AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

A Project Phase-I

Report On

HANDWRITTEN CHARACTER RECOGNITION SYSTEM

Submitted in the partial fulfillment of the requirements of

Bachelor of Engineering in

Electronics and Telecommunication

By

Rishikesh Nikam	**17ET040**
Prathmesh Pardeshi	**17ET042**
Rakesh Sawant	**17ET043**

Under the guidance of

Dr. D S Bormane

co-guide: Prof. R Itkarkar

सत्याला मरण नाही

Department of Electronics and Telecommunication Engineering
All India Shri Shivaji Memorial Society's College of Engineering, Pune-01

Affiliated to SPPU Pune

A.Y 2020-21

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

CERTIFICATE

This is to certify that the Project titled "**Handwritten Character Recognition System"**

Submitted by,

Mr. RISHIKESH NIKAM	**B150213040**
Mr. RAKESH SAWANT	**B150213046**
Mr. PRATHMESH PARDESHI	**B150213041**

is a bonafide record of the Project Phase-II work carried out by them towards the partial fulfillment of the requirements of Savitribai Phule Pune University, for the award of Bachelor of Electronics & Telecommunication Engineering under my supervision and guidance.

Signature of the Guide:

Name of the Guide: Dr. D S Bormane
Co-guide: Prof. R R Itkarkar

Prof. R R Itkarkar
Project coordinator

Dr. D G Bhalke
HOD- E & TC

Date:
Place**:** Pune

This report has been examined as per Savitribai Phule Pune University requirements at **AISSMS COE, Pune-01** on ________________

Signature:

Signature:

Name of Internal Examiner

Name of External Examiner

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and We are extremely privileged to have got this all along the completion of Plase-II of our project.

We respect and thank Dr. D S Bormane, for giving us all support and guidance which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs .

We owe our deep gratitude to our project co-guide and project co-ordinator Prof. R R Itkarkar, who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system.

We are grateful to Dr D G Bhalke, Head of E & TC Engineering Department and all the teaching and non-teaching staff members of E& TC Engineering Department for their direct or indirect help in completion of the phase-II of project.

RISHIKESH NIKAM	B150213040
RAKESH SAWANT	B150213046
PRATHMESH PARDESHI	B150213041

TABLE OF CONTENTS

HANDWRITTEN CHARACTER RECOGNITION

ABSTRACT

Handwriting recognition(HWR) is the ability of a machine to receive and interpret handwritten input from multiple sources like paper documents, photographs, touch screen devices etc. The main aim of this project is to design expert system for, Handwritten Character Recognition that can effectively recognize a particular English character using the Neural Network approach. Neural networks are able to learn features from analyzing a dataset, and then classify an unseen image based on weights. Using Convolutional Neural Networks(CNN) layers it is possible to perform this with the world largest dataset of Handwritten characters- EMNIST dataset. This model followed by softmax classifier gives an accuracy of 93% for real time inputs through webcam.

CHAPTER 1: INTRODUCTION

Handwriting has become an essential part of our daily life. It absorbs a number of patterns in it like signature verification, writer identification. Nowadays it is possible for a computer to extract all these patterns by recognizing character(English) given as in the input image. Once the input image of the character is given to proposed system, then it will recognize input character which is given in image. This has got many applications in various fields like-as reading postal addresses, bank check amounts and forms, to restore ancient scripts by digitizing them. Online handwritten character recognition has also been researched extensively and has got many unique applications like in the forensic domain.

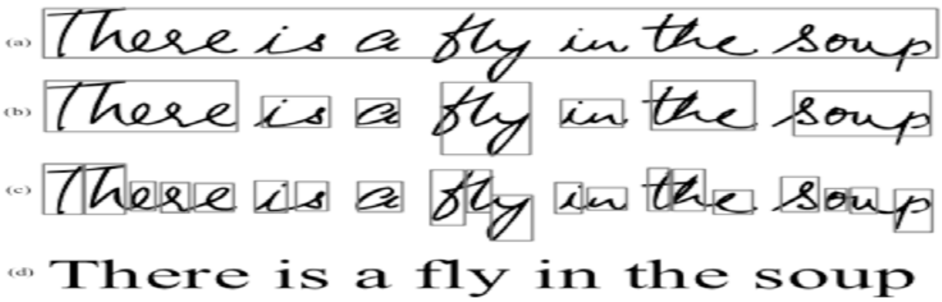


Fig 1. Example of character segmentation[2]

The technique by which a computer system can recognize characters and other symbols written by hand in natural handwriting is called handwriting recognition system. Handwriting recognition is classified into offline handwriting recognition and online handwriting recognition [3]. If handwriting is scanned and then understood by the computer, it is called offline handwriting recognition. In case, the handwriting is recognized while writing through touch pad using stylus pen, it is called online handwriting recognition. From the classifier perspective, character recognition systems are classified into two main categories i.e. segmentation free (global) and segmentation based (analytic). The segmentation free also known as the holistic approach to recognize the character without segmenting it into subunits or characters. Each word is represented as a set of global features, e.g. ascender, loops, cusp, etc.

Whereas segmentation-based approach each word/ligature is segmented into subunits[4]. CNN method comes under the segmentation free category.

For the image recognition problem such as handwritten classification, it is very important to make out how data are represented in images. The data here is not the row pixels, but should be the features of images which has high level representation. For the problem of handwritten digit recognition, the digit's structure features should be first extracted from the strokes. Then the extracted features can be used to recognize the handwritten digit. The high performance of large-scale data processing ability is the core technology in the era of big data.[5]

Deep learning is a multilayer neural network learning algorithm which emerged in recent years. Applications of deep learning to various problems have been the subject of a number of recent studies ranging from image classification and speech recognition to audio classification[4]. It has brought a new wave to machine learning, and making artificial intelligence and human-computer interaction advance with big strides. Deep Learning algorithms are highly efficient in image recognition tasks such as MNIST digit recognition. The three mainstream algorithms of deep learning; the Convolutional Neural Network (CNN), the Deep Belief Network (DBN) and the Deep Neural Network (DNN) [5]

Optical Character Recognition (OCR) is one of the leading dynamic applications of an image classification and gaining additional interest due to its numerous applications. The applications include automatic postal card sorting , digital signature verification, automatic processing of bank cheques , processing of historical documents and automatic handwritten text detection in classroom teaching.[6]

Many researchers worked on HCR and also achieved good recognition accuracy; however, it is impossible to get 100% accuracy to any character recognition system in real-life example. Here, reliability is also very important than high recognition rate. In many real-life applications which are mentioned above required high reliability to reduce losses. The higher complexity of handwritten character versus in print character can be attributed to: i) Presence of clatter during collection process for while writing, ii) Individual's writing style causing significant discrepancy and variation in strokes of a character, iii) Influence of a situation resulting variation in Individual's handwriting on different occasions, iv) Form and shape resemblance and v) Combinations with matra and composite letters add further intricacies. Due to the above stated reasons, it is not always feasible option to implement a common classifier to classify handwritten characters written by various writers.[10]

The most widely used neural network for deep learning work is the convolutional neural network, which converts unstructured image data into structured object tag data [5]. Generally, the working principle of a CNN is as follows: first, the convolution layer scans the input image to generate a feature vector; second, the activation layer determines which feature should activate the image under inference; third, the pooling layer reduces the size of the feature vector; and finally, a fully connected layer connects each potential tag to all outputs of the pooling layer.[9]

Collection of samples for dataset is a clumsy task as benchmark dataset requires bulky and diverse handwritten samples. Having a standard dataset provides a platform for researchers to evaluate and compare performance of different HTR techniques on same ground truth data thus eliminating any bias. Like any other scientific domain, document analysis and recognition community (DAR) has also developed large number of datasets for HTR. Despite deep research, no publically available dataset for Handwritten Character Recognition (HCR) was found. In most of the existing datasets writers were asked to copy the specific given text in their own unconstrained cursive handwriting, due to which these datasets provide segmented words, sentences or paragraphs rather than isolated characters. Secondly these datasets come at substantial cost and are not publically available. These setbacks compel most of the researchers to compile their own limited datasets for implementing and testing their system other state of art techniques. Due to which they are unable to compare performance of their system with other state of art techniques.[4]

CHAPTER 2: PROBLEM STATEMENT

To implement a handwritten character recognition system using Convolutional Neural Networks algorithm.

OBJECTIVE OF PROJECT

1. To collect datasets of various strokes that make up alphabets and numbers the of English language.

2. To use Neural networks for feature extraction and as a classifier.

3. To design a supervised deep neural network model for recognition.

4. To design a system that gives accurate output with less iteration time.

CHAPTER 3: LITERATURE SURVEY

[1] J. Pradeep Et. al propose diagonal feature extraction for offline character recognition using neural networks. The neural network recognition system is trained using the horizontal and vertical feature extraction methods.

[2] puts forth some of the basic techniques required to build a complete recognition model [1]. They discuss various examples of systems and softwares, research-based or commercial, major hurdles in designing such systems and their solutions.

[3] used LeNet-5, a Convolutional Neural Network (CNN) trained with gradient based learning and backpropagation algorithm is used for classification of Malayalam character images.

[4]Shardul Singh Chauhan in his review paper on Handwritten Character Recognition describes the basic working principle of character recognition followed by a detailed literature survey.

[5]This paper compared three most famous NN approaches are deep neural network (DNN), deep belief network (DBN) and convolutional neural network (CNN) and evaluated in terms of many factors such as accuracy and performance.

[6]concluded two main problems which they face while using CNN for handwriting recognition (i)The first problem in CNN is the manual training of data is very time consuming and it is labor intensive. (ii) The second problem is the design and parameter problem all depend on experiences.

[7]successfully implemented a Hindi handwritten Character recognition using Deep Convolution Neural Network. This system is an expansion of LeNet-5 architecture. Using 96000 character sets they obtaining a validation set accuracy of 95.72 percent using Adam optimizer and 93.68 percent using RMSprop optimizer.

[8] described in this paper how to use the mobile to collect data, process the data, and construct the data set. Furthermore they also compared the CNN models.

[9] proposed work highlighting on fine-tuning approach and analysis of state-of-the-art Deep Convolutional Neural Network (DCNN) designed for Devanagari Handwritten characters classification.

CHAPTER 3.1: Conclusion from literature survey:

The literature survey summarized for us the use of CNN techniques for applications including handwriting recognition and given below is a comparison of those techniques.

The following techniques are found related to HWR:

Year	Method	Accuracy	Details
2015	LeNet-5, Convolutional Neural Network (CNN)	92%	UsedLeNet-5, a Convolutional Neural Network (CNN) trained with gradient based learning and backpropagation algorithm is used for classification of Malayalam character images.
2019	RNN (Recurrent Neural Network)	87%	Used residual network to extract features from input images, then employ a RNN to model the contextual information within feature sequences and predict recognition
2019	Classification of Calligraphy Style Based on Convolutional Neural Network	82%	proposed CNN model skipped the complicated preprocessing, which effectively avoids the complex steps of features extraction and data reconstruction.
2019	Deep Convolutional Neural Network (DCNN)	95.72	Proposed deep convolution neural network for recognition

			of handwritten character in Hindi. is an expansion of LeNet-5 architecture. It was trained using 96000 character sets.
2020	‘H-WordNet’, DCNN	96.17	In this study, a deep convolutional neural networkbased holistic method termed ‘H-WordNet’ is proposed for handwritten word recognition. The H-WordNet model includes merely four convolutional layers and one fully connected layer to effectively classify the word images', which lead to a significant reduction in parameters.

CHAPTER 4: DATASET

For training and testing we have used the EMNIST dataset (Extended Modified NIST). The EMNIST dataset is a set of handwritten character digits derived from the NIST Special Database 19 and converted to a 28x28 pixel image format and dataset structure that directly matches the MNIST dataset. It is a subset of original NIST data, has a training set of 60,000 examples of handwritten characters.

The dataset is provided in two file formats. Both versions of the dataset contain identical information, and are provided entirely for the sake of convenience. The first dataset is provided in a Matlab format that is accessible through both Matlab and Python (using the scipy.io.loadmat function). The second version of the dataset is provided in the same binary format as the original MNIST dataset. The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset.

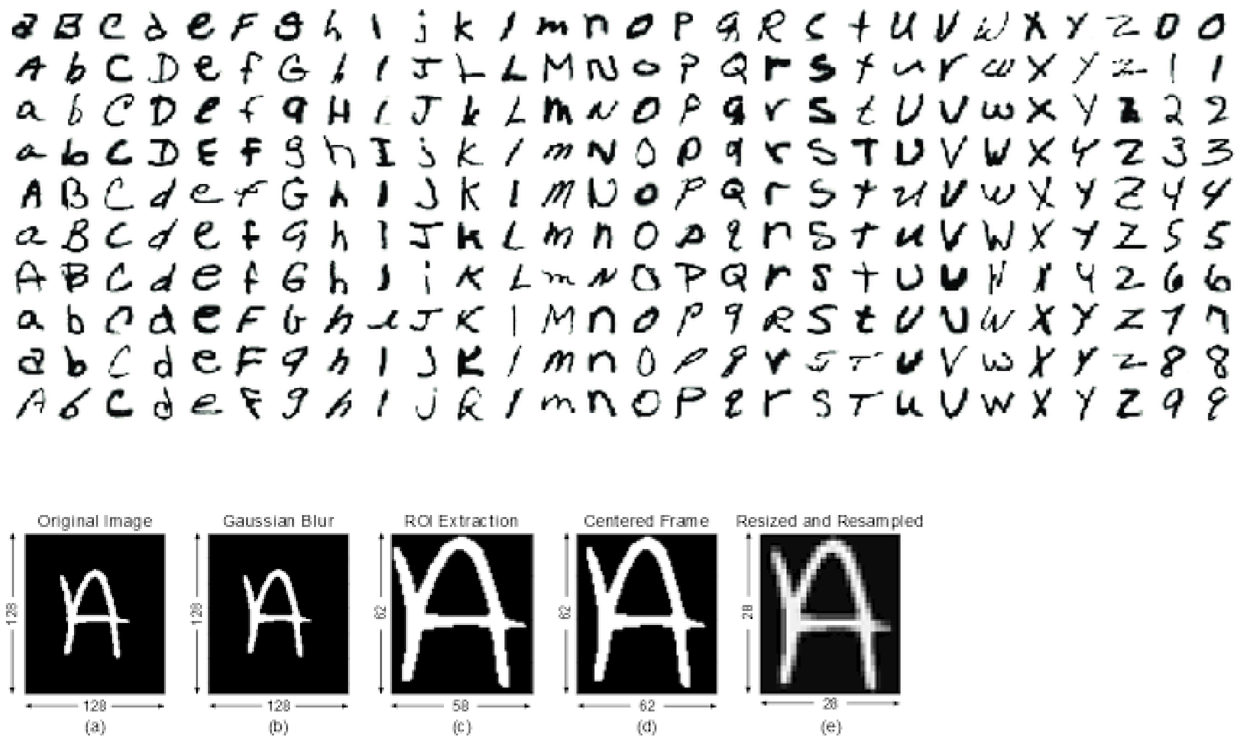


Fig. 4.1 **EMNIST dataset[6]**

CHAPTER 5: METHODOLOGY

The initial stage of the system adjusts the pixels and size of the input image by preprocessing the word images. Feature extraction methods select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.

In a convolutional network, we basically have three types of layers: convolution layer, pooling layer and fully connected layer. The classifier is a hypothesis or discrete-valued function that is used to assign (categorical) class labels to particular data points.

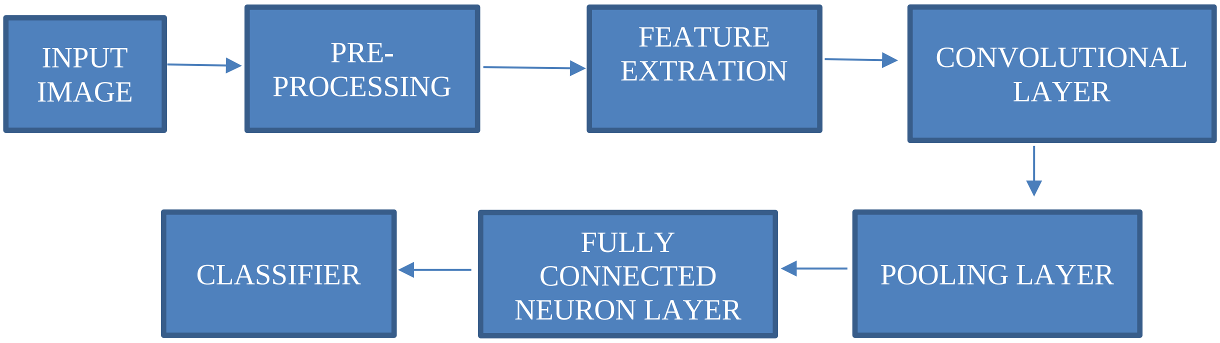


Fig. 5 Block diagram of the complete process of HWR

CHAPTER 5.1 ARCHITECTURE

LeNet was introduced in the research paper "Gradient-Based Learning Applied To Document Recognition" in the year 1998 by Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner.LeNet-5 CNN architecture is made up of 7 layers. The layer composition consists of 3 convolutional layers, 2 subsampling layers and 2 fully connected layers.

We will be using the modified LeNet-5 Architecture boosts the error rate upto 0.7% compared to LeNet -5 and others.

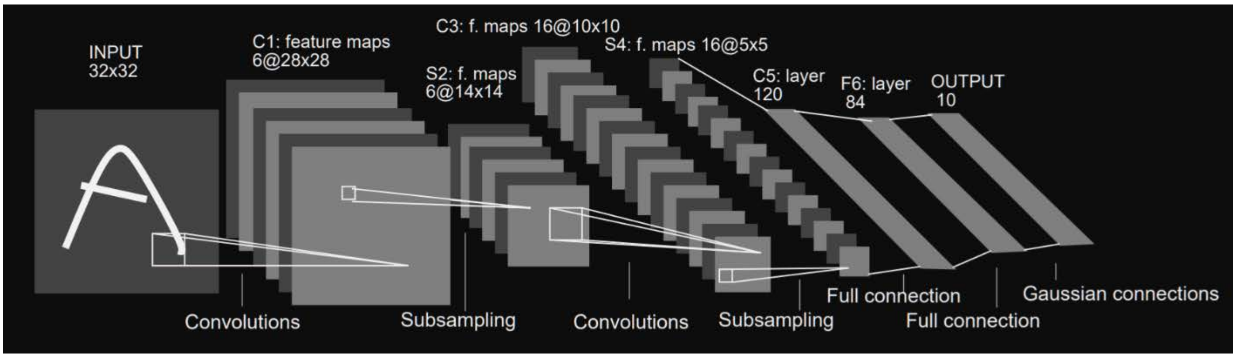


Fig. 5.1 LeNet-5 Architecture. img source: https://bit.ly/3aR

CNN is a class of deep neural networks which operates on learnable parameters and unique features of data . In a convolutional network, we basically have three types of layers:

- Convolution layer
- Pooling layer — Sub-sampling Layers
- Fully connected layer.

The official first layer convolutional layer *C1* produces as output 6 feature maps, and has a kernel size of *5x5*. The kernel/filter is the name given to the window that contains the weight values that are utilized during the convolution of the weight values with the input values. *5x5* is also indicative of the local receptive field size each unit or neuron within a convolutional layer. The dimensions of the six feature maps the first convolution layer produces are 28x28. To reduce the size of the input feature matrix, we usually use pooling layer. We will be using Max pool in this model. Max pooling takes the largest element from the rectified feature map.

A subsampling layer 'S2' follows the 'C1' layer'. The 'S2' layer halves the dimension of the feature maps it receives from the previous layer; this is known commonly as downsampling.
The 'S2' layer also produces 6 feature maps, each one corresponding to the feature maps passed as input from the previous layer. This link contains more information on subsampling layers. The fully connected layer takes the inputs from the feature analysis and applies weights to predict the correct label.

Activation Function:

A mathematical operation that transforms the result or signals of neurons into a normalized output. An activation function is a component of a neural network that introduces non-linearity within the network. Softmax is an activation function that is utilized to derive the probability distribution of a set of numbers within an input vector. The output of a softmax activation function is a vector in which its set of values represents the probability of an occurrence of a class/event.

More generally, CNNs work well with data that has a spatial relationship. Therefore CNNs are go-to method for any type of prediction problem involving image data as an input. The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image. This allows the model to learn position and scale in variant structures in the data, which is important when working with images.

Loss Function:

We cannot calculate the perfect weights for a neural network; there are too many unknowns. Instead, the problem of learning is cast as a search or optimization problem and an algorithm is used to navigate the space of possible sets of weights the model may use in order to make good or good enough predictions. In the context of an optimization algorithm, the function used to evaluate a candidate solution (i.e. a set of weights) is referred to as the objective function. With neural networks, we seek to minimize the error. As such, the objective function is often referred to as a cost function or a loss function and the value calculated by the loss function is referred to as simply "*loss*."

Categorical crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.

CHAPTER 6: System Specifications:

Hardware Requirements :

- Pre-Processer or GPU
- 128 MB RAM
- Up to 10 GB Hard Disk.

Software Requirements :

- Windows OS
- Language: Python
- Libraries: OpenCV (4.4.0)

  Keras (2.1.3)

  Tensorflow (2.3.1)*

  scikit-learn (0.23.2)

  Pandas (1.1.5)

  Numpy (1.19.3)

  Kaggle

  Dataset

CHAPTER 7: ALGORITHM

Algorithm: Convolutional Neural Networks (CNN)

Step1: Collect the relevant datasets for training and testing of the CNN model.

Step2: Open a python enabled ide like google collaboratory, jupyter notebooks and starting importing the packages-

packages used - tenserflow-keras,

- EMNIST dataset from MNIST

- sklearn

-openCV

-numpy

Step 3: CNN model with 8 layers

Model Details:

number of epochs - 10

image size = 28*28

The proposed CNN model has 8 layers i.e. -Conv2D(relu activated) x2

- maxpooling2D (relu activated)

- dropout x2

- flatten

- dense(softmax and relu activated) x2

Step 4: To implement the above layers in Tensorflow make use of functions like flatten, dropout,etc. which are in-built functions in keras used for feature extraction

CNN Layers used:

Flatten -> Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.

Dropout -> A Simple Way to Prevent Neural Networks from Overfitting .

Overfitting -> Overfitting indicates that your model is too complex for the problem that is solving, i.e. your model has too many features in the case of regression models and ensemble learning, filters in the case of Convolutional Neural Networks, and layers in the case of overall Deep Learning Models.

Conv2D -> Conv2D parameter is the numbers of filters that convolutional layers will learn from.

Maxpooling2D -> Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest, value in each patch of each feature map.

Dense -> A Dense layer feeds all outputs from the previous layer to all its neurons, each neuron providing one output to the next layer. It's the most basic layer in neural networks.

Loss function -> Categorical cross-entropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one.

- WHY SEQUENTIAL MODEL? - A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.

Training and testing dataset: EMNIST

The model uses train_test_split method 25% data for testing and 75% for training purpose.

CNN Used for better accuracy , sequential model.

CHAPTER 8: RESULTS AND DISCUSSION

Dataset plays a major role in any supervised learning model. More the date, more accurate will be the recognition. EMNIST dataset contains around 60,000 grayscale images of alphabets and numbers.

Many types of models have been created to work with handwriting recognition over the years. Shown below are some simulation results that we got using CNN model and Multilayer Perceptron model.
Below are the results we get using CNN model in Python.

CNN-
CNN is a class of deep neural networks which operates on learnable parameters and unique features of data . In a convolutional network, we basically have three types of layers: Convolution layer, Pooling layer and Fully connected layer.

Convolutional layer-

- Convolution is the first layer to extract features from an input image.
- It is a mathematical operation that takes two inputs such as image matrix and a filter or kernel.

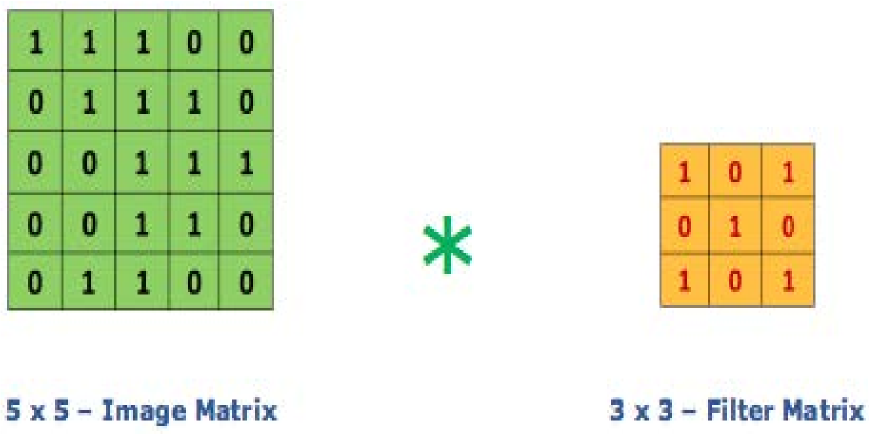


Fig 4.2 Convolutional Layer.[4]

Pooling layer:

- To reduce the size of the input feature matrix, we usually use pooling layer.
- We will be using Max pool in this model. Max pooling takes the largest element from

the rectified feature map.

Fully Connected Layer of Neurons:

- In this layer the matrix is flattened into vector and feed it into a fully connected layer like a neural network.
- The fully connected layer takes the inputs from the feature analysis and applies weights to predict the correct label.

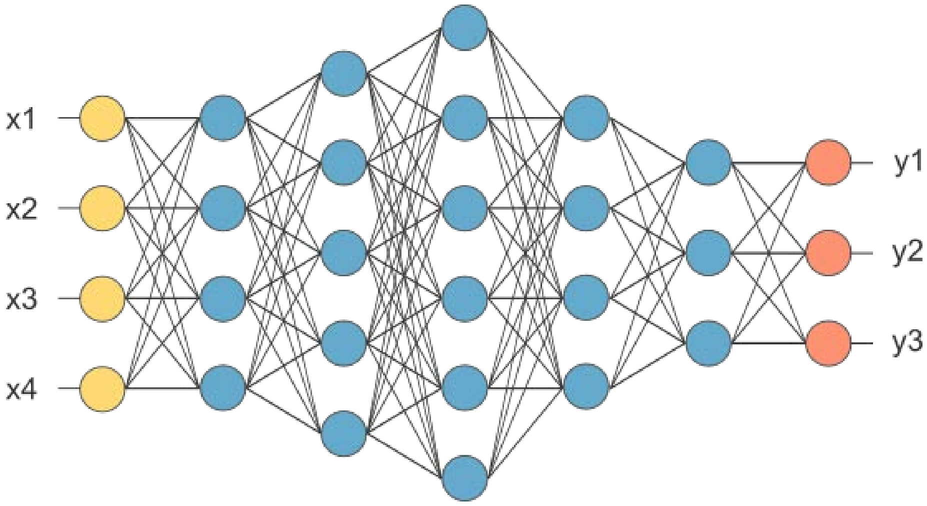
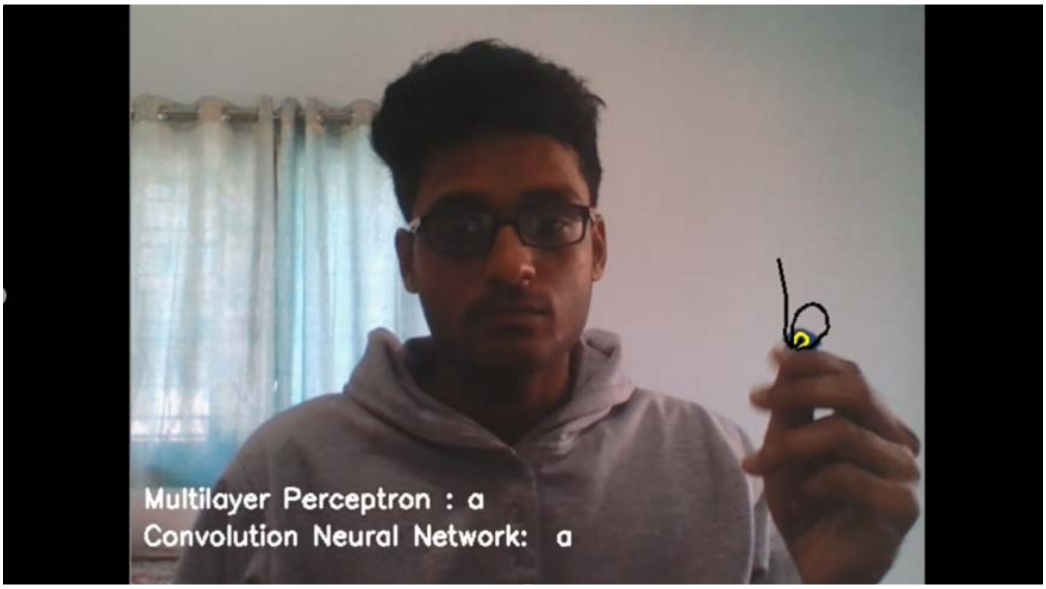


Fig 8.1 Fully-connected layer of neurons.
Img source: https://technoelearn.com/convolutional-neural-network-tutorial/

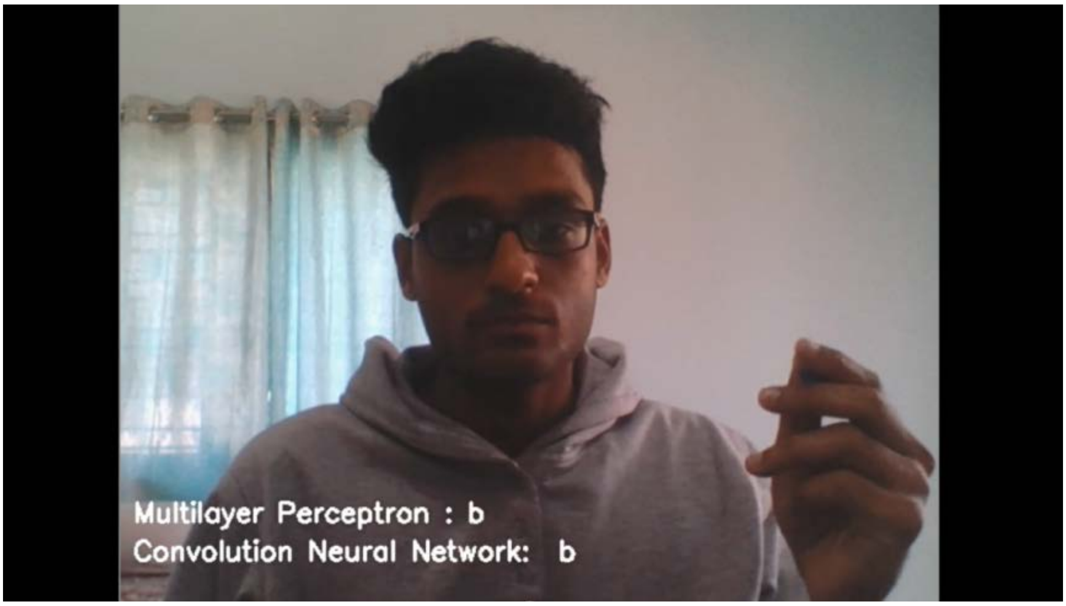
The panning of filters (you can set the stride and filter size) in CNN essentially allows parameter sharing, weight sharing so that the filter looks for a specific pattern, and is location invariant — can find the pattern anywhere in an image. This is very useful for object detection. Patterns can be discovered in more than one part of the image.

Additionally, it can also find the similar pattern even if the object is somewhat rotated/tilted using a concept called Pooling, which makes CNN more robust to changes in the position of the feature in the image.

Shown below are some results after training of the CNNmodel-



Drawing ‘b’ alphabet using a blue object



Recognized result of multilayer perceptron and CNN
Fig 8.2

Fig 8.3 Simulation results for alphabets ‘b’ and ‘c’

Fig 8.4 Simulation results for alphabet ‘e’

A multilayer perceptron is a class of feedforward artificial neural network. The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons.

On comparing the results we can conclude that both the techniques- Convolutional Neural Networks and Multilayer Perceptron were giving accurate results but with different accuracies.

- Results are put forth using loss curve & Model curve

Loss function :

The categorical crossentropy loss function calculates the loss of an example by computing the following sum:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\ \text{size}}} y_i \cdot \log \hat{y}_i$$

where hat{y}/*y*^*i* is the *i*-th scalar value in the model output, y_i*yi* is the corresponding target value, and output size is the number of scalar values in the model output.

This loss is a very good measure of how distinguishable two *discrete probability distributions* are from each other. In this context, y_i*yi* is the probability that event i*i* occurs and the sum of all y_i*yi* is 1, meaning that exactly one event may occur. The minus sign ensures that the loss gets smaller when the distributions get closer to each other.

The categorical crossentropy is well suited to classification tasks, since one example can be considered to belong to a specific category with probability 1, and to other categories with probability 0.

```
# loss curve
plotgraph(epochs, loss, val_loss)
```

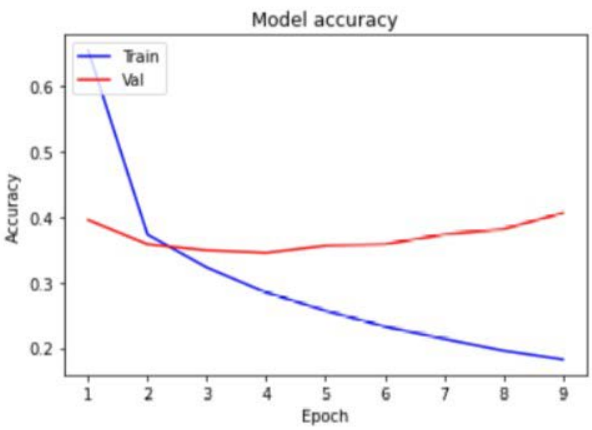


Fig 8.5 Loss curve

The loss curve shows what the model is trying to reduce. The training procedure tried to achieve the lowest loss possible with every epoch. The loss is calculated using the number of training examples that the models gets right, versus the ones it gets wrong. Or how close it gets to the right answer for regression problems. The test loss is higher than the training loss, meaning the model is slightly overfitting the training data, but that’s inevitable, it doesn’t seem problematic.

In [25]:
```
# Accuracy curve
plotgraph(epochs, acc, val_acc)
```

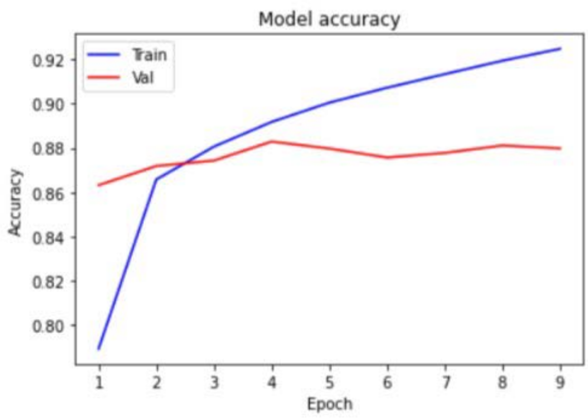


Fig. 8.6 Model accuracy

In this case Training Accuracy > Validation Accuracy. This happens when the training data is underfitting. Even though the data is under fitted, the validation data may perform well under circumstances that the validation data fits better in your model than does training data.

CHAPTER 9: COMPARATIVE ANALYSIS

Algorithm comparison

CNN vs MLP:

- MLPs (Multilayer Perceptron) use one perceptron for each input (e.g. pixel in an image) and the amount of weights rapidly becomes unmanageable for large images. It includes too many parameters because it is fully connected. Each node is connected to every other node in next and the previous layer, forming a very dense web.
  This results in redundancy and inefficiency.

- The benefit of using CNNs is their ability to develop an internal representation of a two-dimensional image. This allows the model to learn position and scale in variant structures in the data, which is important when working with images.

- Another common problem is that MLPs react differently to an input (images) and its shifted version — they are not translation invariant. For example, if a picture of a cat appears in the top left of the image in one picture and the bottom right of another picture, the MLP will try to correct itself and assume that a cat will always appear in this section of the image.

- The filters in CNN essentially allows parameter sharing, weight sharing so that the filter looks for a specific pattern, and is location invariant — can find the pattern anywhere in an image. This is very useful for object detection. Patterns can be discovered in more than one part of the image. Additionally, it can also find the similar pattern even if the object is somewhat rotated/tilted using a concept called Pooling, which makes CNN more robust to changes in the position of the feature in the image.

- Model accuracy and loss-

```
Test loss: 0.2338077425956726
Test accuracy: 0.930801272392273
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 24, 24, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 12, 12, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 9216)              0
_________________________________________________________________
dense_1 (Dense)              (None, 128)               1179776
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_2 (Dense)              (None, 26)                3354
=================================================================
Total params: 1,201,946
Trainable params: 1,201,946
Non-trainable params: 0
_________________________________________________________________
None
```

CNN

```
Test accuracy: 90.9519%
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 784)               0
_________________________________________________________________
dense (Dense)                (None, 512)               401920
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 26)                13330
=================================================================
Total params: 677,914
Trainable params: 677,914
Non-trainable params: 0
```

MLP

Fig 9.1 CNN vs MLP parameters

CHAPTER 10.1: MERITS

1. Less manual labour required and more cost efficient. It is cheaper than paying someone amount to manually enter great deal of text data. Moreover it takes less time to convert within the electronic form.

2. This process is much faster as compared to the manual typing the information into the system.

3. Information can be read with high degree of accuracy.

4. A paper based form are often became an electronic form which is straightforward to store or send by mail.

5. Easier data retrieval.

6. Electronic data storage**.** Once converted into digital form the documents can be stored into digital depository and can be accessed from anywhere. This also discards the need for physical storage units.

10.2: DEMERITS

1. Not 100% accurate, there are likely to be some mistakes made during the method.

2. Difficult to retrieve and mask accurate spacing of letters or words.

3. It may not support all the Different languages.

4. Somewhere we might lose the Unique style of writing.

5. Once converted into digital format the data is hackable.

10.3: APPLICATIONS

1. Orion Live Ink is a digitisation method which uses character recognition technology and question paper rubrics to publish examination results. It uses a digital pen to write marks on the top sheet. The pen digitizes marks in real time and transfers data to the table.

2. historic document processing and thus preserving the ancient text and culture. Some handwritings that are difficult to recognize the characters. A digital character conversion system identifies characters easily and converts them into a people-readable format.

3. Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statements, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation.

4. Sketch recognition is the automated recognition of hand-drawn diagrams by a computer.

5. one language can be converted into another language. Many people wrote stories, documents, novels, research works in their own languages. This meaning translation system can be used to convert these images into another language.

6. Keyword spotting system can extract any required word in the document image.
7. Signboard Translation is necessary to translate one language to another language. When people from other countries need to understand these display boards, they use this Signboard Translations system.

8. Analyzing human behavioral characteristics through text processing and handwriting recognition.

9. Signature verification.

10.4 FUTURE SCOPE

This work further extended to the character recognition for other languages. For e.g. -In future, deploying further compact deep convolution network to classify handwritten Devanagari character is planned.

Designing a text recognition system for mobile device deployment which can used by anywhere.

Future work will include offline handwritten character into consideration for joint learning with increased accuracy and less computational power.

CHAPTER 11: CONCLUSION

In this project, we used a convolutional neural network to handwritten character recognition. We trained the model using the EMNIST data set. Using this dataset we trained two most used handwriting recognition models- Convolutional Neural Networks and Multilayer Perceptron. On comparing the results based on different parameters we can conclude that-

Convolutional neural networks (CNNs) are very effective in perceiving the structure of handwritten characters/words in ways that help in automatic extraction of distinct features.

On implementing we found that Real time inputs through webcam are recognized with 91% accuracy using LeNet-5 architecture based CNN model.

CNN model works better at feature extraction and classification than MLP(multilayer perceptron) for handwriting recognition.

The performance of the model is mostly dependent on the dataset used while training of the CNN model and number of Convolution layers used.

References:

[1] J. Pradeep, E. Srinivasan, S. Himavathi “Diagonal based feature extraction for handwritten character recognition system using neural network” *in 3rd International Conference on Electronics Computer Technology*. 07 July 2011.

[2] Jin Hyung Kim, Bong-Kee Sin, Dr. David Doermann, Dr. Karl Tombre “Handbook of Document Image Processing and Recognition”in *Chapter 26 - Handbook of Document Image Processing and Recognition*. Published 12 May 2014.

[3] Anil R., Manjusha K., Kumar S.S., Soman K.P. (2015) “Convolutional Neural Networks for the Recognition of Malayalam Characters”. In: Satapathy S., Biswal B., Udgata S., Mandal J. (eds) *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014. Advances in Intelligent Systems and Computing, vol 328. Springer, Cham.*

[4] Purohit, Ayush & Chauhan, Shardul. (2016). *A Literature Survey on Handwritten Character Recognition. International Journal of Computer Science and Information Technology*. 7. 1-5.

[5] M. M. Abu Ghosh and A. Y. Maghari, "A Comparative Study on Handwriting Digit Recognition Using Neural Networks," *2017 International Conference on Promising Electronic Technologies (ICPET)*, 2017, pp. 77-81, doi: 10.1109/ICPET.2017.20

[6] Cohen, Gregory & Afshar, Saeed & Tapson, Jonathan & van Schaik, André. (2017). *EMNIST: an extension of MNIST to handwritten letters.*

[7]Jia xiaodong, gong wednog, yuan jie, “Handwritten Yi Character Recognition with Density Based Clustering Algorithm and Convolutional Neural Network”, *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference Embedded and Ubiquitous Computing (EUC*).

[8] Karishma Verma, Dr.Manjeet Singh, “Hindi handwritten Character recognition using Convolutional neural network”, *International journal of computer sciences and Engineering*, vol.6, Issue.6, pp.909-914, 2018.

[9] Weng, Y., Xia, C. “A New Deep Learning-Based Handwritten Character Recognition System on Mobile Computing Devices.” *Mobile Netw Appl* 25**,** 402–411 (2020).

[10] Deore, S.P., Pravin, “A. Devanagari Handwritten Character Recognition using fine-tuned Deep Convolutional Neural Network on trivial dataset” *Sādhanā* **45,** 243 (2020).

[11] Dibyasundar Das, Deepak Ranjan Nayak, Ratnakar Dash, Banshidhar Majhi,Yu-Dong Zhang “H-WordNet: a holistic convolutional neural network approach for handwritten word recognition*”. IET Image Processing* (Volume: 14 , Issue: 9 , 7 20 2020)

[12] Deepak Chaudhary, Kaushal Sharma “Hindi Handwritten Character Recognition using Deep Convolution Neural Network”. *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)* .13 February 2020.

[13] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. Retrieved from http://arxiv.org/abs/1702.05373.

[14] Shinde A., Bhave A., Haribhakta Y.V. (2021) Handwriting Recognition on Filled-in Forms Using CNN. In: Singh S.K., Roy P., Raman B., Nagabhushan P. (eds) Computer Vision and Image Processing. CVIP 2020. Communications in Computer and Information Science, vol 1376. Springer, Singapore.

[15] Geetha, R., Thilagam, T. & Padmavathy, T. Effective offline handwritten text recognition model based on a sequence-to-sequence approach with CNN–RNN networks. *Neural Comput & Applic* (2021)

[16] Hamida S., Cherradi B., Ouajji H., Raihani A. (2020) Convolutional Neural Network Architecture for Offline Handwritten Characters Recognition. In: Serrhini M., Silva C., Aljahdali S. (eds) Innovation in Information Systems and Technologies to Support Learning Research. EMENA-ISTL 2019. Learning and Analytics in Intelligent Systems, vol 7. Springer, Cham. https://doi.org/10.1007/978-3-030-36778-7_41