ARTIFICIAL NEURAL NETWORK BASED HANGUL CHARACTER RECOGNITION

Debrelie Cruz*, Shaira Yvonne Cruz*, Krystal Dhanilei Decembrada*, Lucky Deogrades*, Earl Richard Dizon*, Jerica Mae Fabia*, Joanna Marion Flores*, Mark Daniel Marmeto*, Caryl Andre Martin*, Carl Joshua San Diego*, Jonathan Tolentino*, Ranil Montaril*

**Department of Computer Science, Polytechnic University of the Philippines, Manila, Philippines*

***Abstract* – A tool for recognizing Hangul characters using Artificial Neural Network (ANN) tool is presented on this paper. Using MATLAB R2014a, the tool was implemented with ANN to be able to create networks for the training of data - which is an important process for the recognition of Hangul characters. This paper also discusses how ANN tool is used and how it is being used in image recognition.**

Keywords - Artificial Neural Network; Hangul; Optical Character Recognition; Training;

I. INTRODUCTION

Computer applications are widely used around the globe. Many advanced technologies were invented and computer programs are becoming more and more complex. Data processing is one of the most common use of technologies nowadays. These data includes documents, video files, audio files, and images. As humans, we can analyze data visually, but for computers, it is a requirement to have an algorithm so that machines can effectively extract information from a file. The difficulty of visual pattern recognition becomes apparent if you attempt to write a computer program to recognize digits or letters. What seems easy when we do it ourselves suddenly becomes extremely difficult. Simple intuitions about how we recognize shapes - "a 9 has a loop at the top, and a vertical stroke in the bottom right" - turn out to be not so simple to express algorithmically. When you try to make such rules precise, you quickly get lost in a morass of exceptions and caveats and special cases [1].

In doing so we need to use such techniques like 'Pattern Recognition'. Pattern recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data, although it is in some cases considered to be nearly synonymous with machine learning. Pattern recognition systems are in many cases trained from labeled "training" data (supervised learning), but when no labeled data are available other algorithms can be used to discover previously unknown patterns (unsupervised learning) [2]. Now we have identified a technique that we need to use, we shall add an algorithm to effectively recognize pattern such as Neural Networks or for computers, the Artificial Neural Networks. Artificial neural networks have shown great strength in solving problems that are not governed by rules, or in which traditional techniques have failed or proved inadequate. The inherent parallel architecture and the fault tolerant nature of the ANN is maximally utilized to address problems in variety of application areas relation to the imaging field. Artificial neural networks find their application in pattern recognition (classification, clustering, and feature selection), texture analysis, segmentation, image compression, color representation and several other aspects of image processing [3].

II. BACKGROUND OF THE STUDY

A. Key Terms

a) Artificial Neural Network (ANN) – an information processing paradigm that is inspired by biological nervous systems, such as the brain, process information. It is composed of highly interconnected processing elements working in unison to solve specific problems.

b) Optical character recognition (OCR) – is the translation of optically scanned bitmaps of printed or written text characters into character codes, such as ASCII.

c) Hangul – is the official alphabet used to write the Korean Language that is used by both the North and South Korea. It is composed of 19 consonants and 21 vowel letters.

B. Problem Statement

Hangul is Korea's official alphabet. It is primarily used in the North and South Korean, as well as some other parts of China near these countries. This alphabet can be considered only as a minor one, as it does not reach a global importance. With this, many will struggle in learning and trying to recognize these characters. In connection with AI, Recognition is of the most important problem area, as an agent can only choose and perform an action if it can recognize its environment. Recognition can be a problem when the agent doesn't know what to recognize. So we don't only want an agent that can recognize, we also want an agent who can learn to recognize. To be able to recognize these Hangul characters intelligently, we can use Artificial Neural Networks, in which classification is its domain. Using ANN's, we can classify each Hangul character and thereby reduce the time and complexity an agent can spend in recognizing these characters.

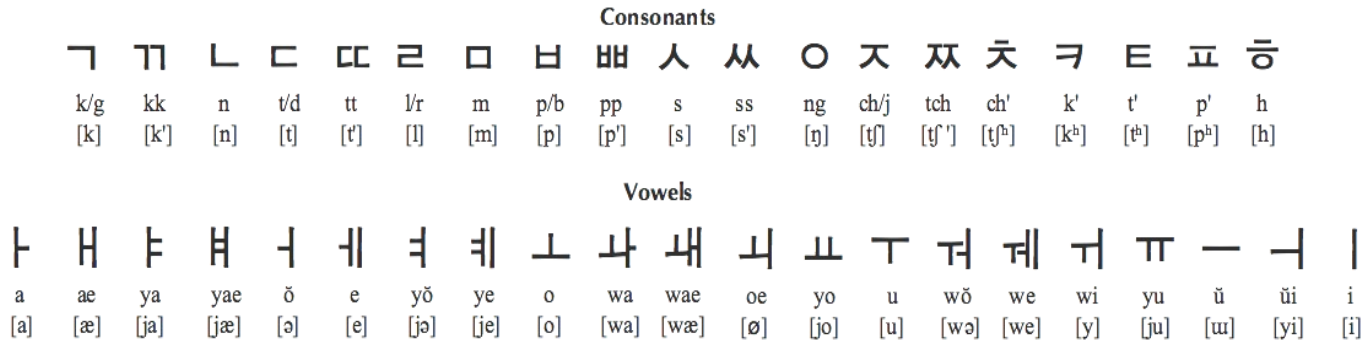

Fig. 2.1 List of Hangul Characters

In Fig. 2.1, it shows the list of Hangul characters such as the consonants and the vowels that this study will recognize using the method of Artificial Neural Network.

C. Applicable Related Studies

According to the International Journal of Research in Engineering and Technology, Optical character recognition is an effective technique which converts image into suitable format such that data can be edited or modified. This technique performs several operations such as, scanning the input image and performing processes so that the image gets converted into portable formats. One very practical application of this according to the IJRET, is the hard copy of old historical books. They cannot be stored safely for a long time. Applying the OCR technique, these historical documents can be stored, modified for a longtime [4]. The Optical Character Recognition technique, however cannot be done without special tools and algorithms. As per stated by Ziga Zadnik, Neural networks can be used, if we have a suitable dataset for training and learning purposes. OCR engines add the multiple algorithms of neural network technology to analyze the different aspects in recognizing a character, such as the stroke edge, the line of discontinuity between the text characters, and the background. It allows for irregularities of printed ink on paper have a match on the known characters and makes a best guess as to which character it is [5]. However, Alexander J. Faaborg said that while neural networks are a promising solution there exist some short term problems. Training a neural network can consume a large amount of time and processor and this has resulted in some researchers advocating 'lazy recognition', not attempting to do character recognition in real time [6].

III. SOFTWARE DESIGN ARCHITECTURE

A. Applicable Equation

Each Hangul character was mapped into a 15-by-13 grid, filling empty spaces with 1's and a drawn cells with 0's. This combination allows to represent each character in its binary form. Figure 3.1 shows an example.

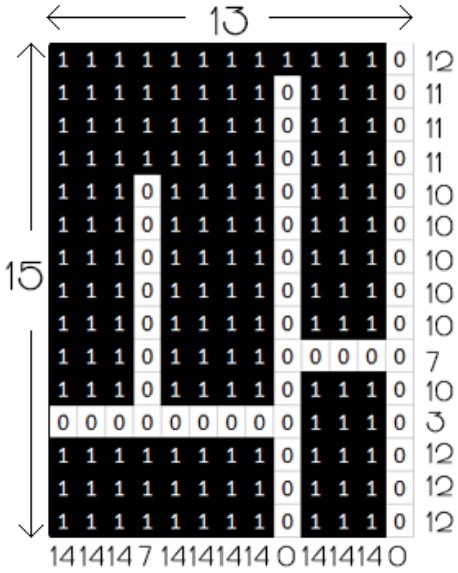

Fig. 3.1 Binary Representation of a Hangul Character

This binary representation can be used to aid the design of the ANN Architecture, as it simplifies how we represent each Hangul character. To do this, we computed the sums of the X and Y planes of each character using the following equations.

For X-Plane:

$$\sum_{i=1}^{X_{13}} A_i = X_1 + X_2 + X_3 + \cdots + X_{13} \qquad (1)$$

WHERE,

i : index of summation

A_i : indexed variable representing each successive term

X_1 : lower bound of summation

X_{13} : upper bound of summation

For Y-Plane:

$$\sum_{i=1}^{Y_{15}} A_i = Y_1 + Y_2 + Y_3 + \cdots + Y_{15} \qquad (2)$$

WHERE,

i : index of summation
A_i : indexed variable representing each successive term
Y_1 : lower bound of summation
Y_{15}: upper bound of summation

B. *Functional Block Diagram*

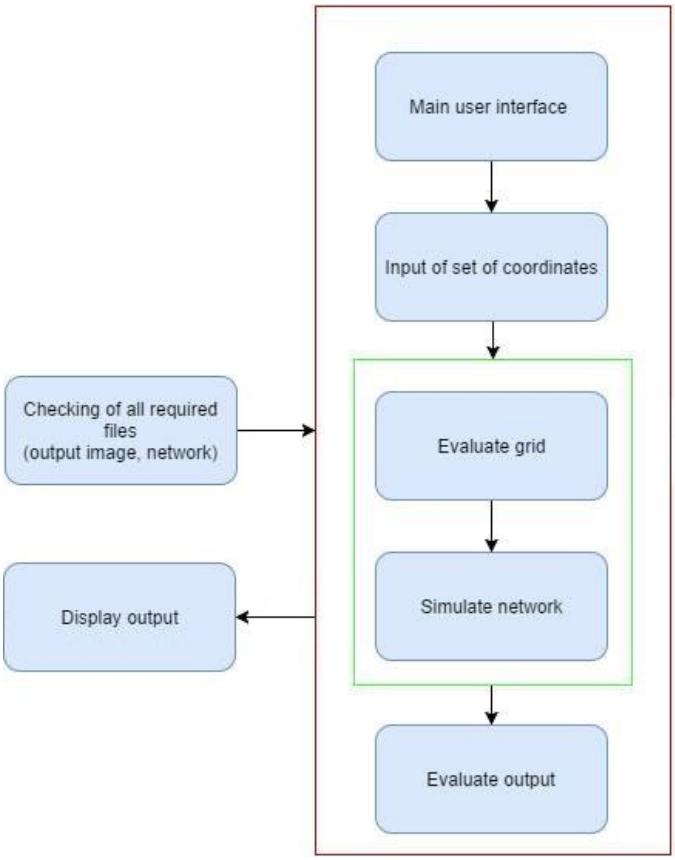

Fig. 3.2 Functional Diagram

The program will first check if all the required files are present in the same folder as with the program. When done validating whether these files (output image and network) exist, the user shall be directed to the main user interface wherein the user will select the coordinates from the grid that represent the Hangul character that the user wishes to recognize. The program will evaluate the grid that contains the input of the user, then it will simulate the network that is created through MATLAB. The evaluated input will be checked against the target output data to see whether the inputted character has a match in the list of characters defined.

The program will output a number that represents the corresponding character. The outputted number has to undergo a series of conditions to find the exact character that corresponds to it. Lastly, when the input character matches any of the target data, the character will be displayed.

C. *Flowchart*

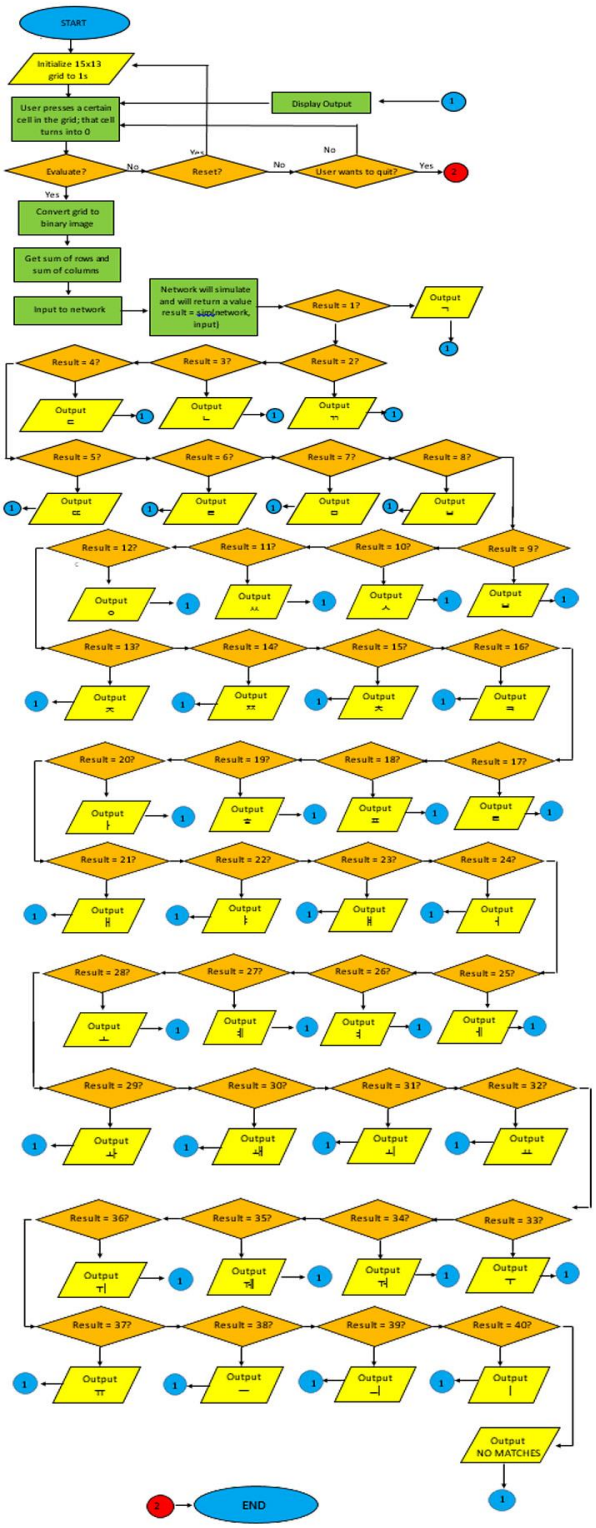

Fig. 3.3 Program Flowchart

Fig. 3.3 shows the flowchart of the Hangul Character Recognition program. The program will initialize the grid to 1s. Second, the user will be allow to press a certain cell in the grid; that cell turns into 0. Then it can now evaluate. It will ask if you want to evaluate, if no, then it will ask again if the user wants to reset the grid, if yes, it will re-initialize the grid to 1s. Else, it will ask if the user wants to quit, if yes, the program will terminate. Else the user will be allow to press a certain cell in the grid again. Going back to the evaluating part, if the user chose yes, then the grid will be converted into a binary image. Next, it will get the sum of rows and sum of columns then these sums will become an input to the network. After that, the network will simulate and will return a result value of sim (network, input). Finally, if the result is equal to a certain number, it will display the corresponding letter. Else it will display an output "No Matches". Then the process will repeat.

D. Artificial Neural Network Architecture

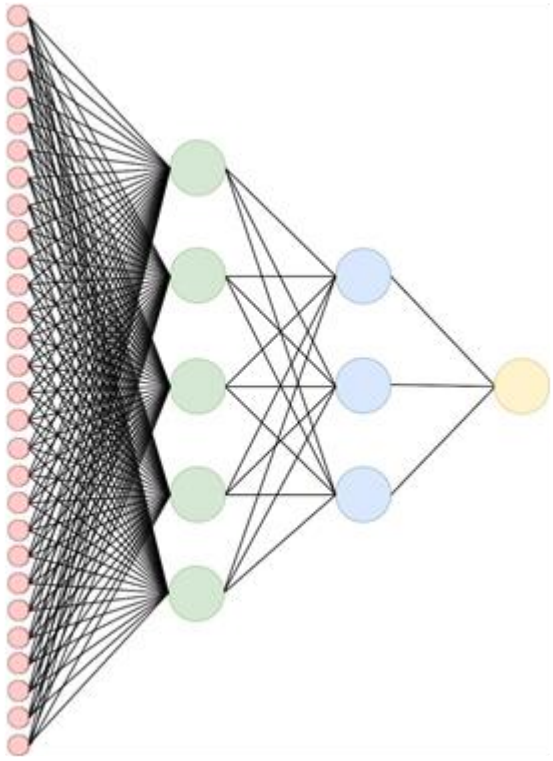

Fig. 3.4 Artificial Neural Network System Architecture

Fig. 3.4 shows the Architecture of the network. The first layer includes the Twenty Eight (28) inputs which is the sum of 1's and 0's per column and row and also responsible for receiving data. These inputs are normalized, produced by the activation function (TANSIG) used in this project.Two hidden layers were present having Five (5) neurons and Three (3) neurons respectively which are responsible for extracting associated patterns. Last layer is the output layer, responsible for producing the correct "Hangul" character needed which is he final network output.

IV. SIMULATION

Fig. 4.1 Neural Network Result

Fig. 4.1 shows the neural network training made for the Hangul Character Recognition based on the ANN architecture in Fig. 3.4

The algorithm used for data division was random (dividerand), Levenberg-Marquadt (trainlm) was used for the training, and Mean Squared Error (mse) was used for the performance.

The training finished after 1000 iterations, with 0:21 seconds time. The performance was 7.34e-10 out of 0.00, the gradient's value was 0.000509 out of 1.00e-07, Mu was 1.00e-07 out of 1.00e+10, and validation checks was 0 out of 6.

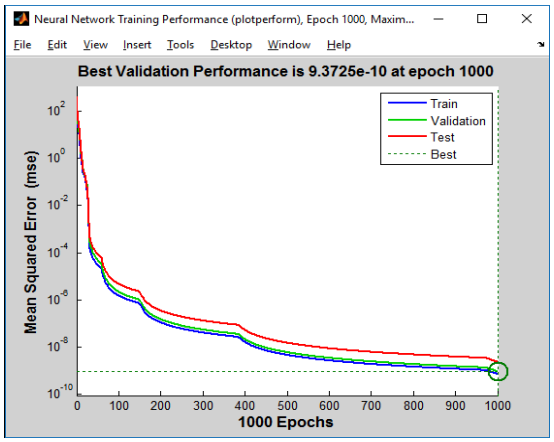

Fig. 4.2 Plot of the Neural Network Training Performance

Fig. 4.2 shows that the best validation performance made by the network trained was 9.3725e-10 at epoch 1000. The graph for train, validation and test performance was indirectly proportional to the number of epochs generated. As the epochs increases, the Mean Squared Error (mse) for train, validation, and test performance decreases, gradually coming close to the best possible performance.

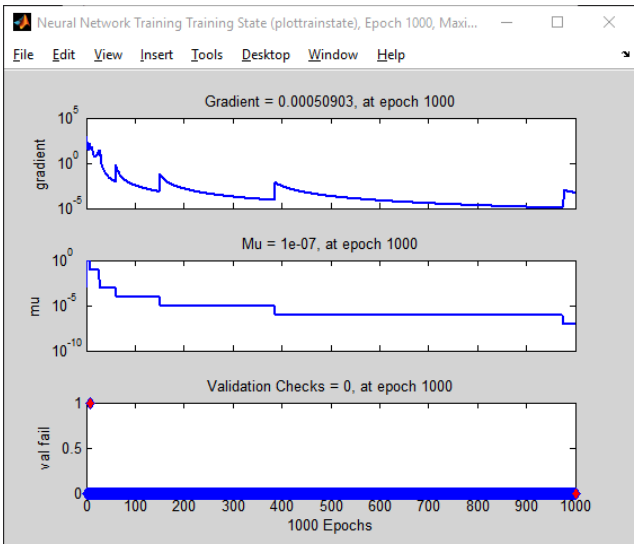

Fig. 4.3 Plot of the Training State of the Neural Network Training

Fig. 4.3 shows the graphs for gradient, mu, and validation checks, all at epoch 1000.

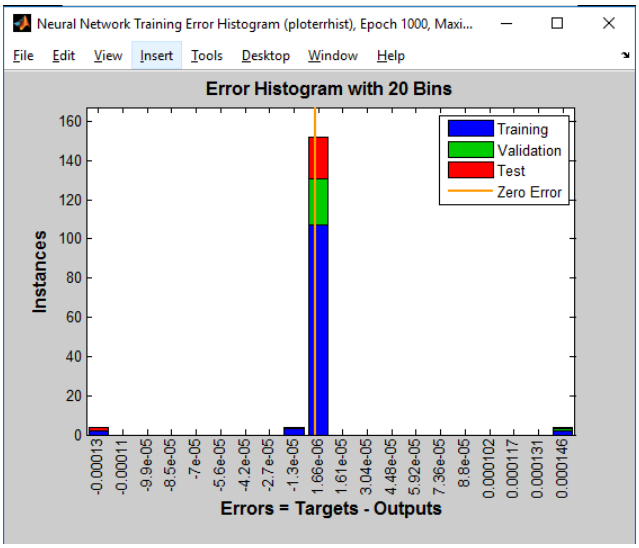

Fig. 4.4 Plot of the Error Histogram of the Network

Fig. 4.4 shows the graph of the Error Histogram with 20 bins. The training, validation, and test, and zero error's values all equaled to 1.66e-06 errors, defined by targets subtracted the outputs, but with different instances, 0-105 for training, 105-130 for validation, 130-155 for test, and 0-160 for zero error. There was a partial result of training and test errors, -0.00013 errors, at instance 0-5. An error for training performance of -1.3e-05 error at instance 0-5 is also visible. A partial result of training and validation error also occurred at instance 0-5 but with 0.000146 errors.

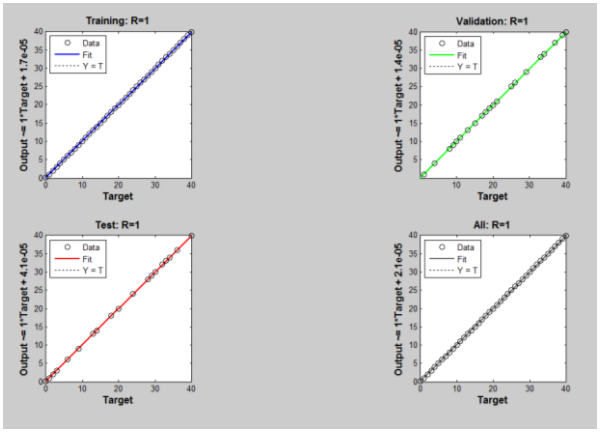

Fig. 4.5 Regression Result of the Neural Network

Fig. 4.5 shows the four graphs that shows the relationship between the target and output with different 1Target plus errors. It is shown that the target data and output data was very close to each other's graphs.

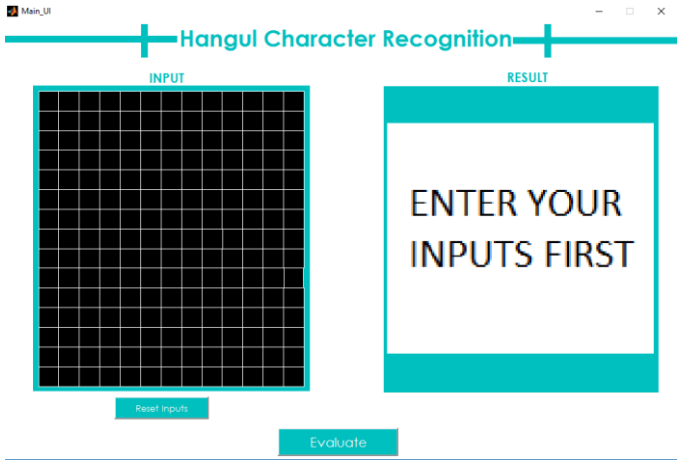

Fig. 4.6 Hangul Character Recognition main interface

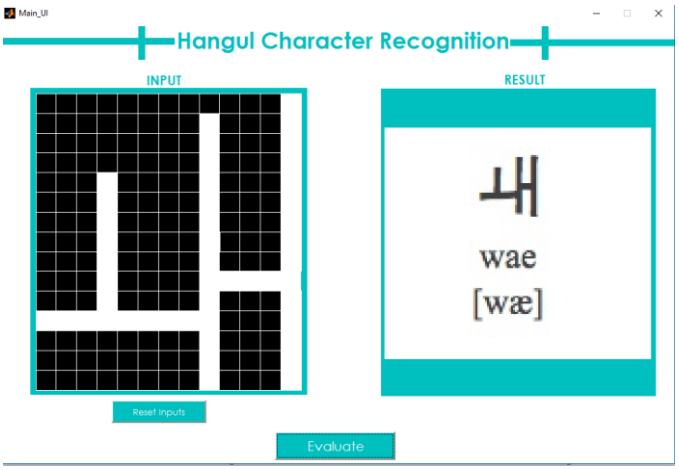

Fig. 4.7 Interface when a valid Hangul is evaluated

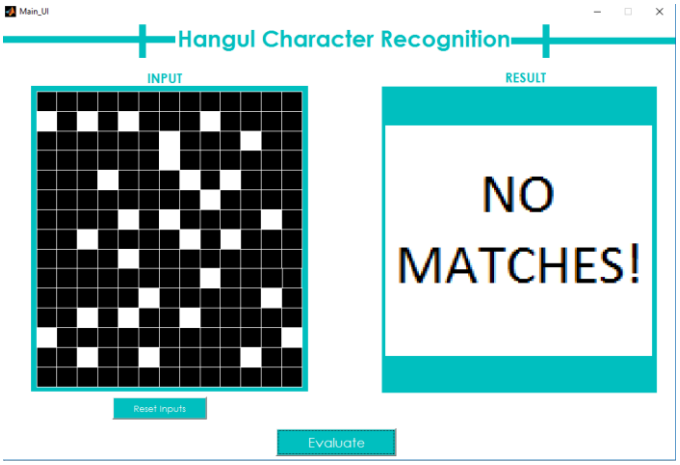

Fig. 4.8 Interface when an invalid Hangul is evaluated

Figures 4.6, 4.7 and 4.8 shows the Main Graphical User Interface of the Hangul Character Recognition that was created using MATLAB. It features a 15x13 grid where users can input a Hangul Character, a button to reset the input, an evaluate button in which it simulates the network and tries to find a match with the input and the Output picture, in where the matching Hangul Character is shown or a No match prompt is shown.

V. Conclusion

In this paper, we have considered using Artificial Neural Network as a tool in Hangul characters recognition. We have applied and trained the neural network to recognize different Hangul characters. The training was proven to be powerful as the tool was able to recognize and classify the characters. In conclusion, we can say that Artificial Neural Network is a good and useful method when it comes to character recognition. In our fast-pacing technology, powerful tools like ANN are needed to effectively and efficiently analyze data visually.

VI. Recommendation

The following recommendations are offered for related research in the field of Character Recognition.

1) Given that this study technologically and literarily helps in the field of character recognition, it may be advantageous to further use an image processing method instead of keying in the Hangul characters in the program manually.

2) With the current progress nature of technology, and global interactions among different countries, it is recommended to expand the scope and limitations of this project, making it available also for the other languages or even dialects to be easily recognized digitally.

VII. Curriculum Vitae

Debrelie O. Cruz is a 3rd year student of Bachelor of Science in Computer Science at Polytechnic University of the Philippines - Manila. Knowledgeable in C, Java, C#, PHP, MATLAB, HTML, CSS, and databases such as MySql and MS Access. Can work as a front-end and back-end developer but enjoys working in front-end. Computer skills also include Microsoft Office, Adobe Photoshop, InDesign, Soundbooth, After Effects, and Adobe Premiere.

Shaira Yvonne A. Cruz is a 3rd year student of Bachelor of Science in Computer Science at Polytechnic University of the Philippines. A friendly individual that can work with any people. Likes to jog, play games, and surf the internet. Mostly do documentations but can also code using C, Java, Visual Basic C#, MATLAB, HTML, and PHP. Also have a background in databases and plan to be working with databases in the future.

Krystal Dhanilei C. Decembrada is a Bachelor of Science in Computer Science student at Polytechnic University of the Philippines - Manila. Knowledgeable in C, C#, Java, MATLAB and Web Development Mostly Back-end. Literate in Microsoft Office. Hobbies are eating and watching movies. Personal interests are music and movies.

Lucky R. Deogrades is a Bachelor of Science in Computer Science student at Polytechnic University of the Philippines - Manila. Knowledgeable in C, C#, Java, MATLAB and Web Development Mostly Front-end. Literate in Microsoft Office, Adobe Photoshop and Adobe InDesign. Hobbies are browsing in the internet and watching movies. Personal interests are music and movies.

Earl Richard P. Dizon is a 3rd year student of Bachelor of Science in Computer Science at Polytechnic University of the Philippines – Manila. Knowledgeable in C, C#, Java, PHP, MATLAB, HTML, CSS and in manipulating databases using MySQL, MS Access and SQL Server. Personal interests includes anime, movies and music.

Jerica Mae M. Fabia is a Bachelor of Science in Computer Science Student at Polytechnic University of the Philippines - Manila. Knowledgeable in C, C#, Java, PHP, MATLAB and Web Development Mostly Front-end and Back-end. Literate in Microsoft Office and Adobe Photoshop. Hobbies are browsing in the internet, reading and watching movies. Personal interest are music and current events.

Joanna Marion S. Flores is a 3rd year student of Bachelor of Science in Computer Science at Polytechnic University of the Philippines - Manila. Knowledgeable in C, C#, Java, MATLAB and HTML. Literate in Microsoft Office. Hobbies are browsing in the internet and watching movies. Personal interests are music and movies.

Mark Daniel L. Marmeto is a 3rd year student of Bachelor of Science in Computer Science at Polytechnic University of the Philippines - Manila. Knowledgeable in C, C#, Java, MATLAB and HTML. Can work as a font-end and back-end developer. Literate in Microsoft Office. Hobbies are browsing in the internet and playing the drums. Personal interests are music and movies.

Caryl Andre D. Martin is a Bachelor of Science in Computer Science student at Polytechnic University of the Philippines - Manila. Knowledgeable in C, C#, Java, MATLAB and Web Development Mostly Back-end. Literate in Microsoft Office. Hobbies are browsing in the internet, reading and watching movies. Personal interests are music and current events.

Carl Joshua R. San Diego is a Bachelor of Science in Computer Science Student at the Polytechnic University of the Philippines - Manila. He has good knowledge in using C, C#, Java, MATLAB, HTML and in manipulating databases using MySQL and MS Access. Can work as both front-end and back-end developer. He likes watching American TV Series and discussing it with his friends.

Jonathan S. Tolentino is a Bachelor of Computer Science Student at Polytechnic University of the Philippines, Mabini Campus. Can program using C/C++, C#, Java, PHP, CSS, HTML, MATLAB, Photoshop, Adobe Illustrator. Hobbies are drawing, watching movies. Personal interests are music and 3d modeling.

REFERENCES

[1] M. A. Nielsen, "Neural Networks and Deep Learning," Determination Press, January 2016. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap1.html. [Accessed 5 October 2016].

[2] Wikipedia, "Pattern Recognition," Wikimedia Foundation, Inc., [Online]. Available: https://en.wikipedia.org/wiki/Pattern_recognition. [Accessed 5 October 2016].

[3] Giakos, "7-giakos-formatted.pdf," [Online]. Available: http://crema.di.unimi.it/~fscotti/nn/7-giakos-formatted.pdf. [Accessed 5 October 2016].K. Elissa, “Title of paper if known,” unpublished.

[4] ” OPTICAL CHARACTER RECOGNITION: AN ENCOMPASSING REVIEW”, International Journal of Research in Engineering and Technology, Nikhil Pai, Vijaykumar S. Kolkure, Bharatratna Indira Gandhi College of Engineering, Affiliated to Solapur University, Solapur, Maharashtra, India, 2015. [Accessed 5 October 2016].

[5] ”CHARACTER RECOGNITION”, Ziga Zadnik, Fakulteta za strojništvo. [Accessed 5 October 2016].

[6] ”Using Neural Networks to Create an Adaptive Character Recognition System”, Alexander J. Faaborg, Cornell University, Ithaca NY, 2012. [Accessed 5 October 2016].