Gabor Filter based Fingerprint Classification using Support Vector Machines

Dhruv Batra, Girish Singhal and Santanu Chaudhury

Abstract— **Fingerprint classification is important for different practical applications. An accurate and consistent classification can greatly reduce fingerprint-matching time for large databases. We use a Gabor filter-based Feature extraction scheme to generate a 384 dimensional feature vector for each fingerprint image. The classification of these patterns is done through a novel two stage classifier in which K Nearest Neightbour (KNN) acts as the first step and finds out the two most frequently represented classes amongst the K nearest patterns, followed by the pertinent SVM classifier choosing the most apt class of the two. 6 SVMs have to be trained for a four class problem, (${}^6C_2$), that is, all one-against-one SVMs. Using this novel scheme and working on the FVC 2000 database (257 final images) we achieved a maximum accuracy of 98.81% with a rejection percentage of 1.95%.This is significantly higher than most reported results in contemporary literature. The SVM training time was 145 seconds, i.e. 24 seconds per SVM on a Pentium III machine.**

Index terms—**Biometrics, Fingerprint classification, Gabor filters, KNN, SVM.**

I. INTRODUCTION

BIOMETRICS is the measurement of biological data. The term biometrics is commonly used today to refer to the authentication of a person by analyzing physical characteristics, such as fingerprints, handprints, eyes and voice, or behavioral characteristics, such as signatures. Fingerprints are the ridge and furrow patterns on the tip of the finger and have been used extensively for personal identification of people. Large volumes of fingerprints are collected and stored everyday for a wide range of applications including forensics, access control, and driver license registration. Automatic recognition of people based on fingerprints requires that the input fingerprint be matched with a large number of fingerprints in a database (FBI database contains approximately 630 million fingerprints) [4]. To reduce the search time and computational complexity, it is desirable to classify these fingerprints in an accurate and consistent manner so that the input fingerprint is required to be matched only with a subset of the fingerprints in the database. Fingerprints can be classified into five distinct classes, namely, whorl (W), right loop (R), left loop (L),arch (A), and tented arch (T) based on the classes identified by the National Institute of Standards and Technology (NIST) to benchmark automatic fingerprint classification algorithms. There are two main types of features in a fingerprint:

(1) Global ridge and furrow structures which form special patterns in the central region of the fingerprint, and

(2) Local ridge and furrow minute details.

A fingerprint is classified based on only the first type of features and is uniquely identified based on the second type of features (ridge endings and bifurcations, also known as minutiae).

Minutiae (○), Core (□), and Delta (△).

Figure 1: a. Minutia, Core and Delta points
b. The experts have labeled this image to belong to two classes, right loop, and tented arch.

Fingerprints have a continuum in the pattern space. For example, there is a continuum of patterns between the two extremes of a "true" arch and a "true" loop. As a result, there exist patterns that lie on any arbitrarily drawn class boundary drawn for an exclusive classification. About 17% images in the NIST-4 [3] database have two different ground truth labels. This means that even human experts could not agree on the true class of the fingerprints for about 17% of the fingerprint images in this database containing 4,000 fingerprint images.

II. PREVIOUS WORK AND MOTIVATION FOR PRESENT WORK

Several approaches have been developed for automatic fingerprint classification. The feature extraction algorithms

Dhruv Batra is student in the Department of Electronics Engineering, Institute of Technology, Benaras Hindu University.
Girish Singhal is student in the Department of Electronics and Communication Engineering, Indian Institute of Technology, Guwahati.
Dr.S.Chaudhury is professor in the Department of Electrical Engineering, Indian Institute of Technology, Delhi.
Corresponding Author: email: santanuc@ee.iitd.ernet.in
phone: +91 (11) 26596167

have been based on orientation fields estimation (also called structure based techniques)[6,7], and minutia points [5] or exclusively global information (classification based on the Henry system [2],[9]). A Gabor filter based approach was developed by Jain et al [2] which used a bank of Gabor filters to produce a feature vector called FingerCode. This technique does not use singular points explicitly, and is robust to noise. This is the technique used in this work, although the core point location scheme used is the maximum concavity location scheme given by Jain et al [10] in 2001 rather than poincare index [2] or orientation field based schemes. The classifications algorithms generally used in the literature are ANNs, e.g. Jain et al [1], Blue et al [11]; nearest neighbours, e.g. Fitz and Green [14] , HMM, e.g Senior [15]; SVMs, e.g. Hao and Gou[16], Shah and Shastri[17].

Due to their fundamental advantages over ANNs we felt that the classification stage should ideally consist of SVMs. Most contemporary works however, reject the use of SVMs due to two reasons:

(1) When "p" (one-against-the others) SVMs are trained, the training dataset involves all "p" patterns and thus the time taken to train these SVMs becomes too large without proportionate increase in accuracy [28].

(2) When "n(n-1)/2" (one-against-one) SVMs are trained (here n is the no. of classes), the testing pattern has to pass through all these networks and then a "voting scheme" [16] has to be devised to deduce the final class. While classifying a large database this technique results in unnecessary testing of each data with irrelevant SVMs, resulting in reduction of operating speed of classification.

However we feel that with an appropriate architecture both these obstacles can be overcome. Since they are primarily binary classifiers a first stage of classification is needed to govern the selection and working of various SVMs. The problem, thus, breaks down to converting an N-class problem to be implemented with the help of binary classifiers. The chosen architecture should not only minimize the number of SVMs to be trained, but should also reduce the number of SVMs against which the test pattern has to be tested, thereby reducing both training and testing time. We thus propose a hybrid algorithm with nearest neighbour acting as the first stage and SVMs acting as the second stage. To the best of our knowledge there is no instance in the literature of any prior attempts along these lines and this is what makes it a novel technique.

III. FEATURE EXTRACTION:

Following steps are observed to extract the feature vector:

(1) Preprocessing of the image (to remove noise) by window wise normalization, Histogram Equalization, low pass and median filtering [26]

(2) Core point location using max concavity estimation [10]

(3) Tessellation of circular region (6 shells, 15 sectors each, 15 pixels wide) around the reference point.

(4) Sector wise normalization followed by application of bank of 4 Gabor filters (0^o, 45^o, 90^o, 135^o), [2].

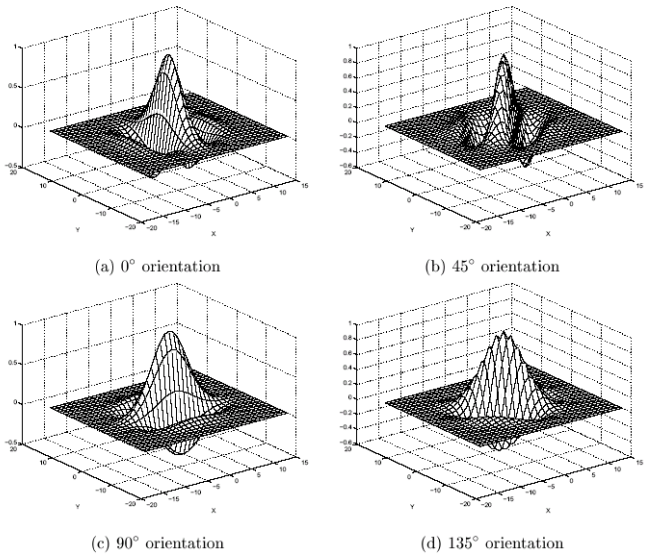
$$G(x, y; f, \theta) = \exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_{x'}^2} + \frac{y'^2}{\delta_{y'}^2}\right]\right\}\cos(2\pi f x')$$

$$x' = x\sin\theta + y\cos\theta$$

$$y' = x\cos\theta - y\sin\theta$$

Where f =freq of the sine plane wave along the direction θ from the x-axis, and $\delta_{x'}$ and $\delta_{y'}$ are the space constants of the Gaussian envelope along X' and Y' axes, respectively.

(5) Finally feature code generation by obtaining standard deviation values of all the sectors (4*96 = 384 sectors), [2].



(a) 0° orientation (b) 45° orientation (c) 90° orientation (d) 135° orientation

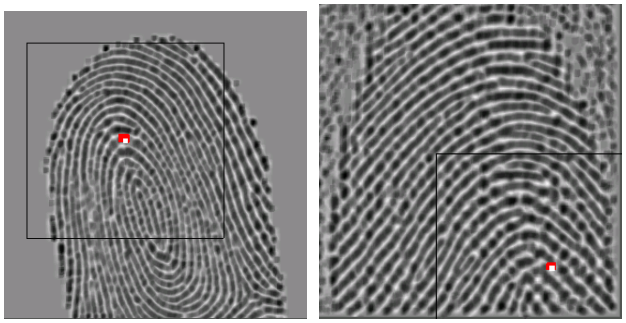
Figure 2: Spatial Orientation of Gabor Filters

Let $F_{i\theta}(x,y)$, be the θ - direction filtered image for sector Si. For i = 0,1,…96 and θ = {0,45,90,135} ,a feature is the variance $V_{i\theta}$,defined as : - $V_{i\theta} = \sqrt{\sum_{K_i}(F_{i\theta}(x,y) - P_{i\theta})^2}$, where K_i is the number of pixels in S_i and $P_{i\theta}$ is the mean of pixel values of $F_{i\theta}(x,y)$ in S_i. The variance of each sector in each of the four filtered images defines our feature vector.

Figure 3: a. Original Image b. Histogram Equalized Image c. Normalized Image d. LPF, median filtered image e. Segmented image f. Smoothened orientation field g. Sine component of field h. Reference point with ROI marked.

Discussions on feature extraction:

(1) Besides false reference point detection due to noisy images, the main problem at the feature extraction stage is that always Region of Interest (ROI) could not be satisfactorily located. The circular ROI drawn around the reference point might not contain enough discerning features to differentiate one class from another. Increasing the marked ROI around the core point leads to a higher rejection of images. We chose (in an image size of 300x300) 6 shells of 15 pixels width each, around the core point requiring a girth of 210 pixels, thereby leaving a central region of 90x90 pixels where the core point must be present for the image to be accepted. Since most solid state fingerprint scanners capture low resolution images, thus this ROI cannot be increased indefinitely.



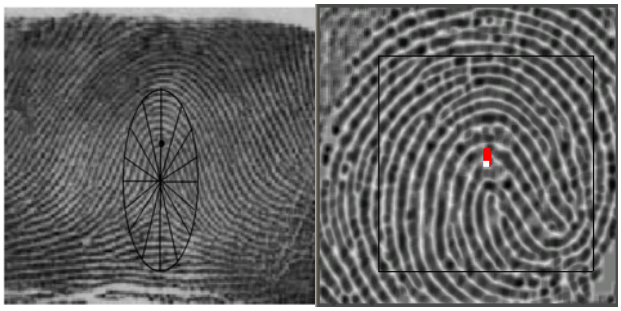


Figure 4 a. Captured Region of Interest poses an ambiguity. b. Image rejected as core point lies near the boundary. c. Elliptical captured region. d. Ambiguity between Twin Whorl and right loop

However, we feel that this tradeoff is unnecessary. As can be seen from the images captured, the northern-most arch region is similar in all categories, the differentiating features lying in the southern regions. Since the core point algorithm detects the point of maximum curvature in concave ridges, thus one of the solutions to eliminate this problem could be to use elliptical tessellation, with the core point lying at one of the foci, instead of a circular tessellation. It must be stated here that this is merely an empirical suggestion, heuristically achieved.

This would result in:

a. Capture of region of interest (ROI), with more discriminating capabilities

b. Less redundancy within the ROI

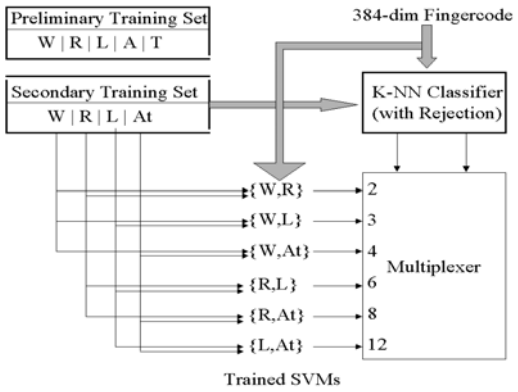
c. Reduced rejection of images at the feature extraction stage

(2) Ambiguity of captured Region of Interest (ROI): For twin loop images which are labeled as whorl in the NIST database, our centre location algorithm picks up the upper core and on considering that as the reference point, the image looks like a loop in the ROI which leads to a misclassification of whorl as left or right loop (refer to figure). A tilted, longitudinally stretched whorl may be misclassified with a left or right loop if pertinent region of interest (covering both the core points of the whorl) is not captured (refer to figure 4)

IV. CLASSIFICATION:

The classification technique used in this paper is a novel two-stage classifier consisting of a K-Nearest Neighbour (KNN) as the first stage and one-to-one Support Vector Machine (SVM) as the second stage. The KNN finds k nearest neighbours to the test case, in the feature space and keeps a list of the classes represented by those neighbours. The pertinent SVM then differentiates the top two most frequent classes. For this stage, 6 ($^4C_2$) SVM classifiers had been trained on the training data.

Figure 5: Proposed architecture



A. K-NEAREST Neighbours :

The K Nearest Neighbour (or KNN) finds out the K nearest neighbours (in feature space) to a test case from the training cases, based on the Euclidean distances between them. Of course the computing time goes up as K goes up, but the advantage is that higher values of k provide smoothing that reduces vulnerability to noise in the training data. In practical applications, typically, k is in units or tens rather than in hundreds or thousands.

B. SVM Overview:

Vapnik introduced Support Vector Machines as powerful learning tools based on statistical learning theoryAn SVM is a binary classifier that makes it's decision by constructing a linear decision boundary or hyperplane that optimally separates data points of the two classes in Feature Hyper space[23] and also make the margin maximized.The use of the kernel $x \rightarrow \phi(x)$ means that explicit transformation of the data into the feature space is not required, known as the Kernel Trick.We have used a Gaussian Radial Basis Function $K(x,y) = exp(-\frac{\|x-y\|^2}{2\sigma^2})$ with sigma=0.5 (empirically determined). From the aspect of implementation training a Support Vector Classifier is equivalent to solving a Quadratic Programming Problem of linear restriction. We used the Sequential Minimization Optimization (SMO) by John C. Platt [24] to do the same.

C. Advantages of SVMs over NN:

The following are advantages of SVMs:

(1) ANN are prone to the danger of over-training [27], resulting in a solution over-fitted to the database being worked on. This could lead to overly optimistic results and accuracy outcomes [28]. This is not possible in SVMs, as it is not the number of patterns used in training that influence the decision hyper plane, but the number of closest patterns, or Support Vectors.

(2) It has been found that SVMs are significantly faster to train that ANNs. Our MatLab implementation, for example takes only ***145*** seconds to train six SVM's on a data set of 128 patterns, i.e., an average training time of ***24*** seconds. In comparison, Neural Networks take much longer to train.

(3) Due to gradient descent algorithm, ANNs are notoriously prone to getting stuck in the local minima. SVMs have no such problems and always converge to solution.

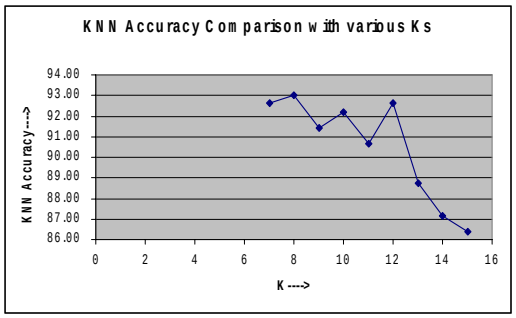
(4) Training data of a Neural Network has to be voluminous because each training data helps in influencing the decision boundary. This is not strictly true for SVM's, since the hyperplane of SVM is not dependent on any patterns other than Support Vectors. Thus the stress in this case is the ability of the data to influence the position the decision hyperplane. The emphasis thus is on diversity of data, rather than its volume.

V. EXPERIMENTS/ OBSERVATIONS AND RESULTS

Although this database is more commonly used for reporting identification and authentication accuracies, we have used FVC 2000, Db1_a (800 images of size 300 X 300 pixels) for core point based fingerprint classification. 257 images passed the feature extraction stage, out of which 128 (32 from each class) were used to train the various SVMs and entire 257 were used for testing.

A. K- NEAREST neighbor: -

The classification accuracy of the KNN does not always increase with increasing K, the accuracy performance of the KNN for various K values are show in Figure below



The K nearest neighbor classier results in an accuracy of **93 %** for the four class (Arch and T.Arch together) classification when 8 nearest neighbors (K=8) are considered. We observed that **92.99 %** of the time, the class with the maximum frequency among the K (=8) nearest neighbors is the correct class and **5.83%** of the time the class with the second highest frequency is the correct class. This **98.82%** of the time this result itself can be used to accurately classify fingerprints into two out of four classes.

B. REJECT option:-

Classification accuracy can be increased by incorporating a rejection option, because in certain cases an incorrect SVM is chosen by the KNN due to lack to examples lying close to the testing example in feature space. Thus even though the pertinent SVM, if given a chance, would have probably classified the pattern correctly, an error is reported instead because an incorrect SVM is chosen. We use the (K-k') nearest neighbor classifier for rejection. If the number of training samples from the majority class amongst the K nearest neighbors of a test pattern is less than k' we reject the test pattern and do not attempt to classify it. Most of the rejected images using this scheme were those images that

appeared to belong to different classes (see section III). The classification accuracies for testing database (both with and without training data) for various values of K and k' are shown in Figures 6 and 7.The percentage of patterns rejected at various values of K and k' are shown in Table 1.

C. TWO STAGE classifier: -

As can be seen from Figure 6, 7 and Table 1, the performance of the Two Stage Classifier peaks (the criteria being maximum accuracy with rejection percentage under 10%) at K=8 and k'=4, resulting in an accuracy of **98.81%** (when training data is included) and **97.63%**(otherwise) with a reject percentage of 1.95%. This is higher than reported accuracies of most contemporary literatures. The confusion matrix for this case for the two stage classification is shown in the table below. It should be noted that we have achieved even higher accuracies like 99.4% and 100% but the rejection rates in those cases are unusually high.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
7	0	0	0	0.39	3.11	12.06	28.40	-	-	-	-	-	-	-	
8	0	0	0	0	1.95	8.17	21.01	38.52	-	-	-	-	-	-	
9	0	0	0	0	1.95	4.28	13.23	28.40	44.36	-	-	-	-	-	
10	0	0	0	0	0.78	1.95	7.39	19.46	35.41	50.19	-	-	-	-	
11	0	0	0	0	0	1.17	3.89	14.79	24.51	40.86	55.25	-	-	-	
12	0	0	0	0	0	0.39	1.56	12.84	18.68	30.35	43.58	60.70	-	-	
13	0	0	0	0	0	0.39	1.17	5.45	14.40	24.90	36.19	47.08	64.59	-	
14	0	0	0	0	0	0	1.17	3.89	7.78	19.46	29.57	40.86	51.36	68.09	
15	0	0	0	0	0	0	0	3.11	5.45	13.23	24.12	34.24	45.14	56.42	70.04

Table 1: Rejection Percentages for various values of K (Y axis) and k'(X axis)

Assigned

Actual Class	Whorl	Right Loop	Left Loop	Arch/T. Arch
Whorl	56	2	0	0
Right Loop	0	98	0	0
Left Loop	1	0	63	0
Arch/T. Arch	0	0	0	32

D. DISCUSSIONS:

(1) Since we are using a two stage classifier whose first stage is a KNN, there need to be a certain minimum number of training patterns of a type for them to come into a majority in the KNN (thereby resulting in the selection of the appropriate SVM), the value of this minimum depending on the value of K. A solution to this could be the use of a KNN with a measure for the degree of closeness of patterns, a qualitative measure than mere quantitative analysis. A weight could be associated with each pattern, with this quantity proportional to the inverse of the distance between the two vectors. In such a case, even a single example with an extremely high degree of closeness would outweigh several remotely spaced patterns.

(2) Time taken to train six SVMs (with a training database of 128 vectors of 384 dimensions) was 145 seconds, giving us an average training time of 24 seconds .It should be noted however that the time taken to train a particular SVM depends upon the two classes concerned. For example, time taken to train the SVM that classifies a left loop from a right loop is considerably less than that taken by the SVM that classifies a Whorl from L/R loop due to inherent ambiguities.

(3) Since in a numerical solution, it is not possible to achieve optimality exactly, thus for checking the validity of KKT conditions a tolerance factor is used, as suggested by Keerthi et al[25], to check them within a practical limit. Thus this parameter should be tuned to deliver a maximized performance.

VI. CONCLUSION:

The results and accuracies achieved validate our claim that our novel architecture is indeed more efficient than existing techniques in literature involving KNNs, ANNs, and SVMs.

VII. REFERENCES:-

[1]. Anil. K. Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti, "Filterbank-based Fingerprint Matching," IEEE Transactions on Image Processing, vol. 9, no. 5, pp. 846–859, May 2000

[2]. Anil K. Jain, Salil Prabhakar, and Lin Hong, "A Multichannel Approach to Fingerprint Classification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 4, pp. 348–359, April 1999

[3]. C. I. Watson and C. L. Wilson, "NIST Special Database 4, Fingerprint Database," National Institute of Standards and Technology, March 1992.

[4]. Federal Bureau of Investigation. www.fbi.gov

[5]. A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," *Proc. IEEE*, vol. 85, pp. 1365–1388, Sept. 1997.

[6]. C. L. Wilson, G. T. Candela, and C.I. Watson, "Neural Network Fingerprint Classification," *J. Artificial Neural Networks*, Vol. 1, No. 2, pp. 203-228, 1993.

[7]. R. Cappelli, D. Maio, and D. Maltoni, "Fingerprint Classification based on Multi-space KL", *Proc. Workshop on Automatic Identification Advances Technologies (AutoID'99)*, Summit (NJ), pp. 117-120, October 1999.

[8]. D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 27–40, Jan. 1997.

[9]. L. Hong and A. K. Jain, "Classification of fingerprint images," in *11th Scandinavian Conf. Image Analysis*, Kangerlussuaq, Greenland, June 7–11, 1999.

[10]. A. K. Jain, Davide Maltoni, Dario Maio, Salil Prabhakar, "Handbook of Fingerprint Recognition", Springer Professional Computing,2003.

[11]. C. L. Wilson, J. L. Blue, and O. M. Omidvar, "Training Dynamics and Neural Network Performance", *Neural Networks*, Vol. 10, No. 5, pp. 907-923, 1997.

[12]. C. L. Wilson, J. L. Blue, and O. M. Omidwar, "Neurodynamics of Learning and Network Performance", *Journal of Electronic Imaging*, Vol. 6, No. 3, pp. 379-385, 1997.

[13]. C. L. Wilson, G. T. Candela, and C.I. Watson, "Neural Network Fingerprint Classification," *J. Artificial Neural Networks*, Vol. 1, No. 2, pp. 203-228, 1993.

[14]. A. P. Fitz and R. J. Green, "Fingerprint Classification Using Hexagonal Fast Fourier Transform," *Pattern Recognition*, Vol. 29, No. 10, pp. 1587-1597, 1996.

[15]. A. Senior, "A Hidden Markov Model Fingerprint Classifier," *Proceedings of the 31st Asilomar conference on Signals, Systems and Computers*, pp. 306-310, 1997.

[16]. Yang He, Zong-Ying Ou, Hao Guo, " A method of Fingerprint Identification based on Space Invariant Transforms and Support Vector Machines", Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003.

[17]. Shah, S.; Sastry, P.S., "Fingerprint Classification Using a Feedback-Based Line Detector", Systems, Man and Cybernetics, Part B, IEEE Transactions on , Volume: 34 , Issue: 1 , Feb. 2004

[18]. M. Kass and A. Witkin, "Analyzing oriented patterns," *Comput. Vis.*

Graph. Image Processing, vol. 37, no. 4, pp. 362–385, 1987.
[19]. T. Chang, "Texture analysis of digitized fingerprints for singularity detection" in *Proc. 5th Int. Conf. Pattern Recognition*, 1980, pp. 478–480.
[20]. M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recogntion.*, vol. 17, no. 3, pp. 295–303, 1984.
[21]. A. R. Rao, *Taxonomy for Texture Description and Identification*, New York: Springer-Verlag, 1990.
[22]. L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 777–789, Aug. 1998.
[23]. Vapnik, V. *The nature of statistical learning theory*. Springer-Verlag, Berlin, 1995.
[24]. John C.Platt,Microsoft Research,*Sequential Minimal Optimization:A Fast Algorithm for Training Support Vector Machines,* Technical Report MSR-TR-98-14 April 21, 1998
[25]. S.S. Keerthi, S.K. Shevade, C. Bhattacharyya and K.R.K. Murthy, Improvements to Platt's SMO algorithm for SVM classifier design, *Neural Computation,* Vol. 13, March 2001, pp. 637-649
[26]. Shlomo Greenberg, Mayer Aladjem, Daniel Kogan and Itshak Dimitrov, "Fingerprint Image Enhancement using Filtering Techniques" Ben-Gurion University of the Negev, Beer-Sheva, Israel
[27]. Bernhard Schölkopf and Alex Smola, "Learning with Kernels" (MIT Press, Cambridge, MA, 2002).
[28]. S. Prabhakar, "Gabor filter bank based Fingerprint classification and identification.", PhD. Thesis, Department of Computer Engg., MSU

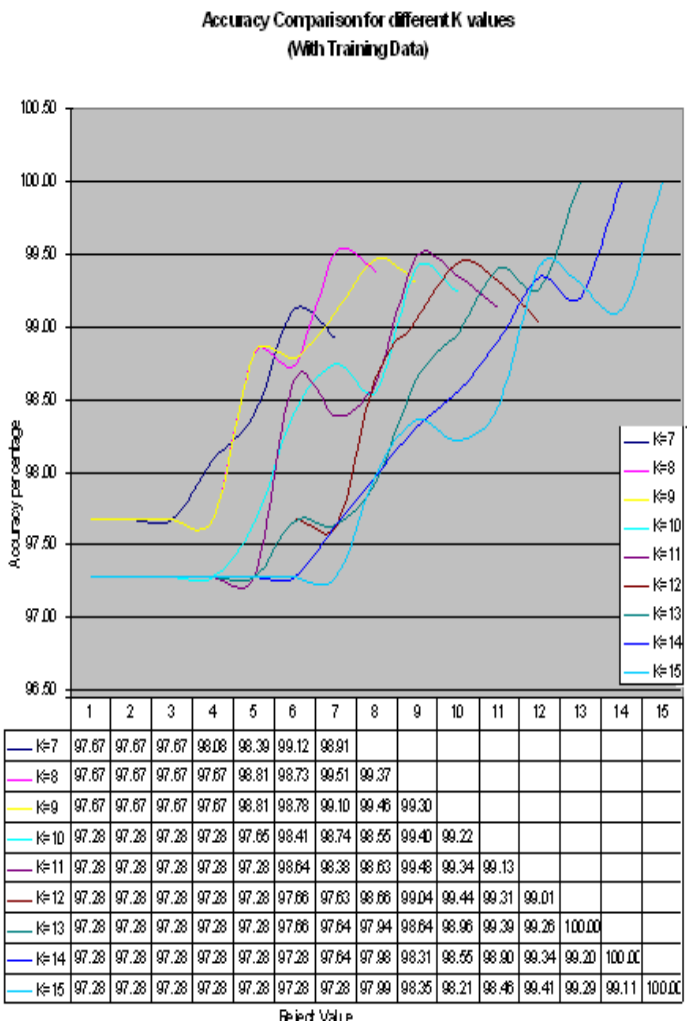


Fig 7:Accuracy Comparisons of 2-stage classifier for various values of K and k' (Training Data included)

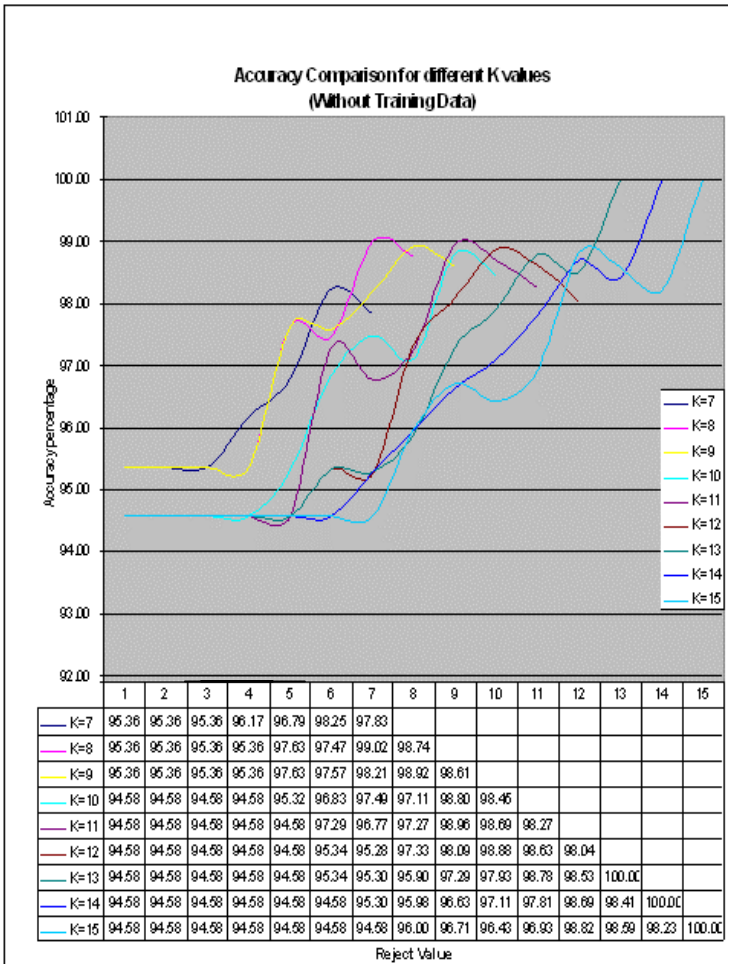


Fig 8:Accuracy Comparisons of 2-stage classifier for various values of K and k' (Training Data excluded)