FPGA Implementation of CNN based Fingertip Digit Recognition

MD Shahbaz Khan, Niharika, Priya Yadav, Rishabh Verma, Indu Sreedevi

Abstract: The paper proposes a novel framework for fingertip digit recognition using Convolutional Neural Network (CNN) and its hardware implementation on FPGA. In the above-mentioned framework, tracking of the fingertip is done using Learning Adaptive Discriminative Correlation Filter (LADCF) algorithm using C++. From the tracked image, digit recognition is done using CNN. CNN model was trained on MNIST dataset of 50000 images of handwritten digits and tested on 10000 such samples with the accuracy rate of 98.4% while recognising an unseen digit. The predicted digit was displayed on seven-segment using SystemVerilog. As per the outcomes of VOT 2018 challenge, LADCF based tracking algorithm has an accuracy of 90.7%. The proposed algorithm is fast in processing as it uses LADCF based fingertip tracking and CNN for digit recognition. Existing works in this field uses image classification tree to recognise digits[12], whose accuracy is 95.8% which is much less than the proposed algorithm in this paper.

Keywords: CNN, Fingertip-based digit recognition, FPGA implementation, image processing.

I. INTRODUCTION

Computer vision finds its wide application in day-to-day life. Character recognition is an important task of computer vision algorithm. With the progress in algorithms, computational efficiency is increasing manifold thus making human-computer interface smooth, faster and more precise. Advancement in machine learning algorithms has also developed few computational efficient computer vision algorithms and one such is Convolutional Neural Network (CNN). Fingertip based gesture system plays a vital role in the life of physically disabled people. Using fingertip gestures, one can communicate in the vacuum without any medium of writing. Such systems should be computationally faster with higher precision. The paper proposes a noble methodology of fingertip based digit recognition. Using camera input, video of the movement of a fingertip is recorded using SystemVerilog and C++ language. On applying the image processing algorithms in C++ language, the gesture of digit is extracted from video into one frame. That frame is further applied on a trained model for digit recognition. The digit predicted by the CNN based algorithm is displayed on seven-segment display, connected to the FPGA board.

II. LITERATURE LAYOUT

Several algorithms have come up to make fingertip-based digit recognition faster. However, accuracy for recognising digits is still low. Algorithms like [12], has an accuracy of 95% only, as compared to ours of 98.4%. Some of the algorithms [2], [3] uses image processing and decision tree based digit recognition, which are computationally less efficient with lower accuracy. In contrary to this, our algorithm uses FPGA implementation of CNN which makes it much faster as compared to the algorithms running on CPUs or GPUs.

Paper like [7] uses R-CNN based algorithm for fingertip tracking. However as per the result of VOT2018 challenge [21], LADCF tracking algorithm out throws the other tracking algorithms. It has an efficiency of 90.7% for tracking an object in continuous frames at a fps of 50. Along with LADCF based fingertip tracking, the paper proposes CNN based handwritten digit recognition system. CNN model for digit recognition is trained on 50,000 handwritten images and been tested on 10,000 images which yielded an accuracy of 98.4%. The paper deals in hardware implementation using FPGA board of digit recognition using CNN. As per the white paper published by Microsoft [5], training of any neural network on FPGA is much faster than the its training on CPUs and GPUs.

III. PROBLEM STATEMENT

The paper is aimed to find the solution for the fingertip digit recognition system. The intent is to implement the CNN based recognition system on FPGA board as to increase its computational efficiency. Earlier work in this field [13] was carried out using Python language on CPUs, which are not as fast as FPGAs. The proposed algorithm also enhances the accuracy of digit prediction as it uses CNN for handwritten digit recognition. Works like [11], [12] deals in decision tree and SVM based classifier for handwritten digit recognition, whose accuracy is not as high as CNN based algorithm. Moreover, on implementing algorithm on FPGA, training speed of model also increases manifold.
IV. EXTRACTING FOREGROUND

The very step in the field of fingertip tracking is the removal of background from each frame. Background subtraction is an important image pre-processing technique to extract the region of interest. This preprocessing of the image is mainly done to subtract the foreground from the background. For the abovementioned task, background segmentation algorithm (based on Gaussian Mixture) was used. This algorithm is based on two research papers [6],[9] given in the year 2004 and 2006 respectively. For each pixel, the algorithm picks the Gaussian distributions by itself. It makes the algorithm flexible for diurnal changes. Moreover, it proves to be efficient for shadow detection and its removal from the foreground. This background subtraction algorithm makes the whole framework independent of diurnal changes and shadow and hence enhances its efficiency.

![Flowchart of proposed algorithm.](image)

V. FINGERTIP DETECTION

Detecting a fingertip in an RGB video, is a tough and challenging task as the dimension of fingertip is small compared to the whole frame. However, recent CNN-based algorithms have come up to detect fingertip precisely and accurately. The paper implements an algorithm [7] which segments the hand region, followed by localization of its centroid. From the segmented region, fingertip is localized with the help of an entropy function, termed as distance-weighted curvature entropy function.

In the above-mentioned entropy function, distance of each contour points from the centroid is calculated and stored in a 1-D matrix. Then the weighted distance is calculated by multiplying the matrix with the distance of curvature points from the centre of hand. Using contours detecting algorithm as proposed by [16], contours of the hand are detected. Let $\mathcal{P}$ be the contours of hand of length $k$. On sampling the contours into $f$ samples, sampling interval is given by,

$$\Delta p = \frac{k}{f}$$  \hspace{1cm} (1)

The turning angle $\theta$ is defined as the change in direction of tangent of two points on contour. While curvature $c$ is the change in tangent direction of points on moving along the curve and is given by,

$$c(p) \propto \frac{\theta}{\Delta p}$$  \hspace{1cm} (2)

Hence the curvature entropy $u$ which is locally proportional to $c(p)$ (curvature of the contour) is approximated by,

$$u(c(p)) \theta = \cos(\Delta p \cdot c(p))$$  \hspace{1cm} (3)

The other factor influencing entropy function is the distance between contour points and the centroid $(x', y')$ which is denoted by $s(p)$. The complete function is denoted by $\chi(p)$ and is defined as the
following,

\[ \hat{x}(p) = u(c(p)).s(p)^* \]  

(4)

Where \( \tau \) is the weighted distance as defined earlier.

It is known that the region of fingertip will have maximum value for curvature entropy along with maximum distance from centroid. Thus, the entropy function \( \hat{x}(p) \) will have its maxima at the fingertip coordinates, which may be expressed as,

\[ x_a, y_a = \arg \max \hat{x}(p) \]  

(5)

Where \((x_a, y_a)\) are the coordinates of fingertip.

VI. FINGERTIP TRACKING USING LADCF

As discussed in [20], LADCF algorithm is based on temporal consistency preserving spatial feature selection method for appearance modelling. On detecting the fingertip in a frame, the tracking algorithm(LADCF) starts its execution. It uses single-channel features which can be further extended to multi-channel. Direction Method of Multiplier (ADMM) is used for optimizing the algorithm.

A. Spatially Feature Selection Technique

Selection of features is done using discriminative and descriptive information stored in filter ‘\( h \)’.

\[ h_{\alpha} = \text{diag}(\alpha)h \]  

(6)

Where, \( \alpha \) is the feature vector.

\( \text{diag}(\alpha) \) is the diagonal matrix of \( \alpha \).

Spatial feature vector is dependent on input by,

\[ h_s^j f = h_s^j f_\alpha \]  

(7)

In learning stage, spatial feature vector can be given as,

\[ \hat{h}^j = \arg \min \left\{ h \bigcup f - \beta \| f \|^2 + \lambda_1 \| f \|_0 \right\} \]  

(8)

For the approximation of sparsity, convex envelope \( l_1 \)- norm and \( l_0 \)- norm (non-convex) were obtained and was restricted to current template ‘\( \hat{x} \)’.

\[ \| \alpha - \alpha_{\text{model}} \|_0 < \xi \]  

(9)

Using \( l_1 \)- norm, spatial feature is given by,

\[ \arg \min \left\{ h \bigcup f - \beta \| f \|^2 + \lambda_1 \| h \|_1 + \lambda_2 \| h - h_{\text{model}} \|_1 \right\} \]  

(10)

Where \( \lambda_1 \) & \( \lambda_2 \) are tuning parameters.

B. Multi-channel Feature Selection Model

In this framework, the appearance model was characterised by multi-channel feature representation. Multi-channel input is denoted as,

\[ h = \{h_1, h_2, ..., h_k\} \]  

(12)

Where \( k \) is the number of channels.

\[ f = [f_1^1, f_1^2, f_1^3, ..., f_1^i] \]  

(13)

In case of multi-channel model, the objective function is given by,

\[ g(h) = \sum_{i=1}^{k} \| h_i \bigcup f_i - \beta \| f_i \|^2 + \lambda_1 \sum_{i=1}^{k} \| h_i \|_1 + \lambda_2 \sum_{i=1}^{k} \| h_i - h_{\text{model}} \|_1 \]  

(14)

Where \( h_i^j \) is the \( j \)th element of \( i \)th channel feature vector.

C. Optimisation

To optimize the above derived equation, eq’(14), convex optimization was employed,

\[ h = \arg \min \left\{ \| h \bigcup f - \beta \| f \|^2 + \lambda_1 \sum_{i=1}^{k} \| h_i \|_1 + \lambda_2 \sum_{i=1}^{k} \| h_i - h_{\text{model}} \|_1 \right\} \]  

(15)

We define a set of Lagrange multipliers, given by

\[ \eta = \{\eta_1, \eta_2, ..., \eta_k\} \]  

(16)

With a penalty controlling parameter \( \mu \) & \( \mu > 0 \).

For updating \( h \), for a given \( h^*, \eta \) & \( \mu \). We optimize \( h \) using OCF learning method.

\[ h = \arg \min \left\{ \| h \bigcup f - \beta \| f \|^2 + \lambda_1 \sum_{i=1}^{k} \| h_i \|_1 + \lambda_2 \sum_{i=1}^{k} \| h_i - h_{\text{model}} \|_1 \right\} \]  

(17)

\[ h^* = \arg \min \left\{ \| h \bigcup f - \beta \| f \|^2 + \lambda_1 \sum_{i=1}^{k} \| h_i \|_1 + \lambda_2 \sum_{i=1}^{k} \| h_i - h_{\text{model}} \|_1 \right\} \]  

(18)

\[ \eta = \arg \min \left\{ \| h \bigcup f - \beta \| f \|^2 + \lambda_1 \sum_{i=1}^{k} \| h_i \|_1 + \lambda_2 \sum_{i=1}^{k} \| h_i - h_{\text{model}} \|_1 \right\} \]  

(19)

TABLE I. TRACKING ALGORITHM

<table>
<thead>
<tr>
<th>LADCF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pre-requisites</strong></td>
</tr>
<tr>
<td>Input image, ( I ), filter (model) ( h_{\text{model}} ), centres coordinate (target) ( c_{x_{-1}} ) &amp; frame size ( h \times h ) of frame ( i-1 ).</td>
</tr>
<tr>
<td>Tracking:</td>
</tr>
<tr>
<td>1. Search windows were extracted of frame sizes ( I ) from input image at its centre coordinate.</td>
</tr>
<tr>
<td>2. Corresponding feature representation was obtained ( g(F(i)) ).</td>
</tr>
<tr>
<td>3. Response scores ( r ) was calculated.</td>
</tr>
<tr>
<td>4. Centre coordinate and scale size were set.</td>
</tr>
<tr>
<td>Observation:</td>
</tr>
<tr>
<td>1.</td>
</tr>
<tr>
<td>2. ‘( \alpha )’ was optimised for ( k ) iterations.</td>
</tr>
<tr>
<td>Update:</td>
</tr>
<tr>
<td>1. Filter model ( h_{\text{model}} ) was updated.</td>
</tr>
<tr>
<td>Output:</td>
</tr>
<tr>
<td>1. Bounding box was drawn using centre coordinate and frame size.</td>
</tr>
<tr>
<td>2. Updated filter model ( h_{\text{model}} ) for the given frame ( i ).</td>
</tr>
</tbody>
</table>

Where sparsity of \( h \) is increased, by increasing the similarity between the estimated \( h \) and the template model \( h_{\text{model}} \).
VII. DIGIT RECOGNITION

After tracking the fingertip movement in consecutive frames, single frame is constructed from it as shown in the figure 23123. The image thus formed is further used to recognise the numeral character present in it. For recognising the handwritten digit in the frame, we have deployed CNN based algorithm. As proposed in [18], CNN is a computationally faster algorithm as compared to other multi-layer perceptron. CNN works on 2D convolution on the input pixel values. CNN based algorithm implemented in our paper for digit recognition task can be divided into following subtasks:

A. Convolution Layer

The very first layer of a CNN is the convolution layer. An RGB image consists of three channels i.e., Red, Green and Blue. However, convolution layer can only work on a single channel at a time. Thus, 3 channel RGBV image was converted into one channel by taking average of it. Hence, in this layer we have a 2-D matrix of 28 x 28 pixels. We picked three filters randomly of 5x5 from the image. Convolution of all the three filters with the image was performed resulting into three 2-D matrices of 28 x 28 pixels. The whole task could be summed up in a table:

<table>
<thead>
<tr>
<th>TABLE II. CONVOLUTION LAYER ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: RGB image was converted into a single channel image.</td>
</tr>
<tr>
<td>Step 2: - Three filters of 5 x 5 pixel was picked from the image randomly.</td>
</tr>
<tr>
<td>Step 3: - Each filter was multiplied by the corresponding pixel of 28 x 28 image.</td>
</tr>
<tr>
<td>Step 4: - All the elements of resultant 5 x 5 matrix was summed and divided by 25.</td>
</tr>
<tr>
<td>Step 5: - Filter was moved and centered at the next pixel of image and step 2 to 4 was repeated for each filter separately.</td>
</tr>
</tbody>
</table>


![Three Filters used in Convolution.](image)

B. Activation Layer

The resultant matrices of convolution layer may or may not consists of negative values. However, a negative value as the pixel of an image, holds no meaning. In order to remove negative values from the pixel matrix of filtered images, we applied an activation function. An activation function only lets the positive values to pass through it. There are many activation functions as such sigmoid function, Rectified Linear Unit (ReLU) function, tanh function etc. In our algorithm we applied ReLU as an activation function for the filtered images. A ReLU only activates a node if its value is greater than 0 and deactivates the negative node. A ReLU layer can be mathematically defined as follows:

![ReLU Activation Function.](image)

C. Max Pool Layer

After passing the image matrices from ReLU layer, the dimension of the matrix is very high i.e., 28 x 28 that makes up to 724 elements. In order to reduce the size of image matrix, we created a window of 2 x 2 size and put it at the beginning of image pixel matrix. Maximum value of that window was picked and replaced in the image matrix. Window was slided over the whole matrix, thus reducing its size to its half.

After pooling layer, the whole process starting from convolution to pooling, was repeated thrice. This resulted in 4 x 4 matrices of three channels. All the elements of three channels were arranged in a single vector/list in a row major form. The vector thus formed after training, was unique for each digit and were 10 in number. Unknown handwritten digits had to undergo all the above-mentioned steps, which yielded a 1-D vector for them. The resultant vector was compared with all the 10 vectors that were formed during training of model. The unknown digit was predicted on the basis of highest resemblance of its vector with the precalculated 10 vectors.

VIII. DISPLAY ON SEVEN SEGMENT

After the fingertip drawn digit is recognized by CNN algorithm, it is displayed on seven-segment display board. This is task is performed using the modules created in SystemVerilog. Steps performed for displaying the digit are as follows:

![Fig. 3.ReLU Activation Function.](image)
TABLE III. DISPLAYING THE RECOGNISED DIGIT

<table>
<thead>
<tr>
<th>STEPS FOR DISPLAYING THE PREDICTED DIGIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1: Read the predicted digit using ‘readmemb()’ function</td>
</tr>
<tr>
<td>Step 2: Assigned the data read to a register named ‘digits’.</td>
</tr>
<tr>
<td>Step 3: Converted the binary data to the anode and cathode connection values using ‘case()’.</td>
</tr>
<tr>
<td>Step 4: Produced the data if data lies in the range 0-9 else turned off the 7-segment display.</td>
</tr>
</tbody>
</table>

IX. RESULT

The above proposed algorithm was trained on MNIST datasets containing 60,000 images of handwritten digits. Dataset was divided into training and testing data in the ratio 5:1. The CNN model was trained with the following parameters:

- Number of epochs = 17
- Learning rate = 0.003
- Number of hidden layers = 2
- Optimizer used = Stochastic Gradient Descent

On testing the trained model on 10,000 images, accuracy of the algorithm was 98.4% which is much higher than [7] which was 95%.

IX. CONCLUSION

The paper has proposed an algorithm for fingertip writing digit recognition system using CNN. Proposed recognition system processes both pre-processed image algorithms and fingertip writing numbers recognition algorithms in hardware. The CNN model has been trained on 50,000 handwritten digits provided by an MNIST and has been tested on 10,000 images. The computational efficiency of this algorithm is very high as it implements FPGA for digit recognition and the accuracy of digit recognition is 98.4%. The proposed algorithm makes use of both SystemVerilog as well as C++ for fingertip hand recognition. SystemVerilog is used just to communicate between hardware and software. Direct programming interface facility of SystemVerilog has been used to read an image and to convert it in its RGB value using C++ program. That corresponding RGB value has been further used for digit recognition and the recognised digit was displayed on SevenSegment using SystemVerilog. The future expectation of this algorithm is the betterment of the recognition system and increasing its computational efficiency manifold. In our coming papers, we will be proposing the algorithm to implement the CNN on hardware only as it will increase the computational efficiency and will make the algorithm faster with a high accuracy rate. So that our main motive of helping humankind by making use of machines will come true.
ACKNOWLEDGMENT

We express our deepest gratitude to all those who helped us make this research complete and meaningful in all sense. First of all, we would like to thank our faculty in-charge, Prof. S. Indu, whose contribution to this research work is incredible. She helped us to get through this research work by providing her valuable suggestions and sparing her precious time. Furthermore, we acknowledge the contribution of other staffs of the ECE department (DTU), who helped us by providing their valuable suggestions related to our topic. Last but not least, we are highly thankful to our parent and friends, who provided us moral support in times of distress and failure and played a vital role to make this work complete in all sense.

REFERENCES

13. X. Yang, J. pu, MDig: Multi-digit Recognition using Convolutional Neural Network on Mobile, 14th International Conference on Natural Computation, Huazhong, China.

AUTHORS PROFILE

MD Shahbaz Khan currently pursuing B. Tech in Electronics & Communication branch from Delhi Technological University, Delhi India. His fields of interest for research are Computer Vision, Machine Learning and FPGA.

Niharika currently pursuing B. Tech in Electronics & Communication branch from Delhi Technological University, Delhi India. Her field of research are Machine Learning and Data Science.

Priya Yadav currently pursuing B. Tech in Electronics & Communication branch from Delhi Technological University, Delhi India. Her fields of interest are Machine Learning, FPGA and Verilog.