Dimensionality Reduction on Handwritten Character Dataset

Akash Kumar Kondaparthi
Dept. of Electrical and Computer Engineering
University of Florida
Gainesville, FL, USA
akash.kondaparth@ufl.edu

***Abstract*—This paper presents a comprehensive study of dimensionality reduction techniques applied to a custom handwritten character dataset to reduce the dimensionality of the data, enhancing data visualization, and improving classification performance. Given the high dimensionality of the dataset, traditional classification approaches face computational and performance challenges. To address these, manifold learning algorithms, namely t-SNE, MDS, Isomap, and Locally Linear Embedding (LLE), along with Principal Component Analysis (PCA), Fisher's Linear Discriminant Analysis (LDA), and Recursive Feature Elimination (RFE) were employed to reduce the dimensionality by which classification becomes more tractable, and the data becomes easier to visualize. Methodologies involved rigorous data preprocessing, application of feature selection, feature extraction, and manifold learning algorithms. Classifiers were trained on both the original and reduced feature spaces to compare performances. The results demonstrate that dimensionality reduction not only accelerates the training process but also enhances classifier accuracy.**

I. Introduction

In the realm of machine learning and pattern recognition, the classification of high-dimensional data presents a significant challenge. Particularly in the case of image recognition, where each pixel serves as a distinct feature, the curse of dimensionality can severely hamper the ability of classifiers to efficiently and effectively classify between different classes. This paper addresses the problem of classifying handwritten characters by using dimensionality reduction techniques.

The dataset used for this project comprises 6720 grayscale images (in the training set, and 2880 images in the test set) of ten unique characters, each image being a 300x300 pixel representation, thereby resulting in a feature space of 90,000 dimensions. To tackle the computational complexity and enhance classifiers' performance, dimensionality reduction techniques, such as Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), Fisher's Linear Discriminant Analysis (LDA), and manifold learning methods including MDS, t-SNE, Isomap, and Locally Linear Embedding (LLE) were used. These methods were also used to reduce the dimensionality so that the datasets are easier to visualize in a 2-dimensional space. The reduced feature spaces were ensured to establish a balance between the computational efficiency and the practicality of the classification task.

II. Methodology

A. Data Preprocessing

The initial step involved correcting mislabeled instances within the handwritten character dataset. This task was accomplished manually, ensuring that each image was correctly assigned its intended character class label. Next, images were resized from their original dimensions of 300×300 pixels to 50×50 pixels. This reduction, selected after trying out different possibilities as in Figure 1, substantially decreased the feature space from 90,000 to 2,500 dimensions without a significant loss in image quality, thereby accelerating computation and optimizing resource usage.

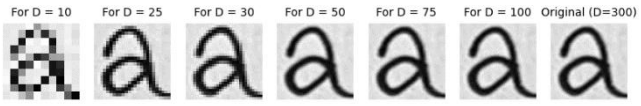

Figure 1. Choosing a pixel size to reduce from 300 x 300

B. Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) was applied with two estimators. Logistic Regression and Random Forest Classifier were evaluated in combination with RFE. Various step sizes and numbers of features were tested, to achieve the best performance using the classifiers. RFE was used with Logistic Regression and Random Forest Classifier to choose and retain the best 900 features.

C. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was then employed to reduce dimensionality to components accounting for 90% of the explained variance. This step aimed to reduce the feature space while maintaining the data's core characteristics necessary for accurate classification.

The training was conducted on both the original and reduced datasets using a Random Forest Classifier. The obtained eigenvectors, mainly the top 10, were visualized. The images from the dataset were later reconstructed using only these eigenvectors.

D. Fisher's LDA and t-SNE

Fisher's Linear Discriminant Analysis (LDA) and t-distributed Stochastic Neighbor Embedding (t-SNE) were used to create two-dimensional embeddings of the dataset to facilitate data visualization.

E. Manifold Learning for Visualization and Classification

Three manifold learning algorithms—Multidimensional Scaling (MDS), Isometric feature Mapping (Isomap), and Locally Linear Embedding (LLE)—were utilized to project the dataset into a two-dimensional space. These projections were visualized, with points color-coded by their target label, to illustrate the underlying data structure.

Upon transforming the data into optimal low-dimensional representations, a Random Forest Classifier was trained within each manifold-reduced feature space. The data was reduced to 100 components for this task using every Manifold Learning algorithm listed before. Metrics such as accuracy, precision, recall, and the F1 score were used to evaluate the performance of manifold learning techniques in enhancing classifier performance.

III. RESULTS

A. *Recursive Feature Elimination (RFE)*

RFE with Logistic Regression: After grid-searching different combinations of components and step sizes, the best algorithm turned out to select 900 features with a step size of 0.05 for best performance. The features selected by RFE with Logistic Regression, when mapped as a mask image as shown in Figure 2, show that mostly the center pixels are selected.

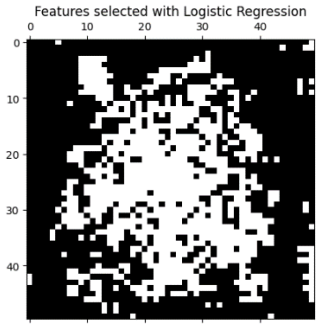

Figure 2. Feature Mask with Logistic Regression

In the feature mask, white pixels represent the selected features, and the black pixels represent the discarded ones. Using the obtained mask, a few masked images are shown in Figure 3.

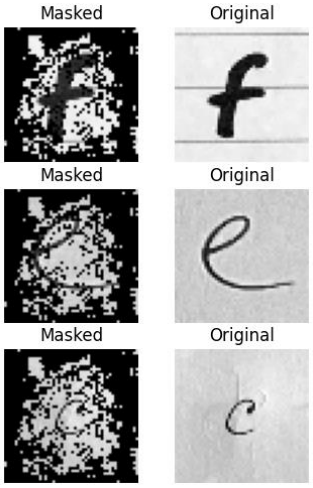

Figure 3. Masked examples using Logistic Regression

RFE with Random Forest: After grid searching through different components and step sizes for best performance, the best model was configured to select 900 features with a step size of 0.1.

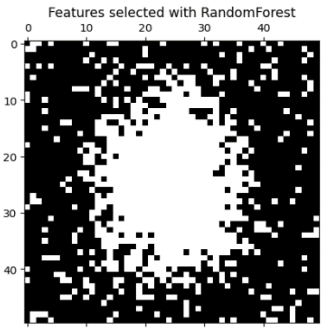

Figure 4. Feature Mask with Random Forest

The plotted mask of these selected features as an image is shown in Figure 4. The masked examples from the training set using this obtained feature mask are shown in Figure 5.

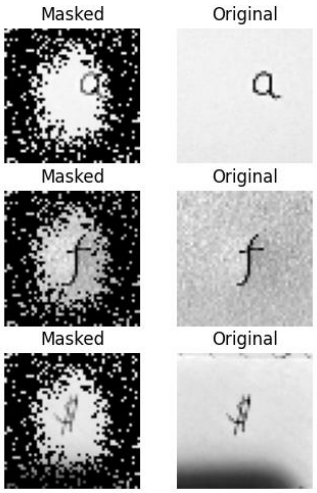

Figure 5. Masked examples using Random Forest

B. *Principal Component Analysis (PCA)*

PCA was used to reduce the dimensionality of the dataset to explain 90% of the explained variance. This number turned out to be 165 components, which means that the first 165 components (eigenvectors) were enough to preserve 90% of the explained variance. The top 10 eigenvectors (PCA components) were visualized as shown in Figure 6.

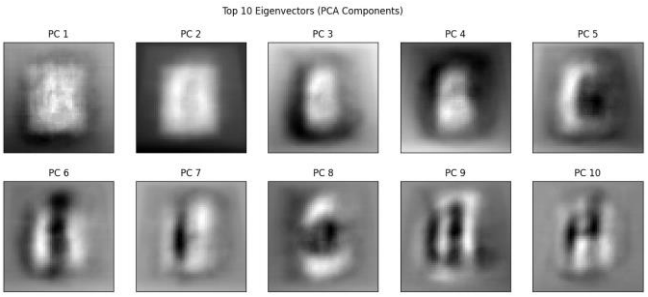

Figure 6. Visualization of top 10 Eigenvectors

Training the Random Forest Classifier on the original dataset took ~79.28 seconds compared to the reduced dataset which took ~24.98 seconds. The classification accuracy on the test set improved when using the reduced dataset (56.03%) compared to the original dataset (46.06%). The performance metrics for both datasets are detailed in Table 1.

Performance	Precision	Recall	F1-score	Accuracy
Original Dataset	0.48	0.46	0.46	46.06%
Reduced Dataset	0.58	0.56	0.56	56.03%

Table 1. Performance comparison with and without PCA

The images from the training set were reconstructed using just the eigenvectors required to explain 90% of the explained variance. These reconstructions are shown in Figure 7.

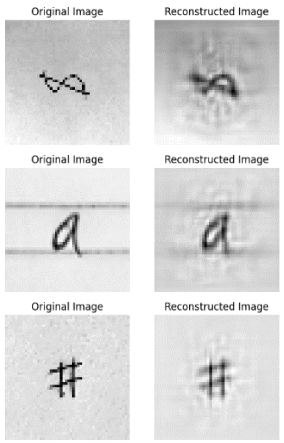

Figure 7. Image reconstructions from PCA

C. *Fisher's LDA and t-SNE*

Linear Discriminant Analysis (LDA): Using LDA, the dataset was reduced to 2 dimensions. The 2-dimensional projection using LDA is depicted in Figure 8.

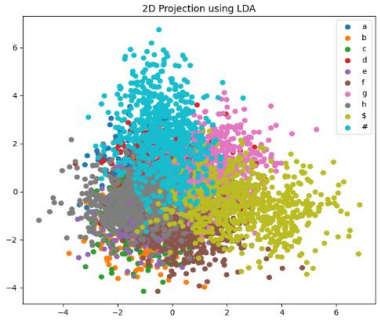

Figure 8. 2D Projection using LDA

t-Distributed Stochastic Neighbor Embedding (t-SNE): Using t-SNE, the dataset was also reduced to 2 dimensions to visualize it. Figure 9 illustrates the t-SNE projection.

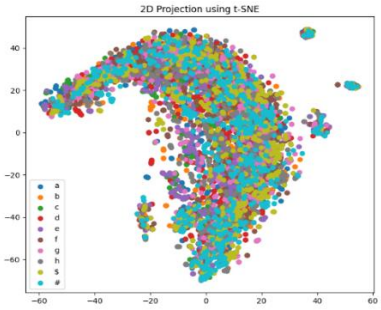

Figure 9. 2D projection using t-SNE

Principal Component Analysis (PCA): For comparison, PCA was also used to reduce the dataset to 2-dimensional space for visualization, which is shown in Figure 10.

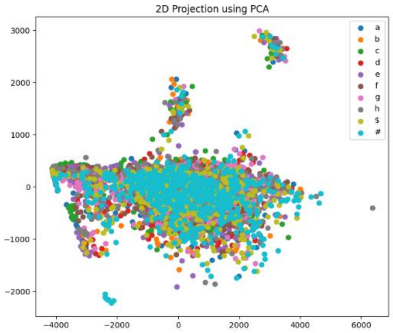

Figure 10. 2D projection with PCA

D. *Manifold Learning for Visualization and Classification*

Multidimensional Scaling (MDS): MDS was used to reduce the dataset to 100 features and a Random Forest was used to train on the transformed dataset. MDS was also used to reduce it to 2D to visualize it as shown in Figure 11.

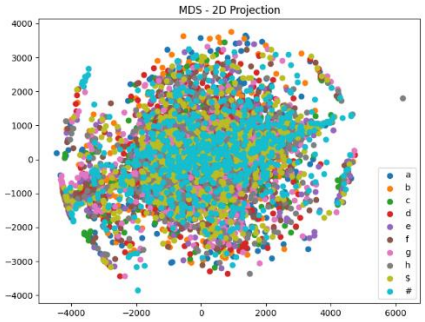

Figure 11. 2D projection using MDS

Isomap: ISOMAP was also used to reduce the dimensions to 100 features to train a Random Forest Classifier on it. The 2D Projection using ISOMAP is displayed in Figure 12.

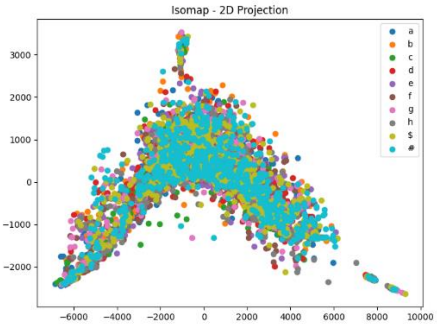

Figure 12. 2D projection using ISOMAP

LLE: LLE was also used to reduce the dataset to 100 features. A Random Forest Classifier was then trained on the LLE-transformed data. LLE was also used to reduce them to 2D and visualize it as shown in Figure 13.

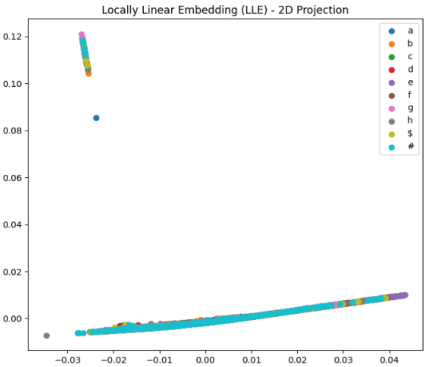

Figure 13. 2D projection using LLE

Accuracy	MDS	ISOMAP	LLE
Random Forest	14%	31%	37%

Table 2. Performances using Manifold Learnings and Random Forest

The accuracies obtained by the Random Forest Classifiers on these datasets are summarized in Table 2.

IV. Discussion

A. *Recursive Feature Elimination (RFE)*

Comparing both RFE approaches, it is evident that different classifiers focus on varying aspects of the images

when determining feature importance. But in this case between Logistic Regression and Random Forest, they both select features in the center of the image. This makes sense because most of the images contain the handwritten character positioned in the center of the image. It also looks like the center of the image is better captured by Random Forest than Logistic Regression.

B. Principal Component Analysis (PCA)

From the obtained training times of a Random Forest Classifier with and without PCA, we can see that the time taken for training on the PCA-reduced Dataset is much shorter than the time taken for training on the Original Dataset. This is because since we are going down from 2500 features to 165, the computations required are reduced drastically, reducing the training times as well.

From the performance results, we can say that the performance improved from the original dataset to the reduced dataset at a lower computational cost. This might be because PCA extracts more useful features and hence the performance will be better and faster using a smaller number of but more important features.

From Figure 6, the top 10 eigenvectors can represent the following:

- PC1, PC2 - These eigenvectors have higher values in the center in the shape of a rectangle. This might mean that the first eigenvectors capture the data in the center of the image, which makes sense because the characters in the images are mostly in the center of the image.
- PC3, PC4 - These eigenvectors also capture the center data but this center area, here, is smaller than the first two eigenvectors. This eigenvector might represent the smaller-sized written symbols.
- PC5 - This eigenvector captures the data around the periphery of the center area of the image.
- PC6, PC7 - These two eigenvectors capture and represent the data on the left and right sides of the center area of the image.
- PC8 - This eigenvector captures and represents the data in the top and bottom portions of the center of the image.
- PC9, PC10 - These eigenvectors form a complex shape but look for the data mostly in the center of the image.

From Figure 7, we can notice that even though the reconstructions are not as good as the original image, they retain the important information to classify the images.

C. Fisher's LDA and t-SNE

From the visualizations from Figures 8, and 9, it is apparent that LDA has a better reduction to 2 dimensions than t-SNE in this case. In the LDA projection, we can see the groups of classes are well separated. Whereas, in the visualization by t-SNE, all the classes seem to take the same shape/distribution making it harder to differentiate between classes. This is because LDA is a supervised algorithm and has the advantage of knowing the class labels to make the class separation better.

Using LDA, it might be feasible to select just 2 features, but the performance is going to be much worse since there are classes that still overlap. But t-SNE performs much worse at the same 2 dimensions. So, for classification applications, reducing 300 × 300 dimensions to 2 dimensions is not a good idea. The exact number of dimensions to select can be found out from further experimentation using a classifier. We cannot use visualization to select the number of features because we cannot visualize more than 3 dimensions. In LDA, the number of discriminant axes is at most c - 1, where c is the number of classes, which might be the best option to select. In t-SNE, opting for a much higher dimensional space is preferred because as seen from the visualizations, the class separation is not good enough to achieve reasonable performance.

From Figure 10, the 2D visualization using PCA, the obtained plot is like that of t-SNE. Here, the class separation is not as good as LDA (again, this is because PCA is also unsupervised, but LDA is not) but the instances of all classes seem to follow the same distribution.

One major observation that is apparent from the visualizations of t-SNE and PCA is that these visualizations aid very well in identifying the outliers in each class. There are a few (mainly two) blobs of instances located off the main distribution, which could indicate outliers in the dataset.

D. Manifold Learning for Visualization and Classification

Since the accuracy obtained from reducing the features to 100 using MDS is not impressive, MDS might not be a good choice for this classification. Furthermore, 2D visualization using MDS gives us a single circular distribution which tells us that the performance is not as good. The training with MDS also takes significantly and unreasonably longer than the other Manifold Learning Algorithms. Hence, MDS is not a good choice. And since, for the same number of features, LLE has a relatively higher accuracy than ISOMAP in the test case, LLE is preferable over ISOMAP.

V. Conclusions

This project on dimensionality reduction has demonstrated the advantages of extracting meaningful patterns from complex, high-dimensional data. The application of these methods, particularly RFE, Fisher's LDA, PCA, and LLE, has provided ways for more efficient data processing and improved classification accuracy. By reducing computational overhead and highlighting critical features within the data, these techniques offer substantial benefits for boosting performance, reducing costs, and improving the interpretability of models.

Manifold learning techniques are also invaluable tools for dimensionality reduction, especially when dealing with complex datasets where linear projections like PCA fall short. The results obtained in this project demonstrate that methods like LLE can provide a significant boost in performance for classification tasks, affirming the importance of choosing the right dimensionality reduction technique that aligns with the underlying data structure.

References

[1] https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

[2] https://github.com/UF-AppliedMLSystems-Fall23/Lectures/blob/main/Lecture%2019/Lecture%2019-in-class%20edits.ipynb

[3] scikit-learn.org/stable/modules/generated/sklearn.manifold.LLE.html