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**)]++;**

**}**

**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**),**

**};**

**};**

**},**

**},**

**};**

</script>

<style>

**/\* Стили \*/**

**input[type="file"] {**

**margin: 10px 0;**

**}**

**svg {**

**border: 1px solid #ddd;**

**margin-top: 10px;**

**}**

</style>

* 1. Код App.vue

<template>

<div id="app">

<h1>Анализатор изображения</h1>

<ImageAnalyzer />

</div>

</template>

<script>

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

export default {

name: 'App',

components: {

ImageAnalyzer,

},

};

</script>

* 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.