Thesis for Master's Degree

2013

Sign Language Recognition using a Modified FMM Neural Network

SEUNG-KANG LEE

Department of Information Technology
Handong Global University

Sign Language Recognition using a Modified FMM Neural Network

수정된 FMM 신경망을 이용한 수화 인식 기법

Sign Language Recognition using a Modified FMM Neural Network

Advisor : Professor Ho-Joon Kim

By

Seung-Kang Lee

Department of Information Technology

Handong Global University

A thesis submitted to faculty of the Handong University in partial fulfillment of the requirements for the degree of Master of Engineering in the Department of Information Technology

November 29, 2013

Approved by

Professor Ho-Joon Kim

Thesis Advisor

Sign Language Recognition using a Modified FMM Neural Network

Seung-Kang Lee

Accepted in partial fulfillment of the requirements for the degree of Master of Engineering

November 29, 2013

Thesis Advisor ______________________

Prof. Ho-Joon Kim

Committee Member ______________________

Prof. Keum-Young Sung

Committee Member ______________________

Prof. Whan-Ki Yong

IT Seung-Kang Lee 이승강
21231002 Sign Language Recognition using a Modified FMM Neural Network. 수정된 FMM 신경망을 이용한 수화 인식 기법. 한동대학교 일반대학원 정보통신공학과 석사논문. Department of Information Technology, 2013 44p. Advisor : Prof. Ho-Joon Kim

Abstract

The study proposes a recognition system model for sign language. To extract features for pattern classification, motion history volume(MHV) was used as an input data. Three-dimensional extended convolutional neural network is used for feature extraction. For pattern classification, modified fuzzy min-max(FMM) neural network was proposed and used. Overlapping problem of original FMM model was improved adapting frequency factor concept. Modified FMM neural network was trained, and relevance factors were calculated using weight values of neural network to build rules between features and pattern classes. To evaluate proposed model, Fisher's Iris data set was used as an input sample. The experimental results show the performance proposed model comparing other neural network models. And an experiment applying proposed model to a sign language recognition is performed. We analyzed its performance by evaluating its error rate and recognition rate.

본 연구에서는 수정된 FMM(Fuzzy Min-Max) 신경망을 이용한 수화 인식 시스템을 제안한다. 수화 영상의 특징을 추출하기 위해 모션 히스토리 볼륨(MHV)이라는 3차원 데이터를 사용한다. 모션 히스토리 볼륨을 입력값으로 3차원으로 확장된 CNN(Convolutional Neural Network)구조를 이용하여 수화 영상에서의 변이를 수용하는 특징 추출을 적용한다.

본 연구에서는 수화 패턴 분류를 위해서 FMM 신경망을 적용하였다. 기존의 FMM의 하이퍼박스 중첩 문제를 해결하기 위하여 패턴의 빈도 요소를 고려한 수정된 형태의 FMM을 제안한다. 추출된 특징 지도를 이용해 수정된 FMM 신경망을 학습하고, 수화 패턴 클래스와 특징 지도 간의 빈도 요소가 적용된 가중치를 이용하여 패턴과 특징 사이의 연관도 요소를 산출하였다. 산출된 연관도 요소를 이용하여 IF-THEN 형태의 규칙을 생성하였다.

본 연구에서 제안한 수정된 FMM 신경망의 성능을 평가하기 위해 Fisher의 Iris data를 이용하여 분류 실험을 수행하고 기존의 FMM 신경망과 성능을 비교하였다. 그리고 제안된 수화 인식 시스템을 평가하기 위해 수화 영상 데이터를 사용하여 수화 인식 실험을 수행하였고, 실험 결과로부터 제안된 방법의 타당성을 고찰하였다.

CONTENTS

List of Figures

List of Tables

Ⅰ. Introduction

1. Research Objective

A pattern recognition refers to classification of set of data into several groups based on the input value and certain criteria. In artificial intelligence area, many researches related to pattern recognition are developed, influencing human life significantly. Representative examples of product using pattern recognition technique include speech recognition, handwriting recognition and face recognition mounted on smart cellular phone.

Many automatic sign language analysis systems have used tracker-based supporting equipments or wearable trackers for an effective feature extraction and recognition. It is because sign language data contain various variations which have complex formation. The system using additional devices, however, has a severe constraint of the environment and conditions.

In this research, sign language recognition system is presented without an additional device such as a motion sensor and but use sign language video image data as input of pattern recognition system. The proposed model of the research has three essential parts. The first part is preprocessing part for preparing a feature extraction. The second part, which is the feature extraction part, extended convolutional neural network(CNN) is used as a feature extractor with three-dimensional receptive field. Motion history volume is an input of feature extractor and convoluted by CNN. The third part consists of learning and classifying with extracted features as input. We suggest modified fuzzy min-max(FMM) neural network as classification model. For pattern classification, we propose a method for rule extraction using a frequency factor.

2. Related Works

Artificial neural networks have been well known technique for pattern recognition and classification. Generally, multi-layer perceptron(MLP) model is a famous and preferred classifier because of its ability of solving complex non-linear boundary classification problems. However, its learning algorithm takes very long time to classification because it take a great number of passes for the training data to reach the convergence state. Moreover, if new pattern is added to classifier, it has to do retraining job of the neural network, regardless of whether it is a new training data or old data even it was trained before or not.

Fuzzy min-max neural networks are introduced by Simpson[1]. They are built using hyperbox fuzzy sets that define regions of the n-dimensional pattern space. A membership function of hyperbox completely defined by its minimum point and maximum point. Recently, many researches use the FMM neural network to build pattern classification and recognition system.

By applying the theory of Simpson's FMM, Garbrys introduce a general FMM neural network for clustering and classification[2]. And Quteishat suggests a modified FMM neural network with rule extraction and its application to fault detection and classification[3] and a genetic-algorithm-based rule extractor[4].

The FMM neural network has a simple and powerful learning algorithm. Without spoiling old class information, new classes can be learned and existing classes can be refined quickly, and short training time is required even including non-linear separability.

To extract features in sign language recognition, the study used extended convolutional neural network with 3-dimensional receptive field. Convolutional neural network is suitable for image pattern recognition which have a lot of variation and distortion. Simard shows neural networks achievement on handwriting recognition task using CNN[6]. And Sawrence present a face recognition research with a convolutional neural network approach[5].

Ⅱ. Underlying System

1. System Structure for Sign Language Recognition

The underlying system consists of multi-layer structure for sing language recognition. As shown in Figure 1, there are three modules: preprocessing, feature extraction and pattern classification. In detail, two sub steps exists in each module.

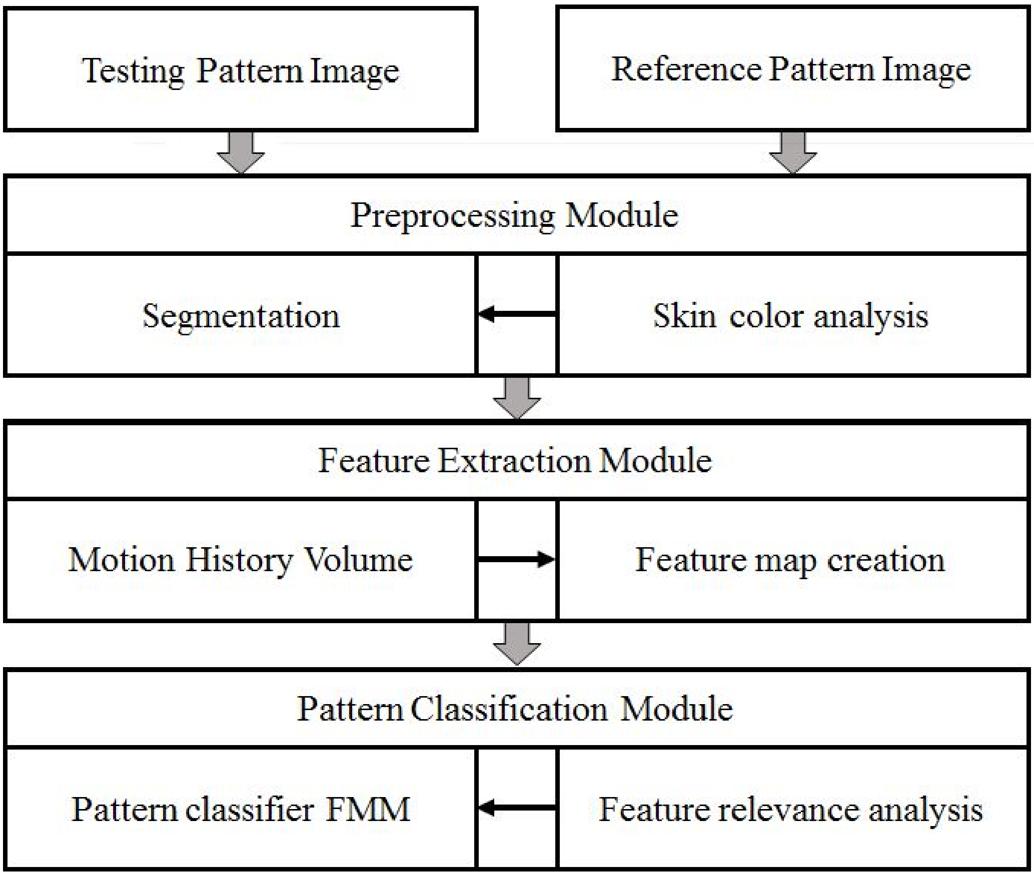


Figure 1. The underlying sign language recognition system structure

Input pattern data can be divided into two types. One is reference patterns for training and learning the pattern classifier and the other is test patterns for performance evaluation and experiments.

In the preprocessing module, feature region of image are detected through skin color analysis and segmentation. Then the feature extraction process is proceeded. Using preprocessed image data, wes build a 3-dimensional data : motion history volume(MHV). To build a feature map, convolutional neural network(CNN) is used for a feature extraction. MHV generated in the previous step is used as an input of CNN. It is possible that processing the module not only can reduce the number of features significantly but also can be robust to spatio-temporal variation. Moreover, with feature values that has been extracted through CNN, fuzzy min-max(FMM) neural network classifier is trained. In learning process of the FMM classifier, it is feasible to calculate relevance factor, which can used as frequency analysis.

Ⅲ. Feature Extraction using Extended CNN

1. Motion History Volume

To extract features from sign language video data, motion history volume(MHV) was built and used. The MHV is a 3-dimensional volume data that represents a movement of motion information over time sequence. In the preprocessing module, skin color detection and segmentation is processed from input sign language video data to create motion history volume(MHV). Executing preprocess module stage to each frame from sign language video data, the MHV is created by extending its motion variation over time domain.

Figure 2. is an examples of MHV data built using korean sign language video data ‘regret’. x and y axes represent spatial variance, and t axis represents the motion variation of the time domain. The dim gray color means early time, and darker color is late frame data. And Figure 3. is the other MHVs of sign language video data. As shown, the location of motion and the length of the time axis can be different among the sign language data.

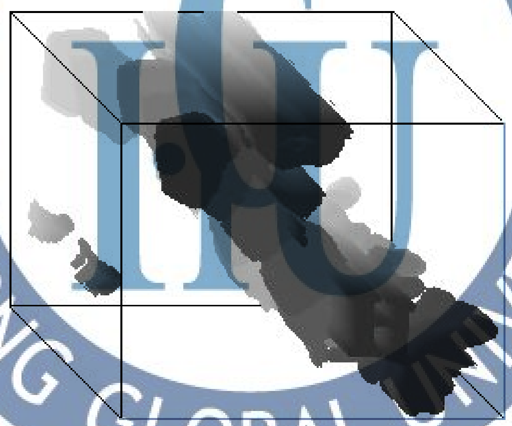


Figure 2. An example of MHV of sing language ‘regret’

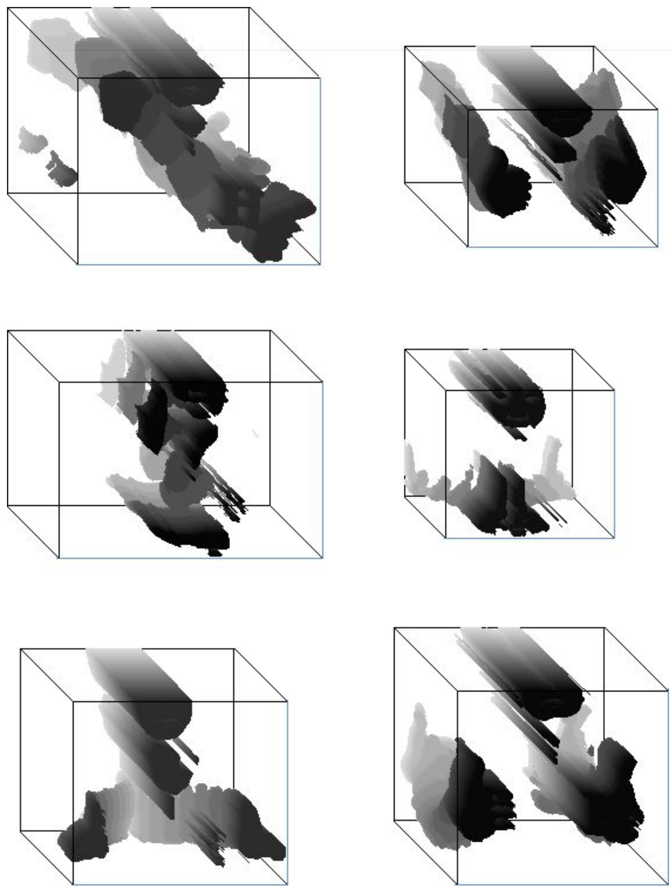


Figure 3. Examples of motion history volume (MHV)

2. Extended Convolutional Neural Network

2.1 Convolutional Neural Network

Convolutional neural network(CNN) are often used in image and gesture recognition system[5-6]. Convolution is a mathematical operation on two functions. It produces the third function that is typically viewed as a modified version of one of the original functions, giving the area overlap between the two functions as a function of the amount that one of the original functions is translated. In digital image processing, convolutional filtering plays an important role in many important algorithms. Convolutional neural networks(CNN) incorporate constraints and achieve some degree of shift and deformation invariance using spatial subsampling and local receptive fields[5]. CNN consisits of two layers, convolution layer and subsampling layer. The convolution layers have orientation-selective filter banks where elementary visual features are extracted from the spatial template. The filtered image is then subsampled by the subsampling layer.

2.2 Extended Convolutional Neural Network

In this study, extended version of the CNN is used for feature extraction as shown in Figure 4.

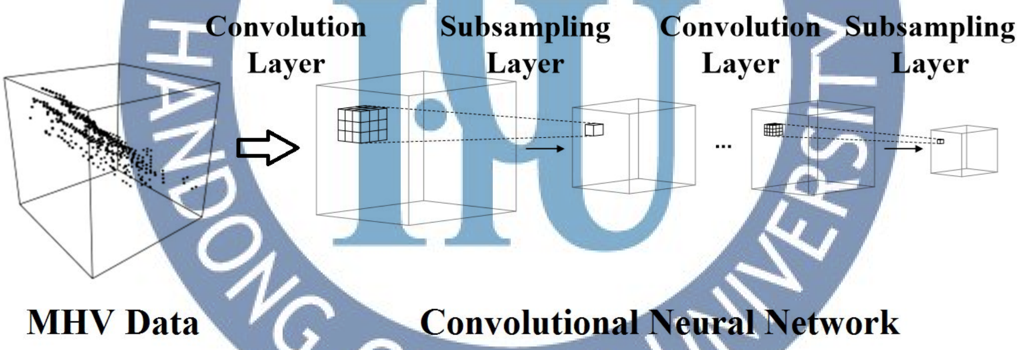


Figure 4. The extended CNN model for feature extraction

The input data for the feature extraction are the MHV data which is introduced in the previous section. The structure of receptive field is extended along the time axis. The center of the 3-dimensional processing element shifts through the spatial and temporal domain of the MHV. Feature value C_i^l of the ith position of layer level l represented as equation (3.1).

$$C_i^l = g(N_{i,l} \bigcirc W_l + B_{i,l}) \tag{3.1}$$

In the equation (3.1), $N_{i,l}$ means the 3-dimensional neighborhood area of the ith position of layer level l. And $B_{i,l}$ means a bias value and $\bigcirc$ is a 3-dimensional convolution operation. Function g is a sigmoid function in the form of $\tanh(x)$.

The extended model of CNN is not only robust to spatial variation but also to temporal variation. A final feature map size for our system is (3 X 3 X 3), 27 extracted features as show in Figure 5. This feature map becomes the input of the patten classifier.

Figure 5. An example of the spatio-temporal volume and the feature map

After extraction, 27 features are generated. 27 features extracted from MHV are named as $M = \{m1, m2, \cdots, m27\}$. We name each feature from its position and time sequence. Figure 6 represent the position of features and its symbolic names.

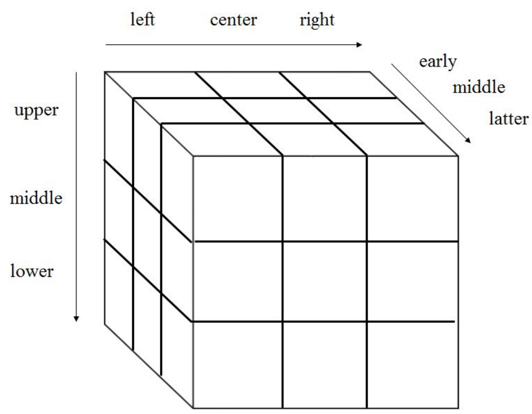


Figure 6. Feature names from its position and time sequence

Table 1. represents the symbolic character of features and its name.

Symbolic Character	x axis	y axis	t axis	name
m1	left	upper	early	left-upper-early
m2	center	upper	early	center-upper-early
m3	right	upper	early	right-upper-early
m4	left	middle	early	left-middle-early
m5	center	middle	early	center-middle-early
m6	right	middle	early	right-middle-early
m7	left	lower	early	left-lower-early
m8	center	lower	early	center-lower-early
m9	right	lower	early	right-lower-early
m10	left	upper	middle	left-upper-middle
m11	center	upper	middle	center-upper-middle
m12	right	upper	middle	right-upper-middle
m13	left	middle	middle	left-middle-middle
m14	center	middle	middle	center-middle-middle
m15	right	middle	middle	right-middle-middle
m16	left	lower	middle	left-lower-middle
m17	center	lower	middle	center-lower-middle
m18	right	lower	middle	right-lower-middle
m19	left	upper	latter	left-upper-latter
m20	center	upper	latter	center-upper-latter
m21	right	upper	latter	right-upper-latter
m22	left	middle	latter	left-middle-latter
m23	center	middle	latter	center-middle-latter
m24	right	middle	latter	right-middle-latter
m25	left	lower	latter	left-lower-latter
m26	center	lower	latter	center-lower-latter
m27	right	lower	latter	right-lower-latter

Table 1. Representation of symbolic characters and names of each feature

Ⅳ. Classification Model : Modified FMM Neural Network

1. Fuzzy Min-Max Neural Network

1.1 Overview

Fuzzy set means a set of data that were not precise, but rather fuzzy. Fuzzy min-max(FMM) neural network was proposed by Simpson[1]. It is supervised-learning classification model and has simple and powerful learning algorithm. Fuzzy min-max classification neural networks are built using hyperbox fuzzy sets. The fuzzy theory measures the degree of event occurring. Hyperboxes are defined by its pairs of min-max points and their membership functions that used to create fuzzy subsets of the n-dimensional space. Figure 7. is the implementation of fuzzy min-max neural network classifier. F_a is input node for classifier. Each extracted feature values of n number is an input for nodes. And F_b is hyperbox nodes. There are two sets of connections between F_a and the jth F_b node : the minimum points vector V_j and the maximum points vector W_j. The transfer function of F_b is a hyperbox membership function. Each F_c node means each pattern class. The connections between F_b nodes and the p output nodes F_c are binary valued and are determined as each F_b node is added during learning.

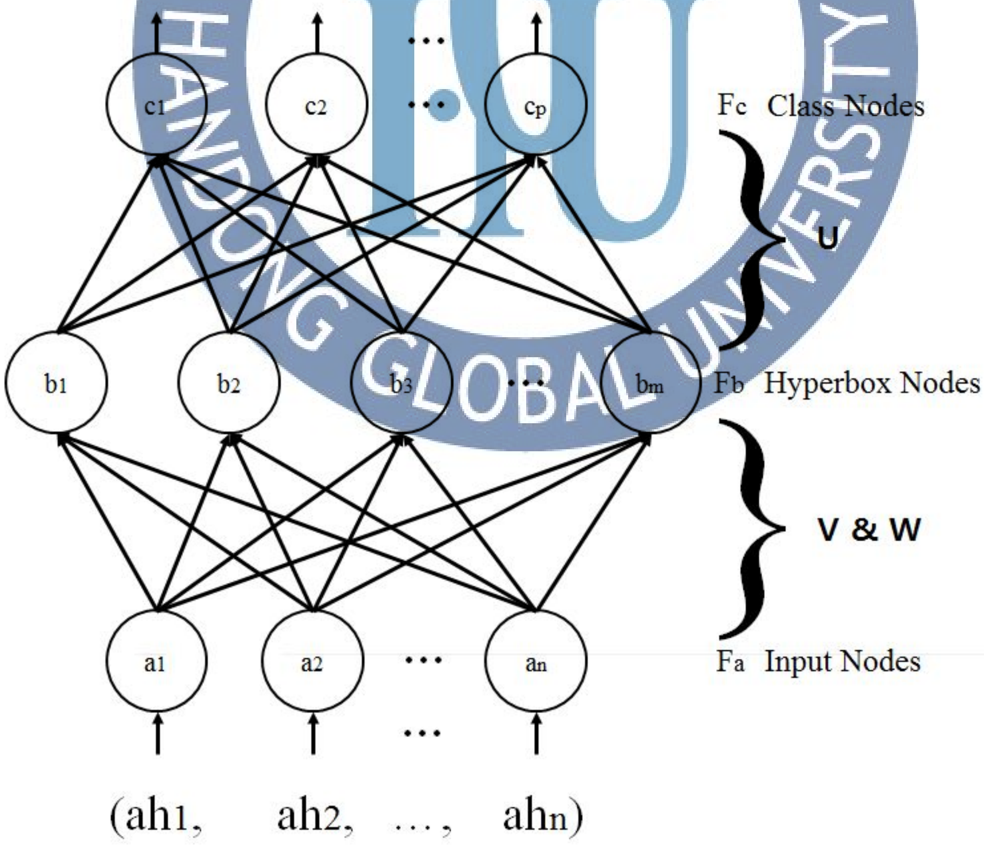


Figure 7. The three-layer structure of fuzzy min-max neural network

Let each hyperbox fuzzy set B_j be defined by the ordered set

$$B_j = \{X, V_j, W_j, f(X, V_j, W_j)\} \qquad \forall X \in I^n \tag{4.1}$$

Using this definition of a hyperbox fuzzy set, the aggregate fuzzy set that defines the kth pattern class C_k is define as equation (4.2).

$$C_k = \bigcup_{j \in K} B_j \tag{4.2}$$

where K is the index set of those hyperboxes associated with class k. Note that the union operation in fuzzy sets is typically the max of all of the associated fuzzy set membership function.

The membership function for the jth hyperbox $b_j(A_h)$ must measure the degree to which the hth input pattern A_h falls outside of the hyperbox B_j. The resulting membership function is defined as equation (4.3).

$$b_j(A_h) = \frac{1}{2n}\sum_{i=1}^{n}[\max(0, 1-\max(0, \gamma\min(1, a_{hi}-w_{ji}))) + \max(0, 1-\max(0, \gamma\min(1, v_{ji}-a_{hi})))] \tag{4.3}$$

1.2 Learning Algorithm

Fuzzy min-max learning is an expansion/contraction process[1]. The fuzzy min-max classification learning algorithm has a three-step process.

1) Expansion : Identify the hyperbox that can be expanded and expand it. If an expendable hyperbox cannot be found, add a new hyperbox for that class.

2) Overlap test : Determine if any overlap exists between hyperboxes forming different class.

3) Contraction : If overlap between hyperboxes that represent different classes does exist, eliminate the overlap by minimally adjusting each of the hyperboxes.

1.2.1 Hyperbox Expansion

The maximum size of a hyperbox is bounded above by $0 \le \theta \le 1$, a user defined value. For the hyperbox B_j, to be expanded to include X_h, the following constraint must be met :

$$n\theta \ge \sum_{i=1}^{n} (\max(w_{ji}, x_{hi}) - \min(v_{ji}, x_{hi})) \tag{4.4}$$

If the expansion criterion has been met for hyperbox B_j, the min point of the hyperbox is adjusted using the equation (4.5).

$$v_{ji}^{new} = \min(v_{ji}^{old}, x_{hi}) \qquad \forall i = 1, 2, \cdots, n \tag{4.5}$$

and the max point is adjusted using the equation (4.6)

$$w_{ji}^{new} = \max(w_{ji}^{old}, x_{hi}) \qquad \forall i = 1, 2, \cdots, n \tag{4.6}$$

1.2.2 Hyperbox Overlap Test

Overlapping hyperboxes that represent the same class do not present a class formation problem. But it is necessary to eliminate overlap between hyperboxes that represent different classes.

To determine if this expansion created any overlap, a dimension by dimension comparison between hyperboxes is performed.

Assuming $\delta^{old} = 1$ initially, the four test cases and the corresponding minimum overlap value for the ith dimension are as follows

Case 1 : $v_{ji} < v_{ki} < w_{ji} < w_{ki},\ \delta^{new} = \min(w_{ji} - v_{ki}, \delta^{old})$

Case 2 : $v_{ki} < v_{ji} < w_{ki} < w_{ji},\ \delta^{new} = \min(w_{ki} - v_{ji}, \delta^{old})$

Case 3 : $v_{ji} < v_{ki} < w_{ki} < w_{ji},\ \delta^{new} = \min(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old})$

Case 4 : $v_{ki} < v_{ji} < w_{ji} < w_{ki},\ \delta^{new} = \min(\min(w_{ji} - v_{ki}, w_{ki} - v_{ji}), \delta^{old})$

1.2.3 Hyperbox Contraction

If the result of overlapping test represent overlap, we have to proceed hyperbox contraction step. To determine the proper adjustment to make, the same four cases are examined.

Case 1 : $v_{j\Delta} < v_{k\Delta} < w_{j\Delta} < w_{k\Delta}$,
$$w_{j\Delta}^{new} = v_{k\Delta}^{new} = \frac{w_{j\Delta}^{old} + v_{k\Delta}^{old}}{2}$$

Case 2 : $v_{k\Delta} < v_{j\Delta} < w_{k\Delta} < w_{j\Delta}$,
$$w_{k\Delta}^{new} = v_{j\Delta}^{new} = \frac{w_{k\Delta}^{old} + v_{j\Delta}^{old}}{2}$$

Case 3a: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$,
$$v_{j\Delta}^{new} = w_{k\Delta}^{old}$$

Case 3b: $v_{j\Delta} < v_{k\Delta} < w_{k\Delta} < w_{j\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$,
$$w_{j\Delta}^{new} = v_{k\Delta}^{old}$$

Case 4a: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) < (w_{j\Delta} - v_{k\Delta})$,
$$w_{k\Delta}^{new} = v_{j\Delta}^{old}$$

Case 4b: $v_{k\Delta} < v_{j\Delta} < w_{j\Delta} < w_{k\Delta}$ and $(w_{k\Delta} - v_{j\Delta}) > (w_{j\Delta} - v_{k\Delta})$,
$$v_{k\Delta}^{new} = w_{j\Delta}^{old}$$

2. Other Researches using Fuzzy Min-Max Neural Network

2.1 General Fuzzy Min-Max Neural Network

Bogdan Gabrys and Andrzej Bargiela proposed a general fuzzy min-max(GFMM) neural network in 2000, which is a generalization and extension of the fuzzy min-max clustering and classification algorithms developed by Simpson[2]. The GFMM method combines the supervised and unsupervised learning within a single training algorithm. The characteristics of the GFMM method can be explained in the following points.

1) Input patterns can be fuzzy hyperboxes in the pattern space, or crisp-points in the pattern space.

2) The fuzzy hyperbox membership function and basic hyperbox expansion constraint proposed in original model have been modified.

3) The labeled and unlabeled input patterns can be processed at the same time which resulted in an algorithm that can be implemented as pure clustering, pure classification, or hybrid clustering/classification system.

4) The parameter regulating the maximum hyperbox size can be changed adaptively in the course of GFMM neural network training.

2.1.1 GFMM Algorithm

(1) Basic Definitions

The input is specified as the ordered pair $\{X_h, d_h\}$ where $X_h = [X_h^l \; X_h^u]$ is the hth input pattern in lower bound, X_h^l, and upper bound, X_h^u, in the n-dimensional feature space I^n. And is d_h is the index of pattern classes, where $d_h = 0$ means that the input vector is unlabeled.

A new membership function of GFMM has been defined as

$$b_j(X_h) = \min_{i=1\cdots n}\left(\min\left(\left[1-f\left(x_{hi}^u - w_{ji}, \gamma_i\right)\right], \left[1-f\left(v_{ji} - x_{hi}^l, \gamma_i\right)\right]\right)\right) \tag{4.7}$$

where $f(r,\gamma) = \begin{cases} 1 & \text{if } r\gamma > 1 \\ r\gamma & \text{if } 0 \le r\gamma \le 1 \\ 0 & \text{if } r\gamma < 0 \end{cases}$. γ is a sensitivity parameter regulating how fast membership value decrease.

(2) GFMM Learning Algorithm

When a new hyperbox created, its min points, V_j, and max points, W_j, are initialized and adjusted by the creation and expansion process. The V_j and W_j are initially set to 0. In expansion stage, when the hth imput pattern X_h is presented, the hyperbox B_j with the highest degree of membership is found and allows expansion. The hyperbox B_j can expand to include the input X_h as following two steps:

$$\forall_{i=1\cdots n}\left(\max\left(w_{ji}, x_{hi}^u\right) - \min\left(v_{ji}, x_{hi}^l\right)\right) \le \Theta$$

and

$$\begin{array}{l} if\ d_h = 0\ then\ adjust\ B_j \\ else\ if\ class(B_j) = \begin{cases} 0 & \Rightarrow & \begin{array}{c} adjust\ B_j \\ class(B_j) = d_h \end{array} \\ d_h & \Rightarrow & adjust\ B_j \\ else & \Rightarrow & \begin{array}{c} take \\ another\ B_j \end{array} \end{cases} \end{array}$$

with the $adjust\ B_j$ operation defined as

$$v_{ji}^{new} = \min\left(v_{ji}^{old}, x_{ji}^{l}\right), \text{ for } each\ i = 1, \cdots, n$$
$$w_{ji}^{\neq w} = \max\left(w_{ji}^{old}, x_{hi}^{u}\right), \text{ for } each\ i = 1, \cdots, n.$$

If neither including process nor expanding process, then a new hyperbox is created, adjusted, and labeled by setting $class(B_k) = d_h$.

The other expansion process constraint result from admitting both labeled and unlabeled input patterns. If the input pattern X_h is not labeled, then the hyperbox B_j can be adjusted to include this pattern.

Because of applying of labeled and unlabeled input patterns, GFMM hyperbox overlapping test must consider not only different classes but also of all hyperboxes that are not labeled.

Assuming that hyperbox B_j was expanded, test for overlapping with the hyperbox B_k if

$$class(B_j) = \begin{cases} 0 \Rightarrow & \text{test for overlapping with all the other hyperboxes} \\ else \Rightarrow & \text{test for overlapping only if } class(B_j) \neq class(B_k) \end{cases}$$

The four cases are being considered where initially $\delta^{old} = 1$

Case 1: $v_{ji} < v_{ki} < w_{ji} < w_{ki}$
$$\delta^{new} = \min\left(w_{ji} - v_{ki}, \delta^{old}\right)$$

Case 2: $v_{ki} < v_{ji} < w_{ki} < w_{ji}$
$$\delta^{new} = \min\left(w_{ki} - v_{ji}, \delta^{old}\right)$$

Case 3: $v_{ji} < v_{ki} \le w_{ki} < w_{ji}$
$$\delta^{new} = \min\left(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old}\right)$$

Case 4: $v_{ki} < v_{ji} \le w_{ji} < w_{ki}$
$$\delta^{new} = \min\left(\min(w_{ki} - v_{ji}, w_{ji} - v_{ki}), \delta^{old}\right)$$

If overlap for the ith dimension has been occurred and $\delta^{old} - \delta^{new} > 0$, then $\Delta = i$, $\delta^{old} = \delta^{new}$ and $case = l(l = \{1,2,3,4\})$: the case for which the smallest overlap was found. If overlap for the ith dimension has not been occurred, set $\Delta = -1$ signifying that the contraction step is not necessary. The contraction process is the same as in min-max classification algorithm.

2.2 A Fuzzy Min-Max Neural Network with a Genetic-Algorithm-Based Rule Extractor

Anas Quteishat proposed a two-stage pattern classification and rule extraction system based on a modified fuzzy min-max network and a genetic algorithm rule extractor[4]. This model has two parts which compose pattern classification and rule extraction. In the first stage, a modified FMM model is used to generate hyperboxes[3]. By pruning process, some hyperboxes are cut off based on its calculated confidence factor. In the second stage, “open hyperboxes” are created from the original hyperboxes and the genetic algorithm is used to minimize the input features in the rules and to maximize classification accuracy[4].

2.2.1 Two-Stage Pattern Classification and Rule Extraction

By pruning process, some hyperboxes are cut off based on its calculated confidence factor. In the second stage, “open hyperboxes” are created from the original hyperboxes and the genetic algorithm is used to minimize the input features in the rules and to maximize classification accuracy[4].

(1) The Modified FMM Pattern Classifier

In this stage, the original FMM network is modified by introducing a pruning procedure after its learning phase[4]. The original FMM has two data set, training and testing. The model has the input-output data set divided into three subsets: training, prediction, and testing.

The learning process is the same as original FMM, and a pruning procedure is incorporated into modified FMM after its learning step. The pruning method is processed by calculated confidence factor(CF_j) for each hyperbox B_j in terms of its usage frequency and predictive accuracy on a prediction data set.

$$CF_j = (1-\gamma)U_j + \gamma A_j \tag{4.8}$$

where U_j and A_j are the usage and accuracy indices of hyperbox B_j and $\gamma \in [0, 1]$ is a weighing factor. The usage of a hyperbox is the number of prediction set patterns classified by any hyperbox B_j divided by the maximum number of prediction patterns of any hyperbox with the same classification class. And the accuracy index of a hyperbox is the number of correctly predicted set of patterns divided by the maximum correctly classified patterns with the same classification class. Hyperboxes with a confidence factor smaller than a predefined threshold are pruned.

(2) Genetic Algorithm Based Rule Extractor

After pruning process, the remaining hyperboxes are used to generate open hyperboxes. All hyperboxes are set to the genetic algorithm for evolution.

A hyperbox which is defined by its minimum and maximum points is called closed hyperbox. If a hyperbox has dimensions that are not defined by its min-max points. the hyperbox is called an open hyperbox and the non-declared dimension is regarded as the don't care dimension.

A genetic algorithm is used to evolve and choose a set of hyperboxes that has a good accuracy rate with a small number of features. The chromosome of the algorithm is a binary string that represents a solution comprising all the possible open hyperboxes as follows:

$$S = \{D_1^1, D_2^1, \cdots, D_d^1, D_1^2, D_2^2, \cdots, D_d^2, \cdots, D_1^p, D_2^p, \cdots, D_d^p\} \quad (4.9)$$

A value of one in S indicates that the dimension is a closed. On the other hand, a value of zero for the allele in S indicates that the dimension is a don't care dimension and its membership value is set to one. the length of S is $p \times d$, and the genetic algorithm search space is $p(2^d - 2)$. In FMM-GA, the length of S and the genetic algorithm search space are subject to the number of hyperboxes, which is controlled by pruning. The curse-of-dimensionality problem in this model can be mitigated by pruning.

For rule extraction, each hyperbox is transformed into on fuzzy rule[4]. The procedure starts by quantifying first the min-max values of each input feature. The quantization level Q equals the number of fuzzy partitions in the quantized rule. For example, if $Q = 3$, then feature A_q is quantized to three level such as low, medium, or high in a fuzzy rule. Interval $[0,1]$ is divided into Q intervals, and the feature is assigned to the quantization points evenly.

$$A_q = \frac{q-1}{Q-1} \quad (4.10)$$

where $q = 1, \cdots, Q$. The if-then rules extracted in the following format:

Rule R_j : IF x_{p1} is A_q and ... x_{pn} is A_q , THEN x_p is class C_j with $CF = CF_j, j = 1, 2, \cdots, N$

where N is the number of hyperboxes, $x_p = (x_{p1}, \cdots, x_{pn})$ is an n-dimensional pattern vector, A_q is the antecedent value, and CF_j is the confidence factor for the jth hyperbox.

3. Proposed Model : A Modified Fuzzy Min-Max Neural Network

3.1 Overview

Original FMM neural network has an advantage of powerful learning algorithm and easy to implement. In learning algorithm, however, there could be a distortion of leaning process. Assume that a sample of pattern which has a large amount of error becomes an input of well-trained FMM pattern classifier. By performing its hyperbox expansion operation, a well trained hyperbox could be distorted even with a single sample's error.

To overcome this drawback, the study propose a modified FMM neural network model with a frequency factor. Definition of modified hyperbox membership function for pattern classifier is as follows.

$$B_j(A_h) = \frac{1}{Z}\sum_{i=1}^{n} w_{ji} f(a_{hi}, I_{ji}) \qquad (4.11)$$

In equation (4.7), $A_h = (a_{h1}, a_{h2}, \cdots, a_{hn}) \in I^n$ is the hth input pattern which consists of n features. I_{ji} is the feature value ranges of the ith dimension in the jth hyperbox. w_{ji} means the connection weight between ith feature and the jth hyperbox. The function f is defined following equation (4.12).

$$f(a_{hi}, I_{ji}) = \begin{cases} 1.0 & if\ I_{ji}^L < a_{hi} < I_{ji}^U \\ 1.0 + \gamma(a_{hi} - I_{ji}^L) & if\ a_{hi} < I_{ji}^L \\ 1.0 - \gamma(a_{hi} - I_{ji}^U) & if\ a_{hi} > I_{ji}^U \end{cases} \tag{4.12}$$

The parameter γ controls the slope of the fuzzy membership function at the boundaries of the feature range.

3.2 Learning Process

Unlike original FMM neural network model, our proposed model doesn't need a overlapping test and contraction process because of frequency feature concept.

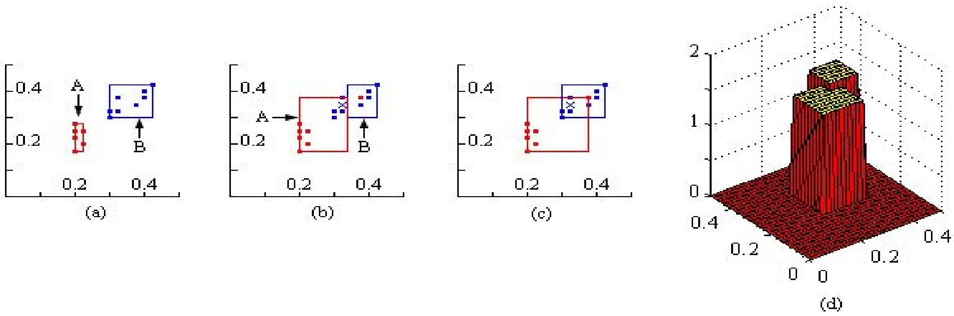


Figure 8. Comparison of the two decision making of different FMM model

In the Figure 8, (a) is original state of classifier, (b) is after contraction process of original FMM model and (c) is operation of modified FMM model with frequency concept. Red dots are class A and blue dots are class B. As an input of (0.4, 0.4), class A is entered in this classifier, hyperbox overlapping is occurred and then contraction is performed. In this case, if an input pattern marked × is near the data points of class B, the original classifier will determine this as of class A, while it would seem more appropriate to classify it as class B. Futhermore, half of the input training patterns that were subject to the hyperbox of class B are being excluded after the hyperbox contraction occurs. Although an overlap exists between the two hyperboxes, the disambiguation can be achieved in the proposed model by comparing the hyperbox gain as shown in (d).

The learning of modified FMM neural network model consists of two processes : hyperbox creation and hyperbox expansion. In the learning process, the weight updating equation is define as follow (4.13).

$$w_{ji}^{new} = \begin{cases} w_{ji}^{old} + \lambda & if\,(|I_{ji}|^{old} = |I_{ji}|^{new}) \\ \dfrac{w_{ji}^{old} \cdot |I_{ji}|^{old}}{|I_{ji}|^{new}} & otherwise \end{cases} \tag{4.13}$$

$|I_{ji}|$ means the size of ith feature range of the jth hyperbox. In other words, each weight value increases in proportionals to the frequency factor of the extracted feature value and inverse proportional to the size of the feature range. The weights between the features and the hyperboxes represent the frequency concept, so the hyperbox overlapping problem can be improved without contraction process.

We have defined the feature ranges of each dimension as 5 fuzzy partitions as shown in Figure 9.

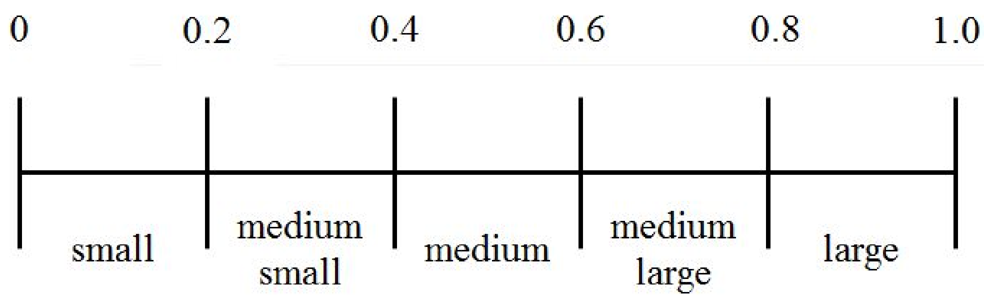


Figure 9. Fuzzy partitions for feature ranges

3.3 Rule Extraction Method

After training the modified FMM neural network classifier, relevance factor between the features and each classes can be calculated using its weights between the feature nodes and the hyperbox nodes. Relevance factor function RF equation between the feature range I and the pattern class k is defined as shown in the equation (4.14).

$$RF(I_{ji},k)=\left(\frac{1}{N_k}\sum_{B_j\in C_k} w_{ji}-\frac{1}{N_B-N_k}\sum_{B_j\notin C_k} w_{ji}\right)/\sum_{B_j\in C_k} wji \tag{4.14}$$

In equation (4.10), C_k means a set of hyperboxes belongs to class k. N_B is the number of all hyperboxes and N_k is the number of hyperboxes that belongs to class k.

If the relevance factor $RF(I_{ji,}k)$ has a positive value, it represents that the feature I_{ji} has an excitatory characteristic for the pattern class k. In contrast, if the $RF(I_{ji,}k)$ has a negative value, it means that feature I_{ji} has an inhibitory relationship for the pattern class k.

Using this analysis of $RF(I_{ji,}k)$, set of rules can be built for the pattern recognition. Association degree between the features and our goal patten classes can be analyzed. Threshold value for relevance factor to reduce a feature dimension for improving performance of pattern classifier can be meas as well.

Another step is finding the $IF-THEN$ rules by analyzing the relevance factor. If the relevance factor $RF(I_{ji,}k)$ has a negative value n, and $I_{ji}=0.15$, the following rule can be built.

IF (x_i is SMALL) $THEN$ not pattern k with $(cf = |n|)$

where cf means 'confidence factor' that is defined as the absolute value of the relevance factor value.

In our sign recognition system, 27 features extracted from MHV are named as $M = \{m1, m2, \cdots, m27\}$. Each feature has the meaning related with its location in the MHV, so we can build a symbolic rules like the following rule. If the relevance factor $RF(I_{ij}, k)$ value 0.235 and the feature m4 means 'left-middle-early', I_{ji} = 0.45 and pattern class k is sign language 'depart', then rules like the following example can be built.

IF (left-middle-early motion MEDIUM)

$THEN$ the pattern is 'depart' with $(cf = 0.235)$

Ⅴ. Experimental Results

1. Performance of Pattern Classifier on Sign Language Recognition

The first experiment is to evaluate the classification performance of the proposed modified FMM neural network. Two neural network pattern classifiers were implemented, original FMM neural network and modified FMM neural network to evaluate their performance comparison experiment. In order to compare these models, the performance error is defined as following equation (5.1).

$$E = \frac{1}{pm} \sum_{k=1}^{p} \sum_{l=1}^{m} |a_{kl} - d_{kl}| \tag{5.1}$$

Where p is the number of test patterns, and m is the number of pattern classes. a_{kl} is the actual output of test and d_{kl} is the desired output value of class l for the kth input pattern.

As a training samples and a testing samples, Fisher's Iris data set is used. Fisher's Iris data set consists of 150 pattern cases in three classes and each pattern case has four features, sepal length, sepal width, petal length, and petal width. For training neural networks, 30, 60, 90, 120, and 150 patterns are used for each experiment step. For testing pattern classifiers, all iris pattern data (150) are used. Table 2 shows the result of experiment.

# of training patterns	Original FMM		Modified FMM	
	# of error patterns	Error rate(%)	# of error patterns	Error rate (%)
30	8	5.334	4	2.667
60	5	3.334	3	2.0
90	5	3.334	3	2.0
120	4	2.667	3	2.0
150	4	2.667	2	1.334

Table 2. Comparison of performance according to the number of training pattern using Fisher's Iris data

As shown in Table 2, reduction of classification performance by training distorted pattern data was improved in the modified FMM neural network.

In order to investigate what is improved more deeply, we analyzed and compared the error rate of each pattern classes in the experiment of 30 number of training patterns. Analysis revealed that most of the error were P2, respectively. Many of wrong results of classification of P2 are classified as the P3. As shown in Table 3, by improving hyperbox overlapping, the error rate of P2 decrease using proposed modified FMM neural network.

# of training patterns	Original FMM		Modified FMM	
	# of error patterns	Error rate(%)	# of error patterns	Error rate(%)
P1	1	2.0	1	2.0
P2	6	12.0	3	6.0
P3	1	2.0	1	2.0

Table 3. Error rate comparison data for each Iris pattern class

The second experiment is to evaluate the classification performance of the proposed model. For experiments, 105 sign language video data was used. Each five person speaks seven korean sign languages three times. 70 video data are used for training and learning pattern classifiers, and the other 35 video data are used for test and performance evaluation process. 7 korean sign language were chosen for pattern class as shown in Table 2. In preprocessing module, skin color detection was processed using RGB data of video image. Afterwards, left, right, top, bottom of motion data position was found to cut off unnecessary data. After processing this task, MHV for each speaker could be built even if the location of speaker in video data is different from the others. Figure 10. is the example of skin color detection, segmentation and building MHV data. (a) is a skin color detection result. (b) is 3-dimensional data by extending time-domain sequences. (c) means the result of searching motion area. (d) means segmented MHV.

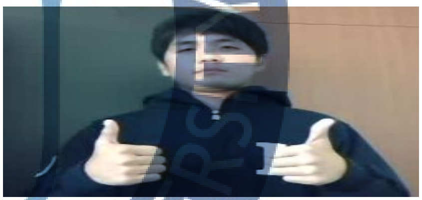
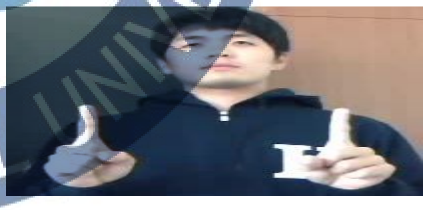
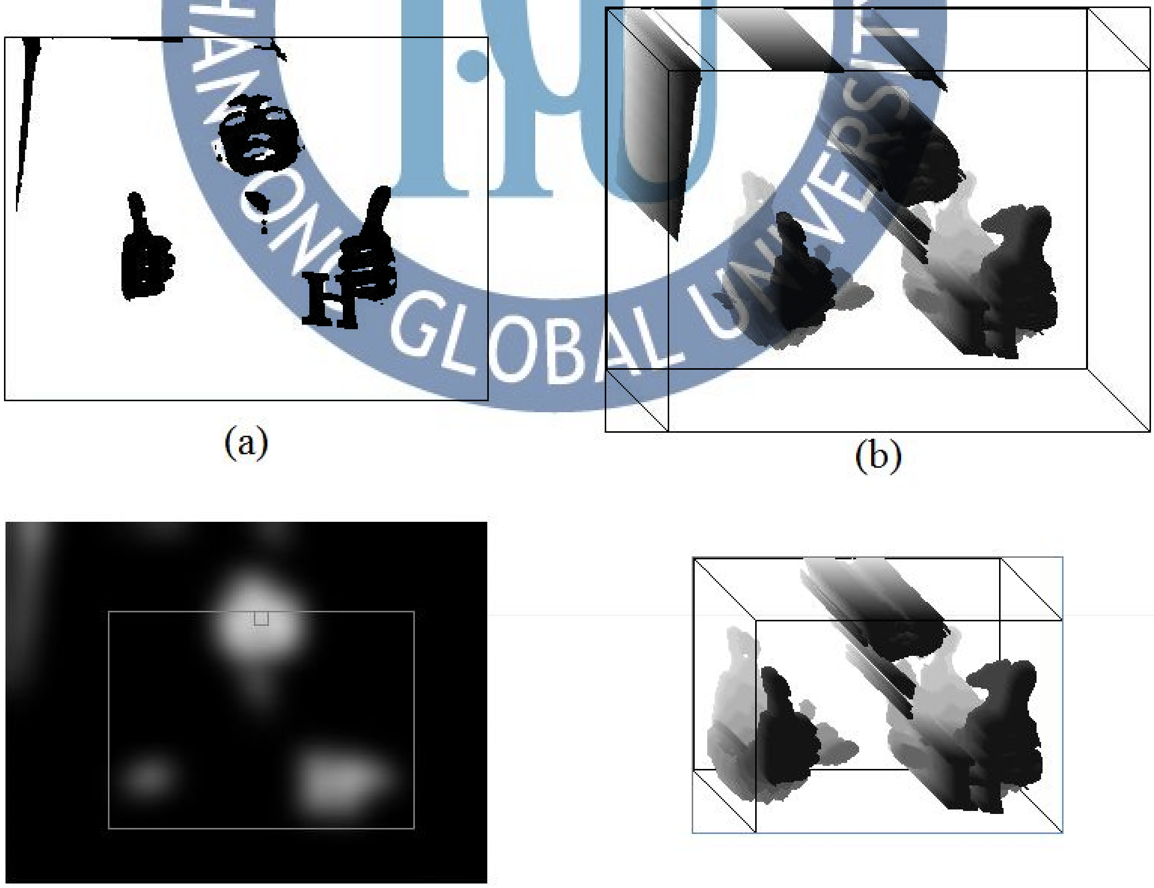
Pattern Class	Sign Language	Sample Image
P1	Greeting	
P2	Meet	
P3	Depart	
P4	Glad	
P5	Regret	
P6	Thank You	
P7	Very	

Table 4. Chosen 7 korean sign languages for pattern classification

(d) becomes an input of a feature extraction module.



(c) (d)

Figure 10. Example of results of preprocessing module

MHV data are divided into (18 X 18 X 18) data for feature extraction module. We use convolutional neural network to reduce feature space dimensions and to improve the distortion of variation of sign language image data. The output of CNN has (3 X 3 X 3) feature map data. Table 5 shows examples of created feature maps.

Pattern / Feature	P1	P2	P3	P4	P5	P6	P7
m1	0.253	0.188	0.240	0.148	0.573	0.343	0.045
m2	0.741	0.833	0.666	0.821	0.998	0.877	0.947
m3	0.140	0.337	0.060	0.573	0.155	0.146	0.689
m4	0.594	0.451	0.105	0.309	0.203	0.117	0.072
m5	0.390	0.247	0.370	0.577	0.477	0.471	0.466
m6	0.718	0.553	0.108	0.543	0.468	0.131	0.765
m7	0.784	0.622	0.368	0.243	0.039	0.129	0.171
m8	0.369	0.202	0.950	0.622	0.437	0.911	0.326
m9	0.768	1.000	0.246	1.000	0.627	0.298	0.915
m10	0.254	0.181	0.246	0.184	0.461	0.285	0.056
m11	0.761	0.886	0.669	0.871	1.000	0.879	0.975
m12	0.123	0.308	0.057	0.677	0.148	0.165	0.657
m13	0.425	0.365	0.128	0.339	0.208	0.135	0.417
m14	0.371	0.648	0.344	0.489	0.571	0.617	0.529
m15	0.570	0.482	0.163	0.656	0.410	0.162	0.284
m16	0.947	0.530	0.429	0.210	0.059	0.141	0.704
m17	0.479	0.619	0.694	0.429	0.436	0.968	0.419
m18	1.000	0.858	0.462	0.767	0.560	0.313	0.562
m19	0.307	0.169	0.253	0.089	0.570	0.378	0.080
m20	0.776	0.901	0.672	0.839	0.998	1.000	1.000
m21	0.166	0.291	0.055	0.902	0.146	0.153	0.626
m22	0.861	0.274	0.114	0.153	0.487	0.120	0.554
m23	0.439	0.801	0.257	0.381	0.542	0.482	0.371
m24	0.769	0.423	0.241	0.969	0.409	0.138	0.177
m25	0.650	0.416	0.470	0.240	0.054	0.100	0.824
m26	0.336	0.783	0.478	0.537	0.375	0.820	0.103
m27	0.632	0.767	1.000	0.730	0.810	0.302	0.443

Table 5. Examples of feature maps of each pattern

After creating feature maps, using these feature map values, modified FMM neural network are built and trained. Total 70 sign language image data, 2 image for 5 sign language speaker of 7 sign language are used for training, and the other 35 sign language image data are used for sign language pattern classification test. Table 6 is the result of testing 35 patterns.

Input Pattern	Classification Output	Result
1	1	T
2	2	T
3	3	T
4	4	T
5	5	T
6	6	T
7	7	T
1	4	F
2	2	T
3	3	T
4	4	T
5	3	F
6	6	T
7	7	T
1	1	T
2	2	T
3	3	T
4	4	T
5	5	T
6	6	T
7	7	T

1	1	T
2	2	T
3	3	T
4	4	T
5	5	T
6	6	T
7	7	T
1	4	F
2	2	T
3	3	T
4	4	T
5	5	T
6	6	T
7	7	T

Table 6. Result of the test using 35 sign language pattern data

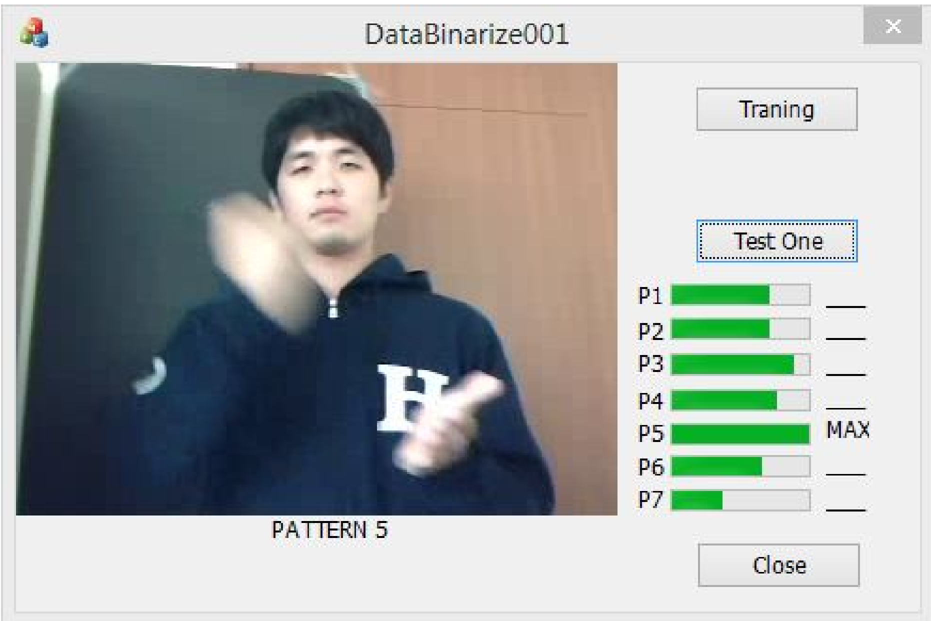


Figure 11. Application of sign language recognition system

Figure 11 is an application user interface of sign language recognition system.

2. Relevance Factors and Rule Extraction

By using the weight value of trained modified FMM neural network classifier, relevance factor between the features and each classes can be calculated. Table 7 shows a relevance factors calculated using weight value of modified FMM neural network through experiments.

Pattern / Feature	P1	P2	P3	P4	P5	P6	P7
m1	-0.523	-0.257	-0.341	-0.755	-0.298	-0.460	0.717
m2	-0.104	-0.013	-0.104	0.515	0.515	-0.898	-3.719
m3	0.011	0.093	0.574	-0.538	-0.503	-0.980	0.011
m4	-0.436	-0.106	-0.436	-0.436	-1.148	0.098	0.660
m5	-0.276	-0.644	-0.235	0.139	-0.029	0.139	0.373
m6	-0.350	-0.283	-0.144	-0.172	-0.004	0.701	-0.598
m7	0.123	-0.017	-0.017	-0.051	-0.295	0.016	0.144
m8	0.314	-0.540	-2.290	-0.340	-0.849	0.553	0.307
m9	-1.569	0.395	-0.291	-0.026	0.019	0.600	-0.446
m10	-0.708	0.264	0.018	-1.049	-0.088	-3.098	0.668
m11	-0.104	-0.013	-0.104	0.515	0.515	-0.898	-3.719
m12	0.154	0.154	0.607	-4.041	-0.408	-0.858	0.076
m13	0.167	-0.247	0.167	-0.055	-2.285	0.167	0.321
m14	-0.086	0.004	-0.321	0.477	-1.367	0.274	-0.254
m15	-1.789	0.037	-0.829	-0.125	0.037	0.717	-1.387
m16	-1.161	0.375	-0.228	0.375	0.160	0.533	-1.151
m17	0.101	-0.063	-0.692	-0.286	-0.777	0.529	0.101
m18	0.151	-0.394	-0.741	0.151	-0.665	0.478	0.015
m19	-1.171	0.004	0.524	-2.163	-0.833	-0.085	0.524
m20	-0.364	-0.255	-0.364	0.381	0.381	0.381	-0.467
m21	-0.106	0.167	0.614	-1.833	-0.389	-0.833	0.090

m22	-0.251	0.407	-1.172	-0.499	-0.233	0.344	0.187
m23	0.224	0.000	-0.054	-0.054	-1.148	0.116	0.224
m24	-0.666	0.248	-0.283	-0.833	-0.019	0.695	-0.369
m25	-0.656	0.000	-1.388	0.542	-0.124	0.151	-0.018
m26	0.252	0.317	0.252	-0.597	-0.362	-0.182	-0.319
m27	-0.226	0.287	0.265	-0.301	-0.256	0.555	-1.022

Table 7. Calculated relevance factors between features and pattern classes

Relevance factor $RF(I_{ji},k)$ can be a positive value or a negative value. Positive relevance factor presents that the feature related relevance factor has an excitatory characteristic for the related pattern class. In contrast, a negative relevance factor means it has an inhibitory relationship between feature and pattern class.

Using this analysis of $RF(I_{ji},k)$, set of rules can be built for the pattern recognition. Association degree between the features and our goal patten classes can be analyzed. Threshold value for relevance factor to reduce a feature dimension for improving performance of pattern classifier can be meas as well.

Another step is finding the $IF-THEN$ rules by analyzing the relevance factor. The following rules are IF-THEN rules built using relevance factor and hyperbox fuzzy partition.

IF m20(=center-upper-latter) is 'LARGE' THEN P5(=regret) with cf = 0.381

IF m19(=left-upper-latter) is 'SMALL' THEN P7(=very) with cf = 0.524

IF m10(=left-upper-middle) is 'MEDIUM' THEN not P1(=greeting) with cf = 0.708

Ⅵ. Conclusion

Sign language recognition is an extension research of gesture recognition system. Proposed underlying system consists of multi-layer structure for sing language recognition. In this study, a recognition system model for sign language was proposed. Three-dimensional data, named as Motion history volume (MHV) is used as input of feature extraction module. Also three-dimensional extended convolutional neural network is used as feature extractor. By the effect of CNN model, a spatio-temporal variation of motion data can be adjusted. For pattern classifier, modified fuzzy min-max(FMM) neural network is proposed. Overlapping problem of original FMM model is improved adapting frequency concept of input pattern. Modified FMM neural network is trained, and relevance factor is calculated to build a rule extraction method. The experiments are processed into two step. First experiment is comparison test between original FMM and proposed modified FMM neural network. The experimental results show the improvement of performance of proposed model comparing original FMM network model by solving hyperbox overlapping problem. Second experiment is a recognition test using a sign language image data. We apply our proposed model to a gesture recognition problem, especially using a sign language data. The performance result shows that our proposed model can be used to a gesture recognition problem such as sign language recognition. The IF-THEN rules could be built using calculated relevance factor and its related fuzzy partition of feature space. The advantage of rule extraction is an availability of relevance factor as an initial weight value for pattern classification model.

Ⅶ. References

1. Simpson, P. K. 1992. Fuzzy min-max neural networks-part 1: classification. *IEEE Transactions on Neural Networks* 3(5), 776-786.

2. Gabrys, B., A. Bargiela. 2000. General fuzzy min-max neural network for clustering and classification. *IEEE Transaction on Neural Networks* 11(3), 769-783

3. Quteishat, A., C. P. Lim. 2008. A modified fuzzy min-max neural network with rule extraction and its application to fault detection and classification. *Applied Soft Computing* 8, 985-995.

4. Quteishat, A., C. P. Lim, K. S. Tan. 2010. A modified fuzzy min-max neural network with a genetic-algorithm-based rule extractor for pattern classification. *IEEE Transactions on Systems, Man, and Cybernetics − Part A: Systems and Humans* 40(3), 641-650.

5. Sawrence, S., C. L. Giles, A. C. Tsoi, A. D. Back. 1997. Face recognition: a convolutional neural-network apparoach. *IEEE Transactions on Neural Networks* 8(1), 98-113.

6. Simard, P. Y., D. Steinkraus, J. C. Platt. 2003. Best practices for convolutional neural networks applied to visual document analysis. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pp. 958-963.

IEEE Conference Publications.

7. Zaki, M. M., S. I. Shaheen. 2011. Sign language recognition using a combination of new vision based features. *Pattern Recognition Letters* 32, 572-577.

8. Ho-Joon Kim. 2012. Two-Stage Neural Networks for Sign Language Pattern Recognition. *Journal of Korean Institute of Intelligent Systems* 22(3), 319-327

9. Ho-Joon Kim, So-Jung Park. 2013. A hyperbox-based neural network for dynamic hand gesture recognition. *Proceedings International Conference, Advances in Computer and Computing Science* pp. 78-81. IST COMCOMS ICMS