EMBEDDED SYSTEMS REPORT

A SELF-CONFIGURABLE SYSTOLIC ARCHITECTURE FOR FACE RECOGNITION SYSTEM BASED ON PRINCIPAL COMPONENT NEURAL NETWORK

November 27, 2016

Amit Manchanda	Animesh Guupta	Tanvi Sharma
14116013	14116014	14116070

Contents

Abstract 3
Introduction 3
Implementation 5
    Approach 5
    Code implementation 6
Problems faced 7
Results 7
References 8

List of Figures

1 PCNN 4
2 PE structure and control flow diagram 5
3 Pre-processing image 6
4 Xilinx ISE testbench output 8

ABSTRACT

This report is based on a research paper titled "A Self-configurable Systolic Architecture for Face Recognition System based on Principal Component Neural Network" by N.Sudha, A.R.Mohan and P.K. Meher. An efficient self-configurable systolic architecture is proposed in this paper for very large scale integration implementation of a face recognition system. We have tried to build a face recognition system on XILINX Spartan-3E using the same approach.

INTRODUCTION

In the next few subsections, the principal concepts are discussed on which the paper is based.

Systolic architecture

In parallel computer architectures, a systolic array is a homogeneous network of tightly coupled data processing units (DPUs) called cells or nodes. It is based on the principle that by replacing a single PE by a regular array of PEs, and by carefully orchestrating the flow of data between the PEs, high throughput can be achieved without increasing memory bandwidth requirements.

Conventional Approaches

Many approaches can be used for image processing. Some of them are given below with a short description.

1. PCA: Principal Component Analysis. It is the best approach using eigen-based recognition which is performed on covariance matrix. It is very suitable for security-related applications since only few samples per subject are required for analysis.

2. LDA: Linear discriminant analysis. It is performed on a separation matrix (obtained from the within-class and between-class scatter matrices). It requires more number of samples.

3. LPP: Locality preserving projections. It gives superior results for frontal images.

4. Bayesian analysis: It is a probabilistic approach based on image differences between the data set elements. It has the disadvantage of being computationally expensive.

Neural networks

It is a computer system modelled on the human brain and nervous system. They can be types like RNN (radial based) or CNN (complex) and can be easily used for building recognition systems. Generally, they are based on complex learning algorithms and non linear neurons which is inefficient for hardware implementation.
Here, we have used a single-layer feed forward network of neurons (PCNN, Principal Component Neural Network) based on the Hebbian learning algorithm. In this rule, the weights between two neurons (x and y) is strengthened when the neurons on input and output have highly correlated outputs.

Figure 1: PCNN

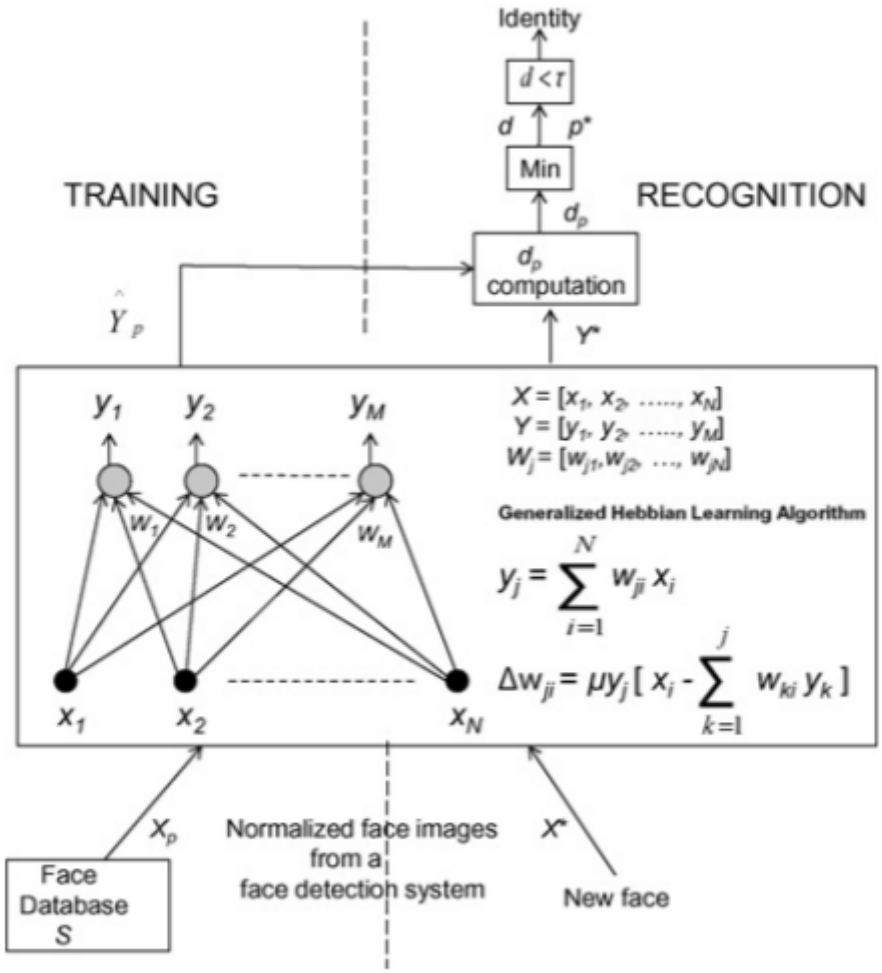


Fig. 1. PCNN-based face recognition system.

IMPLEMENTATION

Approach

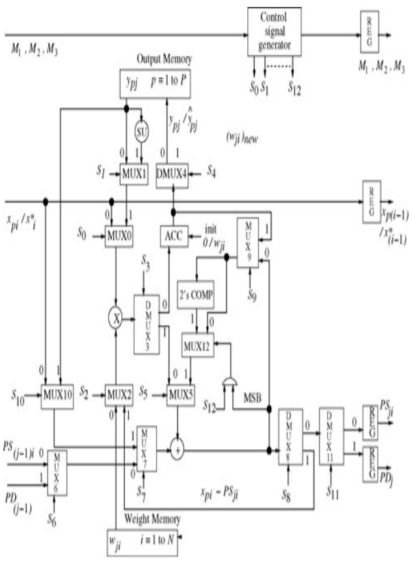
As proposed in the paper, PCNN approach includes two stages, namely training and recognition. Training in turn includes

- T-1 : Computation of Neuron Outputs for Training Faces
- T-2 : Updating Weights
- T-3 : Computation of Projections of Training Faces onto Eigenfaces

While recognition includes

- R-1 : Computation of Neuron Outputs for a New Face
- R-2 :Computation of Difference Between Projections

Figure 2: PE structure and control flow diagram



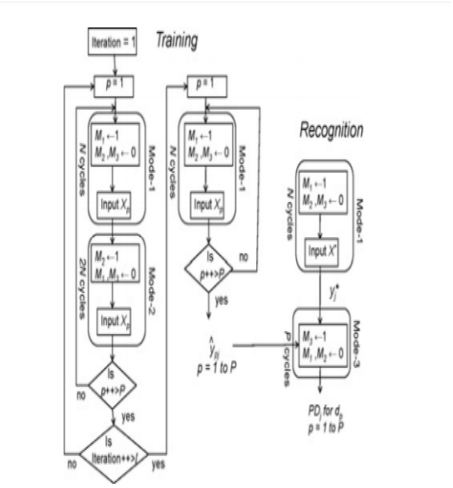
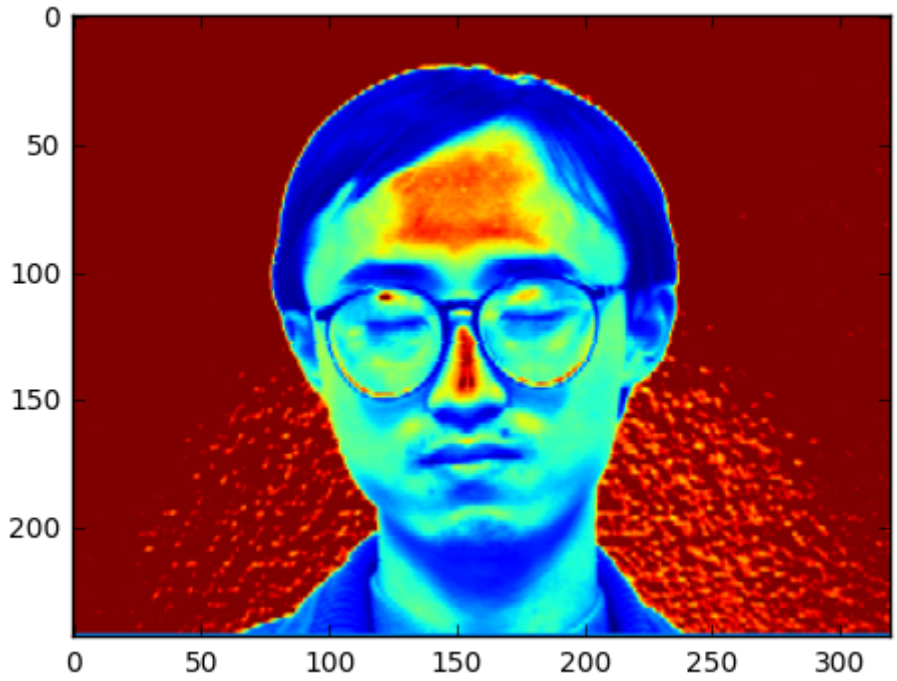


Fig. 8. Combined control flow diagram for the jth PE in training and recognition phases.

These computations are further classified into 3 modes for simplification in implementation:

1. Mode-1: computation of neuron outputs (Corresponding to T-1,T-3,R-1)
2. Mode-2: updating weights (Corresponding to T-2)

Figure 3: Pre-processing image



3. Mode-2: computation of dp (Corresponding to R-2)

The change from one mode to other is based on the control unit flow. This is very well illustrated in the given flow charts.

Code implementation

First of all, the image data set taken from Yale university is pre-processed to reduce the pixel size and get the input grayscale values starting from left upper corner to bottom right corner. It involved extraction of face from the image and normalising the pixel values to be used in training and recognition phase.

Training Phase

The training part is done on computer in python language because of performance limitations in FPGA board.* It involved selection of proper hyperparameters and training of the model 150 images (Yale Dataset). Generalised Hebbian algorithm was used for updation of weights and for the calculation of Principal Components and the projection along these Principal Components.

Recognition Phase

The recognition part is implemented on FPGA and is developed in VHDL.*Involved providing an image for testing and the output was the index of the closest image from our dataset.

*The code can be taken from the Github link given in references.

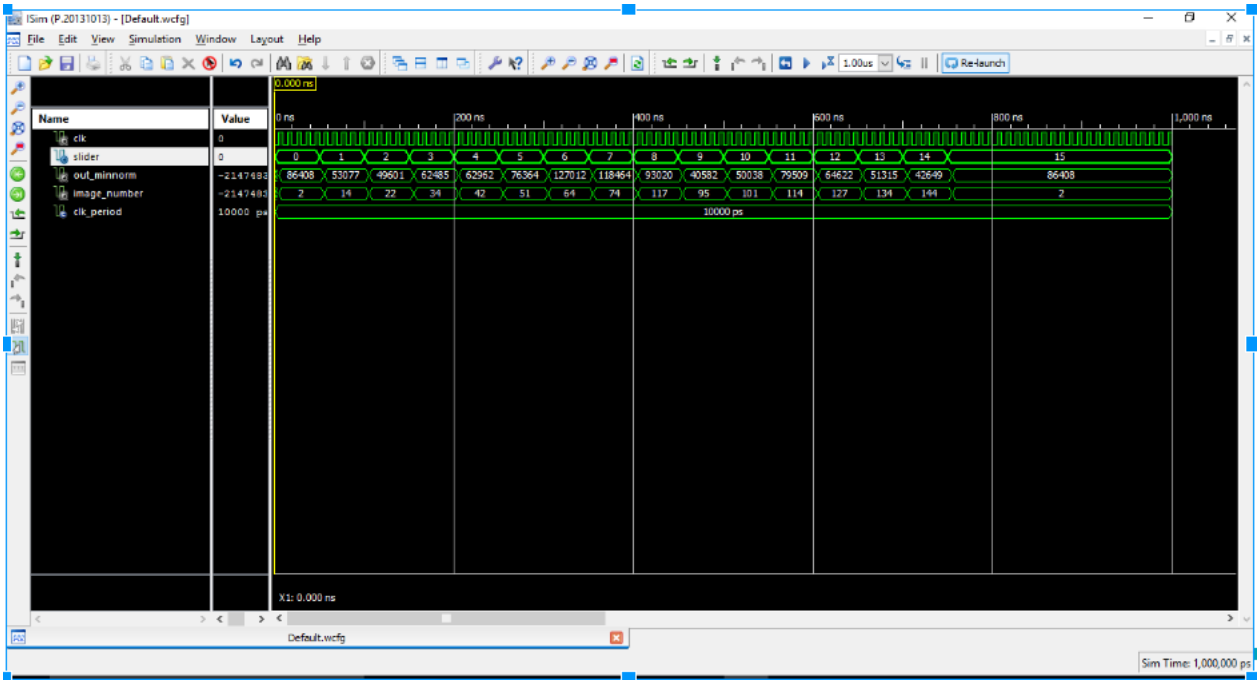
PROBLEMS FACED

- Extraction of face from the image: We used opencv library in python to overcome it.
- Use of floating point in VHDL: We had to multiply the weights by 1000, input normalised pixels by 20 and output principal components by 20000 in order to reduce the dependency of floating point number in our project.
- Large dataset, so large collection of weights
- Infinite long synthesis time and Over-utilization of FPGA resources: Due to the large image size and the for loops required to predict the image, there was over utilization of FPGA resources and long synthesis time. We had to reduce the image size from 32X32 to 8X8 in order to run the algorithm in VHDL

RESULTS

We were able to identify 14 out of 15 different faces which were not used while training of datasets, so were new to the algorithm. If a threshold is added 10 faces were identified correctly. We used python to verify the results.

Figure 4: Xilinx ISE testbench output



REFERENCES

1. "A Self-Configurable Systolic architecture for face recognition system based on principal component neural network - IEEE Xplore document," 2016. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5739514.

2. amitmanchanda 1995, "Amitmanchanda1995/Face-Recognition-using-PCNN," GitHub, 2016. [Online]. Available: https://github.com/amitmanchanda1995/Face-Recognition-using-PCNN.

3. T. D. Sanger, âĂIJOptimal unsupervised learning in a single layer feedforward neural network,âĂİ Neural Netw., vol. 2, pp. 459âĂŞ473, 1989

4. Yale Database [Online]. Available: http://cvc.yale.edu/projects/yalefaces/yalefaces.html

5. M. A. Turk and A. P. Pentland, âĂIJEigenfaces for recognition,âĂİ J. Cognitive Neurosci., vol. 3, no. 1, pp. 71âĂŞ86, 1991