Handwritten Character Recognition

Enrol. No. (s) – 20103012, 20103016, 20103020

Name of Students – Pratishtha Bhateja , Muskan Mittal , Manika Agarwal

Name of Supervisor – Ms. Kirti Aggarwal

December-2022

Submitted in partial fulfilment of the Degree of

Bachelor of Technology

In

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)

TABLE OF CONTENTS

(II)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: …… Signature:

Date: 06-12-2022 Name: Pratishtha Bhateja

Enrollment No: 20103012

Place: …… Signature:

Date: 06-12-2022 Name: Muskan Mittal

Enrollment No: 20103016

Place: …… Signature:

Date: 06-12-2022 Name: Manika Agarwal

Enrollment No: 20103020

(III)

CERTIFICATE

This is to certify that the work titled " **HANDWRITTEN CHARACTER RECOGNITION**" submitted by " **PRATISHTHA BHATEJA** ", "**MUSKAN MITTAL** " and "**MANIKA AGARWAL** " in partial fulfilment for the award of degree of Bachelor of Technology of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma

Signature of Supervisor : ……

Name of Supervisor : Ms. Kirti Aggarwal

Designation : Assistant Professor - II

Date : 06-12-2022

(IV)

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our project supervisor **Ms. Kirti Aggarwal** as well as Jaypee Institute of Information Technology, Noida who gave us the golden opportunity to do this wonderful project on the topic **"Handwritten Character Recognition"**, which also helped us in doing a lot of research and we came to know about so many new things which we are really thankful for.

Secondly, we would also like to thank our friends who helped us a lot in the collection of the text descriptions required for this project within the limited time frame.

Signature of Student : ……

Name of Student : Pratishtha Bhateja

Enrollment Number : 20103012

Date : 06-12-2022

Signature of Student : ……

Name of Student : Muskan Mittal

Enrollment Number : 20103016

Date : 06-12-2022

Signature of Student : ……

Name of Student : Manika agarwal

Enrollment Number : 20103020

Date : 06-12-2022

(V)

SUMMARY

The goal of this project is to recognize the handwritten – characters using deep learning. This project has a research-based nature, and majority of the work is in the form of implementation of CNN (Convolutional Neural Network) model with the help of tensorflow and keras which will recognise Handwritten characters.We have created two different models one for each dataset of digits and devnagri characters. The accuracy of the model is 98% with 5 epochs and 96%on 4 epochs respectively. The accuracy can be further improved by using even more advanced hardware and software, together with a very large dataset and this model can be utilised by any person who use laptops and computers . The model can be refined by adding more layers and more data so that better results could be obtained. Some advanced pre trained models may also be tried for the same.

_____ _____

Signature of Student Signature of Supervisor

Name : Pratishtha Bhateja Name : Ms. Kirti Aggarwal

Date : 06-12-2022 Date : 06-12-2022

_____

Signature of Student

Name : Muskan Mittal

Date : 06-12-2022

_____

Signature of Student

Name : Manika Agarwal

Date : 06-12-2022

(VI)

LIST OF FIGURES

1. INTRODUCTION

1.1 General Instruction

The handwritten Character recognition is the ability of computers to recognize human handwritten characters. This is Deep learning project, or we say Machine learning project in which we will create a Convolutional neural network(CNN) model with the help of tensorflow and keras which will recognise Handwritten characters. We can use any dataset from our choice i.e. Digits, Devanagari, Chinese symbols e.t.c. For image recognition and processing, there is a very popular artificial neural network used that is Convolutional neural network (CNN) that is specifically designed to process pixel data.

1.2 Problem Statement

We are developing a deep learning model to recognize human handwritten characters as the handwriting varies from person to person and sometimes it is very difficult to type those letters which are not present in our qwerty keyboard. So by using a CNN approach the characters can be recognized and can be easily understood and can be converted into a formattable text for further uses.

1.3 Significance/Novelty of the problem

India is a multi-lingual multi-script country and there are twenty-two languages. Eleven scripts are used to write these languages and Devnagari is the most popular script in India. It contains 12 vowels and 36 consonants. It is very difficult to type Hindi language using a popularly used qwerty keyboard. Today also, in many government offices most of the work is done in Hindi language only and due to the absence of proper Hindi characters in our computers and laptops many people complete their work themselves by writing each character which sometimes become very hectic, tiring and quite frustrating for these people also. It cannot be much better if we can type those letters easily on our desktop by using this character recognition technique. In this way we can solve their problem and this will also decrease their workload .

1.4 Empirical Study (Field Survey/Existing Tool Survey/Experimental Study)

1.4.1 Tool Survey

1.4.1.1 IPython Notebook - Jupyter

Interactive Python (or IPython) is a tool or basically a command prompt/shell for interactive computing in Python programming language, but now it has extended support for different other languages like R. Although it is based on browser, it offers a notebook-like interface with support for code, math expressions, plots that are inline, text and other media. The notebooks offers rich graphs/diagrams, shell syntax and tab completion. The shells are interective, cell-based where each cell can be executed independently. It also offers support for data visualization

1.4.1.2 Google Colab

Colaboratory is a free Jupyter notebook environment provided by Google that runs entirely in the cloud. It requires no setup and is a platform for building machine learning models with GPU support. Google Colab enables one to write and execute code and access computing resources for free from the browser. It comes with libraries which are used for accessing various services provided by Google conveniently. It saves files to Google Drive and allows easy sharing of Jupyter notebooks Google Colaboratory is hosted by Google designed for data experimentation and analysis But on the downside, there are some limitations, like for example the entire environment (i.e. libraries, packages, and data files) do not persist across sessions Programs cannot be executed for longer than a certain amount of time.

1.4.2 Libraries Survey

1.4.2.1 TensorFlow

TensorFlow is an open source programming library for programming of dataflow over a scope of work. It is a general arithmetic library which is likewise utilized for AI applications, for example neural systems . It is utilized at Google for both research and creation .

Its design is flexible to the point that it takes into account the simple arrangement of calculation over a scope of stages , for example – CPU's, GPU's, TPU's and from work areas to portable and edge gadgets to bunch of servers. TensorFlow calculations are spoken to as stateful dataflow charts. The name TensorFlow gets from the activities that such sort of neural systems perform on multi-dimensional exhibits and these clusters are alluded to as "Tensors".

1.4.2.2 Keras

Keras is an open source library for implementing neural network written in Python. It is capable of running on top of TensorFlow , Theano and Microsoft Cognitive tool . It focuses on being user – friendly , modular, and extensible and is designed to enable fast experimentation with depp neural networks.

A Keras model is realized as a sequence or a stand – alone , fully- configurable module's graph that can be plugged together with a little as restrictions as possible. New models can be created by combining various modules like layers,loss functions, e.t.c. Keras is suitable for advanced research in neural networks as new modules are simple to add just as new classes and functions.

1.4.2.3 Numpy

NumPy is a Python library used for working with arrays. Array programming provides a powerful, compact and expressive syntax for accessing, manipulating and operating on data in vectors, matrices and higher-dimensional arrays. NumPy is the primary array programming library for the Python language. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python. The NumPy array is a data structure that efficiently stores and accesses multidimensional arrays17 (also known as tensors), and enables a wide variety of scientific computation. It consists of a pointer to memory, along with metadata used to interpret the data stored there, notably 'data type', 'shape' and 'strides'.

1.4.2.4 Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications. Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

1.4.2.5 Pandas

Pandas is defined as an open-source library that provides high-performance data manipulation in Python. It is used for data analysis in Python. Data analysis requires lots of processing, such as restructuring, cleaning or merging, etc. There are different tools are available for fast data processing, such as Numpy, Scipy, Cython, and Panda. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools. Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., load, manipulate, prepare, model, and analyze.

1.4.2.6 Joblib

Joblib is a library in python which is used to save your model for future use to make a prediction on unseen data. It also helps to compare the models with other models. The saving of data is called Serialization while restoring the data is called Deserialization. Also, we deal with different types and sizes of data. When we need the same trained data in some different projects or later sometime, to avoid. it also helps to transfer your model to someone who wants to make predictions.

1.4.2.7 CV2

CV2 is nothing but the latest version of OpenCV. OpenCV is a library of programming functions mainly aimed at real-time computer vision. In simple language it is a library used for Image Processing. It is mainly used to do all the operation related to images. From OpenCV you can do can whatever you want to do from images. OpenCV helps in manipulation of images. Like read an image, write an image, convert coloured to Gray, binary, HSV etc.

1.4.2.8 Sklearn

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib. Scikit-learn is mainly coded in Python and heavily utilizes the NumPy library for highly efficient array and linear algebra computations. Some fundamental algorithms are also

built in Cython to enhance the efficiency of this library. Support vector machines, logistic regression, and linear SVMs are performed using wrappers coded in Cython for LIBSVM and LIBLINEAR, respectively. Expanding these routines with Python might not be viable in such circumstances.

1.5 Brief description of the solution approach

Neural networks reflect the behaviour of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning. We will use a neural network-based approach to train the model on available training data, and then test it. Neural networks rely on training data to learn and improve their accuracy over time. However, once these learning algorithms are fine-tuned for accuracy, thus allowing us to classify and cluster data at a high velocity. One of the most well-known neural networks is Google's search algorithm.

1.6 Comparison of existing approaches to the problem framed

1.6.1 Neural Networks

The key insight to understand deep learning is to understand neural networks. Artificial Neural Networks (ANN), have been made to contribute in resolving problems of different realms in the recent several decades. Generally speaking, ANN can be depicted as a numerical model of a certain structure, comprising of some of the single processing components, constructed between inter-connected layers. Every unique layer comprises hidden neurons that are used for transforming the input data and figuring out the outputs to the next associated neurons. Deep learning methods have successfully demonstrated their advantages in many applications, where the input values are characterized by high dimensionality, huge quantities and highly-structured. Consequently, deep learning methods are widely applied in image classification and have good performance. Because of the framework of the image that comprises millions of pixels that can be clearly aligned into well-defined objects, deep learning tools are quite useful in the field of image classification.

1.6.2 CNN

One of the most effective models for deep learning is the Convolutional Neural Network (CNN). The CNN comprises two unique types of layers, pooling layers and convolution. Layers of CNN contain well-designed filters to handle with input data. They convolve the range of input values, and finally get smaller range of them. And then, CNN can detect essential or specific features within the

range we acquired before. The CNN generally consists of the input layer, convolution layer and Rectified Linear Unit. Rectified Linear Unit (ReLU) is mathematically expressed as Max (0, x), Convolution layer can produce a matrix of smaller dimension than input matrix, and max pooling can transmit the maximum value from amongst a rather small batch of data of the input matrix to the output. The output layer is fully connected, and based on the activation function.

HOW IT WORKS ?

Once an input layer is determined, weights are assigned. These weights help determine the importance of any given variable, with larger ones contributing more significantly to the output compared to other inputs. All inputs are then multiplied by their respective weights and then summed. Afterward, the output is passed through an activation function, which determines the output. If that output exceeds a given threshold, it “fires” (or activates) the node, passing data to the next layer in the network. This results in the output of one node becoming in the input of the next node. This process of passing data from one layer to the next layer defines this neural network as a feedforward network.

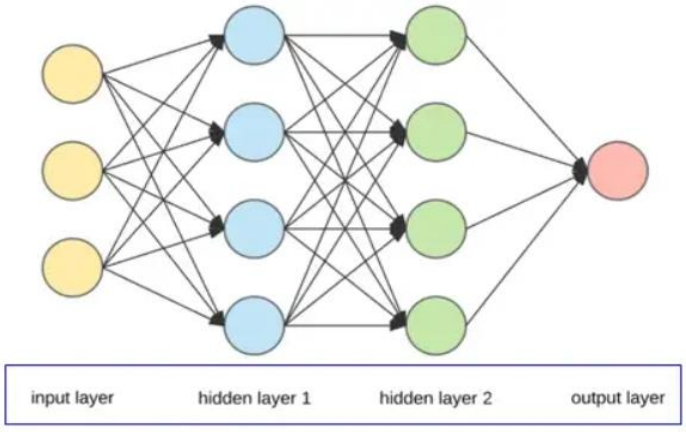


Fig. 1 Basic Structure of Neural Network

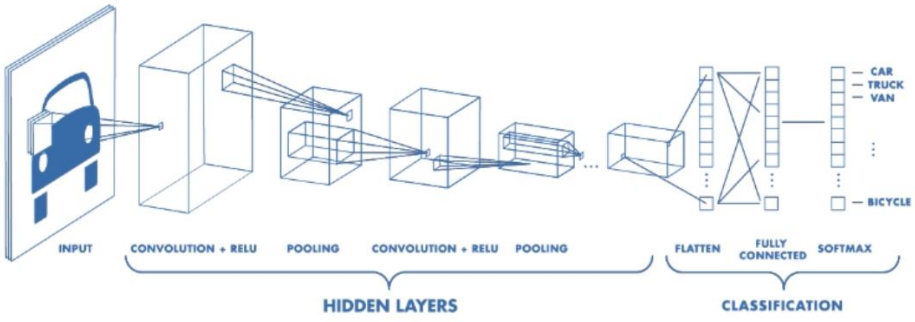


Fig. 2 Construction Model of CNN

2. LITERATURE SURVEY

2.1 PAPER 1: Diagonal based feature extraction for handwritten character recognition system using neural network

.A new method, called, diagonal based feature extraction is introduced for extracting the features of the handwritten alphabets. Steps involved are –

1) *Image – Acquisition* - Acquires a scanned input image of a specific format such as JPEG, BMT etc.

1)*Pre- processing* - It includes the steps that are required to shape the input image into a form suitable for segmentation such as Binarization , Global thresholding , Detection of edges.

2)*Segmentation* - In the segmentation, the input image is segmented into individual characters and then, each character is resized into m × n pixels towards training the network.

3)*Feature- extractions.* In the feature extraction process, resized individual character of size suppose- 90×60 pixels is further divided into 54 equal zones, each of size 10×10 pixels. The features are extracted from the pixels of each zone by moving along their diagonals. This procedure is repeated for all the zones leading to extraction of 54 features for each character. These extracted features are used to train a feed forward back propagation neural network employed for performing classification and recognition tasks. Fifty data sets, each containing 26 alphabets written by various people, are used for training the neural network and twenty different handwritten alphabets characters are used for testing._The scanned image is taken as dataset/input and feed forward architecture is used. The structure of neural network includes an input layer with 54/69 inputs, two hidden layers each with 100 neurons and an output layer with 26 neurons.The neural networks have emerged as the fast and reliable tools for classification towards achieving high recognition accuracy. Extensive simulation studies show that the recognition system using diagonal features provides good recognition accuracy of 96.52% with 54 features and 97.84% with 69 features. It is suitable for several applications including handwritten name recognition and conversion of any handwritten document into structural text form.

2.2 PAPER 2: Comparative Study of Devnagari Handwritten Character Recognition Using Different Feature and Classifiers

Twelve different classifiers like projection distance, subspace method, linear discriminant function, support vector machines, modified quadratic discriminant function, mirror image learning, Euclidean distance, nearest neighbour, k-Nearest neighbour, modified projection distance, compound projection distance and compound modified quadratic discriminant function are used for comparative study.

1)*Computation of gradient feature* - To get gradient feature, at first, a 2×2 mean filtering is applied 4 times on the input image and a non-linear size normalization is done. The 148×148 pixels image is then used.To obtain 49×49 segmented blocks from a pattern of 148×148 pixels we used $k=(49/148)*(i-1)+1$ and $1=(49/148)*(j-1)+1$, where (i, j) is the coordinates of 148×148 patterns and (k, 1) is the coordinates of 49×49 blocks

2)*Computation of curvature feature* - The curvature c is computed by bi-quadratic interpolation method and quantized into 3 levels using a threshold t (for concave, linear and convex regions). For concave region $c \leq -t$, for linear region $-t<c<t$ and for convex region $c \geq t$. We assume t as 0.15 in our experiment. The alphabet of the modern Devnagari script consists of 14 vowels and 33 consonants. These characters are called basic characters. and we used these 47 basic characters for our experiment. Writing mode in Devnagari script is from left to right. The concept of upper/lower case is absent in Devnagari script.The following observations and results are obtained.

Classifier	Grey image		Binary image		
	Gradient (392 dim.)	Curvature (392 dim.)	Gradient (392 dim.)	Curvature (392 dim.)	Average
PD	92.76	93.76	92.77	93.53	93.21
SM	92.61	93.62	92.61	93.35	93.04
LD	86.76	89.02	86.78	88.88	87.86
SVM	93.38	94.52	93.59	94.36	93.96
MQDF	94.24	94.78	94.14	94.50	94.42
MIL	94.74	95.19	94.74	95.09	94.94
ED	77.94	80.06	77.89	80.08	78.99
NN	87.16	86.72	87.69	87.22	87.19
k-NN	89.85	89.78	90.06	89.96	89.91
MPD	93.92	94.44	93.83	94.44	94.15
CPD	94.45	94.97	94.33	94.75	94.62
CMQDF	94.39	94.92	94.32	94.62	94.56

2.3 PAPER 3: Beyond human recognition: A CNN-based framework for handwritten character recognition.

The deep learning model can be seen as “black box” and we only need to input the image and then the recognition result is obtained. In contrast, the traditional methods usually require artificial feature design and manually tuning of the classifier. Among the deep learning models, the convolutional neural networks (CNN) are the most popular one especially for image recognition. This is because CNN is very suitable to represent the image structure. First, the pixels of the image are strongly related to their neighbour pixels but has little correlation with the far away pixels. Second, in CNN, the weight sharing strategy ensures that different part of the image can share similar properties, such as texture and brightness. For training MNIST, an 8-Layer network is used. (including pooling layers). The “In”, “Cony” “MaxP”, “Full” and “Out” indicates the input layer, convolution layer, max pooling layer, fully-connected layer and softmax output layer, respectively and For training CASIA, we used a 15-Layer network since Chinese characters have much more complex structure than digits. MNIST(handwrit-ten written digits) and CASIA(handwritten Chinese characters). MNIST contains 10 classes of handwritten digits. There are 50,000 training samples and 10,000 test samples. For CASIA, we used a subset of 3,755 commonly used Chinese characters. For MNIST, the average error rate of the 5 models was 0.24% while the voting was only 0.18%. For CASIA, the average error rate of the models was 3.42% while the voting was only 3.21 %.

2.4 PAPER 4: Handwritten Character Recognition using Neural Network

Objective of this paper is to recognize the characters in a given scanned document and study the effects of changing models of ANN. OCR is a widespread use of Neural Networks. Parameters like number of hidden layers, size of hidden layers and epochs are considered. Multilayer Feed Forward with back propagation .The parameters listed in these papers are skewing, slanting, thickening, cursive handwriting, joint characters. If all these parameters are taken care in the preprocessing

phase then overall accuracy of the Neural Network would increase.Preprocessing has been done for separating the Characters and normalizing each character. The image would be in RGB format so it is converted into binary format. To do this, input image is converted to grayscale format, and then threshold is used to convert image to binary. Skew correction has been done by Line fitting i.e. Linear Regression. Tool used is MATLAB

•A small number of nodes in the hidden layer lower the accuracy.

•A large number of neurons in the hidden layer help in increasing the accuracy

•As number of hidden layer increases the accuracy increases initially and then saturates at certain rate

•Mostly Accuracy is increased by increasing the number of cycles.

•Accuracy could be increased by increasing the training set.

2.4 PAPER 5: Hindi Handwritten Character Recognition using Deep Convolutional Neural Network

Convolution Neural Network (CNN) is turning out to be a very powerful tool for solving Machine Learning (ML) problems, especially in multiclass image classification. With the availability of a huge handwritten dataset, it is possible to achieve a never thought machine accuracy in image classification. In this paper, a Deep Convolutional Neural Network (DCNN) for Hindi handwritten character recognition is proposed. Convolutional Layer reduces the trainable weight parameters and also allows us the parameter sharing in one feature map. The filter slides over the input volume. In one slide it takes the dot product of all the overlapping pixels and projects this scalar result into output volume. In every layer of Convolution, ReLU (Rectified Linear Unit) non-linearity activation function . To reduce the size of the input feature matrix, we usually use a pooling layer. It speeds up the computation. Devanagari Character Set, Nov 2018, [online] Available: https://www.kaggle. com/rishianand/devanagari-character-set. The model is trained using 96,000 characters, The data is split into a 70:20:10 ratio. 70% for the training set, 20% for cross-validation and 10% for the test set.The model is trained using two optimizers Adam and RMSprop which respectively give 95.72% and 93.68% accuracy in the validation set. The training is done using 25 epochs. The accuracy seems to increase drastically in the initial epochs and gets saturated in the higher iterations.

2.4 PAPER 6: Deep learning based large scale handwritten Devanagari character recognition

In this paper a new public dataset for Devanagari script :Devanagari Handwritten Character Dataset (DHCD) is introduced. Deep Convolutional Neural Network(CNN) have shown superior results. Dropout and dataset increment approach has been used to increase accuracy.The ability of CNN to correctly model the input dataset can be varied by changing the number of hidden layers. The control flow includes: Character Extraction(scanning,cropping,labelling) , Pre-processing (resizing,

conversion to grayscale, suppressing background pixels, padding) , separating into train and test data, run CNN trainer, run test on testing data , check for accuracy. The dataset consists of 92 thousand images of 46 different classes of characters of Devanagari script segmented from handwritten documents. The Dataset is randomly split into Training (85 %) and Testing set(15 %). Training set consists of 78, 200 images and the Testing set consists of the remaining 13, 800 images. Each image is 32×32 pixels and the actual character is centered within 28×28 pixels. The proposed architecture scored the highest test accuracy of 98.47% on the dataset. 50 epochs have been used. Increasing the number of training samples is effective to increase performance of a wide and deep network with a large bank of parameters. Extending the dataset also resulted in slight improvement in Test accuracy.

2.7 PAPER 7: Handwritten Digit Recognition Using Machine Learning Algorithms

Recognising handwritten text is one of the practical applications of symbol recognition. In this paper the text used is a digit which is handwritten and it is to be recognized. Handwritten digits recognition will help in form data entry, check processing in banks etc. The problem is to develop an off-line application that can recognize hand written digits by using different machine learning techniques. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten digits. Data used is a digit dataset provided by Austrian Research Institute for Artificial Intelligence, Austria. This data set indicate that arbitrary scaling and a blur setting of 2.5 for the Mitchell down-sampling filter should perform well and used down-sample to 16x16 pixels**.** Waikato Environment for Knowledge Analysis (WEKA) is a prominent suite of machine learning. It contains a collection of algorithms and visualization tools for predictive modelling, data analysis, along with graphical user interfaces for smooth access to this functionality. Several machines learning algorithm namely, Multilayer Perceptron, Support Vector Machine, NaFDA5; Bayes, Bayes Net, Random Forest, J48 and Random Tree has been used for the recognition of digits using WEKA. The result of this paper shows that highest accuracy has been obtained for Multilayer Perceptron. Hence a model was implemented that was using Multilayer Perceptron algorithm for digits recognition and an accuracy of 90.37% was obtained.

2.8 PAPER 8: Handwritten character recognition using convolutional neural network

Handwritten character recognition is a prominent application for detection of characters from images, documents and other sources. Convulational Neural Network (CNN) is a deep learning

architecture, that trains a model to learn a input and output of a dataset with each iteration. The paper discusses about CNN capability to recognize the characters from the image dataset and the accuracy of recognition with training and testing. It is a very well-known deep learning architecture motivated by the natural visual perception technique of human brain. The dataset used here is NIST. The CNN algorithm is implemented in MATLABR20015a. The accuracy obtained from 200 training images as 65.32% see Fig.4 is improved gradually with increasing training images. The accuracy reaches to 92.91% with the 1000 training images. Thus, further increment of training images will continue to enhance the accuracy towards to certain limit – which cannot be exceeded due to numerical errors, and the constraints on the CNN capability of image differentiability for labels

Test Results for Handwriting Character Recognition with NIST Dataset.

No. of Training Images	No. of Testing Images	Average Accuracy (%)
200	200	65.32%
300	200	74.43%
500	200	80.84%
600	200	85.21%
800	200	87.65%
1000	200	92.91%

2.9 PAPER 9: Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)

Pre-processing techniques are the first step in a character recognition system. This paper deals with the various pre-processing techniques involved in character recognition system with different kind of images ranges from simple handwritten form-based documents and documents containing coloured and complex background and varied intensities.

Various pre-processing techniques were used in this paper like:

- Image enhancement techniques: it is to modify attributes of image to make it more suitable for a given task and improve the quality of image for the human perception

- Binarization (Thresholding): Pre-processing techniques are applied only on grey or binary images; a grey image is one in which pixel density value is in between 0 and 255 and a binary image is one in which pixel density value is in the form of 0 and 1 where 0 stands for white i.e., background of image and 1 stand for black i.e., foreground of image.
- Skew detection and correction
- Morphological processing
- Character segmentation

All image enhancement techniques have to be applied for accuracy in pre-processing system. Pre-processed image can be used in feature extraction phase and then in neural network phase. Even applying all these techniques, we cannot obtain the 100% accuracy in a pre-processing system.

2.10 PAPER 10: The MNIST Database of Handwritten Digit Images for Machine Learning Research.

This paper presents the modified National Institute of Standards and Technology (MNIST) resources, consisting of a collection of handwritten digit images used extensively in optical character recognition and machine learning research. Handwritten digit recognition is an important problem in optical character recognition, and it has been used as a test case for theories of pattern recognition and machine learning algorithms for many years. The freely available MNIST database of handwritten digits has become a standard for fast-testing machine learning algorithms for this purpose. General site for the MNIST database:http: //yann.lecun.com/exdb/mnist
Code to read the MNIST database:
http: //www.mathworks.com/matlabcentral/fileexchange/27675-read-digitsand-labels-from-mnist database
There are 60,000 training images (some of these training images can also be used for cross validation purposes) and 10,000 test images, both drawn from the same distribution. All these black and white digits are size normalized, and centred in a fixed size image where the center of gravity of the intensity lies at the centre of the image with 28 # 28 pixels. Thus, the dimensionality of each image sample vector is 28 * 28 = 784, where each element is binary. The MNIST database gives a relatively simple static classification task for researchers and students to explore machine learning and pattern recognition techniques, saving unnecessary efforts on data pre-processing and formatting. MNIST has been used for image and more general classification tasks.

2.11 Integrated Summary of the Lecture studied

Data exploration:. Devnagari is the most popular script in India. It contains 12 vowels and 36 consonants. It is difficult to type Hindi language using a popularly used qwerty keyboard. There are only few research reports available on devnagri[2].So it becomes a good choice to work on. The MNIST is a widely used dataset for handwritten digits as it saves unnecessary efforts on data pre-processing and formatting.[10]

Splitting data for training and testing : After the data has been pre-processed, the next step is to split the data into parts to be used to create and train the model and for testing and evaluating the model that is produced. Data can be split into test and train[6] or test ,train and cross validation[5].

Image pre-processing: Data pre-processing is an important step in the machine learning model building process because the model can perform well only when the data it is trained on is good and well prepared. It includes converting the image from RGB format into binary format[4].Other steps include resizing , suppressing background pixels, padding, skew correction[4][6].Morphological processing and character segmentation can also be used[9].

Preparing a classification model : There are several classification models that are popular and have been proven to perform with high accuracy . The neural networks have emerged as the fast and reliable tools for classification towards achieving high recognition accuracy.[1] But, mostly CNN model gives high accuracy and is termed as a best model as compared to others[8].Different layers like pooling layer and convolutional layer can be added.[5] Optimizers like Adam and RMSprop are used.[5]

Running predictions on the model : After the model is trained, it is ready for some analysis. In this step, the data that was reserved for testing the model is used to run predictions. The data is blindfolded without any outputs and is passed on . The predicted output is collected for evaluation against the actual results.

Evaluating and visualizing model performance : We observe that the accuracy of a model increases by increment in number of training images[8][6][4], greater the number of epochs run,

higher is the accuracy[6]. Parameters like number of hidden layers, size of hidden layers effect accuracy[4].

3. REQUIREMENTS ANALYSIS

Description: The project is used to recognize handwritten characters. It does so by using Neural Network Models. The model is first trained using a large dataset and its accuracy is obtained. For predicting handwritten characters, the image is first pre-processed. The characters in this image are then predicted by the model.

3.1 Programming Language Used:

a. Python

3.2 Framework:

a. tensorflow

3.3 Libraries:

a. matplotlib
b. tensorflow
c. seaborn
d. keras
e. numpy
f. pandas
g. joblib
h. sklearn

3.4 Features:

➢ *keras.models.Sequential* – Create a sequential model.

➢ *keras.layers* - Keras layers are the primary building block of Keras models. Each layer receives input information, do some computation and finally output the transformed information.

➢ *seaborn.heatmap*- Plot rectangular data as a color-encoded matrix.

➢ *joblib.dump* – Dump our model into a .pkl file.

➢ *sklearn.model_selection.train_test_split* – split our dataset into train and test data

4. MODELLING AND IMPLEMENTATION DETAILS

4.1 Control flow Diagram

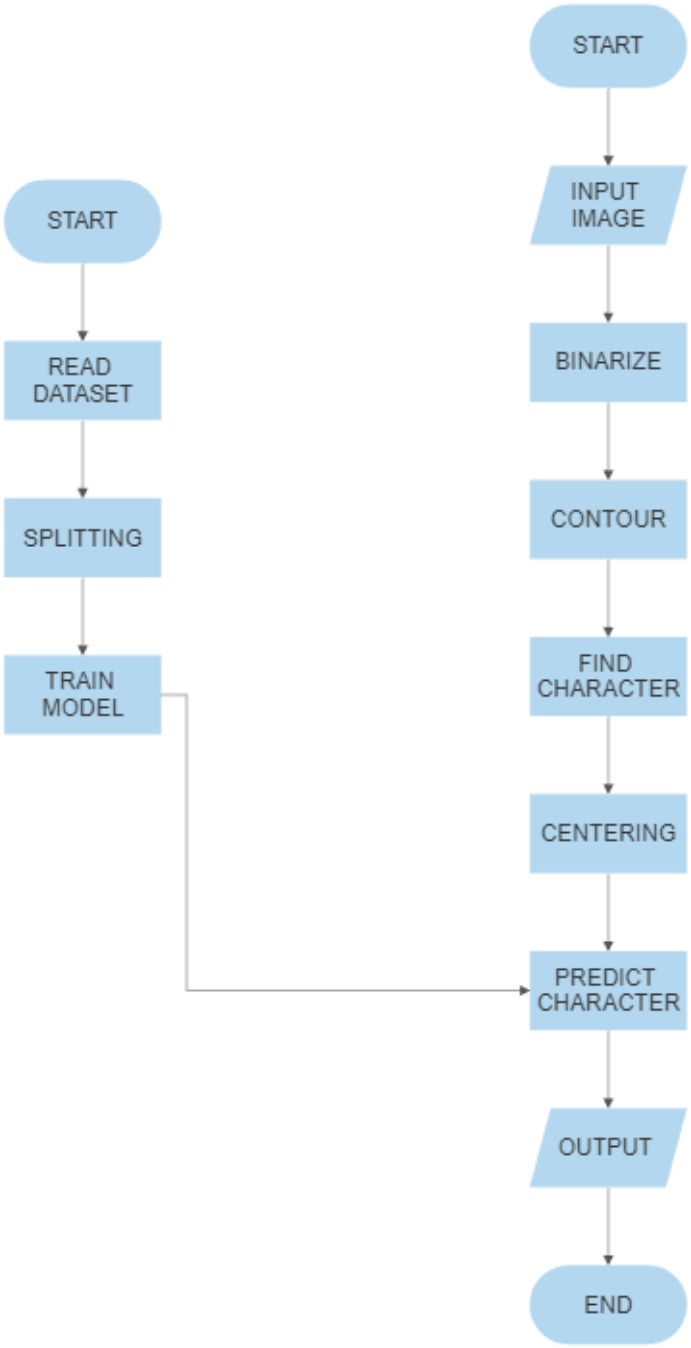


Fig.3 Flowchart

4.2 Implementation details

The objective is to classify and identify the character trained and tested on following dataset:

1) Numerical Digits
2) Devnagri Characters

This Project is composed of 4 phases that are pre-processing, training model, test and evaluation.

4.2.1 Hindi dataset

Our dataset contains 92000 rows (sample images) and 1025 columns. Each image is 32x32 pixels , so the image is split into 1024 columns containing the greyscale value of each pixel (0 to 255). The last column indicates the result of all the 1024 column. There are 2000 images for each character in this dataset. As this dataset is large therefore the model can have improved accuracy. All these data values of pixels are stored in .csv file that has to be read and then it is split for training and testing purposes. The dataset is split such that the 67% of the dataset is used for training model and the other 33% is used for testing the accuracy of our model. After splitting the dataset into X(sample image data) and Y(name of the character). The string format of Y was a hindrance for our model so we have to perform mapping of each string to an integer for ease.

4.2.2 Numerical Dataset

The MNIST database of handwritten digits, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centred in a fixed-size image. MINIST dataset is widely used dataset in machine learning for handwritten recognition, image classification and many more. The dataset contains 6000 images for each digit in training dataset and 1000 images for each digit in test dataset.

The MNIST dataset is short form for the Modified National Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28×28 pixel greyscale images of handwritten single digits between 0 and 9.Handwritten recognition project specifically performs the classification a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9, including both 0 and 9. The MNIST handwritten digit classification problem is a standard dataset used in machine learning.

4.2.3 Convolutional Neural Network

A convolutional neural network is a feed-forward neural network that automatically detects the important features without any human supervision. This is why CNN would be an ideal solution to computer vision and image classification problems. Filters or kernels are the building blocks of CNNs. Kernels are used to extract the relevant features from the input using the convolution operation.

A. *Convolutional layer:* This is the first layer of the convolutional network that performs feature extraction by sliding the filter over the input image. The output or the convolved feature is the element-wise product of filters in the image and their sum for every sliding action.

B. *Pooling layer:* The primary purpose of this layer is to reduce the number of trainable parameters by decreasing the spatial size of the image, thereby reducing the computational cost. The image depth remains unchanged since pooling is done independently on each depth dimension. Max Pooling is the most common pooling method, where the most significant element is taken as input from the feature map. Max Pooling is then performed to give the output image with dimensions reduced to a great extent while retaining the essential information.

C. *Fully Connected Layer:* The last few layers which determine the output are the fully connected layers. The output from the pooling layer is Flattened into a one-dimensional vector and then given as input to the fully connected layer. The output layer has the same number of neurons as the number of categories we had in our problem for classification, thus associating features to a particular label. After this process is known as forwarding propagation, the output so generated is compared to the actual production for error generation. The error is then back propagated to update the filters(weights) and bias values. Thus, one training is completed after this forwarding and backward propagation cycle. The output layer, also known as the feature map, corresponds to original images like curves, sharp edges, textures, etc. In the case of networks with more convolutional layers, the initial layers are meant for extracting the generic features while the complex parts are removed as the network gets deeper.

4.2.4 Open CV2

A. ***imread():***The function imread loads an image from the specified file and returns it

B. ***adaptiveThreshold():*** Applies an adaptive threshold to an array.Adaptive thresholding is the method where the threshold value is calculated for smaller regions and therefore, there will be different threshold values for different regions.

C. ***THRESH_BINARY(thresholdType):*** It uses-

$$dst(x,y) = \begin{cases} \texttt{maxValue} & \text{if } src(x,y) > T(x,y) \\ 0 & \text{otherwise} \end{cases}$$

D. ***ADAPTIVE_THRESH_GAUSSIAN_C (adaptiveMethod):*** threshold value is the weighted sum of neighbourhood values where weights are a Gaussian window.

E. ***findcontours():*** The function retrieves contours from the binary image.

F. ***RETR_EXTERNAL(Contour retrieval mode.) :*** Retrieves only the outer contours. This is the fastest mode.

G. ***CHAIN_APPROX_TC89_L1(Contour approximation method)*** : This is a more accurate approximation algorithms. This should be used when the shapes curved and are not simple polygons.

H. ***moments():***Moment is a particular weighted average of image pixel intensities

I. ***erode():***Erodes away the boundaries of the foreground object.It is used to diminish the features of an image.

5. Findings, Conclusion, and Future Work

5.1 Findings and Result

5.1.1 For handwritten digits

We trained our model for 5 epochs and it is giving an accuracy of 98%. It is capable of recognizing the handwritten digits, we tested our model on 4 handwritten characters of our own unseen images and found out that our model was efficiently detecting the 4 out of them.

```
Epoch 1/5
1875/1875 [==============================] - 8s 4ms/step - loss: 0.2684 - accuracy: 0.9247
Epoch 2/5
1875/1875 [==============================] - 6s 3ms/step - loss: 0.1211 - accuracy: 0.9650
Epoch 3/5
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0841 - accuracy: 0.9744
Epoch 4/5
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0655 - accuracy: 0.9804
Epoch 5/5
1875/1875 [==============================] - 6s 3ms/step - loss: 0.0510 - accuracy: 0.9840
<keras.callbacks.History at 0x7f712e3bc2e0>
```

Fig. 4 Epochs in digit recognition

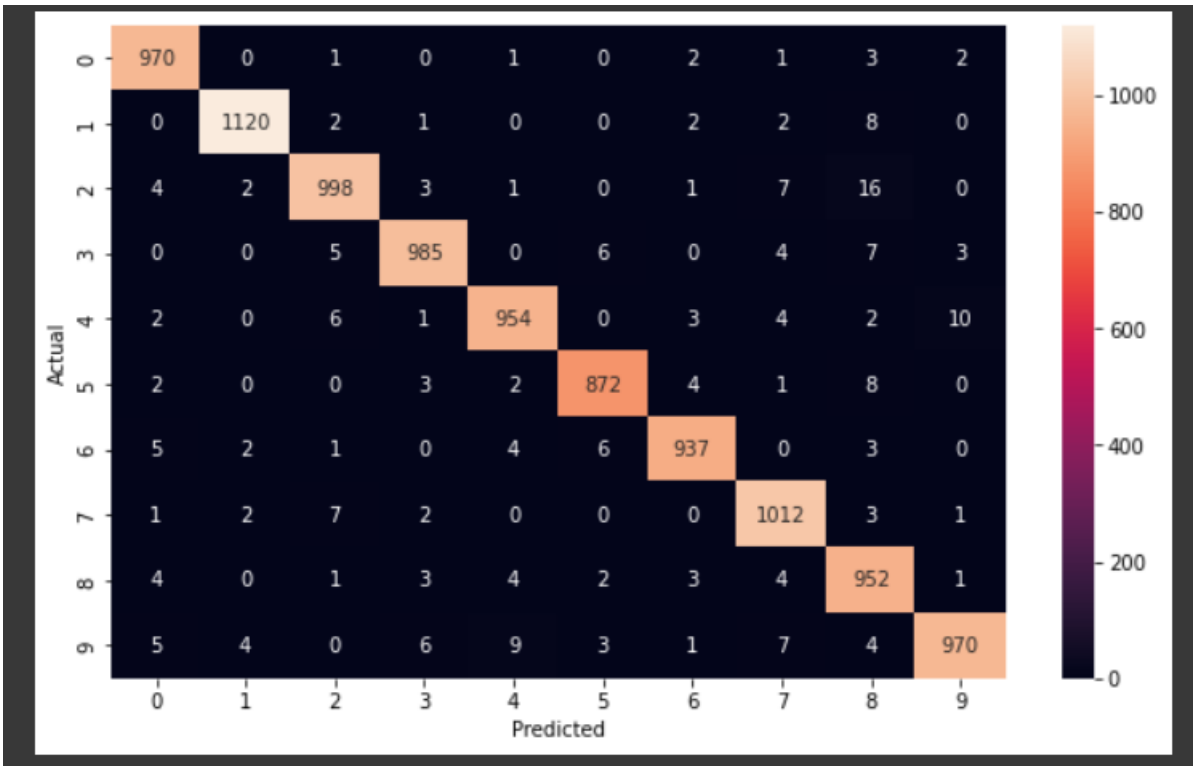


Fig.5 Confusion matrix for digits

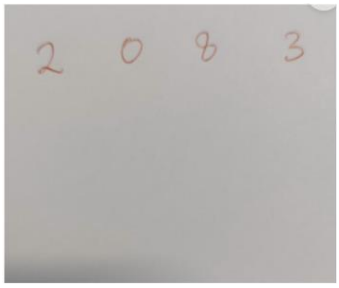
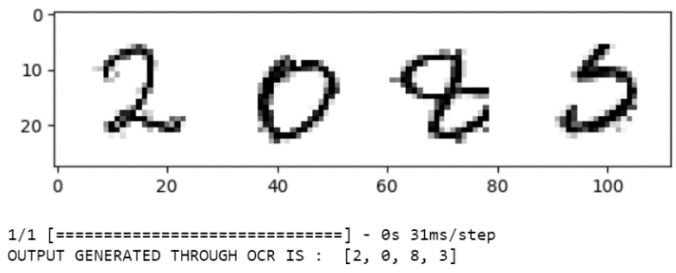


Fig-6 –original digits




```
1/1 [==============================] - 0s 31ms/step
OUTPUT GENERATED THROUGH OCR IS :  [2, 0, 8, 3]
```

fig-7-predicted digits

5.1.2 For handwritten Hindi character

We trained our model for 4 epochs and it is giving an accuracy of 96%. It is capable of recognizing the handwritten Hindi character , we tested our model on 3 handwritten characters of our own unseen images and found out that our model was efficiently detecting the 3 out of them.

```
Epoch 1/4
1927/1927 [==============================] - 134s 69ms/step - loss: 1.0888 - accuracy: 0.6854 - val_loss: 0.2523 - val_accurac
y: 0.9297
Epoch 2/4
1927/1927 [==============================] - 133s 69ms/step - loss: 0.4892 - accuracy: 0.8508 - val_loss: 0.1706 - val_accurac
y: 0.9515
Epoch 3/4
1927/1927 [==============================] - 133s 69ms/step - loss: 0.3615 - accuracy: 0.8878 - val_loss: 0.1431 - val_accurac
y: 0.9582
Epoch 4/4
1927/1927 [==============================] - 132s 68ms/step - loss: 0.2979 - accuracy: 0.9074 - val_loss: 0.1167 - val_accurac
y: 0.9652
```

Fig-8 Epochs for training Hindi model

```
949/949 [==============================] - 13s 14ms/step - loss: 0.1167 - accuracy: 0.9652
[0.11672161519527435, 0.9652174115180969]
```

Fig-9 Test Hindi accuracy

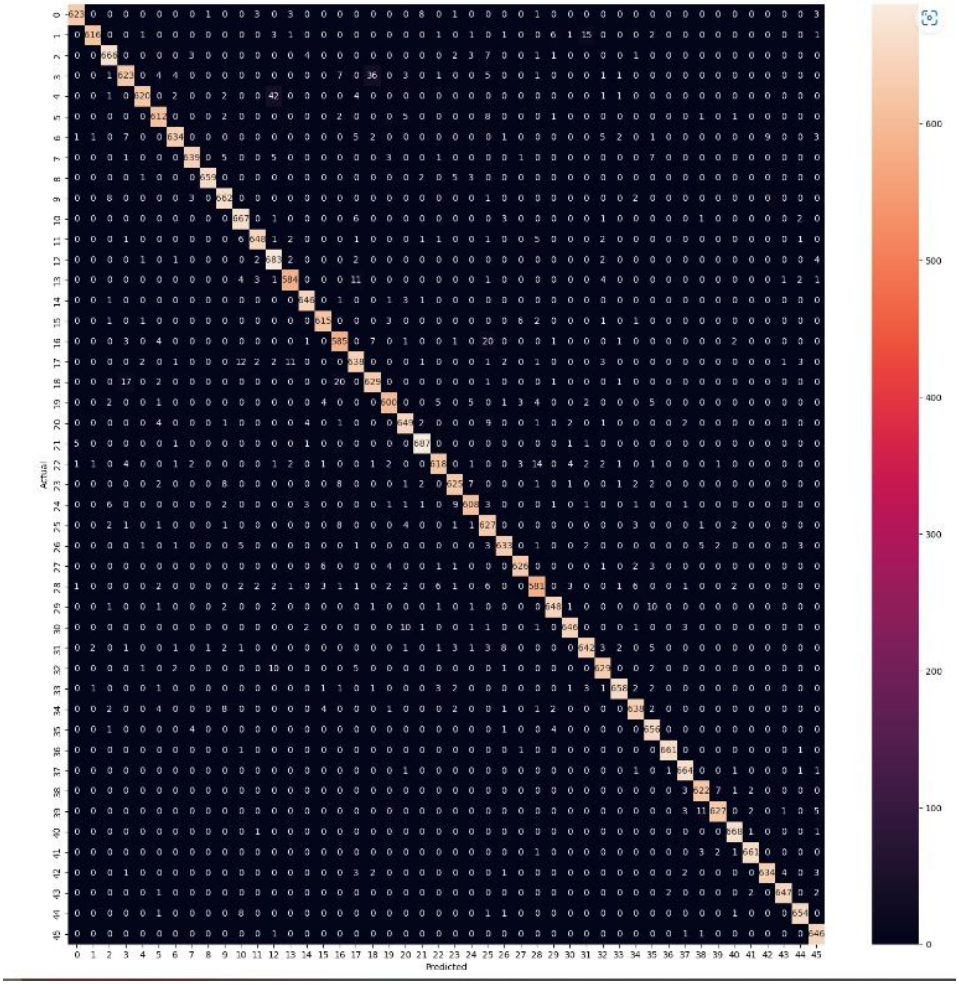


Fig.10 Confusion Matrix for Hindi Characters

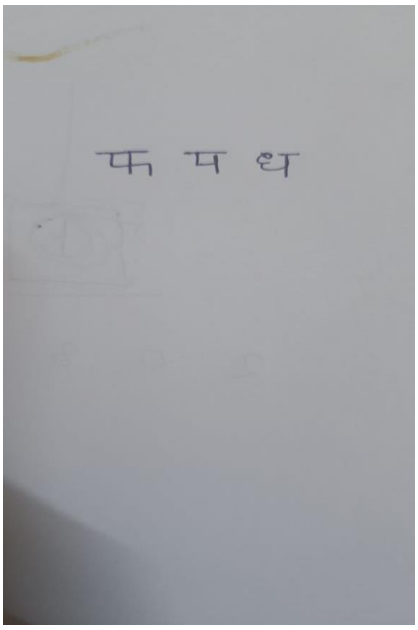


Fig.11 Original Hindi Characters

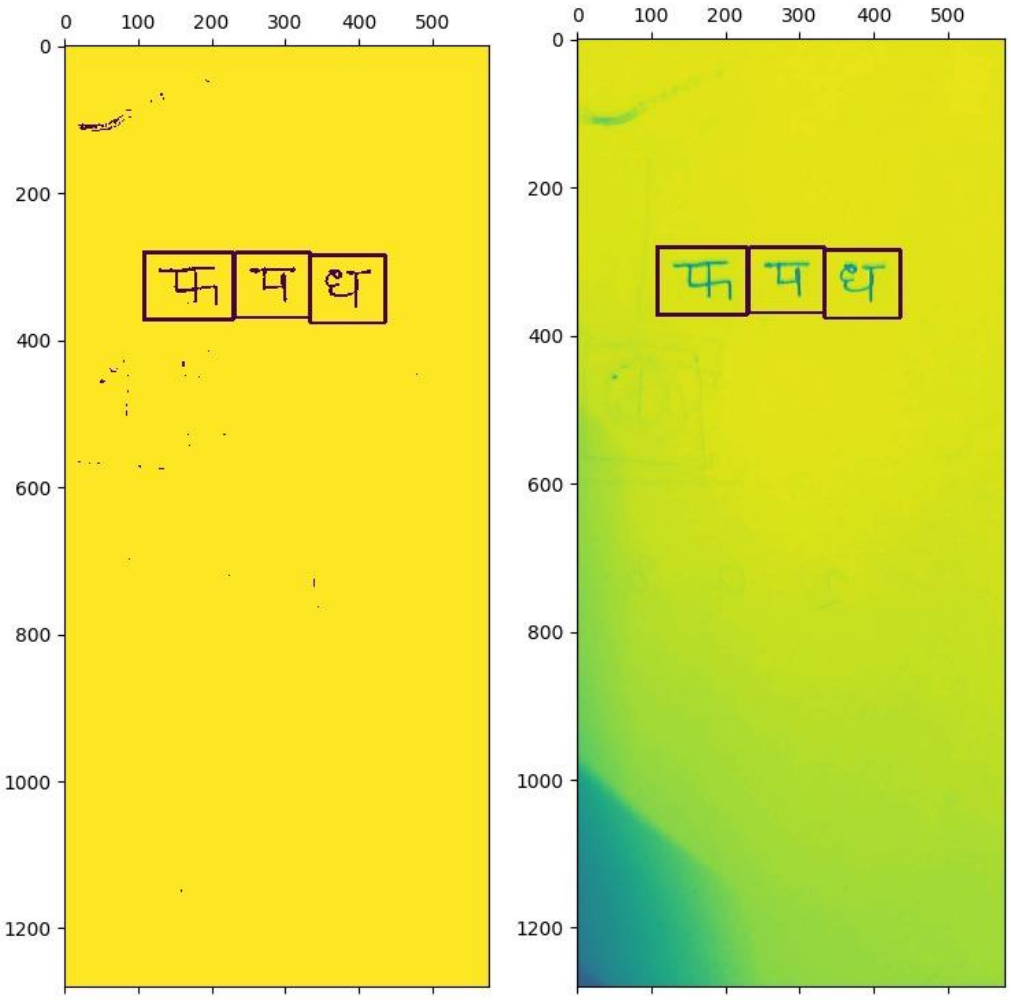


Fig12. Contoured Hindi Characters

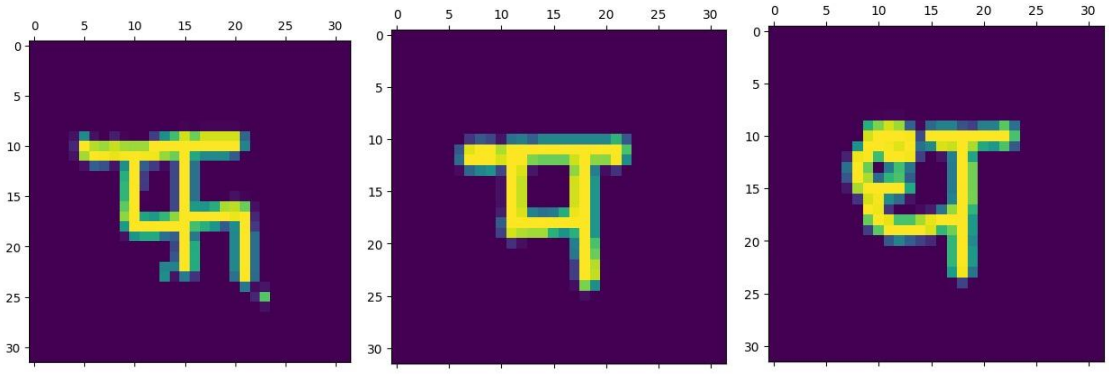


Fig-13 Pre Processed Hindi characters

```
1/1 [==============================] - 0s 22ms/step
Total output:
फ
प
ध
```

Fig-14 Predicted Hindi Characters

5.2 Conclusion

In this study, Neural Networks are used to create a model for the prediction of handwritten digits and Hindi characters. It can be observed that by proper pre-processing and training of model, accurate prediction of handwritten characters can be done. This prediction can be utilized to recognize and type Hindi characters easily.

5.3Future Work

The model can be refined by adding more layers and more data so that better results could be obtained. The model can be used to make websites and applications that provide Hindi OCR.

REFERENCES

1. Pradeep, J., Srinivasan, E., & Himavathi, S. (2011, April). Diagonal based feature extraction for handwritten character recognition system using neural network. In *2011 3rd international conference on electronics computer technology* (Vol. 4, pp. 364-368). IEEE.
2. Pal, U., Wakabayashi, T., & Kimura, F. (2009, July). Comparative study of Devnagari handwritten character recognition using different feature and classifiers. In *2009 10th International Conference on Document Analysis and Recognition* (pp. 1111-1115). IEEE.
3. Chen, L., Wang, S., Fan, W., Sun, J., & Naoi, S. (2015, November). Beyond human recognition: A CNN-based framework for handwritten character recognition. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)* (pp. 695-699). IEEE.
4. Patel, C. I., Patel, R., & Patel, P. (2011). Handwritten character recognition using neural network. *International Journal of Scientific & Engineering Research*, *2*(5), 1-6.
5. Chaudhary, D., & Sharma, K. (2019, March). Hindi Handwritten Character Recognition using Deep Convolution Neural Network. In *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 961-965). IEEE.
6. Acharya, S., Pant, A. K., & Gyawali, P. K. (2015, December). Deep learning based large scale handwritten Devanagari character recognition. In *2015 9th International conference on software, knowledge, information management and applications (SKIMA)* (pp. 1-6). IEEE.
7. Shamim, S. M., Miah, M. B. A., Angona Sarker, M. R., & Al Jobair, A. (2018). Handwritten digit recognition using machine learning algorithms. Global Journal Of Computer Science And Technology.
8. Khandokar, I., Hasan, M., Ernawan, F., Islam, S., & Kabir, M. N. (2021, June). Handwritten character recognition using convolutional neural network. In Journal of Physics: Conference Series (Vol. 1918, No. 4, p. 042152). IOP Publishing.
9. Kumar, G., Bhatia, P. K., & Banger, I. (2013). Analytical review of preprocessing techniques for offline handwritten character recognition. International Journal of Advances in Engineering Sciences, 3(3), 14-22.
10. Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine, 29(6), 141-142.