International Journal of Advanced Culture Technology Vol.7 No.2 209-217 (2019)
DOI 10.17703/IJACT.2019.7.2.209

IJACT 19-6-26

Character Classification with Triangular Distribution

Suk Won Yoo

SeoKyeong Univ., Dept. of Software, Seoul, Korea
swyoo@skuniv.ac.kr

Abstract

Due to the development of artificial intelligence and image recognition technology that play important roles in the field of 4th industry, office automation systems and unmanned automation systems are rapidly spreading in human society. The proposed algorithm first finds the variances of the differences between the tile values constituting the learning characters and the experimental character and then recognizes the experimental character according to the distribution of the three learning characters with the smallest variances. In more detail, for 100 learning data characters and 10 experimental data characters, each character is defined as the number of black pixels belonging to 15 tile areas. For each character constituting the experimental data, the variance of the differences of the tile values of 100 learning data characters is obtained and then arranged in the ascending order. After that, three learning data characters with the minimum variance values are selected, and the final recognition result for the given experimental character is selected according to the distribution of these character types. Moreover, we compare the recognition result with the result made by a neural network of basic structure. It is confirmed that satisfactory recognition results are obtained through the processes that subdivide the learning characters and experiment characters into tile sizes and then select the recognition result using variances.

Keywords: *Image Comparison, Classification, Character Recognition, Feature Extraction, Distribution, Neural Network, Machine Learning*

1. Introduction

Today, due to the development of science and technology, people use automation machines or robots in many fields. The spread of intelligent automation systems [1] has made human life more convenient, and the things that people are doing are being replaced by these automation systems. In particular, the character recognition technology [2] used in document automation is used in many other fields. For example, the existing documents are automatically recognized and classified [3] in the public organization, the factory automatically recognizes and classifies the goods [4], the bank automatically processes the issued checks [5], and the transportation system automatically recognizes [6] car plate numbers using camera systems to track vehicles [7] that have been subjected to signal violations, to handle accidents, and to prevent over-speeding. It is possible to obtain a high recognition rate when specific fonts are used in these fields. Numerous studies have been carried out for number recognition because number sequence does not have any word arrangement and any context meaning, and also the recognition error might cause serious problem.

As a research method for character recognition, it might be classified into a method of finding a pattern that determines characteristics of characters [8], a method using neural network [9], a method using deep

Manuscript received: May 06, 2019 /revised: May 17, 2019 / Accepted: May 23, 2019
Corresponding Author: swyoo@skuniv.ac.kr
Tel:+82-2-940-7751
SeoKyeong Univ., Dept. of Software, Korea

learning [10], and a method using a chaos theory [11]. The method of finding a pattern that determines the characteristics of a character is to detect the positional features of the character by examining pixels on vertical lines or horizontal lines, or to extract the skeleton of the character using thinning techniques to define characteristics of the character [12]. However, this method has a disadvantage that it is limited to a specific font. The method of machine learning using neural network is one of the most widely used typical research method. Because it uses a method that mimics human discrimination and cognitive abilities, it might recognize various new fonts. However, this method has several drawbacks in that 1) it requires a lot of learning time, 2) it does new learning procedure to add new learning data, 3) different recognition result might be obtained for the same experimental data depending on the number of nodes of hidden layer composing neural network structure, and 4) different recognition result might be obtained depending on how to connect edges of neural network and also depending on initial values assigned randomly to these edges as weights. In addition, studies using chaos theory have been carried out to recognize changes in the parts constituting characters using fractal techniques. However, in spite of various researches on character recognition, there is no way to fully recognize characters with artificially modified fonts. In this paper, we propose an algorithm that obtains the variances of tile values composing learning characters and experiment character, and then recognizes experimental characters according to the distribution of three characters with the smallest variance values. The expected advantages of the proposed classification method are that it is easy to include new learning data, it obtains a high recognition rate, and same recognition result is obtained for the same experimental data.

2. The Main Subject

For 100 number of learning characters and 10 number of experimental characters with a new font not used for those learning characters, each character is defined as the number of black pixels belonging to 15 tile areas. Then, the information for learning characters is stored in the array LearningData and the information for the experimental character is stored in the array TestData. For each experimental character, variances of the differences of those tile values with the 100 learning characters are obtained and then arranged in the ascending order. Then, the 3 learning characters with the minimum variance values are selected, and the final recognition result for the experimental character is selected according to the distribution of these 3 selected learning characters. Finally, we use a basic neural network composed of 16 input layer nodes, 20 hidden layer nodes, and 1 output layer node in order to compare the recognition result with the result made by the proposed classification algorithm.

2.1 Character Classification with Triangular Distribution

The character classification with triangular distribution algorithm will be described in more detail in the following steps.

Step 1) Learning characters consist of 10 numbers, and each number has 10 different fonts. So, the learning characters consist of a total of 100 characters. Each learning character is 15x15 in size, and it shows the text area in black color on a white background. The experimental characters consists of 10 numbers with a new font not used for the learning characters. Each experimental character is also 15x15 in size with a black text on a white background.

Step 2) Each learning character is divided into 25 tiles of 3x3 size, and the number of black pixels belonging to the 15 tile regions in the middle portion are counted. Then, each learning character can be defined as the number of these black pixels belonging to 15 tile regions. In the same way, each experimental character is also divided into 25 tile regions, and it is defined as the number of black pixels belonging to the 15 tile regions in the middle portion.

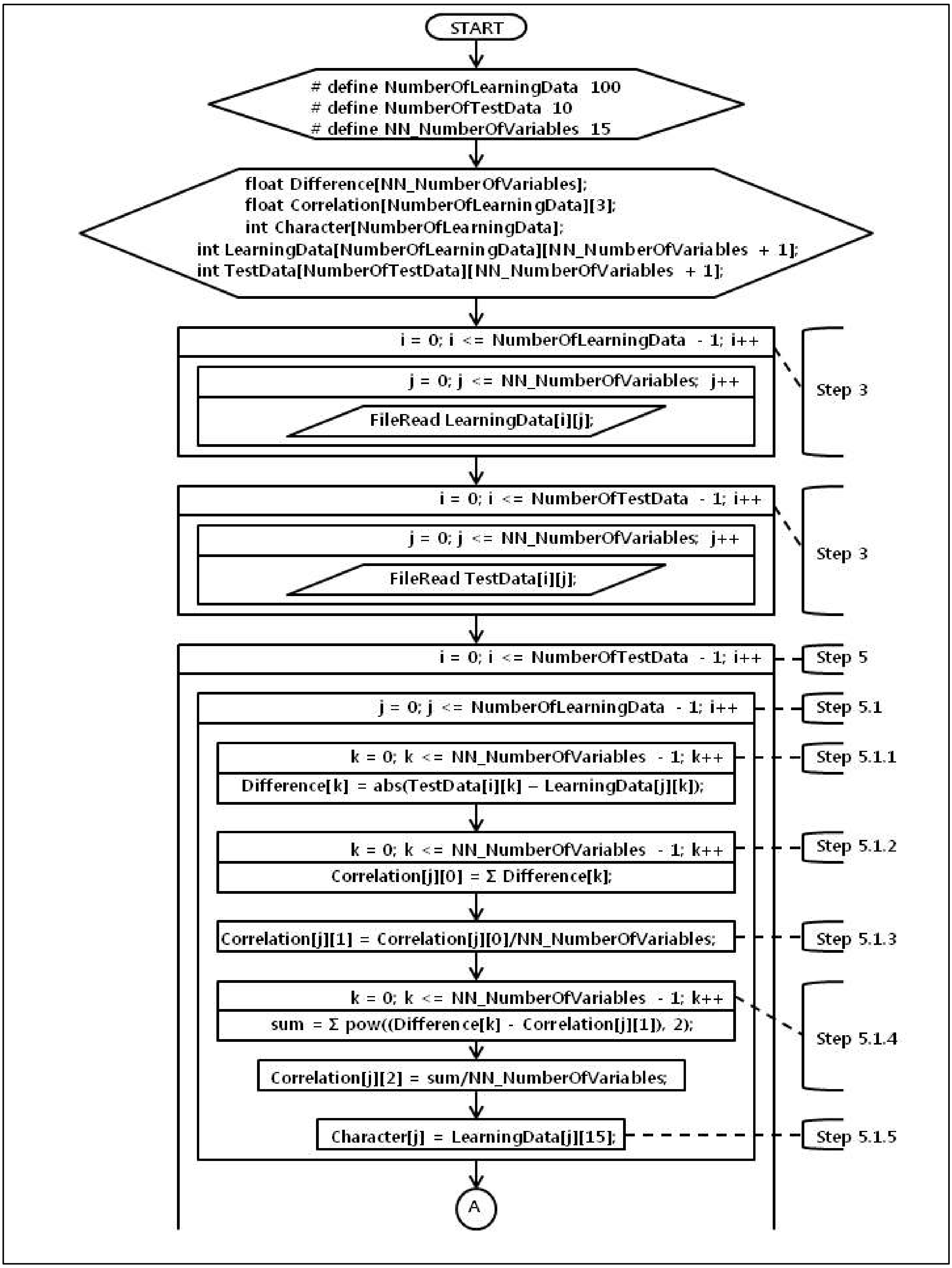
START
# define NumberOfLearningData 100
# define NumberOfTestData 10
# define NN_NumberOfVariables 15
float Difference[NN_NumberOfVariables];
float Correlation[NumberOfLearningData][3];
int Character[NumberOfLearningData];
int LearningData[NumberOfLearningData][NN_NumberOfVariables + 1];
int TestData[NumberOfTestData][NN_NumberOfVariables + 1];
i = 0; i <= NumberOfLearningData - 1; i++
j = 0; j <= NN_NumberOfVariables; j++
FileRead LearningData[i][j];
Step 3
i = 0; i <= NumberOfTestData - 1; i++
j = 0; j <= NN_NumberOfVariables; j++
FileRead TestData[i][j];
Step 3
i = 0; i <= NumberOfTestData - 1; i++
Step 5
j = 0; j <= NumberOfLearningData - 1; i++
Step 5.1
k = 0; k <= NN_NumberOfVariables - 1; k++
Difference[k] = abs(TestData[i][k] – LearningData[j][k]);
Step 5.1.1
k = 0; k <= NN_NumberOfVariables - 1; k++
Correlation[j][0] = Σ Difference[k];
Step 5.1.2
Correlation[j][1] = Correlation[j][0]/NN_NumberOfVariables;
Step 5.1.3
k = 0; k <= NN_NumberOfVariables - 1; k++
sum = Σ pow((Difference[k] - Correlation[j][1]), 2);
Correlation[j][2] = sum/NN_NumberOfVariables;
Step 5.1.4
Character[j] = LearningData[j][15];
Step 5.1.5
A

Figure 1. Flowchart

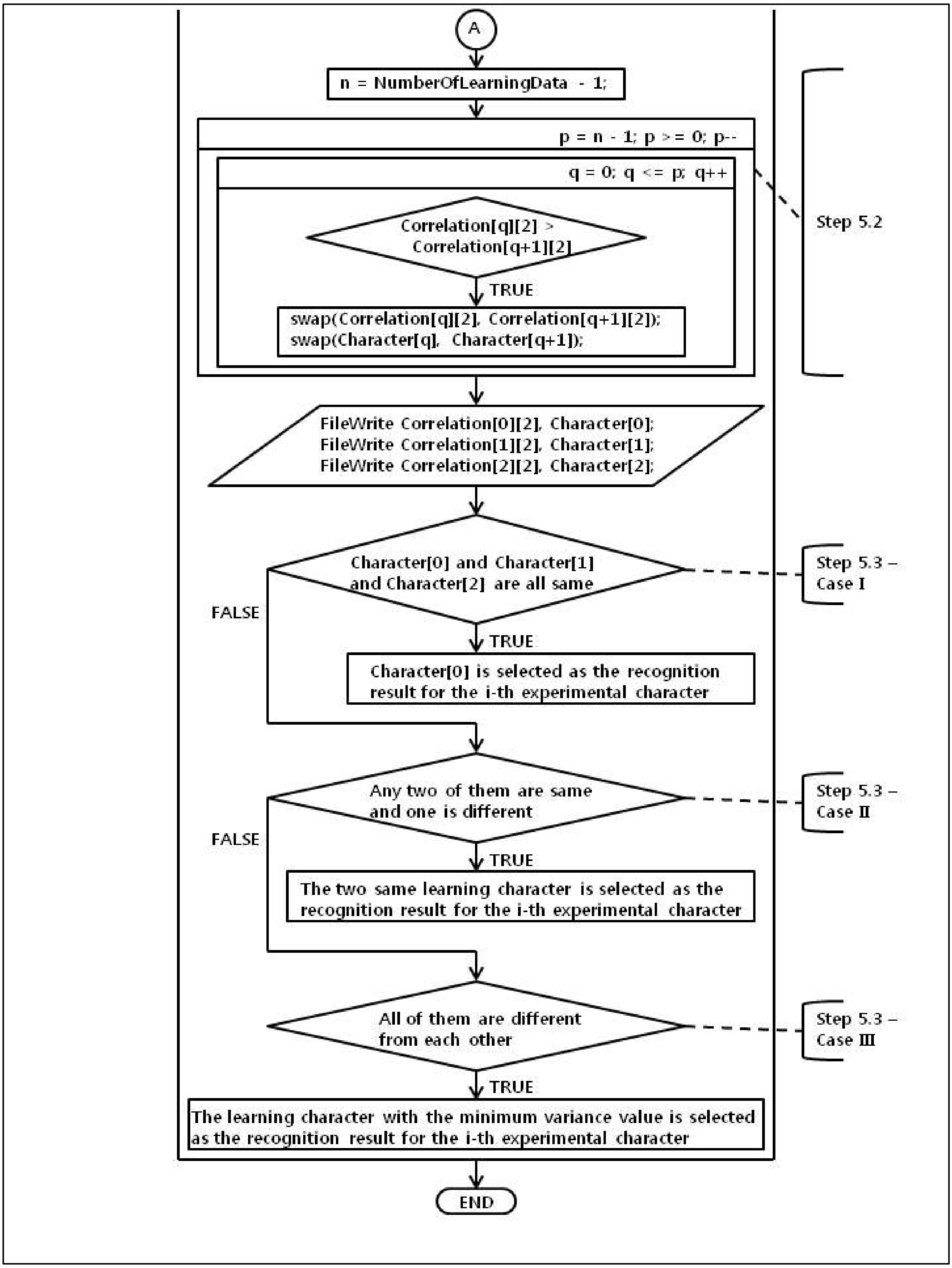
A
n = NumberOfLearningData - 1;
p = n - 1; p >= 0; p--
q = 0; q <= p; q++
Correlation[q][2] > Correlation[q+1][2]
TRUE
swap(Correlation[q][2], Correlation[q+1][2]);
swap(Character[q], Character[q+1]);
Step 5.2
FileWrite Correlation[0][2], Character[0];
FileWrite Correlation[1][2], Character[1];
FileWrite Correlation[2][2], Character[2];
Character[0] and Character[1] and Character[2] are all same
FALSE
TRUE
Character[0] is selected as the recognition result for the i-th experimental character
Step 5.3 – Case I
Any two of them are same and one is different
FALSE
TRUE
The two same learning character is selected as the recognition result for the i-th experimental character
Step 5.3 – Case II
All of them are different from each other
TRUE
The learning character with the minimum variance value is selected as the recognition result for the i-th experimental character
Step 5.3 – Case III
END

Figure 1. (continued) Flowchart

Step 3) Each of the 100 learning characters is defined as the number of black pixels belonging to 15 tile areas now. After declaring the array LearningData[100][16] of integer data type, the numbers of black pixels belonging to the 15 tile areas for k-th learning character are stored in order from LearningData[k][0] to LearningData[k][14](for k = 0 to 99). The last element of the array, LearningData[k][15], stores the corresponding character. In other words, the k-th learning character with LearningData[k][15] is defined as a set of 15 numbers from LearningData[k][0] to LearningData[k][14]. Similarly, for the 10 experimental characters, the array TestData[10][16] of integer data type is declared, and the numbers of black pixels belonging to the 15 tile areas for the k-th experiment character are stored in order from TestData[k][0] to TestData[k][14](for k = 0 to 9). TestData[k][15], the last element of the array, stores -1 as an undecided value.

Step 4) Three Arrays, Difference[15] of integer data type, Correlation[100][3] of real data type, and Character[100] of integer data type are declared.

Step 5) For the i-th experimental character (for i = 0 to 9),

Step 5.1) For the j-th learning character (for j = 0 to 99),

Step 5.1.1) For the k-th tile (for k = 0 to 14), the difference between the k-th tile value of the i-th experimental character and the k-th tile value of the j-th learning character is stored in the array Difference[k].

Step 5.1.2) Sum of the values in the array Difference[] is calculated and stored in Correlation[j][0].

Step 5.1.3) Average of the values in Difference[] is calculated and stored in Correlation[j][1].

Step 5.1.4) Variance is calculated using the values in Difference[] and the average in Correlation[j][1], and stored in Correlation[j][2].

Step 5.1.5) Corresponding learning character is stored in Character [j].

Step 5.2) The array Correlation is sorted in ascending order based on Correlation[][2], where variance values are stored. At this time, the corresponding array Character[] is also sorted in ascending order.

Step 5.3) For the three types of learning characters, Character[0], Character[1], and Character[2], in ascending order, there are 3 possible cases:

Case I) If all three kinds of learning characters are the same, then the corresponding learning character is selected as the recognition result for the i-th experimental character.

Case II) If two kinds of learning characters are the same character and one is different, then the two same learning character is selected as recognition result for the i-th experimental character.

Case III) If all three types of learning characters are different from each other, then the learning character with the minimum variance value is selected as the recognition result for the i-th experimental character.

For reference, when determining the recognition result for the i-th experimental character, we did not use Character[0], which is the first element of the array Character sorted in ascending order. The reason is that the numbers stored in the Character[0], Character[1], and Character[2] are not the values of the operation result but merely the kind of learning characters. So, it considers the distribution of these three character types, not the values of these three characters. More detailed descriptions for these steps are given by the flowchart in Fig. 1.

Step 6) We construct a neural network to compare recognition result with the result of the character classification method proposed in this paper. The neural network basically has a structure composed of 16 input layer nodes, 20 hidden layer nodes, and 1 output layer node. The reason why the number of input layer nodes is 16 is because one node is added for bias, and the number of output layer node is one because it represents a character as recognition result. By the rule of the neural network learning method, the neural network uses these 100 learning characters and corresponding characters as a set of inputs and outputs, and repeated learning is performed on the neural network until all these learning characters are completely recognized as corresponding characters. After the neural network is constructed through this repetitive learning process, 10 experimental characters are sequentially provided as the input data of the neural network, and the execution results for the given experimental characters are obtained.

2.2 The Results

100 number of learning characters [13] consist of 10 numbers, and each number has 10 different fonts. 10 number of experimental characters [13] also consist of 10 numbers with a new font not used in the learning characters. The proposed classification algorithm stores the values of the 15 tile regions for each learning character in the array LearningData, and also does the values of the 15 tile regions for each experimental character in the array TestData. For each experimental character, the algorithm finds variances of the difference values for each of the 100 learning characters and then sorts the results in ascending order. The classification algorithm compares the three kinds of learning characters that have the smallest variance values, and it finally selects one of them as a recognition result for the given experimental character.

Table 1. Output of 10 Number of Experimental Characters Obtained by Execution of Character Classification Algorithm

TD#	Var	Char	Var	Char	Var	Char	Result
0	0.00000	0	0.00000	0	0.42667	1	0
1	0.37333	1	0.64889	1	0.64889	0	1
2	0.69333	2	1.09333	8	1.26222	2	2
3	0.22222	3	0.48889	3	0.51556	9	3
4	0.38222	4	0.48889	4	0.64000	6	4
5	1.04889	2	1.04889	5	1.04889	7	2,5,7
6	0.19556	6	0.38222	6	0.51556	8	6
7	0.50667	7	0.50667	7	0.86222	7	7
8	0.16000	8	0.51556	8	0.64000	8	8
9	0.19556	9	0.72889	9	0.72889	8	9

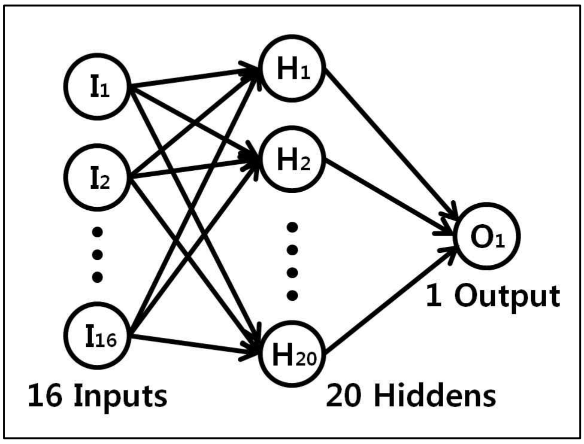

Figure 2. Neural Network with 16 Input Nodes, 20 Hidden Nodes and 1 Output Node

Table 1 shows the results obtained by the proposed classification algorithm for 100 number of learning characters and 10 number of experimental characters. The table shows that the experimental character 5 is recognized as 2, 5, and 7, and 100% recognition result has obtained for other experimental characters except 5. For the operation on experimental character 7, the three minimum variance values are 0.50667, 0.50667, and 0.86222, and the corresponding three types of learning characters are 7, 7, and 7. Therefore, since the three kinds of learning character types are all the same, the learning character 7 is selected as the recognition result for the experimental character 7. For the operation on experimental character 2, the three minimum variance values are 0.69333, 1.09333, and 1.26222, and the corresponding three types of learning characters are 2, 8, and 2. Therefore, the learning character 2 is selected as the recognition result for the experimental character 2 since two kinds of the three learning characters are the same character 2, and one is the other character 8. And, for the calculation for experimental character 5, the three minimum variance values are 1.04889, 1.04889, 1.04889, and the corresponding three types of learning characters are 2, 5, and 7. Therefore, since the three kinds of learning characters are all different, a learning character with a minimum variance should be selected as a recognition result for the experimental character 5. However in this case, since all the variance values of the selected three learning characters are equal to 1.04889, the experimental character 5 is recognized as learning characters 2, 5, and 7 redundantly.

Table 2. Output of 100 Number of Learning Characters Obtained by Execution of Neural Network After 2,000,000 or More Number of Repetition of Learning

#	Char	Output	#	Char	Output	#	Char	Output	#	Char	Output
0	0	0.00009	25	2	1.99998	50	5	4.99785	75	7	7.00184
1	0	0.00001	26	2	1.99990	51	5	4.99996	76	7	6.99995
2	0	-0.00001	27	2	2.00082	52	5	4.99997	77	7	7.00025
3	0	-0.00002	28	2	1.99999	53	5	4.99996	78	7	7.00000
4	0	0.00002	29	2	2.00003	54	5	4.99997	79	7	7.00013
5	0	0.00000	30	3	3.00000	55	5	4.99990	80	8	8.00044
6	0	0.00000	31	3	3.00000	56	5	4.99996	81	8	8.00001
7	0	0.00000	32	3	3.00000	57	5	4.99997	82	8	8.00001
8	0	0.00000	33	3	3.00119	58	5	5.00171	83	8	8.00002
9	0	-0.00002	34	3	2.99801	59	5	5.00037	84	8	8.00001
10	1	1.00000	35	3	2.99999	60	6	5.99997	85	8	8.00001
11	1	0.99840	36	3	2.99998	61	6	5.99997	86	8	8.00001
12	1	0.99994	37	3	2.99998	62	6	6.00000	87	8	8.00183
13	1	0.99966	38	3	3.00049	63	6	6.00003	88	8	7.99991
14	1	1.00195	39	3	3.00030	64	6	6.00000	89	8	7.99794
15	1	1.00000	40	4	3.99987	65	6	6.00002	90	9	9.00000
16	1	1.00000	41	4	4.00001	66	6	6.00001	91	9	9.00000
17	1	0.99994	42	4	3.99940	67	6	6.00000	92	9	9.00000
18	1	1.00000	43	4	4.00069	68	6	6.00006	93	9	9.00000
19	1	1.00000	44	4	4.00000	69	6	6.00000	94	9	9.00000
20	2	2.00000	45	4	3.99995	70	7	6.99803	95	9	9.00000
21	2	2.00000	46	4	3.99885	71	7	6.99965	96	9	9.00000
22	2	2.00000	47	4	4.00114	72	7	6.99995	97	9	9.00000
23	2	1.99795	48	4	4.00003	73	7	7.00011	98	9	9.00000

24	2	2.00115	49	4	4.00000	74	7	6.99997	99	9	8.99995

In order to compare the recognition result with the result obtained by the proposed algorithm, we construct a basic neural network composed of 16 input nodes, 20 hidden nodes, and 1 output layer node as shown in Fig. 2. Then, by the rule of the neural network learning method, the neural network executes learning steps repeatedly until all the learning characters are completely recognized as the corresponding learning characters. Table 2 shows actual calculation results obtained after executing the repeated learning of 2,000,000 or more times until all the 100 number of learning characters are completely recognized as corresponding learning characters on the neural network with the structure shown in Fig. 2. Since the learning characters have 10 different types from 0 to 9, it can be said that the neural network has been learned perfectly so that 100 number of learning characters are completely recognized as the corresponding characters when rounding operations are performed on the output values.

Table 3. Output of 10 Number of Experimental Characters Obtained by Execution of Neural Network After 2,000,000 or More Number of Repetition of Learning

TD#	Output	Round	O/X
0	0.00000	0	O
1	2.20351	2	X
2	4.68251	5	X
3	3.91755	4	X
4	3.95756	4	O
5	6.09718	6	X
6	5.51196	6	O
7	8.74179	9	X
8	7.98944	8	O
9	7.01381	7	X

Table 3 shows the recognition results obtained when 10 experimental characters are given as inputs on the neural network, where the neural network has learned enough to recognize all 100 number of learning characters as corresponding characters. For these 10 experimental characters, the neural network got 40% recognition result.

Even though the neural network recognizes 100 learning characters completely as the corresponding characters by executing 2,000,000 times or more iterative learning steps, there might exist several reasons why the neural network got the 40% recognition result for the 10 experimental characters with a new font unused for those learning characters. Typical reasons are as follows: 1) The recognition result might be better if multi-layered structures are used for the neural network. 2) When constructing a neural network, the result might vary depending on the number of hidden layer nodes. 3) The weight values used in the neural network are initially given as random values, so the results might be different. In addition, when the neural network is used to recognize characters, it requires long learning time to recognize all learning characters as corresponding characters. As a result, the proposed character classification algorithm obtains better recognition result than a basic neural network.

2.3 The Pros and Cons of the proposed Character Classification Algorithm

Similar to the conventional recognition methods, there are some advantages and disadvantages in the proposed algorithm. Advantages include: 1) It takes less computation time than a neural network, and better recognition result might be obtained. 2) When existing learning character set is modified, additional computation is not required, but neural network should be newly learned. The disadvantage is that the recognition rate might be different if the experimental character has artificially over-deformed fonts.

3. Conclusion and Future work

The proposed character classification algorithm compares the learning characters with the experimental character on a tile basis and uses the variance of the differences to determine the recognition result. We can make the significance of this study that the recognition of characters with new fonts can be performed through this process by using the local characteristics of the characters. In a real experiment, 9 out of 10 experimental characters were classified and recognized correctly by the proposed triangular distribution method, and only one of them was recognized redundantly, and also the processing time was less than that of the conventional neural network circuit. The proposed character classification method might be improved to get better recognition result if the concept of the triangular distribution in 2D space is expanded to that of the tetrahedral distribution in 3D space. As a future research direction, we are studying a voice recognition method that recognizes a voice and converts it into a corresponding word or sentences effectively.

References

[1] L. Jaulin, *Automation for Robotics*, Wiley, pp.47-56, 2015.

[2] S. Annadurai, R. Shanmugalakshmi, *Fundamentals of Digital Image Processing*, Pearson, pp. 103-126, 2007.

[3] M. Kang, "Object Recognition Using the Edge Orientation Histogram and Improved Multi-Layer Neural Network", *International Journal of Advanced Culture Technology(IJACT)*, Vol.6, No.3, pp.142-150, 2018.

[4] D. Sule, *Manufacturing Facilities*, CRC Press, pp.49-95, 2008.

[5] W. Pratt, *Introduction to Digital Image Processing*, CRC Press, pp.139-154, 2013.

[6] F. Shih, *Image Processing and Mathematical Morphology*, CRC Press, pp.25-35, 2009.

[7] Y. Yang, S. Lee, "An Object Tracking Method for Studio Cameras by OpenCV-based Python Program", *The Journal of Convergence on Culture Technology(JCCT)*, Vol.4, No.1, pp.291-297, Feb 2018.

[8] B. Jeong, M. Kang, Y. Jung, "A study on the Facial Expression Recognition using Deep Learning Technique", *International Journal of Advanced Culture Technology(IJACT)*, Vol.6, No.1, pp.60-67, 2018.

[9] D. Rumelhart, J. McClelland, *Parallel Distributed Processing*, MIT Press, pp. 121-127, 1987.

[10] S. Lee, "Deep Structured Learning: Architectures and Applications", *International Journal of Advanced Culture Technology(IJACT)*, Vol.6, No.4, pp.262-265, 2018.

[11] R. Kautz, *Chaos – The Science of Predictable Random Motion*, Oxford University Press, pp. 19-42, 2011.

[12] G. Dougherty, *Pattern Recognition and Classification*, Springer, pp. 123-134, 2012.

[13] S. Yoo, "Character Recognition Algorithm using Accumulation Mask", *International Journal of Advanced Culture Technology(IJACT)*, Vol.6, No.2, pp.123-128, 2018.