ISSN No: - 2456 – 2165

Hand Gesture Recognition System Using Kohonen Self-Organizing Map

Juliet P. Cagampang[1]
[1]Computer Engineering Department
University of Science and Technology of
Southern Philippines, Cagayan de Oro City Philippines

Sherwin A. Guirnaldo[2]
[2]Mechanical Engineering Department
Mindanao State University-Iligan Institute of Technology
Iligan City Philippines

Abstract—Hand gesture is a nonverbal communication which is very useful to the deaf and mute. It is also used as an alternative way to communicate with computers. Hand gesture recognition has a wide range of applications such as recognizing of sign language, interfaces for human-computer interaction, robot control, machine vision, smart surveillance, computer games, keyboards and mice replacement. This paper described a hand gesture recognition system, a vision-based approach, to recognize static hand gesture images using Kohonen Self-Organizing Map (SOM), an artificial neural network which learns to classify data without supervision. A set of 29 hand gesture images representing letters of the alphabet, enter, space and backspace keys were captured using a CMU camera. The images were cropped using a photo editor and the edited images were converted to grayscale using the MATLAB software. These images in 1D form were used as training set for the Kohonen Self-Organizing Map. After the unsupervised training, the system was tested using 29 actual hand gestures and 10 trials for each gesture. The system achieved an average of 91% accuracy with only 9% error. The system's recognition accuracy may be further improved by increasing the number of epochs in the training phase, experimenting to find a better learning rate, using a high-resolution camera to capture the image more precisely to minimize the amount of background noise resulting to a more defined input feature vector to be fed to the SOM.

Keywords—Artificial Neural Network, Image Processing, Kohonen Self-Organizing Map, Vision-Based Approach.

I. INTRODUCTION

Hand gesture is the most expressive and frequently used among a variety of human gestures for communication. It is also the most natural way for humans to communicate with computers. This kind of human-computer interaction is possible now that computers are equipped with cameras to allow vision based interfaces. Using the hand as an input device provides an easy and fast way for humans to interact with computers rather than using the keyboards and mice. Hand gesture recognition has a wide range of applications such as recognizing of sign language, interfaces for human-computer interaction, robot control, machine vision, smart surveillance, computer games, keyboards and mice replacement [1].

There are two different approaches in hand gesture recognition, glove-based and vision-based [1][7]. Glove-based approach requires the user to wear gloves with sensors like magnetic field tracker which configures hand position such as angles, rotations and movements [7]. This glove-based approach can give very accurate results, however the cost for the hardware is expensive and it requires the user to wear a cumbersome device and requires complex calibration and setup procedures to be able to obtain precise measurements [6][7]. On the other hand, vision-based approach is a promising alternative where it only needs a camera to capture the input from the user. It is the most natural way of human-computer gestural interface.

Hand gestures can be classified in two categories: static and dynamic [1][7]. A static gesture is a particular hand posture determined by a particular finger thumb-palm configuration and represented by a single image. A dynamic gesture is a moving gesture, represented by a sequence of images [1].

Using artificial neural networks for pattern (image) recognition has many advantages compared to other techniques like statistical modeling, programming logic or using database of patterns which require much memory space [1]. Neural networks can learn to recognize patterns that exist in a data set. They are flexible in a changing environment. Rule-based systems or programmed systems are limited to the situation for which they were designed. When conditions change, they are no longer valid. Neural nets are excellent at adapting to changing information [1].

A neural network (net) can be supervised or unsupervised. Examples of neural nets that use the supervised technique are Perceptron and Backpropagation [2] in which a target output is required so that the error can be computed which is the basis for updating or adjusting of the network's connection weights.

ISSN No: - 2456 – 2165

Kohonen Self-Organizing Map is a neural net that use the unsupervised technique [2][4][5]. This neural net does not require a target output. It is able to learn the input pattern on its own.

This paper focused on the vision-based approach for recognizing static hand gestures representing letters of the alphabet, enter, space and backspace keys captured using a CMU camera and converted to computer readable form by image processing and used as input for the training, setting boundaries and testing of Kohonen Self-Organizing Map.

II. IMAGE PROCESSING

The following stages in image processing are involved in preparing the hand gesture images for the three phases of the system's development (SOM training, setting boundaries and SOM testing), 1) background subtraction, 2) image cropping, 3) resizing, 4) gray scaling and 5) conversion to 1D.

Background subtraction involves separating potential hand pixels from non-hand pixels. Since the camera is mounted above a nonmoving workspace, a simple background subtraction scheme is used to segment any potential foreground hand information from the fixed background scene. At system startup, a background image I_B is captured to represent the static workspace from the camera view. Subsequent frames then use the background image to segment out foreground data. For each pixel in frame I_i, we compute the foreground mask image I_F as follows:

$$I_F = \begin{cases} I_i & \text{if } |I_i - I_B| > \sigma_B \\ 255 & \text{otherwise} \end{cases}$$

where σ_B is a fixed threshold to differentiate foreground data from the background data. Background subtraction is performed in RGB color space with 8 bits per color channel. The resulting I_F is RGB image with a single 8-bit channel. After some experimentation, a value of 10 for the threshold was found to provide good results.

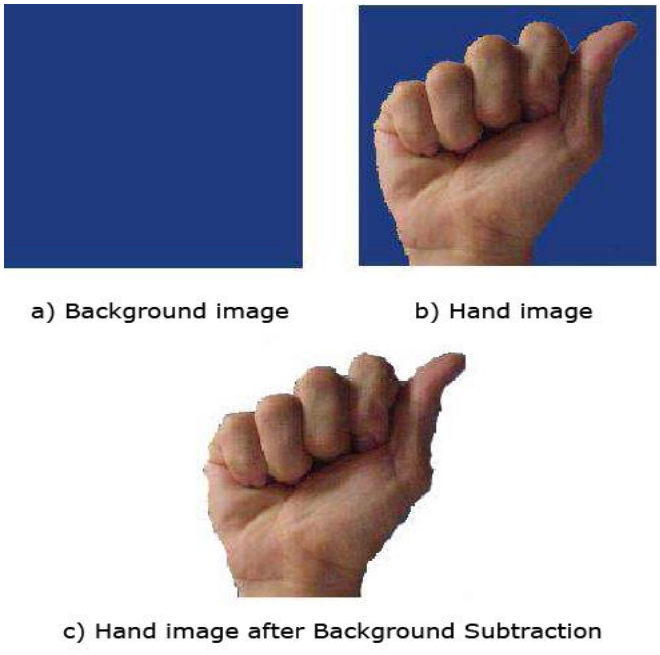

Fig. 1: Background Subtraction

Fig. 1 is an example of background subtraction. Fig. 1a is a snapshot of a static background. A captured image of a hand gesture is presented in Fig. 1b. Each component of ***a*** is being subtracted from **b:** ***b(i) – a(i) = \|c(i)\|***. If the absolute value of the difference ***c(i)*** is greater than the threshold which is 10, then the value of ***c(i)*** is equal to ***b(i)***, else it is 255 for white color. Fig. 1c is the result of background subtraction.

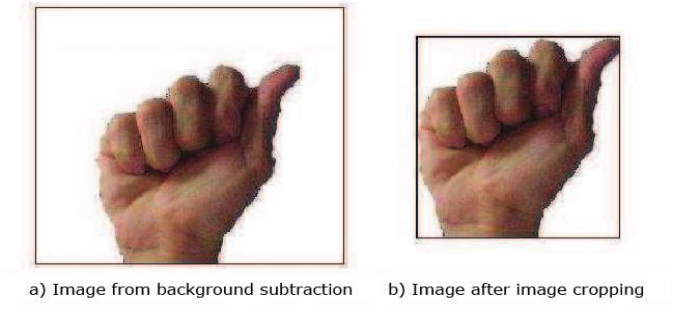

Fig. 2: Image Cropping

After background subtraction, the image shown in Fig. 2a is cropped to magnify the primary subject of the image which is the hand or the skin color. Image cropping is done by evaluating the four sides of the image and removing all lines of pixels having white values. Fig. 2b is the resulting image which is then resized to 30 x 40 pixels and converted to grayscale. The MATLAB code below do this magic.

ISSN No: - 2456 – 2165

```
x = double(image) / 1000;
y = imresize(x, [30 40]);
gray_scale_img = rgb2gray(y);
```

Finally, the image is converted from 30 x 40 pixels to 1D 1200 x 1 pixels using the MATLAB code below. The images were transformed to meet the MATLAB Neural Network Toolbox requirements [3].

```
z = gray_scale_img(:);
```

III. KOHONEN SELF-ORGANIZING MAP

Kohonen Self-Organizing Map, invented by Teuvo Kohonen, a professor of the Academy of Finland, assume a topological structure among the cluster units. This property is observed in the brain, but it is not found in other artificial neural networks. There are m cluster units, arranged in a one- or two-dimensional array. The input signals are n-tuples [2][4].

The weight vector for a cluster unit serves as an exemplar of the input patterns associated with that cluster. During the self-organizing process, the cluster unit whose weight vector matches the input pattern most closely, typically the square of the minimum Euclidean distance, is chosen as the winner. The winning unit and its neighboring units, in terms of the topology of the cluster units, update their weights [2][5]. The weight vectors of neighboring units are not close to the input pattern [2]. Fig. 3 below shows a sample SOM network architecture [9]. It consists of 16 cluster units and 2 input units. Each input unit is connected to all cluster units. A cluster unit has 2 connection weights since there are 2 input units.

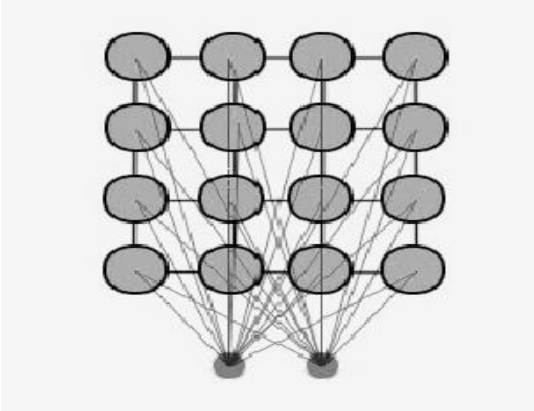

Fig. 3: SOM Architecture 4x4-16 clusters and 2 input nodes

IV. TRAINING THE SOM

There are three phases in the development of the system as shown in Fig. 7. First is the training phase, second is setting boundaries, and third is the testing phase. In the training phase, ten samples of each hand gestures representing letters A to Z, Space, Backspace and Enter keys were collected from different hand owners. Fig. 6 shows the 29 hand gesture images and the background image. In each phase the five stages of image processing were performed on all images. All of the processed images were concatenated to create a single output before being presented to the SOM. The MATLAB Artificial Neural Network (ANN) Toolbox [3] is utilized. The SOM was set to 20 x 20 dimensions, creating 400 cluster

1 i = 1..n input units, j = 1..m clusters, wij = connection weight from input i to cluster j
2 Initialize connection weights wij
3 Set topological neighborhood parameters.
4 Set learning_rate parameters.
5 For some number of epochs do line 6 to 12:
6 For each input vector x, do line 7 to line 11:
7 For each j, compute:
8 $D(j) = \sum (wij - xi)^2$
9 Find index j, such that D(j) is the minimum
10 For all units j within a specified neighborhood of j, and for all i:
11 wij(new) = wij(old) + learning_rate [xi – wij(old)]
12 Update the learning_rate.

Fig. 4: SOM Training/Learning Algorithm

ISSN No: - 2456 – 2165

Fig. 4 shows the training or learning algorithm of a Kohonen SOM [2]. Fig. 5 below illustrates the training or learning of SOM [8]. The blue blob is the distribution of the training data set, and the small white region of the blue blob is the current training sample drawn from the distribution [8]. At the start, in Fig. 5a the SOM nodes are randomly positioned in the data space. The node, in yellow color, nearest to the training node is selected, and is moved towards the training data, the white region, as shown in Fig. 5b as well as its neighbors on the grid to a lesser extent. After many iterations or epochs, the grid tends to approximate the data distribution Fig. 5c [8]

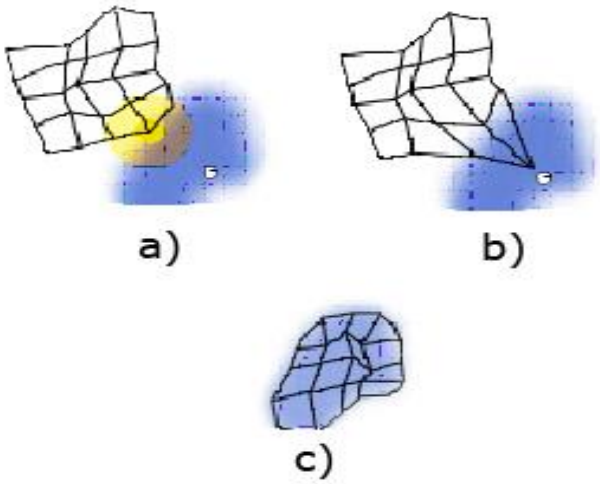

Fig. 5: Illustration of SOM Training or Learning

Units. The training took 10K epochs and it lasted about a total of 24 hours. The trained SOM (connection weights) is saved and is used in the succeeding phases, setting boundaries and testing phase. Fig. 8 shows the trained SOM network and Fig. 9 shows the neighbor distances of all hand gesture images or the topological structure.

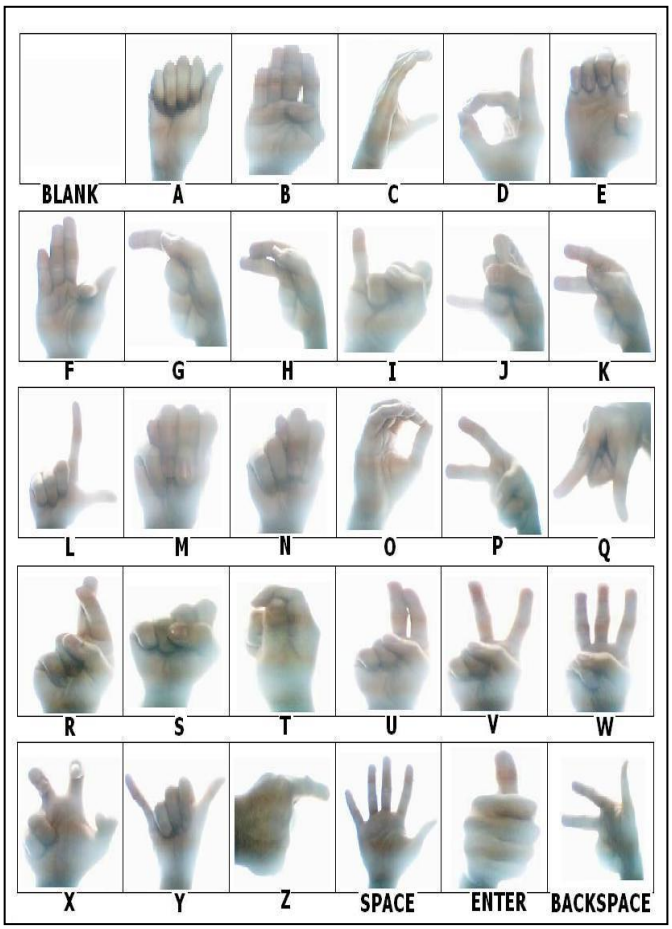

Fig. 6: Hand Gesture Images

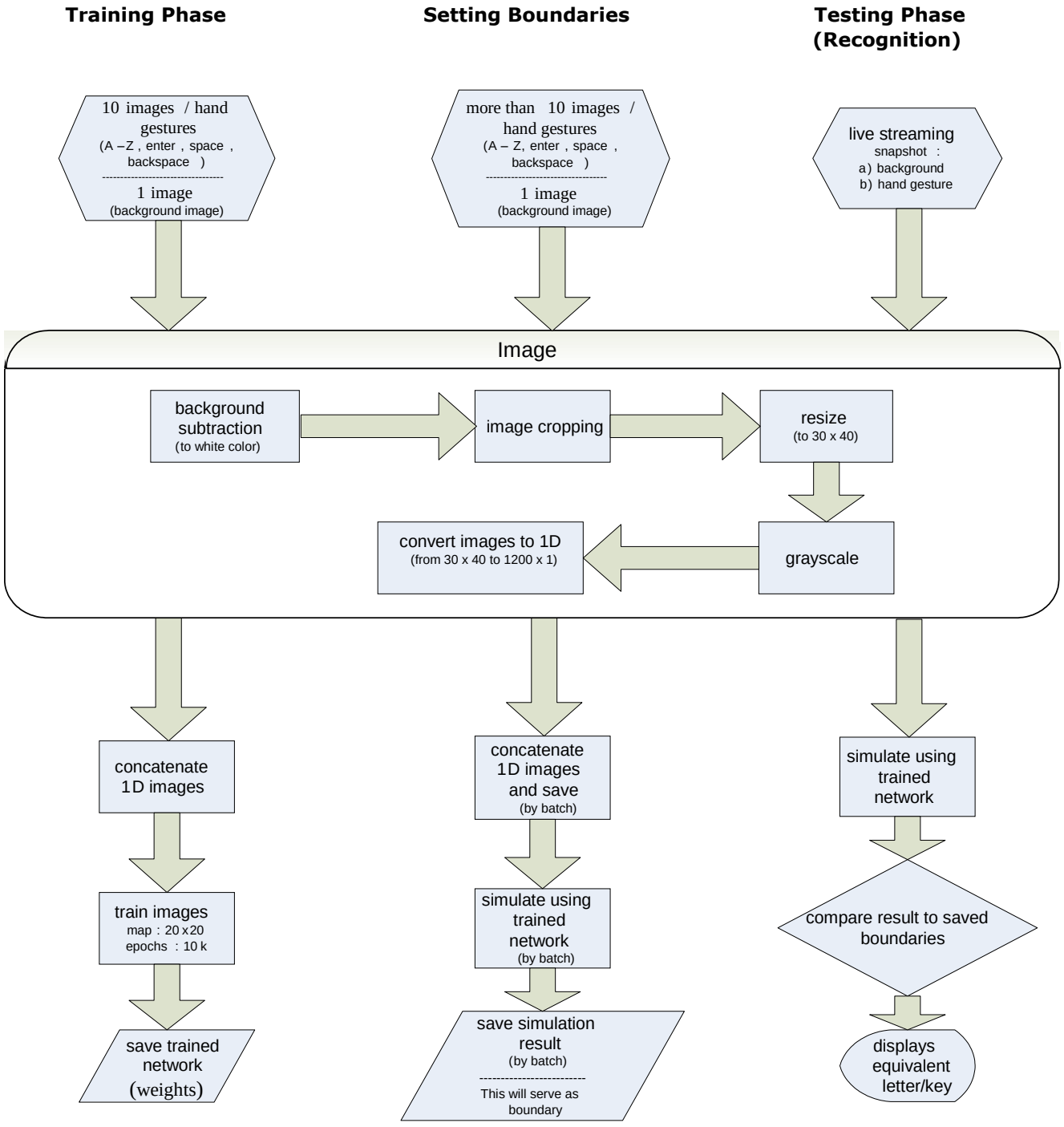

Fig. 7: Conceptual Framework of the System

ISSN No: - 2456 – 2165

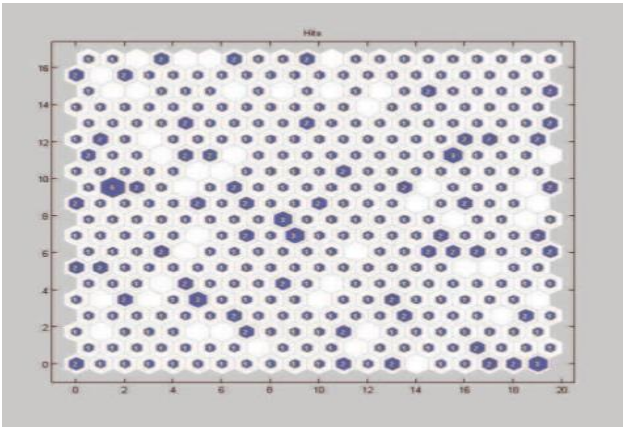

Fig. 8: SOM Training Result

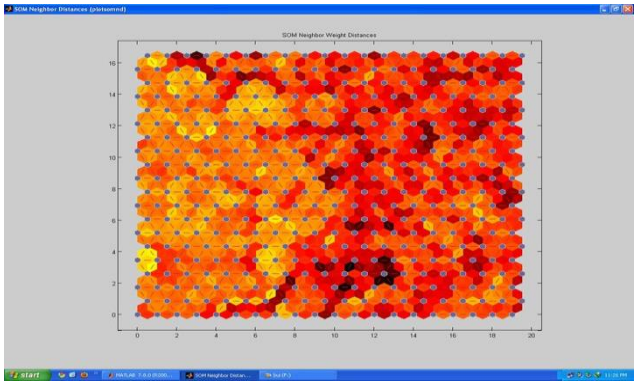

Fig. 9: Neighbor Distances of all Hand Gesture Images

After the training, each of the 1D image is concatenated with its hand gesture representation and each is simulated using the trained SOM network. The output is saved and used as a boundary of the hand gesture. Fig. 10 shows the boundaries of selected hand gestures for letters A, B and C.

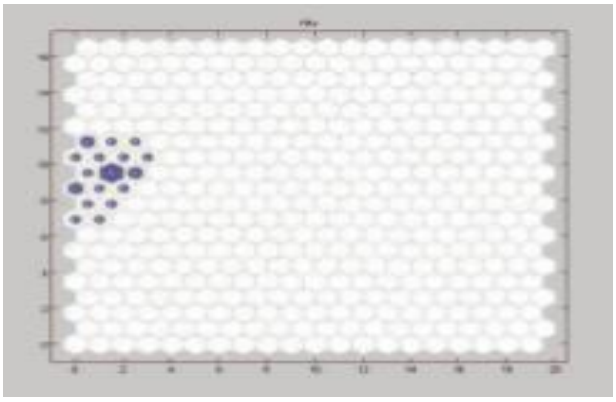

Fig. a) Boundary of "A"

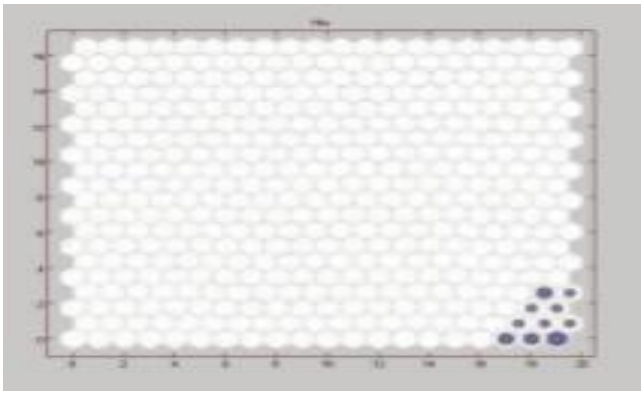

Fig. b) Boundary of "B"

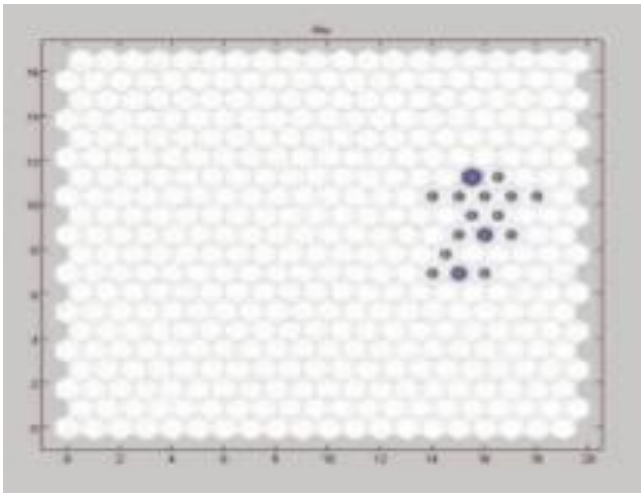

Fig. c) Boundary of "C"

Fig. 10: Hand Gesture Boundaries for letters A, B and C.

V. RESULTS AND DISCUSSION

In the testing or recognition phase, a web camera on live streaming is used. A snapshot from the live stream will pass through stages of image processing. The converted image is simulated or fed to the trained SOM network. The result of the SOM is compared to the saved boundaries and if there is a match then the input hand gesture is identified. Otherwise, the output would be NM or No Match.

Table I below shows the results of the testing phase. There were 10 trials per hand gesture image, making the total number of images 290. The table shows that 264 out of 290 trials matched, resulting to an average of 91% accuracy and an error of 9%.

ISSN No: - 2456 – 2165

Hand Gestures	Matches (out of 10)
A	9
B	10
C	8
D	10
E	9
F	10
G	9
H	9
I	9
J	9
K	8
L	10
M	8
N	8
O	8
P	8
Q	10
R	9
S	9
T	9
U	8
V	8
W	9
X	10
Y	10
Z	10
Space key	10
Backspace key	10
Enter key	10
TOTAL	264 matches
AVERAGE	264 matches / 290 trials **ACCURACY = 91%** and ERROR = 9%

Table 1: Results of the Testing Phase

VI. CONCLUSION

In this paper, we presented a system to recognize static hand gestures using trained Kohonen Self-Organizing Map. The hand gesture images were first converted to computer readable form using image processing before presenting them to the SOM. The advantage of Kohonen Self-Organizing Map compared to other popular ANNs such as Perceptron and Backpropagation is that it can learn to classify data without supervision. Results show that the system achieved an accuracy rate of 91% with only 9% error rate. The system's recognition accuracy may be further improved by increasing the number of epochs in the training phase, experimenting to find a better learning rate, using a high-resolution camera to capture the image more precisely to minimize the amount of background noise resulting to a more defined input feature vector to be fed to the SOM.

VII. ACKNOWLEDGMENT

The authors would like to thank the Computer Engineering Department of University of Science and Technology of Southern Philippines, Cagayan de Oro City for the support extended to this project. Further, the authors would like to acknowledge the technical support of Neil Fergus B. Macas, Marc Joel P. Yañez, Jeram Ray C. Tejano, Ryan Kim B. Janobas and Carlo Mark J. Rana for making this research a success.

REFERENCES

[1]. Klimis Symeonidis, "Hand Gesture Recognition Using Neural Networks". August 23, 2000.
[2]. Laurene Fausett, Fundamentals of Neural Networks-Architectures, Algorithms, and Applications. 1994, pp. 169-185.
[3]. Howard Demuth, Mark Beale, Martin Hagan (2010), "Neural Network Toolbox: User's Guide", 6th Edition.
[4]. Teuvo Kohonen (1989), Self-organization and Associative Memory (3rd edition), Berlin: Springer-Verlag.
[5]. Kevin Pang (2003), Self-organizing Maps. Neural Networks.Available:https://www.cs.hmc.edu/~kpang/nn/som.html#refer (Accessed: September 2017)
[6]. Mohamed Alsheakhali, Ahmed Skaik, Mohammed Aldahdouh, Mahmoud Alhelou, Hand Gesture Recognition System. Computer Engineering Department, The Islamic University of Gaza, Gaza Strip, Palestine, 2011.
[7]. Nadira Nordin, Introduction Human Computer Interaction (HCI) – Hand Gesture. Available: http://urrg.eng.usm.my/TO_DELETE/index.php?option=com_content&view=article&id=169:introduction-human-computer-interaction-hci--hand-

gesture&catid=31:articles&Itemid=70 (Accessed: September 2017)
[8]. Available: https://en.wikipedia.org/wiki/Self-organizing_map (Accessed: September 2017)
[9]. Available:http://www.aijunkie.com/ann/som/som1.html (Accessed: September 2017)