

Sketch Recognition for Image Classification and Retrieval

Aishwarya M S

Department of Computer Science
BNM Institute of Technology
Bangalore, India

Dr. Vimuktha Evangeleen Salis

Department of Computer Science
BNM Institute of Technology
Bangalore, India

Arvindh K

Department of Computer Science
BNM Institute of Technology
Bangalore, India

Abstract— The approach employed here focuses on the sketches of objects. Here, the user interface asks for a sketch input from the user on which the search has to be performed. Then, the low-level features of the sketch such like boundary lines, outlines are extracted. The images from the database are also processed in the same way and the extracted details are stored in database. Once the database makes the classification, algorithms are applied to produce appropriate results. Here, some image processing and classification algorithms are used to match the input sketch and the training example sketches present in the database. Once the match is found, the related information and images is retrieved. The main objective of Sketch Recognition project is to bring in Human-Computer interaction. One should be able to draw any object, upload it to the recognition system and recognize and classify the objects easily. To improve retrieval accuracy through sketch-to-sketch comparison. “To develop an approach, to find the similarities between the natural images and sketch images, classify and retrieve them with efficient features.”

Keywords—Sketch Recognition, Image Processing, Block/Template Matching

I. INTRODUCTION

Image-based search is a valuable tool that is attached to the computer vision problems in object recognition and also classification, recently, ability to recognize real world objects using rough hand drawn sketches has been of particular interest.

Sketch recognition mainly focuses on matching the sketches of the objects with the image database present. Based on the result obtained we classify them and retrieve the appropriate images. In a system that performs image searches, it is desirable for both to find close image matches and to classify an input image. But extraction of distinct features from input query is quite challenging in object sketch recognition systems. More effort is needed to determine optimal feature form a certain set, based on attributes of the object that needs to be recognized and classifiers to be used. However, due to the ambiguity and limitations the procedure of object classifications has been recently over- powered by multiple approaches on neural networks such as deep learning [2].

The main advantage in deep neural network methodologies is that training process self- determines the

best features from data and comparative data set. The disadvantage is that the training procedure could be very time consuming. Neural network has provided promising performance in numerous image classification. However, recent research has demonstrated a imperious power of neural network method called as deep learning.

Sketch recognition is the automated recognition of object diagrams, by a computer. It focuses on matching the sketches of the objects with the image database present. Based on the result obtained we classify them and retrieve the appropriate images. The main objective is to extract a compressed and straight forward sketch from a query image. The feature extraction from a sketch is mainly established on histogram of orientation accession and sketches are represented by global/local captions.

Since sketch creation from natural images is a crucial part of sketch-based image retrieval (SBIR) [1] a saliency detection method is employed, which is based on enhanced Markov Chain. . The sketches are stored in database and compared with the input sketch obtained from the user interface. The system proposed focuses on the specific problems of matching sketched query images to a database of sketches and classifying those query images.

A user interface is provided for creating sketches and the related images are searched accordingly. The dataset consisted of a group of 20,000 human sketched image by Eitz et al.2012 collected from Amazon Mechanical Turk. The dataset included 250 categories of objects ranging from simple (e.g. apple, sun) to complex (e.g. clock etc). Each category has 80 images, and the images contain only the sketch lines of the object (i.e., no other scene context is contained within the sketches).

II. LITERATURE CITED

A. Photo to Sketch Transformation from a Complex Background.

Photo-to-Sketch Transformation, the main aspect is to

examine the problem related to sketch creations for sketch-based image retrieval (SBIR). Dominant feature extraction algorithm is alone incompetent without clarifying this problem to obtain exceptional retrieval outcome. Transposing images from basic pixels into pseudo-sketches plays a compelling role in SBIR. Other works are present on face-sketch synthesis [3]

based on recognitions. Wang et al. presented a transudative face sketch-photo synthesis method [4] that combines test samples during learning to obtain optimal performance, also proposed a Bayesian framework [5] for face sketch synthesis. Three main methodologies implemented are:

- Saliency detection.
- Gabor filter.
- Sobel operator.

Saliency detection is implemented to excerpt the dominant objects from an image. Then, Gabor filter is constructed in addition to apprehend the “real-major” object. The Sobel operator, is used to achieve the required pseudo-sketch. Dataset used here is a Flickr15k dataset which has a wide range of categories and has one free-hand sketch inside every folder(category) for better comparison and retrieval.

Therefore, performing all the methodologies on a particular input and conducting experiments over Flickr15k dataset to obtain pseudo-sketches that are reasonable and produces most advanced level results in certain category. Major advantage is that it enhanced random walk model depended on Markov-chain to achieve saliency map.

B. Object Recognition Using Deep Convolutional Method by a Recursive Network Structure.

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them Proposed system focuses on efficient feature extraction methodology by incorporating deep convolutional neural network trained on huge dataset. It is a combination of AlexNet and RNN [6] structure. This model involves many low level layer of the AlexNet trained upon the ImageNet dataset and the network was also trained using fixed/variable size RGB images from ImageNet.

This network layer is made of 8 layers where the first 5 are convolutional and the rest are fully connected layers. These layers together perform the action of feature extraction from the input image. Further, RNN unit consisting of various RNN structure connected to process the obtained features, before they are categorized by the Softmax classifier. The input data is segregated into divisions of equal proportions, calculate element wise product of every such division with a common array of weights, sum up the results in sync and process it through sigmoidal or similar squash functions.

Table 1: Performance evaluation

Method	Accuracy	Feature Size
AlexNet-RNN + Softmax	89.3 ± 1.6	32768
AlexNet-RNN + SVM	89.7 ± 1.7	32768

In case of RNN, set of weights are randomly assigned based on the input of data structure and it is treated as a constant throughout, thus avoiding the pre-training as in CNN. Unlike CNN, RNN makes use of only non-overlapping input patches and thus require lesser cost than CNN. The recursive

model of the RNN captures repetitive patterns from the input, whereas random weights and squash function benefit in distinguishing among them

C. Sketch2Tag: Automatic Hand-Drawn Sketch Recognition

This paper deals with a system called Sketch2Tag which can be used for hand drawn sketch recognition. It is a sketch recognition system, which aims at recognizing object that a child is capable of recognizing. This system provides the user with a query panel on which the sketch has to be drawn and then provides recognition results in real-time. The database used contains one million clip art images.

Sketch2Tag is a sketch recognition system which produces the results of the search in real-time. The sketch has to be drawn by the user on the query panel provided and then click on the search button in order to start the recognition process.

Once the process is completed the result page will contain the object name of the sketch and also the recommended tags and their corresponding probabilities.

A more effective sketch-based image search methodology is adapted which will helps systems discover the images which are visually similar to those hand drawn sketches contained in the database.

The shapes [7] of these images are compared to the shape of the sketch and the most representative tags will be derived and will be considered as the result for the hand drawn sketch.

Apart from sketch recognition, Sketch2Tag can be used to recommend the related tags along with corresponding probabilities so that the user’s search content can be narrowed down in the case of an ambiguous sketch.

D. Sketch-based Object Recognition

The equations are an exception to the prescribed specifications of this template This system aims at providing both closest match image results as well as ranked guesses at the object portrayed in a query sketch.

This system is provided with a user interface for creating sketches and performing searches based on those sketches. The upper-left section of the interface provides a canvas on which the user can draw the sketch which is the query image.

When the user performs a search, the right side of the window will be populated with the 30 closest matches from the dataset which contains the sketches. This system extracts features from a sketch by using HoG descriptors.

In order to perform image searches in a reasonable amount of time, the size of the image descriptions have to be reduced. To accomplish this, a bag of words model is used.

Once the HoG features for each training image are computed, a dictionary of code words are generated by aggregating the HoG descriptor using k-means algorithm. The HoG features of a query image are matched to a dictionary of code words.

Below the canvas, the top five most likely object classes are listed in decreasing order from top to bottom.

The advantages of this system are incremental sketch editing and quantitative analysis. The major disadvantage of

this system is that the orientation and perspective of a sketch may cause problems in recognition since the algorithm used is not invariant to rotation.

The limitations also include inter-class ambiguity and intra-class ambiguity.

E. Parsing Methodology for Sketch Recognition Systems.

This paper presents a structure for modeling a sketch language and also generating a good parser recognition system.

The parsing method mentioned here throws light two issues namely; stroke gathering and ambiguity resolution. The paper addresses these issues. Pattern matching can help in recognition of a sketch up to an extent.

However, it does not work well with accepting incorrect patterns. Sketches may vary from person to person which causes drawing ambiguity, and this has to be treated well to achieve a good approach to sketch recognition.

Sketch recognition deals with two major T's, identification of patterns and interpretation of them. Context of a sketch or a stroke plays an important role in interpretation and resolving ambiguity.

The direct extension of a context-free string grammar is represented by XPGs, where generic relations are allowed. This idea of formalism helps in overcoming the inefficiency of visual language parsing techniques and algorithms.

In particular, the XPG formalism is completely based on one of the extensions of LR parsing called as XpLR methodology [8]. This strictness interprets a sentence to be set of symbols with some attribute.

The entire approach is integrated into a Sketch-Bench which consists of a symbol rewriter, production symbol database. Sketch-Bench also works with an incremental parser and a sketch editor to achieve the goal of sketch recognition

III. PROPOSED SYSTEM

The proposed system takes the input sketch from users and extracts the low-level features of this input image. Similarly, the low-level features of all images from the dataset are extracted along with storing. These features are compared with extracted features of the input query to retrieve the appropriate results.

The main distinction between the existing and the proposed systems is that the images in the dataset are first converted into sketches and the low-level features containing these converted sketches extracted to improve the search results.

The detailed working of the proposed system is as follows:

Step 1: The colored training images as well as the sketch training images are contained in two different folders on the system.



Figure 1: Training images

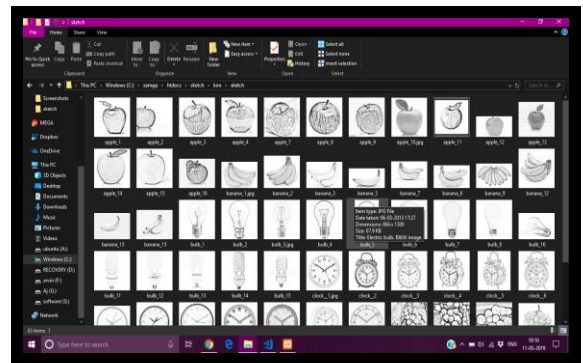


Figure 2: Sketch Training images

Step 2: Python programs for extracting the Color, Edge and Contour of the training images are executed and the values obtained are stored in a csv file under respective columns.

	A	B	C	D	E	F	G	H	I	J	K
1	Color	Edges	Contours	Class							
2	0.018229	84.884	106	apple							
3	0.019052	37.07133	136.9872	grape							
4	0.016858	4.880667	78.47368	apple							
5	0.018329	35.38267	95	apple							
6	0.017734	26.51333	108.5	apple							
7	0.017004	10.10933	109	apple							
8	0.018229	84.884	106	apple							
9	0.017938	29.97667	94.875	apple							
10	0.018558	44.064	92	apple							
11	0.017564	19.244	74.5	apple							
12	0.018182	28.50333	113.8333	banana							
13	0.018085	30.498	144.2	banana							
14	0.020998	58.93333	75.57143	banana							
15	0.018421	36.38	117.5	banana							
16	0.017895	26.06667	88	banana							
17	0.019378	32.844	125.5	banana							
18	0.018312	31.20067	93.92857	banana							
19	0.017566	14.994	140.625	banana							
20	0.018572	31.484	126	banana							
21	0.019879	59.98733	83.375	banana							
22	0.018173	31.39333	129.75	grape							
23	0.022822	80.172	84.75	grape							

Figure 3: Features of the training sketches in a csv file

Step 3: A user interface is provided to the user to choose the preferred algorithm to perform classification. The options provided are KNN and SIFT.

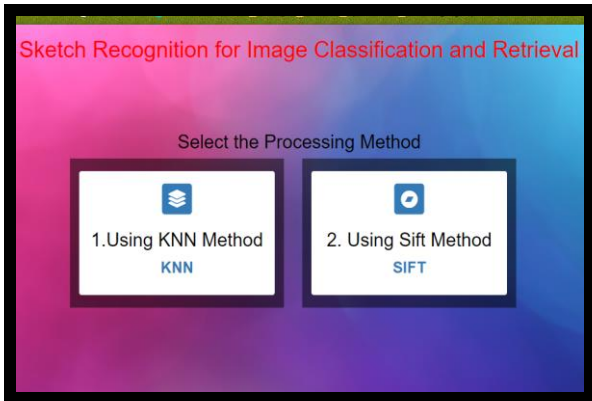


Figure 4: User Interface

Step 4: Once the user chooses the algorithm, the user can upload the sketch image as input.

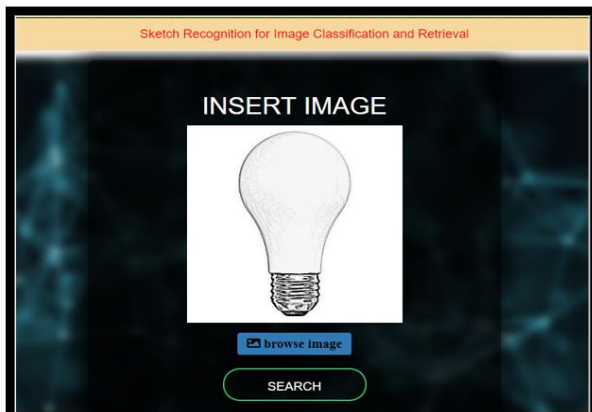


Figure 5: Uploading sketch

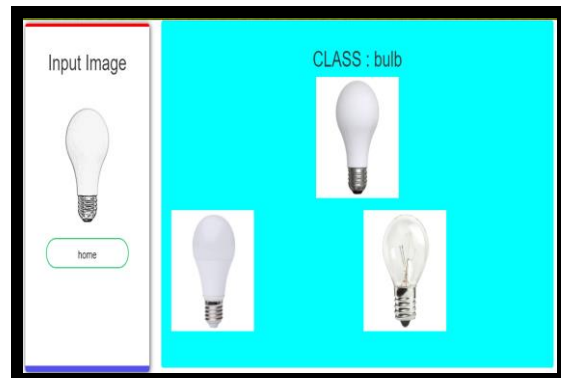
Step 5: The python programs run and extract the features containing this input query and stores values in a query image csv file.

	A	B	C	D	E	F	G	H
1	Color	Edge	Contour					
2	0.01875	11.68467	217.4782					
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								

Figure 6: Features of the input image in a csv file

Step 6: The query image csv file and the training dataset csv file are given to the respective algorithm for classification.

Step 7: Once the algorithm is done classifying the images, it maps the sketch images to the colored images, retrieves the colored images from the folder and displays 3 related images to the user.



IV. METHODOLOGY

There are a wide range of methodologies that can be used in order to design a system. The most important aspects when it comes to selecting a methodology for a project are:

- The accuracy
- The complexity
- The implement ability

These are the aspects that need to be taken into consideration while selecting an ideal way in which a project can be implemented.

The design of the system must be done in such a way that all the above-mentioned conditions are taken into account and are used in the best possible way. This section discusses about the various algorithms that are used, and the reason behind using them.

A. System Design

The physical design is related to the original input and output processes of the proposed system. This can be said in terms of how data is given into a system and how it is verified, how it may be processed, and how it is represented/displayed. The physical portion of system design can be broken down into three sub tasks:

- User interface
- Data design
- Process design

User Interface design is troubled with how user adds data to the system and with how the system displaces data back to them. Data design is troubled with how the data is presented and stored in the system. The Process design is troubled with how data moves throughout the system, and with how and where it is verified, secured and/or transformed as it flows into the system. Documentation outlines the three sub tasks produced and made accessible in the next phase. There are multiple solutions to a problem. The objective is to find the best and the most feasible one. Some of the methods or algorithms used for the design of the system are: Feature

extraction, KNN algorithm, chi-squared distance, Scale Invariant Feature Transform (SIFT).

B. Algorithms/Methods used

Feature extraction:

Feature extraction is a localization process, where an original set of raw variables is decreased to more convenient group of features for processing, while maintaining accuracy and meaning the original data.

When the input data is too huge to be processed and it is expected to be redundant (e.g. the repetitiveness of image in form of pixels), then it can be transformed into a decreased set of features, called a feature vector. Determining the subset of the starting feature is known as feature selection. The selected features are assumed to contain the appropriate data from the query data, so that the expected task can be performed by using this decreased presentation instead of the complete initial data.

- Edge detection: Canny edge detection is a method to filter out useful structural data/details from various objects and gradually decrease the amount of information to be processed. It has been applied in multiple computer vision systems.

cv2.Canny(img, minVal, maxVal)

img: Path of the image,

minVal: intensity gradient value, below which represents non-edges

maxVal: intensity gradient value, above which represents edges

The steps followed by Canny edge detection algorithm is as follows:

- Apply Gaussian filter to smoothen the image to eliminate the noise:

As most edge detection outcomes are easily influenced by noise present in image, it is important to filter out the noise to avoid wrong detection caused by noise. To smoothen the image, Gaussian filter is used to convolve with the image. This will slightly smoothen the image to prevent the effects of general noise on the edge detector. The Gaussian filter equation for a kernel of size $(2k+1) \times (2k+1)$ is given by

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k+1) \dots \text{Equation 5.1}$$

- Finding intensity gradients in the image:

An edge present in the image may indicate to a collection of directions, therefore the Canny method uses 4 filters to recognize horizontal, vertical and diagonal edges in the blurred image. The edge detection operator like Roberts or Sobel gives a value for the initial derivative in the horizontal and vertical direction as (G_x) and (G_y) respectively. The edge gradient and direction can be shown as:

$$G = \sqrt{G_x^2 + G_y^2} \dots \text{Equation 5.2}$$

$$\Theta = \text{atan2}(G_y, G_x),$$

- Trace the edge by hysteresis:

The tracking down of edges by restraining all the other edges that may be weak and not connected to strong edges, are given in Fig. 5.1. To trace the edge connection, blob analysis is implemented by looking for the weak edge pixel and its connected neighborhood pixels. As far as there is at least one strong edge pixel that is included in the blob, that weak edge point can be focused as one that should be conserved.



- Color histogram:

It represents the distribution of the proportion of colors in the picture. It shows various types of colors present and the number of overall pixels in each type of colors obtained. The alliance between a color histogram and that of luminance histogram is that a color histogram can also be articulated as “Three Luminance Histograms”, each one shows the brightness distribution of individual Red-Green-Blue color channel.

cv2.CalcHist(img, hist, accumulate, mask) img: Source image with same depth and same size.

hist: Output histogram.

accumulate: If the histogram is not that clear in the initial stage when it is allocated. If the matrix is not null, it should be an 8-bit array of the same size as image. The non-zero mask elements indicate the array elements counted in the histogram. Color histogram can be constructed for any kind of color space, even if the term is more often used for 3- dimensional spaces like RGB or HSV.

When monochromatic images are considered, the term intensity histogram may be used. When multi-spectral images are considered, where each pixel is presented by an arbitrary number for measurements, the color histogram is N-dimensional, with N being the number of readings taken. If set of visible color values is sufficiently less, each of the colors may be replaced on a range by itself; then the histogram is just the count of pixels that have possible color. More often, the space is separated into an appropriate number of ranges, mostly arranged as a regular grid, each consisting many similar color values.

The color histogram can be represented and shown as a smoothen function defined above the color space that approximates the overall pixel counts. Similar to other kinds of histograms, the color histogram is a statistic that can be seen as an approximation of a continuous distribution of colors attributes. They are flexible constructs that can be constructed from pictures in various color spaces, or even RGB or any color space of any dimension other than that.

For example, a Red-Blue chromaticity histogram are formed by normalizing color pixel attributes by dividing RGB

values as a sum of R+G+B, later quantizing the normalized R and B co-ordinates into the N bins each. A 2-dimensional histogram of Red-Blue chromaticity sliced into four bins (N=4) might yield a histogram as shown in table:

		red			
		0-63	64-127	128-191	192-255
blue	0-63	43	78	18	0
	64-127	45	67	33	2
	128-191	127	58	25	8
	192-255	140	47	47	13

➤ **Contours:**

It can be expressed as simply as a curve connecting all the continuous points along the edges/boundary, containing similar color or intensity. Contours are a useful tool for structure analysis, object recognition.

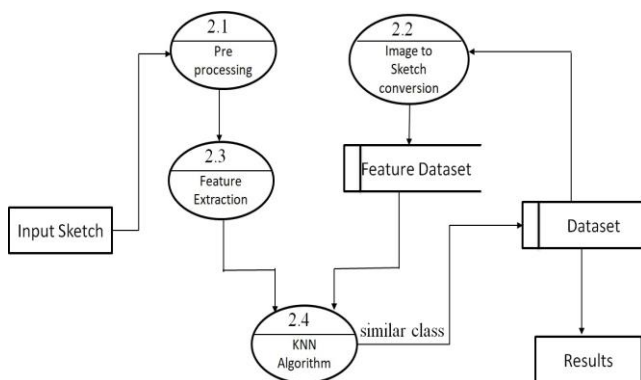
- ❖ Binary pictures can be used for better accuracy.
 - ❖ Obtained Contours function alters the source image. If original picture is required later, store it somewhere else by hand.
 - ❖ In OpenCV, finding contours is similar to finding white element from dark/black background. Therefore, object that need to be found should be white and the background should be dark/black.
- cv2.findContours(img,retrieval_mode,approx)**

img: Source picture,

retrieval_mode: retrieval contour mode,

approx: it's the approximation technique use.

cv2.drawContours() is implemented/used to draw contours. It's used to draw any shape given you have its edge/boundary points. Its arguments are source picture/image, contours that should be passed as a Python list, contours index.



KNN Algorithm

The output of KNN depends on whether it is used for classification or regression:

- ❖ In KNN classification, the result is a group(class) membership. An object is categorized by a

plurality vote of its neighbors, with object being appointed to the category most common among the k nearest neighbors

- ❖ In KNN regression, the result is the property value for the object. Value is the average of the values among k nearest neighbors.

The training data are vectors in a multidimensional space, each one with a category label. The training part of the algorithm contains only the storing of the feature vectors and category labels of the training data. A generally used distance metric is Euclidean distance. For text categorization the metric used is the overlap metric or Hamming distance. In context expression micro array data, for instance the KNN has been employed with correlation coefficients, such as individual and Spearman, as a metric. Frequently, the categorization accuracy of KNN can be improved automatically if the distance metric is learned with specially designed algorithms like Large Margin Nearest Neighbor or Neighborhood components analysis.

The drawback in regard to the basic "majority voting" categorization occurs when the class distribution is skewed. One way to overcome the problem is to weight the categorization, taking into concern the distance from the test-point to each of its k nearest neighbors. The category of each of the k nearest points is the product of the weight proportional to the inverse of distance from that point to the test-point. Another alternative to overcome skew is by abstraction in data presentation. Fig. 5.2 is an example of 2 classes, represented graphically using KNN.

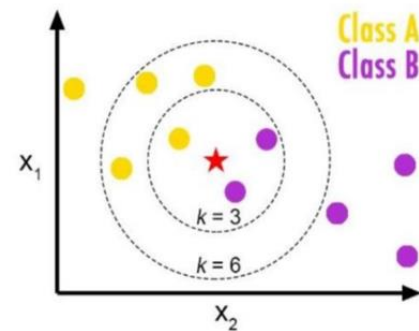


Fig 5.2: Graphical representation of KNN Algorithm

Scale Invariant Feature Transform (SIFT):

The SIFT is a feature detection method in computer vision as to detect and illustrates local features in images. Applications contains object re-cognition, robotic mappings and guiding, image stitching's, 3D structuring, gesture re-cognition, video tracing, separate identification of wildlife and match movements. SIFT key points for objects are first obtained from sets of resource images and stored. An object is recognized in a new query by separately comparing each feature from the new query to database along with finding the candidate identical features based on Euclidean distance.

Subsets of key points that accept on objects and its location, scale, orientation in the new query image are identified to filter out the good results. The assurance of consistent clusters is performed rapidly by an efficient hashing table implementation of the generalized Hough transformation. Each cluster of 2 or more features that accept

on objects along with its pose is there after subject to further detailed model verification and subsequently outliers are deleted. At last, the probability that particular set of features point out the presence of an object is calculated, given the accuracy of fit and also number of probable false results. Object results that qualify all the tests can be identified as correct with high confidence.

❖ Scale-invariant feature detection:

Lowe’s method for image feature generation transforms an image into a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion.

❖ Feature matching and indexing:

Indexing consists of storing SIFT keys and identifying matching keys from the new image. Lowe used a modification of the k-d tree algorithm called the best-bin-first search method that can identify the nearest neighbors with high probability using only a limited amount of computation. Each of the SIFT key points specifies 2D location, scale, and orientation, and each matched key point in the database has a record of its parameters relative to the training image in which it was found. The similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object and also does not account for any non-rigid deformations.

As shown in Fig 5.3, scale space extreme is detected. SIFT algorithm removes low- contrast key points (middle image), there after filters out located on edges. Result of key points is shown (last image).



Fig 5.3 SIFT key points

The features of the various images stored in the dataset can be extracted using the various feature extraction techniques mentioned here. Once the feature vector is obtained, the mean values are extracted and stored in the .csv file, along with the path where the image is stored in the system. In the same way, the mean feature values of the input image are also extracted. Once the .csv files created, K-NN algorithm are applied to detect the class to which the input image belongs to. If K=3, the top 3 matches are displayed and if K=5 top 5 matches are displayed and so on. SIFT algorithm directly extracts features as key points and descriptors and stores it. It matches these

key points with the input image to show appropriate output images.

V. RESULT ANALYSIS

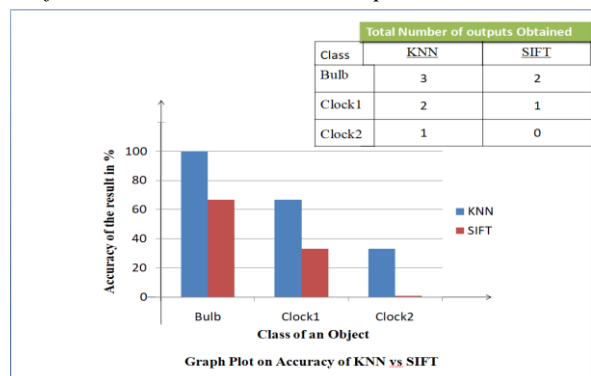
A. KNN

Test ID	Test input	Excepted result	Actual Result	Remarks
T_01	Clock_1	Images of at least two clocks.	Three clocks. So class is defined as clock	Pass
T_02	Clock_2	Images of at least two clocks.	One clock. So class is undefined	Fail
T_03	Bulb_1	Images of at least two bulbs.	Three bulbs. So class is defined as bulb.	Pass
T_04	Apple_1	Images of at least two apples.	Two apples. So class is defined as apple.	Pass
T_05	Apple_2	Images of at least two apples.	Cherry, grapes and an apple. So class is undefined.	Fail
T_06	Apple_3	Images of at least two apples.	Two bananas and an apple. So class is defined as banana.	Fail

B. SIFT

Test ID	Test input	Excepted result	Actual Result	Remarks
T_01	Clock_1	Images of at least one clock.	Two clocks.	Pass
T_02	Glass_1	Images of at least one glass.	Two clock.	Pass
T_03	Box_1	Images of at least one box.	One box.	Pass
T_04	Pisa_1	Images of at least one pisa tower.	Images of different objects.	Fail

C. Performance Evaluation and Comparison



The performance evaluation metrics show that the KNN algorithm has an accuracy rate of 77% for the considered dataset while SIFT has an accuracy of 51%. Hence KNN performs better, with higher accuracy and speed compared to the SIFT algorithm. Moreover the KNN algorithm predicts the class to which a particular object might belong, but SIFT just provides the names of each individual result making the object recognition task difficult for the user.

CONCLUSION

The project, Sketch Recognition for Image Classification and Retrieval, is developed with the objective of recognizing hand drawn sketches and retrieving related images of that hand drawn object or sketch. The main motive behind the project is to improve human computer interaction by developing an approach to help computers identify or classify a simple sketch drawn by a human being. A good user interface is provided to the user to upload a sketch input image and receive related images. At the back end of the project various python codes are run in order to extract the low-level features of the query image as well as the training images which are maintained in the system. The two main algorithms employed for classification of images are KNN and SIFT. The performance evaluation conducted after the completion of the project, for the respective algorithms, show that the KNN algorithm works best for classification. It has an accuracy rate of 100% for most of the images among all the dataset images. This helps in concluding that the project employees' two methods or algorithms to serve the purpose of image classification, although it works best when the KNN algorithm is chosen.

REFERENCES

- [1] M. Eitz, J. Hays, and M. Alexa, "How do humans sketch objects?" ACM Trans. Graph., vol. 31, no. 4, pp. 1–10, Jul. 2012. [2] Photo-to-Sketch Transformation in a Complex Background by XIANLIN ZHANG1, XUEMING LI2, SHUXIN OUYANG1, AND YANG LIU31 Institute of Information and Communication, Beijing University of Posts and Telecommunications, Beijing 100089, China.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436–444, May 2015.
- [3] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "A comprehensive survey to face hallucination," Int. J. Comput. Vis., vol. 106, no. 1, pp. 9–30, 2014.
- [4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740-741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982 N. Wang, D. Tao, X. Gao, X. Li, and J. Li, "Transductive face sketchphoto synthesis," IEEE Trans. Neural Netw. Learn. Syst., vol. 24, no. 9, pp. 1364–1376, Sep. 2013
- [5] N. Wang, X. Gao, L. Sun, and J. Li, "Bayesian face sketch synthesis," IEEE Trans. Image Process., vol. 26, no. 3, pp. 1264–1274, Mar. 2017.
- [6] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in Proc. 28th Int. Conf. Mach. Learn. (ICML), 2011, pp. 129–136.
- [7] Z. Sun, C. Wang, L. Zhang, and L. Zhang. Query-adaptive shape topic mining for hand-drawn sketch recognition. In ACM Multimedia, 2012.
- [8] G. Costagliola and G. Polese, "Extended Positional Grammars", in Proceedings of 2000 IEEE Symposium on Visual Languages, Seattle, WA, USA, September 10-13, 2000, pp. 103-110.