**International Journal of Electrical and Computer Engineering (IJECE)**
Vol. 11, No. 3, June 2021, pp. 2660~2665
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i3.pp2660-2665 

# A method to convert non-numeric characters into numerical values in dynamic time warping for string matching

**Hyun Jun Park**
Division of Software Convergence, Cheongju University, Cheongju, Korea

**Article Info**

***Article history:***

Received Jul 31, 2020
Revised Sep 22, 2020
Accepted Oct 11, 2020

***Keywords:***

Dynamic time warping
Image processing
Non-numeric data
Sequence matching
String matching

**ABSTRACT**

Dynamic time warping (DTW) is one of the well-known algorithms for measuring similarity between two temporal sequences, and it can be used for character matching. It uses a distance of two character strings. However, since the characters are non-numeric, it must be assigned to numerical values to calculate a distance between two character strings. Therefore, in this paper, we propose a method to convert non-numeric characters into numerical values in dynamic time warping for string matching. The proposed method uses normalized correlation coefficient, and it makes DTW gives more accurate results. Experimental results show that the proposed method gives excellent results.



***Corresponding Author:***

Hyun Jun Park,
Division of Software Convergence,
Cheongju University, Cheongju 28503, Korea
Email: hyunjun@cju.ac.kr

## 1. INTRODUCTION

Recent advances in deep learning are raising interest in how to solve problems in various fields using computer vision. As a result, a variety of applications is being created in artificial intelligence, robotics, factory automation, security, and so on [1-5]. Among them, one of the most widely used fields of computer vision is character recognition [6-8]. With the development of the character recognition technology using the convolutional neural network (CNN) [9, 10], the performance of the character recognition is much better than before. Printed documents or text can be recognized with high accuracy using image processing technology and convolutional neural network. However, if there is a lot of noise in the image, there is a character that is damaged enough to be not recognized, if the character is not hidden, it is still difficult to accurately recognize the character even with the latest technology. There may be partially misrecognized or unrecognized characters, and some characters may be added or dropped, so pure character recognition results are not available.

Therefore, in these cases post-processing is required. Various methods [11-13] can be used here. If there is a list of character strings to be recognized, the recognition result can be greatly improved. This is because misrecognized characters can be corrected by comparing the recognized result with a list to see if there are similar items. To do this, you need a method to compare list items and character strings to check similarity and find the most similar items. Methods of comparing the similarity of character strings include Jaccard similarity [14], Hamming distance [15], Jaro-Winkler distance [16], Levenshtein distance [17], and dynamic time warping [18].

In this paper, we match the list with the recognized results using dynamic time warping (DTW). DTW is one of the well-known algorithms for measuring similarity between two temporal sequences. DTW

has been used to solve various problems such as online signature recognition [19], shape matching [20], speech recognition [21], speaker recognition [22], and so on. The DTW algorithm calculates the similarity using the distance of two data. Therefore, in order to find the distance of data, the data must be represented as a numerical value. But what we want to match is a character string, which is not numeric data. Therefore, in order to use DTW, a method of converting nonnumeric characters to numeric values is required. Using the converted numeric values, the similarity is calculated with each item in the list and the most similar item can be found. This method has a great influence on the matching accuracy.

In this paper, we propose a method of converting nonnumeric characters into numerical data to use DTW. This method makes it possible to properly calculate the distance between characters by expressing the characters with appropriate numerical values using the shape differences of the characters. Therefore, by using the proposed method, DTW performance can be improved by converting non-numerical data into numerical data properly, and as a result, the final recognition result of incorrect character string recognized by computer vision technique can be improved.

## 2. DYNAMIC TIME WARPING

DTW is a method to calculate an optimal match between two given sequences using certain restrictions and rules [23]. The length of the character string recognized by the image processing may be different even for the same character string. However, DTW can match without considering the length of the two sequences, so you can use DTW to match the list. The DTW distance $DTW(Q, C)$ of two sequences $Q=\{q_1, \ldots, q_n\}$ and $C=\{c_1, \ldots, c_n\}$ of length $n$ is defined recursively as in (1) [24].

$$DTW(\{\},\{\}) = 0$$
$$DTW(Q,\{\}) = DTW(\{\},C) = \infty$$
$$DTW(Q,C) = \sqrt{dist(q_1,c_1) + \min\begin{cases} DTW(\{q_2,\ldots,q_n\},\{c_2,\ldots,c_n\}) \\ DTW(\{q_2,\ldots,q_n\},C) \\ DTW(Q,\{c_2,\ldots,c_n\}) \end{cases}} \quad (1)$$

where {} is null sequence, $dist(q_1, c_1)$ is the Euclidean distance between $q_1$ and $c_1$.

In general, DTW is implemented using dynamic programming technique. It calculates the warping matrix $\gamma(1\ldots n, 1\ldots n)$ as in (2) to get the DTW distance.

$$\gamma(i,j) = dist(q_i,c_i) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\} \quad (2)$$

Therefore, to calculate the warping matrix, it has to know the distance between two sequences ($dist(q_i, c_i)$). If the sequence is a numeric data, there is no problem to calculate the distance, but if the sequence is a non-numeric data, it has to be converted into numeric data.

## 3. METHODS TO CONVERT CHARACTERS INTO NUMERIC DATA

In this paper, DTW is used to improve the accuracy of the material code recognized using CNN in the noisy image as shown in Figure 1. There are 41 characters used in the material code, *0 ~ 9, A ~ Z,-, _, #, [,]*. The accuracy improvement depends on how these characters are converted to numeric data.


Figure 1. Images with material code

### 3.1. Method using ASCII code

The simplest method to convert a character to a numeric value is to use the American Standard Code for Information Interchange (ASCII) code. ASCII is one of the character codes, where every single bit represents a unique character. Table 1 shows ASCII code for some characters.

Table 1. ASCII code

| Char | '0' '1' '2' '3' '4' '5' '6' '7' '8' '9' 'A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' |
|---|---|
| ASCII | 48 49 50 51 52 53 54 55 56 57 65 66 67 68 69 70 71 72 73 74 75 |
| Char | 'L' 'M' 'N' 'O' 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z' '#' '-' '_' '[' ']' |
| ASCII | 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 35 45 137133135 |

Usually, computer represents characters using ASCII. Therefore, the distance between two characters can be simply calculated by ASCII difference. For example, the distance between the characters 'A' and 'Z' can be calculated as | 65-90 |.

### 3.2. Manual method

The ASCII method is simplest and fastest because there is no need to convert characters to numeric values. However, for character recognition, we should think about whether it is reasonable that the distance between 'A' and 'Z' is 25 and the distance between 'A' and 'B' is 1. When matching with lists using DTW, there is a significant difference between matching digitized strings and matching strings recognized by the CNN. There is no error in the digitized string. However, in character recognition using image processing, there may be characters added or missing due to misrecognition, unrecognition, and incorrect recognition. In particular, similarly shaped characters such as 'B' and '8', 'I' and 'l', 'S' and '5' are often incorrectly recognized.

For example, the recognition result of the string "[3033] N133S-13S-U" in the image using CNN, "[3033] N133S-135-U" was obtained. '5' came from misrecognition of 'S'. To improve the recognition result, DTW using ASCII was used to match incorrect strings and lists. As a result, the most similar item came out as "[3033] N143P-1B-U" (DTW distance = 19) even though there were "[3033] N133S-13S-U" in the list. 'S' and '5' can be misrecognized each other in CNN because the shape is similar. However, the distance using ASCII is 18 (| 53-35 |), so the other item is matched. If the distance difference between 'S' and '5' was small, DTW would have given the correct item.

Therefore, in order to obtain accurate results when matching strings recognized by CNN using DTW, the distance difference between characters that can be misrecognized should be small. It means that characters with similar shapes must have similar values. On the other hand, the distance difference between characters that do not easily misrecognized each other should be large. It means that the characters with different shapes should have a large difference. Table 2 shows the numeric values manually set for each character. 'B' and '8', 'I' and 'l', 'S' and '5', which are similar in shape, are set to similar values to reduce the distance difference, but the characters which have different shape such as '1' and '_' are set to different values to increase the distance.

Table 2. Manual setting

| Char | '0' '1' '2' '3' '4' '5' '6' '7' '8' '9' 'A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' |
|---|---|
| Num | 0 2 4 6 8 10 12 14 16 18 17.514.5 19 10 8 7 10.5 11 0.5 1 12 |
| Char | 'L' 'M' 'N' 'O' 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z' '#' '-' '_' '[' ']' |
| Num | 6.5 13 14 9 15 10 16 8.511.515.516.5 17 18.5 19 20.525 15 40 19 2.5 |

### 3.3. Character shape based method

In this paper, we propose a method to convert non-numeric character string to numerical values. Now we know that there should be a dependency on the character shapes and the numeric values, so the proposed method uses the shape of the characters to calculate the values. First, to calculate the numerical values based on the shapes of the characters, each character is created as an image as shown in Figure 2. Since the shape of the characters depends on the font, the result is calculated differently depending on the font. In this paper, the image is generated using Console font.

The proposed method uses the normalized correlation coefficient (NCC), which is widely used in template matching [25], to find the differences in character shapes. NCC is one of the methods for calculating the similarity between two images of the same size. It can be calculated using (3).


Figure 2. Character images used in this paper

$$\frac{1}{n}\sum_{x,y}\frac{\left(f(x,y)-\bar{f}\right)(g(x,y)-\bar{g})}{\sigma_f\sigma_g} \tag{3}$$

where $n$ is the number of pixels, $f$ and $g$ are the images, $f(x, y)$ and $g(x, y)$ are the pixels of the $(x, y)$ coordinates of the image, $\bar{f}$ and $\bar{g}$ are the average of the image pixels, and $\sigma_f$ and $\sigma_g$ are the standard deviation of the image $f$ and $g$.

Table 3 shows the calculated NCC values. The high NCC value indicates that the two characters have a more similar shape. The most similar to '0' is 'O', the most similar to '1' is 'I', '5' is 'S', '8' is 'B', and 'E' is 'F'. It shows that the similarity is appropriately represented. 41 × 41 NCC matrix should be reduced to one dimension. In this paper, the average value of NCC is calculated and numerical value is set for each character. Table 4 shows the calculated values.

Table 3. Calculated NCC

| | 'A' | 'B' | 'C' | 'F' | 'G' | 'H' | 'I' | 'J' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| '0' | 0.32 | 0.75 | 0.62 | 0.50 | 0.70 | 0.57 | 0.34 | 0.47 | 0.51 | 0.55 | **0.81** | 0.61 | 0.74 | 0.62 | 0.57 | 0.27 | 0.71 |
| '1' | 0.27 | 0.38 | 0.43 | 0.30 | 0.42 | 0.15 | **0.75** | 0.33 | 0.21 | 0.33 | 0.33 | 0.28 | 0.29 | 0.35 | 0.47 | 0.58 | 0.21 |
| '5' | 0.34 | 0.74 | 0.45 | 0.62 | 0.55 | 0.37 | 0.51 | 0.62 | 0.28 | 0.38 | 0.49 | 0.47 | 0.44 | 0.55 | **0.81** | 0.40 | 0.39 |
| '8' | 0.43 | **0.82** | 0.52 | 0.58 | 0.60 | 0.46 | 0.40 | 0.55 | 0.42 | 0.48 | 0.63 | 0.60 | 0.57 | 0.68 | 0.76 | 0.30 | 0.53 |
| 'E' | 0.36 | 0.72 | 0.66 | **0.86** | 0.60 | 0.41 | 0.54 | 0.47 | 0.29 | 0.38 | 0.47 | 0.60 | 0.43 | 0.63 | 0.62 | 0.38 | 0.40 |

Table 4. Character shape based setting

| Char | '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9' | 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Num | 0.27 | 0.13 | 0.19 | 0.21 | 0.11 | 0.23 | 0.23 | 0.13 | 0.27 | 0.22 | 0.11 | 0.30 | 0.19 | 0.23 | 0.24 | 0.20 | 0.23 | 0.17 | 0.16 | 0.17 | 0.16 |
| Char | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z' | '#' | '-' | '_' | '[' | ']' | |
| Num | 0.13 | 0.15 | 0.18 | 0.23 | 0.21 | 0.20 | 0.24 | 0.23 | 0.11 | 0.20 | 0.10 | 0.15 | 0.13 | 0.11 | 0.19 | 0.14 | 0.00 | 0.02 | 0.03 | 0.04 | |

## 4. EXPERIMENTAL RESULTS

The experimental environment for evaluating the proposed method is as follows. We used laptop computer with Intel i7-7500U 2.70 GHz CPU and 8.0 GB RAM, and the proposed method is implemented in Visual Studio 2017 with MFC-based C ++ language. OpenCV 3.4.2 was used as an image processing library. In the experiment, a total of 1.309 images including the material code shown in Figure 1 were used, and the resolution was 1920×1080.

Since there is a lot of noise in the image, the recognition rate is about 30% when all the characters are perfectly recognized. To improve this, we used DTW to match the material code list. Therefore, the recognition success in this paper is defined as follows. The character string recognized in the image is matched with the material code list using the DTW. If the top five items with a small DTW distance contain the correct material code, the recognition is successful. Figure 3 shows the results of DTW matching using the proposed method. The characters on the left are strings recognized using CNN, and the DTW distance and matching results are shown on the right. Table 5 shows the accuracy according to the method of converting characters to numerical value.

The DTW using ASCII successfully matched character strings in 975 of 1,309 images. It improved the pure recognition rate, which was about 30%, to 74.48%. It can be seen that DTW effectively matches strings. The DTW using the manual method correctly matched 1,238 images and showed a 94.58% accuracy. It means that the DTW matching results are improved when the characters with similar shapes are set to similar values as analyzed in this paper. Finally, the proposed method was experimented, and it showed a 95.49% success rate by accurately matching 1,250 images. This means that the normalized correlation coefficient used to compare the similarity of character shapes properly represented the similarity of characters, and it gives better DTW matching results than other methods.

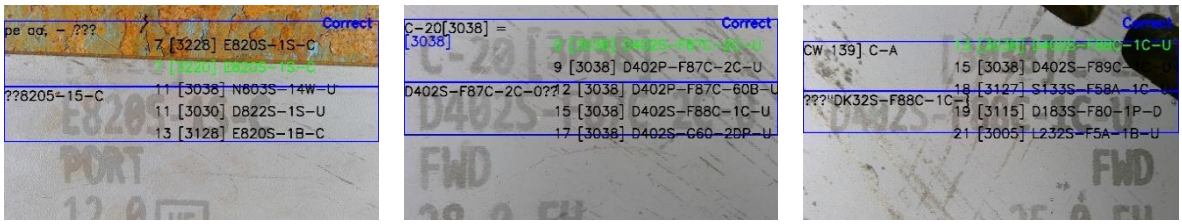


Figure 3. Results of the DTW matching using the proposed method

Table 5. Accuracy according to the converting method

| Method | Success/Total | Accuracy |
|---|---|---|
| ASCII | 975/1309 | 74.48% |
| Manual | 1238/1309 | 94.58% |
| Proposed | 1250/1309 | 95.49% |

## 5. CONCLUSION

In this paper, we proposed a method of converting non-numeric characters into numerical values when using DTW to improve the recognition rate of incorrectly recognized character strings in images. As a result of analyzing the DTW result for string matching, it was found that the matching accuracy is improved by assigning similar values to characters with similar shapes. Therefore, the proposed method calculates the shape similarity between characters using normalized correlation coefficient, and sets numerical values for each character using calculated values. Experimental results showed that the proposed method gives better results (95.49%) than simply converting characters to ASCII (74.48%) or manually converting letters to numbers (94.58%). The proposed method improved the pure recognition rate, which was about 30%, to 95.49%. Therefore, we can confirm that DTW matching using the proposed method is very effective for string matching. However, in the proposed method, since the average is simply used in the process of converting the similarity calculated using NCC into one dimension, it may not accurately reflect the similarity difference of each character. Although the proposed method shows good results, but it is not optimal, so further study on this is needed.

## REFERENCES

[1] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197-387, 2014.
[2] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375-9389, 2018.
[3] D. Yu and L. Deng, "Deep learning and its applications to signal and information processing [exploratory dsp]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145-154, 2011.
[4] E. Rahimy, "Deep learning applications in ophthalmology," *Current opinion in ophthalmology*, vol. 29, no. 3, pp. 254-260, 2018.
[5] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017, pp. 759-760.
[6] J. Sun, Y. Ma, H. Yang, N. Wang, Y. Ding, A. Song, Y. Zhu, and J. Xi, "Character Recognition Method for Low-Contrast Images of Numerical Instruments," *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, Chongqing, 2018, pp. 157-162.
[7] R. Baran, P. Partila, and R. Wilk, "Automated text detection and character recognition in natural scenes based on local image features and contour processing techniques," *Intelligent Human Systems Integration*, pp. 42-48, 2018.
[8] X. Y. Zhang, Y. Bengio, and C. L. Liu, "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark," *Pattern Recognition*, vol. 61, pp. 348-360, 2017.
[9] N. S. Rani, N. Chandan, A. S. Jain, and H. R. Kiran, "Deformed character recognition using convolutional neural networks," *International Journal of Engineering & Technology*, vol. 7, no. 3, pp. 1599-1604, 2018.
[10] H. H. Kim, J. K. Park, J. H. Oh, and D. J. Kang, "Multi-task convolutional neural network system for license plate recognition," *International Journal of Control Automation and Systems*, vol. 15, no. 6, pp. 2942-2949, 2017.
[11] S. Drobac and K. Lindén, "Optical character recognition with neural networks and post-correction with finite state methods," *International Journal on Document Analysis and Recognition*, vol. 23, no. 4, pp. 279-295, 2020.
[12] A. Marial and J. Jibrael, "Feature extraction of optical character recognition: survey," *International Journal of Applied Engineering Research*, vol. 12, no. 7, pp. 1129-1137, 2017.
[13] A. Priya, S. Mishra, S. Raj, S. Mandal, and S. Datta, "Online and offline character recognition: A survey," *2016 International Conference on Communication and Signal Processing (ICCSP)*, Melmaruvathur, 2016, pp. 0967-0970.

[14] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," *The 2013 IAENG International Conference on Internet Computing and Web Services (ICICWS'13)*, vol. 1, no. 6, 2013, pp. 380-384.
[15] M. Norouzi, D. J. Fleet and R. R. Salakhutdinov, "Hamming distance metric learning," *Advances in neural information processing systems*, pp. 1061-1069, 2012.
[16] M. Rajabzadeh, S. Tabibian, A. Akbari, and B. Nasersharif, "Improved dynamic match phone lattice search using Viterbi scores and Jaro Winkler distance for keyword spotting system," *The 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP 2012)*, Shiraz, Fars, 2012, pp. 423-427.
[17] L. Yujian and L. Bo, "A normalized Levenshtein distance metric," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1091-1095, 2007.
[18] P. Anantasech and C. A. Ratanamahatana, "Enhanced Weighted Dynamic Time Warping for Time Series Classification," *Third International Congress on Information and Communication Technology*, vol. 797, pp. 655-664, 2019.
[19] R. Al-Hmouz, W. Pedrycz, K. Daqrouq, A. Morfeq, and A. Al-Hmouz, "Quantifying dynamic time warping distance using probabilistic model in verification of dynamic signatures," *Soft Computing*, vol. 23, no. 2, pp. 407-418, 2019.
[20] J. Zhao and L. Itti, "Shapedtw: shape dynamic time warping," *Pattern Recognition*, vol. 74, pp. 171-184, 2018.
[21] V. M. Kumar and D. S. H. Thipesh, "Robot ARM performing writing through speech recognition using dynamic time warping algorithm," *International Journal of Engineering, Transactions B: Applications*, vol. 30, no. 8, pp. 1238-1245, 2017.
[22] M. Adel, M. Afify, A. Gaballah, and M. Fayek, "Text-Independent Speaker Verification Based on Deep Neural Networks and Segmental Dynamic Time Warping," *2018 IEEE Spoken Language Technology Workshop (SLT)*, Athens, Greece, 2018, pp. 1001-1006.
[23] M. Müller, "Dynamic time warping," in *Information retrieval for music and motion*. Berlin, Springer, ch. 4, 2007, pp. 69-84.
[24] M. Lee, S. Lee, M. J. Choi, Y. S. Moon, and H. S. Lim, "HybridFTW: hybrid computation of dynamic time warping distances," *IEEE Access*, vol. 6, pp. 2085-2096, 2017.
[25] R. Brunelli, "Template matching techniques in computer vision: theory and practice," *John Wiley & Sons*, 2009.

## BIOGRAPHY OF AUTHOR

**Hyun Jun Park** He received his M.S. and Ph.D. degrees from the Department of Computer Engineering, Pusan National University, Busan, Korea, in 2009 and 2017, respectively. In 2017, he was a postdoctoral researcher at BK21PLUS, Creative Human Resource Development Program for IT Convergence, Pusan National University, Korea. From 2018 to the present, he is an associate professor at the Division of Software Convergence, Cheongju University, Korea. His research interests include computer vision, image processing, factory automation, neural network, and deep learning applications.