A Comparative Study of Dimension Reduction Techniques: PCA, ICA, LDA and NMF in Facial Recognition using SVM Classifier

CS7IS2 Project (2019-2020)

Prateek Tulsyan, Rushikesh Vinayak Joshi, Shubham Dhupar, Mrinal Jhamb

tulsyanp@tcd.ie, joshiru@tcd.ie, dhupars@tcd.ie, jhambm@tcd.ie

Abstract. This paper provides a comparison between Principle Component Analysis (PCA), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF) and Linear Discriminant Analysis (LDA) while performing face recognition using the SVM classifier. The metrics used are accuracy of the prediction and the training time the classifier needs. This paper also explains the underlying differences in each of the four-dimension reduction methods.

1 Introduction

The advancements in contemporary technologies imply that Face Recognition is becoming increasingly important application of Artificial Intelligence. Amalgamating AI with Computer Vision enables commercial systems to perform face recognition in real-time speed. Drilling down into these systems reveals that there's a two-step process involved in face recognition; Subspace projection (dimensionality reduction) followed by performing Classification.

Furthermore, it has been observed that algorithms for dimensionality reduction largely affect the accuracy of face recognition systems. For the scope of this paper, four prominent algorithms, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF) and Linear Discriminant Analysis (LDA) have been chosen for comparison. These will be used in conjunction with the SVM classifier to perform face recognition on the 'labelled faces in the wild[1]' dataset.

In addition, these algorithms have been handpicked due to the contradicting results available in numerous research papers comparing them. For instance, Bartlett [1, 2], Liu and Wechsler [3] and, Yuen and Lai [4] argue that in the domain of face recognition, ICA outperforms PCA. However, Baek et al. [5] in their works conclude that PCA is better than ICA. Moreover, Moghaddam [6] settles that there's no significant difference in the performance of these algorithms.

[1] http://vis-www.cs.umass.edu/lfw/

Of course, there is always a trade-off between accuracy and speed. Hence, the total time taken for training the classifier has also been chosen as a metric for comparing these algorithms. The best choice of algorithm would find a balance between these two metrics. Meanwhile, the classifier (SVM) is kept common to avoid any unfair advantages while the results are being compared.
Briefly, this paper performs face recognition using SVM classifier along with PCA, ICA, NMF and, LDA. The accuracy and the training time have been chosen as metrics to evaluate and compare these algorithms, and results have been published to conclude the research. These results hope to put an end to the open-ended question of the best dimensionality reduction algorithm to be used in a facial recognition application.
Finally, following is a roadmap for this paper. Section 2 discusses the existing literature available in the field of dimensionality reduction methods in the domain of face recognition. Section 3 provides basic theoretical knowledge of PCA, ICA, NMF, LDA and the SVM classifier. Section 4 lays down the steps taken to complete the research along with the results obtained. Lastly, Section 5 concludes with practical recommendations.

2 Related Work

Research in the field of face recognition started back in 1960's [7]. With the emergence of appearance-based work in 1980's and 1990's, most current face recognition techniques emerged. PCA was first applied to face images by Kirby and Sirovich [8, 9]. They showed that using it, images can be compressed to their reconstructions with minimum mean squared error between them, i.e. it is an optimal compression scheme. PCA for face recognition was also used by Turk and Pentland [10]. In fact, they popularized its use. In their work compression to compressed subspace (eigenspace) was performed from a database of face images by computing a set of subspace basis vectors (eigenfaces) using PCA. The success of this method leads to the popularization of matching images in the compressed subspace. PCA produces spatially global feature vectors.
Researchers also put in efforts to create techniques that create spatially localized feature vectors, in the hopes that they would implement recognition by parts. ICA is the most general way to generate spatially localized features, which does it by producing statistically independent basis vectors [11]. One other method to generate localized feature vector is Non-negative Matrix Factorization (NMF) [12]. ICA can also be used to create feature vectors that uniformly distribute data samples in subspace [1, 13]. This is conceptually different from what ICA is believed to do, i.e. rather than producing features vectors which are spatially localized, it produces feature vectors which produce very fine distinction between images in order to spread the samples in subspace. To keep up with the terminology, we refer to former as architecture I, and the latter as architecture II.
There are certain techniques which are formed by the combination of local linear subspaces. For instance, mix local PCA which is used to compress face data by Kambhatla and Leen [14], and a mixture of factor analyzers produced by Frey [15]. Although they have not yet been applied to face recognition, Tipping and Bishop, and

Lee provide an algorithm for optimizing mixture models of PCA and ICA subspaces respectively [16, 17].
Alternative to these algorithms is a supervised learning algorithm knows as Fisher's linear discriminant analysis (LDA, a.k.a. "fisherfaces") [18]. Its goal is to produce N-1 basis vector in order to maximize and minimize the intra and inter class distances between N-classes of N-class problem. Even though LDA differ PCA in terms that one is supervised learning algorithm and the other is unsupervised but when the data is labeled then either of the two can be used and in these cases LDA has been compared to PCA [7, 14, 6, 19]. One common characteristic of both PCA and LDA is that they produce spatially global feature vectors.

3 Problem Definition and Algorithm

This paper tries to implement numerous dimension reduction methods in conjunction with SVM classifier to aid facial recognition process. The dimensionality reduction methods used are Principle Component Analysis (PCA), Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF) and, Linear Discriminant Analysis (LDA).

3.1 Principle Component Analysis (PCA)

PCA was introduced by Karl Pearson in 1901 and is mostly used for EDA and predictive models. It is an eigenvector based multivariate analysis and is the simplest in its kind. It is basically reducing the dimensionality of the data in a way that it explains the variation (major features) in the best possible way to make it more understandable. Say there are $n * n$ dimensional images i.e. n^2 dimensions (pixels). When such a multivariate dataset is visualized as set of coordinates in high dimensional data space, PCA can provide with the lower dimensional picture (eigen faces) of the object when viewed from its most informative viewpoint.
Mathematically by definition PCA is a procedure that converts M correlated into a set of K uncorrelated variables called principal components using orthogonal transformations (where $K \leq M$).
Since PCA selects the principle components which show the direction of data and each proceeding component shows less direction and more noise. Hence, first few principle components are enough to represent the original data.
Once we have found the k principle components then image in the dataset or incoming new data can be represented as the linear combination (weighted sum) of k components, i.e. $W * K$ (W is a matrix of weight and k is a matrix of eigenfaces).
When we represent the data this way it reduces the number of values needed to recognize it. This makes the process faster and more error free because this discards the noise in the dataset. It is done by eigenvalue decomposition of a data covariance matrix. Results are usually discussed in form of principle components (how much of each of the k principle components) and loadings (weight).

3.2 Independent Component Analysis (ICA)

PCA is about finding correlation and the way that it does this is by maximizing variance, which in turn gives us the ability to reconstruct. ICA, on the other hand, tries to maximize independence. This when simply put means that it tries to find linear transformation of feature space into new feature space such that each of the individual new features are mutually independent. The new components produced are perpendicular to each other and there is no information loss between moving from observables to independent components.
There are certain hidden variables (independent components) which are random and mutually independent (value of one doesn't tell us anything about the value of other). Then there are known observable variables which are given rise to by the linear combination of these hidden variables. The job of unobservable learner in this case is to find the hidden variables given the observables (under the assumption that hidden variables are independent of one another). A classic example of this is the Blind Source Separation problem.

	PCA	**ICA**
Mutually Orthogonal	Yes	No
Mutually Independent	No	Yes
Maximal Variance	Yes	No
Maximal Mutual Interaction	No	Yes
Ordered Features	Yes	No
Bag of Features	Yes	Yes

Table 1: *Difference in PCA and ICA*

PCA is directional, and when it comes to face recognition problem it captures brightness, avg. (Eigen) face. Since PCA is based on global orthogonality hence it captures global features.
ICA on the other hand in face recognition problem captures nose, eye selectors, mouth selectors, hair selectors. Since ICA is based on local search hence it finds parts of face.

3.3 Non-negative Matrix Factorization (NMF)

Basic idea of NMF is to reduce a given matrix into two matrices which are both easier to work with and which when multiplied produce the original matrix. For a matrix V with $n * m$ dimensions the NMF will try and decompose it into two matrices W and H with $n * r$ and $r * m$ dimensions respectively, such that:

$$W * H = V \quad \text{and} \quad W_{ij}, H_{ij}, V_{ij} \geq 0$$

This problem cannot be solved analytically so it is generally solved numerically. NMF is relatively a new way of reducing dimensionality of our data into a linear combination of bases which in turn is necessary for machine learning algorithms for ease of

computation. In the image example it's very difficult to consider all the pixels each time the image is handled so it is better to reduce the image into few representative pixels. Due to fact that NMF has this non-negative constraint it can be used to can be used to depict data with non-negative features. It is like PCA, but weights are only allowed to have positive values. This is compatible with the intuitive notion of combining parts to form the whole. Sparse bases and spares weightings are constructed by NMF assuming that there is an underlying structure in the data.
When it comes face recognition NMF forms bases that are parts of the face. The bases in this case are mostly empty and weighting matrix is also sparse, i.e. all parts are not used to form the image. On the contrary PCA has both its matrices densely populated and forms bases of positive and negative pixels and the weight blends them together.

3.4 Linear Discriminant Analysis (LDA)

As the name suggests, LDA tries to identify the best discriminator for the classes. Mathematically, LDA tries to create a linear combination of the independent features which maximizes the interclass mean difference. Let c be the number of classes, n_j be the number of samples in class j and, μ_j be the mean for class j. Then, for given sample x_i^j and mean of all classes μ, we define the within-cluster distance S_w,

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{n_j} (x_i^j - \mu_j)(x_i^j - \mu_j)^T$$

And, the between-cluster distance S_b as,

$$S_b = \sum_{j=1}^{c} (\mu_j - \mu)(\mu_j - \mu)^T$$

The goal here would be the maximize S_b while minimizing S_w. This can be done by either maximizing $det\,|S_b| / det\,|S_w|$ or minimizing $\left(det\,|S_b| / det\,|S_w|\right)^{-1}$.

3.5 Support Vector Machine

Support Vector Machines work on the very idea of decision planes which states decision boundaries. Members belonging to different classes are separated by using decision planes. An example is shown below. The objects belong either to class red or they belong to the green one in this example. Both the classes are separated by the separating line such that all objects to the right are red and to the left are green. The new object falling to the right will be labelled red and to the left will be labelled green (or classified as RED should it fall to the left of the separating line).

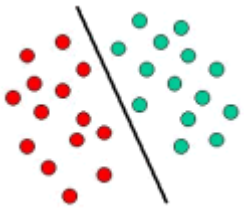

Figure 1: *Linear Classifier*

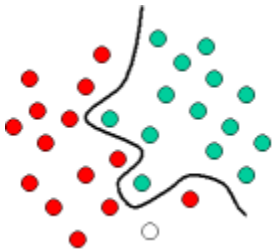

Figure 2: *Non-linear Classifier*

The example shown (Figure 1) is of linear classifier i.e., it separates the set of objects into red and green classes with a line. Most of the classification tasks we come across are complex than this one and often more complex decision structures are required to classify them properly i.e., correctly classify new objects (test cases) based on the examples that are available (train cases). This is shown in the image above (Figure 2). As compared to the last example the complete separation of red and green classes is not possible unless a curve is used. These type of classification tasks which use drawing separating lines are called hyperplanes. Support Vector Machines are particularly suited to handle such tasks.

Reasons for choosing these classifiers:

- Previously used.
- Gives the option to compare with previous models.
- These use different techniques and so it is reasonable to detect whether different defects are detected by each and whether the prediction consistency is different among the classifiers.

4 Experimental Results

The paper implements dimension reduction methods in conjunction with SVM classifier using sklearn[2] package in python to aid facial recognition process.

4.1 Methodology

Evaluation Criteria: Results are being evaluated based on the accuracy of prediction, on training (75% images) as well as testing (25% images) data, using each of the dimension reduction techniques, namely PCA, ICA, LDA and NMF when used in conjunction with the SVM classifier.

Data: Labelled dataset named 'labelled faces in the wild' is used which consists of 5 users with more than 100 faces each.

---

[2] https://scikit-learn.org/

Modus Operandi:

⇒ The training data available at hand was already greyscale. So, it didn't require any tweaking.

⇒ Dimensionality is reduced using PCA (L1 distance matrix based), ICA (architecture I), LDA and NMF. Four models were trained on 50, 100, 150 and 200 eigenfaces/fisherfaces for each of the dimensionality reduction technique.

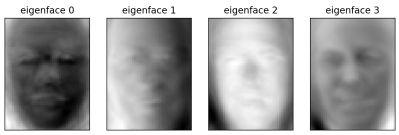

Figure 3: *Eigenfaces for PCA*

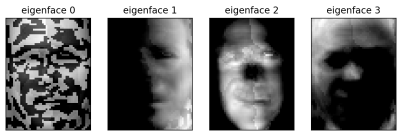

Figure 4: *Eigenfaces for ICA*

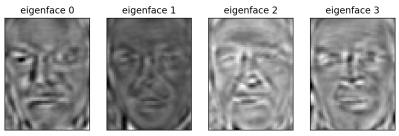

Figure 5*: Eigenfaces for NMF*

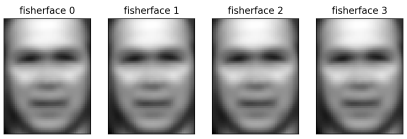

Figure 6: *Fisherfaces for LDA*

As expected, it is quite evident that the eigenfaces generated by PCA captures global features like brightness, avg. eigenfaces and that each next eigenface has less information and more noise. ICA in accordance with theory captured various facial elements in each of its eigenface (localized features like nose, eye selectors etc.). NMF also performed as expected and produced sparsely information rich eigenfaces as per its inherent property due to sparsely positively populated decomposed matrices. As per the theory it is evident that scatter between classes is greater in fisherfaces than in eigenfaces.

⇒ SVM classifier is fit to this training data using each set of eigenfaces/fisherfaces and then accuracy of SVM classifier is compared on the training data and test data based on the accuracy of prediction and time taken for an SVM classifier to be trained on it.

4.2 Results

Training dataset of 855 images at hand after being reduced to 50 eigenfaces/fisherfaces using PCA, ICA, NMF and, LDA were compared for their accuracy of prediction and for the time it took for the SVM classifier to run on them. The number of eigenfaces/fisherfaces were also increased gradually (step size 50) to get a more comprehensive view about the performance of dimensionality reduction algorithm with respect to number of eigenfaces/fisherfaces considered.

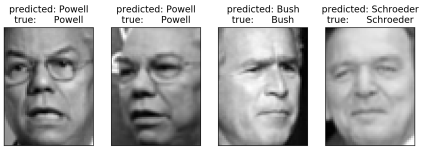

Figure 7: *Prediction for PCA*

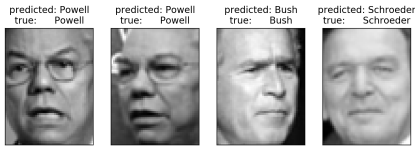

Figure 8: *Prediction for ICA*

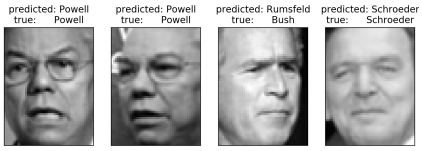

Figure 9*: Prediction for NMF*

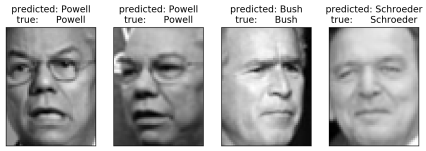

Figure 10: *Prediction for LDA*

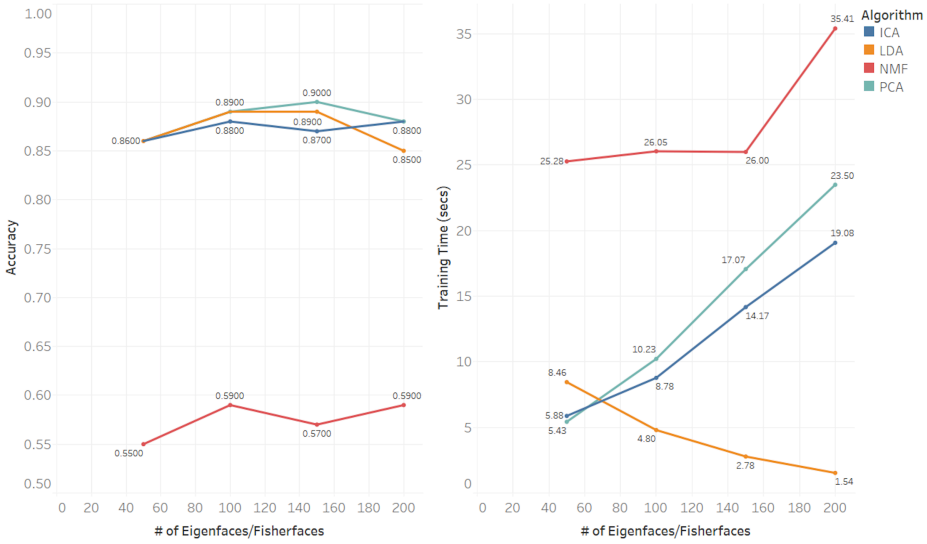

Figure 11: *Performance and Training Time of SVM classifier when trained on varying number of eigenfaces/fisherfaces*

4.3 Discussion

The results show that performance of SVM classifier when trained with dimensionally reduced data using NMF is the worst in terms of accuracy of prediction and training time for every number of eigenfaces considered. The performance of model trained on dimensionally reduced data using ICA, PCA and LDA is almost same when it comes to accuracy of prediction and varies between 85% and 90% but there is a huge difference between the time taken by classifier to train on them.

When it comes to accuracy, model trained using 150 eigenfaces generated by PCA performs the best with prediction accuracy of 90% and in terms of time taken to train the model LDA performs the best for 200 fisherfaces taking around 1.54 seconds and its behaviour was peculiar as the time taken to train the model kept on decreasing with increasing number of fisherfaces.
PCA and ICA perform almost equally well in terms of prediction accuracy and time taken to train the model with PCA edging ICA slightly this deduction is not in sync with the data in [20, 21].

5 Conclusions

Comparison between PCA, ICA, NMF and, LDA is complex because of differences in underlying tasks, architectures and distance metrics. This paper explores the performance of PCA, ICA, NMF and LDA based on their accuracy of prediction when used in conjunction with SVM classifier and suggests that PCA, ICA and LDA perform equally well, rather PCA edges over ICA & LDA slightly in terms of accuracy of prediction which is in contrast with the literature which says that ICA edges over PCA slightly when it comes to facial identity recognition [20, 21].
Based on the trade-off between accuracy of prediction and the time it took for the model to be trained on the generated eigenfaces/fisherfaces, LDA performed best for 150 fisherfaces and had accuracy of 89% and training time of 2.78 seconds.
This paper is limited in its scope because it compares the algorithms only based on prediction accuracy and the time it took for the SVM classifier to be trained. This can be extended in future by finding the factors which influence the performance of algorithms and the time it takes for the classifiers to be trained on these eigenfaces/fisherfaces. Numerous variations of these algorithms which are based on varied architectures (in ICA) and the distance matrices (in PCA) can be considered. These can further be explored and compared based on underlying pattern of the data distribution.
Also, this work can be extended to check if the performance of dimensionality reduction algorithms in terms of accuracy and training time varies with change in the classification algorithm like KNN, Naïve Bayes etc.

References

1. M. S. Bartlett, H. M. Lades, and T. J. Sejnowski, "Independent component representations for face recognition", SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III, San Jose, CA, 1998.
2. M. S. Bartlett and T. J. Sejnowski, "Viewpoint invariant face recognition using independent component analysis and attractor networks," in Neural Information Processing Systems - Natural and Synthetic, vol. 9, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA: MIT Press, 1997, pp. 817-823.
3. C. Liu and H. Wechsler, "Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition," presented at International Conference on Audio and Video Based Biometric Person Authentication, Washington, D.C., 1999.

4. P. C. Yuen and J. H. Lai, "Independent Component Analysis of Face Images," presented at IEEE Workshop on Biologically Motivated Computer Vision, Seoul, 2000.
5. Baek, Kyungim & Draper, Bruce & Beveridge, J. & She, Kai. (2004). PCA vs. ICA: A comparison on the FERET data set. Proceedings of the Joint Conference on Information Sciences.
6. B. Moghaddam, "Principal Manifolds and Bayesian Subspaces for Visual Recognition," presented at International Conference on Computer Vision, Corfu, Greece, 1999.
7. W. W. Bledsoe, "The model method in facial recognition," Panoramic Research, Inc., Palo Alto, CAPRI:15, August 1966.
8. M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 103-107, 1990.
9. L. Sirovich and M. Kirby, "A Low-dimensional Procedure for the Characterization of Human Faces" Journal of the Optical Society of America, vol. 4, pp. 519-524, 1987.
10. M. Turk and A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, vol. 3, pp. 71-86, 1991.
11. M. S. Bartlett, Face Image Analysis by Unsupervised Learning: Kluwer Academic, 2001.
12. D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 201, pp. 788-791, 1999.
13. M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face Recognition by Independent Component Analysis" IEEE Transaction on Neural Networks, Vol 13, pp. 1450-1464, 2002.
14. N. Kambhatla and T. K. Leen, "Dimension Reduction by Local PCA," Neural Computation, vol. 9, pp. 1493-1516, 1997.
15. B. J. Frey, A. Colmenarez, and T. S. Huang, "Mixtures of Local Linear Subspaces for Face Recognition" presented at IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998.
16. M. E. Tipping and C. M. Bishop, "Mixtures of Probabilistic Principal Component Analysers" Neural Computation, vol. 11, pp. 443-482, 1999.
17. T.-W. Lee, T. Wachtler, and T. J. Sejnowski, "Color oppency is an efficient representation of spectral properties in natural scenes," Vision Research, vol. 42, pp. 2095-2103, 2002.
18. D. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, pp. 831-817, 1996.
19. H. Moon and J. Phillips, "Analysis of PCA-based Face Recognition Algorithms," in Empirical Evaluation Techniques in Computer Vision, K. Boyer and J. Phillips, Eds. Los Alamitos, CA: IEEE Computer Society Press, 1998.
20. M. S. Bartlett, G. Donato, J. R. Movellan, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Image representations for facial expression coding" in Advances in Neural Information Processing Systems, vol.12. Cambridge, MA: MIT Press, 2000, pp. 886-892.
21. L. Sirovich and M. Kirby, "A Low-dimensional Procedure for the Characterization of Human Faces" Journal of the Optical Society of America, vol. 4, pp. 519-524, 1987.