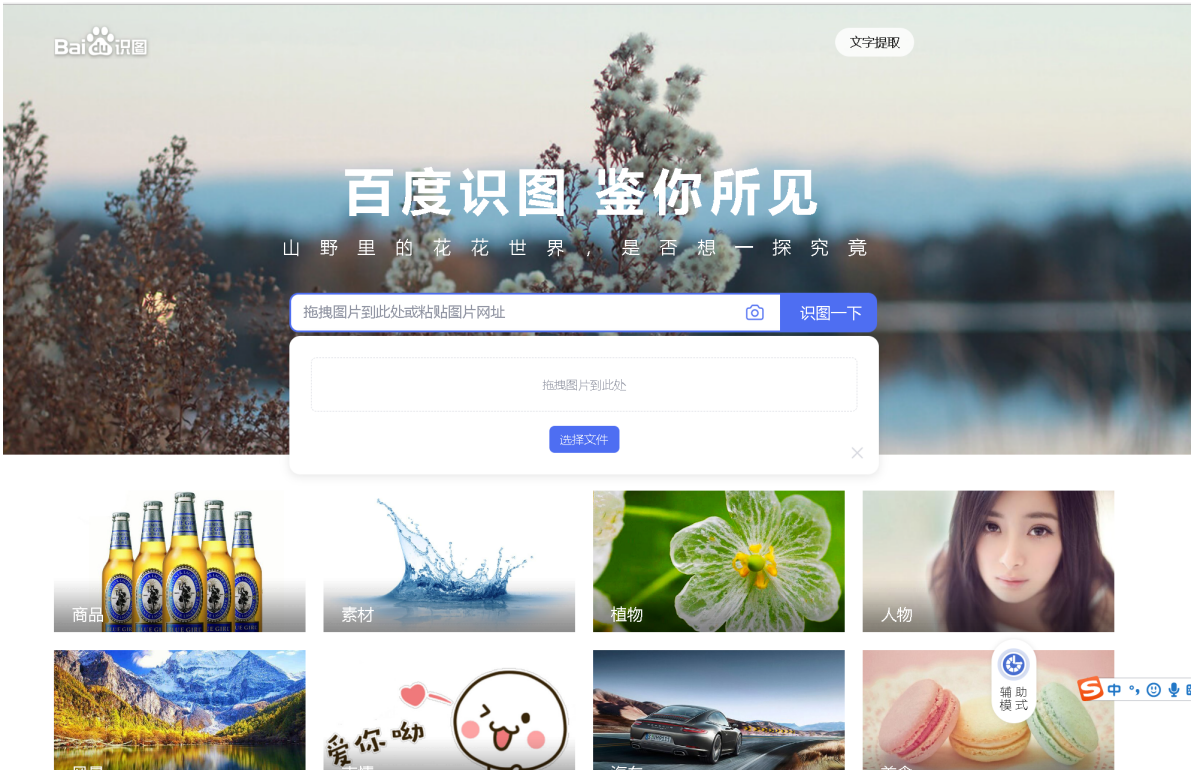
Lab 2 Information Retrieval

I. Describe the requirements of an image search task

In the following, I will describe some basic requirements for the image search task, using Baidu Map Recognition as an example.

1. Bold text or image cues to inform users of the page's functionality for searching

Take Baidu Maps as an example:



2. Upload images in diffenrent ways

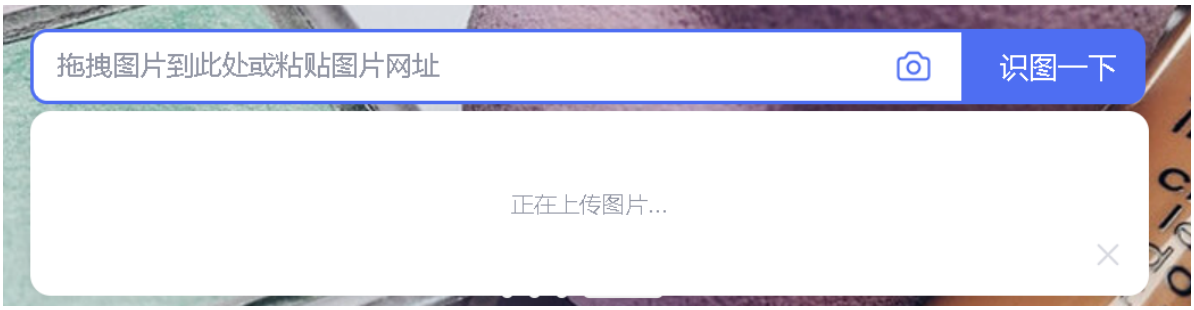
As shown above, Baidu supports uploading images from local or input links.

3. Confirm upload and start search

For local files, Baidu will directly jump to the image search results page; For image links, users need to click on the image to get search results.

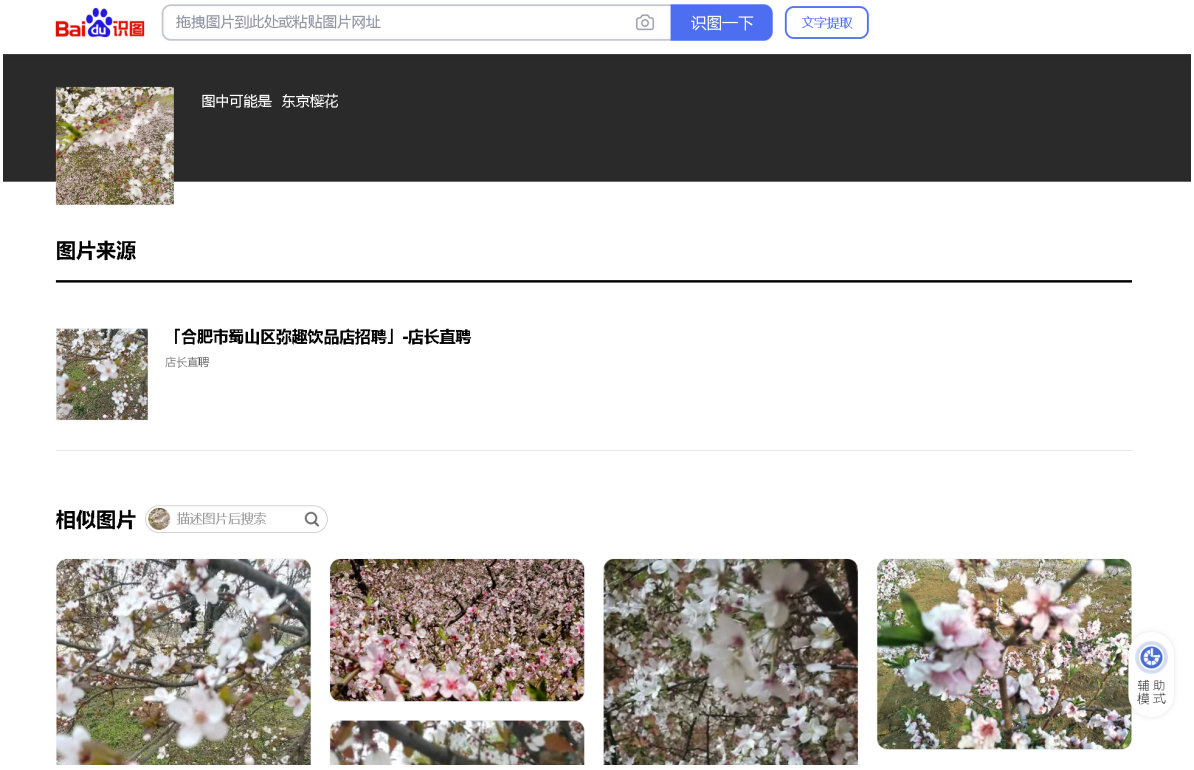
By the way, I think there is one aspect of the design that is not very user-friendly: the user needs to click on the camera icon to open the drag-and-drop area, and if you drag a local image directly into the text field, it will open the image directly in the browser, not the search results.

4. Load icon to indicate a search is underway



In the picture, before the search is completed, Baidu will display "uploading pictures".

5. Display the identified images and other information



Upload a photo of cherry blossoms in Tongji, Baidu identified the object as "Tokyo cherry blossom", and gave the possible source of the picture with a large number of similar pictures.

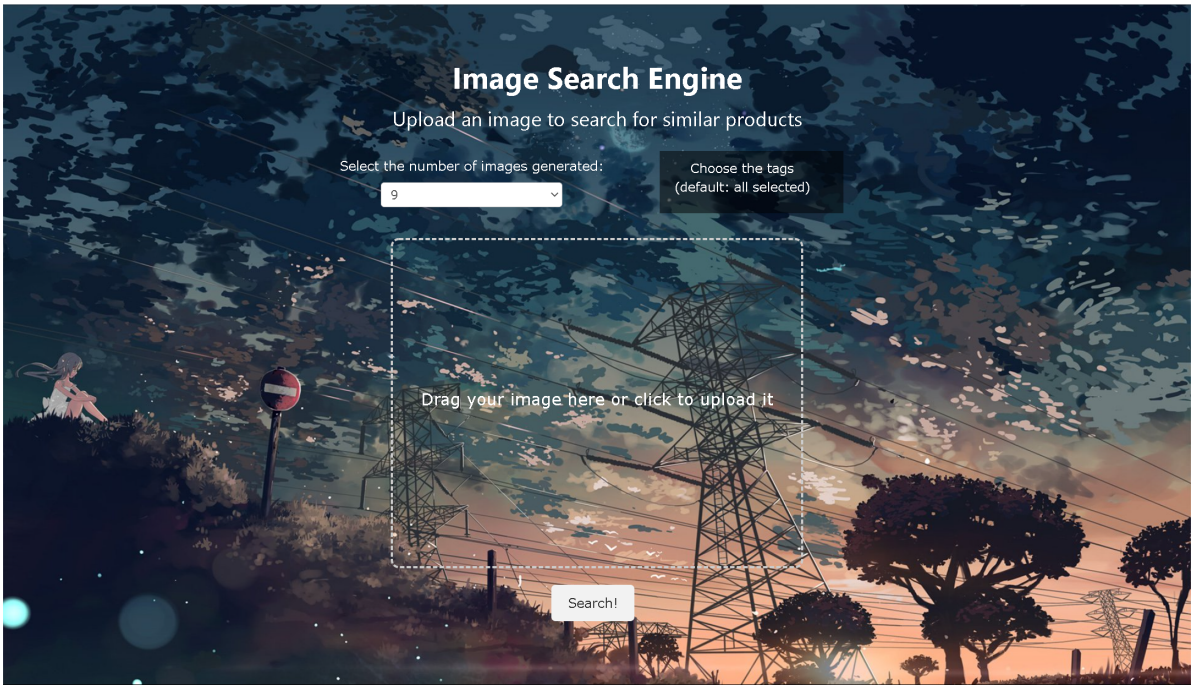
6. Clear and return, or more operations

After the first upload, users can continue to upload new images in the top text box for Baidu to identify, click on similar images or photo sources to see details, or enter more keywords in similar images to narrow the search scope, and so on.

II. My designs for five stages

1. Formulation

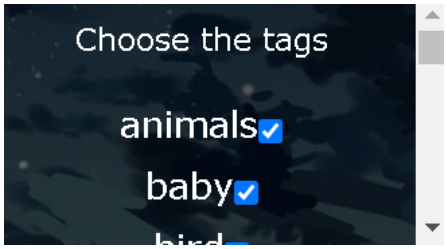
Home page:



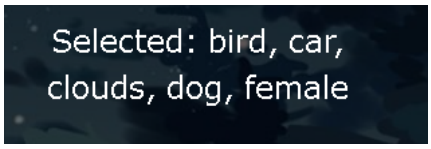
(1) Use tags to limit the search

Each image in the image set has a number of tags, and users can select or deselect certain tags to specify which image sets are available.

Click "Choose the tags" at the top right to open the multiplexer:



The system will check all checkboxes by default, and you can change them according to your needs. When the user clicks somewhere else, the drop-down box shrinks and shows all the selected tags:

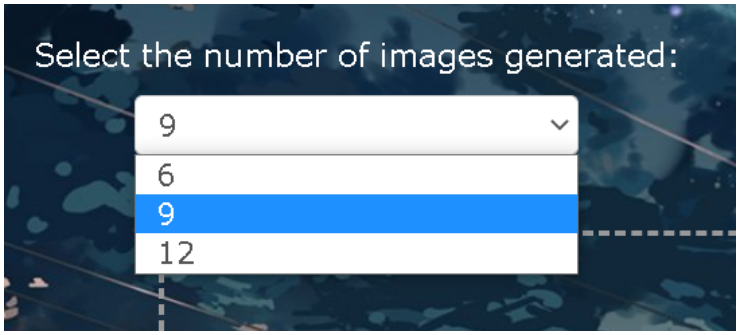


(2) Allow relaxation of search constraints

Users can click the dotted box in the middle of the page to select a file, or drag and drop the image directly into the box (I made the box large enough to reduce the possibility of users dragging in the wrong place). If the wrong image is uploaded, click or drag again to replace the image.

(3) Control the size of the initial result set

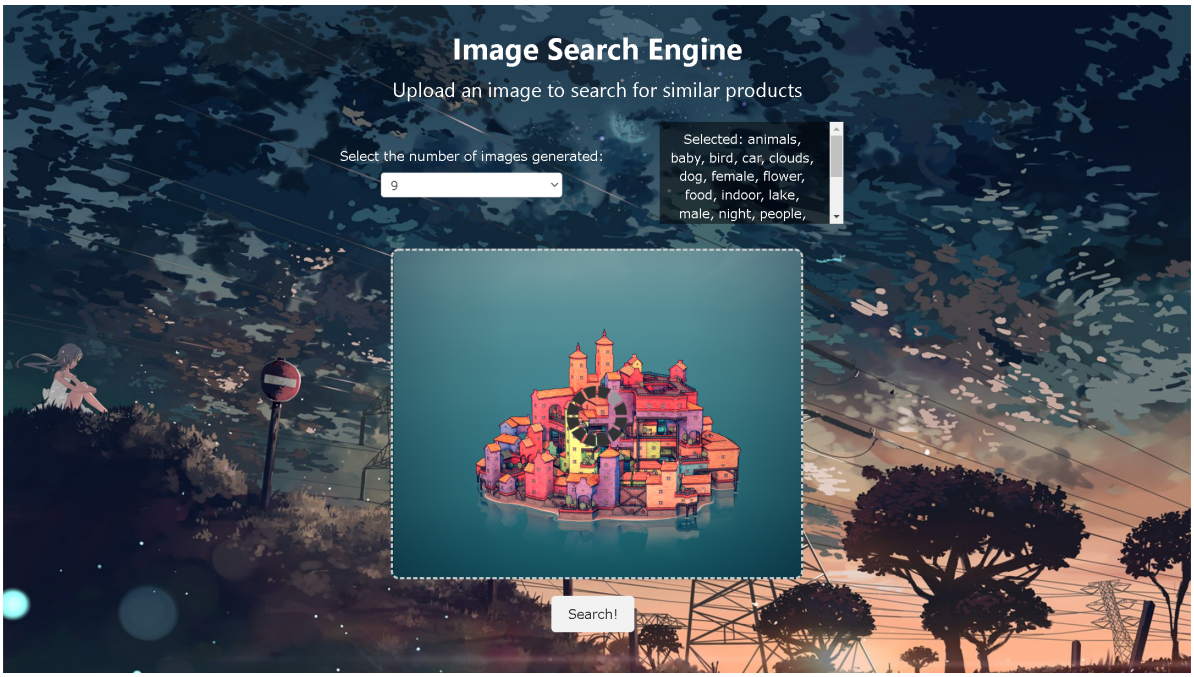
Users can control the number of images to be generated to 6, 9, or 12 using a drop-down selector.



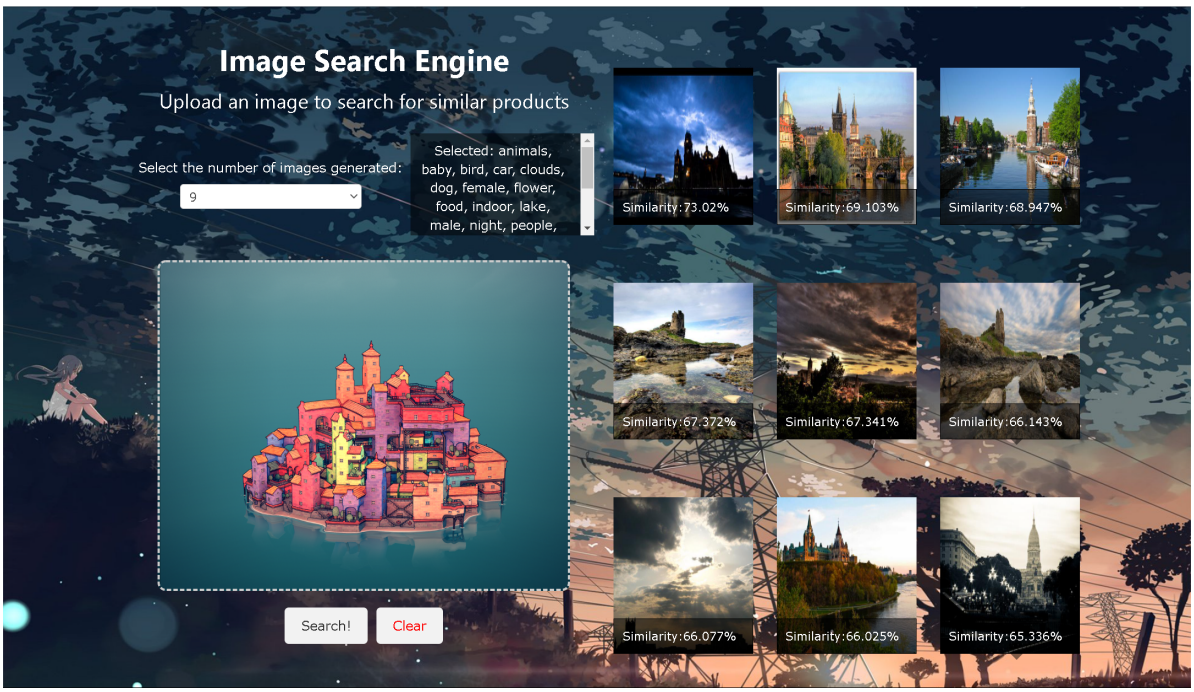
2. Initiation of action

(1) Click search, clear or other buttons for explicit actions

After uploading the image, click 'Search!' to start search (loading icon appears)

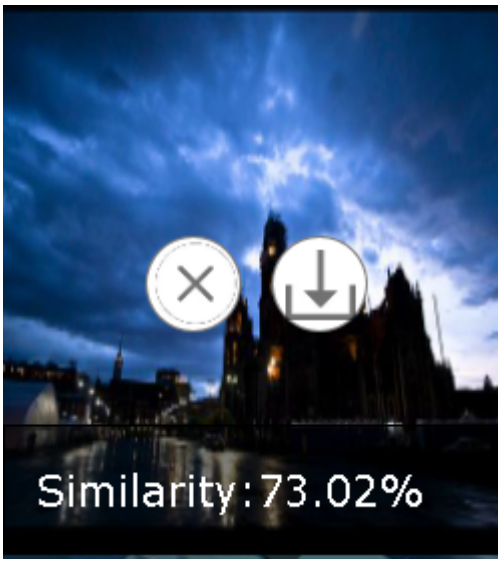


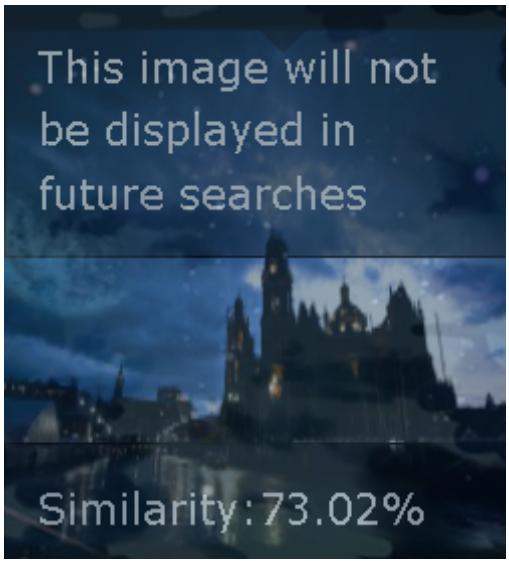
Once the search is complete, the user can continue to modify tags, the number of pictures, re-upload pictures or re-search, or click clear to return to the home page



(2) Click "dislike", and the backend will update the user data, which will take effect from the next search

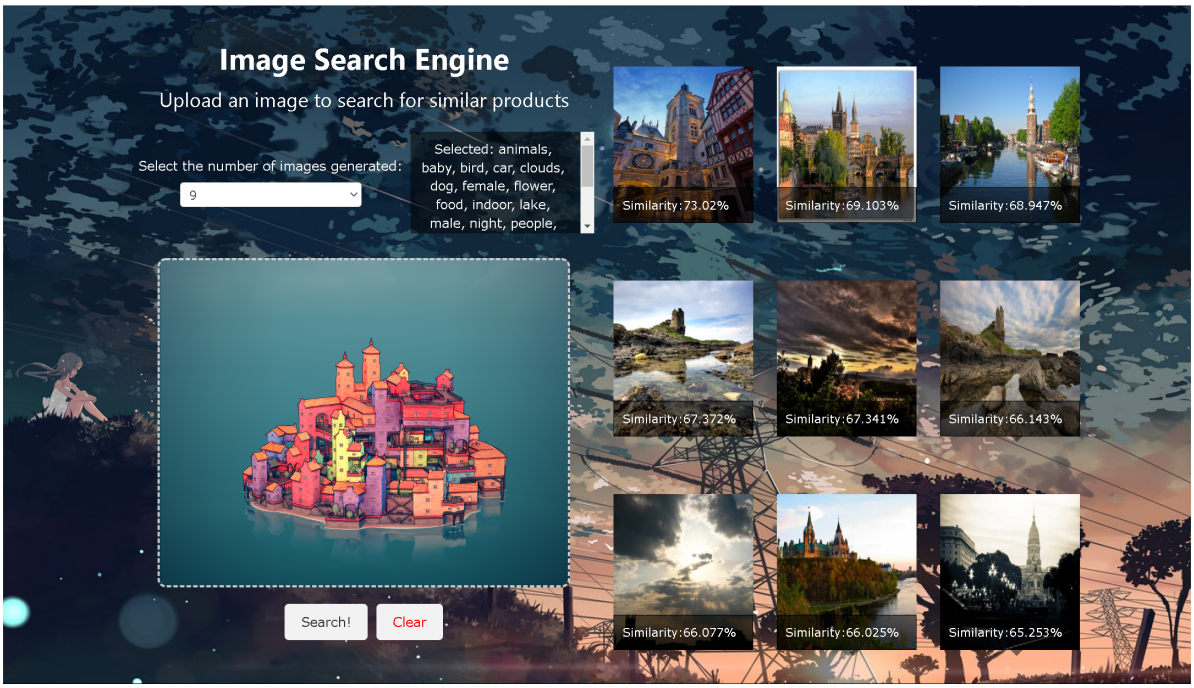
The "Dislike" and "Save" images appear when the mouse hovers over them. Users can click "Save" to download the image directly, and click "dislike" to get the text prompt shown in the right image and set the image to translucent:





After clicking 'Dislike', the backend will maintain a dislike.txt file to store the image number that the user doesn't like. In the next search, if the image is found, it will be skipped and not displayed again.

Click 'Search!' again, we can see that the image has been replaced with another image:



3. Review of results

(1) Keep search conditions and constraints visible

The user hits 'Search!' and the original image, radio, and multi-select boxes are still visible.

(2) Provides the similarity for each result item

Below each image, the similarity between the image and the original image is displayed.

The backend does the filtering of similar images in search.py by computing the cosine distance and selecting the top k items with the smallest distance. I go through the following equation

$$similarity\_percentage = (1 - cosine(image\_data, pred[neighbor])) * 100$$

to convert the cosine distance to a similarity percentage.

(3) Allow parameters to be readjust

Users can go ahead and resize tags, result sets, or change images, and click 'Search!' to search again.

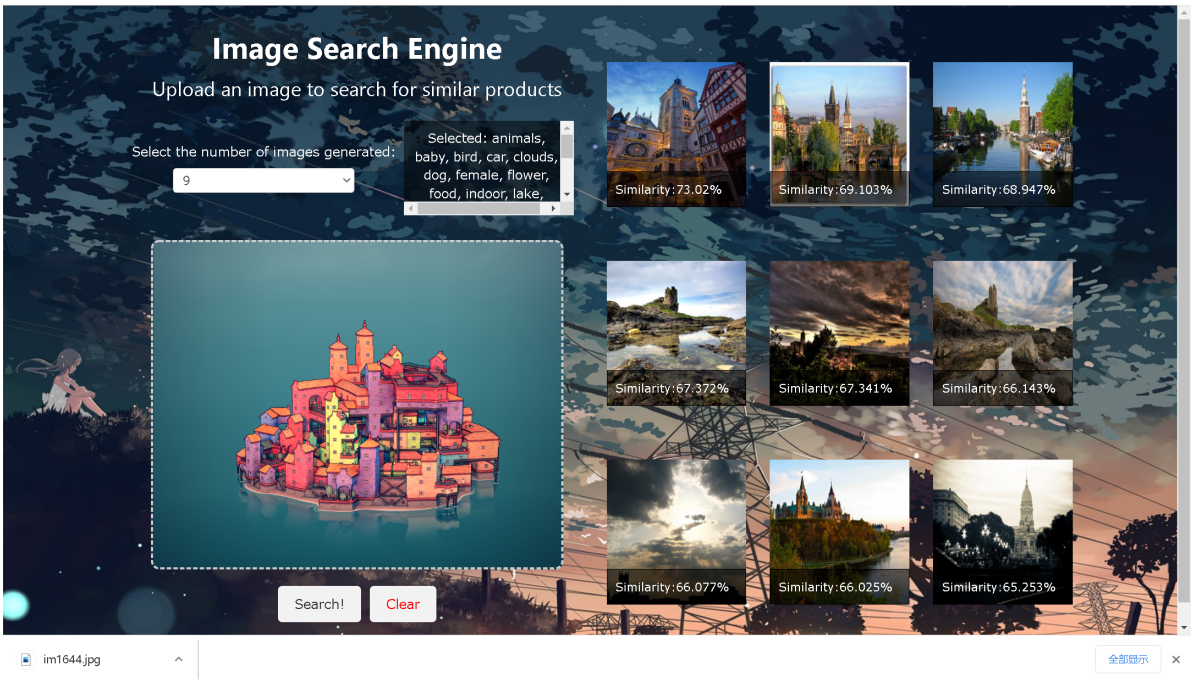
4. Refinement

As mentioned above, the user can be more specific about his or her needs by changing the tag, the result set size, choosing not to like the image, or simply changing the image.

5. Use

(1) Allow saving

Click "Save" to download the image



(2) Collect feedback

As shown above, the user can click on the "Dislike" icon to give feedback to the system. The system will no longer display images that the user does not like in future searches.