

# Lab2 Information Retrieval Report

1952107 Zixuan Wang

## 1 Requirements

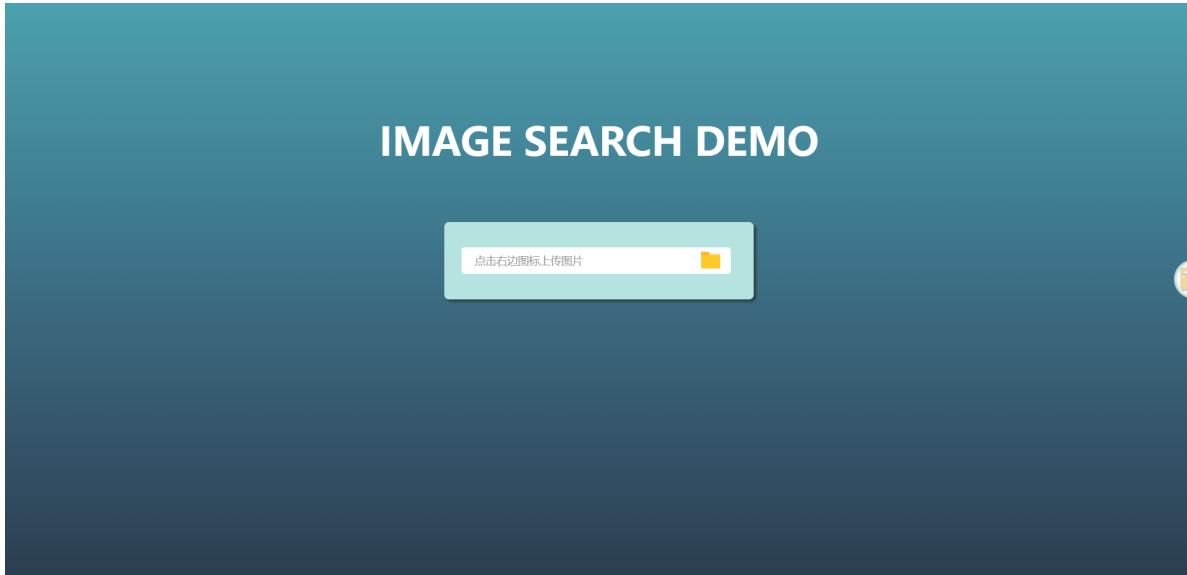
1. Choose an image and upload it
2. Show the thumbnail of image which has been uploaded
3. Preview the enlarged image by hovering the thumbnail
4. Search similar images by clicking the search button
5. Show the total number and the tags of searching results
6. Select certain tag when reviewing results
7. Add selected images to a favorite list
8. Delete images from the favorite list

## 2 Designs and features

### 2.1 Formulation

There is a large title **IMAGE SEARCH DEMO** displayed in the center of the page tells users the function of this search engine.

A fresh card has an input box that tells users how to upload image. You can click the button in the right of this input box to upload image.



The box will make a gradual transformation to show the thumbnail of image and the file name of the image. The clear button and the search button is visible too.

this design can avoid to search before upload image.

## IMAGE SEARCH DEMO



Users can view the large image by hovering the thumbnail with mouse.

## IMAGE SEARCH DEMO



Click the clear button can return to initial state.

## 2.2 Initiation of action

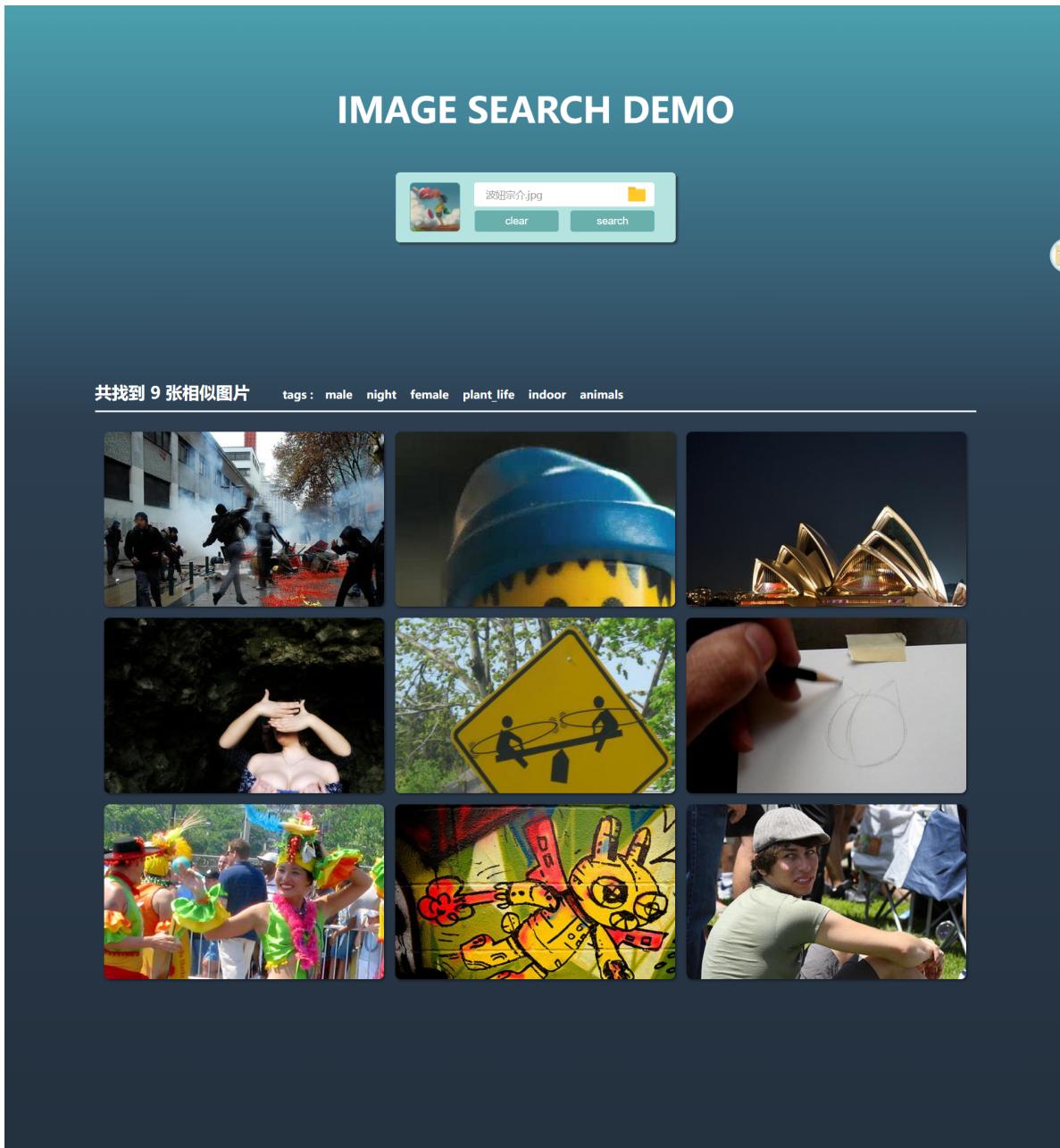
When user clicks the search button, the system will start the search task. When waiting to the result, there is a gif indicate the searching is processing.

## IMAGE SEARCH DEMO



## 2.3 Review of results

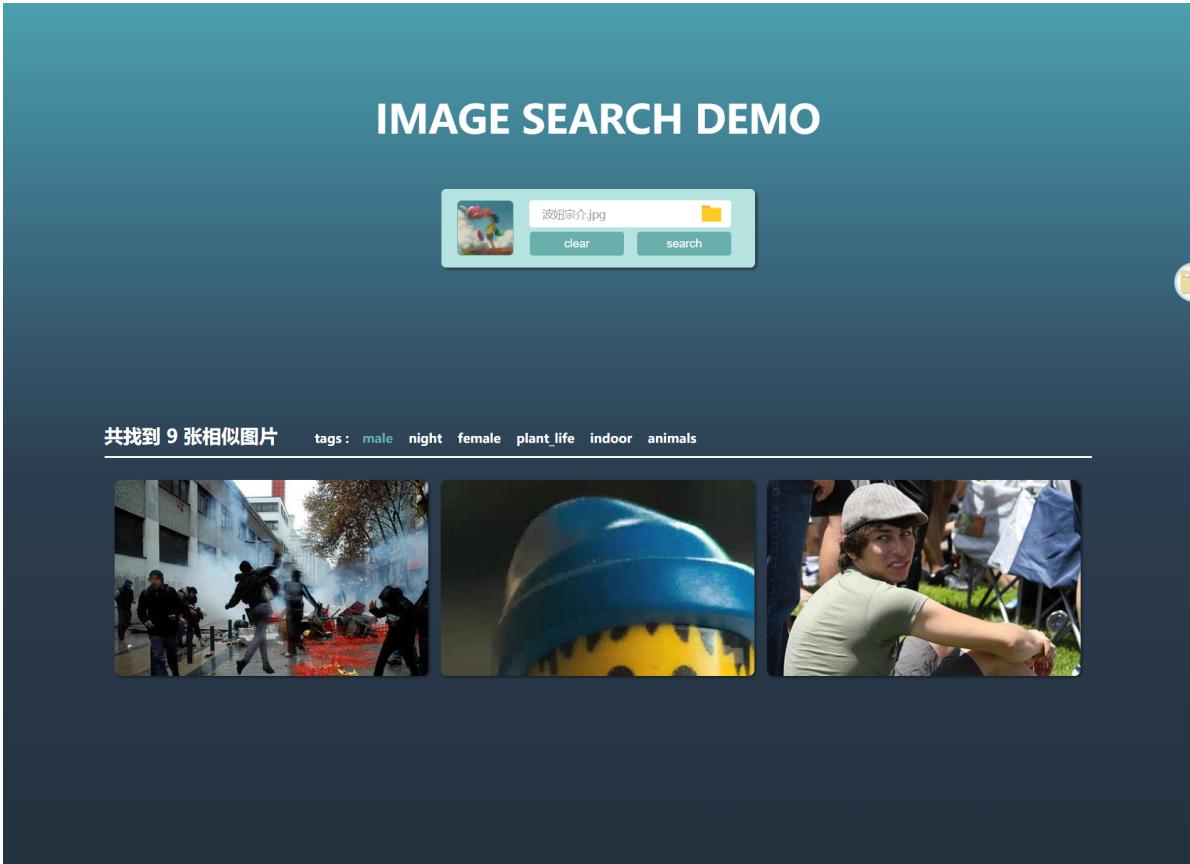
The website will provide the total num of result and the tags of images. And the similar images will be orderly displayed in the page.



## 2.4 Refinements

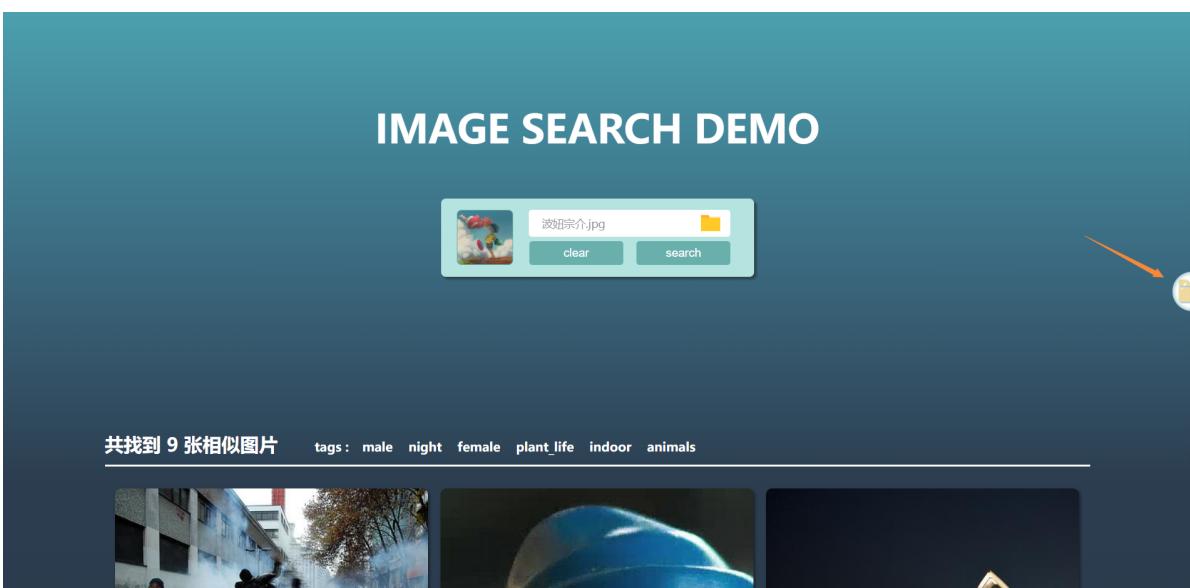
When user choose a tag, the website will filter relative image by their tags. Images which have the same tag to the selected tag will be visible and the different ones will be not visible.

## IMAGE SEARCH DEMO



### 2.5 Use

The is a favorite list on the right of the page. When user clicks the button fixed on the right side of the screen, the favorite list will move out slowly from the right of screen.



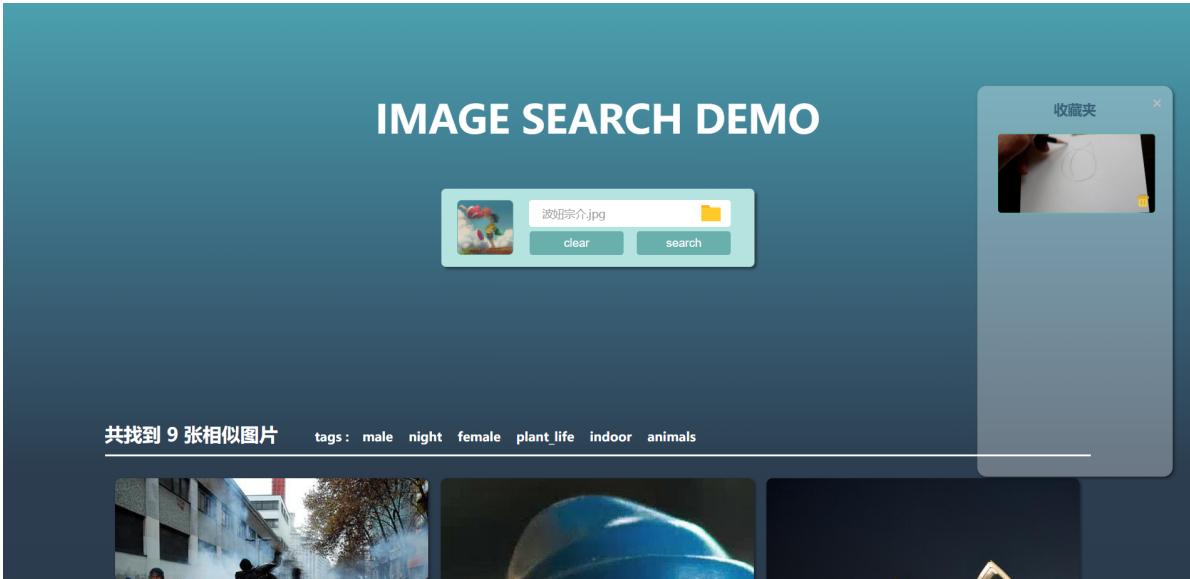
# IMAGE SEARCH DEMO

The screenshot shows a search interface. At the top, there's a file selection area with a thumbnail of a cartoon character, a text input field containing "波妞宗介.jpg", and two buttons labeled "clear" and "search". To the right is a large, semi-transparent gray overlay labeled "收藏夹" (Favorites) with a close button "x". Below this, a message says "共找到 9 张相似图片" (Found 9 similar images) and lists tags: male, night, female, plant\_life, indoor, animals. A row of three small images is displayed.

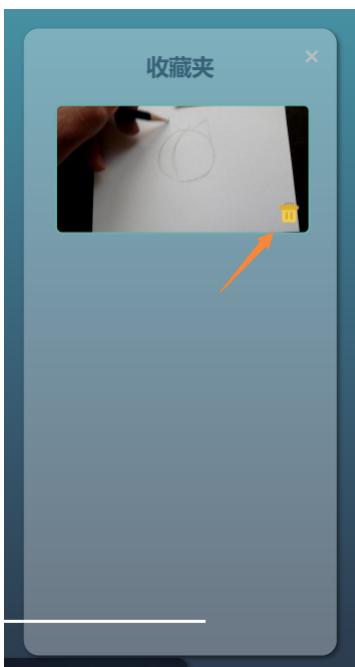
If user likes an image, he/she can click the favorite icon fixed on the right-bottom of it to add it into the favorite list.



# IMAGE SEARCH DEMO



If user want to delete the image from favorite list, just click the delete icon fixed on the right-bottom of this image.



## 3. Kernel Code

### 3.1 upload and preprocessor function

1. Add an click event for the upload button
2. Record the file information and use dom to change the css of relative elements

```

uploadPicture(e) {
    this.picture.file = e.target.files[0]
    this.picture.url = URL.createObjectURL(e.target.files[0])
    let input = document.querySelector('#search_bar')
    input.className = 'search_bar2'
    let div = document.querySelector('#preview')
    div.style.backgroundImage = "url(" + this.picture.url + ")"
    div.style.display = "inline-block"
    let btn = document.querySelector('.button_group')
    btn.style.display = "block"
}

```

## 3.2 Searching function

1. Send the image uploaded by user to the back-end by axios api.
2. Back-end system searching the pictures and sent back pictures in json format.
3. Show the overview of the results and the images on the page by using dom.

```

async getSimilarPicture() {
    let loading = document.querySelector("#loading")
    loading.style.display = "block"
    let formData = new FormData()
    formData.append("file", this.picture.file)
    const {data:data} = await axios({
        url: 'http://127.0.0.1:5000/imgUpload',
        method: 'post',
        data: formData
    })
    this.pictureList = []
    this.tags = []
    this.currentTag = ''
    let set = new Set()
    for (let i = 0; i < 9; i++) {
        let key = "image" + i;
        let url = data[key][0].substring(0, 7) + '/' + data[key][0].substring(7);
        let tag = data[key][1]
        this.pictureList.push({
            url: "http://127.0.0.1:5000" + url,
            tag: tag
        });
        if (!set.has(tag)) {
            set.add(tag)
            this.tags.push(tag)
        }
    }
    let body = document.querySelector("#search_body")
    body.style.height = "80%"
    let content = document.querySelector("#search_content")
    content.style.display = "block"
    loading.style.display = "none"
},

```

### 3.3 Favorite list function

1. Get the favorite list from the localStorage of web cache in the created hook function
2. Add the image to the favorite list if it has not been saved
3. Add the image to the localStorage of web cache in order to get the save the image a longer time
4. Remove the image from list and localStorage

```
created: function () {
  if(window.localStorage.getItem('collectList')!==''&&
    window.localStorage.getItem('collectList')!=null)
  {
    this.collectList = window.localStorage.getItem('collectList').split(',')
    for (let item of this.collectList) {
      this.collectSet.add(item)
    }
  }
},
methods:{
  favorite(i) {
    if (!this.collectSet.has(this.pictureList[i].url)){
      this.collectList.push(this.pictureList[i].url)
      this.collectSet.add(this.pictureList[i].url)
      let collectList = ''
      for (let item of this.collectList) {
        collectList += item + ','
      }
      collectList = collectList.substring(0, collectList.length - 1)
      window.localStorage.setItem('collectList', collectList)
      this.$message.success("成功加入收藏夹")
      return
    }
    this.$message.warning("该图片已在收藏夹中")
  },
  deletePicture(index){
    this.collectSet.delete(this.collectList[index])
    this.collectList.splice(index,1)
    let collectList = ''
    for (let item of this.collectList) {
      collectList += item + ','
    }
    collectList = collectList.substring(0, collectList.length - 1)
    window.localStorage.setItem('collectList', collectList)
    this.$message.success("成功将次图片移出收藏夹")
  }
}
```