Performance of Different Algorithms for Face Recognition

Parul Jain,Nilay Jain and Rohit Raja

Abstract—**This paper evaluates and performed of different algorithm for face recognition. We evaluate a variety of well-known face recognition algorithms (PCA, LDA, ICA, SVMs) against holistic performance speed, memory usage, storage size and metrics of accuracy. SVMs perform best with ~65% accuracy, but lower accuracy algorithms such as IPCA are orders of magnitude more efficient yielding a more feasible system in memory speed and consumption.**

***Index Terms*—PCA, LDA, ICA, SVM, Face Recognition, Algorithms**

I. Introduction

Face recognition is a popular research topic, maturing as researchers develop sophisticated algorithms to achieve more accurate results on increasingly difficult face datasets [1]. Since the original "eigenface" approach [2], other techniques such as Fisherfaces [3], Independent Component Analysis [4], and Support Vector Machines [5] have been proposed. More difficult face databases, such as the FERET [6] database, introduce variations in illumination, expression, pose and time lapses, leading to recent advances such as 3D modeling techniques [7].

With such advances in face recognition and algorithms claiming accuracies of greater than 90% one wonders how these algorithms would fare on a real system with real data. Ruiz-del-Solar and Navarrette in [8-10] provide an excellent review of face recognition algorithms, but only evaluate them for accuracy. Our contribution is utilizing a new source of different face database to evaluate performance of several face recognition algorithms.

Figure.1: Sample AT&T face image from the database.

Manuscript received Dec, 2014.

***Parul Jain**, Department of Computer Science and Engineering, SSGI, SSTC, Bhilai., Durg, India, 7587488017*

***Rohit Raja**, Department of Computer Science and Engineering, SSGI, SSTC, Bhilai., Durg, India, 9827164467*

II. Face Extraction Approach

For simplicity, we focused on frontal faces detected using the Viola-Jones face detector [11]. Because of the large number of false positives encountered, eye detection was employed to augment the face extraction [12]. Detected faces were only accepted if they met certain geometric criteria that relate the presence and location of both eyes $e(l), e(r)$, to the face f:

$$C(f, e(l), e(r)) = S(l) + S(r) + S_b$$

$$S(i) = \left(\frac{2f_{size}}{e(i)_{size}} - \alpha\right)^2 + \left(\frac{|f_x - e(i)_x|}{0.01f_{radius}} - \beta_x\right)^2 + \left(\frac{|f_y - e(i)_y|}{0.01f_{radius}} - \beta_y\right)^2$$

$$S_b = \left(\frac{f_{size}}{e(i)_{size}} - \frac{f_{size}}{e(r)_{size}}\right)^2 + \left(\frac{d(e(l), e(r))}{0.01f_{radius}} - \gamma\right)^2$$

The cost function C calculates the score for each combination of face f and left/right eyes $e(l)/e(r)$. The function d measures Euclidean distances. Both faces and eyes are modeled as circles; f_x, f_y refer to the center of the circle representing the face. Through experimental observation using available different face images, appropriate values of $\alpha = 20$, $\beta_x = 25, \gamma = 75$ were chosen, which correspond to the eyes residing in the center of the upper quadrants of the face. The system only accepts the face if the cost is below a threshold, usually a value between 2000-5000 depending on the desired strictness.

Once accepted, a face is identified is located within the circle representing the face. The face is subsequently rotated by aligning the eyes horizontally. The system extracts the face and normalizes it with the procedure developed by [13], namely by a elliptical mask crop, histogram equalization, grayscale conversion and pixel value normalization. The final image is resized to 56x64 pixels. The entire processing of a face from a tagged image takes 1-3 seconds on average.

A. Resulting Face Datasets

We collected different face database and applied the techniques described to automatically construct seven different face datasets. The statistics of each database are described in Table 1 and compared to the different Database of Faces [14]. See Fig. 1 for sample faces of AT&T datasets.

ISSN: 2278 – 7798

Table 1: Statistics for different datasets

Sr. No.	Dataset Name	No. of Person	Total Images
1.	ATT	40	400
2.	ACR	10	282
3.	BCB	22	839
4.	JBK	65	4009
5.	EGO	148	7950
6.	RGA	215	12394
7.	LLP	264	17602
8.	SSB	222	18599
9.	FERET	329	3290
10.	IIT-Kanpur	40	400

III. FACE RECOGNITION APPROACHES

To achieve our goal we have applied different face recognition algorithms to the different face datasets, we evaluate well-known, classical algorithms: PCA, ICA, LDA, and SVMs.

A. Results

When possible, we chose preexisting algorithm implementations that have been tested and accepted by the community. For LDA, we used [17]'s code; for IPCA, we used code accompanying [15]. LIBSVM [19] is a popular SVM library. For ICA, we interfaced to the Architecture I code from [4]. Also, we hybridize SVM with other architectures as in [5]. Except LIBSVM, which is a C library, all implementation work has been done in MATLAB®.

B. Experimental Setup

To ensure a useful interpretation of results, we maintain a consistent setup between experiments. As described earlier, the extraction and preprocessing of faces were identically and automatically performed to construct each dataset. Each dataset was randomly divided into 60% training and 40% testing for validation. Tests showed different selections of random training/testing sets changed the results by less than a percent on average.

Each algorithm is evaluated with several criteria: accuracy, memory usage during the training and testing phases, time spent to train and test, and the size of the model. While implementations differ in optimizations and efficiency, the results provide a realistic estimate. All experiments were performed on an Intel® Core 2 Duo 2.6 GHz computer with 3 GB of RAM. Running times do not include loading images as a real implementation may store them in a database, on a disk. Running times are normalized by the size of the dataset and thus listed in milliseconds (ms) per image. Overall memory (RAM) consumption is sampled at approximately 10 Hz and averaged over the algorithm phase (train or test). Unless the algorithm processes faces incrementally, memory consumed by the batch training images is included in the training phase.

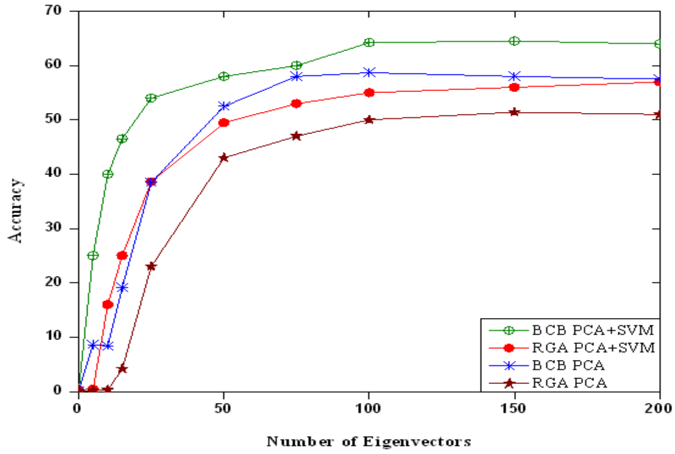

Figure 2: Varying the number of eigenvectors for a small (BCB) and large (RGA) dataset to determine a satisfactory threshold.

Table 2: The number of eigenvectors for a small (BCB) and large (RGA) dataset

Number of Eigenvectors	Accuracy			
	BCB PCA+SVM	RGA PCA+SVM	BCB PCA	RGA PCA
200	64	57	57.5	51
150	64.5	56	58	51.43
100	64.2	55	58.7	50
75	60	53	58	47
50	58	49.5	52.5	43
25	54	38.6	38.5	23
15	46.5	25	19.2	4.2
10	40	16	8.4	0.3
5	25	0.4	8.6	0.2
0	0.3	0.3	0.1	0.1

To choose the number of eigenvectors to use for subspace algorithms, we varied eigenvectors for a small and large dataset, BCB and RGA respectively. Since accuracy levels out around 75-150 in both cases, PCA/ICA use 100 basis vectors. To reduce the effect of illumination in PCA/ICA, the first 10 eigenfaces are discarded based on findings [1]. Individual IPCA methods discard the first eigenface for better accuracy as well.

For each algorithm, we present a tabular form of the system characteristics for each dataset. Tables 3-8 are arranged by evaluation criteria so an interested user can scan and directly compare datasets and approaches.

1) Analysis and Evaluation

As expected, the results show that the system performance varies significantly and accuracy on real data is lower than reported on most face datasets. In a typical face recognition paper focused on accuracy, we would conclude that a SVM approach is best. However, SVMs require half a gigabyte of memory, over half an hour to train or classify, and a half a gigabyte for storage. Without a very large cluster, deploying SVMs to service any significant part of the face datasets population is simply not feasible.

ISSN: 2278 – 7798

Table 3: Accuracy across approaches and datasets (%)

	ATT	ACR	BCB	JBK	EGO	RGA	LLP	SSB	FERET	IITK
PCA	93.1	69.7	57.2	56.5	52.8	50.0	47.1	56.2	89.2	92.3
IPCA	94.4	65.1	56.0	56.2	51.6	49.9	46.9	55.6	91.4	93.3
Ind. IPCA	93.1	69.7	59.3	50.9	48.3	43.4	38.9	44.1	89.2	92.3
ICA	91.3	72.5	56.9	51.4	50.1	45.5	43.0	48.9	88.1	90.3
ILDA	96.3	70.6	66.7	59.4	55.6	52.4	49.3	57.6	92.3	95.3
PCA/VM	97.5	72.5	71.3	64.0	61.7	58.1	55.5	62.4	93.2	95.6
ILDA/SVM	96.3	72.5	63.6	62.3	64.3	60.3	57.4	64.5	92.5	95.3
SVM	96.9	70.6	70.9	66.6	64.3	60.2	58.2	66.0	93.2	95.8

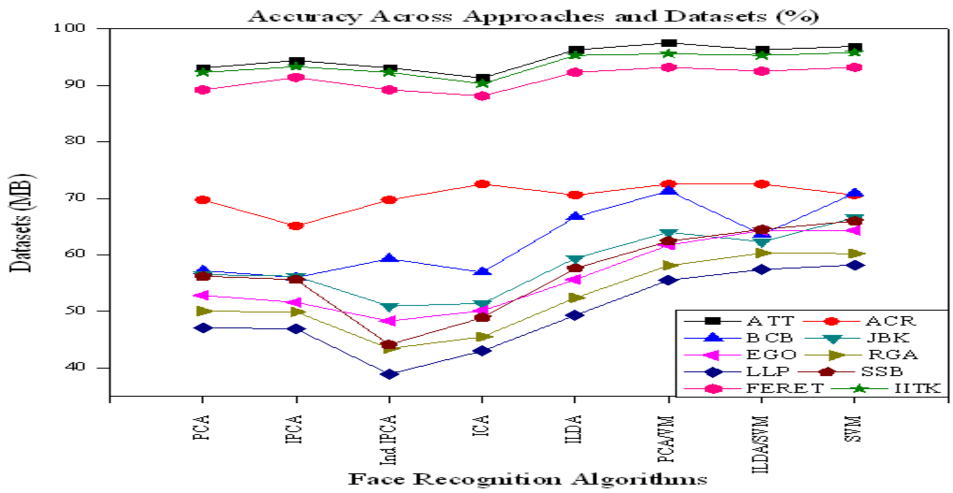

Fig.ure 3: Accuracy across approaches and datasets (%)

Table 3: Training time across approaches and datasets (ms/img)

	ATT	ACR	BCB	JBK	EGO	RGA	LLP	SSB	FERET	IITK
PCA	1.3	1.1	3.6	53.1	54.4	35.7	26.6	25.3	1.1	1.2
IPCA	23.7	16.7	22.4	24.4	22.5	24.4	17.5	24.4	20.4	22.3
Ind. IPCA	1.7	3.8	3.5	3.3	3.5	3.5	3.6	3.7	1.3	1.5
ICA	504	735	261	132	439	231	160	146	423	489
ILDA	2.8	1.4	4.5	66.7	75.8	51.8	37.9	35.7	1.9	2.6
PCA/VM	1.6	1.2	3.9	53.7	55.8	39.0	30.3	29.0	1.4	1.5
ILDA/SVM	1.9	1.3	4.6	67.0	77.3	51.6	40.3	37.4	1.6	1.8
SVM	5.9	4.8	8.4	29.2	53.1	77.2	109	108	5.4	5.7

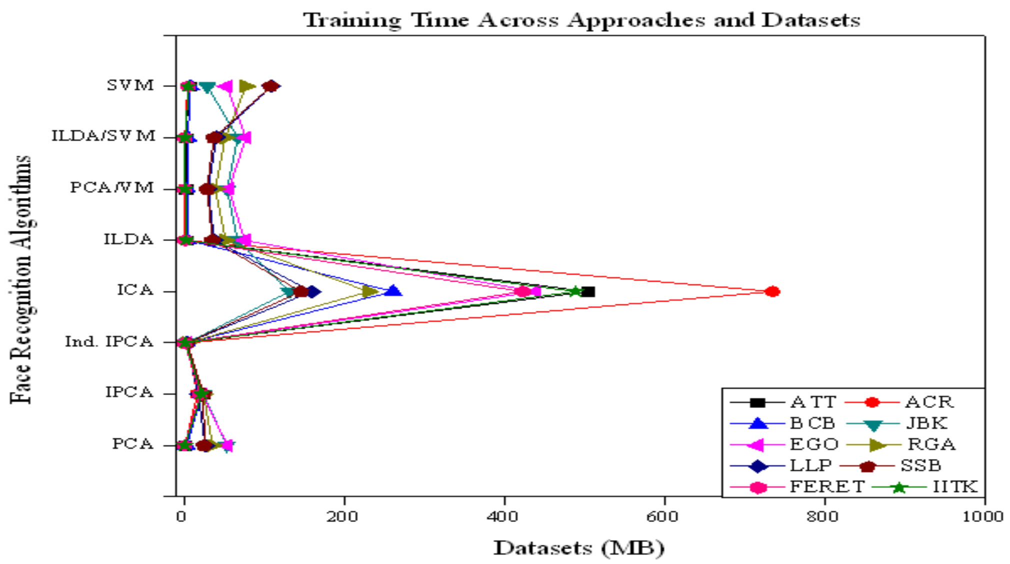

Figure. 4: Training time across approaches and datasets (ms/img)

ISSN: 2278 – 7798

Table 4: Testing time across approaches and datasets (ms/img)

	ATT	ACR	BCB	JBK	EGO	RGA	LLP	SSB	FERET	IITK
PCA	0.1	0.1	0.1	0.2	0.6	1.1	1.5	1.6	0.1	0.1
IPCA	0.3	0.1	0.1	0.2	0.5	0.1	1.5	1.6	0.1	0.2
Ind. IPCA	3.2	0.6	2.0	4.6	12.3	14.2	17	17.3	2.9	3.1
ICA	0.2	0.1	0.1	0.2	0.6	1.0	1.4	1.5	0.1	0.2
ILDA	0.0	0.0	0.1	0.2	0.6	1.1	1.5	1.5	0.1	0.0
PCA/VM	0.2	0.1	0.1	1.4	4.3	8.2	12.5	11.4	0.1	0.2
ILDA/SVM	0.0	0.0	0.1	0.8	4.3	8.0	12.9	12.1	0.0	0.0
SVM	3.3	1.9	6.3	32.1	64.5	105	146	156	2.6	31

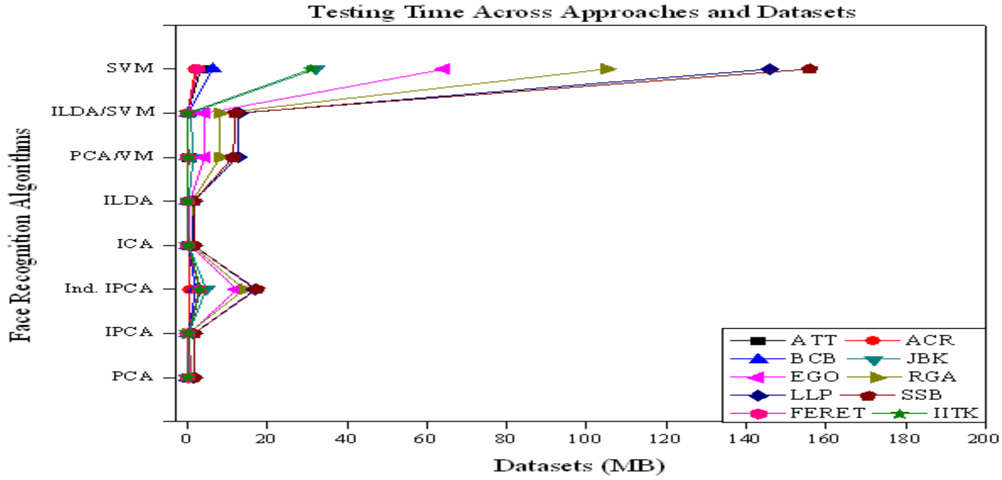

Figure 5: Testing time across approaches and datasets (ms/img)

Table 5: Training memory across approaches and datasets (MB)

	ATT	ACR	BCB	JBK	EGO	RGA	LLP	SSB	FERET	IITK
PCA	7.7	5.0	25.5	241	457	602	772	804	7.1	7.2
IPCA	8.3	6.0	16.0	68.4	133	206	291	308	7.9	8.1
Ind. IPCA	0.4	0.5	1.4	1.1	2.8	19.2	24.8	24.9	0.1	0.2
ICA	21.6	16.0	39.6	281	593	902	1158	1206	20.3	21.1
ILDA	8.7	5.2	27.2	274	564	687	832	859	7.8	8.3
PCA/VM	8.0	5.1	25.3	239	450	580	720	746	7.5	7.9
ILDA/SVM	7.8	5.0	27.4	273	558	668	787	813	7.1	7.4
SVM	8.6	5.6	21.8	129	261	409	581	614	8.3	8.2

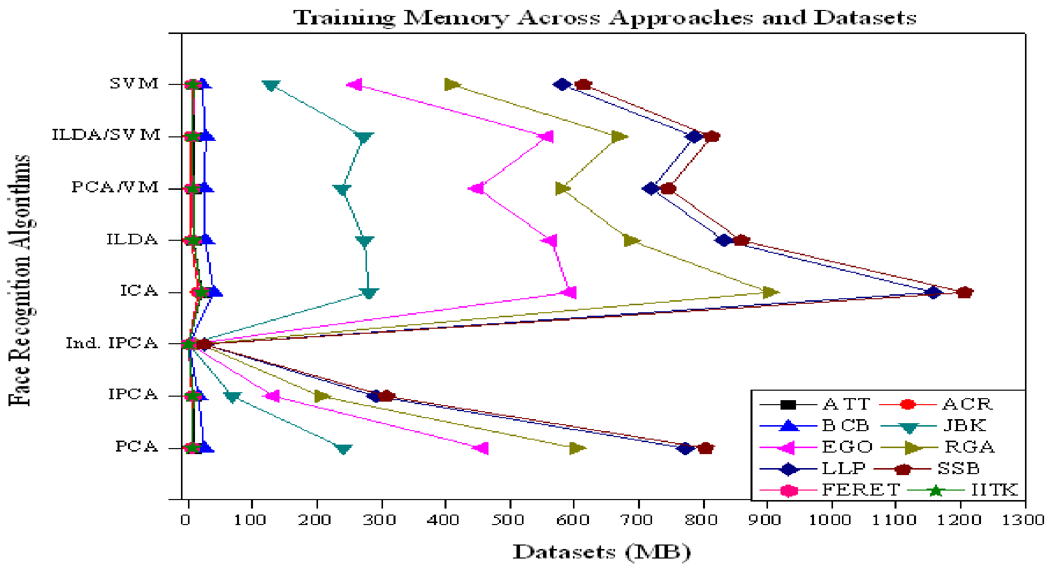

Figure. 6: Training memory across approaches and datasets

ISSN: 2278 – 7798

Table 6: Test memory across approaches and datasets (MB)

	ATT	ACR	BCB	JBK	EGO	RGA	LLP	SSB	FERET	IITK
PCA	2.7	2.6	2.8	4.4	7.2	10.8	15.0	15.8	2.1	2.2
IPCA	2.2	2.3	2.5	4.0	6.4	10.1	13.9	14.7	2.0	2.0
Ind. IPCA	5.7	1.9	3.7	9.5	21.5	31.8	38.3	32.3	5.2	5.2
ICA	2.8	2.7	2.9	5.0	10.4	19.8	30.0	31.9	2.6	2.6
ILDA	1.0	0.2	0.5	1.7	3.4	5.3	7.1	7.5	1.0	1.0
PCA/VM	3.1	2.8	3.9	12.3	35.5	69.8	106	86.5	2.8	3.0
ILDA/SVM	1.2	0.3	0.7	8.0	32.9	65.4	99.4	79.5	1.0	1.1
SVM	11.4	7.0	24.1	118	250	405	580	575	10.6	11.1

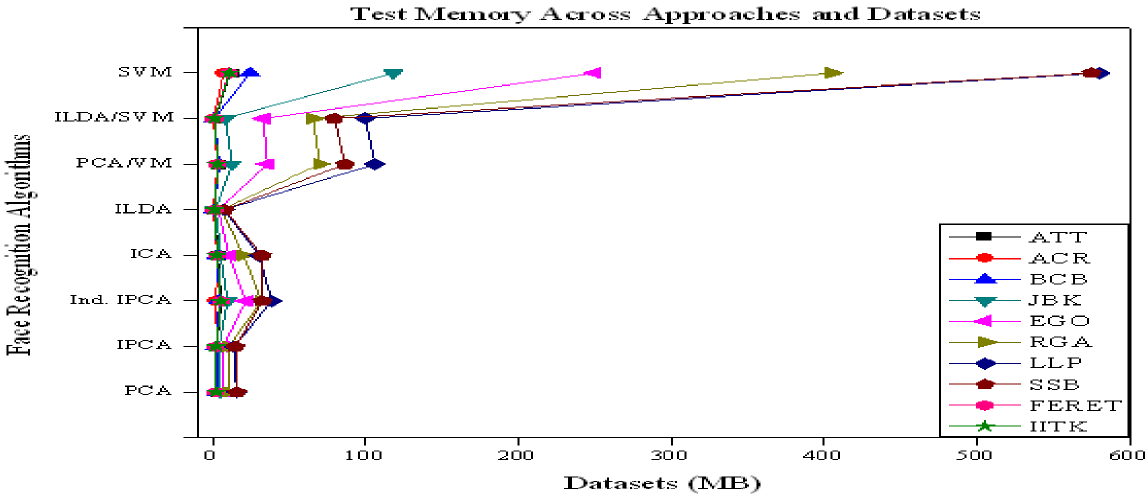

Figure. 7: Test memory across approaches and datasets

Table 7: Model size across approaches and datasets (MB)

	ATT	ACR	BCB	JBK	EGO	RGA	LLP	SSB	FERET	IITK
PCA	2.7	2.6	2.8	4.2	6.0	8.0	10.3	10.7	2.2	2.6
IPCA	2.4	2.3	2.5	3.8	5.4	7.2	9.3	9.7	2.1	2.2
Ind. IPCA	5.7	1.9	3.7	9.4	21.4	31.1	38.2	32.1	5.3	5.6
ICA	2.8	2.7	3.0	4.4	6.2	8.2	10.5	10.9	2.2	2.6
ILDA	1.0	0.2	0.5	1.6	2.5	2.5	2.5	2.5	1.0	1.0
PCA/VM	3.1	2.8	3.5	8.6	18.5	32.4	48.1	43.6	2.9	3.0
ILDA/SVM	1.2	0.3	0.7	4.7	15.6	28.0	41.9	36.5	1.0	1.1
SVM	10.8	7.1	22.7	113	231	367	525	535	9.7	10.3

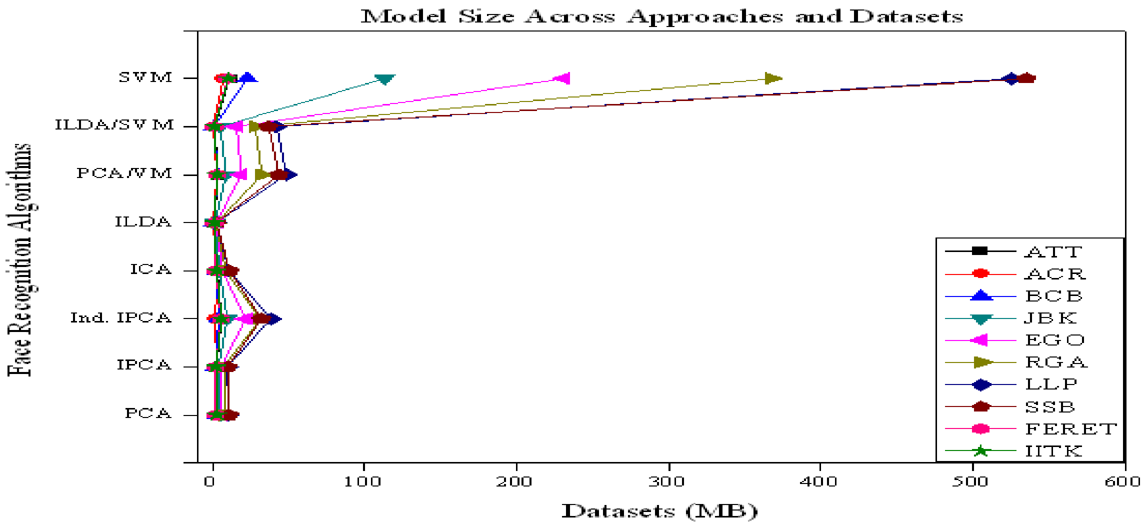

Figure 8: Model size across approaches and datasets

ISSN: 2278 – 7798

By sacrificing accuracy, other approaches are much more feasible. PCA, the traditional “eigenface” algorithm, fares well on the different datasets. As the first 10 eigenfaces encode illumination changes, discarding them significantly improved the accuracy, as shown in the Table 2 for datasets BCB through SSB. While training time and storage are more reasonable than SVMs, the memory requirements for training are high (>500MB). Using the incremental version of PCA [15] in batch mode reduces the training memory requirements by over half while still achieving comparable accuracy. IPCA is not fully incremental in that the mean face of the training images must be available a-priori; incrementally calculating the mean causes a large drop in accuracy.

Individual IPCA is unique because it is the only algorithm that is completely incremental. An eigenspace is built for each person, with the mean face calculated incrementally as each image is processed. A new face can be trained on by creating or updating the eigenspace using IPCA. Furthermore, training is extremely fast. However, the accuracy is 10-20% less than SVMs and recognition is slower because of the projections into multiple eigenspaces.

ICA improves the accuracy of straight PCA by significantly increasing the computation times and memory requirements. However, when discarding the top 10 eigenfaces for each algorithm, PCA performs better, perhaps because the top PCA eigenfaces encode illumination variations. ICA’s learning logistic function [4] executes a fixed number of iterations, leading to a bottleneck and the counter-intuitive result that larger datasets exhibit a lower training time per image.

LDA [13] was a top performer on the small datasets, but scaled so poorly we did not run it on the three largest datasets. Since LDA is infeasible on a large system, we instead list the batch-mode ILDA [17] results. ILDA’s accuracy is slightly greater than PCA and the run times are more favorable than ICA. Additionally the model size is smaller because of the decreased eigenspace dimensionality of LDA.

Perhaps the best compromise is the combination of PCA or ILDA and SVMs. Instead of the expensive computation of training SVMs on the full images and generating very large model files, we can use the eigenspace representation of the faces to greatly reduce the complexity while only sacrificing 2-4% accuracy. However even in this scenario, it takes roughly five minutes and a gigabyte of memory for a normal user.

IV. CONCLUSION AND FUTURE WORK

In conclusion, we have utilized a new, real-world source of images to test a variety of algorithms for holistic performance with respect to the potential application of face recognition. SVM and ILDA methods yield fair ~65% accuracy at the cost of high computation and memory requirements. Further, they must be completely retrained with each new image. Likewise, the Individual IPCA approach is ideally suited to a real-world implementation, but yields a low accuracy intolerable to most users. However if the scope was scaled back from full autonomy, Individual IPCA could aid in tagging by automatically detecting faces and suggesting most likely identities.

Future work includes the exploration of iterative SVMs and its parameter space to yield more optimal results in memory consumption. In addition, more recent approaches such as 3D face reconstruction may correct pose problems inherent to the algorithms presented.

The detection and extraction of faces is tightly constrained by Viola-Jones and other parameters, which further reduces the accuracy of recognizing faces because they are rejected by the face extraction stage. Further work should add face detection methods of increased accuracy and a filtering step to remove outliers.

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," ACM Computing Surveys, vol. 35, pp. 399-458, 2003.
[2] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," in IEEE CVPR, 1991, pp. 586-591.
[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," in IEEE TPAMI. vol. 19, 1997, pp. 711-720.
[4] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face Recognition by Independent Component Analysis," IEEE Transactions on Neural Networks, vol. 13, pp. 1450-1464, 2002.
[5] G. Guo, S. Z. Li, and K. Chan, "Face Recognition By Support Vector Machines," Image and Vision Computing, vol. 19, pp. 631-638, 2001.
[6] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms," IEEE TPAMI, vol. 22, pp. 1090-1104, 2000.
[7] Y. Hu, D. Jiang, S. Yan, and L. Zhang, "Automatic 3D Reconstruction for Face Recognition," in IEEE FG, 2004, pp. 843-848.
[8] Facebook, "Facebook | Statistics," 2008. http://www.facebook.com/press/info.php?statistics.
[9] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," University of Massachusetts, Amherst 2007.
[10] J. Ruiz-del-Solar and P. Navarrete, "Eigenspace-Based Face Recognition: A Comparative Study of Different Approaches," IEEE Trans. on Sys., Man. & Cyb. C., vol. 35, p. 315, 2005.
[11] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in IEEE CVPR, 2001, pp. 511–518.
[12] M. Castrillón-Santana, O. Déniz-Suárez, L. Antón-Canalís, and J. Lorenzo-Navarro, "Face and Facial Feature Detection Evaluation," in VISAPP, 2008.
[13] D. S. Bolme, J. R. Beveridge, M. Teixeira, and B. A. Draper, "The CSU Face Identification Evaluation System: Its Purpose, Features, and Structure," in ICVS Graz, Austria, 2003.
[14] A. T. L. Cambridge, "The Database of Faces," http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
[15] J. Weng, Y. Zhang, and W. S. Hwang, "Candid Covariance-Free Incremental Principal Component Analysis," IEEE TPAMI, vol. 25, pp. 1034-1040, 2003.
[16] X. Liu, T. Chen, and S. M. Thornton, "Eigenspace Updating for Non-Stationary Process and Its Application to Face Recognition," Pattern Recognition, vol. 36, pp. 1945-1959, 2003.

ISSN: 2278 – 7798

[17] T. K. Kim, S. F. Wong, B. Stenger, J. Kittler, and R. Cipolla, "Incremental Linear Discriminant Analysis Using Sufficient Spanning Set Approximations," in IEEE CVPR, 2007, pp. 1-8.

[18] B. Heisele, P. Ho, and T. Poggio, "Face Recognition with Support Vector Machines: Global versus Component-Based Approach," in ICCV. vol. 2 Vancouver, Canada, 2001, pp. 688–694.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Parul Jain has completed her B.E. in Computer Science & Engineering Branch in 2007 with 1st Division from Shri Shankacharya College of Enginnering & Technology, Bhilai (Pt. Ravi Shankar Shukla University, Raipur). She is pursuing his M.E. in Computer Technology and Application from Shankaracharya Group of Institution, Bhilai.

Nilay Jain has completed his B.TECH. in Computer Science & Engineering Branch in 2012 with 1sst division from Kavikulguru Institute Of Technology & Sciences(RTMNU).He is currently working at Tech Mehendra,Pune as a software engineer

Rohit Raja is working as a Professor in Shankaracharya Group of Institutions, Bhilai(C.G.). He has received his M.E. degree in Computer Technology and Application in 2009 from SSGI, Bhilai (C.G) and B.E. degree in Computer Science & Engineering in 2004 from R.I.T (R.S.U) Raipur (C.G.). Presently he is pursuing his Ph.D. from C.V Raman University, Bilaspur. He has published more than 15 Papers in various reputed Journals, National & International Conferences.

Corresponding Address-
Parul Jain
2/2, Vasundhara Nagar (North)
Near Ispat Club, Bhilai-3
District-Durg (C.G.)-490021

ISSN: 2278 – 7798