Assignment 3

Image search
1951095 Yiwen Liang

Introduction

Image retrieval is a visual search task where the goal is, given a query image, to retrieve all images in a potentially very large database of images that contain the same object instances as the query image. The image retrieval implemented by this project is based on the similarity measure between images. When the similarity between two images is greater than a certain threshold, the n images with the highest scores are selected as the retrieval result. The image feature extraction and score comparison in the project is a deep model built using the tensorflow framework.

The main interface codes and function codes are in `"server/template/index.html"` and `"server/static/js/main.js"` respectively.

Project design principles

- Image retrieval follows the five-stage search framework.
- Searching in a large number of documents should be efficient, and the returned results should be real and effective.
- Users can operate the retrieved pictures according to their needs.
- The image retrieval interface should be user-friendly, and the page design should be beautiful, neat and generous.
- The interface is attractive to users.

How to run:

1. Download imagenet folder, extraxt and keep it in server directory.
2. Download datasets for footwares, apparels keep them inside a directory under upload folder. Final folder strcture will be as below.

```
root folder
|
└──lib
|
└──server
|   |──rest-server.py
|   |──imagenet
|   |──static
|   |──templates
|   |──uploads
|       |──dogs_and_cats
|           |──shoes
|           |──sandals
|           |──slippers
|           |──boots
|           |──apparels
```

3. .Run image vectorizer which passes each data through an inception-v3 model and collects the bottleneck layer vectors and stores in disc. Edit dataset paths accordingly indide the image_vectorizer.py.

```
python server/image_vectorizer.py
```

This will generate two files namely, image_list.pickle and saved_features.txt. Keep them inside lib folder where search.py script is available.

4. Start the server by running rest-server.py. This project uses flask based REST implementation for UI.

```
python server/rest-server.py
```

5. Once the server starts up, access the url 127.0.0.1:5000 to get the UI. Now upload any file and see 9 similar images. You can change the value of K from 9 to any values, but dont foreget to update the html file accordingly for displaying.

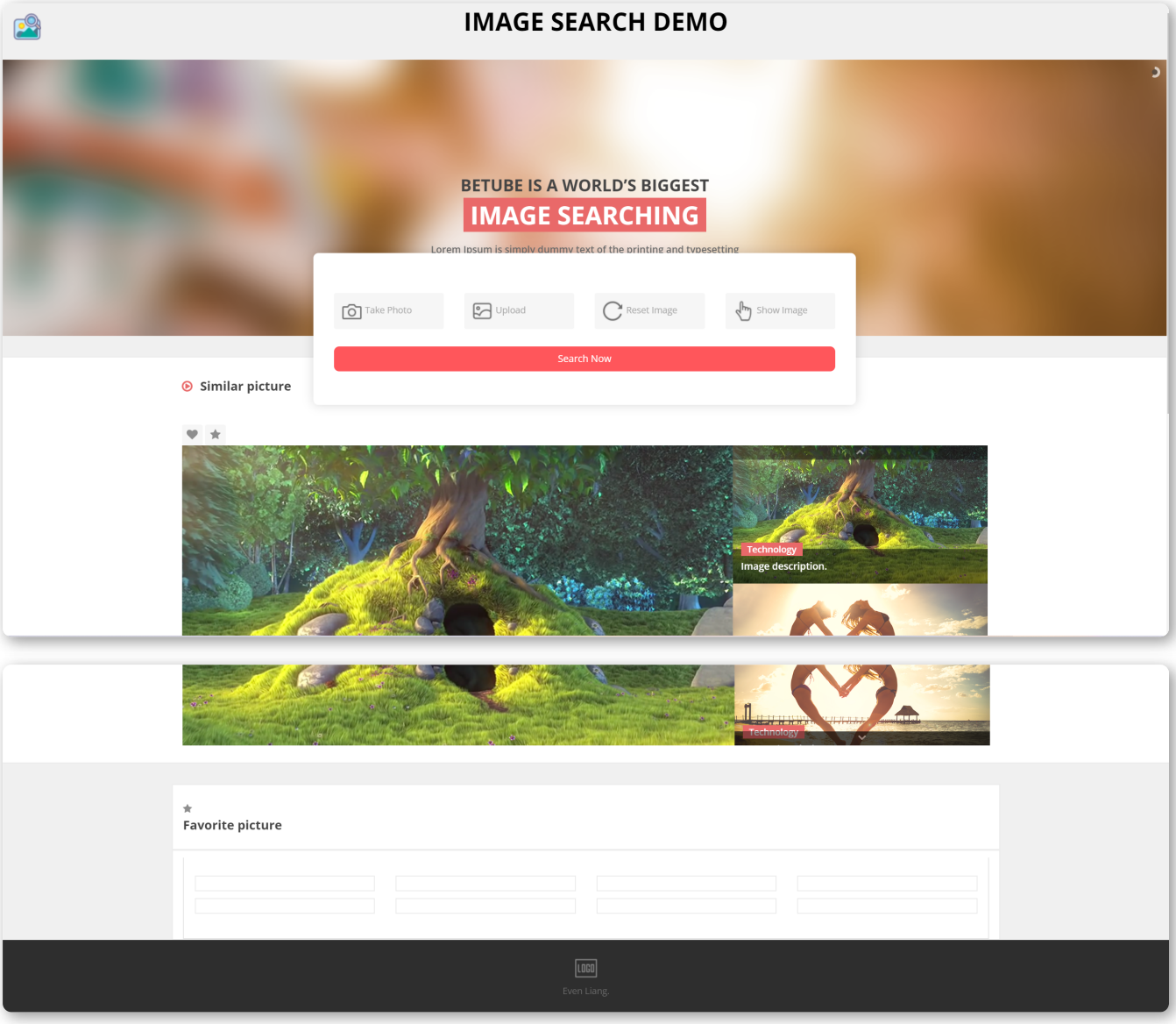
Five-stage search framework

Brief

A five-stage search framework help to coordinate design practices and satisfy the needs of all users. The five-stage framework in my project are:

- Formulation: the source of the information, the input image. User can click `"Upload"` button.
- Initiation of action: Execute the searching. User can click `"Search Now"` button.
- Review of results: User review results in a list. All retrieved pictures are placed in the sidebar, and users can browse by swiping.
- Refinement: Search interfaces can provide meaningful messages to explain search outcomes. Output category information of searched images to help users filter images. And implemented a filter box to help users refine their needs.
- Use: Results may be used, such as sharing, starring etc. We set up `"like"` and `"star"` button for the pictures.

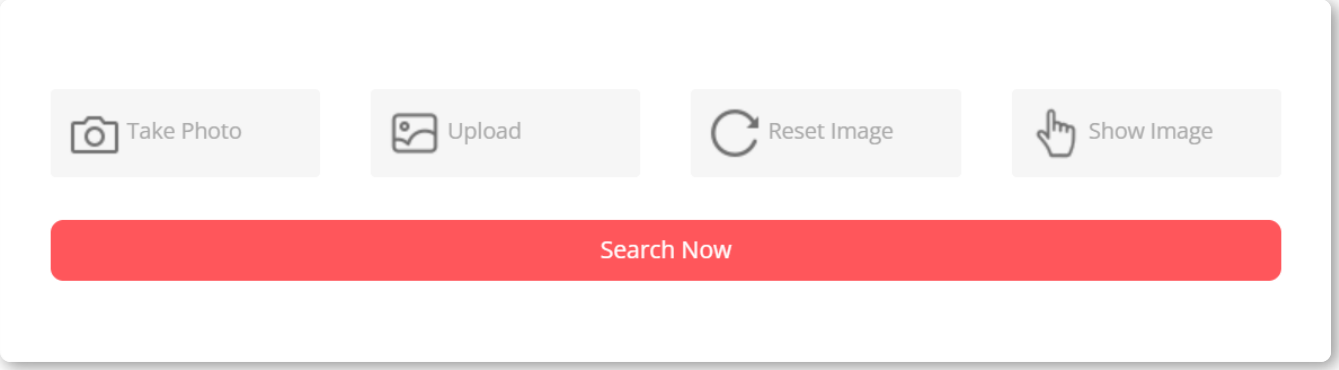
Main user interface



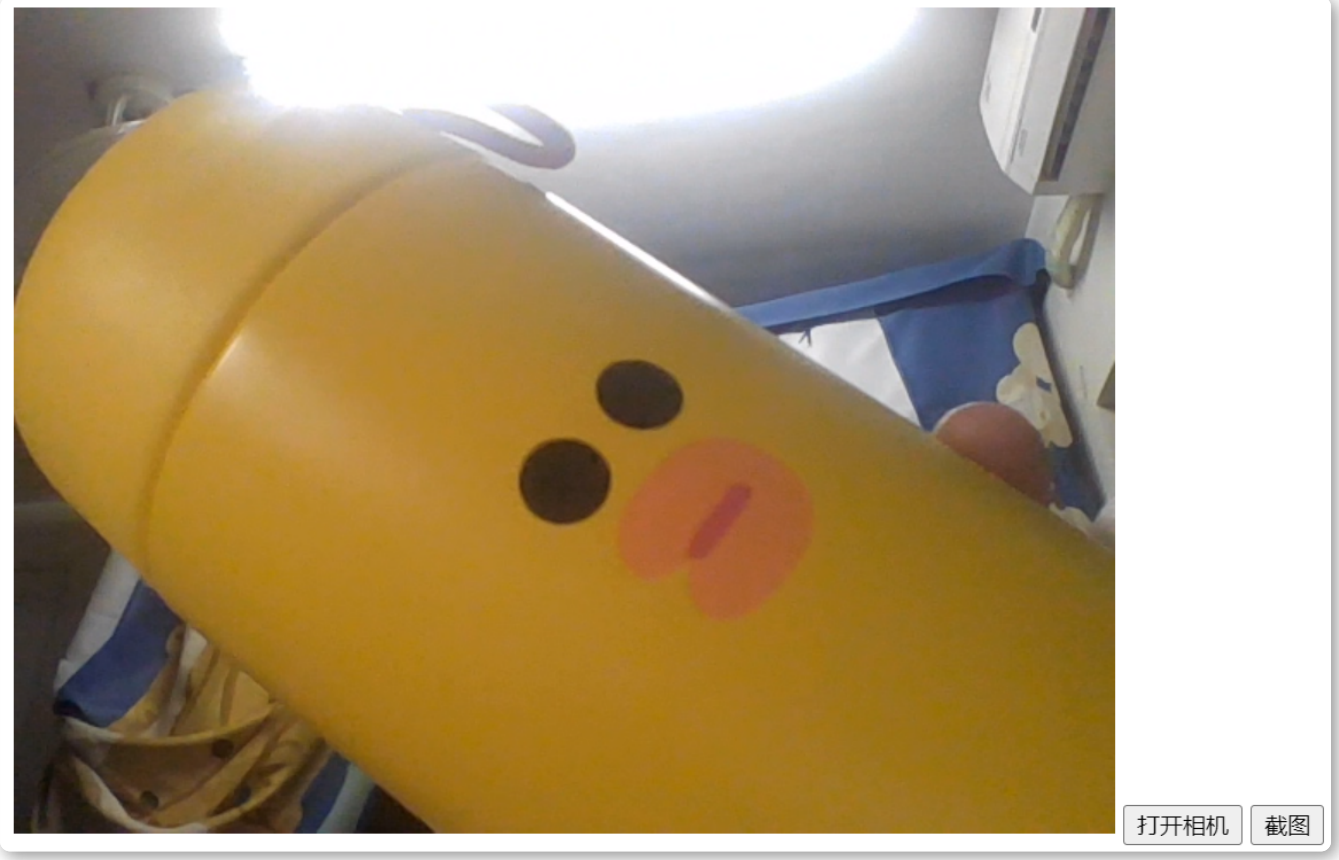
Formulation

This part is mainly implemented by the search bar part, which consists of 4 buttons:

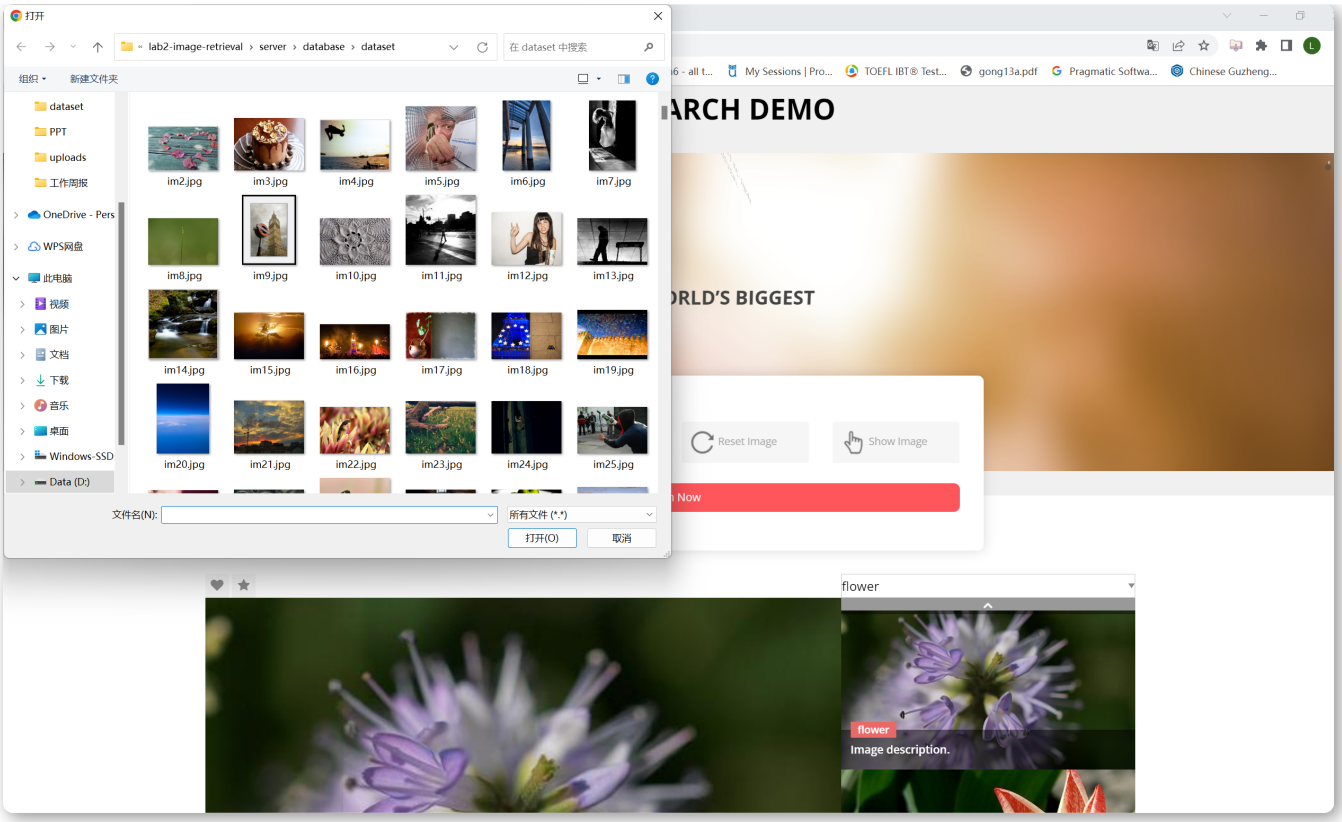
- Take Photo： When the user has no pictures stored on the PC, he can select this button to jump to a new interface to take pictures.
- Upload: The user uploads the image to be retrieved.
- Reset Image: The user is dissatisfied with the pictures uploaded by themselves, or the upload is wrong, delete the uploaded pictures.
- Show Image：Show uploaded images.



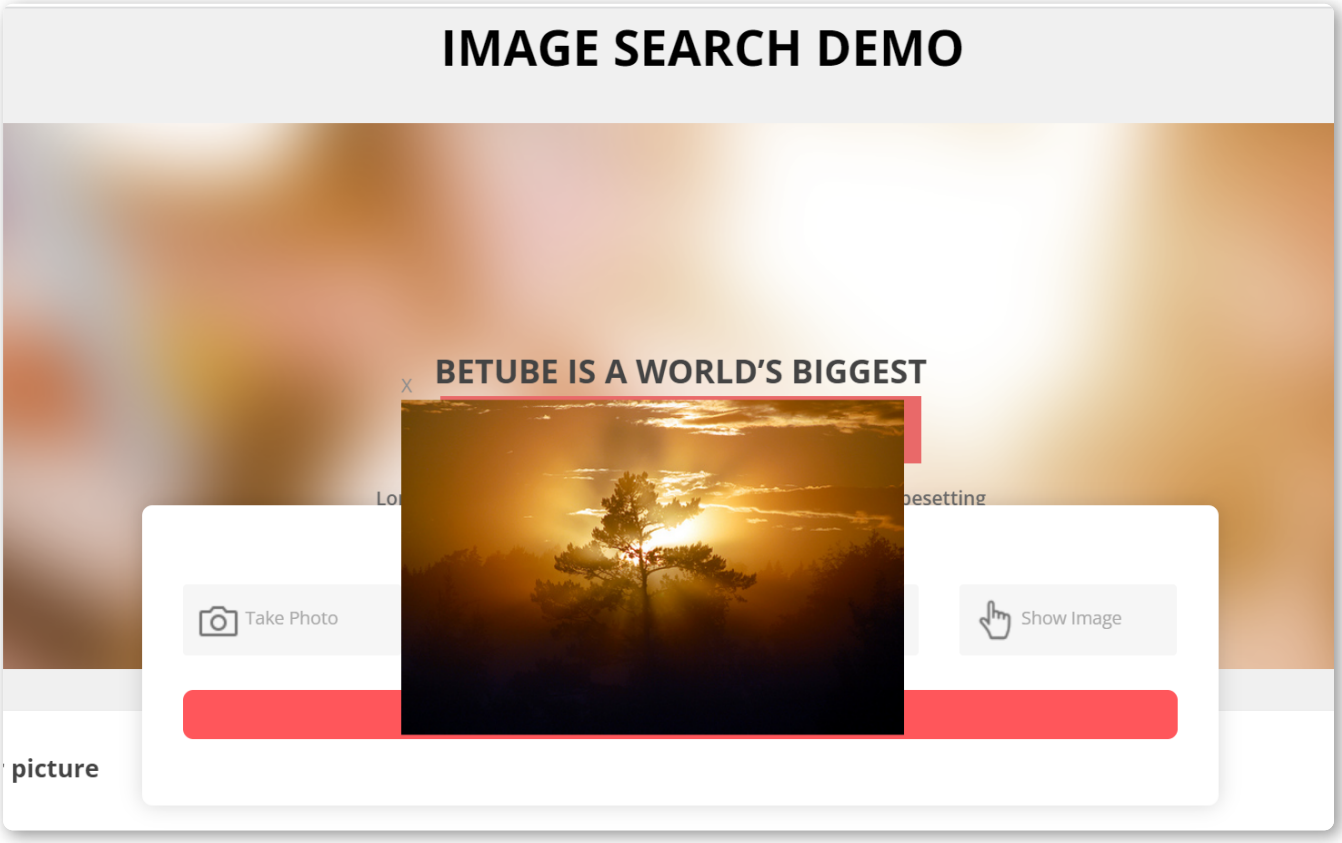
Click `take photo`:



Click `Upload`:



Click `Show Image` : (Note: your select file must be in `"/static/upload"`)



code:

(1) Take photo

```
<script language="javascript">
    // Grab elements, create settings, etc.
var video = document.getElementById('video');
```

```
// Get access to the camera!
    if(navigator.mediaDevices &&
navigator.mediaDevices.getUserMedia) {
        // Not adding `{ audio: true }` since we only want video
now
        navigator.mediaDevices.getUserMedia({ video: true
}).then(function(stream) {
            //video.src = window.URL.createObjectURL(stream);
            video.srcObject = stream;
            video.play();
        });
    }

    function openCamera(){
            // Grab elements, create settings, etc.
            debugger;
    var video = document.getElementById('video');

    // Get access to the camera!
     if(navigator.mediaDevices &&
navigator.mediaDevices.getUserMedia) {
         // Not adding `{ audio: true }` since we only want video
now
         navigator.mediaDevices.getUserMedia({ video: true
}).then(function(stream) {
             //video.src = window.URL.createObjectURL(stream);
             video.srcObject = stream;
             video.play();
         });
     }
    }
</script>
```

(2)Show image

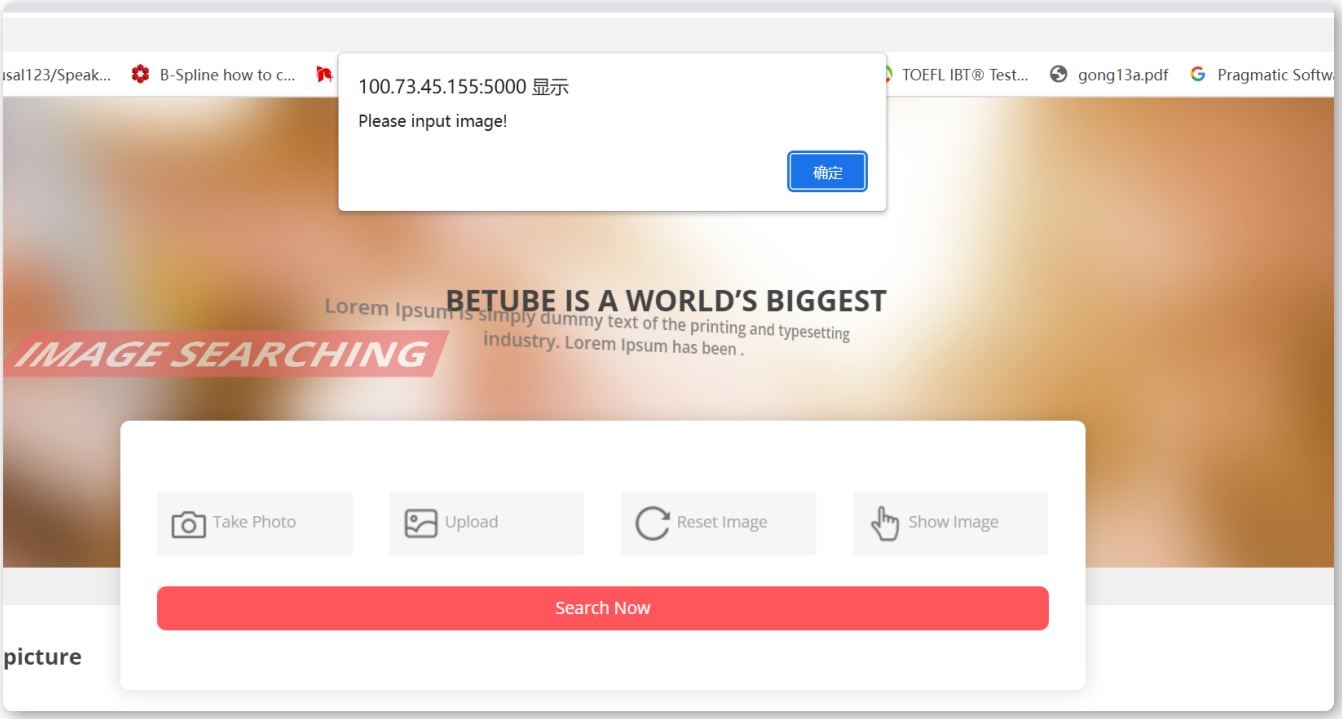
```
function show_image(){
    // $("#show_image_popup").fadeIn();
    debugger;
    var selectedFile = document.getElementById('file').files[0];
    var name = selectedFile.name;//读取选中文件的文件名
    var path = "../static/uploads/"
    console.log("文件名:"+name, "路径:" +path+name )
    document.getElementById("large-image").src = path+name;
    $("#show_image_popup").show()
}
```

```
function close_img(){
    $("#show_image_popup")[0].style.display="none"
}
```

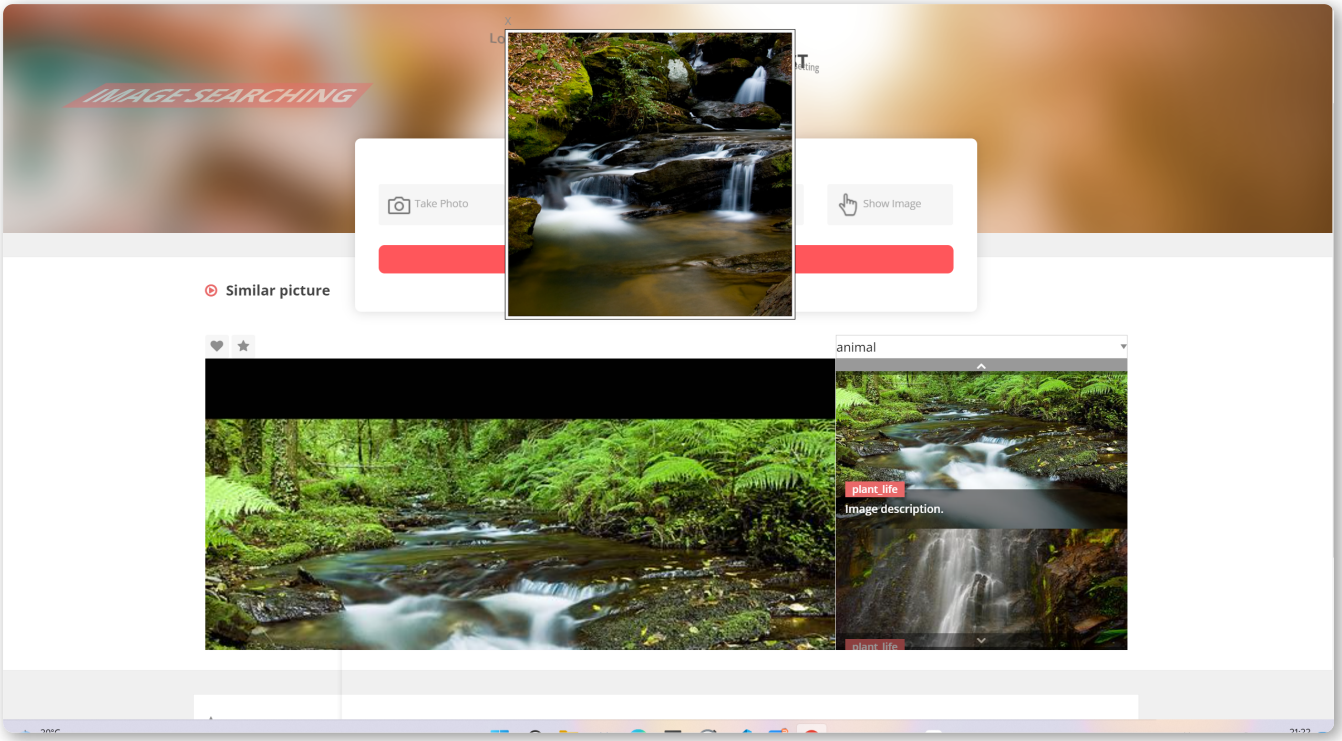
Initiation of action

Click `Search Now`:

(1) the uploaded file is **not** an image:



(2) the uploaded file is an image:



code:

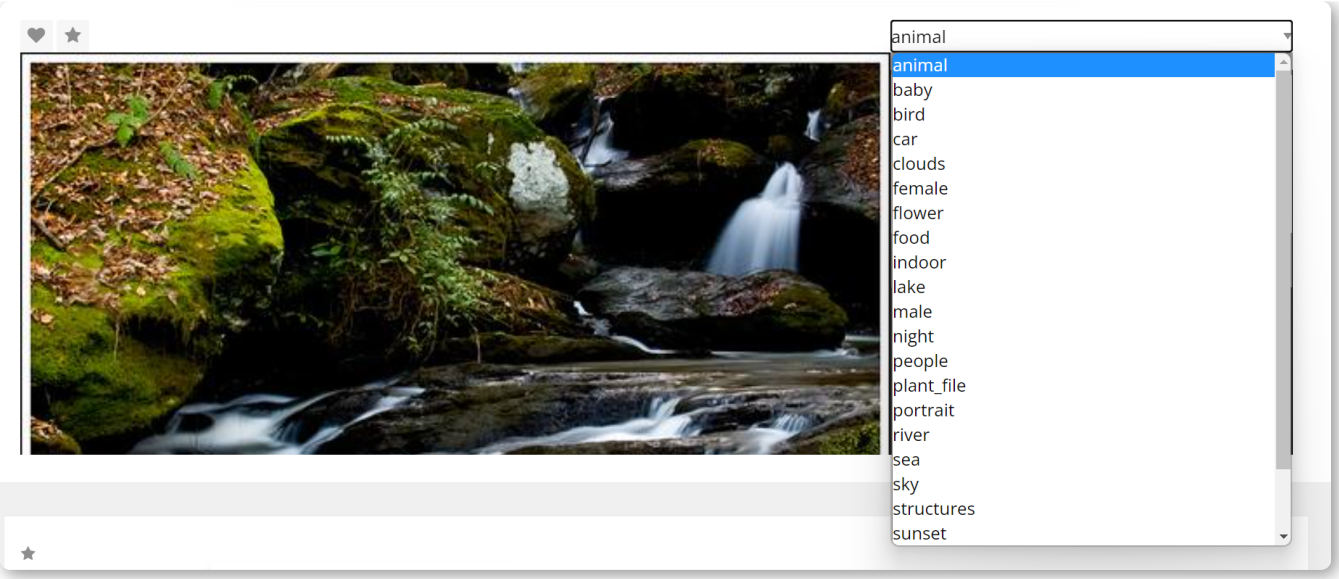
```
#=====================================================================
=====================================================================
#

#  This function is used to do the image search/image retrieval
#

#=====================================================================
=====================================================================
@app.route('/imgUpload', methods=['GET', 'POST'])
def upload_img():
    print("image upload")
    result = 'static/result'
    if not gfile.Exists(result):
          os.mkdir(result)
    shutil.rmtree(result)

    if request.method == 'POST' or request.method == 'GET':
        print(request.method)
        # check if the post request has the file part
        if 'file' not in request.files:
            print('No file part')
            return redirect(request.url)

        file = request.files['file']
        print(file.filename)
        # if user does not select file, browser also
        # submit a empty part without filename
        if file.filename == '':

            print('No selected file')
            return redirect(request.url)
        if file:# and allowed_file(file.filename):
            filename = secure_filename(file.filename)
            file.save(os.path.join(app.config['UPLOAD_FOLDER'],
filename))
            inputloc = os.path.join(app.config['UPLOAD_FOLDER'],
filename)
            recommend(inputloc, extracted_features)
            # os.remove(inputloc)
            image_path = "/static/result"

            # Traverse to find the image's tag
            typename = []
            tagpath = "./database/tags/"
```

```
            tagslist = os.listdir(tagpath)
            for image in os.listdir(result):

                imgName,extension = os.path.splitext(image)
                for tagfile in tagslist:
                    with open(tagpath+tagfile,'r',encoding = 'UTF-
8') as f:
                        class_names = f.readlines()
                    class_names = [c.strip() for c in class_names]
                    if imgName[2:] in class_names:
                        tagName,_ = os.path.splitext(tagfile)
                        typename.append(tagName)
                        break


            image_list =[os.path.join(image_path, file) for file
in os.listdir(result)
                                if not file.startswith('.')]

            images = {
            'image0':image_list[0],
            'image1':image_list[1],
            'image2':image_list[2],
            'image3':image_list[3],
            'image4':image_list[4],
            'image5':image_list[5],
            'image0_type':typename[0],
            'image1_type':typename[1],
            'image2_type':typename[2],
            'image3_type':typename[3],
            'image4_type':typename[4],
            'image5_type':typename[5],

            # 'image6':image_list[6],
            # 'image7':image_list[7],
            # 'image8':image_list[8]
              }
            return jsonify(images)
```

Review of results

The searched image results are on the sidebar, and the search bar includes the type and image description of each image. Users can click the up and down buttons to swipe to browse all the pictures.

The left side is the big picture of the picture. Click a picture in the sidebar, the big picture on the left will be displayed.



code:

```
function fun(){

    var selectedFile = document.getElementById('file').files[0];
    var name = selectedFile.name;//读取选中文件的文件名

    //获取最后一个.的位置
    var index= name.lastIndexOf(".");
    //获取后缀
    var ext = name.substr(index+1);
    //判断是否是图片
    console.log("该文件是否为图片: " + isAssetTypeAnImage(ext));

    if(!isAssetTypeAnImage(ext)){
        alert("Please input image!");
        return false;
    }

    $('#load').show();
    $("form").submit(function(evt){
    //$('#loader-icon').show();
```

```
    evt.preventDefault();

            //$('#loader-icon').show();
        var formData = new FormData($(this)[0]);

    $.ajax({
            url: 'imgUpload',
            type: 'POST',
            data: formData,
            //async: false,
            cache: false,
            contentType: false,
            enctype: 'multipart/form-data',
            processData: false,

            success: function (response) {
        $('#load').hide();
        $('#row1').show();
        //$('#clear').show();
                //console.log(response[1]);

 //document.getElementById("predictedResult").innerHTML= response;

                document.getElementById("img0").src =
response.image0;
                document.getElementById("img1").src =
response.image1;
                document.getElementById("img2").src =
response.image2;
                document.getElementById("img3").src =
response.image3;
                document.getElementById("img4").src =
response.image4;
                document.getElementById("img5").src =
response.image5;

                document.getElementById("main_img0").src =
response.image0;
                document.getElementById("main_img1").src =
response.image1;
                document.getElementById("main_img2").src =
response.image2;
                document.getElementById("main_img3").src =
response.image3;
                document.getElementById("main_img4").src =
response.image4;
                document.getElementById("main_img5").src =
response.image5;
```

```
                document.getElementById("sidebar-name1").innerHTML
= response.image0_type;
                document.getElementById("sidebar-name2").innerHTML
= response.image1_type;
                document.getElementById("sidebar-name3").innerHTML
= response.image2_type;
                document.getElementById("sidebar-name4").innerHTML
= response.image3_type;
                document.getElementById("sidebar-name5").innerHTML
= response.image4_type;
                document.getElementById("sidebar-name6").innerHTML
= response.image5_type;

            }
    });

return false;
 })};
```

Refinement

The user clicks a category in the search box to filter images. The selected picture will be displayed on the top layer, and other parts will display "None" .

code:

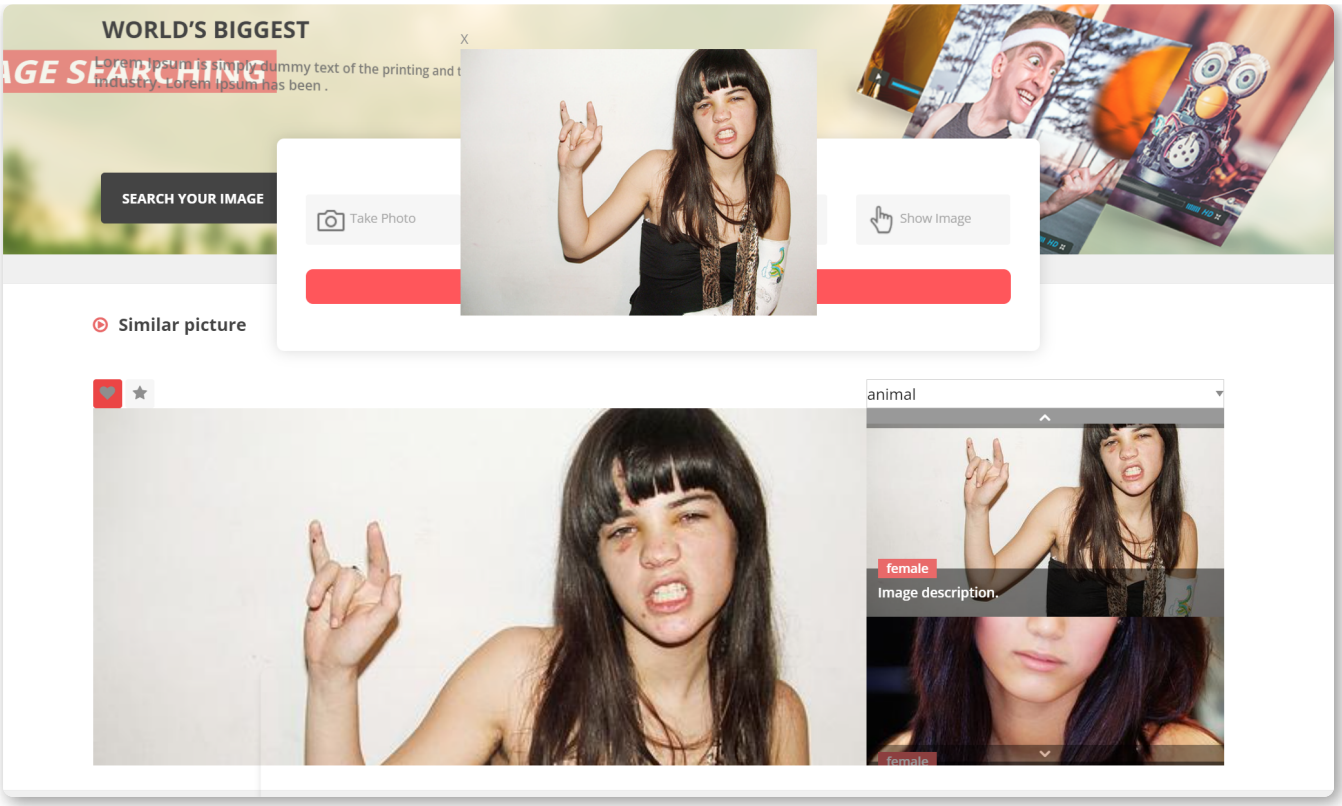
```
function typeSelect(){
debugger;
 var value = $('#typechange')[0].value;
 //alert(value);
 var satisfy_url = [];
for(var i=0;i<6;i++){
```

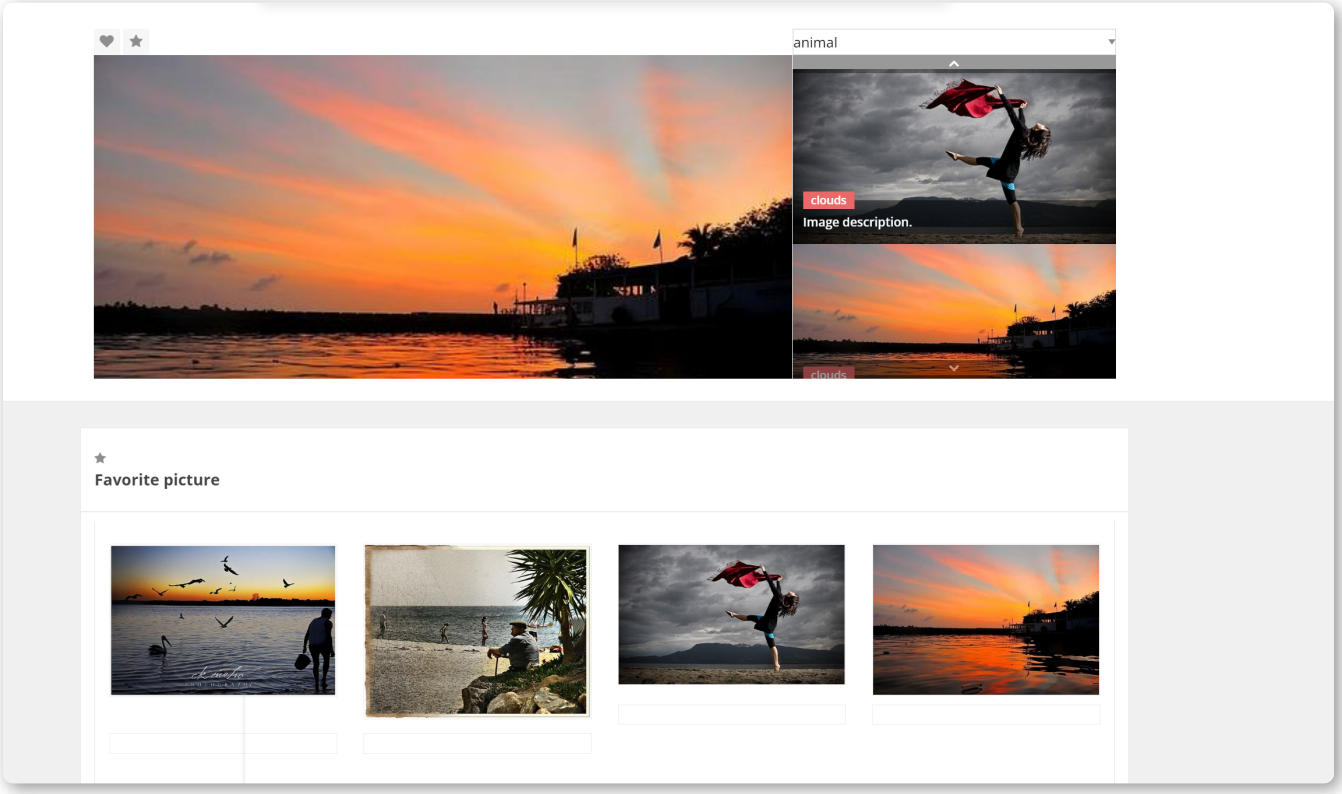
```
    var caption = document.getElementById("sidebar-name"+
(i+1)).innerText;
    if (caption != value)
    {
        //document.getElementById("sidebar-name" + (i+1)).value =
None;
        var imgName = "img"+i;
        //document.getElementById("sidebar-name" + (i+1)).src =
"../static/image/gray.jpg"
    }
    else{
        var imgName = "img"+i;
        satisfy_url.push(document.getElementById(imgName).src);
    }
    var imgName = "img"+i;
    //先置为灰色，None
    document.getElementById(imgName).src =
"static/image/gray.jpg";
    document.getElementById("sidebar-name" + (i+1)).innerText =
"None";
}

for(var i=0;i<satisfy_url.length;i++){
    document.getElementById("sidebar-name"+ (i+1)).innerText =
value;
    var imgName = "img"+i;
    document.getElementById(imgName).src = satisfy_url[i];

}
}
```

Use

After completing the search, the user can operate on the image:

- Save: Users can right-click the picture and save the picture locally.
- Like: Users can click the heart in the upper left corner of the big picture to like the picture
- Star: Users can click the star in the upper left corner to collect pictures. Favorite pictures will appear below.





code:

```
function changeColor(){
    var heart = document.getElementById("love");
    heart.style.background = "#e82626db";
}

function getCollection(){
        // $("#show_image_popup").fadeIn();
    for (var i=0;i<6;i++)
```

```
{
    var blockName = "image "+(i+1);
    if(document.getElementsByClassName(blockName)
[0].style.display!="none")
    {
        var imgName = "main_img"+i;
        collect.push(document.getElementById(imgName).src);
        break;
    }
}
    for(var i=0;i<collect.length;i++){
        console.log(collect[i])
        document.getElementById("collect_img" + (i+1)).src =
collect[i]
    }
}
```

Innovation

- Based on the teacher's code, this project enriches other interface elements, such as footer, composite picture viewing box, picture collection box, etc.
- The interface design of this project is beautiful and concise. The background of the search box is designed in the form of animation, which is more watchable and technological.
- Implemented conditional judgment for uploading files.
- Realize multiple viewing methods of searched images, including thumbnail view on the sidebar and large image display on the left.
- Realize the category display of pictures and the display of picture descriptions.