Submitted to the Annals of Applied Statistics

APPLY GEOMETRIC DATA ANALYSIS TO THE MNIST DATA

BY LEON YUAN[1,a]

[1]*Department of Statistics and Data Science, Southern Methodist University,* [a]*yuanli@smu.edu*

This report applies the Geometric Data Analysis method to the MNIST dataset. The main goal is to explore how the shape analysis with SRVF works in a real application. The benchmark dataset MNIST is selected to evaluate the performance. The Geodesic distance, Karcher mean, and a basic classification of images were used in this paper. The final accuracy was 74% for 10 classes, and Karcher mean can be a good representative description for simple MNIST images.

1. Introduction. Geometric data analysis is a big part of Statistical Shape Analysis. GDA focuses on the overall shape and underlying structure. Unlike traditional data analysis, where the data is a collection of single units, the data in GDA is the whole manifold shape where observations lie. This makes GDA a valuable and powerful tool in image analysis and object analysis. For example, the study of the shape change of a plant over time and across species, the development of the brain tumor boundary, fiber tracts in MRI, hurricane trajectory, protein structure, and facial surface recognition. To apply any GDA tools, we first need to represent curves/shapes/surfaces in a mathematical representation, including: coordinate function (β_x, β_y), angel function $e^{i\theta(s)}$, curvature function $\kappa(s) = \dot{\theta}(s)$.

To compare the different curves/shapes, the alignment of curves/shapes is important to remove the translation, rotation, and scale, so the differences across curves/shapes are meaningful variations instead of arbitrary transformations. The registration across curves usually does this alignment by representing the curves into the Square-Root Velocity Function (SRVF)

$$q(t) \equiv \frac{\dot{\beta}(t)}{\sqrt{|\dot{\beta}(t)|}} = \sqrt{\phi(t)}\theta(t)$$

where $\beta(t)$ is the mathematical form we mentioned before, $\phi(t)$ is the speed function and $\theta(t)$ is the directional function. The elastic Riemannian metric is defined in this SRVF space.

Finally, the Geodesic distance is defined on the curve/shape manifold to measure the shortest path across curves on the surface instead of the traditional straight line connecting two points in Euclidean space. The

*Statistical Shape Analysis, Stat 6383

Keywords and phrases: tranformers, sentiment analysis, Topological Data Analysis, Mapper Algorithm, word emebddings.

optimal Geodesic distance is defined as follows:

$$\alpha(\tau) = \frac{1}{\sin(\vartheta)} \left[\sin(\vartheta(1-\tau))q_1 + \sin(\tau\vartheta)q_2^*\right]$$

where $\cos(\vartheta) = \cos^{-1}\left(\langle q_1, q_2^* \rangle\right)$ and $q^* = O^*\left(q_2, \gamma^*\right)$. To find the optimal rotation O^* and γ^* between two curves, fix one curve q_1, and solve the following optimization problem:

$$(O^*, \gamma^*) = \underset{O \in SO(2), \gamma \in \Gamma}{\operatorname{argmin}} \left\| q_1 - O\left(q_2 \circ \gamma\right)\sqrt{\dot{\gamma}} \right\|^2$$

proposed by Kurtek and Srivastava (2014).

The following 1 illustrates the elastic Geodesic distance between two handwritten letters "R" and "g".

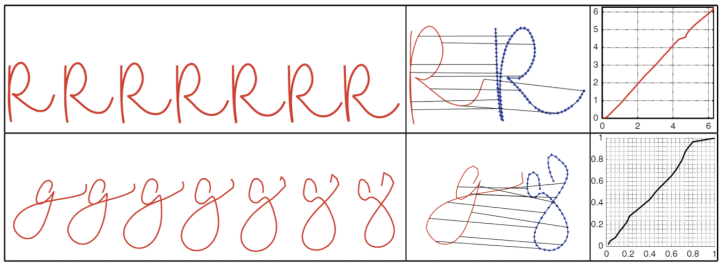

FIG 1. *Elastic Geodesic distance between two handwritten letters*

After registration by SRVF and calculating the Geodesic distance, the Karcher mean and covariance across curves can be computed for the simple image classification as follows:

$$[\mu_q] = \underset{[q] \in \mathcal{A}}{\arg\inf} \sum_{i=1}^{n} d_a\left([q], [q_i]\right)^2$$

2. MNIST Dataset. The dataset I selected for this project is MNIST (Modified National Institute of Standards and Technology) LeCun, Cortes and Burges (2010). This dataset is a large dataset of handwritten digits from 0 to 9, which is commonly used for training different image classification and processing tasks. It had become a benchmark dataset for evaluating new image classification algorithms. It contains 60,000 training images and 10,000 testing images. Each image is a 28 by 28 grayscale image pixel, and the true digital number labels each image. One R package dslabs provides a collection of datasets and functions for statistics and data science projects, including the MNIST dataset in R. The following graph 2 visualizes one random image from each digit.

3. Data Preprocessing. Before we can apply any Geometric Data Analysis to this MNIST dataset, we have to preprocess the images so that the format of the images is compliant with the function requirements specified in the related R packages such as imager, dtw, shapes, fdasrvf.

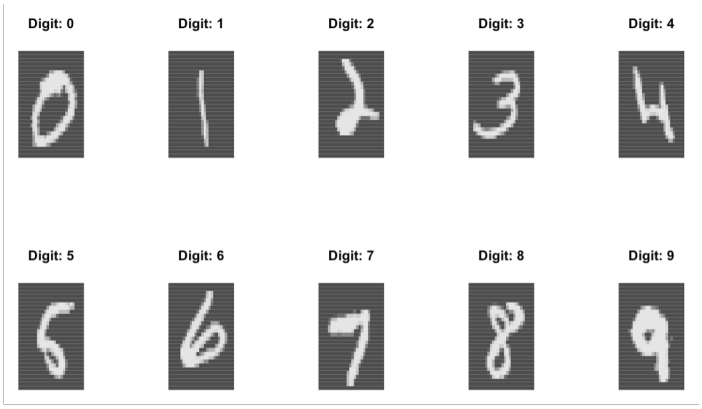

FIG 2. *One random image from each digit in R*

3.1. *Separation.* My first step was to separate the images by their true labels into each digital group. This makes it easier to calculate any Geometric Data Analysis for each class.

3.2. *Binary.* The second step is to convert the images into binary images. A simple "ifelse()" function in R sets all pixel values greater than 0 to 1 and otherwise to 0. The following figure 3 shows the binary images for each digit. We can tell the subtle difference between the binary and the original from the solid pixels on the boundary.

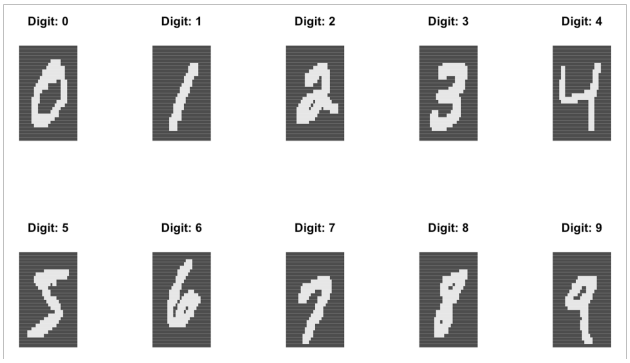

FIG 3. *Binary Images for each digit*

3.3. *Contour.* The third step is to extract the contour points of all images. The R package "imager" provides the function "contours()" to extract the contour points from all images. The following figure 4 shows the contour points along the boundary from the binary images.

3.4. *Resamples.* To conform to the format of some R functions for a group of image analysis, all images of the same group must have an equal number of contour points. This step is to apply the "approx()" function

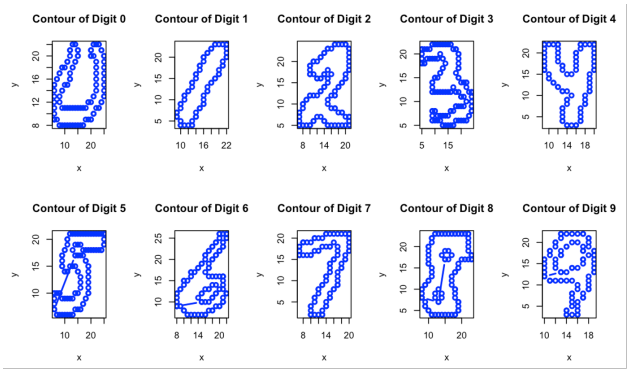

FIG 4. *The contour lines from binary images*

to linearly interpolate the contour points to make them an equal number of points. The following figure 5 shows the resample step.

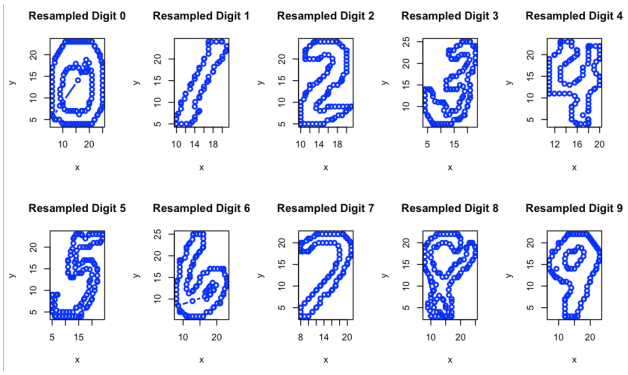

FIG 5. *Resampling images for all digits*

4. Calculate SRVF. The core representation in Geometric Data Analysis is the Square-Root Velocity Function (SRVF). It provides efficient ways of computing the geodesic distance between curves under the elastic Riemannian metric. The R function "curve_to_q()" in the package "fdasrvf" computes the SRVF space of a curve. The following figure 6 displays the SRVF pictures for each random digital image.

Then we took a look at the inverted digits from SRVF to see how much difference there is between the original digits and the inverted ones. The R function "q_to_curve()" in "fdasrvf" package converts the SRVF back to curves. The following figure 7 displays the converted random digits.

As we can see, the conversion of digits from SRVF still makes sense to us based on the outlines of each digit. We can tell the digit number by the images themselves. However, for the digit 1 and 7, they show somehow a bit of distortion from the standard digit form. The top part of the digit 1 curves towards the right. The middle part of the digit seven shows outward directions to both sides.

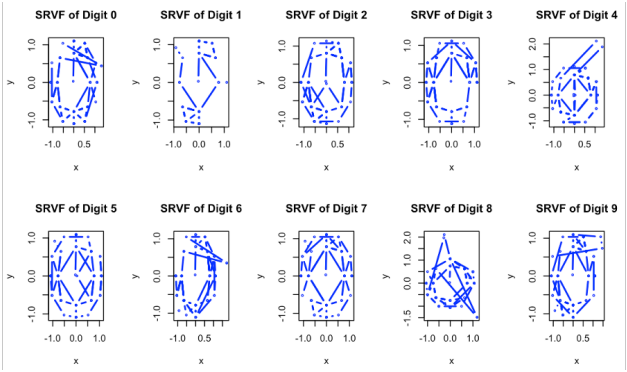

FIG 6. *SRVF for each random digital image*

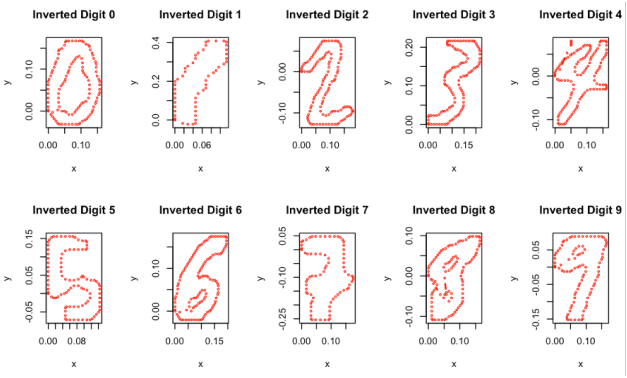

FIG 7. *Conversion of digits from SRVF*

5. Geodesic Path. After calculating the SRVF of all images, it's time to calculate the Geodesic path for each digit of two random images. The Geodesic path shows the trajectory of deformation that one curve needs to be as close as possible to another one under the SRVF in the Riemannian metric. The following figure 8 shows the geodesic path for each digit of two random images. To magnify the Geodesic path, I selected the digit seven as an example to illustrate this deformation path. The following figure 9 displays this deformation path.

6. Calculate Karcher Mean. A simple idea is to look at the Karcher mean of amplitudes for each digit. The Karcher mean can represent one class. The R function "curve_karcher_mean()" in the R package "fdasrvf" calculates the Karcher mean or median of a collection of curves using the elastic square-root velocity (SRVF) framework. This function allows us to set up the mode = "O", which means the input curves should be considered as open curves, rotated = FALSE, and scale = TRUE. The following 10 figure shows the Karcher Mean for each digit. As we can see, except for digit 6, the Karcher Mean can describe all other digits very well with our eyes. Digit 6 has some massive distortions and folding shapes. I assume

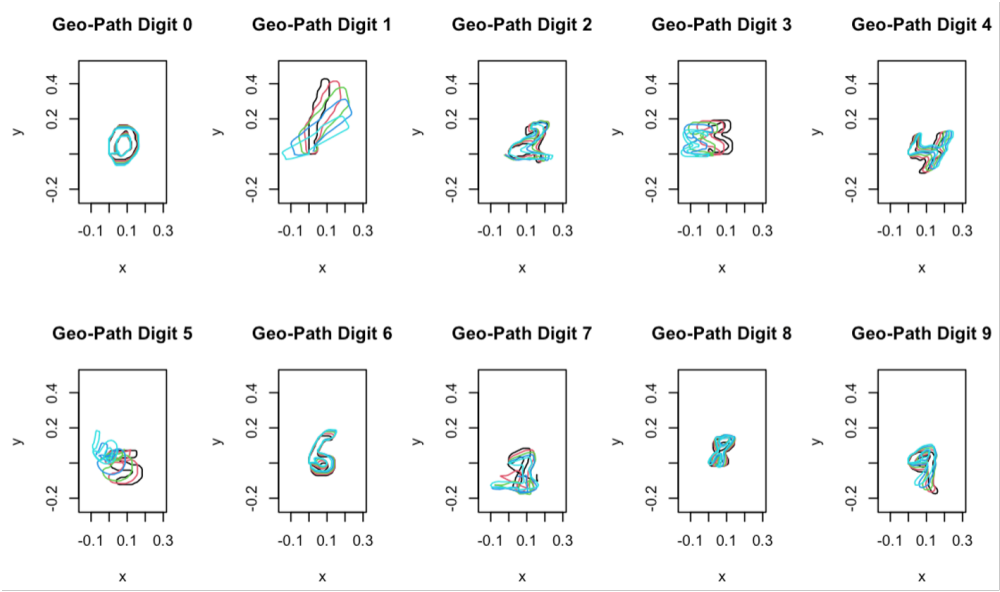

FIG 8. *Geodesic Path for each digit*

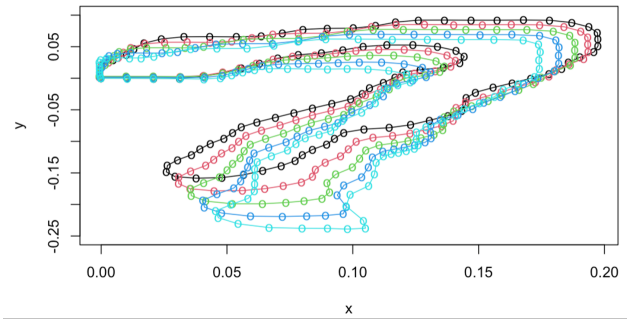

FIG 9. *Geodesic Path for digit 7*

the reason for digit six is that when people write digit six by hand, there are many variations and many deviations from the standard format of digit 6.

7. FPCA. I also calculated the principal directions of a set of curves for each digit. The principal directions of one group can be used to simplify a lot of analysis with less data and lower dimensions. The R function "curve_principal_directions()" in the R package "fdasrvf" calculates the principal directions of a set of curves. The following figure 11 shows the first three principal components for each digit. However, only digits zero, one, and four have some meaningful PCA in their figures.

8. Classification. Now it's time to build a simple classification based on this Karcher, which means classifying the new test image into different groups. The idea is that each new test image is calculated

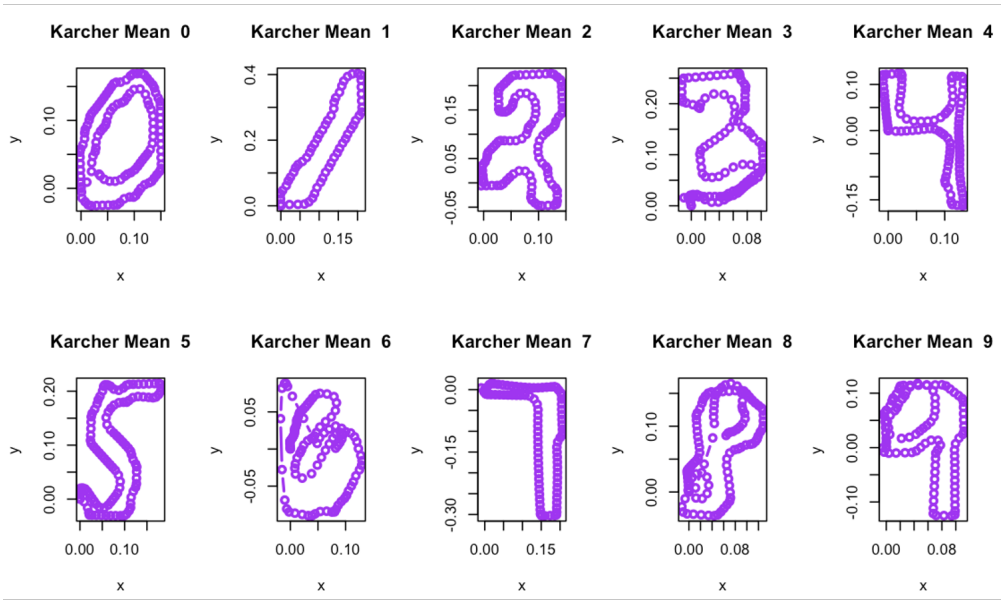

FIG 10. *The Karcher Mean for each digit*

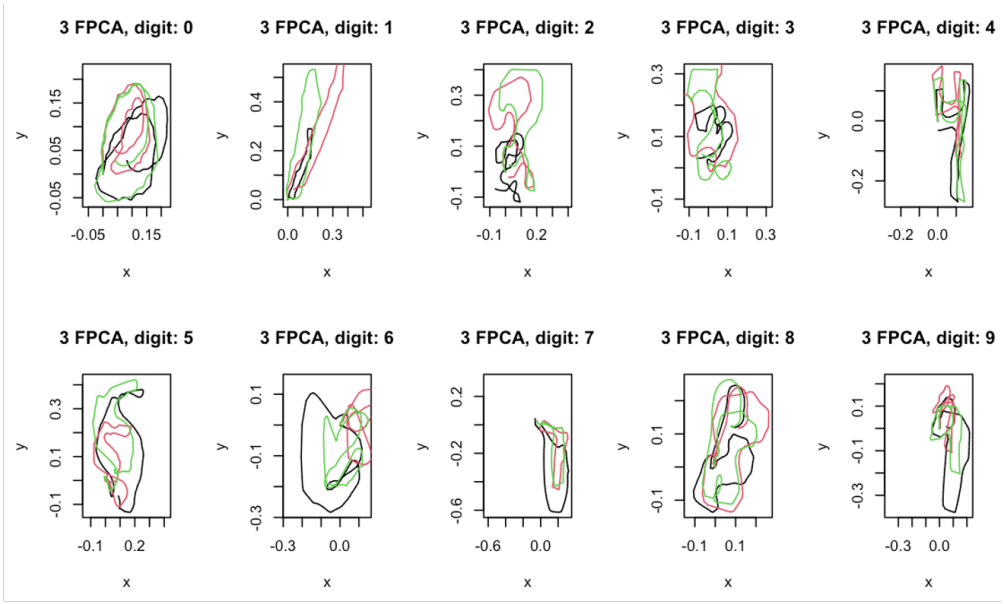

FIG 11. *Functional PCA for each digit*

by the Geodesic distance to each digital Karcher mean, and the minimal distance digit is used as the new prediction. Due to the slow computation of the elastic shape distance in R, only 100 test images are evaluated for each digit. The R packages "future" and "furrr" are used to build the double-nested parallel processing to compute the elastic distance. In total, 23 cores are used for parallel computing. In total, 2.4 hours were taken

to finish computations. The following figure 12 shows the prediction accuracy for each digit test group, and the overall accuracy for all digits is 74%.

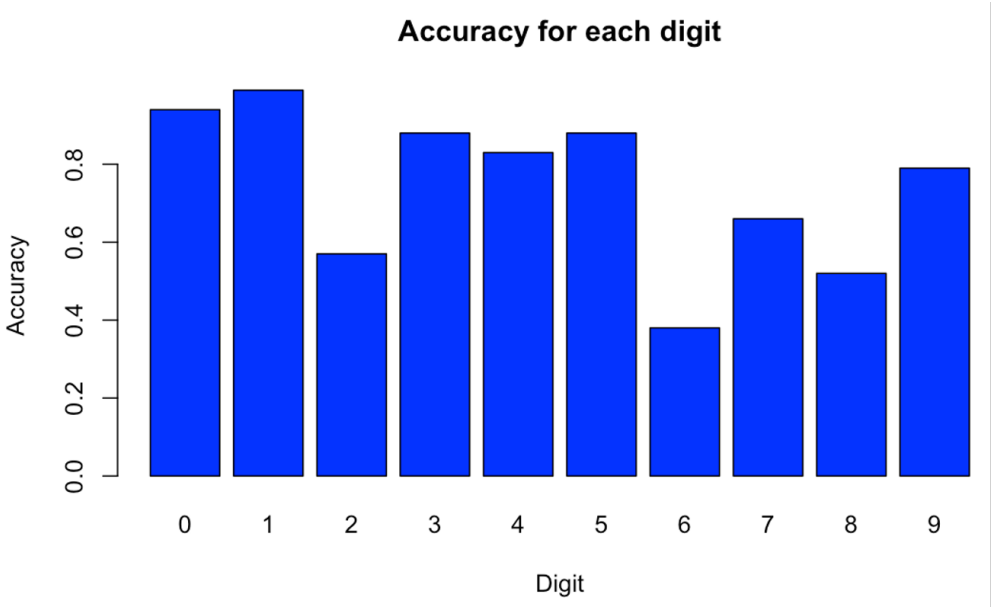

FIG 12. *Prediction Accuracy for each digit*

The confusion matrix for the prediction of each digit is shown below 13. While the index of this confusion matrix is from 1 to 10, the actual digit number is the index minus 1 for both the x and y axes. The y-axis is the true labels, and the x-axis is the predicted labels. As we can see, except for digits 2, 6, 7, and 8, all other digits have very good predictions given their highest numbers are on the diagonals. Many digit 2 images are classified into 4, 6, and 7. This is because, from the appearance, handwritten digit two can be very close to 4 and 7 on the right side of this number. In terms of digit 7, many digit 7s are classified into digit 2 for the same reason as digit 2.

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]
[1,]	94	0	0	1	1	0	4	0	0	0
[2,]	0	99	0	0	0	0	0	1	0	0
[3,]	0	0	57	1	14	0	12	11	1	4
[4,]	1	0	1	88	0	8	1	0	0	1
[5,]	0	1	1	4	83	1	2	0	1	7
[6,]	1	0	0	1	4	88	0	0	4	2
[7,]	1	1	4	3	2	1	38	10	2	38
[8,]	0	0	25	1	5	1	2	66	0	0
[9,]	9	0	3	1	5	1	14	2	52	13
[10,]	2	1	0	1	1	5	4	7	0	79

FIG 13. *Confusion Matrix*

9. Future Directions. This classification is a very simple and basic way, combining it with a machine learning method can be designed to improve accuracy, such as KNN or SVM. The computation time is very slow; an optimal algorithm must be designed to speed it up. There are two ways to speed up the computations. One is to reduce the number of resampled contour points for each digit so that computation in the geodesic distance can be much less. Another way to further classify the training set of each digit is into different subgroups to improve the Karcher mean on a more granular level, so that each subgroup can capture the subtle differences in handwritten forms for each digit. Another trial is the Karcher median for each digit group is calculated because the median can be more robust to the outliers and significant variations. This may help the better Karcher representation for digit 6.

10. Discussion. The final conclusion is that the Geodesic distance helps reveal the difference between curves and measure their deformation effectively. The Karcher means that it can be a good representative description for each class. However, this method is sensitive to the contour extractions. This method is an intensive computation.

REFERENCES

KURTEK, S. and SRIVASTAVA, A. (2014). Handwritten Text Segmentation Using Elastic Shape Analysis. In *2014 22nd International Conference on Pattern Recognition* 2501-2506. https://doi.org/10.1109/ICPR.2014.432

LECUN, Y., CORTES, C. and BURGES, C. (2010). MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist* **2**.