Improving the Online Smart CBIR Interface, to Make it Work in Auto Mode

Vikram Sunil Bajaj
Summer Intern
Visualization and Perception Lab
Department of Computer Science and Engineering
IIT Madras

Abstract

Content-Based Image Retrieval (CBIR), also known as Query By Image Content (QBIC), is the application of Computer Vision to the image retrieval problem, that is, the problem of searching for digital images in large databases.

The proposed CBIR system takes as input a query image (via webcam capture or image upload), and retrieves the top six most similar images from a gallery of image samples.

I. Introduction

THE need to find a desired image from a collection is shared by many professional groups, including journalists, design engineers and art historians. This need can be fulfilled by CBIR systems.

Content-Based Image Retrieval (CBIR) consists of retrieving the most visually similar images to a given query image from a database of images.

"Content-based" means that the search will analyze the actual contents of the image, as opposed to the traditional "Concept-based" image retrieval techniques that rely purely on metadata such as captions or keywords.

Using textual descriptors is intuitive and simple, but isn't very efficient. This is because having humans manually annotate images by entering keywords or metadata in a large database can be time consuming and may not capture the keywords desired to describe the image. Hence, CBIR systems are in demand.

The aim of this project is to create a Content-Based Image Retrieval system and a GUI to demonstrate its working. The system should also be made to run in an Auto mode (i.e. It should run from start to end repeatedly, without human intervention).

A. Common Applications of CBIR Systems

CBIR systems have applications spanning across various fields of work. Some of them include:

- Fingerprint identification
- Architectural and engineering design
- Crime prevention
- Military
- Fashion industry

B. Some Commercially Available CBIR Systems

The earliest commercial CBIR system was developed by IBM and was called QBIC (Query by Image Content).
A few others include:

- Google's Image Search
- Virages VIR Image Engine
- Excaliburs Image RetrievalWare
- Chic Engine: fashion search engine
- ID My Pill

II. ALGORITHMIC DESCRIPTION

The CBIR system has two phases:

- **Offline Phase**: This phase involves the extraction of features from images in the dataset, and the training of a classifier using the extracted features. These features are even stored in a categorized feature database. This phase also involves the training of an object detection module using Discriminatively Trained Part Based Models (DPM) [5].
- **Online Phase**: This phase involves the acquisition of a query image from the user, with the aim of classifying it into one of several predefined categories, and then retrieving the top six rank-ordered images from the gallery using a similarity estimation measure.

A. The Offline Phase

The offline phase involves three tasks:

- Feature Extraction
- Training the Recognition Module
- Training the Detection Module

These tasks have to be performed in order to train the CBIR system to recognize a set of objects, and need not be performed once the system is ready to be used.

1) Feature Extraction:
The dataset used contains 28 categories of objects, with about 30 images in each category.

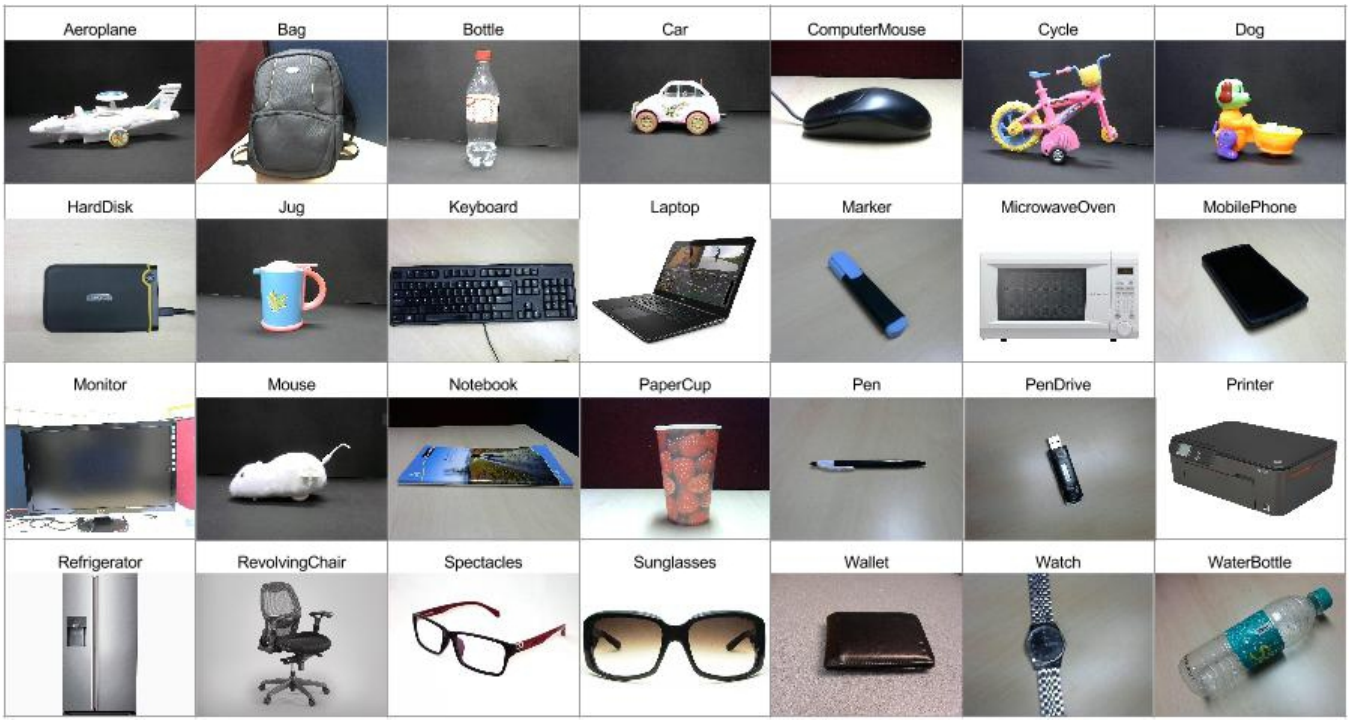

Fig. 1: CBIR Dataset.

Feature extraction is the heart of content-based image retrieval. Instead of using the whole image, only an expressive representation of the most significant information should be extracted. The process of finding this expressive representation is known as **feature extraction** and the resulting representation is called the **feature vector**.

There are several local feature descriptors that can be extracted from images. A few of them include:

- Speeded Up Robust Features (SURF) [10]
- Scale Invariant Feature Transform (SIFT)
- Histogram of Oriented Gradients (HOG) [7]
- Gradient Location and Orientation Histogram (GLOH), and so on.

For this project, **SURF** features were extracted from the training images. SURF is a robust image detector and descriptor (Bay et al., 2006). Analysis shows it is 3 times faster than SIFT while performance is comparable to SIFT. SURF is good at handling images with blurring and rotation, but not good at handling viewpoint change and illumination change. SURF detector is based on Hessian matrix measures and uses 2D Haar wavelet transform for descriptor, employing only 64 dimensions, which leads to fast feature computation and matching. Its dimensions can be increased to 128 and it has been proved (Juan et al., 2009) that doing so does not affect the speed much. SURF sometimes provides with more than 10% improvement compared to several other descriptors (Bay et al., 2006).
A **Bag of Words (BoW)** [11], or Bag of Visual Words model was then used to represent image features as words. The BoW model is based on the frequency of occurrence of a word in a document. In the context of CBIR, these words (or *codewords*)

refer to the representative of similar patches in an image. These codewords combine to form a *codebook*. To identify codewords, **K Means Clustering** is performed over the feature vectors obtained from applying SURF. The centroids of the formed clusters are taken as the codewords and the number of clusters gives the size of the codebook (The size of the codebook created in this project is 500). A histogram of these codewords summarizes the content of the image and serves as the feature vector for that image. These feature vectors were extracted from all the training images and were used to train the classifier in the recognition module.
Also, HOG features were extracted from all the images and a collection of these feature vectors formed the **categorized feature database**.

2) Training the Recognition Module:
In Machine Learning, **Support Vector Machines** (SVMs) [6] are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, with their corresponding class labels, an SVM training algorithm builds a model that assigns new examples into one of the classes.
The classifier used for this project was a multi-class SVM with a Gaussian (RBF i.e. Radial basis function) kernel. It showed 77% classification accuracy.
The following bar graph denotes the category-wise accuracies achieved by the recognition module.

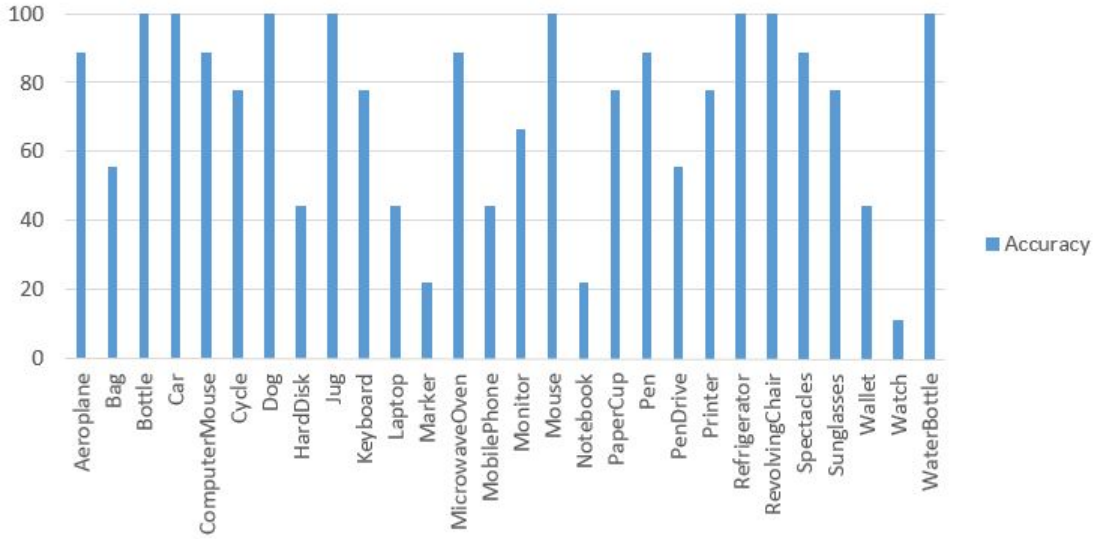

Fig. 2: Category-wise accuracies.

3) Training the Detection Module:
The object detection system is based on mixtures of multi-scale deformable part models. These models are trained using a discriminative procedure that only requires bounding boxes for the objects in a set of images. Pictorial structures represent objects by a collection of parts arranged in a deformable configuration. Each part captures local appearance properties of an object while the deformable configuration is characterized by spring-like connections between certain pairs of parts. Deformable part models such as pictorial structures provide an elegant framework for object detection.

B. The Online Phase

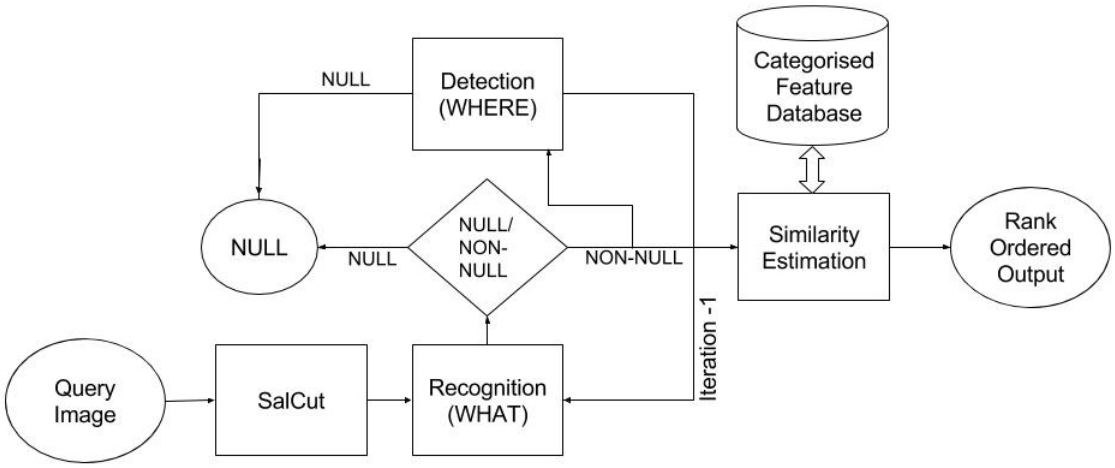

Fig. 3: The Online Phase.

The online phase is where the system takes a query image and tries to retrieve the top 6 most similar images from the gallery of images in the dataset.
This phase has five tasks:

- Acquiring the Query Image
- Salient Object Detection
- Iterative Feedback Cycle [8], [9]
- Similarity Estimation
- Displaying the Rank-Ordered Output

1) Acquiring the Query Image:
A MATLAB GUI was built to acquire the query image. The image can be acquired either through a webcam or by uploading it. The following figures show the GUI created using MATLAB, and the two modes of image acquisition using the GUI.

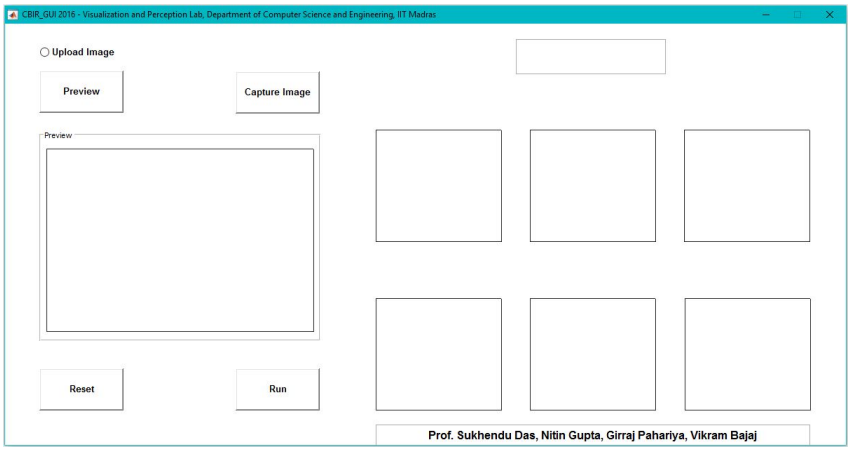

Fig. 4: CBIR GUI.

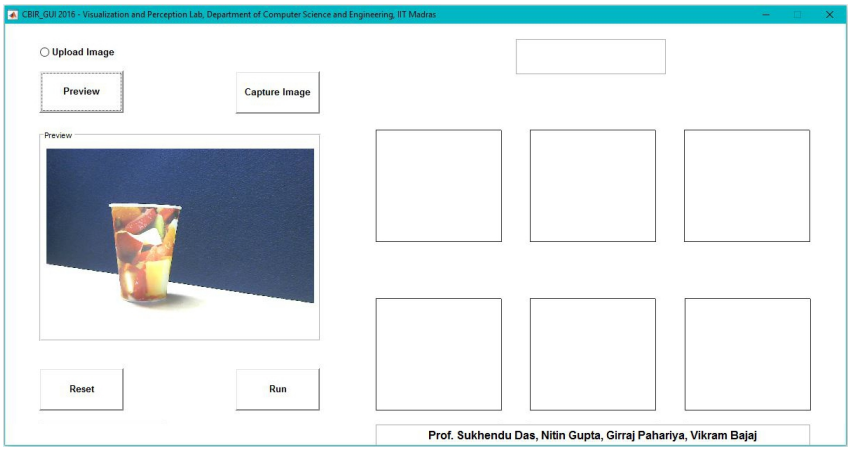

Fig. 5: Image Acquisition by Webcam.

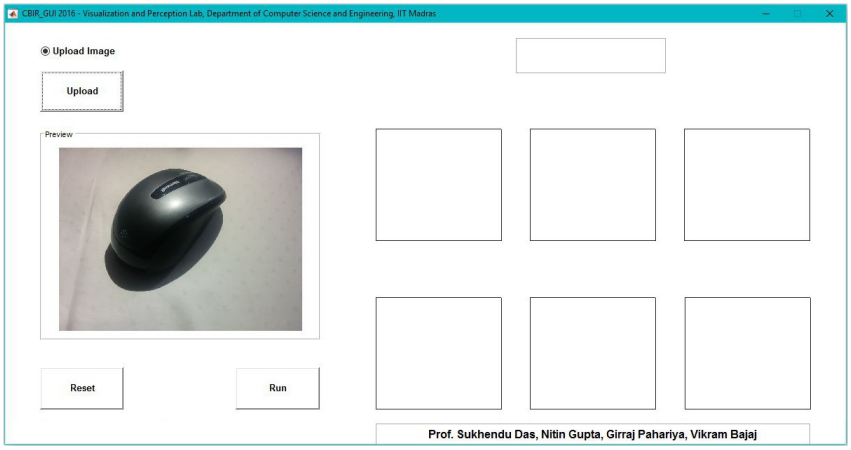

Fig. 6: Image Acquisition by Upload.

2) Salient Object Detection:
The salience (also called saliency) of an item – be it an object, a person, a pixel, etc. is the state or quality by which it stands out relative to its neighbors. Thus, saliency detection in images involves computing the likelihood of a location in an image to attract the attention of a human.
In this project, saliency detection is done using an algorithm called SalCut.
SalCut is an unsupervised segmentation technique for object localization. It is a combination of two state-of-the-art techniques, namely GBVS [3] (Graph-Based Visual Saliency) and GrabCut [4].
The limitation of SalCut is that it helps in localization of only a single, prominent object in an image, but is mostly accurate.

Fig. 7: SalCut Output.

3) Iterative Feedback Mechanism:
The image returned by SalCut is considered as the test image. Features are extracted from this test image in the same way as in the offline phase. These features are given to the recognition (WHAT) module, where the SVM outputs one of the 28 class labels.
Now, the image is passed to the detection (WHERE) module, which attempts to locate the recognized object. If found, the output image of the detection module is passed to the recognition module, which attempts to confirm that the detected object is the same as the one recognized initially.
This WHAT-WHERE loop is referred to as the Iterative Feedback Cycle.
The output of this step is either a class label (when an object is both recognized and detected) or NULL (when no object is recognized or detected).

4) Similarity Estimation:
The categorized feature database created in the offline phase contains HOG feature vectors of all the images in the dataset. The aim of similarity estimation is to compare the features of the test image with those in the categorized feature database. To do so, first HOG features are extracted from the test image, and a dot product is used as the similarity measure.
If the output of the iterative feedback cycle is a class label, the features of the test image are compared only with those of the images belonging to that class in the dataset. If the output of the iterative feedback cycle is NULL, the features of the test image are compared with those of all the images in the dataset.

5) Displaying the Rank-Ordered Output:
The output of the similarity estimation task is an array of dot products between features of the test image and those of images in the dataset. The top six elements in this array correspond to dot products that denote the highest similarity between the test image and corresponding images in the dataset. These six images are retrieved from the dataset and are displayed as a rank-ordered set of images on the axes of the GUI.

III. AUTOMATIC MODE

This project also aimed at making the CBIR system run repeatedly from start to end, without human intervention. This meant that the system was to be made capable of automatically capturing the test image, processing it and retrieving similar images from the dataset, over and over again.

IV. OUTPUT

The following images depict the outputs of the CBIR system for a few test images. The numeric scores above the retrieved images depict the similarity estimates between the test image and the retrieved image.

A. *A few successful test cases*

It has been observed that the system works well for test images that are well-illuminated and contain less background clutter. Also, the detection module performs better if the object doesn't cover most part of the image.

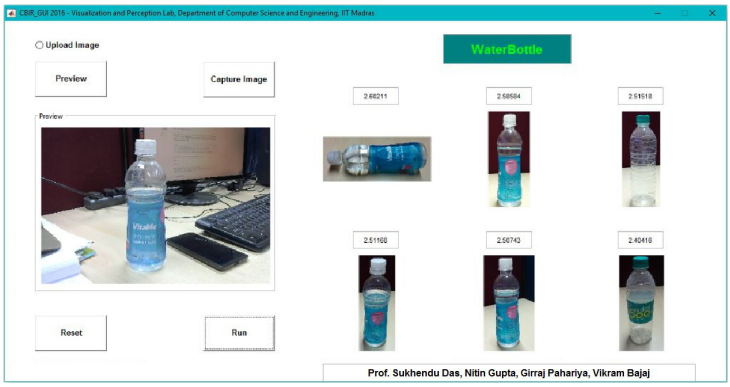

Fig. 8: Test image with a little background clutter.

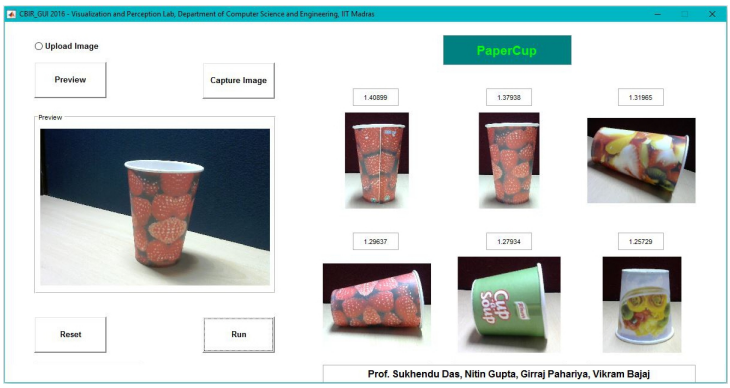

Fig. 9: Well-illuminated test image.

Fig. 10: Test image where the object doesn't cover most part of the image.

Fig. 11: Test image uploaded to the GUI.

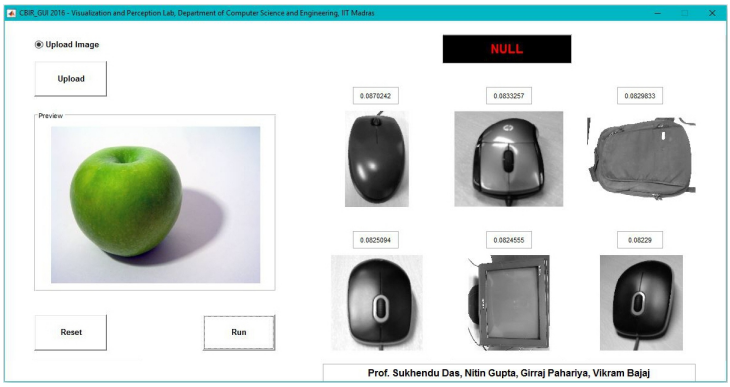

Fig. 12: Test image containing an object that doesn't belong to any of the 28 categories.

B. An unsuccessful test case

The system sometimes misclassifies objects, or fails to recognize objects that it has been trained to recognize. The following image depicts a misclassification.

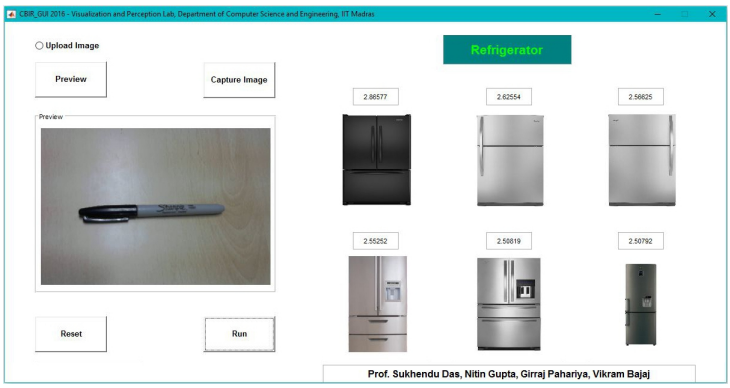

Fig. 13: A misclassification.

C. Precision-Recall and ROC curves

In the context of CBIR, precision is the ratio of the number of relevant images retrieved to the total number of irrelevant and

relevant images retrieved, while recall is the ratio of the number of relevant images retrieved to the total number of relevant images in the dataset.

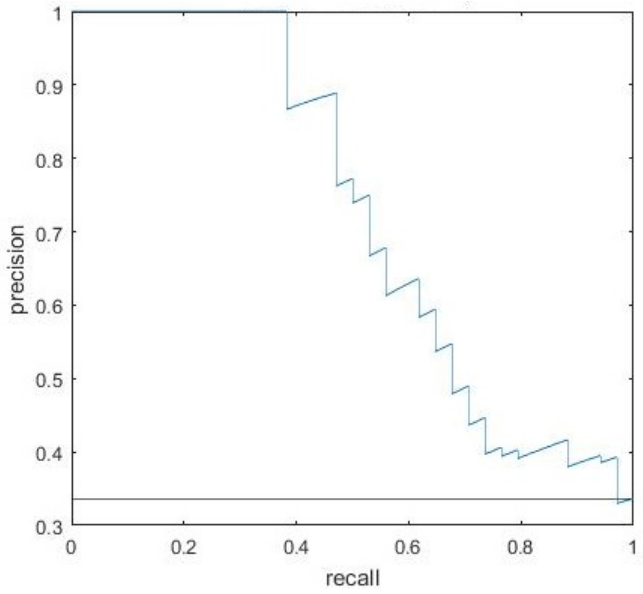

Fig. 14: Precision-Recall curve (for the ComputerMouse class).

A Receiver Operating Characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

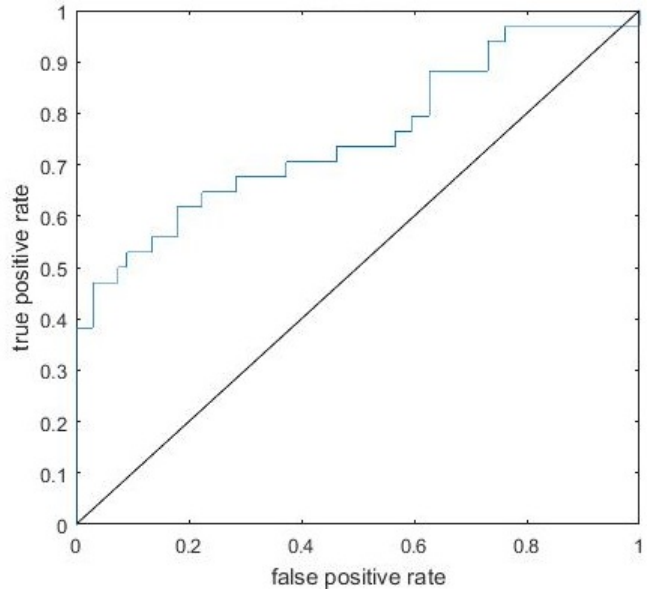

Fig. 15: ROC curve (for the ComputerMouse class).

V. Conclusion

A content-based image retrieval system was created and a GUI was developed to demonstrate its working. The system was also made capable of running from start to end repeatedly, without human intervention. Though the system achieved decent performance, there is room for improvement. What makes this system stand out is its Simultaneous Localization and Recognition (SLAR) framework [2], implemented by the iterative feedback cycle.

References

[1] Nitin Gupta, Sukhendu Das and Sutanu Chakraborti, "Extracting Information from a Query Image, for Content Based Image Retrieval", in ICAPR, 2015.

[2] G. Dwivedi, S. Das, S. Rakshit, M. Vora, and S. Samanta, "SLAR (Simultaneous Localization And Recognition) framework for smart CBIR", in PerMIn, vol. 7143. LNCS, 2012.

[3] Jonathan Harel, Christof Koch and Pietro Perona, "Graph-based visual saliency", in NIPS, 2006.

[4] Carsten Rother, Vladimir Kolmogorov and Andrew Blake, "GrabCut: interactive foreground extraction using iterated graph cuts", in ACM graph-cuts, ACM TOG, vol. 23, 2004.

[5] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan, "Object Detection with Discriminatively Trained Part Based Models", in IEEE TPAMI, vol. 32, 2010.

[6] Christianini, N. and J. C. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods", Cambridge University Press, 2000.

[7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in CVPR, 2005.

[8] B. J. Baars and N. M. Gage, "Cognition, Brain, and Consciousness, Introduction to Cognitive Neuroscience", Academic Press, 2010.

[9] E. T. Rolls and G. Deco, "Computational Neuroscience of Vision", Oxford, 2004.

[10] Bay, H., Tuytelaars, T. and Van Gool, L, "SURF: Speeded Up Robust Features", in ECCV, 2006.

[11] G. Csurka, C. Dance, L.X. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints", Workshop on Statistical Learning in Computer Vision, ECCV, 2004.