ABSTRACT

Optical Character Recognition (OCR) software is used for converting handwritten text into electronic word format. Also many of document scanners nowadays have this technology incorporated into them. This is used abundantly in corporate world. However, most of the work done in this field is limited to English Language.

Hence, we attempt to recognize characters of an obsolete regional script, namely ‘Modi.’ Though not in use nowadays, many of the historical documents are written in this script. These documents need to be translated into intelligible script for the inexpert people to understand them. But currently there is no such OCR software. Having said that, the focus of our project is on designing a Deep Learning Artificial Neural Network (A.N.N.), using 'Neo- Cognitron' architecture which uses unsupervised learning algorithm, for recognition of any available pattern (including different languages like 'Modi').

Also, the image processing algorithms like pre-processing and feature extraction techniques are used for simultaneously working on supervised learning algorithm. Evaluation of performance is obtained for the same, in terms of confusion matrix and is compared with that of the Neo-cognitron architecture.

Once this network is successfully designed and working, we will try to reduce the training time taken by the network.

ACKNOWLEDGEMENT

We would sincerely like to thank our project guide **Prof. Mr. Alwin Anuse who** inspired us to take up the project and guided us throughout the year from the first day of college. Apart from imparting valuable knowledge in us, he helped us in acquiring all the project related requirements at all the times. It was his firm support and backing which always made us feel confident to accomplish the project work. We would always want him to be our mentor in future.

We are very much thankful to our HOD, **Prof. (Dr.) G. N. Mulay** for their guidance along with the inspiration and proper suggestions without which this project would not have been completed.

Our genuine sense of gratitude goes to the **Savitribai Phule Pune University** that gave us a chance to brighten our academic qualification that provided us this opportunity to have a practical knowledge of relevant fields.

Also, we thank all our friends for their appraisal and criticism, which has helped us to complete our project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 INTRODUCTION AND SCOPE

Artificial neural networks (ANNs) are a family of models inspired by biological neural networks (the central nervous systems of animals, in particular the brain) and are used to estimate or approximate functions that can depend on a large number of inputs and are generally unknown. Artificial neural networks are generally presented as systems of interconnected "neurons" which exchange messages between each other. The connections have numeric weights that can be tuned based on experience, making neural nets adaptive to inputs and capable of learning.

A neural network for handwriting recognition is defined by a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by a function (determined by the network's designer), the activations of these neurons are then passed on to other neurons. This process is repeated until finally, an output neuron is activated. This determines which character was read.

Recognition can be defined as the methodology of identifying something that you know or interpreting something that you may not know. Character Recognition is a very important area in image processing and pattern recognition fields. The aim of character recognition is to translate human readable characters to machine readable characters. Handwritten character recognition (HCR) has received extensive attention in academic and production fields because of its various application potentials. Some of its potential application areas are Postal automation, Bank cheque processing, automatic data entry, etc.

Optical character recognition (OCR) based systems ease the process of automatic translation of document images into equivalent character codes with huge savings of human energy and cost.

Artificial Intelligence concepts like neural networks are used to perform the work as human mind can do. This explores the idea of how humans recognize text in general and are used to develop machines that simulated this process. A Neural Network acquires knowledge as human brain acquires knowledge from learning process. For recognition of handwritten characters, Script depends on learning process in which back propagation/ neocognitron algorithm takes input from the user. In this learning process, training and testing of characters is done. A library is made in which different characters written by different persons are stored which are used for future comparisons, this library helps in acceptance and rejection of characters.

Deep learning is a class of machine learning algorithms that use a cascade of many layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. The algorithms may be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised), are based on the (unsupervised) learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation. are part of the broader machine learning field of learning representations of data, learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

The neocognitron is a self-organizing neural network which excels at visual pattern recognition. In order to perform character recognition, the methodology must be able to handle variations in the signal. Typically, written characters have not only spatial variations for each user, but also vary greatly in appearance between different individuals. This problem drives the need for a robust character recognition system that can handle these variations. The neocognitron provides a platform for character recognition which is resilient to changes in character appearance and spatial location. The neocognitron is a self-organizing neural network; this means that the features which are extracted are determined during training.

This designed network should detect human faces just as animal's visual system and brain detects various pattern. A specific face can be thought of as a specific pattern. Thus, scope of this project increases manifolds as it can be applied to any pattern. New researches are coming forward regularly in this field. Being a part of Artificial Intelligence, this field is the future of computing and robotics.

1.1 OVERVIEW OF 'MODI' SCIPT

Most of the Indian scripts are distinguished by the presence of matras (or, character modifiers) in addition to main characters as against the English script that has no matras. Thus the challenging part of Indian handwritten character is the distinction between the similar shaped components. A very small variation between two characters or numerals leads to recognition complexity and change in the degree of recognition accuracy. Therefore, algorithms developed for them are not directly applicable to Indian scripts.

'Modi' is a script used to write the Marathi language, which is the primary language spoken in the state of Maharashtra in western India. There are at least two different theories concerning its origin. Modi was an official script used to write Marathi until the 20^{th}century when the *Balbodh* style of the Devanagari script was promoted as the standard writing system for Marathi. Thus, all the written communication in Marathi which happened before 20^{th} century has been written in Modi script. This work which was discovered in the form of letters needs to be translated in more understandable form for the amateur people which aren't well-versed in Modi. And this is what this project intends to achieve. Following figure shows the characters in Modi script against those in Devnagri.

1.2 ORGANIZATION OF THE REPORT

- Chapter 1 :Introduction and scope

  This chapter provides an overview of the basic functionality of the system and describes its scope of expansion

- Chapter 2 : Literature Survey and present scenario

  We present the Literature Survey of the work done in this field so far as well as the present scenario.

- Chapter 3 : System Block Diagram

  Explain in detail the design and development process of the system. Includes system specifications block diagram.

- Chapter 4:System Design

  It explains the Neural Network architecture employed in this project

- Chapter 5 : Tasks

  It includes the tasks undertaken by us to understand the project on fundamental level and displays its result.

- Chapter 6 : Appendix

  Includes important IEEE Papers referred for algorithms of the system.

Chapter 2 LITERATURE SURVEY

Recognition of handwritten characters has been a popular research area for many years because of its various practical application potentials such as reading aid for the blind, bank cheques, vehicle number plates, automatic pin code reading of postal mail to sort. There are many pieces of work towards handwritten recognition of Roman, Japanese, Chinese and Arabic scripts, and various approaches have been proposed by the researchers towards handwritten character recognition. There are many scripts and languages in India and not much research has been done for the recognition of handwritten Indian characters.

Many pieces of work have been done towards the recognition of Indian printed characters and at present Optical Character Recognition (OCR) systems are commercially available for some of the Indian printed scripts. Towards off-line handwritten character recognition of Indian characters only a few attempts have been made. Among off-line handwritten work on Indian scripts, maximum research has been done for Bangla. Systems are available for off-line Bangla handwritten numerals and characters. Also some systems have been developed for unconstrained Bangla handwritten word recognition for Indian postal automation.

Recently on 24thJune 2014 a paper titled, 'Offline Handwritten MODI Character Recognition Using HU, Zernike Moments and Zoning,' was put forward by Mr. Sadanand A. Kulkarni and group in Cornell University. No more work is found on Modi lipi character recognition.

'Neocognitron' architecture was first proposed by Dr. Kunihiko Fukushima, Kansai University, Japan in 1980 in his paper titled, 'Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position.' He went on to publish subsequent papers in years 1988 AND 2003. Neocognitron for handwriting recognition has gain impetus in c.

2.1 PRESENT SCENARIO

As can be seen from the Literature survey, a lot of work has been done in Character Recognition. Character Recognition techniques and algorithms have been put to use in recognizing postal codes, numbers of license number plates of cars etc. However, the efficiency of these algorithms needs improvement.

Moreover, not much work has been done in recognition of 'Modi' script characters. Thus, we were motivated to work towards 'Modi' script in order to make some progress in the field of Handwritten Modi Character Recognition.

MATLAB, software which is at the forefront of making programming easier for engineers, has an Artificial Neural Network toolbox which can be directly used to design an Artificial Neural Network for various applications. However, it provided nothing related to Deep Learning Neural Network until recently.

Recently, in the month of March, MATLAB launched its R2016a version. In this version there is a Deep Learning Neural Network toolbox. This contains encoders and decoders for constructing various hidden layers of the network. We are thinking of using this toolbox as well.

Chapter 3 SYSTEM DEVELOPMENT

3.1 SYSTEM SPECIFICATIONS.

Operating system:

- ✓ PC with Windows XP/Vista/7 OS

Platform / Language used:

- ✓ MATLAB R2012b

3.2 SYSTEM BLOCK DIAGRAM

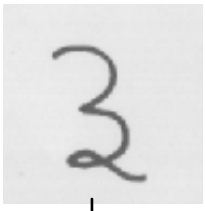

Image Pre-processing

1. Image Resizing
2. Image Binarization (Thresholding)

Input Layer

(16*16)

p -> No. of cell planes

Hidden Layers

(S Layer:-16*16*p)

(V Layer:-16*16*p)

(C Layer:-12*12*p)

(S Layer:-8*8*p)

(V Layer:-8*8*p)

(C Layer:-6*6*p)

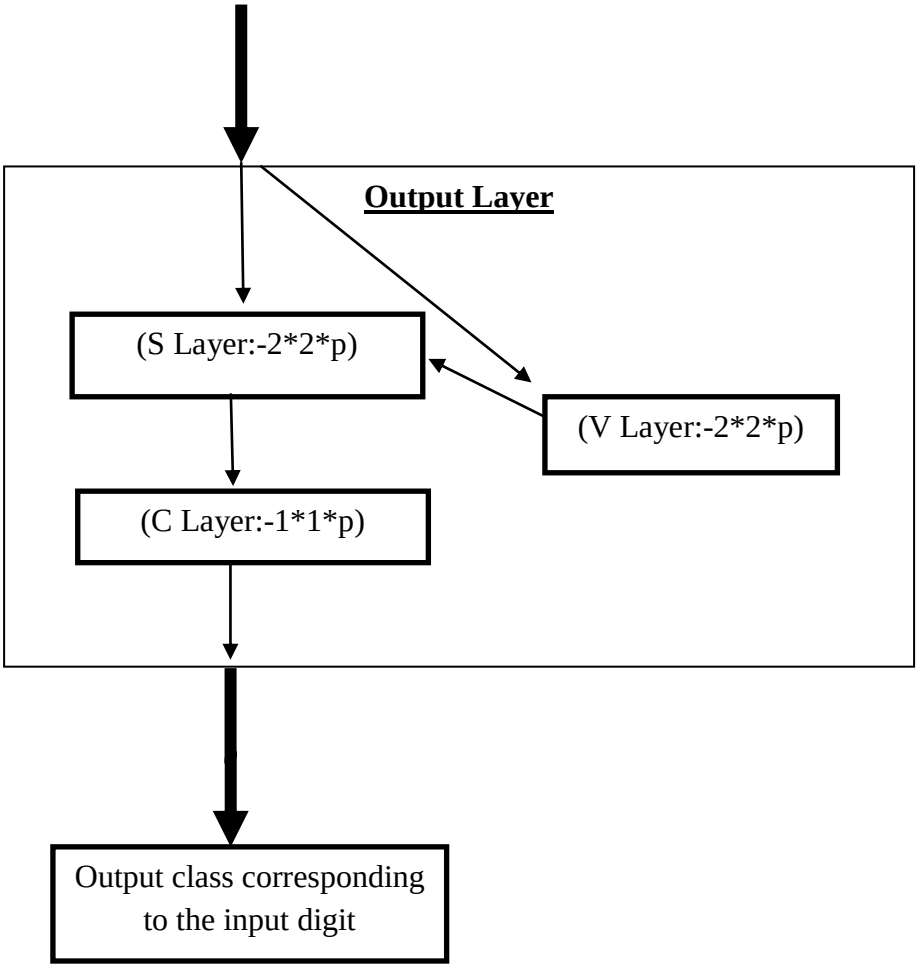

Fig.1.Neural Network implementation using Neocognitron architecture

(System Block Diagram).

3.2. EACH BLOCK IN NEO-COGNITRON ARCHITECTURE

A. IMAGE PRE-PROCESSING:-

The image from the database cannot be directly fed to the network; it has to be preprocessed for many reasons. Reasons may include noise in the image, image being larger than the network could handle, image containing more number of planes than the network requires etc.

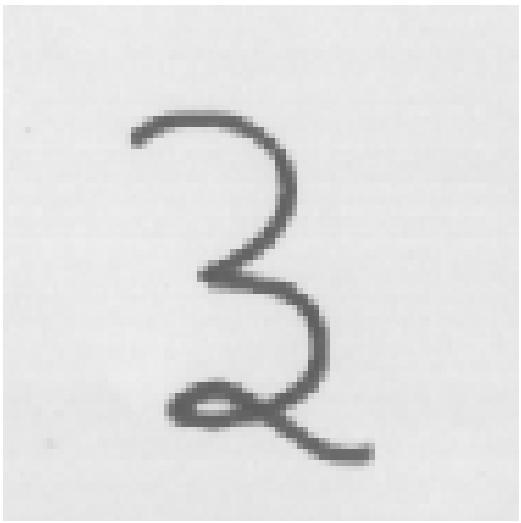

Fig.2. Original Image in the Database.

In our case, the Image in the database is of approximately, '93 * 93 * 3' pixels, resolution, whereas our input layer in the network accepts an Image of size '16*16' pixels (for neo-cognitron). Moreover, the image our network required is binary. Thus, we need to **resize** the image to 16*16 pixels after applying '**thresholding**' process to it.

After applying Averaging operation, the image was resized to 16*16 pixels. This made the image compatible to our network. The Input Layer could accept this image and send it for further processing. But still there would be a problem with accuracy of recognition and time complexity as the intensity range of the image is still from '0 t 255'. If the network has only two intensity levels for processing, then it becomes easier for the network to do its task.

Hence, as said earlier, the resize operation is followed by 'thresholding.' Here the image is converted to Binary form. The overall output after preprocessing is as follows:

Fig.3. Final '16*16' pixels binary image.

B. INPUT LAYER:-

The function of this layer is just to accept the input image and store it until the network is ready to accept it. This layer has one plane of 16*16 cells. These '16*16' cells store the image which is also of the same size. From here the image is sent to first S-layer, in the Hidden Layer domain, which contains variable number of planes (p) of 16*16 cells. The Input Layer sends the input image to each of its plane.

C. HIDDEN LAYERS:-

a. First S-Layer (16*16*p):-

This S-layer consists of p planes of 16*16 cells each. Function of each S-cell is to extract certain feature at the certain position in the input layer (i.e. in its receptive field). For extraction of this feature S-cell uses only information obtained from its connection areas and information about average activity in these areas obtained from corresponding V-cell. All S-cells in one S-plane always extract the same feature. The feature extracted by S-cell is determined by weights for this cell.

The weights related to this cells are 'd', 'a' and 'b'. It is connected to V-layer by 'b' weight, to C-layer by 'd' weight and to the C-layer of previous hidden layer by 'a' weight.

b. First C-Layer (8*8*p):-

This layer consists of p planes consisting of 8*8 cells each. The features extracted by the S-cells are processed by the C-cells. C-plane contains a blurred representation of S-plane

content. Ability of C-cell to compress content of connection area in the certain way is the next consequence of C-cell function. Hence, we can decrease the density of cells in C-layer to the half of density of cells in previous S-layer, as is done in our network.
C-cell function is ensuring of the Neocognitron's tolerance of feature shifts. The C-cell is active only if there is an active S-cell in its connection area. It corresponds to presence of correct feature at the certain position in the input layer. When this feature is shifted to another position another S-cell is activated. If the activated S-cell belongs to the marked connection area again our C-cell remains active.
The weights related to this cells are ‘d’, ‘a’ and ‘c’. It is connected to V-layer by ‘c’ weight, to S-layer by ‘d’ weight and to the S-layer of the next hidden layer by ‘a’ weight.

c. First V-Layer (16*16*p):-

The number of cells in each plane of V-layer is same as that of S-Layer in the same Hidden layer. Each V-cell in the Neocognitron evaluates outputs of C-cells from the certain connection areas from previous C-layer. It inhibits the extraction of features which are extracted in the previous layer to be repeated in the next layer.
The weights related to these cells are ‘b’ and ‘c’. It is connected to S-layer by ‘b’ weight and to the C-layer by ‘c’ weight.

c. Subsequent Hidden Layers:-

The subsequent Hidden layer operates in similar fashion. The S-layer consists of p planes of 8*8 cells each. The C-layer consists of p planes consisting of 6*6 cells each. And V-layer same as S-layer.

C. OUTPUT LAYERS:-

The output of the network can be taken from the C-layer of the Output Layer. This C-layer contains p planes of 1*1 cells. At the end for each input pattern (Modi character), one cell is excited. This cell corresponds to that pattern. Thus that pattern is recognized by the network.

The key to this Character Recognition is updation of the weights ‘a’, ’b’, ’c’, ’d’ and ‘e’.

To sum up the block working:-

First the image (pattern) is read by the system. This image may or may not contain noise. The image is preprocessed to make it favorable for the Neural Network to process it without ambiguity.

The pre-processing stage, which includes averaging, cropping and resizing, image binarization (thresholding) and noise reduction, can make the initial image more suitable for later computation.

After pre-processing the image is accepted by the Neural Network using its input layer.

This layer only stores the image for the network and does not do any computations but may break down the image into the segments of its patterns. The input layer passes these individually divided patterns to the next hidden layer. The number of layers in this stage depends on the whether the network is deep Learning Network or Shallow Learning Network. Deep Learning Network has multiple layers. Finally these subdivided patterns are run through the networks library to check for a match. Then the image goes to the output layer where it is either sent back to the hidden layers for updating their weights (if there is no exact match in the hidden layers) or prepared to be displayed.

Chapter 4 SYSTEM DESIGN

4.1 PREPROCESSING

The starting point of the project is to create a database with all the character images that would be used for training (if supervised learning used) and testing. An organization is expected to provide us the training set. If not, this training set would be collected from the people in the department. The collected documents would be scanned using a document scanner. The digitized images are stored as binary images in BMP format.

Preprocessing includes the steps that are necessary to bring the input data into an acceptable form for feature extraction. The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing stages. Preprocessing stage involves size normalization, image binarization and thresholding and noise reduction. The corresponding objectives of Pre-processing methods are as follows: -

- Image Binarization (Thresholding) - The binarization stage plays an important role in separating the characters from the background accurately. Thresholding is the simplest method of image binarization. From a gray-scale or RGB scale image, thresholding is used to create a binary image.

  If not done, there would be a problem with accuracy of recognition and time complexity as the intensity range of the image is still from '0 t 255' or '0 to '255'*3. If the network has only two intensity levels for processing, then it becomes easier for the network to do its task.

- Resizing - In our case, the Image in the database is of approximately, '93 * 93 * 3' pixels, resolution, whereas our input layer in the network accepts an Image of size '16*16' pixels. Thus Image needs to be resized before given to the network.

4.2 DATA COLLECTION

We have created our own database. This contains handwritten characters written by 8 people. Some of the images of it are as follows.

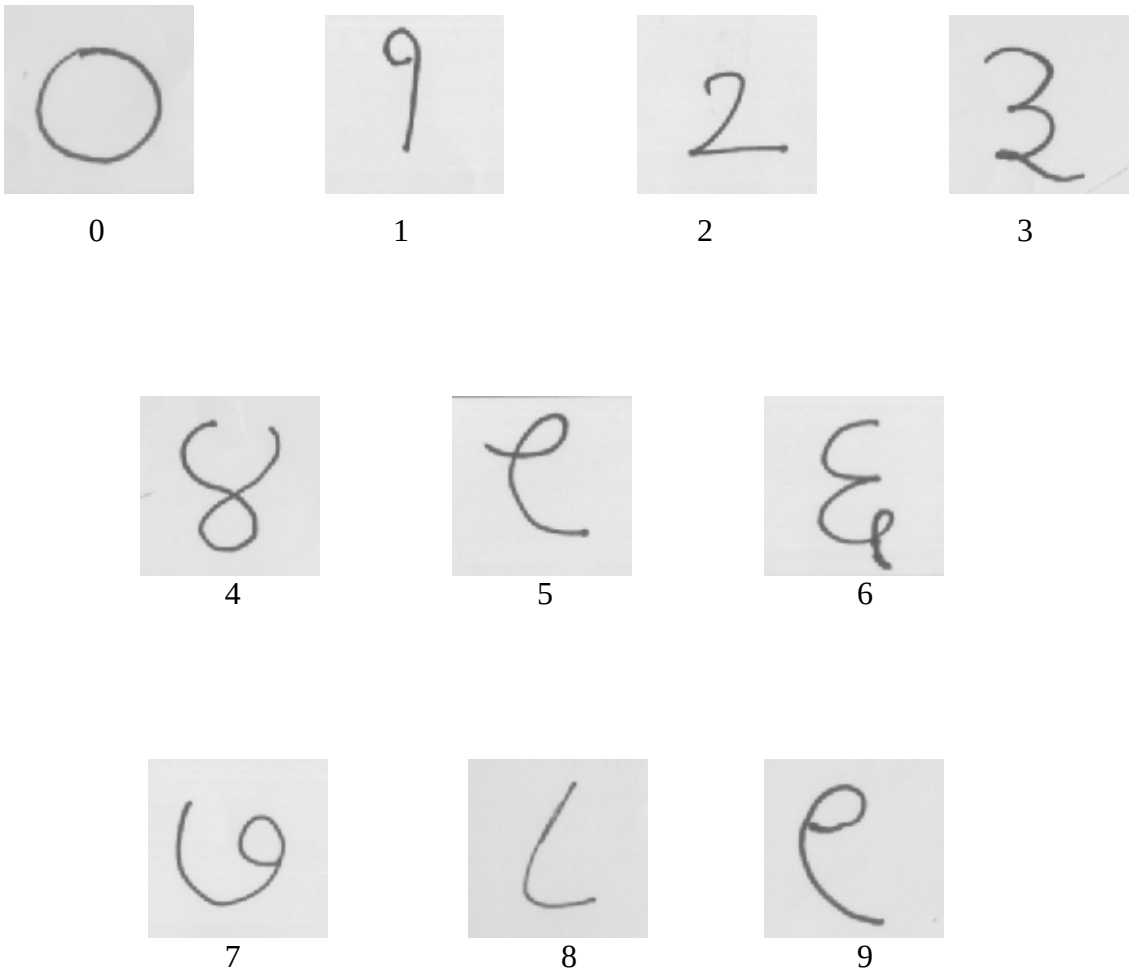

Fig.4. Database generated

These are Modi digits. This database was created by scanning image in high resolution. We also worked on printed database which consisted of numbers from '0' to '4'. Few of these images are:

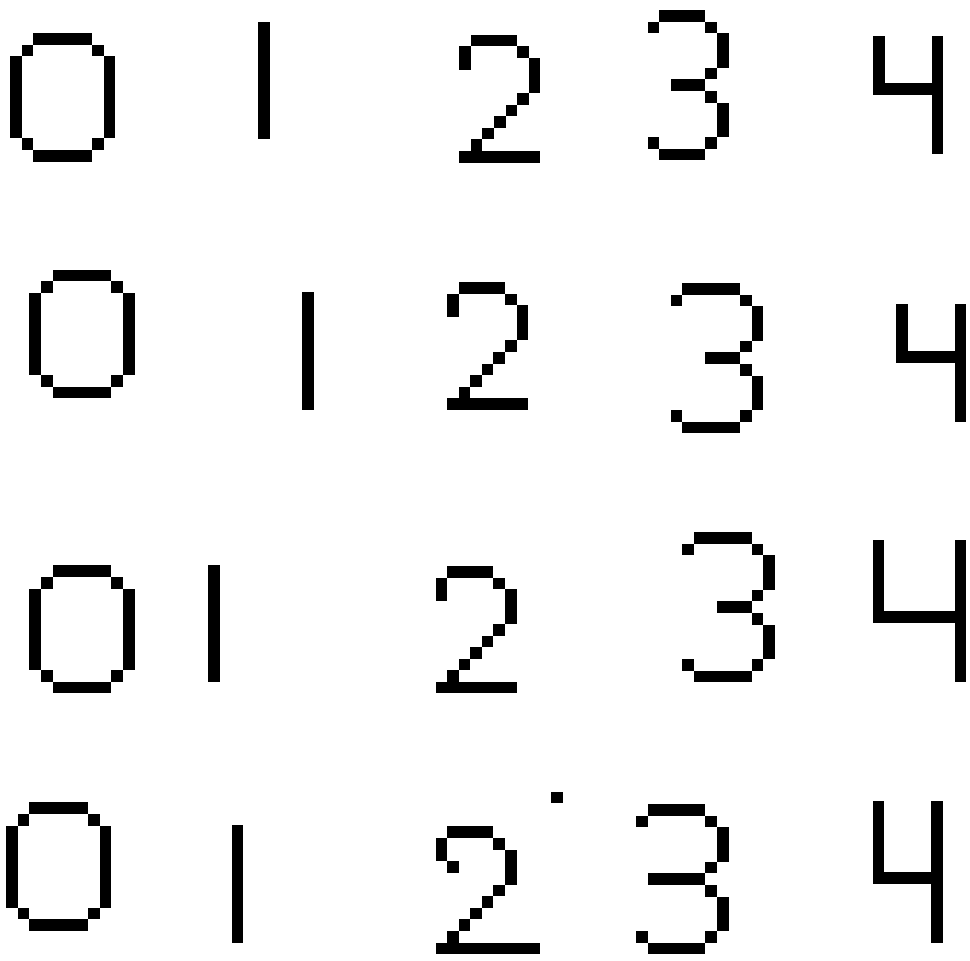

Fig.5. Printed Character Database

This database consists of shifted and scaled versions of same characters.

4.3 NEOCOGNITRON ARCHITECTURE AND STUCTURE OF THE NETWORK:-

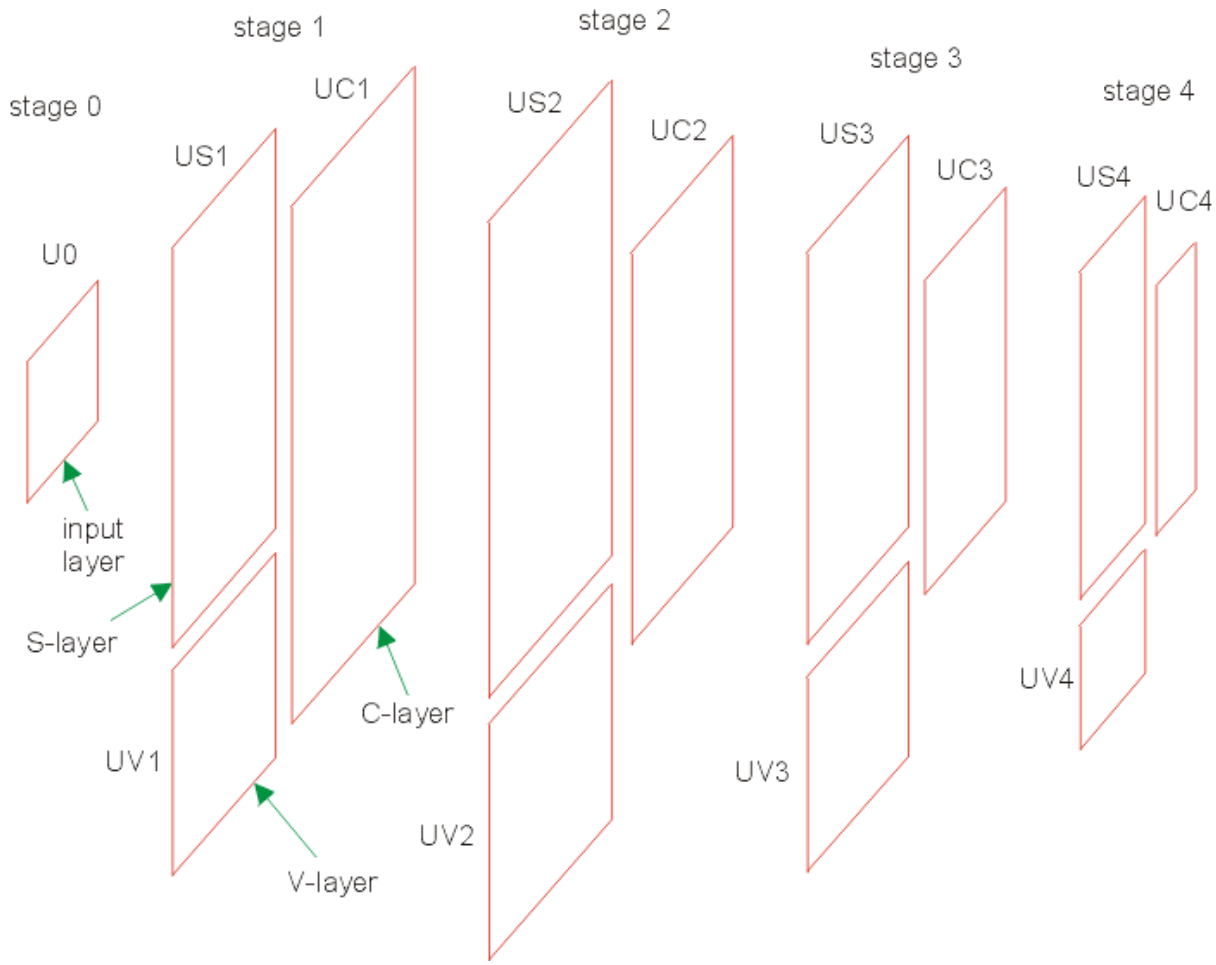

Fig.6.Neocognitron Architecture.

Neocognitron is a hierarchical, multilayered artificial neural network proposed by Kunihiko Fukushima in the 1980s.

It has been used for handwritten character recognition and other pattern recognition tasks, and served as the inspiration for convolutional neural networks. The Neocognitron consists of multiple types of cells, the most important of which are called S-cells and C-cells. The local features are extracted by S-cells, and these features' deformation, such as local shifts, is tolerated by C-cells. Local features in the input are integrated gradually and classified in the higher layers. The network consists of an input layer (photoreceptor array) followed by a cascade connection of a number of modular structures, each of which is composed of two layers of cells connected in a cascade.

This network is self-organized by "learning without a teacher", and acquires an ability to recognize stimulus patterns based on the geometrical similarity (Gestalt) of their shapes

which is affected neither by their position nor by small distortion of their shapes. Total number of stages of the Neocognitron depends on the complexity of recognized patterns. The more complex recognized patterns are, the more stages of hierarchy of extracted features we need and the higher number of stages of the Neocognitron is. All the cells, regardless of their type, process and generate analog values.

Size of cell arrays is the same for all cell planes in one layer and it decreases with increasing of the network stage.

Each V-cell in the Neocognitron evaluates outputs of C-cells (or receptor cells) from the certain connection areas from previous C-layer (or input layer).Size of connection areas is the same for all V-cells and S-cells in one stage of the network and it is determined at construction of the network. Function of each S-cell is to extract the certain feature at the certain position in the input layer (i.e. in its receptive field). For extraction of this feature S-cell uses only information obtained from its connection areas and information about average activity in these areas obtained from corresponding V-cell. All S-cells in one S-plane always extract the same feature. The feature extracted by S-cell is determined by weights for this cell. Each C-cell in the Neocognitron evaluates outputs of S-cells from the certain connection area from one of S-planes from previous S-layer. C-plane contains a blurred representation of S-plane content. Ability of C-cell to compress content of connection area in the certain way is the next consequence of C-cell function. Hence we can decrease the density of cells in C-layer to the half of density of cells in previous S-layer in some cases.

C-cell function is ensuring of the neocognitron's tolerance of feature shifts. The C-cell is active only if there is an active S-cell in its connection area. It corresponds to presence of correct feature at the certain position in the input layer. When this feature is shifted to another position another S-cell is activated. If the activated S-cell belongs to the marked connection area again our C-cell remains active.

The neocognitron is characteristic not only by large number of cells but also by large number of connections. These connections serve for transfer of information between cells in adjoining layers. Particular cell obtains by means of connections information from all cells which are located in its connection areas.

4.4 SELF-ORGANIZATION OF THE NETWORK:-

The self-organization of the neocognitron is performed by means of "learning without a teacher". During the process of self-organization, the network is repeatedly presented with a set of stimulus patterns to the input layer, but it does not receive any other information about the stimulus patterns.

One of the basic hypotheses employed in the neocognitron is the assumption that all the S-cells in the same S-plane have input synapses of the same spatial distribution, and that only the positions of the pre-synaptic cells shift in parallel in accordance with the shift in position of individual S-cells' receptive fields.

In order to make self-organization under the conditions mentioned above, the modifiable synapses are reinforced by the following procedures.

At first, several "representative" S-cells are selected from each S-layer every time when a stimulus pattern is presented. The representative is selected among the S-cells which have yielded large outputs, but the number of the representatives is so restricted that more than one representative are not selected from any single S-plane. The detailed procedure for selecting the representatives is given later on.

All the other S-cells in the S-plane, from which the representative is selected, have their input synapses reinforced by the same amounts as those for their representative. These relations can be quantitatively expressed as follows.

The modifiable synapses *al(k(l - l), v,k1)* and *b1(k1),* which are afferent to the S-cells of the kcth S-plane, are reinforced by the amount shown below:

$$d(al(k(l - l), v,k1))=ql.c(l - l)(v).Uc(l - l(k(l\_ l),n + v)$$

$$d(b1(k1)) = (ql/2). V(cl - l)(n);$$

Where, ql is a positive constant prescribing the speed of reinforcement.

These are the errors in the weights, which are added calculated and updated after every iteration.

The cells in the S-plane from which no representative is selected, however, do not have their input synapses reinforced at all. These are the cells of the first S-layer.

In the initial state, the modifiable excitatory synapses *al(k(l - l), v,k1)* are set to have small positive values such that the S-cells show very weak orientation selectivity, and that the preferred orientation of the S-cells differ from S-plane to S-plane. The initial values of modifiable inhibitory synapses *b1(k1)* are set to be zero.

The procedure for selecting the representatives is given below. It resembles, in some sense, to the procedure with which the reinforced cells are selected in the conventional cognitron.

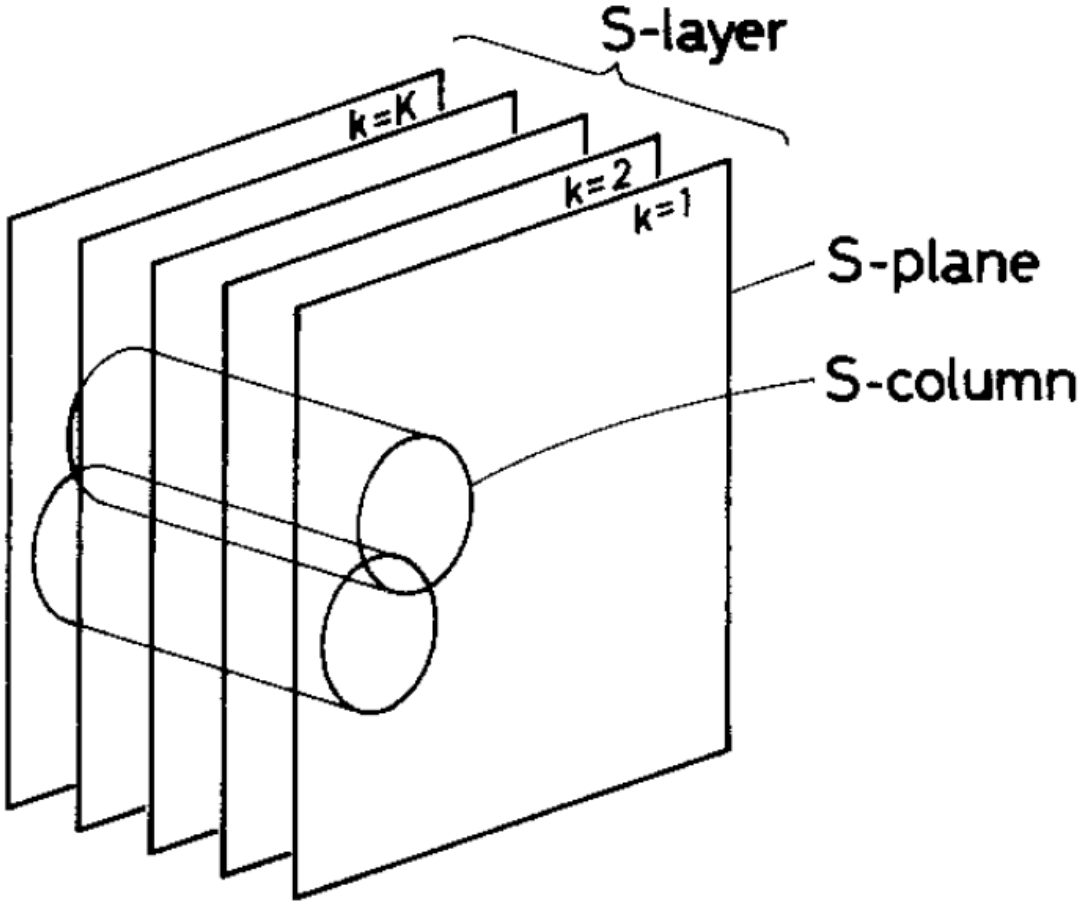

Fig.7. Relation between S-plane and S-columns within an S-layer

At first, in an S-layer, we watch a group of S-cells whose receptive fields are situated within a small area on the input layer. If we arrange the S-planes of an S-layer in a manner shown in Fig. 9, the group of S-cells constitutes a column in an S-layer. Accordingly, we call the group as an "S-column". An S-column contains S-cells from all the S-planes. That is, an S-column contains various kinds of feature extracting cells in it, but the receptive fields of these cells are situated almost at the same position. There are a lot of such S-columns in a single S-layer. Since S-columns have overlapping with one another, there is a possibility that a single S-cell is contained in two or more S-columns.

From each S-column, every time when a stimulus pattern is presented, the S-cell which is yielding the largest output is chosen as a candidate for the representatives.

Hence, there is a possibility that a number of candidates appear in a single S-plane. If two or more candidates appear in a single S-plane, only the one which is yielding the largest output among them is selected as the representative from that S-plane. In case only one candidate appears in an S-plane, the candidate is unconditionally determined as the

representative from that S-plane. If no candidate appears in an S-plane, no representative is selected from that S-plane.

Since the representatives are determined in this manner, each S-plane becomes selectively sensitive to one of the features of the stimulus patterns, and there is not a possibility of formation of redundant connections such that two or more S-planes are used for detection of one and the same feature. Incidentally, representatives are selected only from a small number of S-planes at a time, and the rest of the S-planes are to send representatives for other stimulus patterns.

Equation for Layers :

$$u_{Cl}(k_l, \boldsymbol{n}) = \psi\left[\frac{1+\sum_{v\in D_l} d_l(v)\cdot u_{Sl}(k_l, \boldsymbol{n}+v)}{1+v_{Sl}(\boldsymbol{n})} - 1\right]$$

$$\psi[x] = \varphi\left[\frac{x}{\alpha+x}\right]$$

$$v_{Cl-1}(\boldsymbol{n}) = \sqrt{\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{v\in S_l} c_{l-1}(\boldsymbol{v}) \cdot u^2_{Cl-1}(k_{l-1}, \boldsymbol{n}+\boldsymbol{v})}$$

$$v_{Sl}(\boldsymbol{n}) = \frac{1}{K_l} \cdot \sum_{k_{l=1}}^{K_l} \sum_{v\in D_l} d_l(\boldsymbol{v}) \cdot u_{Sl}(k_l, \boldsymbol{n}+\boldsymbol{v})$$

$$u_{Sl}(k_l, \boldsymbol{n}) = r_l \cdot \varphi\left[\frac{1+\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{v\in S_l} a_l(k_{l-1}, v, k_l)\cdot u_{Cl-1}(k_{l-1}, \boldsymbol{n}+v)}{1+\frac{2r_l}{1+r_l}\cdot b_l(k_l)\cdot v_{Cl-1}(\boldsymbol{n})} - 1\right]$$

$$\varphi[x] = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Equation for fixed weights :

$$c_l(\boldsymbol{v}) = \gamma_l^{|v|}$$

$$d_l(v) = \bar{\delta}_l \cdot \delta_l^{|v|}$$

$$\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{v\in S_l} c_{l-1}(\boldsymbol{v}) = 1$$

4.5 ROUGH SKETCHES OF THE WORKING OF THE NETWORK:-

At first, let us assume that the neocognitron has been self-organized with repeated presentations of stimulus patterns like "A", "B", "C" and so on. In the state when the self-organization has been completed, various feature-extracting cells are formed in the network as shown in Fig. 8. (It should be noted that Fig. 8 shows only an example. It does not mean that exactly the same feature extractors as shown in this figure are always formed in this network.)

Here, if pattern "A" is presented to the input layer Uo, the cells in the network yield outputs as shown in Fig. 10. For instance, S-plane with k 1 = 1 in layer Us1 consists of a two-dimensional array of S-cells which extract /\-shaped features. Since the stimulus pattern"A" contains /\-shaped feature at the top, an S-cell near the top of this S-plane yields a large output as shown in the enlarged illustration in the lower part of Fig. 8.

A C-cell in the succeeding C-plane (i.e. C-plane in layer Ucl with kl = 1) has synaptic connections from a group of S-cells in this S-plane. For example, the C-cell shown in Fig. 8 has synaptic connections from the S-cells situated within the thin-lined circle, and it responds whenever at least one of these S-cells yields a large output. Hence, the C-cell responds to a /\-shaped feature situated in a certain area in the input layer, and its response is less affected by the shift in position of the stimulus pattern than that of presynaptic S-cells. Since this C-plane consists of an array of such C-cells, several C-cells which are situated near the top of this

C-plane respond to the /\-shaped feature contained in the stimulus pattern "A". In layer Ucl, besides this C-plane, we also have C-planes which extract features with shapes like '/-', '-\' and so on.

In the next module, each S-cell receives signals from all the C-planes of layer *Ucl.* For example, the S-cell shown in Fig.8. receives signals from C-cells within the thin-lined circles in layer *Ucl.* Its input synapses have been reinforced in such a way that this S-cell responds only when /\-shaped, /--shaped and -\ shaped features are presented in its receptive field with configuration like /--\ Hence, pattern "A" elicits a large response from this S-cell, which is situated a little above the center of this S-plane. If positional relation of these three features are changed beyond some allowance, this S-cell stops responding. This S-cell also checks the condition that other features such as ends-of-lines, which are to be extracted in S-planes with k 1 =4, 5 and so on, are not presented in its receptive field. The inhibitory cell

Vcl, which makes inhibitory synaptic connection to this S-cell, plays an important role in checking the absence of such irrelevant features.

Since operations of this kind are repeatedly applied through a cascade connection of modular structures of S- and C-layers, each individual cell in the network becomes to have wider receptive field in accordance with the increased number of modules before it, and, at the same time, becomes more tolerant of shift in position of the input pattern. Thus, one C-cell in the last layer *Uc3* yields a large response only when, say, pattern "A" is presented to the input layer, regardless of the pattern's position. Although only one cell which responds to pattern "A" is drawn in Fig. 10, cells which respond to other patterns, such as "B', "C" and so on, have been formed in parallel in the last layer.

From these discussions, it might be felt as if an enormously large number of feature extracting cellplanes become necessary with the increase in the number of input patterns to be recognized. However, it is not the case. With the increase in the number of input patterns, it becomes more and more probable that one and the same feature is contained in common in more than two different kinds of patterns. Hence, each cell-plane, especially the one near the input layer, will generally be used in common for the feature extraction, not from only one pattern, but from numerous kinds of patterns. Therefore, the required number of cell-planes does not increase so much in spite of the increase in the number of patterns to be recognized.

Viewed from another angle, this procedure for pattern recognition can be interpreted as identical in its principle to the information processing mentioned below.

That is, in the neocognitron, the input pattern is compared with learned standard patterns, which have been recorded beforehand in the network in the form of spatial distribution of the synaptic connections. This comparison is not made by a direct pattern matching in a wide visual field, but by piecewise pattern matchings in a number of small visual fields. Only when the difference between both patterns does not exceed a certain limit in any of the small visual fields, the neocognitron judges that these patterns coincide with each other.

Such comparison in small visual fields is not performed in a single stage, but similar processes are repeatedly applied in a cascade. That is, the output\ from one stage is used as the input to the next stage. In the comparison in each of these stages, the allowance for the shift in pattern's position is increased little by little. The size of the visual field (or the size of the receptive fields) in which the input pattern is compared with standard patterns, becomes larger in a higher stage. In the last stage, the visual field is large enough to observe the whole information of the input pattern simultaneously.

Even if the input pattern does not match with a learned standard pattern in all parts of the large visual field simultaneously, it does not immediately mean that these patterns are of different categories. Suppose that the upper part of the input pattern matches with that of the standard pattern situated at a certain location, and that, at the same time, the lower part of this input pattern matches with that of the same standard pattern situated at another location. Since the pattern matching in the first stage is tested in parallel in a number of small visual fields, these two patterns are still regarded as the same by the neocognitron. Thus, the neocognitron is able to make a correct pattern recognition even if input patterns have some distortion in shape.

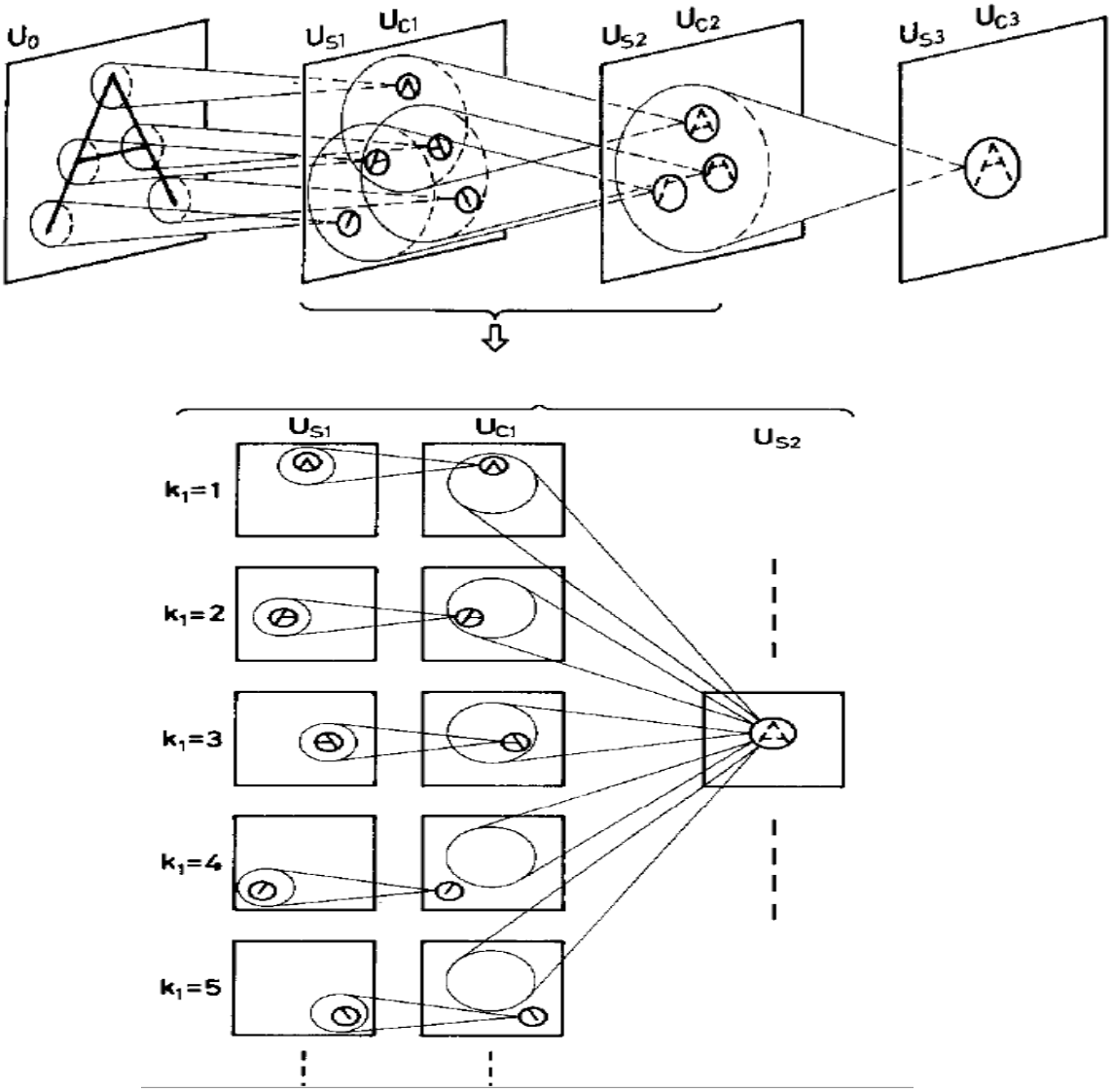

Fig.8. An example of interconnections between cells and the response of the cells after completion of self-organization.

4.6 DEEP LEARNING NEURAL NETWORKS:-

Deep learning (deep machine learning, or deep structured learning, or hierarchical learning, or sometimes DL) is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers with complex structures or otherwise, composed of multiple non-linear transformations. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction.

Following is a figurative comparison between Deep Learning Neural Networks and Shallow Learning Neural Networks.

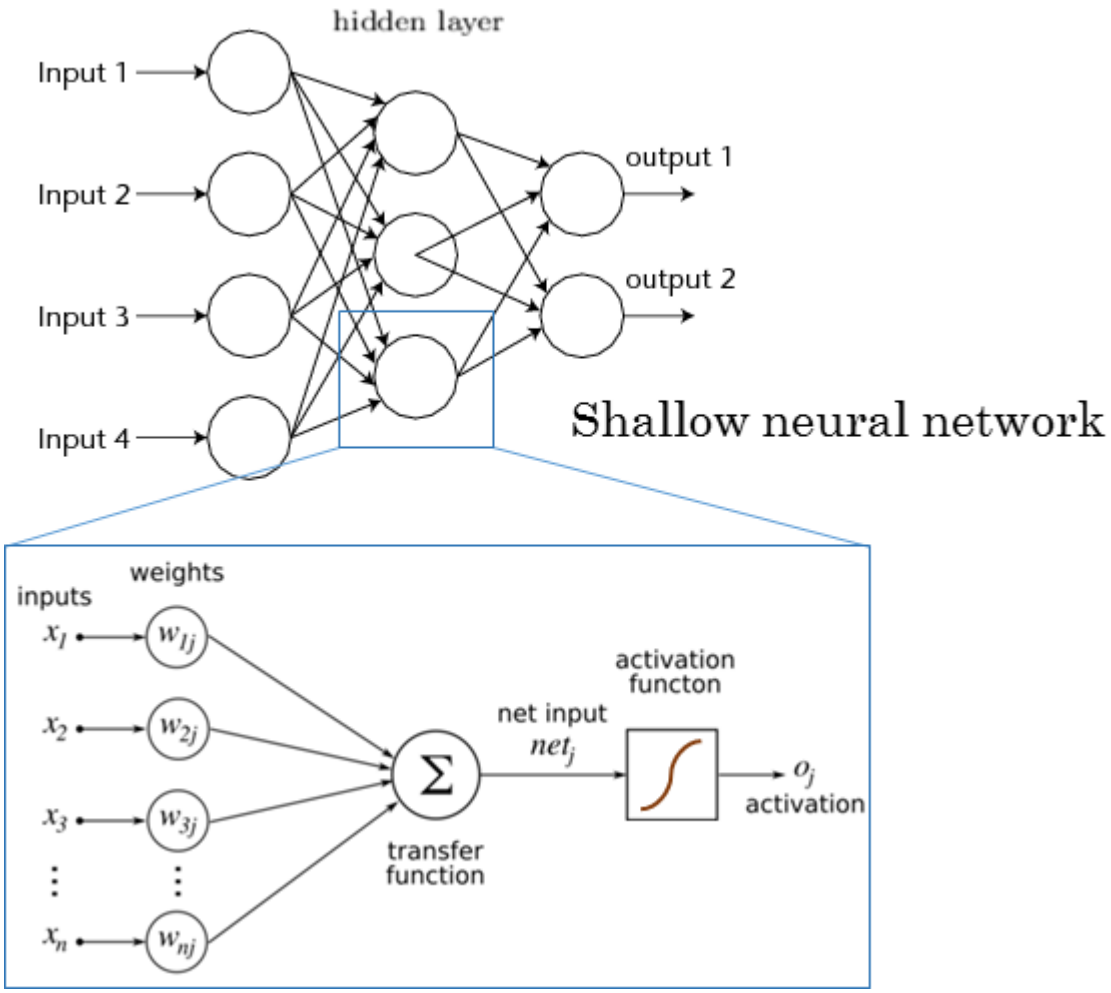

Fig.9.Shallow Learning.

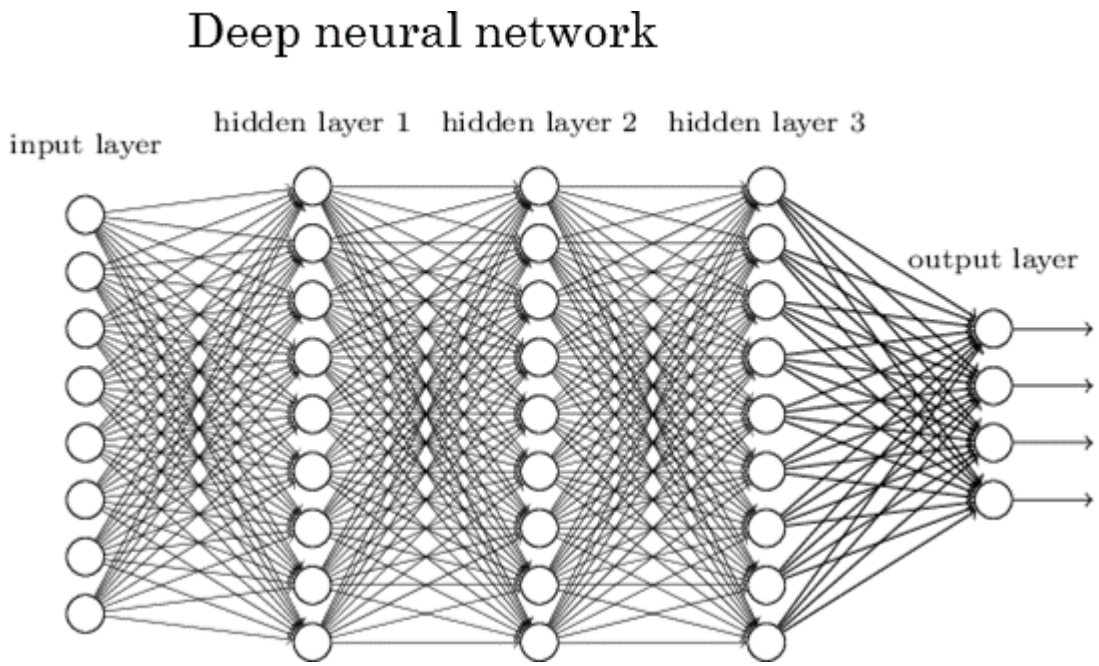

Fig.10.Deep Learning.

Deep learning is part of a broader family of machine learning methods based on learning representations of data. An observation (e.g., an image) can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of particular shape, etc. Some representations are better than others at simplifying the learning task (e.g., face recognition or facial expression recognition) from examples. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for unsupervised or semi-supervised feature learning and hierarchical feature extraction.

Deep Learning Networks have more number of layers, for instance, if we're doing visual pattern recognition, then the neurons in the first layer might learn to recognize edges, the neurons in the second layer could learn to recognize more complex shapes, say triangle or rectangles, built up from edges. The third layer would then recognize still more complex shapes. And so on. These multiple layers of abstraction seem likely to give deep networks a compelling advantage in learning to solve complex pattern recognition problems.

More number of layers provides them an extra computational and processing ability. Thus each pattern will be processed more number of times. This ensures a more accurate result. More complex the pattern becomes more number of layers can be employed.

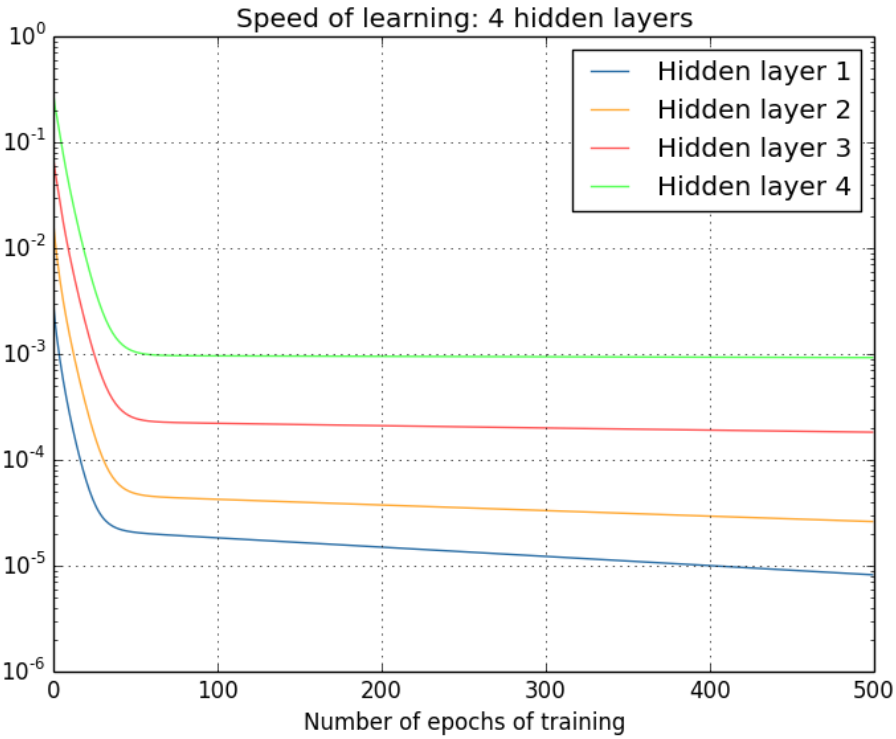

Fig.11. Speed of learning with multiple hidden layers

Deep learning algorithms are based on distributed representations. The underlying assumption behind distributed representations is that observed data are generated by the interactions of factors organized in layers. Deep learning adds the assumption that these layers of factors correspond to levels of abstraction or composition. Varying numbers of layers and layer sizes can be used to provide different amounts of abstraction.

Deep learning exploits this idea of hierarchical explanatory factors where higher level, more abstract concepts are learned from the lower level ones. These architectures are often constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features are useful for learning For supervised learning tasks, deep learning methods obviate feature engineering, by translating the data into compact intermediate representations akin to principal components, and derive layered structures which remove redundancy in representation Many deep learning algorithms are applied to unsupervised learning tasks. This is an important benefit because unlabeled data are usually more abundant than labeled data.

Examples of deep structures that can be trained in an unsupervised manner are neural history compressors and deep belief networks There are huge number of variants of deep architectures. Most of them are branched from some original parent architectures. It is not always possible to compare the performance of multiple architectures all together, because they are not all evaluated on the same data sets. Deep learning is a fast-growing field, and new architectures, variants, or algorithms appear every few weeks.

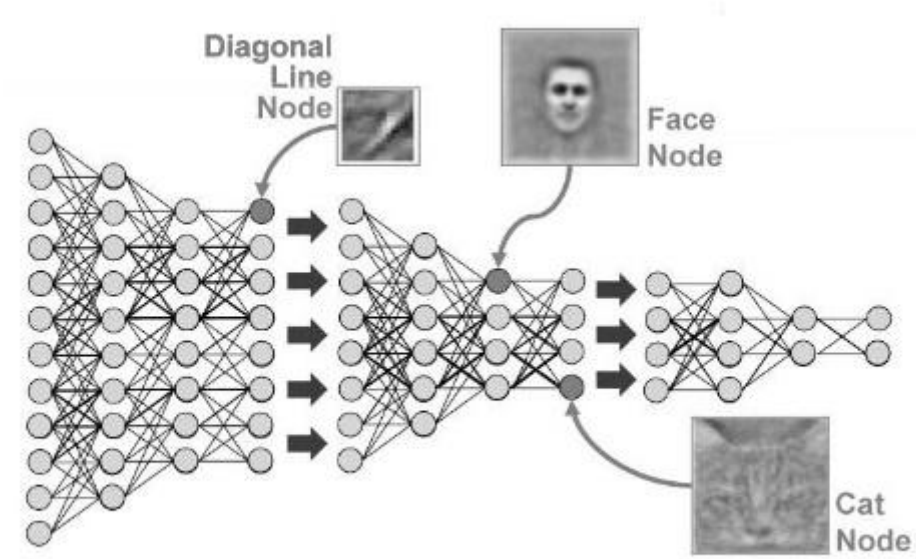

Fig.12. Nodes in Deep Learning Network

5.1 TASKS:-

To understand the concepts of the project, we undertook few tasks. These tasks were:-

1. To thin a given character in an image. - This is important for getting the exact character and removing any ambiguity.
2. To take Vertical and Horizontal plots of number of pixels in an image:- These plots can be seen as a feature. As plots of every character will vary from that of the other, these task helped us to understand feature extraction.
3. Apply High pass and Low pass filters:- Here we applied High and Low-pass filters successively to the same image till the point that the pattern in the image started losing connectivity and got subdivided into small patterns. These patterns provide an insight for ***Filter based Approach for Neocognitron.***

For all these tasks, platform used was MATLAB.

5.2 RESULTS:-

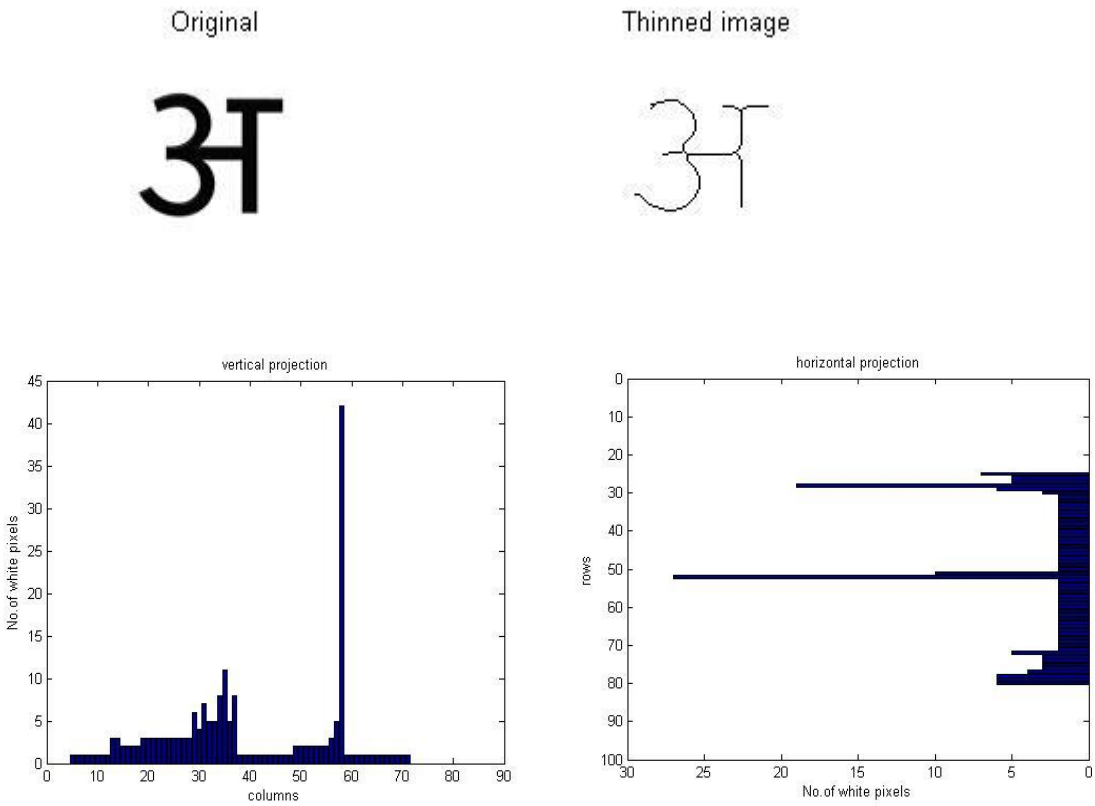

Fig13.1: Result set 1:- Image Thinning and plots (Printed Image).

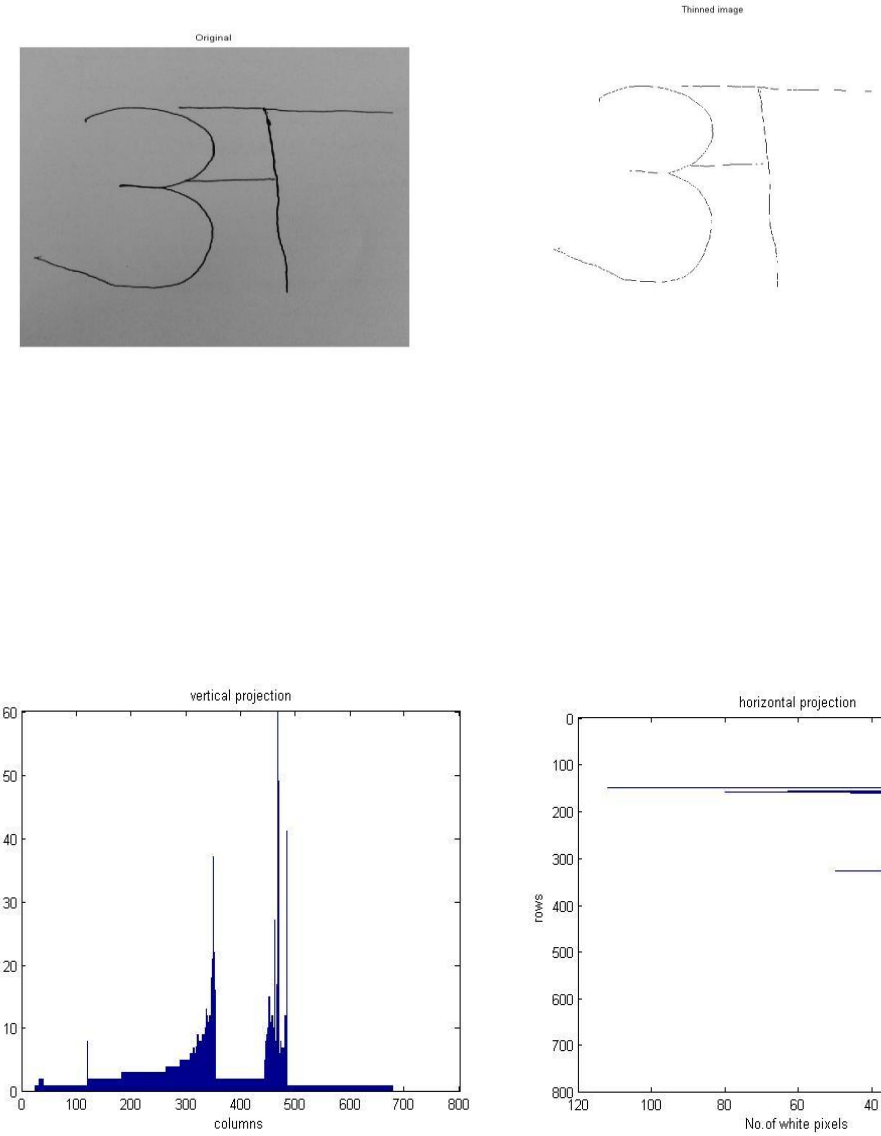

Fig13.2: Result set 2:- Image Thinning and plots (Handwritten Image).

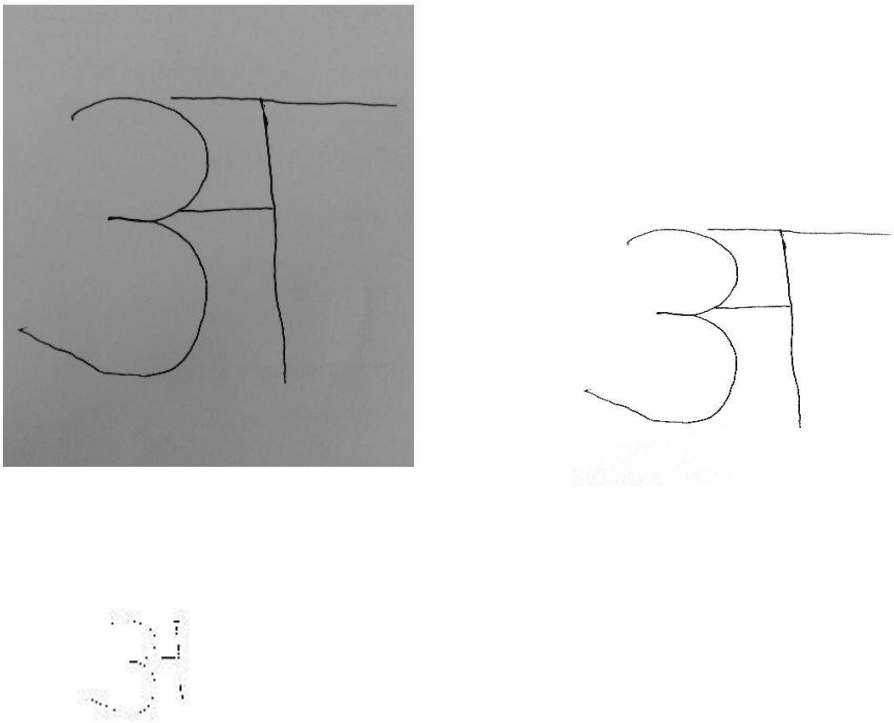

Fig.13.3: Result set 3:- Down sampling and High-pass Filtering

Fig.13.Results from tasks.

A. Starters

Starters are those pixels with one neighbour 1 in the character skeleton. Before character traversal starts, all the starters in the particular zone is found and is populated in a list.

B. Intersections

The definition for intersections is somewhat more complicated. The necessary but insufficient criterion for a pixel to be an intersection is that it should have more than one neighbour. A new property called true neighbours is defined for each pixel. Based on the number of true neighbours for a particular pixel, it is classified as an intersection or not.

C. Features

1. Number of horizontal lines
2. Number of vertical lines
3. Number of right diagonal line
4. Number of left diagonal lines
5. Normalized length of horizontal lines
6. Normalized length of vertical lines
7. Normalized length of right diagonal lines
8. Normalized length of left diagonal lines
9. Normalized area of skeleton
10. Direction of curve
11. Extent
12. Eccentricity

7.1 RESULTS

The set of handwritten Modi digits and digits from ‘0 to 4’ are used to obtain recognition tasks using supervised algorithms like Back Propagation Network and Support Vector Machine and unsupervised Neo-cognitron architecture.

For a start, we have been testing the system only with the digits from ‘0 to 4’. These digits’ database was obtained from various implementations of Neocognitron available. The data set was partitioned into four groups of five digits each. The system was trained for a group and then digits from the other group were given for testing purpose.

This dataset was also used in BPN and SVM networks. The obtained efficiencies and confusion matrix are then compared for all three networks. The confusion matrices can thus be given as:

FOR PRINTED	EFFICIENCY
BPN	100%
SVM	100%
NEO-COGNITRON	75%

FOR MODI	EFFICIENCY
BPN	92%
SVM	88%
NEO-COGNITRON	52.5%

Table.1.Overall Efficiency

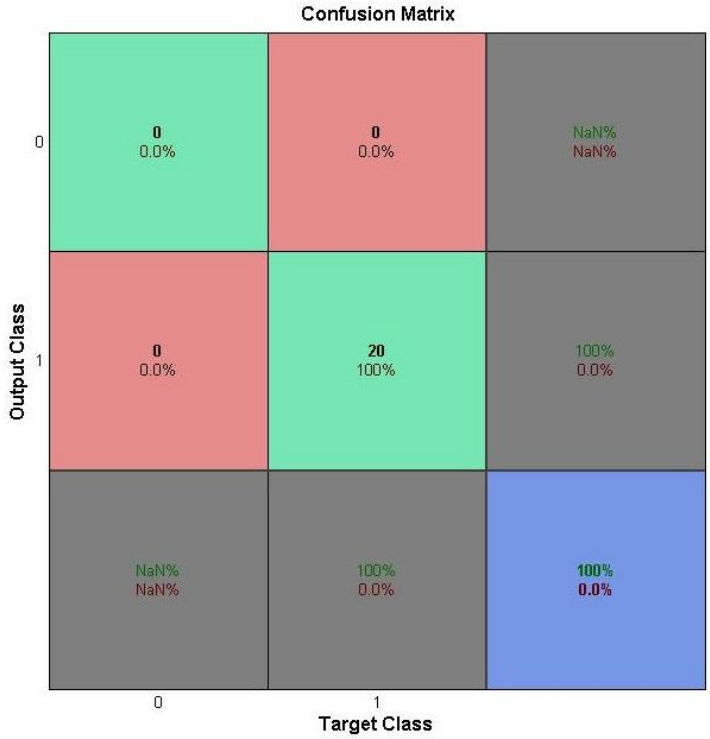

Table.2. BPN Confusion matrix for Printed digits

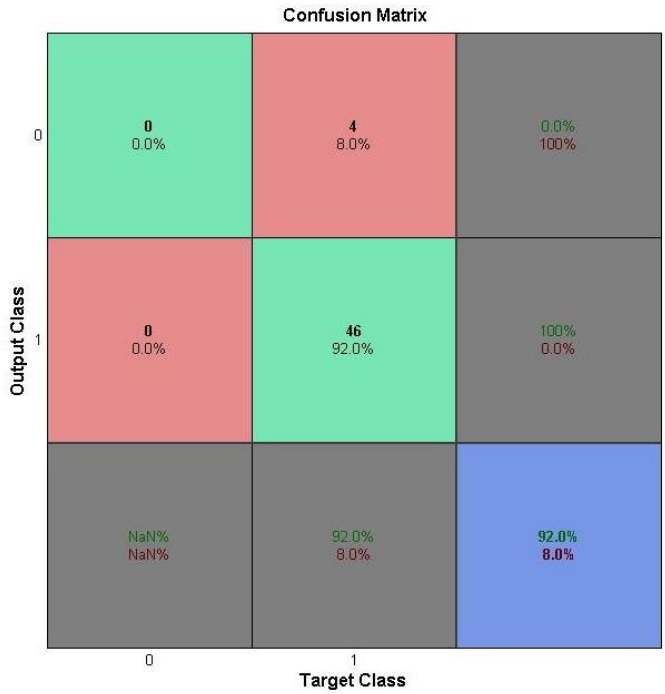

Table.3. BPN Confusion matrix for Modi digits

		CLASSES					CLASSIFICATION RATE
		0	1	2	3	4	
PATTERNS	0	4	0	0	0	0	100.00%
	1	0	4	0	0	0	100.00%
	2	0	0	4	0	0	100.00%
	3	0	0	0	4	0	100.00%
	4	0	0	0	0	4	100.00%
			OVERALL CLASSIFICATION RATE				***100.00%***

Table.4. SVM Classification matrix for Printed digits

		CLASSES										UNCLASSIFIED	CLASSIFICATION RATE
		0	1	2	3	4	5	6	7	8	9		
PATTERNS	0	5	0	0	0	0	0	0	0	0	0	0	100.00%
	1	0	5	0	0	0	0	0	0	0	0	0	100.00%
	2	0	0	4	0	0	0	0	0	0	0	1	80.00%
	3	0	0	0	5	0	0	0	0	0	0	0	100.00%
	4	0	0	0	0	4	0	0	1	0	0	0	80.00%
	5	0	0	0	0	0	4	0	0	0	0	1	80.00%
	6	0	0	0	1	0	0	4	0	0	0	0	80.00%
	7	0	0	0	0	0	0	0	5	0	0	0	100.00%
	8	0	1	0	0	0	0	0	0	4	0	0	80.00%
	9	0	0	0	0	0	1	0	0	0	4	0	80.00%
										OVERALL CLASSIFICATION RATE			***88.00%***

Table.5. SVM Classification matrix for Modi digits

r=[4.81,1.44,2.57] q=[0.1,9.6,13.94] alpha=0.478													M.F=0.4
	CLASSES												Classification Rate
		1	2	3	4	9	10	11	12	13	14	15	16	
PATTERNS	0	0	0	0	3	1	0	0	0	0	0	0	0	75.00%
	1	2	0	0	0	0	0	0	0	0	0	0	2	50.00%
	2	0	0	0	0	0	0	0	0	0	4	0	0	100.00%
	3	0	0	0	1	0	0	2	1	0	0	0	0	50.00%
	4	0	0	4	0	0	0	0	0	0	0	0	0	100.00%
										Overall Classification Rate				75.00%

Table.6. Neocognitron Classification Matrix for Printed Digits Case 1

r=[4.87,1.44,1.57] q=[0.1,9.6,13.94] alpha=0.45													M.F=2
	CLASSES												Classification Rate
		1	2	3	4	5	6	7	8	9	10	12	13	
PATTERNS	0	0	0	0	0	3	0	0	0	0	0	0	1	75.00%
	1	0	0	0	0	0	0	1	3	0	0	0	0	75.00%
	2	0	0	0	2	2	0	0	0	0	0	0	0	50.00%
	3	0	0	1	0	0	0	0	2	0	0	0	1	25.00%
	4	0	0	0	0	0	0	0	0	0	4	0	0	100.00%
										Overall Classification Rate				65.00%

Table.7. Neocognitron Classification Matrix for Printed Digits Case 2

		CLASSES															Classification Rate
		1	2	4	5	6	10	11	13	16	19	21	23	24	26	30	
PATTERNS	0	0	0	0	0	0	0	0	7	0	0	0	0	0	1	0	87.50%
	1	0	0	0	0	0	1	0	0	0	6	0	1	0	0	0	75.00%
	2	0	0	0	0	0	0	2	0	3	0	2	0	1	0	0	37.50%
	3	0	0	0	0	0	5	2	0	0	0	1	0	0	0	0	12.50%
	4	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	100.00%
	5	0	5	0	0	0	1	1	0	1	0	0	0	0	0	0	62.50%
	6	0	3	0	2	0	0	3	0	0	0	0	0	0	0	0	25.00%
	7	2	0	0	0	0	3	0	1	0	1	0	0	0	0	1	37.50%
	8	0	1	1	1	2	0	2	0	0	0	0	0	0	1	0	25.00%
	9	0	0	1	0	0	0	2	0	0	0	0	0	0	0	5	62.50%
												Overall Classification Rate					***52.50%***

Table.8. Neocognitron Classification Matrix for Modi Digits

		CLASSES										CLASSIFICATION RATE
		2	4	10	11	13	16	19	21	26	30	
PATTERNS	0	0	0	0	0	7	0	0	0	1	0	87.50%
	1	0	0	1	0	0	0	6	1	0	0	75.00%
	4	0	0	0	8	0	0	0	0	0	0	100.00%
	5	5	0	1	1	0	1	0	0	0	0	62.50%
	9	0	1	0	2	0	0	0	0	0	5	62.50%
									OVERALL CLASSIFICATION RATE			***62.00%***

Table.9. Neocognitron Classification Matrix for 5 Modi Digits

7.2 COMPLEXITIES INVOLVED

I. We tried 3 different approaches: Procedural, Object oriented and Functional.

For procedural approach, length of the code obtained was much large so as to put it correctly.

Matlab supports Object Oriented Programming but has difficulty in implementation due to incomplete knowledge.

In functional approach, we created different functions. Problems were identified as Matlab uses 'pass by value' for weight updation. Hence, we used handle objects.

II. Random Weights

Initialization of weights was done to a random and very small value. Thus, took a long time to update the weights to an optimized value.

III. Parameter

The reinforcement parameter 'q', selectivity parameter 'r' and saturation parameter 'alpha' are initialized manually, and no specific method is available to obtain them.

These parameters need to be initialized to approx. optimum values for good performance.

IV. Number of cell planes

Performance of the network depends on number of cell planes. More the number of planes, more features can be extracted. A slight variation in the same character can be detected as a different feature.

Chapter 8 CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

In this project, we designed a Deep **Learning Neural Network using Neocognitron** architecture with **unsupervised learning.** We also used the same database to obtain recognition using Back Propagation Network and Support Vector Machine, which use supervised learning. During the research we found that Neocognitron architecture is best suited for any Pattern (character) Recognition the reason being it is independent of the orientation of the pattern. Neocognitron was designed to resemble working of human visual system.

The efficiency thus obtained in the three networks are given as

- For Printed Database : Digits 0 - 4
  $Efficiency_{(BPN)} >= Efficiency_{(SVM)} > Efficiency_{(NEO\text{-}COGNITRON)}$

- For Modi Database: Digits 0 - 9
  $Efficiency_{(BPN)} > Efficiency_{(SVM)} > Efficiency_{(NEO\text{-}COGNITRON)}$

  The Neocognitron architecture requires fine tuning of the parameters to obtain considerable efficiency. Proper adjustment of reinforcement (q), selectivity(r) and saturation (alpha) parameters is critical for performance of the network, especially 'r' and 'alpha'.
  If 'r' is less, then features are not extracted properly.
  If 'r' is large, then network's ability for generalization deteriorates.
  Large 'alpha' overwhelms the output of C cells.
  Large 'q' is equivalent to large learning rate and the weights saturate faster.

Advantages of Neocognitron over other networks:

1. Requires less number of patterns (usually one) for training.
2. Less training time for large database.
3. No external feature extraction required.

8.2 FUTURE SCOPE

The project work can be extended to test the network using different training methods like supervised learning and thus aim to achieve the best efficiency not only in terms of accuracy of recognition but also in terms of the time required to train the network. This process may also include implementing network pruning algorithm, which reduces the size of the network thus reducing the chance of over fitting the training data, and removing the less used weights, thus striking a compromise between accuracy and training time. The network can even be tried for human face detection.

Optimizing of parameters can be used for general application. Also**, C layer** can further be made **adaptive** to improve performance.

- **Nicholas J. Conn, "Character recognition using a neocognitron", Electrical and Microelectronic Engineering Department, Rochester Institute of Technology, Rochester, NY 14623, USA - 2012**

- **K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," Biological Cybernetics, vol. 36, pp. 193-202, 1980.**

- **Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", Third Edition, - Pearson Education**

- **Principles of Soft Computing , S. N. Sivanandam, S. N. Deepa, John Wiley & Sons, 2007**

- **"Modified neocognitron for improved 2-D pattern recognition", C. N. S. Ganesh Murthy and Y.V. Venkatesh**

- **http://www.kiv.zcu.cz/studies/predmety/uir/NS/Neocognitron/en/index.html**

- **"Pattern Recognition in the Neocognitron is Improved by Neuronal Adaptation", A van Ooyeny and B Nienhuis**