# 前端代码部分

## index.js

import Vue from "vue";

import VueRouter from "vue-router";

//组件引入

import layout from "../components/layout/layout";

//心超部分

import echo\_home from "@/components/layout/echo\_component/echo\_home";

//颅内动脉部分

import int\_art\_home from "@/components/layout/int\_art/int\_art\_home";

//脑部影像部分

import brain\_home from "@/components/layout/brain/brain\_home";

//EOD部分

import eod\_home from "@/components/layout/eod\_system/eod\_home";

//眼脑电分析部分

import elec\_home from "@/components/layout/elec/elec\_home";

Vue.use(VueRouter);

const routes=[

{

path:"/",

name:"layout",

component:layout,

children:[

{

path:"/echo",

name:"echo",

component: echo\_home

},

{

path:"/int\_art",

name:"int\_art",

component: int\_art\_home

},

{

path:"/eod",

name:"eod",

component: eod\_home

},

{

path:"/elec",

name:"elec",

component: elec\_home

},

{

path:"/brain",

name:"brain",

component: brain\_home

},

]

},

];

const router=new VueRouter({

routes,

});

export default router;

## layout.vue

<template>

<v-app id="inspire">

<v-app-bar

app

color="white"

flat

height="60"

>

<div class="imed">

<div class="block" >

<el-image

style="width: 100px; height: 40px"

:src="require('@/assets/iMED.png')"

>

</el-image>

</div>

</div>

<span class="font-weight-black" style= "color: #a44848;font-size: 25px; width: 400px">

&nbsp;眼脑联动分析平台

</span>

<v-spacer></v-spacer>

<!-- <v-tabs-->

<!-- centered-->

<!-- class="ml-n9"-->

<!-- color="grey darken-1"-->

<!-- >-->

<!-- <v-tab-->

<!-- v-for="item in links"-->

<!-- :key="item.index"-->

<!-- @click="$router.push({name: item.to})"-->

<!-- >-->

<!-- {{ item.name }}-->

<!-- </v-tab>-->

<!-- </v-tabs>-->

</v-app-bar>

<v-main class="grey lighten-3">

<v-container>

<v-row>

<v-col

cols="12"

sm="2"

>

<div style="position: fixed">

<v-sheet

rounded="lg"

min-height="268"

>

<!--左侧导航栏-->

<div>

<v-list-item>

<v-list-item-content>

<v-list-item-title class="text-h6">

功能菜单

</v-list-item-title>

<v-list-item-subtitle>

</v-list-item-subtitle>

</v-list-item-content>

</v-list-item>

<v-divider></v-divider>

<v-list

dense

nav

>

<v-list-item

v-for="item in items\_1"

:key="item.title"

link

:to="item.route"

>

<!-- <v-list-item-icon>-->

<!-- <v-icon>{{ item.icon }}</v-icon>-->

<!-- </v-list-item-icon>-->

<div style="width:30px"></div>

<v-list-item-content>

<v-list-item-title> {{item.title}} </v-list-item-title>

</v-list-item-content>

<!-- <div style="width:20px"></div>-->

</v-list-item>

</v-list>

</div>

</v-sheet>

</div>

</v-col>

<v-col

cols="12"

sm="10"

>

<!--中间main区域-->

<router-view></router-view>

</v-col>

</v-row>

</v-container>

</v-main>

</v-app>

</template>

<script>

import {mapMutations} from "vuex";

export default {

name:"layout",

watch:{

selection(newValue) {

this.onUpdate(newValue)

}

},

data: () => ({

fileList:[],

selection:[],

FileData:[

{

name: 'Image File',

children:[

{name:"OD",children:[]},

{name:"thick\_gt",children:[]},

{name:"thin\_gt",children:[]},

]

},

],

//顶部导航栏数据

links: [

{

index:0,

name:"Home",

to:"carousel",

},

{

index:1,

name:"Study List",

to:"StudyList",

},

{

index:2,

name:"Result List",

to:"ResultList",

},

{

index:3,

name:"Editor",

to:"editor",

},

{

index:4,

name:"Analyze",

to:"analyze",

},

{

index:5,

name:"About",

to:"about",

},

],

//左侧导航栏数据

initiallyOpen: ['public'],

files: {

html: 'mdi-language-html5',

js: 'mdi-nodejs',

json: 'mdi-code-json',

md: 'mdi-language-markdown',

pdf: 'mdi-file-pdf',

png: 'mdi-file-image',

txt: 'mdi-file-document-outline',

xls: 'mdi-file-excel',

},

tree: [],

paramList:['none','none','Curvature','fractal dimension','vessel center line','vessel length','Area'],

items\_1: [

{ icon: "mdi-pencil-outline", title: "心衰检测系统", route: "/echo" },

{ icon: "mdi-google-analytics", title: "痴呆筛查系统", route: "/eod" },

{ icon: "mdi-google-analytics", title: "糖尿病视网膜筛查系统", route: "/elec" },

{ icon: "mdi-google-analytics", title: "颅内动脉瘤检测系统", route: "/int\_art" },

{ icon: "mdi-pencil-outline", title: "脑影像分析系统", route: "/brain" },

],

//中间main区域数据

//右侧导航栏数据

red: 0,

green: 0,

blue: 0,

Brightness:0,

}),

computed:{

},

methods:{

File(item){

console.log("item",item)

},

...mapMutations(['

Image']),

onClick(item){

console.log("item:",item.name)

},

onUpdate() {

// for(let i=0;i<this.selection.length;i++){

// console.log("selection:",this.selection[i])

// }

},

},

// created(){

// let path1="D:\\ZhiHu\_Store\\SUSTech\_Course\\MarsFindings\\Platform\\front\_end\\static\\pre\\OD"

// this.$store.dispatch('getFileList', {path1}).then((res)=>{

// res.forEach(item=>{

// let area={name:item,file:'png'}

// this.FileData[0].children[0].children.push(area)

// this.fileList.push(item)

// })

// })

// let tmpList=this.fileList

// this.$store.commit("

FileList",{tmpList})

// }

}

</script>

<style scoped>

.e4 {

width: 400px;

margin: auto;

}

.el-carousel\_\_item h3 {

color: #475669;

font-size: 18px;

opacity: 0.75;

line-height: 300px;

margin: 0;

}

.el-carousel\_\_item:nth-child(2n) {

background-color: #99a9bf;

}

.el-carousel\_\_item:nth-child(2n+1) {

background-color: #d3dce6;

}

</style>

## brain\_home.vue

<template>

<div>

<v-col

cols="10"

sm="10"

>

<v-sheet

min-height="70vh"

rounded="lg"

>

<v-card-title>

第一步 数据选择

</v-card-title>

<v-divider></v-divider>

<div style="padding: 20px">

<!-- 表格 -->

<el-table :data="FileData.slice((currentPage-1)\*pageSize,currentPage\*pageSize)" style="width: 100%">

<el-table-column label="文件名" width="400">

<template slot-scope="scope">

<div slot="reference" class="name-wrapper">

<el-tag>{{ scope.row.name }}</el-tag>

</div>

</template>

</el-table-column>

<el-table-column label="操作">

<template slot-scope="scope">

<el-button

size="mini"

@click="select(scope.row.name)">选择</el-button>

<el-button

size="mini"

type="danger"

@click="handleDelete(scope.$index, scope.row)">删除</el-button>

</template>

</el-table-column>

</el-table>

<!-- 分页器 -->

<div class="block" style="margin-top:15px;">

<el-pagination align='center' @size-

="handleSize

" @current-

="handleCurrent

"

:current-page="currentPage"

:page-sizes="[5,10,15]"

:page-size="pageSize"

layout="total, sizes, prev, pager, next, jumper"

:total="FileData.length">

</el-pagination>

</div>

</div>

<v-divider></v-divider>

<div style="padding-bottom: 20px">

<v-card-title>

已选择的数据

</v-card-title>

</div>

<v-row>

<div v-if="this.selected\_sample" style="padding: 20px">

<el-tag

effect="light"

type="success">

{{this.selected\_sample}}

</el-tag>

<v-btn

class="ma-2"

color="secondary"

@click="PreShow()"

>

文件预览

</v-btn>

<div v-show="pro1" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

</v-row>

<div v-show="pre\_show" style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in pre\_urlList"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="pre\_preView\_srcList">

>

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

<v-divider></v-divider>

<v-card-title>

第二步 结果分析

</v-card-title>

<div style="padding: 20px">

<v-btn

class="ma-2"

color="primary"

dark

@click="getResult()"

>

查看分析结果

</v-btn>

<div v-show="pro2" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

<div v-show="show">

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList.slice(0,3)"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

<el-button

type="success"

plain

icon="el-icon-setting"

size="mini"

@click="pluginDownload"

>下载完整参数及结果

</el-button>

</div>

<div>

<v-data-table

:headers="headers"

:items="infoList"

item-key="name"

class="elevation-1"

:search="search"

:custom-filter="filterOnlyCapsText"

>

<template v-slot:top>

<v-text-field

v-model="search"

label="Search (UPPER CASE ONLY)"

class="mx-4"

></v-text-field>

</template>

</v-data-table>

</div>

</div>

</v-sheet>

</v-col>

</div>

</template>

<script>

export default {

name: "brain\_home",

data:()=>({

FileData:[],

selected\_sample:'',

pre\_show:false,

button\_show:true,

show:false,

count:"",

time:null,

brain\_pre:null,

pic\_url:null,

partion\_url:null,

up\_url:null,

pdf\_url:null,

para:null,

pre\_urlList:[],

pre\_preView\_srcList:[],

res\_urlList:[],

res\_preView\_srcList:[],

pro1:false,

pro2:false,

currentPage: 1, // 当前页码

total: 0, // 总条数

pageSize: 5, // 每页的数据条数

infoList:[],

search: '',

calories: '',

}),

created() {

let content="brain"

this.$store.dispatch('getFileList', {content}).then((res)=>{

res.forEach(item=>{

let area={name:item}

this.FileData.push(area)

})

})

},

computed: {

headers () {

return [

{

text: 'Description',

align: 'start',

sortable: false,

value: 'Description',

},

{

text: 'Data',

align: 'end',

value: 'Data'

},

]

},

},

methods:{

pluginDownload(){

console.log(this.res\_urlList)

window.location.href = this.res\_urlList[3]['url'];

},

filterOnlyCapsText (value, search) {

return value != null &&

search != null &&

typeof value === 'string' &&

value.toString().toLocaleUpperCase().indexOf(search) !== -1

},

select: function(name){

this.selected\_sample = name;

this.show = false;

this.pre\_show = false;

// console.log(this.selected\_sample)

this.$message({

message: '已选择',

type: 'success'

});

},

handleSize

(val) {

// console.log(`每页 ${val} 条`);

this.currentPage = 1;

this.pageSize = val;

},

//当前页改变时触发 跳转其他页

handleCurrent

(val) {

// console.log(`当前页: ${val}`);

this.currentPage = val;

},

PreShow: function () {

let sample = this.selected\_sample

this.pre\_urlList = []

this.pre\_preView\_srcList = []

this.pro1 = true

this.$store.dispatch('getBrainPre',{sample}).then((res)=>{

console.log("preshow:",res)

res.forEach(item=>{

let area = {url:item[0],description:item[1]}

this.pre\_urlList.push(area)

this.pre\_preView\_srcList.push(item[0])

})

});

setTimeout(()=>{

this.pre\_show = true

this.pro1 = false

},50)

},

getResult: function(){

// console.log(this.selected\_sample)

let sample = this.selected\_sample

this.res\_urlList = []

this.res\_preView\_srcList = []

this.infoList = []

this.pro2 = true

this.$message({

message: '请求成功',

type: 'success'

});

this.$store.dispatch('getBrainResult',{sample}).then((res)=>{

// console.log('model res -------',res)

let pathList = res[0]

console.log("pathList: ",pathList)

let infoList = res[1]

console.log("infoList: ",infoList)

pathList.forEach(item=>{

let area = {url:item[0],description:item[1]}

this.res\_urlList.push(area)

this.res\_preView\_srcList.push(item[0])

})

infoList.forEach(item=>{

let area = {Data:item[0],Description:item[1]}

this.infoList.push(area)

})

});

setTimeout(()=>{

this.show = true

this.pro2 = false

},50)

}

}

}

</script>

<style scoped>

</style>

## elec\_home.vue

<template>

<div>

<v-col

cols="10"

sm="10"

>

<v-sheet

min-height="70vh"

rounded="lg"

>

<v-card-title>

第一步 数据选择

</v-card-title>

<v-divider></v-divider>

<div style="padding: 20px">

<!-- 表格 -->

<el-table :data="FileData.slice((currentPage-1)\*pageSize,currentPage\*pageSize)" style="width: 100%">

<el-table-column label="文件名" width="400">

<template slot-scope="scope">

<div slot="reference" class="name-wrapper">

<el-tag>{{ scope.row.name }}</el-tag>

</div>

</template>

</el-table-column>

<el-table-column label="操作">

<template slot-scope="scope">

<el-button

size="mini"

@click="select(scope.row.name)">选择</el-button>

<el-button

size="mini"

type="danger"

@click="handleDelete(scope.$index, scope.row)">删除</el-button>

</template>

</el-table-column>

</el-table>

<!-- 分页器 -->

<div class="block" style="margin-top:15px;">

<el-pagination align='center' @size-

="handleSize

" @current-

="handleCurrent

"

:current-page="currentPage"

:page-sizes="[5,10,15]"

:page-size="pageSize"

layout="total, sizes, prev, pager, next, jumper"

:total="FileData.length">

</el-pagination>

</div>

</div>

<v-divider></v-divider>

<div style="padding-bottom: 20px">

<v-card-title>

已选择的数据

</v-card-title>

</div>

<v-row>

<div v-if="this.selected\_sample" style="padding: 20px">

<el-tag

effect="light"

type="success">

{{this.selected\_sample}}

</el-tag>

<v-btn

class="ma-2"

color="secondary"

@click="PreShow()"

>

文件预览

</v-btn>

<div v-show="pro1" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

</v-row>

<div v-show="pre\_show" style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in pre\_urlList"

:key="item.url"

cols="10"

>

<el-image

:src="item.url"

:preview-src-list="pre\_preView\_srcList">

>

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

<v-divider></v-divider>

<v-card-title>

第二步 结果分析

</v-card-title>

<div style="padding: 20px">

<v-btn

class="ma-2"

color="primary"

dark

@click="getResult()"

>

查看分析结果

</v-btn>

<div v-show="pro2" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

<div v-show="show">

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList.slice(0,1)"

:key="item.url"

cols="10"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<!-- <v-card-text>-->

<!-- <div>{{item.description}}</div>-->

<!-- </v-card-text>-->

<el-button

type="success"

plain

icon="el-icon-setting"

size="mini"

@click="pluginDownload"

>下载报告及结果

</el-button>

</v-col>

</v-row>

</div>

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList.slice(1,2)"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

</div>

<!-- <div style="padding: 20px">-->

<!-- <v-card-text>{{this.info}}</v-card-text>-->

<!-- </div>-->

</div>

</v-sheet>

</v-col>

</div>

</template>

<script>

import param from "@/store/modules/param";

export default {

name: "elec\_home",

data:()=>({

FileData:[],

selected\_sample:'',

pre\_show:false,

button\_show:true,

show:false,

count:"",

time:null,

echo\_pre:null,

pic\_url:null,

detect\_url:null,

info:null,

pre\_urlList:[],

pre\_preView\_srcList:[],

res\_urlList:[],

res\_preView\_srcList:[],

pro1:false,

pro2:false,

currentPage: 1, // 当前页码

total: 0, // 总条数

pageSize: 5 // 每页的数据条数

}),

created() {

let content="elec"

this.$store.dispatch('get\_elec\_List', {content}).then((res)=>{

res.forEach(item=>{

let area={name:item}

this.FileData.push(area)

})

})

},

computed:{

preView(){

if (param.state.echo\_pre===null){

return null

}

return param.state.echo\_pre;

}

},

methods:{

pluginDownload(){

console.log(this.res\_urlList)

window.location.href = this.res\_urlList[2]['url'];

},

handleSize

(val) {

console.log(`每页 ${val} 条`);

this.currentPage = 1;

this.pageSize = val;

},

//当前页改变时触发 跳转其他页

handleCurrent

(val) {

console.log(`当前页: ${val}`);

this.currentPage = val;

},

select: function(name){

this.selected\_sample = name;

console.log(this.selected\_sample)

this.show = false;

this.info = ''

this.pre\_show = false;

this.$message({

message: '已选择',

type: 'success'

});

},

PreShow: function () {

let sample = this.selected\_sample

this.pre\_urlList = []

this.pre\_preView\_srcList = []

this.pro1 = true

this.$store.dispatch('getElecPre',{sample}).then((res)=>{

res.forEach(item=>{

let area = {url:item[0],description:item[1]}

this.pre\_urlList.push(area)

this.pre\_preView\_srcList.push(item[0])

})

});

setTimeout(()=>{

this.pre\_show = true

this.pro1 = false

},50)

},

getResult: function(){

console.log(this.selected\_sample)

let sample = this.selected\_sample

this.res\_urlList = []

this.res\_preView\_srcList = []

this.pro2 = true

this.$message({

message: '请求成功',

type: 'success'

});

this.$store.dispatch('getElecResult',{sample}).then((res)=>{

let item1 = res[0]

let item2 = res[1]

let item3 = res[2]

let area={url:item1[0],description:item1[1]}

this.res\_urlList.push(area)

this.res\_preView\_srcList.push(item1[0])

let area1={url:item2[0],description:item2[1]}

this.res\_urlList.push(area1)

this.res\_preView\_srcList.push(item2[0])

let area2={url:item3[0],description:item3[1]}

this.res\_urlList.push(area2)

this.res\_preView\_srcList.push(item3[0])

this.info = res[3]

});

setTimeout(()=>{

this.show = true

this.pro2 = false

},50)

}

},

}

</script>

<style scoped>

</style>

## eod\_home.vue

<template>

<div>

<v-col

cols="10"

sm="10"

>

<v-sheet

min-height="70vh"

rounded="lg"

>

<v-card-title>

第一步 病例选择

</v-card-title>

<v-divider></v-divider>

<div style="padding: 20px">

<!-- 表格 -->

<el-table :data="FileData.slice((currentPage-1)\*pageSize,currentPage\*pageSize)" style="width: 100%">

<el-table-column label="文件名" width="400">

<template slot-scope="scope">

<div slot="reference" class="name-wrapper">

<el-tag>{{ scope.row.name }}</el-tag>

</div>

</template>

</el-table-column>

<el-table-column label="操作">

<template slot-scope="scope">

<el-button

size="mini"

@click="select(scope.row.name)">选择</el-button>

<el-button

size="mini"

type="danger"

@click="handleDelete(scope.$index, scope.row)">删除</el-button>

</template>

</el-table-column>

</el-table>

<!-- 分页器 -->

<div class="block" style="margin-top:15px;">

<el-pagination align='center' @size-

="handleSize

" @current-

="handleCurrent

"

:current-page="currentPage"

:page-sizes="[5,10,15]"

:page-size="pageSize"

layout="total, sizes, prev, pager, next, jumper"

:total="FileData.length">

</el-pagination>

</div>

</div>

<v-divider></v-divider>

<div style="padding-bottom: 20px">

<v-card-title>

已选择的数据

</v-card-title>

</div>

<v-row>

<div v-if="this.selected\_sample" style="padding: 20px">

<el-tag

effect="light"

type="success">

{{this.selected\_sample}}

</el-tag>

<v-btn

class="ma-2"

color="secondary"

@click="PreShow()"

>

文件预览

</v-btn>

<div v-show="pro1" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

</v-row>

<div v-show="pre\_show" style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in pre\_urlList"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="pre\_preView\_srcList">

>

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

<v-divider></v-divider>

<v-card-title>

第二步 结果分析

</v-card-title>

<div style="padding: 20px">

<v-btn

class="ma-2"

color="primary"

dark

@click="getResult()"

>

查看分析结果

</v-btn>

<div v-show="pro2" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

<div v-show="show">

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

<div>

<v-data-table

:headers="headers"

:items="infoList"

item-key="name"

class="elevation-1"

:search="search"

:custom-filter="filterOnlyCapsText"

>

<template v-slot:top>

<v-text-field

v-model="search"

label="Search (UPPER CASE ONLY)"

class="mx-4"

></v-text-field>

</template>

</v-data-table>

</div>

</div>

</div>

</v-sheet>

</v-col>

</div>

</template>

<script>

export default {

name: "eod\_home",

data:()=>({

FileData:[],

selected\_sample:null,

pre\_show:false,

button\_show:true,

show:false,

count:"",

time:null,

echo\_pre:null,

pic\_url:null,

detect\_url:null,

info:null,

pre\_urlList:[],

pre\_preView\_srcList:[],

res\_urlList:[],

res\_preView\_srcList:[],

pro1:false,

pro2:false,

currentPage: 1, // 当前页码

total: 0, // 总条数

pageSize: 5, // 每页的数据条数

infoList:[],

search: '',

calories: '',

svc\_od\_List:[],

svc\_os\_List:[],

dvc\_od\_List:[],

dvc\_os\_List:[],

}),

created() {

let content="eod"

this.$store.dispatch('get\_EOD\_List', {content}).then((res)=>{

res.forEach(item=>{

let area={name:item}

this.FileData.push(area)

})

})

},

computed: {

headers () {

return [

{

text: 'Type',

align: 'start',

sortable: false,

value: 'type',

},

{

text: 'Description',

value: 'tag'

},

{

text: 'Data',

align: 'end',

value: 'data'

},

]

},

},

methods:{

filterOnlyCapsText (value, search) {

return value != null &&

search != null &&

typeof value === 'string' &&

value.toString().toLocaleUpperCase().indexOf(search) !== -1

},

select: function(name){

this.selected\_sample = name;

this.show = false;

this.pre\_show = false;

// console.log(this.selected\_sample)

this.$message({

message: '已选择',

type: 'success'

});

},

handleSize

(val) {

// console.log(`每页 ${val} 条`);

this.currentPage = 1;

this.pageSize = val;

},

//当前页改变时触发 跳转其他页

handleCurrent

(val) {

// console.log(`当前页: ${val}`);

this.currentPage = val;

},

PreShow: function () {

let sample = this.selected\_sample

this.pre\_urlList = []

this.pre\_preView\_srcList = []

this.pro1 = true

this.$store.dispatch('getEODPre', {sample}).then((res)=>{

this.pre\_urlList = []

res.forEach(item=>{

let area={url:item[0],description:item[1]}

this.pre\_urlList.push(area)

this.pre\_preView\_srcList.push(item[0])

})

})

setTimeout(()=>{

this.pre\_show = true

this.pro1 = false

},50)

},

getResult: function(){

// console.log(this.selected\_sample)

let sample = this.selected\_sample

this.res\_urlList = []

this.res\_preView\_srcList = []

this.infoList = []

this.pro2 = true

this.$message({

message: '请求成功',

type: 'success'

});

this.$store.dispatch('getEODResult',{sample}).then((res)=>{

let url\_arr = res[0]

let info\_res = res [1]

console.log("res:",res)

console.log('url\_arr:',url\_arr)

console.log('info\_res:',info\_res)

url\_arr.forEach(item=>{

let area={url:item[0],description:item[1]}

this.res\_urlList.push(area)

this.res\_preView\_srcList.push(item[0])

})

let arr1 = info\_res.svc\_od\_parameter

arr1.forEach(item=>{

let area={type:item[0],data:item[1],tag:item[2]}

this.svc\_od\_List.push(area)

})

let arr2 = info\_res.svc\_os\_parameter

arr2.forEach(item=>{

let area={type:item[0],data:item[1],tag:item[2]}

this.svc\_os\_List.push(area)

})

let arr3 = info\_res.dvc\_od\_parameter

arr3.forEach(item=>{

let area={type:item[0],data:item[1],tag:item[2]}

this.dvc\_od\_List.push(area)

})

let arr4 = info\_res.dvc\_os\_parameter

arr4.forEach(item=>{

let area={type:item[0],data:item[1],tag:item[2]}

this.dvc\_os\_List.push(area)

})

this.infoList = this.svc\_od\_List.concat(this.svc\_os\_List,this.dvc\_os\_List,this.dvc\_od\_List)

});

setTimeout(()=>{

this.show = true

this.pro2 = false

},50);

}

}

}

</script>

<style scoped>

</style>

## int\_art\_home.vue

<template>

<div>

<v-col

cols="10"

sm="10"

>

<v-sheet

min-height="70vh"

rounded="lg"

>

<v-card-title>

第一步 数据选择

<el-tooltip>

<div slot="content" style="width: 550px; " >

<h2>颅内动脉瘤检测系统</h2>

<br />

<h3 >说明：</h3>

<p style="text-indent:2em;">

颅内动脉瘤多为发生在颅内动脉管壁上的异常膨出，

是造成蛛网膜下腔出血的首位病因。该系统旨在通过智能分析算法对全脑血管造影（DSA）影像中存在的颅内动脉瘤进行自动化检测。

进一步提取出颅内动脉瘤的瘤体参数并定位出动脉瘤所在的颅内动脉段位置，辅助医生快速发现动脉瘤，提高诊断的准确率。

</p>

</div>

<span><i class="el-icon-question" /></span>

</el-tooltip>

</v-card-title>

<v-divider></v-divider>

<div style="padding: 20px">

<!-- 表格 -->

<el-table :data="FileData.slice((currentPage-1)\*pageSize,currentPage\*pageSize)" style="width: 100%">

<el-table-column label="文件名" width="400">

<template slot-scope="scope">

<div slot="reference" class="name-wrapper">

<el-tag>{{ scope.row.name }}</el-tag>

</div>

</template>

</el-table-column>

<el-table-column label="操作">

<template slot-scope="scope">

<el-button

size="mini"

@click="select(scope.row.name)">选择</el-button>

<el-button

size="mini"

type="danger"

@click="handleDelete(scope.$index, scope.row)">删除</el-button>

</template>

</el-table-column>

</el-table>

<!-- 分页器 -->

<div class="block" style="margin-top:15px;">

<el-pagination align='center' @size-

="handleSize

" @current-

="handleCurrent

"

:current-page="currentPage"

:page-sizes="[5,10,15]"

:page-size="pageSize"

layout="total, sizes, prev, pager, next, jumper"

:total="FileData.length">

</el-pagination>

</div>

</div>

<v-divider></v-divider>

<div style="padding-bottom: 20px">

<v-card-title>

已选择的数据

</v-card-title>

</div>

<v-row>

<div v-if="this.selected\_sample" style="padding: 20px">

<el-tag

effect="light"

type="success">

{{this.selected\_sample}}

</el-tag>

<v-btn

class="ma-2"

color="secondary"

@click="PreShow()"

>

文件预览

</v-btn>

<div v-show="pro1" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

</v-row>

<div v-show="pre\_show" style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in pre\_urlList"

:key="item.url"

cols="10"

>

<el-image

:src="item.url"

:preview-src-list="pre\_preView\_srcList">

>

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

<v-divider></v-divider>

<v-card-title>

第二步 结果分析

</v-card-title>

<div style="padding: 20px">

<v-btn

class="ma-2"

color="primary"

dark

@click="getResult()"

>

查看分析结果

</v-btn>

<div v-show="pro2" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

<div v-show="show">

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList.slice(0,1)"

:key="item.url"

cols="10"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<!-- <v-card-text>-->

<!-- <div>{{item.description}}</div>-->

<!-- </v-card-text>-->

<el-button

type="success"

plain

icon="el-icon-setting"

size="mini"

@click="pluginDownload"

>下载报告及结果

</el-button>

</v-col>

</v-row>

</div>

</div>

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList.slice(1,2)"

:key="item.url"

cols="5"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

<!-- <div style="padding: 20px">-->

<!-- <div v-for="sig in info"-->

<!-- :key="sig"-->

<!-- >-->

<!-- <v-card-text>{{sig}}</v-card-text>-->

<!-- </div>-->

<!-- </div>-->

</div>

</v-sheet>

</v-col>

</div>

</template>

<script>

import param from "@/store/modules/param";

export default {

name: "int\_art\_home",

data:()=>({

FileData:[],

selected\_sample:'',

pre\_show:false,

button\_show:true,

show:false,

count:"",

time:null,

echo\_pre:null,

pic\_url:null,

detect\_url:null,

info:null,

pro1:false,

pro2:false,

pre\_urlList:[],

pre\_preView\_srcList:[],

res\_urlList:[],

res\_preView\_srcList:[],

currentPage: 1, // 当前页码

total: 0, // 总条数

pageSize: 5 // 每页的数据条数

}),

created() {

let content="int\_art"

this.$store.dispatch('get\_int\_art\_List', {content}).then((res)=>{

res.forEach(item=>{

let area={name:item}

this.FileData.push(area)

})

})

},

computed:{

preView(){

if (param.state.echo\_pre===null){

return null

}

return param.state.echo\_pre;

}

},

methods:{

pluginDownload(){

console.log(this.res\_urlList)

window.location.href = this.res\_urlList[2]['url'];

},

handleSize

(val) {

console.log(`每页 ${val} 条`);

this.currentPage = 1;

this.pageSize = val;

},

//当前页改变时触发 跳转其他页

handleCurrent

(val) {

console.log(`当前页: ${val}`);

this.currentPage = val;

},

select: function(name){

this.selected\_sample = name;

this.show = false;

this.pre\_show = false;

console.log(this.selected\_sample)

this.$message('已选择');

},

PreShow: function () {

let sample = this.selected\_sample

this.pre\_urlList = []

this.pre\_preView\_srcList = []

this.$store.dispatch('getIntArtPre',{sample}).then((res)=>{

res.forEach(item=>{

let area = {url:item[0],description:item[1]}

this.pre\_urlList.push(area)

this.pre\_preView\_srcList.push(item[0])

})

});

setTimeout(()=>{

this.pre\_show = true

this.pro1 = false

},50)

},

getResult: function(){

console.log(this.selected\_sample)

let sample = this.selected\_sample

this.res\_urlList = []

this.res\_preView\_srcList = []

this.pro2 = true

this.$message({

message: '请求成功',

type: 'success'

});

this.$store.dispatch('getIntArtResult',{sample}).then((res)=>{

let item1 = res[0]

let item2 = res[1]

let item3 = res[2]

let area={url:item1[0],description:item1[1]}

this.res\_urlList.push(area)

this.res\_preView\_srcList.push(item1[0])

let area1={url:item2[0],description:item2[1]}

this.res\_urlList.push(area1)

this.res\_preView\_srcList.push(item2[0])

let area2={url:item3[0],description:item3[1]}

this.res\_urlList.push(area2)

this.res\_preView\_srcList.push(item3[0])

this.info = res[3]

});

setTimeout(()=>{

this.show = true

this.pro2 = false

},50)

}

}

}

</script>

<style scoped>

</style>

## brain\_home.vue

<template>

<div>

<v-col

cols="10"

sm="10"

>

<v-sheet

min-height="70vh"

rounded="lg"

>

<v-card-title>

第一步 数据选择

</v-card-title>

<v-divider></v-divider>

<div style="padding: 20px">

<!-- 表格 -->

<el-table :data="FileData.slice((currentPage-1)\*pageSize,currentPage\*pageSize)" style="width: 100%">

<el-table-column label="文件名" width="400">

<template slot-scope="scope">

<div slot="reference" class="name-wrapper">

<el-tag>{{ scope.row.name }}</el-tag>

</div>

</template>

</el-table-column>

<el-table-column label="操作">

<template slot-scope="scope">

<el-button

size="mini"

@click="select(scope.row.name)">选择</el-button>

<el-button

size="mini"

type="danger"

@click="handleDelete(scope.$index, scope.row)">删除</el-button>

</template>

</el-table-column>

</el-table>

<!-- 分页器 -->

<div class="block" style="margin-top:15px;">

<el-pagination align='center' @size-

="handleSize

" @current-

="handleCurrent

"

:current-page="currentPage"

:page-sizes="[5,10,15]"

:page-size="pageSize"

layout="total, sizes, prev, pager, next, jumper"

:total="FileData.length">

</el-pagination>

</div>

</div>

<v-divider></v-divider>

<div style="padding-bottom: 20px">

<v-card-title>

已选择的数据

</v-card-title>

</div>

<v-row>

<div v-if="this.selected\_sample" style="padding: 20px">

<el-tag

effect="light"

type="success">

{{this.selected\_sample}}

</el-tag>

<v-btn

class="ma-2"

color="secondary"

@click="PreShow()"

>

文件预览

</v-btn>

<div v-show="pro1" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

</v-row>

<div v-show="pre\_show" style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in pre\_urlList"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="pre\_preView\_srcList">

>

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

</div>

<v-divider></v-divider>

<v-card-title>

第二步 结果分析

</v-card-title>

<div style="padding: 20px">

<v-btn

class="ma-2"

color="primary"

dark

@click="getResult()"

>

查看分析结果

</v-btn>

<div v-show="pro2" style="float:right;padding-right: 10px">

<v-progress-circular

indeterminate

color="primary"

></v-progress-circular>

</div>

</div>

<div v-show="show">

<div style="padding: 20px">

<div>

<v-row>

<v-col

v-for="item in res\_urlList.slice(0,3)"

:key="item.url"

cols="3"

>

<el-image

:src="item.url"

:preview-src-list="res\_preView\_srcList">

<div slot="error" class="image-slot">

<i class="el-icon-picture-outline"></i>

</div>

</el-image>

<v-card-text>

<div>{{item.description}}</div>

</v-card-text>

</v-col>

</v-row>

</div>

<el-button

type="success"

plain

icon="el-icon-setting"

size="mini"

@click="pluginDownload"

>下载完整参数及结果

</el-button>

</div>

<div>

<v-data-table

:headers="headers"

:items="infoList"

item-key="name"

class="elevation-1"

:search="search"

:custom-filter="filterOnlyCapsText"

>

<template v-slot:top>

<v-text-field

v-model="search"

label="Search (UPPER CASE ONLY)"

class="mx-4"

></v-text-field>

</template>

</v-data-table>

</div>

</div>

</v-sheet>

</v-col>

</div>

</template>

<script>

export default {

name: "brain\_home",

data:()=>({

FileData:[],

selected\_sample:'',

pre\_show:false,

button\_show:true,

show:false,

count:"",

time:null,

brain\_pre:null,

pic\_url:null,

partion\_url:null,

up\_url:null,

pdf\_url:null,

para:null,

pre\_urlList:[],

pre\_preView\_srcList:[],

res\_urlList:[],

res\_preView\_srcList:[],

pro1:false,

pro2:false,

currentPage: 1, // 当前页码

total: 0, // 总条数

pageSize: 5, // 每页的数据条数

infoList:[],

search: '',

calories: '',

}),

created() {

let content="brain"

this.$store.dispatch('getFileList', {content}).then((res)=>{

res.forEach(item=>{

let area={name:item}

this.FileData.push(area)

})

})

},

computed: {

headers () {

return [

{

text: 'Description',

align: 'start',

sortable: false,

value: 'Description',

},

{

text: 'Data',

align: 'end',

value: 'Data'

},

]

},

},

methods:{

pluginDownload(){

console.log(this.res\_urlList)

window.location.href = this.res\_urlList[3]['url'];

},

filterOnlyCapsText (value, search) {

return value != null &&

search != null &&

typeof value === 'string' &&

value.toString().toLocaleUpperCase().indexOf(search) !== -1

},

select: function(name){

this.selected\_sample = name;

this.show = false;

this.pre\_show = false;

// console.log(this.selected\_sample)

this.$message({

message: '已选择',

type: 'success'

});

},

handleSize

(val) {

// console.log(`每页 ${val} 条`);

this.currentPage = 1;

this.pageSize = val;

},

//当前页改变时触发 跳转其他页

handleCurrent

(val) {

// console.log(`当前页: ${val}`);

this.currentPage = val;

},

PreShow: function () {

let sample = this.selected\_sample

this.pre\_urlList = []

this.pre\_preView\_srcList = []

this.pro1 = true

this.$store.dispatch('getBrainPre',{sample}).then((res)=>{

console.log("preshow:",res)

res.forEach(item=>{

let area = {url:item[0],description:item[1]}

this.pre\_urlList.push(area)

this.pre\_preView\_srcList.push(item[0])

})

});

setTimeout(()=>{

this.pre\_show = true

this.pro1 = false

},50)

},

getResult: function(){

// console.log(this.selected\_sample)

let sample = this.selected\_sample

this.res\_urlList = []

this.res\_preView\_srcList = []

this.infoList = []

this.pro2 = true

this.$message({

message: '请求成功',

type: 'success'

});

this.$store.dispatch('getBrainResult',{sample}).then((res)=>{

// console.log('model res -------',res)

let pathList = res[0]

console.log("pathList: ",pathList)

let infoList = res[1]

console.log("infoList: ",infoList)

pathList.forEach(item=>{

let area = {url:item[0],description:item[1]}

this.res\_urlList.push(area)

this.res\_preView\_srcList.push(item[0])

})

infoList.forEach(item=>{

let area = {Data:item[0],Description:item[1]}

this.infoList.push(area)

})

});

setTimeout(()=>{

this.show = true

this.pro2 = false

},50)

}

}

}

</script>

<style scoped>

</style>

## About.vue

<template>

<div>

<div style="padding: 10px">

<v-card

class="mx-auto"

>

<div style="padding: 10px 20px;">

<v-img

src="https://images.pexels.com/photos/5677459/pexels-photo-5677459.jpeg?auto=compress&cs=tinysrgb&dpr=2&h=750&w=1260"

height="200px"

></v-img>

</div>

<v-card-title>

Top western road trips

</v-card-title>

<v-card-subtitle>

1,000 miles of wonder

</v-card-subtitle>

<v-card-actions>

<v-btn

color="orange lighten-2"

text

>

Explore

</v-btn>

<v-spacer></v-spacer>

<v-btn

icon

@click="show = !show"

>

<v-icon>{{ show ? 'mdi-chevron-up' : 'mdi-chevron-down' }}</v-icon>

</v-btn>

</v-card-actions>

<v-expand-transition>

<div v-show="show">

<v-divider></v-divider>

<v-card-text>

I'm a thing. But, like most politicians, he promised more than he could deliver. You won't have time for sleeping, soldier, not with all the bed making you'll be doing. Then we'll go with that data file! Hey, you add a one and two zeros to that or we walk! You're going to do his laundry? I've got to find a way to escape.

</v-card-text>

</div>

</v-expand-transition>

</v-card>

</div>

</div>

</template>

<script>

export default {

name: "About",

data:()=>({

show:false

})

}

</script>

<style scoped>

</style>

## Carousel.vue

<template>

<div>

<v-card

class="mx-auto"

>

<div style="padding: 10px">

<v-carousel hide-delimiters height="60vh" width="60vh">

<v-carousel-item

v-for="(item,index) in carList"

:key="index"

:src="item.src"

>

<v-container

fill-height

fluid

pa-0 ma-0 pb-3

>

<v-layout fill-height align-end pb-4 mb-4>

<v-flex xs12>

<span class=" white--text" v-text="item.src"> </span>

</v-flex>

</v-layout>

</v-container>

</v-carousel-item>

</v-carousel>

</div>

</v-card>

</div>

</template>

<script>

export default {

name: "Carousel",

data:()=>({

path:'',

carList:[],

}),

created() {

// let path1="D:\\ZhiHu\_Store\\SUSTech\_Course\\MarsFindings\\Platform\\front\_end\\static\\pre\\OD"

// this.$store.dispatch('getFileList', {path1}).then((res)=>{

// res.forEach(item=>{

// let area={src:require('../../../../static/pre/OD/' + item)}

// console.log("car !")

// this.carList.push(area)

// })

// })

}

}

</script>

<style scoped>

</style>

## analyze.vue

<template>

<div>

<div style="padding: 10px">

<v-card

class="mx-auto"

flat

>

<!-- <div v-if="EX" style="padding: 10px ;">-->

<!-- <v-img-->

<!-- :src="EX"-->

<!-- alt="expath"-->

<!-- ></v-img>-->

<!-- </div>-->

<div v-if="CUR" style="padding: 10px ;">

<p>Curvature</p>

<v-img

:src="CUR"

alt="cur\_path"

width="200px"

height="200px"

></v-img>

</div>

<div v-if="CEN" style="padding: 10px ;">

<p>Vessel Center-line</p>

<v-img

:src="CEN"

alt="center\_path"

width="200px"

height="200px"

></v-img>

</div>

<v-card-title>

Analyze

</v-card-title>

<v-card-subtitle>

<p class="subtitle-2">Length:{{getLength}}</p>

<p class="subtitle-2">Dim:{{getDim}}</p>

<p class="subtitle-2">Vessel Area:{{getVesselsArea}}</p>

<p class="subtitle-2">Explant Area:{{getExplantArea}}</p>

<p class="subtitle-2">Vessel percentage Area:{{getVPArea}}</p>

</v-card-subtitle>

</v-card>

</div>

</div>

</template>

<script>

import { mapGetters } from "vuex";

import param from "@/store/modules/param";

export default {

name: "analyze",

data:()=>({

show:false,

ex\_path:null,

cur\_path:null,

center\_path:null,

}),

computed:{

...mapGetters(['getLength']),

...mapGetters(['getDim']),

EX(){

if (param.state.extract\_path===null){

return null

}

return require('../../../../static/result/' + param.state.extract\_path)

},

CUR(){

if (param.state.cur\_path===null){

return null

}

return require('../../../../static/result/' + param.state.cur\_path)

},

CEN(){

if (param.state.centerLine\_path===null){

return null

}

return require('../../../../static/result/' + param.state.centerLine\_path)

}

},

}

</script>

<style scoped>

</style>

## file.js

import {

get\_int\_art\_FileList, getFileList, revieveFile, get\_EOD\_FileList, get\_elec\_FileList

} from "@/api/file";

const file={

state:{

fileList:[],

imageList:[],

imageSrc:'',

filename:'result/T01.png',

int\_art\_List:null,

eod\_List:null,

elec\_List:null

},

getters:{

FileList(state){

return state.fileList

},

getFileName:state=>state.filename,

},

mutations:{

FileList(state,fileList){

// console.log("info:",file)

state.fileList=fileList

},

saveImage(state, { src }){

state.imageSrc=src

},

Image(state,file){

console.log("info:",file)

state.filename="pre/OD/"+file

},

Image2(state,file){

console.log("info:",file)

state.filename="result/"+file

}

},

actions:{

saveImage( {commit} , { item } ){

commit('saveImage',{ item })

},

getFileList(context,path){

return new Promise((resolve, reject)=>{

getFileList(path).then(res=>{

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

sendImage(context,data){

return revieveFile(data);

},

//颅内动脉瘤部分

get\_int\_art\_List(context,path){

return new Promise((resolve, reject)=>{

get\_int\_art\_FileList(path).then(res=>{

console.log("List:",res.data)

file.state.int\_art\_List=res.data

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

//EOD部分

get\_EOD\_List(context,path){

return new Promise((resolve, reject)=>{

get\_EOD\_FileList(path).then(res=>{

// console.log("List:",res.data)

file.state.eod\_List=res.data

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

get\_elec\_List(context,path){

return new Promise((resolve, reject)=>{

get\_elec\_FileList(path).then(res=>{

console.log("List:",res.data)

file.state.elec\_List=res.data

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

}

}

export default file;

## param.js

import {

getCurvature, getModelRes , getEchoRes, getEchoPre, getIntArtPre,

getIntArtRes, get\_eod\_pre, getEODRes, getbrainPre,getbrainRes,getElecPre,getElecRes

} from "@/api/param"

const param={

state:{

vesselArea:[],

length:null,

dimension:null,

echo\_pre:null,

echo\_path:null,

brain\_pre:null,

brain\_path:null,

elec\_pre:null,

elec\_path:null

},

getters:{

getLength:state=>state.length,

getDim:state=>state.dimension,

},

mutations:{

Length(state,value){

state.length=value

},

//心超系统

},

actions:{

getModelResult(context,path){

return new Promise((resolve, reject)=>{

getModelRes(path).then(res=>{

console.log("model result:",res.data)

param.state.extract\_path=res.data

console.log("model result:",param.state.extract\_path)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getCurva(context,path){

return new Promise((resolve, reject)=>{

getCurvature(path).then(res=>{

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

//心超系统部分

getEcoPre(context,path){

return new Promise((resolve, reject)=>{

getEchoPre(path).then(res=>{

console.log("pre result:",res.data)

param.state.echo\_pre=res.data

console.log("model result:",param.state.extract\_path)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getEcoResult(context,path){

return new Promise((resolve, reject)=>{

getEchoRes(path).then(res=>{

console.log("model result:",res.data)

param.state.echo\_path=res.data

console.log("model result:",param.state.echo\_path)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

//颅内动脉部分

getIntArtPre(context,path){

return new Promise((resolve, reject)=>{

getIntArtPre(path).then(res=>{

console.log("pre result:",res.data)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getIntArtResult(context,path){

return new Promise((resolve, reject)=>{

getIntArtRes(path).then(res=>{

console.log("model result:",res.data)

param.state.echo\_path=res.data

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

//EOD部分

getEODPre(context,path){

return new Promise((resolve, reject)=>{

get\_eod\_pre(path).then(res=>{

// console.log("pre result:",res.data)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getEODResult(context,path){

return new Promise((resolve, reject)=>{

getEODRes(path).then(res=>{

// console.log("model result:",res.data)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getBrainPre(context,path){

return new Promise((resolve, reject)=>{

getbrainPre(path).then(res=>{

console.log("pre result:",res.data)

param.state.brain\_pre=res.data

console.log("model result:",param.state.extract\_path)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getBrainResult(context,path){

return new Promise((resolve, reject)=>{

getbrainRes(path).then(res=>{

console.log("model result:",res.data)

param.state.brain\_path=res.data

console.log("model result:",param.state.brain\_path)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

//眼脑电部分

getElecPre(context,path){

return new Promise((resolve, reject)=>{

getElecPre(path).then(res=>{

console.log("pre result:",res.data)

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

getElecResult(context,path){

return new Promise((resolve, reject)=>{

getElecRes(path).then(res=>{

console.log("model result:",res.data)

param.state.elec\_path=res.data

resolve(res.data);

}).catch(err=>{

reject(err);

});

});

},

}

}

export default param

# 后端代码部分

## app.py

from flask import Flask,request,json

from flask import jsonify

import utils

import read\_eod

import os

app = Flask(\_\_name\_\_)

# head\_url = r"http://124.223.91.66:5100"

head\_url = r"http://127.0.0.1:5000"

@app.route('/')

def hello\_world(): # put application's code here

return 'Hello World!'

@app.route('/file',methods=['GET', 'POST'])

def file(): # put application's code here

if request.method=='GET':

path=request.args.get('path')

print("path:",path)

path\_dict = json.loads(path)

content = path\_dict['content']

print(content)

# utils.Info\_print(user\_dict['path'])

content=utils.forder\_reader(content)

return jsonify(content)

return 'Hello World!'

@app.route('/param/model',methods=['GET', 'POST'])

def model(): # put application's code here

if request.method == 'GET':

print("start")

path = request.args.get('path')

print("path:", path)

filename =path

print("filename:",filename)

new\_name='ex'+filename

return jsonify(new\_name)

'''

-------------------------心超-----------------------------

'''

@app.route('/param/echo/pre',methods=['GET', 'POST'])

def echo\_pre(): # put application's code here

if request.method == 'GET':

print("start")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

print("filename:",filename)

ext = os.path.splitext(filename)

new\_name= 'raw\_' + ext[0] + '.gif'

print("new\_name:", new\_name)

file\_url = os.path.join(head\_url,'static/echo/output',new\_name)

buffer = []

buffer.append([file\_url,ext[0]])

return jsonify(buffer)

@app.route('/param/echo',methods=['GET', 'POST'])

def echo\_model(): # put application's code here

if request.method == 'GET':

print("start")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

ext = os.path.splitext(filename)

print("filename:",filename)

file\_url = os.path.join(head\_url,'static/echo/output/')

buffer = []

buffer.append([file\_url+'report\_'+ext[0]+'.png','诊断报告'])

buffer.append([file\_url+'en\_'+ext[0]+'.gif','x4超分结果'])

buffer.append([file\_url+'out\_'+ext[0]+'.gif','分析结果'])

buffer.append([file\_url+'fig\_'+ext[0]+'.png','EF变化趋势'])

buffer.append([file\_url+'res\_'+ext[0]+'.zip','结果打包'])

return jsonify(buffer)

'''

-------------------------颅内动脉-----------------------------

'''

@app.route('/file/int\_art',methods=['GET', 'POST'])

def int\_art\_file(): # put application's code here

if request.method=='GET':

# print('nnnnnnnnnnnnnn')

path=request.args.get('path')

print("path:",path)

path\_dict = json.loads(path)

content = path\_dict['content']

print(content)

path = os.path.join('static',content,'data')

pathList = os.listdir(path)

newList = []

for item in pathList:

ext = os.path.splitext(item)

if ext[1] != '.png' or ext[0].endswith('output') or ext[0].endswith('report'):

continue

newList.append(item)

return jsonify(newList)

return 'Hello World!'

@app.route('/param/int\_art/pre',methods=['GET', 'POST'])

def int\_art\_pre(): # put application's code here

if request.method == 'GET':

print("nnnnnnnnnn")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

ext = os.path.splitext(filename)

print("filename:",filename)

file\_url = os.path.join(head\_url,"static/int\_art/data",filename)

buffer = []

buffer.append([file\_url,ext[0]])

return jsonify(buffer)

@app.route('/param/int\_art',methods=['GET', 'POST'])

def Int\_Art\_model(): # put application's code here

if request.method == 'GET':

print("ddddddddd")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

print("filename:", filename)

ext = os.path.splitext(filename)

file\_url = os.path.join(head\_url, "static/int\_art/data/")

buffer = []

txt\_info = utils.readIntArt\_txt(ext[0]+'.txt')

buffer.append([file\_url+ext[0]+'\_report'+'.png',ext[0]+'辅助检测报告'])

buffer.append([file\_url+ext[0]+'\_output'+'.png',ext[0]+'颅内动脉瘤定位结果'])

buffer.append([file\_url+ext[0]+'.zip',ext[0]+'结果打包'])

buffer.append(txt\_info)

return jsonify(buffer)

'''

-------------------------EOD-----------------------------

'''

@app.route('/file/eod',methods=['GET', 'POST'])

def eod\_file(): # put application's code here

if request.method=='GET':

# print('nnnnnnnnnnnnnn')

path=request.args.get('path')

print("path:",path)

path\_dict = json.loads(path)

content = path\_dict['content']

print(content)

patient\_name\_list, eod\_feature\_file = read\_eod.eod\_read\_patient\_list()

return jsonify(patient\_name\_list)

return 'Hello World!'

@app.route('/param/eod/pre',methods=['GET', 'POST'])

def eod\_pre(): # put application's code here

if request.method == 'GET':

print("nnnnnnnnnn")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

print("filename:",filename)

coList = ['OD\_SVC','OD\_DVC','OS\_SVC','OS\_DVC']

patient\_name\_list, eod\_feature\_file = read\_eod.eod\_read\_patient\_list()

image\_path, patient\_index\_id = read\_eod.return\_eod\_image\_path(filename, patient\_name\_list)

pre\_list = []

for item,info in zip(image\_path.values(),coList):

item = os.path.join(head\_url,item)

pre\_list.append([item,info])

print(item)

return jsonify(pre\_list)

@app.route('/param/eod',methods=['GET', 'POST'])

def eod\_model(): # put application's code here

if request.method == 'GET':

print("nnnnnnnnnn")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

print("filename:", filename)

patient\_name\_list, eod\_feature\_file = read\_eod.eod\_read\_patient\_list()

image\_path, patient\_index\_id = read\_eod.return\_eod\_image\_path(filename, patient\_name\_list)

eod\_result\_path, eod\_parameter\_result = read\_eod.return\_eod\_result\_path(patient\_index\_id, eod\_feature\_file)

res\_list = []

ans = []

headList = ['svc\_vessel\_segmentation\_od','dvc\_vessel\_segmentation\_od','svc\_vessel\_segmentation\_os',

'dvc\_vessel\_segmentation\_os','svc\_bifurcation\_od','dvc\_bifurcation\_od','svc\_bifurcation\_os',

'dvc\_bifurcation\_os','svc\_vessel\_direction\_od','dvc\_vessel\_direction\_od','svc\_vessel\_direction\_os',

'dvc\_vessel\_direction\_os','svc\_faz\_od','dvc\_faz\_od','svc\_faz\_os','dvc\_faz\_os']

for item,info in zip(eod\_result\_path.values(),headList):

item = os.path.join(head\_url, item)

res\_list.append([item,info])

print(item)

ans.append(res\_list)

tmp = eod\_parameter\_result

coList = ['ID', 'Name', '类别', '眼别', '层次', 'FAZ面积比', 'FAZ圆度1', 'FAZ轴比', 'FAZ圆度2', 'FAZ坚固度', '血流方向比例', '血流方向面积', '血管弯曲度', '血管面积密度',

'血管长度密度',

'血管分形维数', '血管交叉点数']

# 'Area\_VAD', 'Area\_VAD', 'Area\_VAD', 'Area\_VAD', 'Area\_VAD', 'Area\_VLD', 'Area\_VLD',

# 'Area\_VLD',

# 'Area\_VLD', 'Area\_VLD', 'Area\_Boxdim', 'Area\_Boxdim', 'Area\_Boxdim', 'Area\_Boxdim', 'Area\_Boxdim',

# 'Area\_Tortuosity', 'Area\_Tortuosity', 'Area\_Tortuosity', 'Area\_Tortuosity', 'Area\_Tortuosity',

# 'Area\_B\_num', 'Area\_B\_num', 'Area\_B\_num', 'Area\_B\_num', 'Area\_B\_num']

for key, values in zip(tmp.keys(), tmp.values()):

data\_arr = values.tolist()

elements = []

for data, info in zip(data\_arr,coList):

elements.append([key,data,info])

tmp[key] = elements

ans.append(tmp)

return jsonify(ans)

'''

-------------------------眼脑-----------------------------

'''

@app.route('/param/brain/pre',methods=['GET', 'POST'])

def brain\_pre(): # put application's code here

if request.method == 'GET':

print("start")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

print("filename:",filename)

ext = filename.split('.')

new\_name= ext[0] + '\_正视图.gif'

print("new\_name:", new\_name)

buffer = []

file\_url = head\_url + "/static/brain/output/"

# buffer.append([file\_url,ext[0]+'\_正视图'])

buffer.append([file\_url + ext[0] + '\_正视图.gif', ext[0] + '\_正视图'])

buffer.append([file\_url + ext[0] + '\_左视图.gif',ext[0] + '\_左视图'])

buffer.append([file\_url + ext[0] + '\_俯视图.gif',ext[0] + '\_俯视图'])

return jsonify(buffer)

@app.route('/param/brain',methods=['GET', 'POST'])

def brain\_model(): # put application's code here

if request.method == 'GET':

print("start")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

#ext = os.path.splitext(filename)

ext = filename.split('.')

print("filename:",filename)

file\_url = head\_url + "/static/brain/output/"

buffer = []

buffer.append([file\_url + ext[0] + '\_aseg\_正视图.gif', ext[0] + '\_aseg\_正视图'])

buffer.append([file\_url + ext[0] + '\_aseg\_左视图.gif',ext[0] + '\_aseg\_左视图'])

buffer.append([file\_url + ext[0] + '\_aseg\_俯视图.gif',ext[0] + '\_aseg\_俯视图'])

buffer.append([file\_url + ext[0] + '\_aseg\_res.zip',ext[0] + '参数提取结果'])

info\_list = utils.readBrain\_csv(ext[0]+'\_aseg\_参数.csv')

ans\_list = []

ans\_list.append(buffer)

ans\_list.append(info\_list)

return jsonify(ans\_list)

'''

-------------------------眼脑电-----------------------------

'''

@app.route('/file/elec',methods=['GET', 'POST'])

def elec\_file(): # put application's code here

if request.method=='GET':

# print('nnnnnnnnnnnnnn')

path=request.args.get('path')

print("path:",path)

path\_dict = json.loads(path)

content = path\_dict['content']

print(content)

path = os.path.join('static',content,'data')

pathList = os.listdir(path)

newList = []

for item in pathList:

ext = os.path.splitext(item)

if ext[1] != '.mp4' or ext[0].endswith('\_heat\_map'):

continue

newList.append(item)

return jsonify(newList)

return 'Hello World!'

@app.route('/param/elec/pre',methods=['GET', 'POST'])

def elec\_pre(): # put application's code here

if request.method == 'GET':

print("nnnnnnnnnn")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

new\_name = os.path.splitext(filename)[0]+".gif"

print("filename:",new\_name)

file\_url = head\_url+"/static/elec/data/"+new\_name

buffer = []

buffer.append([file\_url,os.path.splitext(filename)[0]])

return jsonify(buffer)

@app.route('/param/elec',methods=['GET', 'POST'])

def elec\_model(): # put application's code here

if request.method == 'GET':

print("ddddddddd")

path = request.args.get('path')

print("path:", path)

path\_dict = json.loads(path)

filename = path\_dict['sample']

print("filename:", filename)

ext = os.path.splitext(filename)

file\_url = head\_url+"/static/elec/data/"

buffer = []

txt\_info = utils.readElec\_txt(ext[0]+'.txt')

buffer.append([file\_url+ext[0]+'\_report'+'.png',ext[0]+'辅助诊断报告'])

buffer.append([file\_url+ext[0]+'\_heat\_map'+'.png',ext[0]+'热值图'])

buffer.append([file\_url+ext[0]+'.zip',ext[0]+'结果打包'])

buffer.append(txt\_info)

print(buffer)

return jsonify(buffer)

## read\_eod.py

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True, host='127.0.0.1', port=5000)

# app.run(debug=True, host='0.0.0.0', port=5100)

import cv2

import glob

import os

import pandas as pd

import shutil

eod\_root\_path = 'static/eod'

# 返回患者ID及姓名

def eod\_read\_patient\_list():

\_eod\_feature\_file\_ = pd.read\_excel("./static/eod/ETDRS\_model1.xlsx", sheet\_name='OCTA')

patient\_name\_list\_ = pd.DataFrame(\_eod\_feature\_file\_, columns=['ID', 'Name'])

patient\_name\_list\_ = [str(i[0]) for i in patient\_name\_list\_.values]

patient\_name\_list\_ = list(set(patient\_name\_list\_))

return patient\_name\_list\_, \_eod\_feature\_file\_

# 点击患者ID后返回SVC及DVC图像路径

def return\_eod\_image\_path(item, patient\_name\_list\_):

index = patient\_name\_list\_.index(item)

# print(index)

patient\_index\_id = patient\_name\_list\_[index]#.split('\_')[0]

image\_path = {

'svc\_image\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id + '\_OD\_SVC.png'),

'dvc\_image\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id + '\_OD\_DVC.png'),

'svc\_image\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id + '\_OS\_SVC.png'),

'dvc\_image\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id + '\_OS\_DVC.png'),

}

return image\_path, patient\_index\_id

# 点击分析返回患者结果

def return\_eod\_result\_path(patient\_index\_id\_, eod\_feature\_file\_):

eod\_result\_path = {

# 血管分割结果

'svc\_vessel\_segmentation\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_SVC\_vessel\_segmentation.jpg'),

'dvc\_vessel\_segmentation\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_DVC\_vessel\_segmentation.jpg'),

'svc\_vessel\_segmentation\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_SVC\_vessel\_segmentation.jpg'),

'dvc\_vessel\_segmentation\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_DVC\_vessel\_segmentation.jpg'),

# 血管交叉点

'svc\_bifurcation\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_SVC\_Bifurcation.jpg'),

'dvc\_bifurcation\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_DVC\_Bifurcation.jpg'),

'svc\_bifurcation\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_SVC\_Bifurcation.jpg'),

'dvc\_bifurcation\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_DVC\_Bifurcation.jpg'),

# 血流方向图

'svc\_vessel\_direction\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_SVC\_vessel\_direction.jpg'),

'dvc\_vessel\_direction\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_DVC\_vessel\_direction.jpg'),

'svc\_vessel\_direction\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_SVC\_vessel\_direction.jpg'),

'dvc\_vessel\_direction\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_DVC\_vessel\_direction.jpg'),

# FAZ分割结果

'svc\_faz\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_SVC\_FAZ.jpg'),

'dvc\_faz\_od\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OD\_DVC\_FAZ.jpg'),

'svc\_faz\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_SVC\_FAZ.jpg'),

'dvc\_faz\_os\_path': os.path.join(eod\_root\_path, 'eod\_data', patient\_index\_id\_ + '\_OS\_DVC\_FAZ.jpg'),

}

# mass = eod\_feature\_file\_.loc[eod\_feature\_file\_['ID'] == str(patient\_index\_id\_)]

eod\_parameter = eod\_feature\_file\_.loc[eod\_feature\_file\_['ID'] == int(patient\_index\_id\_)]

eod\_parameter\_result = {

'svc\_od\_parameter': eod\_parameter.loc[eod\_parameter['眼别'] == 'OD'].loc[eod\_parameter['层次'] == 'SVC'].values[0],

'svc\_os\_parameter': eod\_parameter.loc[eod\_parameter['眼别'] == 'OS'].loc[eod\_parameter['层次'] == 'SVC'].values[0],

'dvc\_od\_parameter': eod\_parameter.loc[eod\_parameter['眼别'] == 'OD'].loc[eod\_parameter['层次'] == 'DVC'].values[0],

'dvc\_os\_parameter': eod\_parameter.loc[eod\_parameter['眼别'] == 'OS'].loc[eod\_parameter['层次'] == 'DVC'].values[0],

}

return eod\_result\_path, eod\_parameter\_result

if \_\_name\_\_ == '\_\_main\_\_':

# 获取患者列表

patient\_name\_list, eod\_feature\_file = eod\_read\_patient\_list()

# 前端点击患者标签后返回患者SVC 和 DVC图像地址

image\_path, patient\_index\_id = return\_eod\_image\_path("20210902004\_向玲", patient\_name\_list)

# 前端点击分析后根据id返回患者的分析结果（展示图片路径及参数）

eod\_result\_path, eod\_parameter\_result = return\_eod\_result\_path(patient\_index\_id, eod\_feature\_file)

# print(eod\_result\_path, eod\_parameter\_result)

# print(patient\_name\_list)

# print(eod\_result\_path)

# print(eod\_parameter\_result)

tmp = eod\_parameter\_result

# for item in tmp.keys():

# print(item)

# for item in tmp.values():

# print(f'shape:{item.shape}, type:{type(item)}')

for key,values in zip(tmp.keys(),tmp.values()):

tmp[key] = values.tolist()

for item in tmp.keys():

print(item)

for item in tmp.values():

print(f'size:{len(item)}, type:{type(item)}')

coList = ['ID', 'Name', '类别', '眼别', '层次', 'FAZ', 'FAZ', 'FAZ', 'FAZ', 'FAZ', 'Dire', 'Dire', 'Tuot', 'VAD', 'VLD',

'Boxd', 'B\_num','Area\_VAD', 'Area\_VAD', 'Area\_VAD', 'Area\_VAD', 'Area\_VAD', 'Area\_VLD', 'Area\_VLD', 'Area\_VLD',

'Area\_VLD', 'Area\_VLD','Area\_Boxdim', 'Area\_Boxdim', 'Area\_Boxdim', 'Area\_Boxdim', 'Area\_Boxdim',

'Area\_Tortuosity','Area\_Tortuosity', 'Area\_Tortuosity', 'Area\_Tortuosity', 'Area\_Tortuosity',

'Area\_B\_num', 'Area\_B\_num', 'Area\_B\_num', 'Area\_B\_num', 'Area\_B\_num']

print(len(coList))

## change.py

import cv2

import nibabel as nib

import numpy as np

from PIL import Image

def show\_nii\_gif(nii\_file, save\_path="test", show\_frame=True):

#不显示帧数的话show\_frame = False

img = nib.load(nii\_file)

img\_arr = img.get\_fdata()

img\_arr = np.transpose(img\_arr, (0, 1, 2))

img\_arr = np.squeeze(img\_arr)

img\_max, img\_min = img\_arr.max(), img\_arr.min()

img\_arr = (img\_arr - img\_min) / (img\_max - img\_min) \* 255

img\_arr = img\_arr.astype(np.uint8)

img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

assert 3 == len(img\_arr.shape)

arr\_len, h, w = img\_arr.shape

if show\_frame:

for i in range(arr\_len):

img\_RGB\_list[i] = cv2.putText(img\_RGB\_list[i],

'{:>03d}/{:>03d}'.format(i + 1, arr\_len),

(50, 50),

cv2.FONT\_HERSHEY\_COMPLEX,

1,

(255, 255, 255),

2)

imgs = [Image.fromarray(img) for img in img\_RGB\_list]

imgs[0].save(save\_path+"\_正视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

img = nib.load(nii\_file)

img\_arr = img.get\_fdata()

img\_arr = np.transpose(img\_arr, (1,0,2))

img\_arr = np.squeeze(img\_arr)

img\_max, img\_min = img\_arr.max(), img\_arr.min()

img\_arr = (img\_arr - img\_min) / (img\_max - img\_min) \* 255

img\_arr = img\_arr.astype(np.uint8)

img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

assert 3 == len(img\_arr.shape)

arr\_len, h, w = img\_arr.shape

if show\_frame:

for i in range(arr\_len):

img\_RGB\_list[i] = cv2.putText(img\_RGB\_list[i],

'{:>03d}/{:>03d}'.format(i + 1, arr\_len),

(50, 50),

cv2.FONT\_HERSHEY\_COMPLEX,

1,

(255, 255, 255),

2)

imgs = [Image.fromarray(img) for img in img\_RGB\_list]

imgs[0].save(save\_path+"\_俯视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

img = nib.load(nii\_file)

img\_arr = img.get\_fdata()

img\_arr = np.transpose(img\_arr, (2, 1, 0))

img\_arr = np.squeeze(img\_arr)

img\_max, img\_min = img\_arr.max(), img\_arr.min()

img\_arr = (img\_arr - img\_min) / (img\_max - img\_min) \* 255

img\_arr = img\_arr.astype(np.uint8)

img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

assert 3 == len(img\_arr.shape)

arr\_len, h, w = img\_arr.shape

if show\_frame:

for i in range(arr\_len):

img\_RGB\_list[i] = cv2.putText(img\_RGB\_list[i],

'{:>03d}/{:>03d}'.format(i + 1, arr\_len),

(50, 50),

cv2.FONT\_HERSHEY\_COMPLEX,

1,

(255, 255, 255),

2)

imgs = [Image.fromarray(img) for img in img\_RGB\_list]

imgs[0].save(save\_path+"\_左视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

def show\_nii\_gif\_seg(nii\_file, save\_path="test", show\_frame=True):

#不显示帧数的话show\_frame = False

img = nib.load(nii\_file)

img\_arr = img.get\_fdata()

img\_arr = np.transpose(img\_arr, (0, 1, 2))

img\_arr = np.squeeze(img\_arr)

img\_max, img\_min = img\_arr.max(), img\_arr.min()

img\_arr = (img\_arr - img\_min) / (img\_max - img\_min) \* 255

img\_arr = img\_arr.astype(np.uint8)

img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

img\_RGB\_list = np.array(img\_RGB\_list)

seg = nib.load('aseg.nii.gz')

seg\_arr = seg.get\_fdata()

seg\_arr = np.transpose(seg\_arr, (0, 1, 2))

seg\_arr = np.squeeze(seg\_arr)

seg\_max, seg\_min = seg\_arr.max(), seg\_arr.min()

seg\_arr = (seg\_arr - seg\_min) / (seg\_max - seg\_min) \* 255

seg\_arr = seg\_arr.astype(np.uint8)

seg\_RGB\_list = [seg for seg in seg\_arr]

assert 3 == len(img\_arr.shape)

arr\_len, h, w = img\_arr.shape

# if show\_frame:

# for i in range(arr\_len):

# img\_RGB\_list[i] = cv2.putText(img\_RGB\_list[i],

# '{:>03d}/{:>03d}'.format(i + 1, arr\_len),

# (50, 50),

# cv2.FONT\_HERSHEY\_COMPLEX,

# 1,

# (255, 255, 255),

# 2)

# imgs = [Image.fromarray(img) for img in seg\_RGB\_list]

# # imgs = [Image.fromarray(cv2.applyColorMap((cv2.cvtColor(img, cv2.COLOR\_RGB2GRAY)).astype(np.uint8), cv2.COLORMAP\_JET)) for img in img\_RGB\_list]

# imgs[0].save(save\_path+"\_正视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

# if show\_frame:

# for i in range(arr\_len):

# seg\_RGB\_list[i] = cv2.putText(seg\_RGB\_list[i],

# '{:>03d}/{:>03d}'.format(i + 1, arr\_len),

# (50, 50),

# cv2.FONT\_HERSHEY\_COMPLEX,

# 1,

# (255, 255, 255),

# 2)

print(np.max(seg\_arr), img\_arr.shape)

color\_masks = np.random.randint(255, size=(255, 3))

for j in range(256):

print(j)

mask = seg\_arr[j]

img\_RGB\_list[mask] = img\_RGB\_list[mask] \* 0.5 + color\_masks[j] \* 0.5

# img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

# for s, i in zip(seg\_RGB\_list, img\_RGB\_list):

imgs = [Image.fromarray(img) for img in img\_RGB\_list]

# imgs = [Image.fromarray(cv2.applyColorMap((cv2.cvtColor(255-img, cv2.COLOR\_RGB2GRAY)).astype(np.uint8), cv2.COLORMAP\_JET)) for img in img\_RGB\_list]

imgs[0].save('aseg' + "\_正视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

# img = nib.load(nii\_file)

# img\_arr = img.get\_fdata()

# img\_arr = np.transpose(img\_arr, (1,0,2))

# img\_arr = np.squeeze(img\_arr)

#

# img\_max, img\_min = img\_arr.max(), img\_arr.min()

# img\_arr = (img\_arr - img\_min) / (img\_max - img\_min) \* 255

# img\_arr = img\_arr.astype(np.uint8)

# img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

# assert 3 == len(img\_arr.shape)

# arr\_len, h, w = img\_arr.shape

# if show\_frame:

# for i in range(arr\_len):

# img\_RGB\_list[i] = cv2.putText(img\_RGB\_list[i],

# '{:>03d}/{:>03d}'.format(i + 1, arr\_len),

# (50, 50),

# cv2.FONT\_HERSHEY\_COMPLEX,

# 1,

# (255, 255, 255),

# 2)

# imgs = [Image.fromarray(img) for img in img\_RGB\_list]

# imgs[0].save(save\_path+"\_俯视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

#

# img = nib.load(nii\_file)

# img\_arr = img.get\_fdata()

# img\_arr = np.transpose(img\_arr, (2, 1, 0))

# img\_arr = np.squeeze(img\_arr)

# img\_max, img\_min = img\_arr.max(), img\_arr.min()

# img\_arr = (img\_arr - img\_min) / (img\_max - img\_min) \* 255

# img\_arr = img\_arr.astype(np.uint8)

# img\_RGB\_list = [cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR) for img in img\_arr]

#

# assert 3 == len(img\_arr.shape)

# arr\_len, h, w = img\_arr.shape

# if show\_frame:

# for i in range(arr\_len):

# img\_RGB\_list[i] = cv2.putText(img\_RGB\_list[i],

# '{:>03d}/{:>03d}'.format(i + 1, arr\_len),

# (50, 50),

# cv2.FONT\_HERSHEY\_COMPLEX,

# 1,

# (255, 255, 255),

# 2)

# imgs = [Image.fromarray(img) for img in img\_RGB\_list]

# imgs[0].save(save\_path+"\_左视图.gif", save\_all=True, append\_images=imgs[1:], duration=50, loop=0)

show\_nii\_gif\_seg(nii\_file='T1.nii.gz')

## utils.py

import os

import pandas as pd

def Info\_print(info):

print(info)

def forder\_reader(path: str):

path = os.path.join('static',path,'input')

return os.listdir(path)

def readIntArt\_txt(file):

path = os.path.join("./static/int\_art/data/",file)

f = open(path, encoding="utf-8")

tmp = []

for line in f.readlines():

line = line.strip('\n')

tmp.append(line)

item1, item2 = tmp[1].split(' ')

data = []

data.append(tmp[0])

data.append(item1)

data.append(item2)

data.append(tmp[2])

res = ' '.join(data)

print(res)

f.close()

return res

def readElec\_txt(file):

path = os.path.join("./static/elec/data/",file)

f = open(path, encoding="utf-8")

tmp = []

for line in f.readlines():

line = line.strip('\n')

tmp.append(line)

res = ' '.join(tmp)

f.close()

return res

def readBrain\_csv(base\_name):

df = pd.read\_csv('static/brain/output/'+base\_name)

head = df.head(1).values[0]

head = head.tolist()

columns = df.columns.values

columns = columns.tolist()

info\_list = []

for x, y in zip(head,columns):

info\_list.append([x,y])

return info\_list

if \_\_name\_\_ == "\_\_main\_\_":

readBrain\_csv()