Static Hand Sign Recognition System with Som-Hebb Classifier Implemented in FPGA

Anna Alex¹, Lijesh L²
M.Tech Scholar in VLSI and Embedded Systems¹, Associate Professor²
Department of ECE
Musaliar College of Engineering and Technology, Kerala, India

Abstract:
Gesture recognition is useful means of communication that is multifaceted in a number of ways. A hybrid network consisting of self-organizing map (SOM) and Hebbian network is used for hand sign recognition. The input posture images are preprocessed to generate a feature vector. The posture images are mapped to a lower-dimensional map of neurons in the SOM. The Hebbian network is a single-layer feedforward neural network trained with a Hebbian learning algorithm to identify categories. The recognition algorithm is robust to the change in location of hand signs, but it is not immune to rotation or scaling. Its robustness to rotation and scaling was improved by adding perturbation to the training data for the SOM-Hebb classifier. In addition, neuron culling is proposed to improve performance. The whole system is implemented on a field programmable gate array (FPGA) employing novel video processing architecture. The system is designed to recognize 24 American Sign Language (ASL) hand signs, and the system coded in VHDL is stimulated using Matlab and Modelsim.

Key Words: self-organizing map, Hebb learning, hand sign recognition, FPGA.

I. INTRODUCTION
Hand gestures provide an attractive alternative to the cumbersome interface devices used for human-computer interaction (HCI). Thus, integrating the use of hands in HCI would be of great benefit to users. Smart environments have recently become popular to improve our quality of life. Gesture recognition capabilities implemented in embedded systems are very beneficial in such environments for efficient HCI. Real time processing is an essential feature to use hand signs for HCI. The implementation of FPGAs has raised the possibility of achieving portable systems that can recognize hand gestures without bulky PCs while decreasing the response time due to their computing power. Hand gestures are categorized into hand postures and dynamic hand gestures. Hand postures are static hand poses without any movements. While, hand gestures are defined as dynamic movements, which are a sequence of hand postures. This paper focuses on hand posture recognition, and proposes a new real-time system of hand sign recognition that integrates all processing tasks in a single FPGA. The proposed system is based on a hybrid network. The classifier based on self-organizing map (SOM) is one of the most effective classifiers that can be used for efficient recognition. The SOM maps the given high dimensional input to a lower-dimensional map of neurons. The mapping done by the SOM is interpreted by a single layer feedforward neural network that is trained with the hebbian learning algorithm. This hybrid network used to classify vectors is called SOM-Hebb classifier. The number of neurons for the SOM-Hebb classifier to yield the best performance depends on the number of classes. Thus the culling of ineffective neurons is proposed in this work to optimize the number of neurons. Ineffective neurons are disabled so that the number of active neurons is adjusted to the number of hand posture classes. In addition, we also improved the system’s robustness to rotation and scaling by adding perturbation to the data to train the SOM-Hebb vector classifier. The proposed work focuses on recognizing 24 hand signs of American Sign Language. ASL is the language used by auditory handicapped people in the United States of America. In ASL, 26 gestures are used to communicate the 26 letters of alphabet. Among these, 2 of them are dynamic and this project is aimed to recognize 24 static hand gestures. The whole system can easily be accommodated in a single FPGA.

II. RELATED WORK
Now a day, gesture recognition is a growing field of research. There are many methods available for gesture recognition. Drawbacks of these methods are circuit complexity, noise, delay, power consumption etc. Existing methods for gesture recognition are a) wireless data glove method (circuit complexity, delay) b) vision based gesture recognition (power consumption, accuracy), c) GNG algorithm based gesture recognition (time consuming, accuracy, power consumption). Due to these disadvantages of traditional methods, we introduce this classifier based gesture recognition system.

III. PROPOSED METHODOLOGY
The proposed recognition system consists of three basic stages: pre-processing, feature extraction and classification. The input image of P x P pixels in RGB color format is preprocessed to generate a feature vector.
Fig 1. Gestures in American Sign Language

The pre-processing includes binary quantization, horizontal and vertical projection histogram calculations and two discrete fourier transforms (DFTs). The feature vector extraction is done from the magnitude spectrum obtained from DFT and this extracted D-dimensional vector is fed to the SOM-Hebb classifier. The classifier finally identifies the hand posture class.

3.1 BINARY QUANTIZATION
The system requires users to wear a white and red glove to remove the background image and segment the hand region. The segmentation of the hand region is done by detecting the red region and each input colour pixel is quantized to a binary value.

\[ I(x, y) = g(\text{Red}(x, y), \text{Green}(x, y) + \text{Blue}(x, y)) \cdot g(\text{Red}(x, y), \rho) \]

where \( I(x, y) \) is the binary pixel value, and \( \text{Red}(x, y) \), \( \text{Green}(x, y) \), and \( \text{Blue}(x, y) \) are the color component values of a pixel at the \((x, y)\) coordinates. The \( \rho \) is a threshold parameter and \( g(\cdot) \) is a threshold function.

\[ g(x, \rho) = \begin{cases} 1 & \text{if } x \geq \rho \\ 0 & \text{otherwise} \end{cases} \]

3.2 HORIZONTAL AND VERTICAL PROJECTION HISTOGRAM
The projection is defined as an operation that maps binary image into a one-dimensional array called histogram. The sum of the pixel values along a particular direction gives the histogram value. Horizontal projection histogram \( P_H(y) \) and vertical projection histogram \( P_V(x) \) are defined by,

\[ P_H(y) = \sum_{x=0}^{P-1} I(x, y) \]
\[ P_V(x) = \sum_{y=0}^{P-1} I(x, y) \]

3.3 DISCRETE FOURIER TRANSFORM
At the final stage of pre-processing, two DFTs are used to calculate the magnitude spectra \( F_H(n) \) and \( F_V(n) \) of \( P_H(y) \) and \( P_V(x) \).

\[ A(k) = \sum_{n=0}^{P-1} x(n) \cdot \cos\left(\frac{2\pi n k}{p}\right) \\
B(k) = \sum_{n=0}^{P-1} x(n) \cdot \sin\left(\frac{2\pi n k}{p}\right). \]

The magnitude spectrum is given by,

\[ X(k) = \sqrt{A(k)^2 + B(k)^2}. \]

3.4 FEATURE EXTRACTION
The DFT result is used as the feature vector. Each vector element \( \xi \) the classifier network is defined as

\[ \xi = \begin{cases} F(i) & 0 \leq i < D/2 \\ F(i - D/2) & D/2 \leq i < D \end{cases} \]

The magnitude spectra, shown in figure 2, \( F_H(n) \) and \( F_V(n) \) of the same hand posture images placed in different locations are identical because they lack phase information for the hand posture’s position.

Fig 2. Flow Diagram of hand sign recognition system

(a) Binary image (b) Horizontal projection (c) Vertical projection (d) Magnitude spectrum \( F_H(n) \) (e) Magnitude spectrum \( F_V(n) \) (f) Same hand sign in different locations yields identical spectrum.
3.5 SOM-HEBB CLASSIFIER

Fig. 4 outlines the SOM-Hebb vector classifier that is used as the classifier network in Fig. 2. The hybrid network consists of SOM, and a single layer feedforward neural network that is trained with the Hebbian learning algorithm. The classifier reads the D-dimensional vectors from preprocessing, and it classifies them into H classes.

3.5.1. Self-organizing map

The SOM consists of $K = M \times M$ neurons, each of which contains a D-dimensional vector, $\mathbf{m}_i$ called the weight vector.

$$\mathbf{m}_i = \{\mu_0, \mu_1, \ldots, \mu_{D-1}\} \in \mathbb{R}^D.$$ 

The operation of the SOM is divided into two phases. The weight map is trained with a set of input vectors in the learning phase. After that, the map is used in the recall phase. During the learning phase, input vectors are given to the SOM in multiple iterations. The distances to all weight vectors are calculated for each input vector, and then a winner neuron is determined. The weight vector of the winner neuron has the shortest distance to the input vector. The Manhattan distance $d_i$ is given by,

$$d_i = \sum_{j=0}^{D-1} |\xi_j - \mu_{ij}|.$$ 

After the winner neuron is determined, vectors of the neuron and its neighborhood neurons are updated with the following equation so that they are closer to the input vector.

$$\mathbf{m}_i(t + 1) = \mathbf{m}_i(t) + h_{ci} \cdot d_i.$$ 

Neighborhood function $h_{ci}$ is defined as

$$h_{ci} = \alpha(t) \exp \left( -\frac{||\mathbf{r}_c - \mathbf{r}_i||^2}{2\sigma^2(t)} \right).$$ 

Where $\mathbf{r}_c$ and $\mathbf{r}_i$ correspond to the location vectors of the winner neuron-$c$ and neuron-$i$. The neighborhood function provides a topology-preserving nature, i.e., two vectors that are neighbors in the input space will also be represented close to each other on the map. The topology-preserving nature is one of the most important features of SOM.

3.5.2. Hebbian learning network

Winner neuron information is taken from the SOM. The Hebbian learning network performs the category acquisition. Neurons representing a particular class are selected by the hebbian learning rule and they are grouped together. The vector’s class is indicated by the teaching signals $\tau_0, \tau_1, \tau_{H-1}$ and it is fed to the network. Then, the winner neuron is associated with the corresponding class if a strong correlation is found between them. The strong correlation is detected as follows; simultaneous activations of the neurons $i$ and $j$ are counted during the training phase, and if the count exceeds a threshold $T_{Hebb}$, then neuron $i$ is connected to class $j$.

1. SYSTEM PERFORMANCE

Neuron culling and additive perturbation to the training data are done in this work to improve the performance of the system.

4.1 NEURON CULLING

For the SOM-Hebb network to work properly, the SOM must have a sufficient number of neurons. However, the increase in neurons did not always improve the accuracy of recognition. A larger size SOM yields lower recognition rates in some cases. As explained in the previous section, the Hebbian network finds the association between neurons and hand posture classes. However, some neurons may have no connections to any hand posture class and the selection of these neurons as the winner causes false recognition.

If there are too many neurons against the number of classes in the SOM, more ineffective neurons that have no connections tend to be formed. Such neurons are culled after training to avoid false recognition. Ineffective neurons are searched during culling, and their weight vector elements are set to zero so that they do not become winners. If an input vector that would have made a culled neuron a winner is presented, the nearest effective neuron becomes a winner, and its corresponding class is determined as the recognition result. The substitute neuron for the culled one is greatly expected to yield the correct class due to the topology-preserving nature of the SOM.

4.2 ADDITIVE PERTUBATION TO TRAINING DATA

The proposed system is robust to position change in the image, but essentially it is not immune to rotation or scaling. Rotation and scaling invariant features require multiple access to each pixel; hence, the use of frame memory is inevitable. Frame memories are usually large, which increases the cost of systems. Our main goal was to implement a hardware gesture recognition system with minimum hardware resources. Therefore, we avoided the use of memory hungry algorithms and instead, perturbed training data for the SOM-Hebb classifier in rotation and scaling to improve its robustness to both of these. The input vector cluster of each class is enlarged by adding perturbation to the data to include these scaled and rotated hand posture images, and they are all categorized as a single class. As a result, the system is expected to be more robust to rotation and scaling.

IV. SIMULATION RESULT

The simulation of this work is done using Matlab and Modelsim software and the results obtained are shown in the figure 5.
Fig 5. Simulation results (a) Input image (b) Extraction of red part (c) Training of SOM-Hebb classifier (d) Testing of classifier (e) Output image and its class.

V. CONCLUSION

A system of hardware posture recognition based on a SOM and Hebb hybrid vector classifier was proposed in this work. As the feature vectors used in the proposed system were invariant to location changes in input images, the recognition was robust to location changes in hand signs. This paper proposed new learning method for SOM-Hebb classifier, which employs both neuron culling and perturbed training data. The proposed hand sign recognition system could carry out recognition at a speed of 60 fps and 60 recognitions per second with a recognition accuracy of 97.1%.

VI. REFERENCES


