A Literature survey for Object Recognition using Neural Networks in FPGA

T.DaslinPonSanthosh, P.KANNAN
PG Scholar Department of ECE, PET Engineering College, India
Professor Department of ECE, PET Engineering College, India

Abstract—The Recognition of Objects is considered as difficult one in Image Processing. It includes recognition of face, iris, cars, airplanes, etc. For object recognition, many neural networks in FPGA have been proposed. But these neural network methods have high computational complexity with high computation time and also reduce the number of images recognized per second in object recognition. Hierarchical Model and X model (HMAX model) is a biologically inspired model which is used to improve above all complexities in old methods. This literature survey, discusses all the existing neural network methods in FPGA and their performance.

Keywords--- FPGA, HMAX model, Object Recognition.

1. INTRODUCTION

In Image Processing, Object Recognition plays a major role. This object recognition involves the recognition of face, iris, fingerprint, vehicle, etc and also it involves all kind of image and video tracking. The object recognition is done by using machines also. It involves hardware implementation and it has interaction with all kind of environment. The Field Programmable Gate Array (FPGA) consists of logic components which are used to perform complex mathematical functions. It is suitable for the implementation of matrix algorithms. FPGA provide the advantage of the parallelism, low cost, and low power consumption. The object recognition need to face the following problems: the object appearance must be changed depending on its position, scaling and orientations. This kind of object recognition is done by using cortex like mechanisms [2]. In this mechanism, it has the operation similar to human cortex and recognizes the object of different orientations. It needs only less number of training samples to recognize the object. But it is suitable only for the feed forward methods. A reconfigurable FPGA is obtained by the multi layer feed forward neural network [7]. It has a single largest layer which behaves like different layers. Therefore the resource required must be low. The computational complexity is increased by using a single layer for all operations. A scale and rotation invariant feature detection is done by Scale Invariant Feature Transform (SIFT). It is applied to Simultaneous Localization and Mapping (SLAM) problem [4]. Restricted Boltzmann Machine [5] is an unsupervised learning method and it is to provide high performance reconfigurable architecture. It is a divide and conquers method. It does not need any multipliers because it uses logic gates. It does not have any pipelined architecture.

2. LITERATURE REVIEW

MULTILAYER PERCEPTRON-BACK PROPAGATION (MLP-BP):

An MLP[9] network contains neurons structured in parallel layers, from inputs to outputs. The layers are numbered 0 to M and the neurons are numbered 1 to \( N \). When used in conjunction with the error backpropagation (BP) algorithm, each neuron contains two key arithmetic functions which perform forward and backward computations. The forward computation step uses internal weights associated with each neuron for calculating the neuron’s output. The forward computation is given as,

\[
o_k^{(s)} = f \left( H_k^{(s)} \right) = f \left( \sum_{j=1}^{K_{(s-1)}} w_{kj}^{(s)} o_j^{(s-1)} \right) + w_{ko}^{(s)}
\]

Where,

\( s=1,2,\ldots,M \) Network layers;
\( M \) Total number of layers;
\( N^{(s-1)} \) Number of neurons in layer (s-1);
\( o_k^{(s)} \) Output of the current neuron;
\( f \) Activation function
\( H_k^{(s)} \) Weighted input sum;
\( o_j^{(s-1)} \) Output of the jth neuron;
\( w_{kj}^{(s)} \) Synaptic weight;
\( w_{ko}^{(s)} \) Bias weight;
The backward computation step compares the network’s overall output to a target, computes an error gradient, and propagates the error through layers by adjusting each neuron’s weights to correct for it. Backward computation also involves the updation of all weights as,

\[ w^{(s)}_k(n + 1) = w^{(s)}_k(n) + \Delta w^{(s)}_k(n) \]  

However, when reviewing the mathematical properties of the relationship of outputs from inputs, a network with number of layers can be fit into a network with two layers, namely one hidden and one output layer, without loss of these properties. This comes about from the fact that each output of the network must be dependent on a nonlinear combination of any or all inputs. While a network with many hidden layers can learn complex nonlinear input–output relationships efficiently (with few neurons in each layer), the same relationships can be learned with a two-layer topology, though using more neurons in each layer.

By this property, the VHDL network can be constructed, and is designed to be a two-layer structure. MLP-BP networks should have a high precision. Range can be limited as long as inputs and outputs are normalized since weights should be limited for learning progress. The occupied area of an FPGA-based MLP-BP is proportional to the multiplier used.

3. VISUAL CORTEXLIKE MECHANISM:

This system follows the Feedforward path of object recognition in cortex that accounts for the first 100-200 milliseconds of processing in the ventral stream of primate visual cortex. The model itself attempts to summarize in a quantitative way a core of well-accepted facts about the ventral stream in the visual cortex[2]. First, visual processing is hierarchical, aiming to build invariance to position and scale first and then to viewpoint and other transformations. Second, along the hierarchy, the receptive fields of the neurons (i.e., the part of the visual field that could potentially elicit a response from the neuron) as well as the complexity of their optimal stimuli (i.e., the set of stimuli that elicit a response of the neuron) increases.

Third, the initial processing of information is Feedforward (for immediate recognition tasks, i.e., when the image presentation is rapid and there is no time for eye movements or shifts of attention). Fourth, Plasticity and learning probably occurs at all stages and certainly at the level of Infero Temporal (IT) Cortex and prefrontal cortex (PFC), the top-most layers of the hierarchy.

In its simplest form, the model consists of four layers of computational units, where simple S units alternate with complex C units. The S units combine their inputs with a bell-shaped tuning function to increase selectivity. The C units pool their inputs through a maximum (MAX) operation, thereby increasing invariance.

These Cortex like mechanism have the disadvantage of it is used only for the recognition of texture-based objects and shape-based objects.

4. LAYER-MULTIPLEXED FEEDFORWARD NEURAL NETWORK:

In order to implement a given NN of any size with minimum hardware, the concept of multiplexing the layers of NN [7] is used. Layer multiplexing is the implementation of the single largest layer (i.e., layer with maximum neurons) with each neuron having the maximum number of inputs and a control block to coordinate them. Here the largest layer has eight neurons and the largest number of inputs to a neuron in the network is eight. Hence, for the considered example, eight neurons with each neuron having eight inputs is identified as the single largest layer for implementation. The implemented single layer is multiplexed to execute the functions of all layers of the network. The layer multiplexing requires an additional control block to coordinate the computation of every layer of NN. Hence, realization of complete network with a single layer implementation leads to a considerable resource reduction of FPGA.

The largest layer with each neuron having the maximum number of inputs is implemented in this method. A start signal is given to the control block to initiate the operation of the network. The control block places simultaneously the correct set of inputs, weights, and biases to every neuron of the layer. The results of the neurons are computed parallel and sent to the layer control block to be provided as inputs to the next layer. Once the complete network is computed, the end of computation (EOC) signal is issued to latch the output of the NN.

Four different network architectures are chosen such that for all the networks the single largest layer contains five neurons and each neuron has eight inputs.
All those NNs have eight inputs and three outputs. The log sigmoid excitation function is used for the hidden layers and the linear excitation function is used for the output layer. The choice of networks is to illustrate the saving in resource, which increases as the number of layers increases.

The implementation includes the single largest layer and a control block. The layer control block coordinates the computation of the different layers by placing the appropriate inputs, weights, biases, and value of excitation function (from LUT) for each layer. For good recognition rate, it need more resource and high cost. It also increases the computational complexity.

5. SCALE INVARIANT FEATURE TRANSFORM:

The SIFT algorithm is used for scale end rotation invariant feature detection system[13]. The algorithm is divided into three main stages: the first one identifies pixel candidates to be a feature, the next one then applies a set of tests to verify whether the candidates are stable in relation to image transformations in order to accept or reject them. Having detected stable features, the final stage generates a descriptor/signature to be associated with the feature.

![Fig 1 Block Diagram used for detection in SIFT](image)

The above block diagram explains about the operations used to detect features in an image by six gaussian smoothed images. The efficient solution to detect features, is provided by SIFT algorithm. However it has high computational cost caused mainly by the Gaussian filter cascade and the keypoint detection with stability checks. A system configured with 3 octaves and 6 scales per octave based only on software takes 1.1s to extract 514 features from image of 320x240 pixels in an Intel Core 2 1.66GHz processor. SIFT consists of three blocks in hardware (DoG, OriMag, and Kp) and one in software (NIOS II). The DoG block receives as input a pixel stream from a camera and performs two operations, the Gaussian filter and the difference of Gaussian, whose results are then sent to the OriMag and Kp blocks, respectively. The OriMag computes the pixel orientation and gradient magnitude while the Kp block detects keypoints and realizes the stability checks to classify keypoints as features. Finally, the software block, based on the NIOS II soft-core processor, generates a descriptor for each feature detected by the Kp block based on the pixel orientation and gradient magnitude produced by the OriMag block.

The disadvantage of SIFT is, for 320*240 pixels image, it detect at a rate of 30 frames/sec only. (i.e) more time consumption as 33ms.

6. RESTRICTED BOLTZMANN MACHINE (RBM):

A RBM[5] is a generative, stochastic neural network architecture consisting of two layers of nodes representing visible and hidden variables. This work focuses on the family of RBMs where both the visible and hidden variables have binary states. There are weighted connections between every node in opposite layers, and no connections between any nodes in the same layer. Biases are represented by setting the first node.

The following notation system will be used: vi and hj are the binary states of the ith and jth node, where i= {1……I} and j= {1……J}, in the visible and hidden layer, respectively; wij is the connection weight between the ith and jth node. Alternating Gibbs Sampling (AGS) and Contrastive- Divergence learning (CD) has been found to be an effective process to determine the node states and update the weight parameters [8], respectively.

AGS is divided into two phases, generate and reconstruct phases. During the generate phase, the visible layer is clamped and used to determine the node states of the hidden layer. In the reconstruction phase, the opposite occurs by clamping the hidden layer and reconstructing the visible nodes. Here partition of RBM is not done properly. Uses only binary valued node. It doesn’t have pipelined architecture.

7. HMAX MODEL:

Hierarchical model and X(HMAX) model[1] is used for biologically inspired model. HMAX is traditionally known to achieve high accuracy in visual object recognition tasks at the expense of significant computational complexity. It is best to implement in
Field Programmable Gate Array (FPGA) than the above methods. It has two main stages, each consisting of a simple and complex sub stage. It is given as Simple-1 (S1), Complex-1 (C1), Simple-2 (S2), and Complex-2 (C2).

8. S1 Filters:
The S1 layer consists of directionally selective Gabor receptive fields. Here the cells are implemented at four different orientations (0°, 45°, 90° and 135°). Because of symmetry, we need not compute cells at orientations at or above 180°. Each orientation is implemented at 16 different scales and at every location in the image where full support is available. The equations for the filters are the product of a cosine function and a Gaussian window.

\[ F_{\theta}(x, y) = e^{\left(-2 \pi^2 \frac{x^2 + y^2}{2 \sigma^2}\right)} \times \cos\left(\frac{2\pi x}{\lambda} x_0\right) \]

where,

\[ x_0 = x \cos \theta + y \sin \theta \]
\[ y_0 = -x \sin \theta + y \cos \theta \]

Here, \( \lambda \) determines the spatial frequency at the filter’s peak response, \( \sigma \) specifies the radius of the Gaussian window, and \( \gamma \) squeezes or stretches the Gaussian window in the \( y_0 \)-direction to create an elliptical window. For the 0° and 90° cases, we can easily rewrite this equation as the product of two separate functions as shown in (2). The 45° and 135° terms are not separable unless we change the Gaussian window to an isotropic function by specifying \( \gamma = 1 \). By doing this, we arrive at the equations for the 45° and 135° filters shown below

\[ F_{45}(x, y) = E(x, y) \times E(x, y) \]
\[ F_{135}(x, y) = E(x, y) \times E(x, y) \]

\[ G(x, y) = e^{-\left(\frac{x^2}{2\sigma^2}\right)} \cos\left(\frac{2\pi x}{\lambda}\right) \]
\[ O(x, y) = e^{-\left(\frac{x^2}{2\sigma^2}\right)} \sin\left(\frac{2\pi x}{\lambda}\right) \]

Here, \((x, y)\) is the location of the kernel value within the filter, \(O(x, y)\) is an odd Gabor filter, \(E(x, y)\) is an even Gabor filter, and \(G(x, y)\) is a pure Gaussian filter. In this model, the filters are modeled in separable manner (i.e., it is implemented as two passes of a 1-D filter than one pass of a 2-D filter.

\[ MAC_{\text{original}} = 4 \times \sum_{j=1}^{15} [\emptyset(j)^2] = 36416 \]

\[ MAC_{\text{separable}} = 4 \times \sum_{j=1}^{15} [2 \times \emptyset(j)^2] = 2816 \]

where \( \emptyset(j) \) is the side length of filter \( j \). Using separable filters reduces the number of required MAC from 36416 down to 2816.

C1:
The C1 layer requires finding the maximum S1 response over a region of 2_ \times 2_ and subsampling every \_ pixels in both \( x \) and \( y \). It eliminates the need to store nonmaximal S1 results. The computation in C1 is performed on all four orientations in parallel. Each time C1 finishes computing the results for a size band, a flag is set that indicates to S2 that it can begin computation on that size band.

S2:
The data coming into S2 has already been reduced by taking the maximum across a local pool and subsampling in C1, the S2 layer is where most of the computation takes place. Since S2 patches are simply portions of previously computed C1 outputs, thenumber of bits required to store each patch coefficient is 16. The closeness of a patch to a C1 region is computed as the Euclidean distance between the patch and that region. It uses 640 multipliers and requires that 640 patch coefficients be read every clock cycle. Patch coefficients are stored in the FPGA’s internal block RAM since the bandwidth to external RAM would not allow such high data rates.

C2:
C2 simply consists of a running minimum for each S2 patch, computed by comparing new S2 results with the previously stored S2 minimum. This is performed for all 320 S2 patches of the current size simultaneously.
The above figure explains about the comparison between the Cortex like mechanism, HMAX using CPU, HMAX using FPGA. From the figure, we obtain that the object recognition by using FPGA have more accuracies in recognition of airplanes and motorbikes. It also have the overall accuracy of 47.2% at a speed of 5.3 ms. HMAX in FPGA is faster than HMAX CPU and provide a recognition rate of 190 images/sec.

9. CONCLUSION

In this paper, a brief literature survey for object recognition using neural networks is discussed elaborately. From this study it is concluded that all the neural network methods in FPGA like MLP-BP, RBM, spiking neural networks, and feed forward neural networks provide less recognition rate than HMAX model.

ACKNOWLEDGEMENT

Apart from my efforts, the success of any work depends on the support and guidelines of others. We take this opportunity to express my gratitude to the people who have been supported us in the successful completion of this work. We owe a sincere prayer to the LORD ALMIGHTY for his kind blessings without which this would not have been possible. We wish to take this opportunity to express my gratitude to all who have helped us directly or indirectly to complete this paper.

REFERENCES


