ANN Based Digital Circuit Simulator

Sindhushree J M[1]

[1]*Student, Dept. of Electronics and Communication Engineering, P.E.S College of Engineering, Karnataka, India*

---***---

Abstract - *Transforming an electrical diagram or network involving active and passive devices into an netlist ready for simulation is an challenging and interesting work. This makes the faculty and engineers to work at higher abstract level where in the designer interacts with the underlying software through the abstract images. This transformation accelerates the designed and simulation work. The task and developing the software involves the image processing and artificial neural network methods and algorithms for recognition of components and netlist generation. The traditional methods of pattern recognition are not efficient enough in identifying the elements and their connectivity due to the dramatic variations like size, orientation etc. The use of ANN will help to improve the efficiency. The recognition process involves two steps: first step is feature extraction using shape based features, and the second one is a classification procedure using ANN through a back propagation algorithm. The ANN was trained and tested with different hand drawn electrical images. The results show that our proposal is viable and brings good performances.*

Key Words**: Artificial neural network, Backpropogation algorithm, ALEXNET, Feedforward Neural Network, Activation function.**

1. INTRODUCTION

Advances in digital technology have spread out several avenues of leaning. Technology has created information accessible to all varieties of learners. To create the educational a far better expertise technical establishments has to adopt and update their teaching learning method with the inclusion of good integrated technologies.

In technical institutions whereas handling engineering courses machines and mechanics etc for the sake of understanding the ideas teacher need to write the circuits on the board to investigate its outputs and connected parameters with waveforms. This ends up in a tedious time consuming task of analysis. To make learning of engineering or science courses powerful with emphasis on analysis and visualization, sophisticated and smart technology/innovations need to be developed.

The proposed work, ANN based digital circuit simulator is an attempt to overcome the above discussed challenges particularly about electronic circuits. The method uses Artificial neural network (ALEXNET) technology for recognizing the parameters of the electrical circuit diagram.

2. LITERATURE REVIEW

[1] Dr. Mahesh koti, Jayashree M S, Abhilasha H proposed **"Smart Digital Circuit Simulator"** in 2019.

In this paper, they make use of pattern recognition and template matching for recognizing the parameters of the digital circuit diagram. Their method initially performs image pre-processing steps such as normalization, binarization etc. The pre-processed image is then segmented into multiple segments. Then these segmented images are recognized using template matching method. But template matching method is based on measuring the similarities between the structure of the input image and a given set of templates for recognition.

Inherently, this method is sensitive to template mismatch when the input components are not exactly the same style as the templates. Because of the slightly change in the structure of the input components, the highest matches of certain components are not found in their true corresponding templates of components. So in our proposed method we replace the use of template matching with the Artificial neural network for recognizing the parameters of the electrical circuit diagram.

[2] Long Chen'Dan, Liu'Xiuhan, Xi'HuiXie proposed **"Image Preprocessing Algorithm Analysis Based On Roberts Operator And Connected Region"** in 2012.

This paper proposed a new image segmentation algorithm which is based on connected region. The connected region refers to classifying pixels in image and figuring out which pixels are useful, which pixels are extra in the image analysis. The main objective of connected region processing is to eliminate irrelevant information in the image, recover useful real information and enhance the detectability of relevant information, build data to the largest extent to improve the reliability of feature extraction, image segmentation, matching and recognition.

The connectivity between the image pixels needs to be judged by marking algorithm, among which four-connection and eight-connection are main marking algorithms. This image segmentation algorithm based on connected region have been adopted in our work for labelling the connected regions, so that these regions can be easily extracted for recognition.

[3] Saad Albawi, Tareq Abed Mohammed, Saad Al-Zawi proposed **"Understanding of a Convolutional Neural Network"**, in 2017.

In this paper they discuss about significant issues that are identified with convolutional neural system (CNN) and clarify the impact of every boundary on execution of system.

The most significant layer in CNN is convolution layer which takes most of the time within the network. The performance of the network additionally relies upon the quantity of levels inside the system. On the other hand, the quantity of levels expands the time required to prepare and test the system.

Likewise the another significant layer in the neural system is fully connected layer that is associated with each neuron in both past and next layer. Another significant part of CNN, is to acquire conceptual highlights when information engenders toward the more profound layers. Today the CNN consider as power full device inside AI for a great deal of utilization, for example, face location and picture video acknowledgments and voice recognition. From this paper the basic architecture and principles of convolutional neural network have been studied which can be used for classification purpose. They also done a study of most popular architecture such as ALEXNET and LeNet network for classification purpose.

[4] Shakshi Sharma, Dr. Akanksha Juneja, Dr. Nonita Sharma proposed **"Using Deep Convolutional Neural Network in Computer Vision for Real-World Scene Classification",** in 2018.

The scene classification is critical in computer vision. It can automate the procedure of human visual framework. In this paper the system they have utilized for scene classification is Deep Learning which ends up being valuable in extracting features of information. Deep Learning architecture i.e. CNN (ALEXNET) is an awesome programmed feature extraction network which beats the issue of picking hand planned component extraction techniques and CNN handles large amount of data which is accessible nowadays. On the off chance that they would prefer not to make CNN design from scratch, at that point there is likewise a choice to reuse effectively prepared CNN engineering in different tasks i.e. Transfer Learning concept. Here, utilized exchange learning way to deal with lessen computational complexity and accomplish great execution in scene classification.

From the outcomes of this paper, it tends to be inferred that the scene classification accuracy can be accomplished by transfer learning idea, it is valuable and can be additionally enhanced by applying feature extraction techniques to maintain a strategic distance from include redundancies. The basic steps involved in classifying the segmented images using transfer learning concept have been adopted in this paper. For our work we make use of same pretrained network i.e, ALEXNET for classification because the network relatively has high accuracy and it applies transfer learning approach for classification task to reduce the overall computational complexity of the neural network.

3. METHODOLOGY

The proposed work, ANN based digital circuit simulator have been split into different number of stages as shown in the Fig-1. The circuit image that is handwritten is captured and it's taken as input image for pre- processing. The input image could be a combinational circuit diagram, it specifically desires processing on the shapes and links instead of color data therefore it's necessary to convert rgb image to gray image. The binary image is born-again into noise free image is then passed to the segmentation part to get individual circuit components. Once components are extracted we have to acknowledge that components using Artificial neural network (ANN), then the connection among the components generate a netlist and facilitate to seek the output response for all potential conditions.

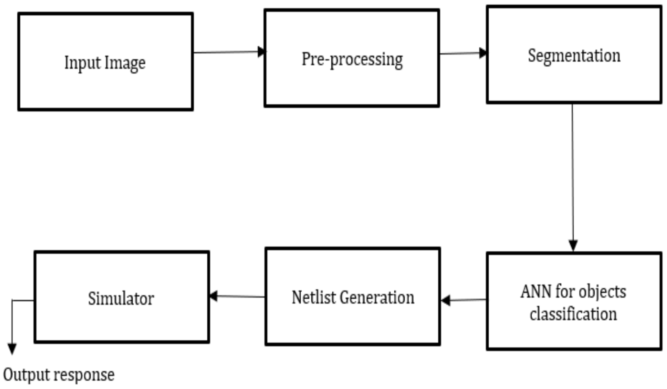


Fig -1: Block Diagram of ANN Based Digital Circuit Simulator.

A. Pre-Processing

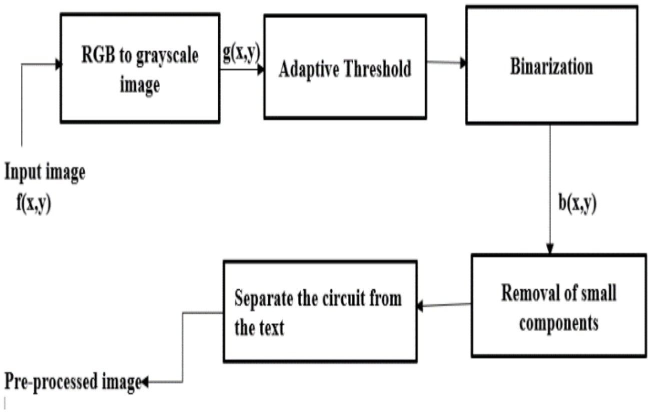


Fig -2: Block Diagram of Pre-Processing Steps

The basic steps involved in processing of the input image which is necessary for segmentation is shown in the Fig-2. Each steps are discussed below in briefly,

RGB to grayscale image

Image processing is specifically desires processing on the shapes and links instead of color information therefore it's necessary to convert rgb to grayscale image.

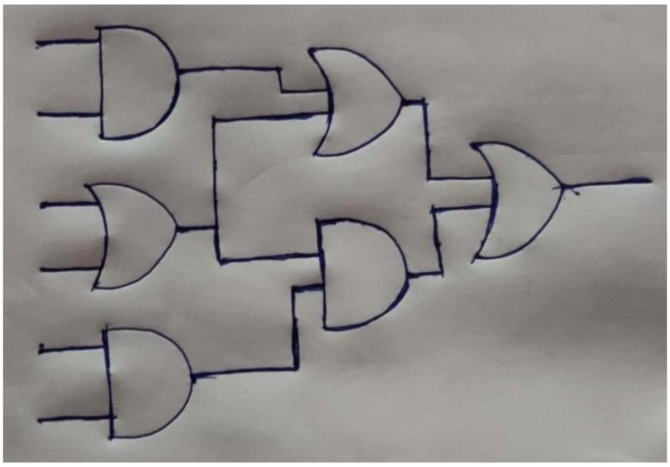


Fig -3: Input RGB Image

We are able to convert rgb values to grayscale values by forming weighed sum of r, g and b elements of f(x, y) as illustrated in the equations (3.1) and (3.2).

$$(R, G, B) = f(x, y) \tag{3.1}$$

$$g(x, y) = 0.2989*R+0.5870*G+0.1140*B \tag{3.2}$$

Where g(x, y) is a grayscale image and f(x, y) is a input RGB image is shown in the Fig-3. The obtained grayscale image g(x, y) is shown in the Fig-4.

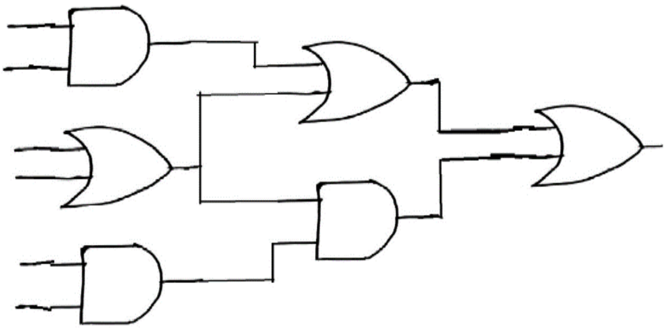


Fig -4: Resultant Grayscale Image

Adaptive Threshold

It is the process of choosing the threshold to minimize the intra class variance of the threshold black and white pixels. For this purpose we make use of Ostu's Thresholding Method. The Ostu's thresholding method is used to computes a global threshold (T) that can be used to convert intensity image to binary image. This involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels either falls in foreground or background.

Threshold is a normalized intensity value that lies in the range [0, 1] of grayscale image g(x, y) as illustrated in the equation (3.3).

$$T = \text{global threshold}(g(x, y)) \tag{3.3}$$

Where, T = global threshold value lies in the range [0, 1] and g(x, y) = grayscale image.

Binarization

The resultant gray scale image is converted into binary image as shown in Fig-5, because it specifically needs processing of shapes and links rather than colours. Thresholding is that the easiest method to convert a grayscale image to a binary image. In thresholding, we tend to choose a threshold value and so all grey level price that is below the chosen threshold value is assessed as 0(black) and every one gray price that is equal or larger than the threshold value classified as 1(white). This can be done as illustrated in the equation (3.4)

$$b(x, y) = \begin{cases} 0 & \text{if } g(x, y) >= T \\ 1 & \text{otherwise} \end{cases} \tag{3.4}$$

Where, b(x, y) = resultant binary image, T = global threshold value, g(x, y) = grayscale image.

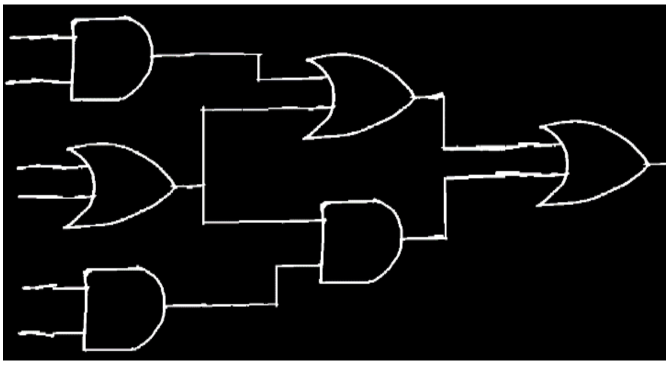


Fig -5: Resultant Binary Image

Removal of small components

In some cases, there would be some smudges or unwanted little dots in the image so we need to be remove this small unwanted objects form an image. The major objective of noise removal is to remove any small components below 30 pixels, which do not have any significance in the output, is as illustrated in equation (3.5).

$$\ln = \begin{cases} 1 & \text{if small components in } b(x, y) < 30 \text{ pixels} \\ 0 & \text{Otherwise} \end{cases} \tag{3.5}$$

B. Segmentation

The preprocessed image is ready to be processed further for classifying the elements. To classify the elements, we need to extract that elements. So we need to segment the resultant pre-processed image into individual segments. Image segmentation is that the method of partitioning a digital image into multiple segments (sets of pixels, conjointly called image objects).

The steps involved in segmentation is as follows,

Labelling connected components in Binary image

L = bwlabel(b(x, y)); (3.6)

The equation (3.6) shows a Matlab function 'bwlabel' which takes in a resultant pre-processed image which contains bunch of objects that are separated from each other. Pixels that belong to an object are denoted with 1 / true while those pixels that are the background are 0 / false. Finally this function creates a label matrix that contains labels for the 8-connected objects or 4-connected objects found in Binary image[2].

Measuring the properties of image regions

The connected regions that are obtained in the above step are fed to 'regionprops' Matlab function as shown in the equation (3.7) for measuring the properties of those regions such as Boundingbox, centroid, area etc to apply bounding boxes around those connected regions.

Props = regionprops(b(x, y), 'BoundingBox', 'Area', 'Centroid'); (3.7)

Finally, this step returns measurements for the set of properties mere by properties for every 8-connected element (object) within the binary image b(x, y) [2].

Applying Bounding Boxes

This step applies a Boundingbox around those 8-connected labelled regions obtained in the previous step. Hence those regions are then can be easily extracted for classification. The "bounding box" of a restricted geometric object is that the box with minimum space or minimum volume, that contains a given geometric object. For any assortment of linear objects like points segments ,polygon ,lines ,etc their bounding box is given by the minimum and most coordinate values for the purpose set S of all the object's n vertices. The bounding box formula is as illustrated in the equation (3.8).

[x y w l] = BoundingBox(b(x, y)); (3.8)

Where, [x y w h] = x coordinate, y co-ordinate, w = width of an image, h = height of an image. The Fig-6 shows the outcome of Boundingbox function where it creates rectangle shaped boxes around each connected regions present in an input binary image b(x, y).

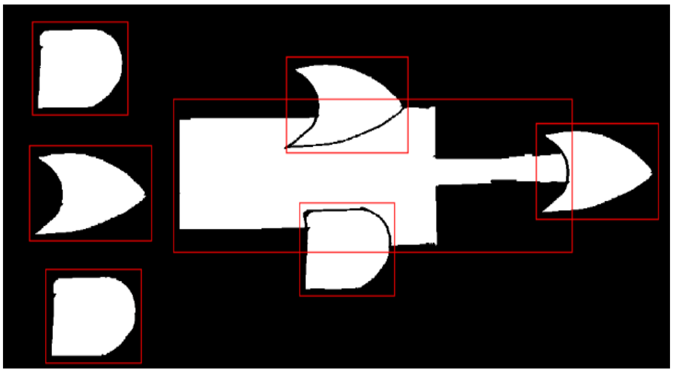


Fig -6: Bounding around Each object in the binary image

By applying the boundingbox around each connected regions obtained by 'bwlabel' we can detect each logic gates in the circuit. But the major disadvantage here is that boundingbox also detects elements other than the logic gates that is, it also extracts non-empty elements other than the logic gates as shown in the Fig-6. So this leads to a inaccuracy in the detection of number of logic gates present in the circuit image. The non-empty elements also includes loops present in the circuit which contain maximum number of pixels than the logic gates. So by getting the number of pixels of each connected region and their index value we can set the corresponding index value that having maximum number of pixels to false. Then only the logic gates present in the circuit will be detected as shown in Fig-7.

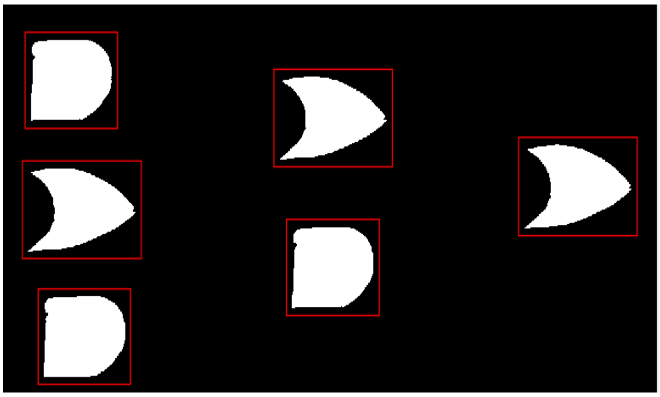


Fig -7: Final segmentation results

Objects Extraction

Each connected regions that are detected using boundingbox function will be extracted in this step. Then these extracted components can be saved to separate folder respectively for further analysis. The Fig-8 shows all the extracted objects.

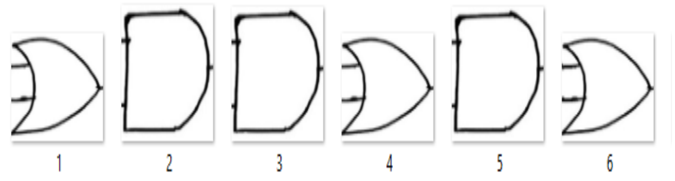


Fig -8: Extracted objects

C. Artificial Neural Network for Objects Classification

The extracted objects shown in the Fig-8 are then fed into the feedforward neural network (ALEXNET) for classifying them into different classes. In a feedforward network, the data moves exactly in one direction – forward – from the input nodes, through the hidden nodes (if any) and to the output nodes. There are not any cycles or loops within the network. An architecture of feedforward neural network is shown in the Fig-9.

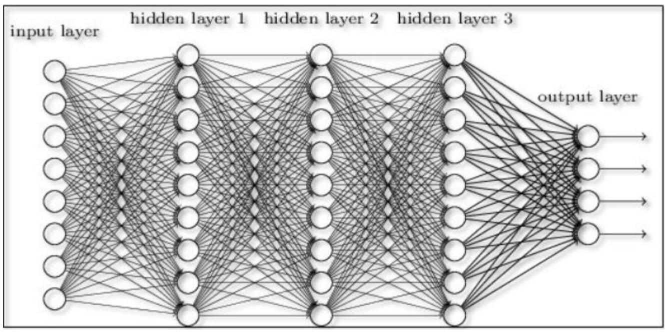


Fig -9: General architecture of Feedforward Neural Network

A feedforward neural network contains three layers:

Input layer – This layer consists of neurons which receive inputs from the outside world and send them to the next layer. Also number of neurons in the input layer will be equal to the attributes or features in the given dataset.

Output layer – The output layer is the predicted feature that depends on the type of model we are building.

Hidden layer – In between the output layer and the input layer there are hidden layers which depends on the type of model. This layer contains more number of neurons which applies transformation on the inputs before passing them. As the network is trained, the weights are updated to be more predictive.

Algorithm for computing the data in four simple steps:

a) The input is multiplied by the assigned weight values.
b) The product obtained in the previous step is added to their respective biases.
c) The activation function then maps summed weighted input to the output of neuron.
d) Finally the weighted sum obtained is turned into an output signal by feeding the weighted sum into an activation function.

The Fig-10 clearly illustrates the above discussed algorithm,

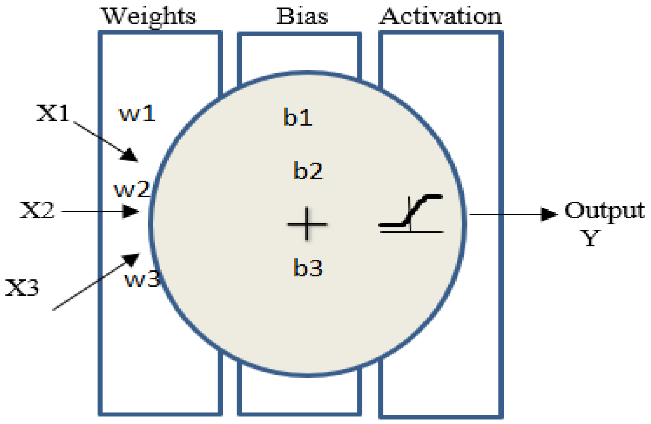


Fig -10: Computing neural network data

Weights

The weights decide how much influence the input will have on the output. Also it controls strength of the connection between the neurons. When the network is initialized all weights values are chosen randomly.

Bias

Bias values are constant values i.e, that are additional input to the next layer will always have the value of 1.

In the Fig-10 where x1, x2, x3 are the three inputs to the network, initial weights of the feedforward network are w1, w2, w3 and biases are b1, b2, b3 and Y is the output signal. Then weighted sum can be calculated as illustrated in the equation (3.9).

$$\text{Weighted sum} = (x1 * w1) + b1 + (x2 * w2) + b2 + (x3 * w3) + b3 \quad (3.9)$$

Activation function

This function maps the calculated weighed sum to the output of the neuron. It is called an activation/transfer function because it governs the inception at which the neuron is activated and the strength of the output signal.

There are several activation functions for different use cases. The most commonly used activation functions are relu, tanh and softmax.

Backpropagation Algorithm

The predicted value of the network is compared to the expected output, and an error is calculated using a function. This error is then propagated back within the whole network, one layer at a time, and the weights are updated according to the value that they contributed to the error. This clever bit of math is called a backpropagation algorithm. The loss is calculated as illustrated in the equation (3.10).

Loss = Y{predicted} – Y{original} (3.10)

ALEXNET – Deep Convolutional Neural Network

The extracted objects are then fed to deep convolutional neural network for classification. Among various available neural networks ALEXNET network gives relatively high accuracy with less number of training data set for our proposed work.
The network make use of transfer learning algorithm which is commonly used in deep learning applications. The transfer learning algorithm includes several steps as follows:

Loading Pretrained network

In this step we load a input dataset for training. We have taken 350 images of varying sizes of 7 different classes such as andgates, orgates, notgates, nandgates, norgates, exorgates and exnorgates each of 50 in number.

The network then divide the data into training and validation datasets. Among these 70% of the images were used for training and 30% for validation purpose.

Then we load a pretrained ALEXNET neural network which consists of 5 convolution layers, 5 max pooling layers, 3 fully connected layers and 7 Relu layers and consists of 1000 neurons in the output and totally 25 layers were used for training. These layers are arranged as shown in the Fig-11 and final calculated weights, bias and activations of the network layers shown in the Fig-12 respectively.

There are mainly four main operation in the convolution neural network are,

- Convolution
- Non – Linearity (Relu)
- Pooling
- Classification (fc layer)
-
- The main parameters of the convolution layer is,

a) Width and height of the convolutional filter.
b) The number of input and output channel.
c) The depth of the convolutional filter must be equal to the number of channels of the input feature map.

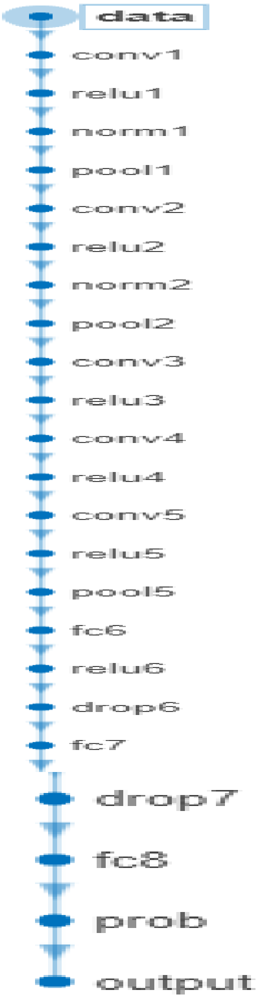
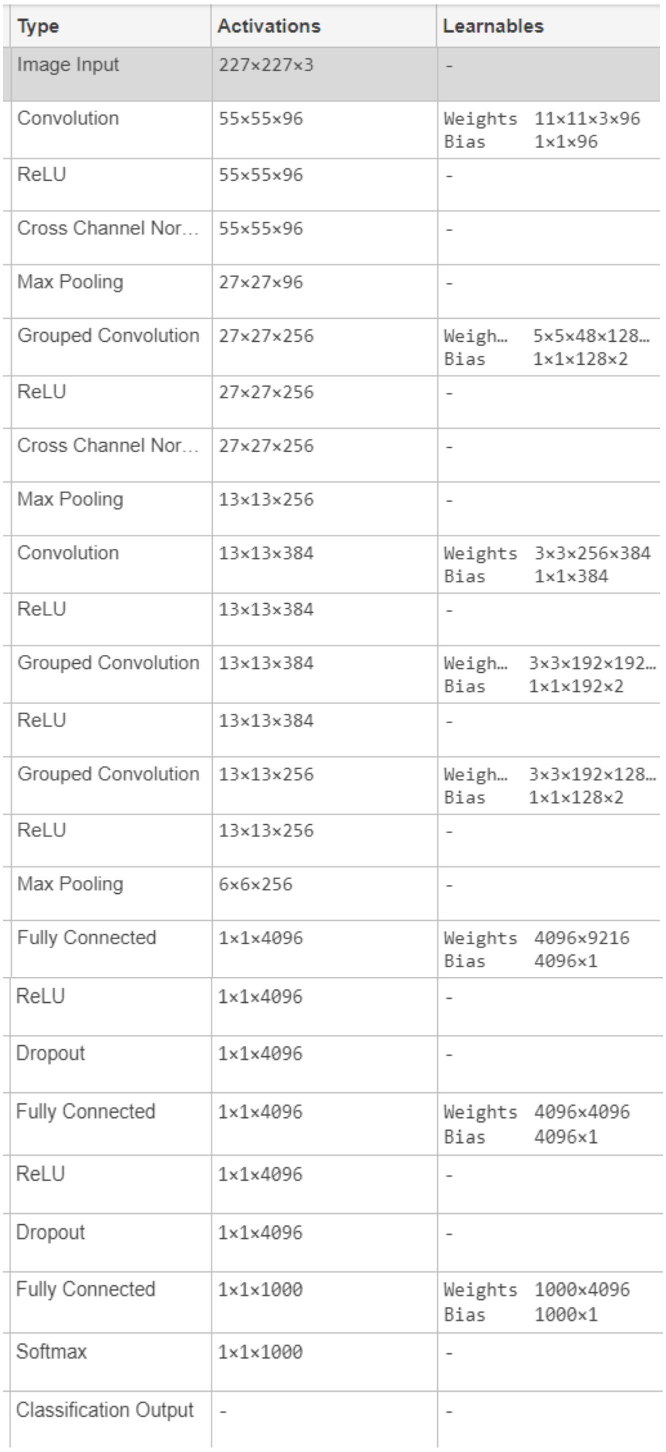


Fig-11: Arrangement of ALEXNET network layers



Type	Activations	Learnables
Image Input	227×227×3	-
Convolution	55×55×96	Weights 11×11×3×96 Bias 1×1×96
ReLU	55×55×96	-
Cross Channel Nor...	55×55×96	-
Max Pooling	27×27×96	-
Grouped Convolution	27×27×256	Weigh... 5×5×48×128... Bias 1×1×128×2
ReLU	27×27×256	-
Cross Channel Nor...	27×27×256	-
Max Pooling	13×13×256	-
Convolution	13×13×384	Weights 3×3×256×384 Bias 1×1×384
ReLU	13×13×384	-
Grouped Convolution	13×13×384	Weigh... 3×3×192×192... Bias 1×1×192×2
ReLU	13×13×384	-
Grouped Convolution	13×13×256	Weigh... 3×3×192×128... Bias 1×1×128×2
ReLU	13×13×256	-
Max Pooling	6×6×256	-
Fully Connected	1×1×4096	Weights 4096×9216 Bias 4096×1
ReLU	1×1×4096	-
Dropout	1×1×4096	-
Fully Connected	1×1×4096	Weights 4096×4096 Bias 4096×1
ReLU	1×1×4096	-
Dropout	1×1×4096	-
Fully Connected	1×1×1000	Weights 1000×4096 Bias 1000×1
Softmax	1×1×1000	-
Classification Output	-	-

Fig -12: Network Analysis Result

Mathematically, convolution operation is represented with an asterisk (*) sign as illustrated in the equation (3.11).

$$Z = x*f \qquad (3.11)$$

Where X is the input image and F = convolutional filter which performs element wise operation.

Input image has size 227*227*3 i.e. height, width and depth of image fed to the input layer i.e, convolution layer of the network and applied a filter of size 11*11 with stride = 1 where stride is the number of pixels by which sliding a filter matrix over the input matrix and there are totally 96 layers as shown in Fig-12. The output of the convolution layer will be feature map with shape illustrated in the equation (3.12)

$$\text{(number of images)} * \text{(feature map height)} * \text{(feature map width)} * \text{(feature map channels)} \qquad (3.12)$$

After this, the number of neurons present in the next layer is calculated using equation (3.13)

$$\text{Number of neurons} = (\text{Width} - \text{FilterSize}+2*\text{Padding})/\text{Stride}+ 1 \qquad (3.13)$$

Where width is the height or width of image i.e. 227 here, filter size is 11, padding=0, Stride=4 in the first layer.

Next, the Relu layer that means Non-linearity functions replace all negative pixels of the feature map with zeros i.e, it performs element wise matrix multiplication and addition. Here we make use of sigmoid function as non-linear function. After convolution, max pooling of size 3*3 is applied with stride=2 which decreases the dimension into halves. i.e, it takes the largest element from the rectified feature map and form a matrix output. The high level features obtained in the convolutional and pooling layer are then fed into the fc layer to use these features for classifying the input image into various different classes based on the training data set. The fc layer uses softmax function as activation function which takes vector real word values and classify it into a vector of values between 0 and 1 that sum to one. The Fig-13 shows some of the hand drawn input images.

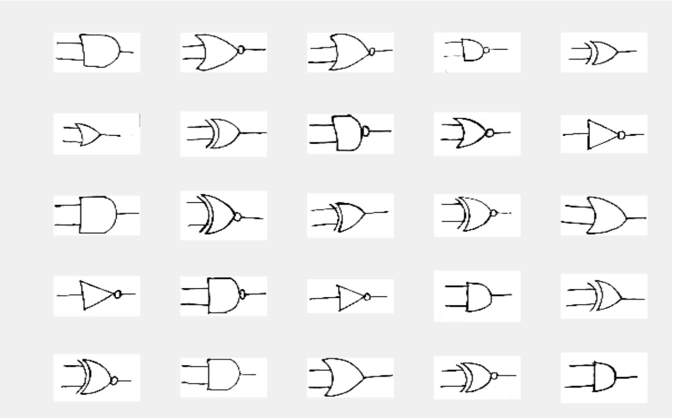


Fig -13: Some models of images from the couple of classification of input datasets

Replace final layers

The last three layers of the loaded pretrained network in the step1 are fine-tuned for our classification problem i.e, these layers are transferred to the new classification task by replacing the last three layers with fc layer, classification output layer and softmax layer as shown in the Fig-12. We can increase the learning rate in these three new layers by increasing the weightLearnrateFactor and BiasLearnrateFactor values of the fullyconnected layer. We set the Weightlearnrate for 20 and Biaslearnrate for 20. This combination of learning rate results in fast learning in the new layers and slow learning in the other layers. Number of classes will be specified in this new fully connected layer.

Train Network

In this step, the ALEXNET network trained with imagedataset of 350 images. But the handwritten images that we are taken for image datastore are of different sizes. So we make use of augmented datastore which automatically resizes all training images to 227*227*3 size. The training process is shown in Chart-1 where percentage of accuracy and loss is plotted with respect to the number of iterations.

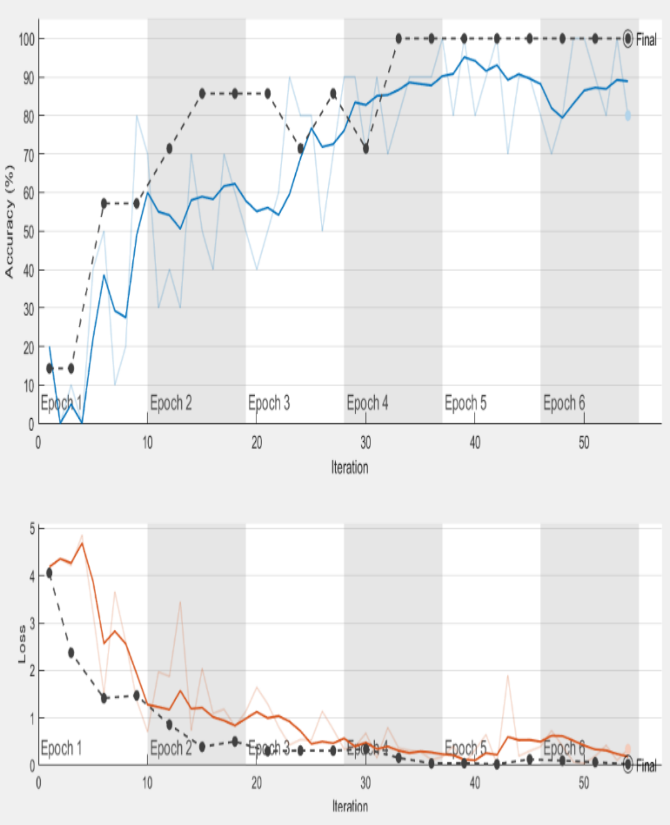


Chart-1: Training process of the ALEXNET neural network

Predict and assess network accuracy

1.

2. The 30% of input dataset which are used for the validation purpose were classified or tested with fine-tuned network and classification accuracy is calculated on the validation set. Accuracy is the fraction of labels that the network predicts correctly. Validation accuracy of various tests have been listed in Table-1.

3.

Table -1: Validation accuracy of ALEXNET neural network

Tests	Number of images taken for tests	Number of Correctly validated images	Accuracy in (%)
Test 1	50	43	86
Test 2	40	40	100
Test 3	120	116	96.66

The extracted objects show in the Fig-8 are fed to the fine-tuned ALEXNET neural network for classification. The classification result is shown in the Fig-14, where all six extracted objects are correctly labelled with class names.

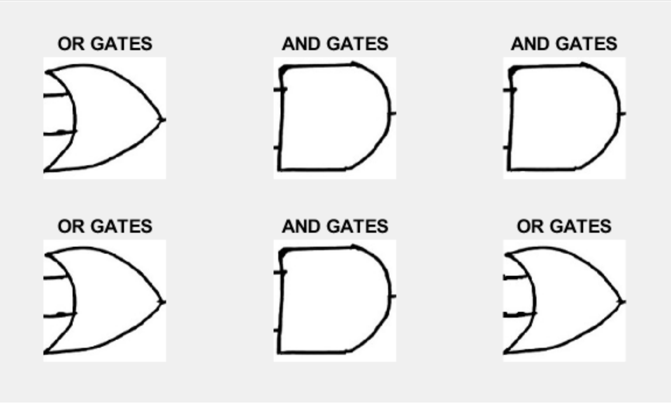


Fig-14: Classification output of ALEXNET neural network for extracted objects

D. Netlist Generation

After the objects are classified it is then necessary to identify the connection among the elements to generate a netlist. In order to do these extract the lines and find out nearest neighbours to those lines and there relation with respect to the element in the right side or left side by using element and line masking.

E. Simulator

The netlist that is generated in the MATLAB is then fed to the stand alone software that converts the schematic of electronic digital circuits for recognized parameters into a output response in the form of tabulation and waveforms. We make use of VIVADO software in this proposed work for the simulation. It is a software suite produced by Xilinx for high level synthesis and synthesis of HDL design.

4. CONCLUSION AND FUTURE ENHANCEMENT

A software package based on artificial neural network for recognizing the parameters of the digital electronic circuits have been developed. The results of tests have shown good qualitative accuracy. The algorithm have been validated with regressive simulation.

In this proposed work neural network have been used only for classifying objects but future research could be done to improve the usage of neural network for both classification and detection purpose. Also system prototype could be improve into a better system by taking the multi loop circuits with different orientations.

ACKNOWLEDGEMENT

I express my earnest gratitude to my guide Dr. Mahesh Koti, partner educator division of Electronics and Communication Engineering for furnishing different assets, for example, research center with all required programming stages, ceaseless Internet association, for my project

REFERENCES

[1]. Shaoqing Ren, Kaiming He, Ross Girshick, Xiangyu Zhang, and Jian Sun “Object detection network on convolution feature map”; 2015 IEEE transaction on pattern analysis and machine intelligence.

[2] Anshika1, Yamuna SV, Nidhi Goee, S.Indu “Neuronal Logic Gates Realization using Vedic Mathematics”; 2015 IEEE 1st International Conference on Next Generation Computing Technologies.

[3] Marc Duranton, Laboratoires d‘Electronique Philips S.A.S. (LEP) “Image Processing by Neural Network”; 1996 IEEE on micro technology.

[4] Shin-Jye Lee1, Tonglin Chen2, Lun Yu2, and Chin-Hui Lai3 “Image Classification Based on the Boost Convolutional Neural Network’’; 2169-3536© 2018 IEEE Access.

[5] Huibao lin, jennie si, glen p Abouseleman “Region-of-interest detection and its application to image segmentation and compression”; ©2007 IEEE.

[6] Hai Gao, Wan-chi Siu, and Chao-Huan Hou “Improved Techniques for Automatic Image Segmentation”; 2001, IEEE transactions on circuits and systems for video technology.

[7] Mahdi Rabbania,*, Reza Khoshkanginib, H.S. Nagendraswamya, Mauro Contib “Hand Drawn Optical Circuit Recognition”; 2015 7th International conference on Intelligent Human Computer Interaction.

[8] D Mishra, Vinayak CS proposed “Image Based Circuit Simulation”; 2013 International Conference on Control, Automation, Robotics and Embedded System(CARE).

[9] Shakshi Sharma, Dr. Akanksha Juneja, Dr. Nonita Sharma proposed “Using Deep Convolutional Neural Network in Computer Vision for Real-World Scene Classification”; 2015 8th International Advance Computing Conference (IACC).

[10] Saad Albawi , Tareq Abed Mohammed, Saad Al-Zawi proposed “Understanding of a Convolutional Neural Network’’; ICET2017, Antalya, Turkey.

[11] Long Chen'Dan, Liu'Xiuhan, Xi'HuiXie proposed “Image Preprocessing Algorithm Analysis Based On Roberts Operator And Connected Region”; 2012 International Conference on Computer Science and Service System.