A Systolic Architecture for Object Recognition Based On Hmax Model

T. Daslin Pon Santhosh¹, P. Kannan²
¹PG Scholar, Department of ECE, PET Engineering College, India
²Professor, 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 have been proposed. But these neural network methods have high computational complexity which require more computation time. Hierarchical Model and X model (HMAX model) is a biologically based model which is used to improve all complexities in the object recognition. This HMAX model need less number of training examples and it is the state-of-the-art system in object recognition task. The basic HMAX model involves the operation of Gabor filter for object recognition. The recognition based on Gabor filter is done in MATLAB. The Gabor filter is used to provide the features of an image of different orientations. It provides independent variables. In the proposed system, the object recognition is done by means of Linear Discriminant Analysis (LDA). It is then implemented in Field Programmable Gate Array (FPGA) which has the advantage of less area, less required power and also provide less computational time.

Index Terms: Object recognition, Neural network, HMAX model, Gabor filter, LDA, FPGA.

I. Introduction

In Image Processing, Object Recognition plays a major role. This object recognition involves the recognition of face, leaves, fingerprint, vehicle, etc. and also it involves all kind of image and video tracking. This systolic architecture could be configured to partition the image and distribute the resulting sections to multiple sections of processing and the processed data provide high computation.

D. Le Ly and P. Chow [5] introduce the Restricted Boltzmann Machine in the year 2010. A RBM 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.

S. Himavathi, D. Anitha, and A. Muthuramalingam [7], introduce the concept of multiplexing the layers of Neural Networks, in order to implement a given NN of any size with minimum hardware. 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. 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 results of the neurons are computed parallel to 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.

A. Savich, M. Moussa, and S. Areibi, introduce a MLP network [9] contains neurons structured in parallel layers. The layers are numbered from 0 to M and the neurons are numbered from 1 to N. When it used with the error back propagation (BP) algorithm, each neuron contains two key arithmetic functions which perform forward and backward computations. In the forward computation, it uses internal weights associated with each neuron for calculating the neuron's output. 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. 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. 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.

In the paper, Fast Neuromimetic Object Recognition Using FPGA Outperforms GPU Implementations by Garrick Orchard, Jacob G. Martin, R. Jacob Vogelstein, and Ralph Etienne-Cummings, the performance of Gabor filter is obtained. But in that it need a very highly modified FPGA hardware for recognition. Else it provide more complexity and also take more time to recognize.

To overcome these, here the LDA method is implemented in FPGA instead of Gabor filter. Linear Discriminant analysis explicitly attemptsto model the difference between the classes of data. LDA is a powerful face recognition technique that overcome the limitation of Principle component analysis technique by applying the linear discriminant criterion. This criterion triestomaximizemetheratio of the between-classscatter matrix of the projected samples to the within-classscatter matrix of the projected samples. Linear discriminant group images of the same class and separates images of different classes of the images. It provide less computational complexity than Gabor filter when it implemented in FPGA. The implementation of FPGA provide less consumption of power also need less area for recognition.
II. System Model

2.1 Recognition based on Gabor filter:

2.1.1. Gabor filter

A Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. A Gabor filter is a linear filter used for edge detection in image processing which is named after Dennis Gabor. Gabor filter frequency and orientation representations are similar to those of human visual system, for texture representation and discrimination it has been found to be remarkably appropriate.

A sinusoidal plane wave has been modulating a 2D Gabor filter which is a Gaussian kernel function in the spatial domain. From one parent wavelet all filters can be generated by dilation and rotation, thus the Gabor filters are self-similar. The overall proposed methodology [1] is given in figure 2. From that figure, Gabor filter is of second block. It has four orientations. With four different orientations of Gabor filter, features of the image are extracting and are combined.

In the HMAX model of object recognition, input images (128 x 128 or 160 x 160 grey scale pixel images) are used and the features are created based on orientations $0^\circ$, $45^\circ$, $90^\circ$, and $135^\circ$ and sizes from 7 x 7 to 29 x 29 pixels. At each pixel of the input image, filters of each size and orientation are centered. By applying the four orientations and 16 different scales, the various features of the input image is created.

2.1.2. Maximum Response

In the next step, filter bands are defined, i.e., groups of Gabor filters of a certain size range (7 x 7 to 9 x 9 pixels; 11 x 11 to 15 x 15 pixels; 17 x 17 to 21 x 21 pixels; and 23 x 23 to 29 x 29 pixels). Within each filter band, a pooling range is defined (i.e.) the features having maximum match within it is taken. Else are rejected which provide simplicity and reduction in the space requirement. The maximum response of the features created is obtained and it is then compared with the dataset which is already created. The maximum response is obtained by setting the threshold value.

2.1.3. Recognition

The training dataset is already trained and stored. The test set image is given and training of that image produce some output arrays. The maximum of these output arrays are taken and then this maximum array of test image is compare with the threshold of the stored dataset. If this equals then it recognize the corresponding image. Else if the array value is not exactly equal to the threshold but somewhat related to the threshold means then also the corresponding image is recognized from the dataset.

2.1.4. Classification

Boosting classifier is better than SVM for the binary classification problem. We used the gentle boosting algorithm with weak learners consisting of tree classifiers, each with a maximum of three decision branches. We used 1280 weak learners in the classifier, each computed in series. For multiclass classification, a linear one-versus-all SVM classifier was chosen. This is a simple linear classifier, but is memory-intensive in its requirement for storing coefficients. In order to not restrict the FPGA implementation to only binary problems or only multiclass problems, the classifier was implemented separately on a host PC.

a. Caltech 101 Dataset

The binary task constituted discriminating the class in question (airplanes, cars, faces, leaves, or motorbikes) from the background class. In each case, half the images from the class in question and half images from the background class were used for training. The remaining images from both the class in question and the background class were used for testing. In each case, 10 trials were run.

b. Flickr Dataset:

The binary Minaret classification task was performed on a dataset containing 662 images of Minarets and 1332 background images. The Minaret (positive) images were obtained from Flicker by searching for “Minaret”, while the negative images were obtained by periodically downloading the most recently uploaded Flicker image.

c. Multiclass One-Versus-All

It uses 15 training examples per category. Testing was performed using 50 examples per category or as many images as remained if fewer than 50 were available. Each of the categories was weighted such that it contributed equally to the result. This is a 102-category problem including the background category. In this, it build N different binary classifiers (i.e.) it has two or more number of multipliers which are connected together to form a multiclass classification. The classification is done by means of taking the argument of maximum value of the created features if the input image. For the
ith classifier, let the positive examples be all the points in class i, and let the negative examples be all the points not in class i. Let \( f_i \) be the ith classifier. It is then classify with the following equation,

\[
f(x) = \arg \max_i f_i(x)
\]

where, \( f(x) \) is classified output of the image.

2.2 Recognition based on Linear Discriminant Analysis:

![Block diagram for face recognition based on LDA](image)

Figure 2: Block diagram for face recognition based on LDA

2.2.1 Algorithm used in LDA:

Step 1:

The LDA method need a training set composed of a relatively large group of subjects with diverse facial characteristics. The appropriate selection of the training set directly determines the validity of the final results. The database should contain several examples of face images for each subject in the training set and at least one example in the test set. These examples should represent different frontal views of subjects with minor variations in view angle. They should also include different facial expressions, different lighting and background conditions, and examples without glasses. It is assumed that all images are already normalized to \( m \times n \) arrays and that they contain only the face regions and not much of the subjects’ bodies.

Step 2:

For each image and subimage, start with the two-dimensional \( m \times n \) array of intensity values \( I(x,y) \). We construct the vector expansion \( \Phi(x,y) \). This vector corresponds to the initial representation of the face. Thus, the set of all faces in the feature space is treated as a high-dimensional vector space.

Step 3:

By defining all instances of the same person’s face as being in one class and the faces of different subjects as being in different classes, we set the training set and perform a cluster separation analysis in the feature space.

Also, having labeled all instances in the training set and having defined all the classes, we compute the within-class and between-class scatter matrices.

Now, within-class scatter matrix ‘\( S_w \)’ and the between-class scatter matrix ‘\( S_b \)’ are defined as follows:

\[
S_w = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (f_j^i - \mu_j)(f_j^i - \mu_j)^T
\]  

(1)

where, \( f_j^i \) is the ith sample of class j, \( \mu_j \) is the mean of class j, \( c \) is the number of classes, \( N_j \) is the number of samples in class j.

\[
S_b = \sum_{j=1}^{c} (\mu_j - \mu)(\mu_j - \mu)^T
\]  

(2)

where, \( \mu \) represents the mean of all classes.

Then the subspace for LDA is spanned by a set of vectors \( W = [W_1, W_2, \ldots, W_d] \), satisfying

\[
W = \arg \max_m \frac{W^T S_b W}{W^T S_w W}
\]  

(3)

The within-class scatter matrix represents how face images are distributed within classes and between-class scatter matrix describes how classes are separated from each other. When face images are projected into the discriminant vector \( W \).

Face images should be distributed closely within classes and should be separated between classes, as much as possible. In other words, these discriminant vectors minimize the denominator and maximize the numerator in equation (3). \( W \) can therefore be constructed by the Eigen vectors of \( S_b^{-1} S_w \). PCA tries to generalize the input data to extract the features and LDA tries to discriminate the input data by dimension reduction.

III. Experimental Results

Gabor filter:

In Gabor filter, the face is recognized by means of first the Gabor filter is created and Fast Fourier Transform (FFT) is taken. Then the input image is taken and the FFT is applied. After that convolve both the transformed input image and the transformed filter which is created already. Now the more orientations and scales of the input image is created. It is
then compared with the trained database and recognition is performed. But it is more complicated to implement in FPGA because of more scaling and orientations.

![Gabor filtered image](image1)

**Figure 3**: Gabor filtered image

The above figure shows the Gabor filtered image and then the maximum of these features are taken up and from that the recognition is take place (figure 4) by means of the Caltech 101 classifier, minaret classifier and Multiclass One-Versus-All.

![Recognized image using Gabor filter](image2)

**Figure 4**: Recognized image using Gabor filter

**LDA:**

![Device Utilization Summary](image3)

**Device utilization summary:**

<table>
<thead>
<tr>
<th>Device utilization summary:</th>
<th>Selected Device: 5vlx30f324-3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Slice Logic Utilization</strong></td>
<td></td>
</tr>
<tr>
<td>Number of Slice Register</td>
<td>2485 out of 19200</td>
</tr>
<tr>
<td>Number of Slice LUTs</td>
<td>9867 out of 19200</td>
</tr>
<tr>
<td>Number used as Logic</td>
<td>5451 out of 19200</td>
</tr>
<tr>
<td>Number used as Memory</td>
<td>4416 out of 5120</td>
</tr>
<tr>
<td>Number used as RAM</td>
<td>4416</td>
</tr>
<tr>
<td><strong>Slice Logic Distribution</strong></td>
<td></td>
</tr>
<tr>
<td>Number of Bit Slices used</td>
<td>10294</td>
</tr>
<tr>
<td>Number with an unused Flip Flop</td>
<td>7809 out of 10294</td>
</tr>
<tr>
<td>Number with an unused LUT</td>
<td>427 out of 10294</td>
</tr>
<tr>
<td>Number of fully used Bit Slices</td>
<td>2058 out of 10294</td>
</tr>
<tr>
<td><strong>IO Utilization</strong></td>
<td></td>
</tr>
<tr>
<td>Number of IOs</td>
<td>69</td>
</tr>
<tr>
<td>Number bonded IOBs</td>
<td>69 out of 220</td>
</tr>
</tbody>
</table>
A Systolic Architecture for Object Recognition Based On Hmax Model

<table>
<thead>
<tr>
<th>IOB Flip Flops/Latches</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Feature Utilization</td>
<td></td>
</tr>
<tr>
<td>Number of BUFG/BUFGCTRLs</td>
<td>16 out of 32 50%</td>
</tr>
<tr>
<td>Number of DSP48E1s</td>
<td>4 out of 32 12%</td>
</tr>
</tbody>
</table>

Timing Summary:

| Speed Grade | 3 |
| Minimum period | 6.489ns (Maximum Frequency: 153.900MHz) |
| Minimum input arrival time before clock | 2.158ns |
| Maximum output required time after clock | 3.135ns |
| Maximum combinational path delay | No path found |

IV. Conclusion

By Gabor filter, the image is recognized properly, but the drawback is it is more complicated to implement in FPGA. Because it has large number of orientations and for that it need FPGA kit of latest version which is more expensive. To overcome this, Linear Discriminant Analysis method has been successfully applied to face recognition which is based on a linear projection from the image space to a low dimensional space. But the major drawback of applying LDA is that it may encounter the small sample size problem. When the small sample size problem occurs, the within-class scatter matrix becomes singular. Since the within-class scatter of all the samples is zero in the null space of Sw, the projection vector that can satisfy the objective of an LDA process is the one that can maximize the between-class scatter.

But face image data distribution in practice is highly complex because of illumination, facial expression and pose variation. The kernel technique is used to project the input data into an implicit space called feature space by nonlinear kernel mapping. Therefore kernel trick is used taking input space and after that LDA performed in this feature space, thus a non-linear discriminant can be yielded in the input data.

Acknowledgement

Apart from our efforts, the success of any work depends on the support and guidelines of others. We take this opportunity to express our gratitude to all who have helped us directly or indirectly to complete this paper.

References


www.iosrjournals.org 28 | Page