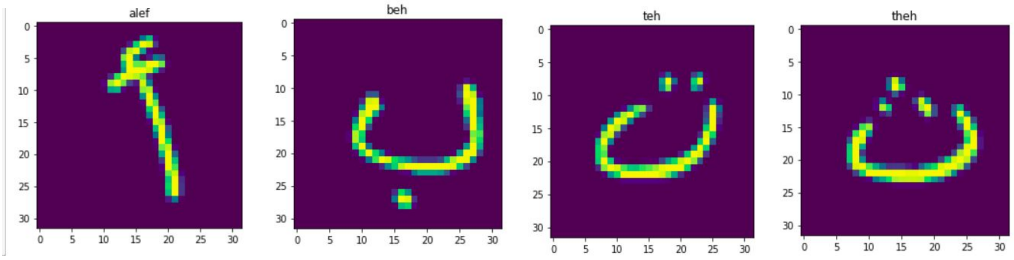
SVD Analysis of Arabic letters dataset
And Handwritten Digits dataset
By applying the nearest neighbor algorithm on both

My project on letters and digit recognition is a classic problem and has many solutions.I used a technique from linear algebra called singular value decomposition (SVD), which is a method for factoring a matrix.It has many different applications in addition to pattern recognition. In my case, I am using it to reduce the dimensionality of the data to make our letter/digit recognition program run in a reasonable amount of time.

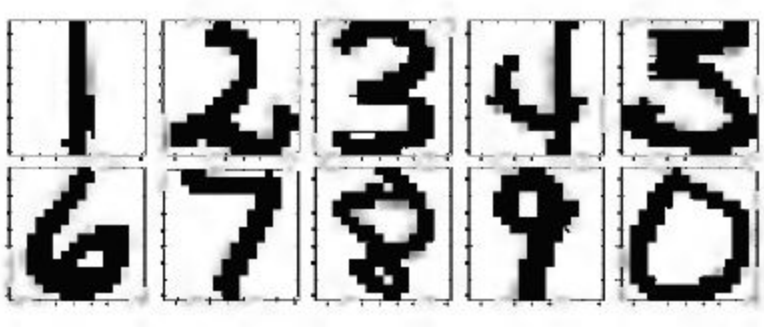
Description of Data

First, Arabic letters handwritten dataset I take it form :
https://www.kaggle.com/mloey1/ahcd1/data
Some examples :



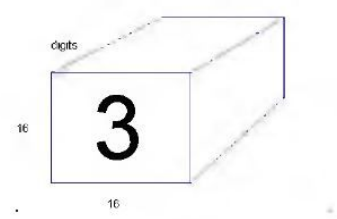
Each one is (32,32) pixel
There are 13440 images for training data and 3360 for testing purpose.

Second ,handwritten digits comes from the U.S. Postal Service database I take it from :
http://users.mai.liu.se/larel04/matrix-methods/computer-assignments/character-recogn.html



Each one is (16,16) pixel
There are 1707 images for training data and 2007 for testing purpose.

to prepare the data I use an array of objects where each object is many matrices of a letter/digit images, for example, the object that holds number 3 “131 copies of 3 ” matrices is something like this :



The method:

The method that I am using is to solve the SVD of the object of matrices for example 3 and save it in a list .

```
def compute_SVD(M): # here compute SVD one time and save results in SVD_List
    U0 , S ,VT = np.linalg.svd(M)
    SVD = [U0 , S ,VT]
    return SVD
```

I uses numpy to perform the SVD, creating three matrices. I am interested in the first matrix, U0. U stands for unitary,

Then when I need to compute the distance between a new image and the matrices I use this code:

```
# find the distance between the image and the matrix of the letter
def find_the_distance(SVD,k,var,trainx ):
    U0 , S ,VT = SVD
    smallU0 = U0[:,0:k]
    d =trainx[var]
    I = np.eye(1024)
    v =np.matmul ((I - np.matmul (smallU0 , smallU0.T)) , d)
    distance = np.linalg.norm(v)
    return distance
```

Where k is number of k and var is index to choses for trainx. After I find all the distances between the new image and all the matrices Then find the nearest neighbor (the shortest distance) and predict the image class.

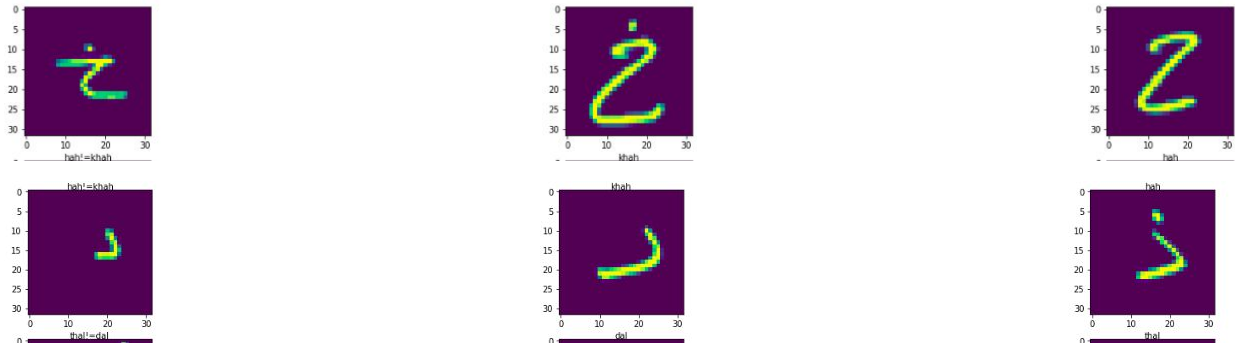
Experiments :

For the Arabic letter dataset After I search for the best k that will be good for the training data, I found it k =19 give me a good result. however, when I try this on the test data give me 52% which can tell there is some recognition on the letters.

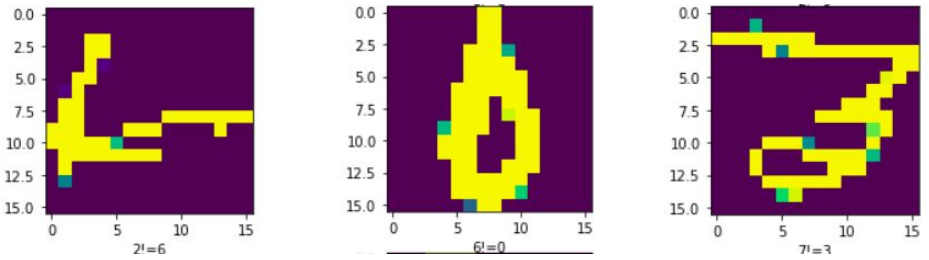
For the Digits dataset, I After I search for the best k that will be good for the training data, I found k=7 give a good result but after applying in on the test set I found it k = 10 give 92% result that I can't see it in the training data

Comparing between the results:

I think since the Arabic letters very much look like each other. My method is not a good method for this data set. While it works well with the digit data set. there are some mistakes but they are understandable. Let show some of the miss predicted images on both data set.



The left image test image. The middle image is a good example of what should be predicted. The right one is a good example of what the model think it is.



Predicted is 2 real is 6 Predicted is 6 real is 0 Predicted is 7 real is 3

Conclusion

This method works good of digit data set but it is not that good for the Arabic letter data set. which we should but that in our consideration when we chose an algorithm for machine learning.

Sources

1. Elden, Lars. *Numerical Linear Algebra and Applications in Data Mining,* Preliminary Version, 2005. (unpublished.) The published version is now available on Amazon under a slightly different title: *Matrix Methods in Data Mining and Pattern Recognition (Fundamentals of Algorithms),* Society for Industrial and Applied Mathematics (April 9, 2007). ISBN 978-0898716269.
2. Wikipedia contributors. Singular value decomposition. Wikipedia, The Free Encyclopedia. December 10, 2018, 21:36 UTC. Available at: https://en.wikipedia.org/w/index.php?title=Singular_value_decomposition&oldid=873057667. Accessed December 13, 2018.