

Intelligent Image Database Searching system

ECE 277A
Project Report

Jelena Tešić

Abstract

An intelligent image database search system is a system capable of searching the digital image databases for user-defined objects. User's information is in the form of acceptance or rejection of retrieved images. This information is used as a collection of positive and negative training examples for a class-specific classification network by identifying clusters in the data. Each network consists of a set of Radial Basis Function neural network with a non-linear perceptron output layer. Relevance feedback is employed to iteratively refine queries. The user can review and adjust the behavior of the network. Color and texture feature vectors represent images for search and retrieval purposes. This project concentrates on examining the effectiveness of the feature set being used and on the RBF training process capability. The ultimate goal is to make more use of user's information and to make a system more operative.

1. Introduction

Many image-database retrieval systems rely on the success of a single query. The optimized feature sets is used to obtain the best possible results. However, user expectance cannot be sufficiently satisfied in such relatively simple terms. If the query fails then the user must try and improve the description using only the available feature descriptors. This project investigates how a system can exploit the user's knowledge.

The technique uses user's information to train radial basis function (RBF) neural networks. The process can be described in three steps: preprocessing data, retrieving queried images and off-line training of RBF networks.

- Data preprocessing means a creation of a query database of image feature vectors. We use a database of 16 000 feature vectors. Shawn Newsam and Baris Sumengen originally developed the database for the Categorical image search engine. The feature vector is combination of a 24 dimensional texture feature vector and a 64 dimensional color feature vectors.
- The feature vector of a key image initiates query. System selects a preliminary set of image thumbnails based on the closeness in the feature space. The user is required to state, for each of the retrieved images, if it matches the original image or not.

- After a query session is completed, an off-line phase employs a complex training algorithm to create a single-class classifier. A set of Radial Basis Function (RBF) nodes is placed appropriately in the feature space amongst the training data vectors. The outputs of the nodes are connected to a single-layer two-output perceptron. The centroids of RBF are bootstrapped using the LVQ algorithm. The output layer is trained using SCG algorithm with the bootstrapped RBF outputs. The output layer is trained so that the balance between color and texture component satisfies user's selection. The multi-stage training process – centroids, output layer, distribution then output layer again – continues until parameter movement is minimized.

The resulted classifier can be used to search for the images that correspond to the user-defined class upon which it was trained.

2. Motivation

Lately, digital-database image browsing became an important research topic in image processing. Different approaches had been taken. Some systems use mixture of optimized image features for on-time query (QBIC) or refined query process (Categorical Image Search). Another group of image systems implements learning component; e.g. FourEyes system [refs] tries to overcome the difficulties of dimensional explosion in feature space by using a "society of models". In an initial off-line phase, a number of different filtering techniques are applied to the data before any queries are made in order to hierarchically cluster the data in as many ways as possible. User feedback is filtered back to alter the clustering according to the success or failure of the query, thus adapting the groupings to the user's needs.

This project is based on the Categorical Image Search System (UCSB) and IDQS system (University of Bristol). The Categorical Image database has 16000 images from WWW and uses global image and texture features to describe them. IDQS system has around 2100 images. It uses localized features and employs region-based refined retrieval. Radial Basis Function (RBF) neural networks were chosen to perform the classifications in this system since only the small amounts of training data (i.e. the initial selections and subsequent feedback of the user) is present at the time.

We are investigating the use of learning component in the large image databases. However, the most promising area of the work is the development of the ability to store useful classes captured from a completely open-domain, e.g. WWW that Categorical Image Search System uses.

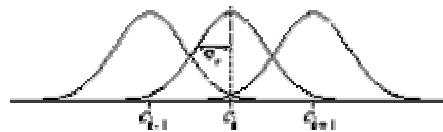


Figure 1 One-dimensional Radial Basis Functions

3. Radial Basis Functions (RBF)

Radial basis functions (RBFs) are the natural generalization of coarse coding to continuous-valued features. Rather than each feature being either 0 or 1, it can be anything in the interval $[0,1]$, reflecting various *degrees* to which the feature is present. A typical RBF feature, i , has a Gaussian response, $\phi_s(i)$, dependent only on the distance between the state, s , and the feature's prototypical or center state, c_i , and relative to the feature's width, σ_i :

$$\phi_s(i) = \exp\left(-\frac{\|s - c_i\|^2}{2\sigma_i^2}\right)$$

The norm or distance metric can be arbitrarily chosen. Figure 1 is one-dimensional example with a commonly used Euclidean distance metric.

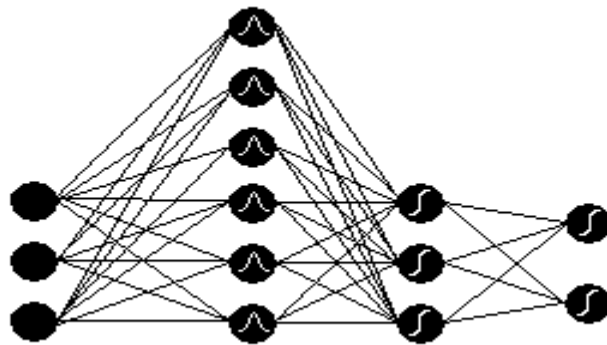


Figure 2 RBF network

Radial basis function (RBF) (Figure 2) networks have Gaussian function as the nonlinearity for the hidden layer processing elements. The Gaussian function responds only to a small region of the input space where the Gaussian is centered. The key to a successful implementation of these networks is to find suitable centers for the Gaussian functions.

The simulation starts with the training of an unsupervised layer. Its function is to derive the Gaussian centers and the widths from the input data. These centers are encoded within the weights of the unsupervised layer using competitive learning. During the unsupervised learning, the widths of the Gaussians are computed based on the centers of their neighbors. The output of this layer is derived from the input data weighted by a Gaussian mixture.

Once the unsupervised layer has completed its training, the supervised segment then sets the centers of Gaussian functions (based on the weights of the unsupervised layer) and determines the width (standard deviation) of each Gaussian. Any supervised topology (such as a MLP) may be used for the classification of the weighted input.

The advantage of the radial basis function network is that it finds the input to output map using local approximators. Usually the supervised segment is simply a linear combination of the approximators. Since linear combiners have few weights, these networks train extremely fast and require fewer training samples.

4. The query

A query is initiated by the selection of a key image. This identifies the elements that should be present in other images to be retrieved. The system selects a preliminary set of images by minimizing the Euclidian distance measure from the region's feature vector to those of potentially similar regions in other images.

Previously rejected regions are not chosen regardless of their proximity to a positive cluster center. This new information is added to the original data and passed once again to the LVQ algorithm for re-clustering. Thus the iterative refinement continues until the user is satisfied with the resulting images. If, at any stage, the user is unhappy with the direction the system appears to be heading, a new key region can be added to the dataset. This has been found to help avoid local minima in the class training. This also helps to deal with classes where the features which best describe it are a subset of complete feature set. This is a highly important aspect of the system's ability to generalize given the user's query description, for classes captured from an open-domain. The user is free at any time to select multiple examples of similar objects even if some of the properties differ between them. The clustering algorithm used will identify the multiple areas in feature space thus providing examples in the results of both types.

At all stages of the on-line session, the user is given the option of retaining the positive and negative class examples to contribute to a class database, which is stored between sessions. A key image and textual tag can be assigned to the class to ease future identification. This means that when a user requires images pertaining to a class upon which a previous query has already been based, the entire relevance feedback process need not be repeated. In addition to this, off-line training can be performed in order to improve the results.

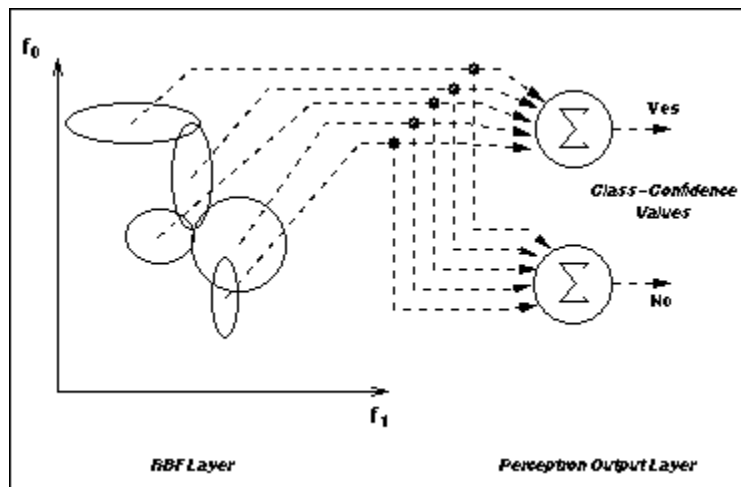


Figure 3 The architecture of an RBF binary classifier.

5. Off-Line Training

After a query session is completed, an off-line phase can be performed based on the user's information. More complex training algorithm is used to create a single-class classifier off-line. 2-D feature space classifier is shown in Figure 3. A set of RBF nodes is placed appropriately in feature space amongst the training data's vectors. The number of nodes used is much lower than for the on-line sessions - usually around 10 split equally between positive and negative clusters. The outputs of the nodes are connected to a single-layer, two-output perceptron which provides a confidence of class membership of a given feature vector applied to the input. The centroids of the RBFs are bootstrapped using the LVQ algorithm and the Gaussian spreads selected according to the location of the training data. Note that the distribution of the RBFs can be varied independently across each axis. The output layer is trained using a Scaled Conjugate Gradient (SCG) algorithm with the boot-strapped RBF outputs derived from the user's training data and optimized according to the RBFs initial settings to minimize the output of the perceptron.

Once training is complete, this mini-classifier can be used to search for the images in the database that correspond to the class of object upon which it was trained and can be loaded in during on-line sessions to aid the user. Thus a user-defined class can be represented by a relatively small set of numbers (i.e. the node centroids, link weights etc).

6. Implementation

The initial stage of the system is implemented in MATLAB, using the software package [Matlab Functions for Radial Basis Function Networks \(1999\)](#), developed by Michael Orr. The software implements a variety of methods based on subset selection and ridge regression to control model complexity and regression trees to generate RBF centers and radii. Therefore, online training work has progressed in two directions, depending on the RBF node modification procedure. Described approach (section 5) re-selects the RBF nodes based on the LVQ algorithm on user's vector selection. Gaussian spreads selected according to the location of the training data. Modified Orr's functions use user's information is to modify the regression tree. The surviving method will be determined based on what direction this project takes.

Images from Categorical Image Search database have been used. The feature vector consists of a color and texture features. The 64 dimensional color feature vectors are extracted from the image histogram, while the 24 dimensional texture feature vectors represents Gabor filter outputs. Due to the time constraint, the evaluation has been carried out on the images from the category number 23 (sunglasses) that contains 337 images. The color and texture feature vectors formed balanced RBF network.

Initial set of images was based only on the smallest overall Euclidean distance. The response for the closest ten retrievals was considered for user's response. From one pass, system learned to recognize the desired shaped and color of sunglasses and found the sunglasses that mach the description.

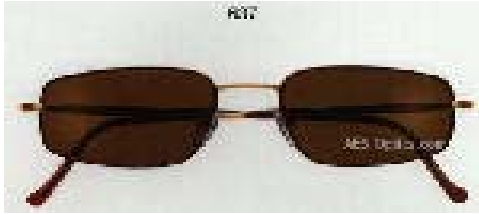


Figure 4 The key image



Figure 5 The retrieved images

1 to 32 of 32 << Prev Next >>

(23, 126) 1.151234 0.616383 0.534851	(23, 135) 1.375848 0.601356 0.774492	(23, 132) 1.468355 0.544664 0.923691	(23, 137) 1.8825281 0.664546 1.217982	(23, 130) 1.9146221 1.362689 0.551933	(23, 139) 2.091704 1.622883 0.468821	(23, 129) 2.217634 1.088926 1.128708	(23, 125) 2.812315 2.022173 0.790142
(23, 124) 3.2203732 2.463387 0.756986	(23, 131) 3.221476 2.547156 0.67432	(23, 138) 3.244458 1.291759 1.952699	(23, 133) 3.36137 1.877719 1.483651	(23, 127) 3.753323 3.051489 0.701834	(23, 180) 3.946082 0.756097 3.189985	(23, 134) 3.974626 1.741809 2.232817	(23, 136) 3.997664 1.913791 2.083873
(23, 182) 4.118095 1.394717 2.723378	(23, 250) 4.145394 1.903948 2.241446	(23, 249) 4.225091 1.991114 2.233977	(23, 198) 4.23662 1.562319 2.674301	(23, 74) 4.23662 1.562319 2.674301	(23, 184) 4.302422 1.259471 3.042951	(23, 183) 4.324541 1.6685 2.656041	(23, 174) 4.4830837 1.557138 2.925946
(23, 195) 4.486979 1.882053 2.604926	(23, 71) 4.486979 1.882053 2.604926	(23, 166) 4.488932 1.830921 2.658011	(23, 185) 4.5342007 1.707587 2.826614	(23, 61) 4.5342007 1.707587 2.826614	(23, 114) 4.5493712 1.559507 2.989864	(23, 179) 4.7121334 2.057249 2.654884	(23, 190) 4.805822 2.098743 2.707079

Figure 6 Categorical Image Search Retrieval

The key image is the 126th image in the 'sunglasses' category. Figure 5 represents the 4 top retrievals from the sunglasses category after one cycle. After two cycles, the Figure 6 represents the output of the Categorical Image Search.

7. Future Work

The initial experiment was conducted only inside one category. The future work includes a creation of a demo using the 16000 image database for categorical image search and testing the ability to create new categories. As for the short-term plans, making the algorithm optimized and operable under generalized conditions is of an ultimate priority.

Next step is to build a hierarchical system for real-time image categorization. An RBF network will be implemented for every feature vector. The user's feedback will be used to learn how to weight the separate RBF outputs in the correct way. As well as examining the effectiveness of the feature set being used and the nature of the RBF training, this work will now concentrate on the class library and how it can be used to make the system more useful and effective. Also, such a system could be supplied to the user with default classes since they can be used on unseen data sets. The user would then expand on these and add new ones.

Qualitative analysis has been carried out into the performance of the RBF networks and it has been seen in many examples that there is a greater tendency towards generalization using this approach.

A feature set that this search engine uses is a trade off between time and effectiveness. One would conceivable require a much more complex feature set in order to discriminate highly similar objects. Here the idea is to retrieve as many images of objects similar to that chosen by the user as the starting point. Thus incorrect classifications do not have serious implications and the speed of the retrieval is not sacrificed.

The most promising area of the work is the development of the ability to store useful classes captured from a completely open-domain database, which can be used in subsequent queries and indeed on other databases.

8. Conclusion

This work has shown how important it is for user feedback to be employed during image database retrieval, since it is a rich source of information regarding the nature of the query. But, from the results we have so far, no further conclusions can be drawn. The superiority of feature vector weights adjusting or learning module training is still to be determined.

9. References

1. M.E.J. Wood, N.W. Campbell, and B.T. Thomas. "Employing Region Features for Searching an Image Database" In The Proceedings of the 1997 British Machine Vision Conference, pages 620-629, 1997.
2. M.E.J. Wood, N.W. Campbell, and B.T. Thomas. "Searching Large Image Databases using Radial Basis Function Neural Networks" In Proceedings of the Sixth International Conference on Image Processing and its Applications, pages 116-120, 1997.
3. W. Y. Ma and B. S. Manjunath, "NETRA: A toolbox for navigating large image databases," Proc. IEEE International Conference on Image Processing, Santa Barbara, California, Vol. I, pp. 568-571, Oct 1997.
4. Y. Deng and B.S. Manjunath, "NeTra-V: toward an object-based video representation", IEEE Transactions on Circuits and Systems for Video Technology, vol.8, (no.5), pp.616-27, Sep 1998.
5. S. Haykin. Neural Networks - A Comprehensive Foundation. Macmillan College Publishing, New York, 1994.
6. R. Gonzalez and R. Woods. Digital Image Processing Addison-Wesley, New York, 1992.

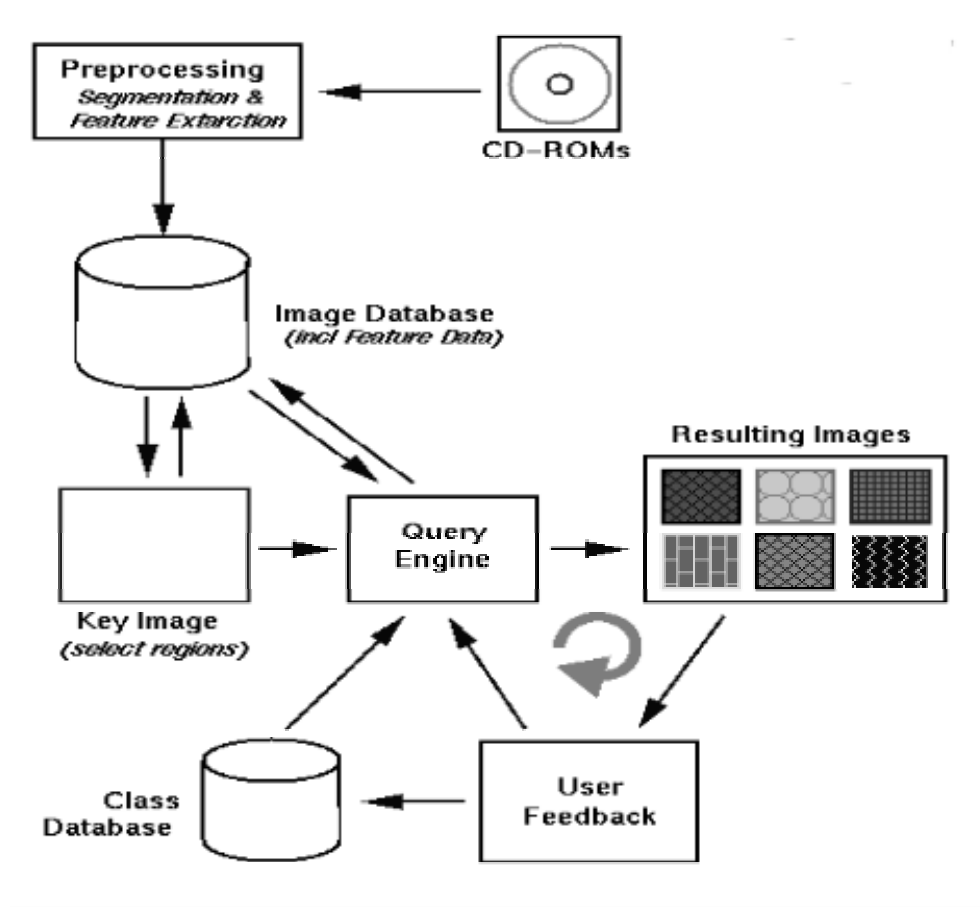


Figure 7 Image Database search engine with relevance feedback