COP6930: Sparse Coding and its Applications

Project Report: Face Recognition using Weighted Sparse Representation

Raheen Mazgaonkar (UF ID: 4714 4316)
Shikha Mehta (UF ID: 4851 9256)

Problem Statement

Face recognition is one of the most intensively investigated topics in biometrics. Law enforcement agencies over the world use biometric software to scan faces in CCTV footage and identify persons of interest in the field. Border control deployments use face recognition to verify the identities of travelers. We can also find many consumer applications in our day-to-day lives based on facial recognition, for example selfie-based authentication on smartphones [1].

Like in other fields in pattern recognition, the identification of faces in face recognition has been addressed through different approaches according to the chosen representation and the design of the classification method. Our project will be focused on a Weighted Sparse Representation based Classification (WSRC) [2]. This work involves studying and implementing a Sparse Representation based Classification (SRC) [3], and improving it to integrate the locality structure of the data. WSRC does a better job in identifying human faces when compared with Nearest Neighbors, Nearest Feature Subspace, and SRC alone. As part of this project, we validate this conclusion and provide the results for the same by conducting a series of experiments.

Core Idea

Face recognition using sparse representation was initially studied in [3]. This Sparse Representation based classification (SRC) is a generalization for popular classifiers such as nearest neighbor (NN) and nearest subspace(NS) [4]. Nearest neighbor classifies images based on the best representation in terms of a single training sample. In contrast, nearest subspace classifies based on the best linear representation in terms of all the training samples in each class. The method used in SRC strikes a balance between both these approaches. It tries to identify the minimal number of training samples required to represent each test image. Furthermore, for face detection, it computes sparse representation for each class of image. For this it uses the property that, if the required solution is sufficiently sparse then the solution to the l0 minimization problem is equal to the solution to the l1 minimization problem [5].

In [6], this idea was further extended, and Weighted Sparse Representation was used for classification. In the previous work l1 minimization was used for creating sparse representation, but this can lead to loss of locality information in lower dimensions which is important in certain applications. Hence, to overcome this drawback, a more robust technique was suggested in which both sparsity and data locality structure were integrated. Additionally, this method also preserves the similarity between the test sample and its neighboring training data.

Literature Review

1. l_0 norm

$$\|x\|_0 = \#(i|x_i \neq 0) \tag{1}$$

that is a total number of non-zero elements in a vector. For finding the sparsest solution for y=Ax, the sparsity of the coefficient vector can be measured by counting the number of non-zeroes in a vector, which is l_0 - norm. The l_0 - norm is non-convex. It is known that non-convex optimization problems are NP hard.

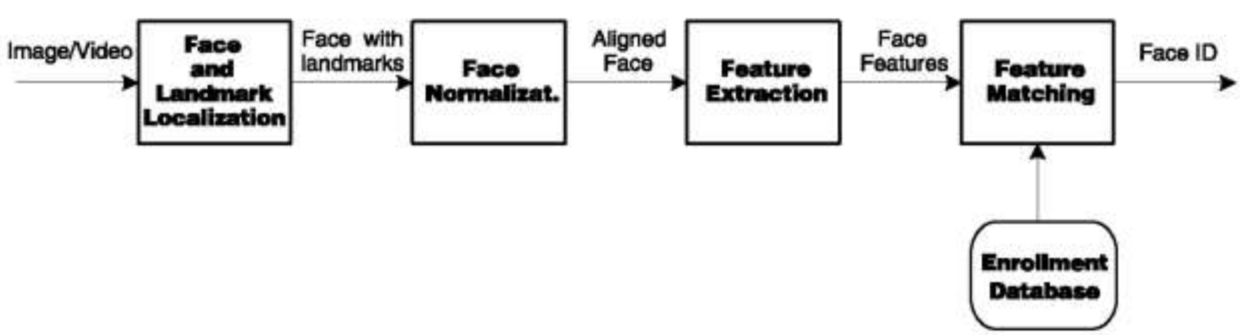
2. l_1 norm

$$\|x\|_1 = \sum_i |x_1| \tag{2}$$

3. l_1 minimization

l_1 - minimization is a convex function close to l_0 - minimization. Literature has shown, if the solution is sufficiently sparse, l_1 - minimization is equivalent to l_0 - minimization.

4. Steps involved in Face Recognition



Face and Landmark Localization

Facial landmark is defined as the detection and localization of certain key points on the face which have a subsequent task focused on the face, like animation, face recognition, gaze detection, face tracking, expression recognition, gesture understanding etc.

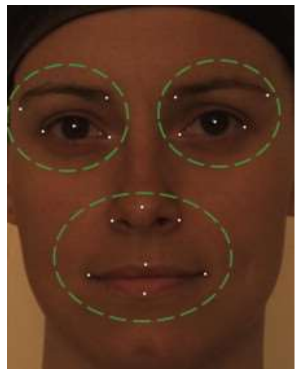


Figure 1: Landmark Localization

Face Normalization

The objective of face normalization is to reduce the effect of useless, interferential and redundant information such as background, hair, cloth etc. to enhance the recognition process.

Feature Extraction

Facial feature extraction is the most crucial step in face recognition. Current face recognition systems can perform very well in controlled environments e.g. frontal face recognition, where face images are acquired under frontal pose with strict constraints as defined in related face recognition standards. However, in unconstrained situations where a face may be captured in outdoor environments, under arbitrary illumination and large pose variations these systems fail to work. The goal of feature extraction is to find a specific representation of the data that can highlight relevant information. Let's know in detail about Eigenfaces, Fisherfaces and Randomfaces as we have used these feature extraction methods in our experiments.

EigenFaces

Face images are projected into a Feature Space ("Face Space") that best encodes the variation among known face images. The face spaces is defined by the eigenfaces, which are the eigenvectors of the set of faces.

Calculation of EigenFaces:

1. Calculate average face : v

2. Collect the difference between training images and average face in matrix A (M by N), where M is the number of pixels and N is the number of images.

$$A = \left[u_1^1 - v, \quad ..., \quad u_n^1 - v, \quad ..., \quad u_1^P - v, \quad ..., \quad u_n^P - v\right] \tag{3}$$

3. The eigen vectors of covariance matrix C (M by M) gives the eigenfaces. M is usually big, so this process is time consuming.

$$C = AA^T \tag{4}$$

If the number of data points is smaller than the dimension (N<M), then there will be only N-1 meaningful eigenvectors.
Instead of directly calculating the eigenvectors of C, we can calculate the eigenvalues and the corresponding eigenvectors of a much smaller matrix L (N by N).

$$L = A^T A \tag{5}$$

If λ_i are the eigenvectors of L then $A\lambda_i$ are the eigenvectors for C. The eigenvectors are in the descending order of the corresponding eigenvalues.
The training face images and the new face images can be represented using as linear combination of the eigenfaces. we have a face image u:

$$u = \sum_i a_i \Phi_i \tag{6}$$

Since the eigenvectors are orthogonal:

$$a_i = u^T \Phi_i \tag{7}$$

Fisherfaces

1. Based on Fisher's Linear Discriminant Analysis(LDA). LDA seeks directions that are efficient for discrimination between the data.

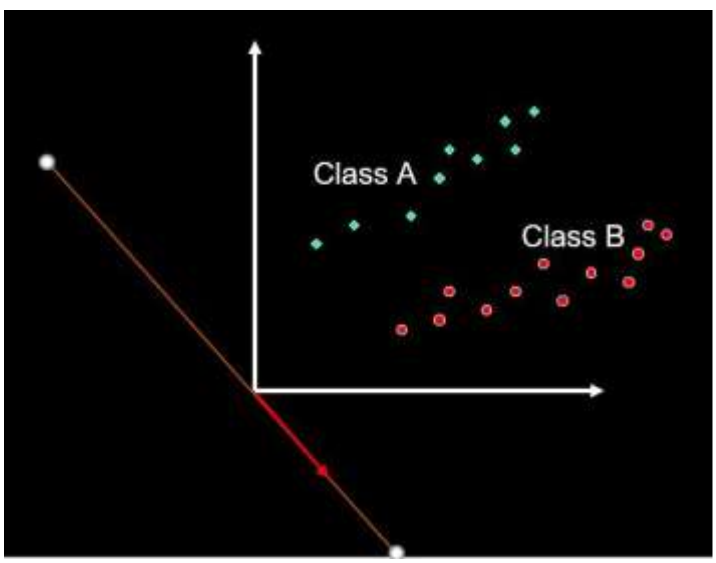


Figure 2: Fisher LDA

2. LDA maximizes the between-class scatter and minimizes the within-class scatter.
   Fisher's LDA considers maximizing the following objective:

$$J(W) = \frac{W^T S_B W}{W^T S_W W} \tag{8}$$

   S_B is the "between classes scatter matrix".
   S_W is the "within classes scatter matrix".

$$S_B = \sum_C (\mu_c - \overline{x})(\mu_c - \overline{x}^T) \tag{9}$$

$$S_W = \sum_C \sum_{i \in C} (x_i - \mu_C)(x_i - \mu_C^T) \tag{10}$$

   where $\bar{x}$ is the overall mean of the data cases.
   To classify a face, project it onto the LDA space:

$$\overrightarrow{x}_{LDA} = W^T \overrightarrow{x}, \overrightarrow{y}_{LDA} = W^T \overrightarrow{y}, \overrightarrow{z}_{LDA} = W^T \overrightarrow{z}, \overrightarrow{w}_{LDA} = W^T \overrightarrow{w} \tag{11}$$

   After the above projection of faces onto the LDA space we can run a classification algorithm like Nearest Neighbor.

Randomfaces

Random features can be viewed as a less-structured counterpart to classical face features, such as Eigenfaces or Fisherfaces. Accordingly, we call the linear projection generated by a Gaussian random matrix Randomfaces.
Consider a transform matrix $R \in R^{d \times m}$ whose entries are independently sampled from a zero-mean normal distribution and each row is normalized to unit length. The row vectors of R can be viewed as d random faces in R^m.
Randomfaces is extremely efficient as the transformation R is independent of the training data set. This advantage is helpful in face recognition systems where we do not know the complete dataset before hand as opposed to Eigenfaces and Fisherfaces.

State of the Art

1. Google currently uses "FaceNet: A Unified embedding for Face Recognition and Clustering". [10]
2. Facebook has its own face recognition research project called DeepFace which is very accurate. It uses advanced deep learning neural network. [11]

Competing Methods

1. K Nearest Neighbor Classification

KNN [12] classifier is best suited for classifying persons based on their images due its lesser execution time and better accuracy than other commonly used methods which includes Hidden Markov Model and Kernel Method. Although methods like SVM and Adaboost algorithms are proved to be more accurate than KNN classifier, KNN classifier has a faster execution time and is dominant than SVM.

KNN Algorithm:

1. Each data pixel value in the data set has a class label in the set, Class = { c_1, .., c_n }
2. The data points k closest neighbors (k being the number of neighbors) are then found by analyzing the distance matrix.
3. The k closest data points are then analyzed to determine which class label is the most common among the set.
4. The most common class label is then assigned to the data point being analyzed.

The Euclidean distance metric is often chosen to determine the closeness between the data points in KNN. A distance is assigned between all pixels in a dataset. The Euclidean metric is the function d: Rn X Rn R that assigns to any two vectors in Euclidean n-space X = (x_1, .., x_n) and Y = (y_1, .., y_n) the number,

$$d(x, y) = \sqrt{((x_1 - y1)^2 + ... + (x_n - y_n)^2)} \qquad (12)$$

This gives the standard distance between any two vectors in R_n. From these distances a distance matrix is constructed between all possible pairings of points (x,y).

When noise is present in the locality of the query instance, the noisy instance(s) win the majority vote, resulting in the incorrect class being predicted. A larger k could solve this problem. When the region defining the class, or fragment of the class, is so small that instances belonging to the class that surrounds the fragment win the majority vote. A smaller k could solve this problem. The KNN shows superior performance for smaller values of k compared to larger values of k. The special case where the class is predicted to be the class of the closest training sample (i.e. when k= 1) is called the nearest neighbor(NN) algorithm.

2. Nearest Feature Subspace Classification

Nearest Feature Space (NFS) [8] is an extension of NN because it accommodates a larger capacity of prototype features. The distance between query and feature space is shortened. To cover sufficient facial variations without much overhead, we generalize the geometrical concept from plane to space. Let { Z_{c1},Z_{c2}, .., Z_{cn_c} } denote the independent prototype features associated with class c. The subspace of R_P with span S^P = *sp*(Z_{c1},Z_{c2}, .., Z_{cn_c}) represents the feature space for person c. The NFS is intended to classify the query by finding the nearest feature space among all classes.

$$d(z, S^c) = \min_{1 \le c \le C} d(z, S^c) = \min_{1 \le c \le C} \|z - p^c\| \qquad (13)$$

When NN is applied for one, two and three prototypes per class these classifiers exactly become NFS.

3. Sparse Representation based Classification

Sparse Representation based Classification (SRC) can be regarded as a generalization of Nearest Neighbor (NN) and Nearest Feature Subspace (NFS). [6]

In equation 14, y is the test image, A is the dictionary, where A_i contains training face images of i-th subject.

$$y = Ax \qquad (14)$$

Given y and A, we need to find a sparse solution for x, which will give us the class to which the test sample y belongs. It has been proven that whenever y = Ax for some x with less than m=2 non-zeros, x is the unique sparse solution. [9] The reconstruction is to solve problem P_0 in equation 15.

$$(P_0): \quad \hat{x} = \arg\min_x \|x\|_0 \qquad s.t. \quad y = \Phi x \qquad (15)$$

However, the problem of finding the sparsest solution of an under-determined system of linear equations is NP-hard and difficult even to approximate [7]. Just as [9] indicates that under mild condition, more specific, just as we learned in class: the theory of compressive sensing (CS) reveals that if the solution is sparse enough, the solution of the l_0 problem is equal to the l_1 problem P_1 in equation 16.

$$(P_1): \quad \hat{x} = \arg\min_x \|x\|_1 \qquad s.t. \quad y = \Phi x \qquad (16)$$

SRC can be considered as a generalization of classifiers such as Nearest Neighbor and Nearest Feature Subspace. It is not critical to the choice of an optimal feature transformation [3]. If the solution is sparse enough, with the overwhelming probability, it can be correctly recovered via l_1-minimization from any sufficiently large number of linear measurements. Unlike (P_0), this problem is convex, which means it can actually be recast as a linear programming problem and is solved efficiently [7, 13].

Weighted Sparse Representation based Classification

The SRC performed well comparing to the classic classifiers like NN or NFS and it is robust to occlusion and noise, which boosted the research of sparsity based face recognition. However, there are some drawbacks in SRC. The algorithm assumes that the training images have been carefully controlled and that the number of samples per class is sufficiently large. If the number of linear measurements is not large enough, the discriminative information from the linearity structure of data in lower dimensional feature subspaces is not enough for SRC. It may even perform worse than NN or NFS. NN related algorithms uses locality structure of data, NFS and SRC use the linearity structure of data. But locality of structure is also important. Due to the mechanism of l_1-minimization, sparse coding coefficients may vary a lot even for similar test samples. Given following example: $x_0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, $\phi = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}$.

We wish to recover x_0 from $y = \phi x_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. x_0 should be $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$. But there's another feasible incorrect solution $\begin{bmatrix} 1/3 \\ 0 \\ 1/3 \end{bmatrix}$. The sparse coding may reconstruct a test sample by training images which are far from the test samples. This produces unstable classification results, i.e. SRC tends to lose locality information.

We can improve upon l_1-minimization to find a different (but still convex) alternative to l_0-minimization, but with a lower measurement requirement than l_1-minimization. If we impose a hypothetical weighting matrix W = *diag*($\begin{bmatrix} 3 \\ 1 \\ 3 \end{bmatrix}$) in the previous example, the unstable result problem can be solved.

A weighted Sparse Representation based classification (WSRC) addresses such problem. WSRC utilizes both data locality and linearity. It is regarded as an extension of SRC, but the coding is local. In equation 17, W is a diagonal matrix, which is the locality adapter that penalizes the distance between y and each training data. A larger distance indicates a farther distance between y and x_{ic}.

$$diag(W) = [dist(y, x_1^1), ..., dist(y, x_{n_C}^C)]^T, \text{ where } dist(y, x_1^C) = \left\|y - x_i^C\right\|^S \qquad (17)$$

$$(WP_0): \quad \hat{x} = \arg\min_x \|Wx\|_0 \qquad s.t. \quad y = \Phi x \qquad (18)$$

$$(WP_1): \quad \hat{x} = \arg\min_x \|Wx\|_1 \qquad s.t. \quad y = \Phi x \qquad (19)$$

Whenever the solution to (P_0) is unique, it is also the unique solution to (WP_0) in equation 19 provided that the weights do not vanish. However, the corresponding l_1 relaxations (P_1) and (WP_1) will have different solutions in general. We may think of the weights as free parameters in the convex relaxation, whose values, if set wisely, could improve the signal reconstruction. It can well characterize the similarity between the test sample and training data. WSRC can generate more discriminative sparse codes which can be used to represent the test sample more robustly. From the perspective of linearity, WSRC is a direct extension of SRC. From the perspective of locality, WSRC can be regarded as an extension of kNN, while WSRC is a neighborhood adaptive algorithm. Considering the sparsity, kNN can be named as k sparse neighborhood representation, while SRC is based on sparse linear representation, and WSRC strikes a balance between them.

Experiments

In this section, we investigate the performance of the WSRC method for face representation and recognition.

1. Face Recognition Verification

In our experiments, we used The Extended Yale B database to verify the WSRC algorithm. This database contains 2414 frontal images of 38 subjects under various lighting conditions. For each subject, we randomly select 10 test images, and the use the remainder for training.

The source code for our experiments [17] comes from the authors of [2], and the SPAMS package [16] was used to solve the stable weighted l_1-minimization problem. We used MATLAB to run and generate the results in Figure 3.

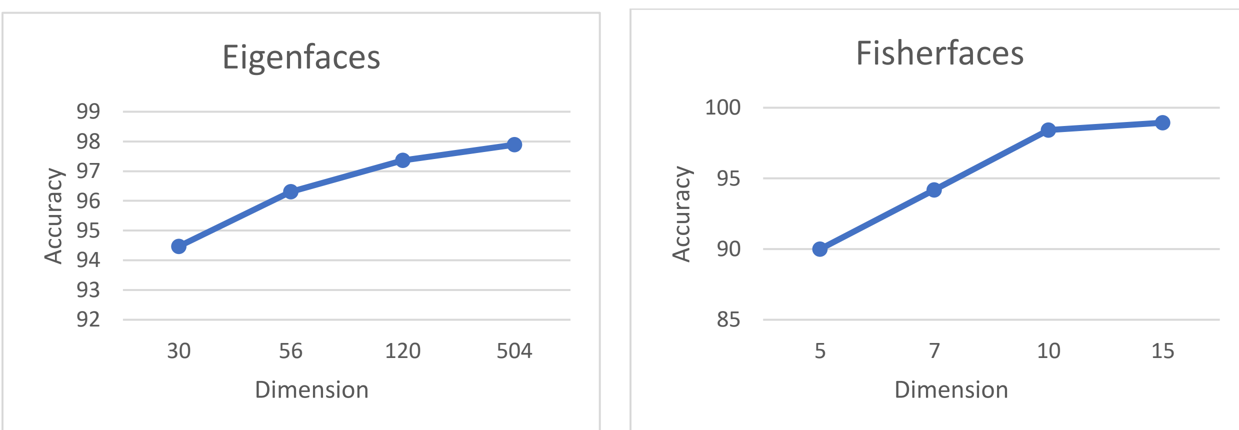
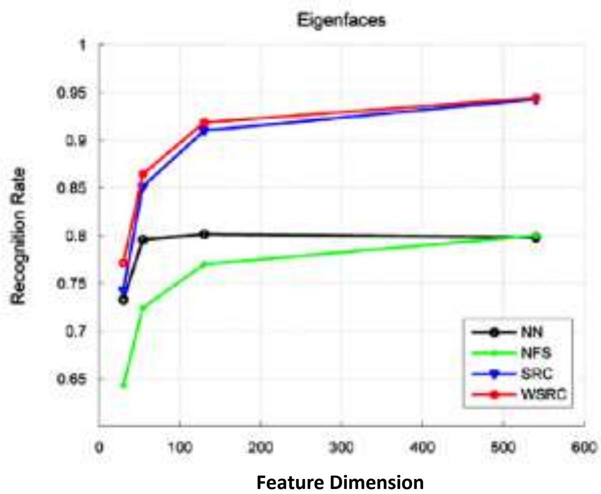


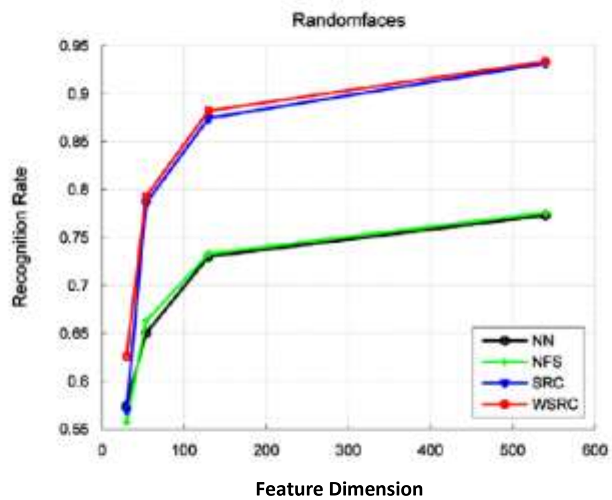
Figure 3: Accuracy versus reduced dimensionality on Extended Yale B database for WSRC: (a) Eigenfaces; (b) Fisherfaces

Our experimental results for WSRC using Eigenfaces matched the recognition rates found in the original research [2]. In the case of Fisherfaces, the accuracies we got were slightly higher than those presented by [2], although the graph showed a similar trend. Our results imply that as dimensionality reduces, the recognition accuracy also drops. However, as we can see in Figure 4, WSRC still performs better than NN, NFS and SRC classification methods with lower dimensional data.

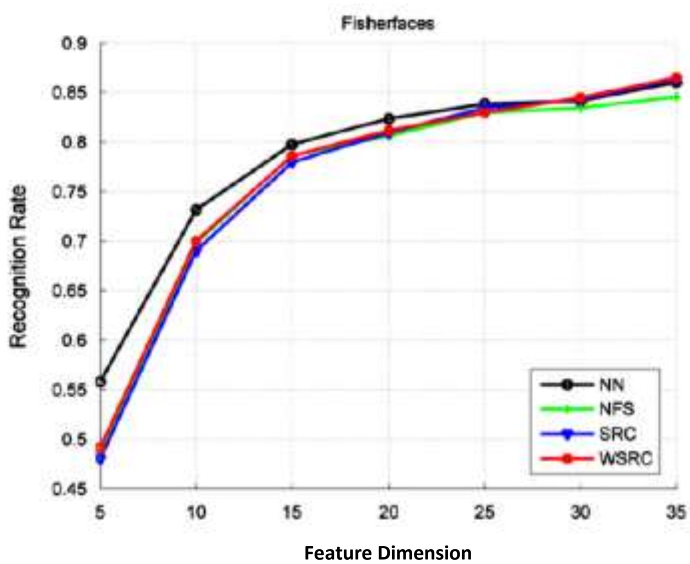
Recognition accuracies of competing classification techniques were recorded in [2] for the same Yale database. From their results (shown in Figure 4), we can see that WSRC outperforms SRC by using Eigenfaces and Fisherfaces. This implies, in lower dimensional subspaces, the data linearity is not enough to separate data from different subspaces, and the imposed data locality is better preserved which help improve recognition performance. However, WSRC is only slightly better than SRC using Fisherfaces. That is because after the projection of LDA, the training data belong to the same subject are very close, the original SRC coding is already local, which is like WSRC. [2] and [3] also compare SRC techniques to tradition face recognition algorithms like NN and NFS. As we can observe from the graphs below, WSRC outperforms all three in most cases.



(a) Eigenfaces



(b) Randomfaces



(c) Fisherfaces

Figure 4: Accuracy versus reduced dimensionality on Extended Yale B database: (a) Eigenfaces; (b) Randomfaces; (c) Fisherfaces

2. Generalization Verification

To test the effectiveness of WSRC on other pattern recognition problems, [2] also experimented WSRC on 15 different data sets. Through the results in Table 1, we can see that WSRC performs better than other classifiers on most of these datasets. The result implies WRSC not only works well for face recognition, but also works well in many other pattern recognition problems. However, due to the complexity and work of processing different data sets, and considering the limited time for the semester, we did not have a chance to verify the generalization result by ourselves.

Table 1: Recognition on 15 data sets

	NN	NFS	SRC	WSRC
Air	94.29 ± 4.84	87.14 ± 2.47	95.14 ± 0.78	95.14 ± 2.78
Austra	78.82 ± 4.05	76.76 ± 4.54	78.38 ± 6.20	76.03 ± 5.80
Breast	72.59 ± 11.48	70.37 ± 0.00	71.48 ± 5.25	74.44 ± 6.86
Breast_gy	70.37 ± 7.41	70.74 ± 3.24	68.89 ± 7.45	70.00 ± 7.70
German	67.20 ± 6.20	70.00 ± 0.00	71.90 ± 4.56	73.10 ± 5.82
Glass	31.50 ± 6.02	26.50 ± 5.76	33.50 ± 6.75	39.00 ± 9.12
Heart	76.30 ± 6.34	76.67 ± 4.64	71.85 ± 5.84	74.44 ± 5.91
Ionosphere	64.71 ± 0.00	40.29 ± 3.41	89.41 ± 5.75	91.18 ± 5.00
Iris	96.00 ± 4.66	94.00 ± 6.63	95.33 ± 4.50	97.33 ± 3.44
Sonar	84.00 ± 8.76	79.50 ± 11.89	84.50 ± 6.43	85.50 ± 5.99
Vote	91.19 ± 2.76	76.43 ± 5.55	93.57 ± 4.05	93.81 ± 4.52
Vowel	98.18 ± 1.44	63.18 ± 7.48	97.27 ± 1.44	98.41 ± 1.53
WBC	94.93 ± 2.46	76.42 ± 4.38	91.94 ± 5.27	92.39 ± 4.31
Wine	95.63 ± 6.62	83.13 ± 9.34	95.63 ± 5.15	96.88 ± 4.42
X8D5K	100.00 ± 0.00	92.20 ± 3.46	100.00 ± 0.00	100.00 ± 0.00
Vote	91.19 ± 2.76	76.43 ± 5.55	93.57 ± 4.05	93.81 ± 4.52
Vowel	98.18 ± 1.44	63.18 ± 7.48	97.27 ± 1.44	98.41 ± 1.53
WBC	94.93 ± 2.46	76.42 ± 4.38	91.94 ± 5.27	92.39 ± 4.31
Wine	95.63 ± 6.62	83.13 ± 9.34	95.63 ± 5.15	96.88 ± 4.42
X8D5K	100.00 ± 0.00	92.20 ± 3.46	100.00 ± 0.00	100.00 ± 0.00

Future Work

We can further enhance the WSRC by using the discriminative nature of human face [14].

Discriminative Sparse Representation

SRC emphasizes more on sparsity and overlooks spatial information. Weighted SRC consider the spatial information but overlooks the different discriminative abilities in different face regions. DSR assign the weights at face locations according to the importance of face region in classification. DSR does this assignment of weights using the information Entropy property.

Information Entropy: entropy is measurement of the uncertainty associated with a random variable. We consider pixel intensity value as random variable for calculating information entropy of Shannons entropy formula of face region i,

$$H_i = -\sum_{k=1}^{n} P(v_k) log(v_k) \tag{20}$$

where p(v) is the probability of the pixel v with intensity value k.

Spatial Weighting strategy: Our task is to assign higher weights to regions rich in texture. To achieve that, we first divide the face images into a few coarse rectangular regions and then the pixels in each region are regrouped into different sets by information entropy measure. these entropy values are passed on to transform function to calculate weights.

$$N(H_i) = \frac{H_i - H_{min}}{H_{max} - H_{min}}(new_{max} - new_{min}) + new_{min} \tag{21}$$

The brightness of the regions in the image indicates their importance. the brightest regions are in the area of two eyes, nose and mouth. The darkest regions (corresponding to low weights) are around the cheek and forehead. This indicates that the learned weights can discover the more discriminative face regions to some extent and well capture the spatial information.

By imposing the weighted spatial information into the sparse coding scheme, our method can be formulated as follows:

$$\min \|W(y - D\alpha)\|_2^2 + \sum_{j=1}^{C} \|\alpha_j\|_2 \tag{22}$$

In the above equation we added an $l_{2,1}$-norm (group Lasso) constraint upon the formulation, which enforces the sparsity at the group level. We find the sparse solution for α using Least Angle Regression Selection (LARS) algorithm.

Learning Outcomes

Through this project, we gained a deeper understanding of the following topics:
1. Existing literature on face recognition techniques and our reference articles [2] and [3].
2. Implementing classification of a given face database using sparse representations.
3. Different feature extraction methods like Eigenfaces and Fisherfaces.
4. Constructing a Dictionary from extracted features for sparse coding.
5. Limitations of using only SRC in real-world face recognition applications.

In the experiments we ran, we found that the construction of dictionary is a crucial step in deciding the performance and accuracy of the algorithm. As we use more intelligent ways to generate the dictionary, the performance gets better and we get the following conclusion:

$$SRC < WSRC < DSRC \tag{23}$$

Further in our study on the future work for our topic, we also learned about information entropy and its role in face recognition to extract discriminative features of the face.

Individual Contribution

Shikha Mehta: Worked on extracting features from the input database using Eigenfaces and Fisherfaces, and building a Dictionary matrix from it. Also developed the weight matrix for the WSRC algorithm and consolidated the comparative results achieved by authors in [2] and [3] w.r.t NN, NFS and SRC techniques.

Raheen Mazgaonkar: Configured the toolbox for solving the l_1-minimization problem and worked on the end-to-end algorithm for WSRC. Ran experiments using only face inputs (high dimensional data) and with feature matrices (low dimensional data) to evaluate the performance of WSRC w.r.t. to each.

Both team members had equal contribution towards drafting this project report.

Acknowledgments

We would like to thank Professor Dr. Alireza Entezari and Teaching Assistant for helpful and informative discussions on sparse coding and face recognition during the semester. We would also like to acknowledge use of following code repositories and datasets in our experiments:
1. http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html
2. http://spams-devel.gforge.inria.fr/downloads.html
3. http://www.openpr.org.cn/index.php/108-Face-recognition-via-Weighted-Sparse-Representation/View-details.html
4. https://github.com/dhingratul/Face-Recognition-Algorithms

References

[1] https://findbiometrics.com/solutions/facial-recognition

[2] Can-Yi Lu, Hai Min, Jie Gui, Lin Zhu, and Ying-Ke Lei. Face recognition via weighted sparse representation. *Journal of Visual Communication and Image Representation*, 24(2):111-116, 2013.

[3] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma, Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210-227, 2009.

[4] R. Duda, P. Hart, D. Stork, Pattern Classification, 2001.

[5] D. Donoho, "For Most Large Underdetermined Systems of Linear Equations the Minimal Solution Is Also the Sparsest Solution", *Comm. Pure and Applied Math.*, vol. 59, no. 6, pp. 797-829, 2006.

[6] Can-Yi Lu, Hai Min, Jie Gui, Lin Zhu, Ying-Ke Lei, Face recognition via Weighted Sparse Representation. *Journal of Visual Communication and Image Representation*, Volume 24, Issue 2, Pages 111-116, 2013. ISSN 1047-3203, https://doi.org/10.1016/j.jvcir.2012.05.003.

[7] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877-905, 2008.

[8] Jen-Tzung Chien and Chia-Chen Wu. Discriminant wavelet faces and nearest feature classifiers for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1644-1649, 2002.

[9] D Donoho and Y Tsaig. Fast solution of l1-norm minimization problems when the solution may be spars. *Preprint*, 1(2), 2006.

[10] Florian Schro, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. pages 815-823, 2015.

[11] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face veri_cation. pages 1701-1708, 2014.

[12] V Vaidehi, S Vasuhi, R Kayalvizhi, et al. Person authentication using face recognition. *Proc. World Congress on Engineering and Computer Science*, 2008.

[13] Allen Y Yang, S Shankar Sastry, Arvind Ganesh, and Yi Ma. Fast l1-minimization algorithms and an application in robust face recognition: A review. *IEEE International Conference on Image Processing*, pages 1849-1852. IEEE, 2010.

[14] Zhihong Zhang, Yuanheng Liang, Lu Bai, and Edwin R Hancock. Discriminative sparse representation for face recognition. *Multimedia Tools and Applications*, 75(7):3973-3992,2016.

[15] http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html

[16] http://spams-devel.gforge.inria.fr/downloads.html

[17] http://www.openpr.org.cn/index.php/108-Face-recognition-via-Weighted-Sparse-Representation/View-details.html

[18] https://github.com/dhingratul/Face-Recognition-Algorithms