Documentation - exercise 4 - Extra
Dataset1 : Credit Card Transactions Fraud Detection Dataset
Dataset2 : Handwriting A-Z alphabet

Professor : Dr. Kheradpisheh
Teacher Assistant : MohammadReza Khanmohammadi

By: Katayoun Kobraei

1. Introduction to Dataset

The dataset contains 26 folders (A-Z) containing handwritten images in size 28*28 pixels, each alphabet in the image is centre fitted to 20*0 pixel box.
Each image is stored as Gray-level. Kernel **CSV_To_Images** contains script to convert .CSV file to actual images in .png format in structured folder. The dataset might contain some noisy image as well.

The images are taken from NIST(https://www.nist.gov/srd/nist-special-database-19) and NMIST large dataset and few other sources which were then formatted as mentioned above.

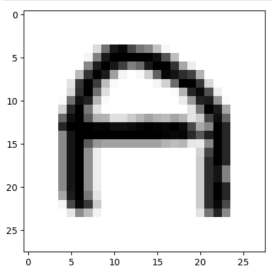
2. Abstract

In this assignment we focus on predicting handwritten alphabets using machine learning methods like KNN. The challenge lies in the high dimensionality of image data and the potential "curse of dimensionality." To overcome this challenge, feature reduction methods such as Principal Component Analysis (PCA) and LDA are explored. These techniques efficiently capture the essential information from the handwritten alphabets while reducing the overall dimensionality of the data. By emphasizing dimensionality reduction, successful alphabet prediction can be achieved using non-deep learning models, while also gaining valuable insights into the inherent structure and characteristics of handwritten data.

3. Overview of the dataset

Train data's shape:
The training dataset contains a total of 372450 entries and 28*28 picture's columns and an extra column for the alphabet label.

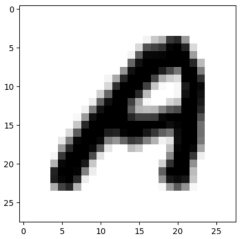
We can see some examples of the dataset now.



And if we check the label it will be:

'A'

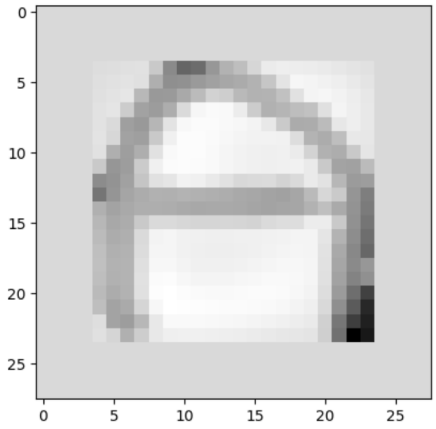
And another example:



Again if we check the label it will be:

'A'

Now before any training we need to preprocess our input data. If we plot normalized input image we see the previous ‘A’ will looks like:



4. Curse of dimensionality

As we now we face high dimensions because images in our data and this lead to the curse of dimensionality issue. The curse of dimensionality arises when dealing with data that has a high number of features or dimensions. It comes with several challenges that can affect our task. When the number of dimensions increases, the data becomes more spread out and sparse. This sparsity makes it harder to accurately estimate statistical quantities and can lead to unreliable models.

Increased computational complexity: High-dimensional data requires more computational resources and time to process and analyze. Algorithms that work efficiently with low-dimensional data become impractical in high-dimensional scenarios, affecting the scalability and efficiency of data analysis. Also with many dimensions, there is a greater risk of overfitting the model to the training data. Overfitting happens when the model becomes overly complex and captures noise or specificities in the training data, resulting in poor generalization to new data. High-dimensional spaces provide more opportunities for models to fit noise, making it harder to identify meaningful patterns. As the number of dimensions increases, the data points become thinly spread across the high-dimensional space. This reduced sample density can compromise the accuracy and reliability of statistical estimates and machine learning models, as there may not be enough data points to adequately represent the underlying distribution or capture the true relationships between variables. In high-dimensional spaces, models tend to become more complex to accurately capture intricate relationships between features. This complexity can make it difficult to interpret and understand the models, making it challenging to extract meaningful insights from the data.

To address the curse of dimensionality issue in the assignment of predicting handwritten alphabets using non-deep learning models, we can apply various feature reduction methods. Here are a few techniques we can consider:

Principal Component Analysis (PCA): we used PCA in our preprocessing. We used 40 principal component as features instead of 728 features. Actually PCA is a widely used technique for dimensionality reduction. It identifies the principal components that capture the most significant variations in the data and projects the data onto a lower-dimensional subspace. By selecting a subset of the principal components, we can effectively reduce the dimensionality of the handwritten alphabet data while preserving the essential information.

Linear Discriminant Analysis (LDA): LDA is other technique that we used before training our model. LDA is also a dimensionality reduction technique that aims to find a linear combination of features that maximizes the separation between different classes. It is particularly useful when the goal is not only to reduce dimensionality but also to enhance the discriminative power of the features for classification tasks. LDA can help to extract informative features that are relevant for distinguishing different alphabet classes.

Beside all these we could use other techniques like t-SNE an Autoencoders neural network. Autoencoders are used for unsupervised dimensionality reduction too. By training an autoencoder on the handwritten alphabet data, we can learn a compressed representation that captures the essential information while reducing dimensionality.

We decided to use KNN model as a classifier because it works better on this data.
We can see after all these we can reach an accuracy of 98.65 with PCA technique and 82.53 with LDA technique.