Chapter 6

Retrieval of Flower based on Sketches

6.1 Preamble

In this chapter, we propose a model for representation and indexing of flower images for the purpose of retrieving flowers of interest based a query sketch. In some applications, where the database is supposed to be very large, the retrieval process typically has an unacceptably a long response time. A solution to speed up the retrieval process is to design an indexing model prior to retrieval for reducing the number of flower hypotheses to be considered during searching by the retrieval algorithm. In this work, we study the suitability of Kd-tree indexing mechanism for flower retrieval system based on shape descriptors viz., Scale Invariant Feature Transform (SIFT), Histogram of Gradients (HOG) and Edge Orientation Histograms (EOH). To corroborate the efficacy of the proposed method, an experiment has been conducted on our own data set of 127 classes of flowers, containing 13169 samples. For querying the database we collected about 100 flower sketches from 20 different users. Experimental results reveal the success of the proposed method.

This chapter is organized as follows. The detail of the proposed method is given in section 6.2. In section 6.3, the process of sketch indexing is presented. In section 6.4 the details on experimental settings and performance analysis along with the results are summarized. Finally conclusions are drawn in section 6.5.

Some parts of the material of this chapter have appeared in the following research papers:

6.2 Proposed Model

The flowers of all classes are segmented using whorl based region merging segmentation. Form a segmented flower image Scale Invariant Feature Transform based features, Histogram of Gradients and Edge Orientation Histograms are extracted and stored in a database. However storing feature in a database in an efficient manner is required such that a list of possible flowers selected for matching process should be very minimum. Hence, there is a need of backend tool called indexing mechanism which stores the data in some predefined manner so that during matching only a few potential flowers can be considered. The multi-dimensional feature vectors obtained from sketch flowers are indexed using the Kd-tree. Top matches will be retrieved from the Kd-tree. The block diagram of the proposed flower retrieval system is given in Figure 6.1.

![Block diagram of the proposed flower indexing model](image)

Figure 6.1: Block diagram of the proposed flower indexing model
6.2.1 Features Descriptors

Different features are chosen to describe different properties of a sketch of a flower. As a sketch can be described using only shape features we choose well known descriptors viz, Scale Invariant Feature Transform (SIFT), Histogram of Gradients (HOG) and Edge Orientation Descriptors (EOH). The following section describes the use of these features separately.

6.2.1.1 SIFT Features Descriptors

The Scale-Invariant Feature Transform (SIFT) bundles a feature detector and a feature descriptor. Detailed discussion of SIFT features is given section 4.2.2.3 of chapter 4.

6.2.1.2 Histogram of Oriented Gradients (HOG)

Histogram of oriented gradients (Dalal and Triggs 2005) can be used as feature descriptors for the purpose of sketch flower retrieval, where the occurrences of gradient orientation in localized parts of a sketch flower image play important roles. The basic idea behind HOG is that the appearances and shapes of local regions within a flower can be well described by the distribution of intensity gradients as the votes for dominant edge directions. Such feature descriptor can be obtained by first dividing the flower into small contiguous regions of equal size, called cells, then collecting a histogram of gradient directions for the pixels within each cell, and finally combining all these histograms.

6.2.1.3 Edge Orientation Histograms (EOH)

Edge orientation histograms (Freeman and Roth, 1995; Levi and Weiss, 2004) are also interesting for our work, since flowers often present strong edges in the petals or whorl areas. They rely on the richness of edge information and maintain invariance properties to global illumination changes. The basic idea is to build a histogram with the directions of the gradients of the edges (borders or contours). It is possible to detect edges in a flower image but it in this work we are interested in the detection
of the angles. The sobel operators could give an idea of the strength of the gradient in five particular directions (Figure 6.2). The convolution against each of this mask produces a matrix of the same size of the original image indicating the gradient (strength) of the edge in any particular direction. It is possible to count the maximum gradient in the final matrix and use that to complete a histogram.

![Image of masks for 5 orientations in extracting EOH: vertical, horizontal, diagonals, non-directional and circle](image)

Figure 6.2: The masks for 5 orientations in extracting EOH: vertical, horizontal, diagonals, non-directional and circle

### 6.3 Sketch Indexing

Any classification process has a time requirement in proportional to the size of a database. That is as size of the database increases, the time taken by the classification process also increases. To overcome this, a retrieval system is proposed. Though the retrieval process takes less time compared to classification process, it is still dependent on the size of a database. Hence retrieval process also takes more time as size of the database increases. A solution to speed up the retrieval process is to design an indexing model prior to retrieval which reduces the number of potential candidates to be considered by the retrieval process. Here, in this work, we propose a Kd-tree based indexing mechanism for flower retrieval based on a query sketch of a flower.

In the proposed indexing model, the multi-dimensional feature vectors obtained from flower images of each individual are indexed separately using Kd-tree. The Kd-tree is one of the most prominent multidimensional space partitioning data structures (Dixit et al., 2008) for organizing points in a k-dimensional space and it is a useful data structure for searching based on a multidimensional key. The construction algorithm of Kd-tree is very similar to the planar case. At the root, we split the set of points into two subsets of roughly the same size by a hyper-plane perpendicular to the x1-axis. In
other words, at the root, the point set is partitioned based on the first coordinate of the points. At the children nodes of the root the partition is based on second coordinate and so on, until depth of \( k - 1 \) at which partition occurs based on last coordinate where \( k \) is the dimension of the feature space. After depth \( k \), again, partitioning is based on first coordinate. The recursion stops only when one point is left, which is then stored at the leaf. Because a \( k \)-dimensional Kd-tree for a set of \( n \) points is a binary tree with \( n \) leaves, it uses \( O(n) \) storage with construction time being \( O(n \log(n)) \). In addition to this, in Kd-tree there is no overlapping between nodes (Samet, 1994). Kd-tree is an appropriate data structure for flower retrieval system particularly in the analysis of execution of range search algorithm and it decreases the search time as it is supporting the range search with a good pruning. When query feature vector of multidimensional is given, range search is invoked using Kd-tree to retrieve top matches that lie within a certain distance (threshold) from the query. These top matches are subsequently used for flower retrieval.

### 6.4 Experimentation

In this section, we present the details of an experimentation conducted to demonstrate our proposed model on our own datasets. Experimentation is performed on Category-127 dataset and on a subset of the Flower Sketch dataset presented in chapter 2.

When the users are asked to rank the similar flower images that are retrieved for a given flower sketch query, they could get us marking only for a subset of the database. The subset has 20 users and for each user 5 flower sketches, totally about 100 sketches.

Given a flower sketch as a query, the first 10 similar flower images are retrieved by comparing with Category-127 dataset. To evaluate the results we asked user rank the top ten images for all the features and their combination for given flower sketch query, as its will more effective for the efficiency of the proposed method.
6.4.1 Experimental Results

In experimentation, we study average retrieval accuracy for all 20 users by considering top ten retrieved images. Figure 6.3 to 6.5 show the average retrieval accuracy of SIFT, HOG, EOH features for each user. Figure 6.6 to 6.8 show the average retrieval accuracy for combination of two features SIFT+HOG, SIFT+EOH, HOG+EOH for each user. Figure 6.9 shows the average retrieval accuracy of final combination SIFT+HOG+EOH for each user. The Table 6.1 shows the average retrieval accuracy for all the features along with the time taken by single query to retrieve. From table 6.1 we can observe that fusion of the all the features achieves a good average retrieval accuracy for all the users.

Further, we created a graphical user interface for sketch based retrieval system (Figure 6.10), in which users can input the sketches through MS-paint or can load from devices. In toolbox, the system displays only top ten images retrieved for the various combinations of features.

6.5 Conclusion

In this chapter, we proposed Kd-tree based indexing approach to index a huge dataset of flowers for given input sketch. In the proposed method we represent each flower by shape descriptors of SIFT, HOG, EOH. Experimentations are conducted on Category-127 dataset and Flower Sketch dataset to assess the advantage of using indexing technology. From Experimentation we can understand that the combination of all the features descriptors achieves a good accuracy with indexing approach.
Figure 6.3: Average retrieval accuracy obtained by the proposed indexing based on only SIFT features for 20 users

Figure 6.4: Average retrieval accuracy obtained by the proposed indexing based on only HOG features for 20 users
Figure 6.5: Average retrieval accuracy obtained by the proposed indexing based on only EOH features for 20 users.

Figure 6.6: Average retrieval accuracy obtained by the proposed indexing based on SIFT+HOG features for 20 users.
Figure 6.7: Average retrieval accuracy obtained by the proposed indexing based on SIFT+EOH features for 20 users

Figure 6.8: Average retrieval accuracy obtained by the proposed indexing based on HOG+EOH features for 20 users
Table 6.1: Average retrieval accuracy obtained for the proposed indexing scheme for various combinations of features along with time taken for a single query to retrieve

<table>
<thead>
<tr>
<th>Features</th>
<th>Accuracy</th>
<th>Time Taken to retrieve a single query in sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>52.3</td>
<td>0.0167</td>
</tr>
<tr>
<td>HOG</td>
<td>51.4</td>
<td>0.0164</td>
</tr>
<tr>
<td>EDH</td>
<td>49.7</td>
<td>0.0155</td>
</tr>
<tr>
<td>SIFT+HOG</td>
<td>61.2</td>
<td>0.0199</td>
</tr>
<tr>
<td>SIFT+EOH</td>
<td>58.2</td>
<td>0.0187</td>
</tr>
<tr>
<td>HOG+EOH</td>
<td>58.4</td>
<td>0.0243</td>
</tr>
<tr>
<td>SIFT+HOG+EOH</td>
<td><strong>78.8</strong></td>
<td>0.0285</td>
</tr>
</tbody>
</table>
Figure 6.10: Snapshot of the toolbox designed for sketch based flower retrieval system