International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

# HANDWRITTEN CHARACTER RECOGNITION

Malla Bala Sowmya[1], Gurram Akhila[2], Gullena Sangeetha[3], Shugufta Fatima[4]

Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Telangana, India

## Abstract:

Handwritten characters are seen everywhere in our day-to-day life. Almost all the things we do involve letters, from writing cheques to writing notes manually. Handwritten character recognition is considered as a core to a diversity of emerging application by using the concepts of machine learning. It is used widely for performing practical applications such as reading computerized bank cheques. However, executing a computerized system to carry out certain types of duties is not easy and it is a challenging matter. There is huge variability and ambiguity of strokes from person to person. Handwriting style of an individual person also varies from time to time and is inconsistent. There are many challenges which we have to deal with while understanding handwritten text. Poor quality of the source document/image due to degradation over time can affect the way of understanding the characters. However, we can find a solution for this using machine learning. This paper illustrates a model which interprets handwritten character accurately with the help of data set which we used to train the data model. The main objective of this paper is to ensure effective and reliable approaches for recognition of handwritten characters.

***Keywords:*** Machine learning, neural networks, pooling, binarization, feature extraction, segmentation, pre-processing, image processing, classification, subsampling, digitalization.

## 1.Introduction

**Handwriting recognition:** Handwriting recognition is the ability of a computer or device to take as input handwriting from sources such as printed physical documents, pictures and other devices, or to use handwriting as a direct input to a touchscreen and then interpret this as text. The input is usually in the form of an image such as a picture of handwritten text that is fed to a pattern-recognition software, or as real-time recognition using a camera for optical scanning.

**Character recognition:** Character recognition is a process which allows computers to recognize written or printed characters such as numbers or letters and to change them into a form that the computer can use. It is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes.

**Handwritten character recognition**: Handwritten character recognition (HCR) is the detection of characters from images, documents and other sources and changes them in machine-readable shape for further processing. The accurate recognition of intricate-shaped compound handwritten characters is still a great challenge. Handwritten character recognition is the way in which we detect handwritten characters and with the help of trained model, we can predict its actual value.

**EMNIST Dataset:** The EMNIST data set is a set of handwritten characters and numbers from the NIST Special Database 19, which can convert into a 28×28-pixel image format and data set structure that directly matches the data set. The image structure is directly matched to the data set used. The testing set and training set appear in the list as a list. Each item in the external list represents an image. The internal list represents the intensity values of 784 pixels, with a range of 0-225. Each item in the external list represents an image. The internal list represents the intensity value of 784 pixels, and the range is 0-225. The test image and the train image have a white foreground and a black background. Both the test image and the training image are flipped horizontally and rotated 90 degrees clockwise. The dataset provides two file formats. Both versions of the data set contain the same data and fully apply for convenience. The first data set is provided in Matlab format, available through Matlab and Python. There are six different splits provided in this dataset. These are EMNIST by class, EMNIST ByMerge, EMNIST Balanced, EMNIST Letters, and EMNIST MNIST.A summary of the dataset is provided. The dataset includes EMNIST Byclass, EMNIST ByMerge, EMNIST Balanced, EMNIST Letters, EMNIST Digits, and

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

EMNIST MNIST. The EMNIST Balanced dataset would use in this project. The EMNIST Balanced dataset mean to address the balance issues in the class and ByMerge dataset. It reduces misclassification errors due to capital and lowercase letters and has an equal number of samples per class. This dataset is most applicable among the other datasets.

**Convolutional Neural Network:** A convolutional neural network is a particular multilayer neural network. CNN is the most current neural network and commonly used for analysing visual data. It is also widely used in natural language processing, image and video recognition, a recommendation system, and a pattern recognition system. A convolutional neural network is a biologically inspirited neural network and very good at image recognition. A neural network is a system of interconnected artificial neurons that can exchange information with each other. Its connection has a uniform numeric weight during training. Therefore, when a well-trained network uses images or patterns for recognition, it will respond accurately. Convolutional nerves have the same function as brain neural networks. Convolutional neural networks are composed of neurons with learnable weights and biases. Each neuron receives several inputs, performing weighted summation on them, passing them to activate the function, then the output response.

The network consists of a multilayer feature detecting "neuron" composition. Each layer has many neurons that respond to multiple input tissues from previous layer. CNN will build layers so that the first layer can detect a set of original models in the input. The second layer detects model patterns, and the third layer detects models of those patterns. A “labelled” dataset of inputs and the input patterns marked with their dedicated output response would display the training process. The training process is the general method to determine the mediate weight and final feature neurons. CNN consists of one or more convolutional layers, usually with a sub-sampling layer. These layers are one or more fully connected layers that follow. The layers would connect the same as in standard neural networks. In a CNN, convolution layers act as a feature extractor. But they are not designed by hand. The right convolution filter kernel focuses on the training process is determined. Convolutional layers can excerpt data features as they limit the receiving field of the hidden layer to confined. CNN can use in various domains includes pattern and image recognition and video analysis. There are many reasons convolutional neural network becomes more and more important for the handwritten recognition system. In the conventional models used for image recognition, the feature extractor is a manual design. The layer fully connects for determining the classification in the training process. CNN improved network architecture may save computational complexity and memory requirements required. At the same time, it provides better enforcement for the application. CNN generally consists of three layers. CNN structure is including convolutional layers, sub-sampling layers, fully connected layers, and layers using SoftMax. These layers can implant into another layer like the SoftMax layer. Each layer would connect to the preceding layer. The softmax layer improves the accuracy of detecting the image. The creases applied to the CNN are not always the same as needed. The difference is that the language that can recognize handwriting will influence the number of layers. The number of layers would use in the future. When another layer is interposed CNN, handwriting recognition accuracy will be higher than that softmax inserted layer CNN in the model.

## 1.1 About project

Handwritten character recognition (HCR) is a mechanism which enables to translate different types of documents into analysable, editable and searchable data. An ultimate aim of HCR is to emulate human reading capabilities in such a way that the machine can read, edit and interact with text as a human in short time. Identification of HCR has drawn great attention of numerous researchers over half a century, and many great achievements have been made in this field. However, in the past years, a significant progress is made on HCR performance, but still now HCR is a challenging task due to the great diversity of handwriting style, the existence of many similar characters and large number of character categories.

To perform handwritten character recognition, we used the CNN model so as to predict accurate character value. We took image of handwritten character as input and we displayed predicted character as output. The input we took undergoes various process of segmentation, binary conversion, feature extraction, recognition so as to give us appropriate output. Convolutional Neural Network (CNN) is a very well-known deep learning architecture motivated by the natural visual perception technique of human brain. Taking advantage of the recent exponential growth in the volume of annotated data and the rapid increases in the capabilities of the graphics processing units, the study on CNN has quickly arisen and obtained state-of-the-art performance on different tasks, e.g., image classification, text detection, pose estimation, object tracking, action detection, visual saliency detection, scene marking, speech and natural language processing. Although there are many variants of CNN architectures, their basic elements are very similar. It comprised of three types of layers, as convolutional, pooling, and fully-connected layers.

Usually, handwritten character recognition models might be in a position to understand upright character or bold character, but here we are implemented on various images where handwritten characters are slightly tilted to the right (most of handwritings are tilted), slightly blurred, overwritten.

## 1.2 Objectives

- To develop a model based on convolution neural network for handwritten character recognition.
- To analyse performance of proposed algorithm with test dataset.
- Recognizing characters with certain exceptions like:
  - Connected characters
  - Overlapping Characters
  - Characters which look similar
- To avoid misinterpretation of characters in:
  - Bank cheques
  - Notes and exam papers
  - Doctors Prescriptions
- To increase efficiency in identifying characters.

## 1.3 Scope

- To interpret various handwritten files like:
  - Banking applications
  - Handwritten mails
  - Application forms
  - Old scripts
- In future we are expecting to explore the Back-propagation algorithm to make recognition of characters faster and more efficient and improve the overall performance.
- It becomes vital scope and it is appealing many researchers because of its using in variety of machine learning and applications.
- This application can be used in future for recognizing the alphabets by scanning of the paper documents or images directly through system.
- This project provides the precise alphabet along with the accuracy up to which the character is determined.

## 1.4 Advantages

1. Higher productivity

Handwritten character recognition software helps businesses to achieve higher productivity by facilitating quicker data retrieval when required. The time and effort which the employees were required to put in for extracting relevant data can now be channelized to focus on core activities.

2. Cost Reduction

Opting for handwritten character recognition will help businesses on cutting down on hiring professionals to carry out data extraction, which is one of the most important benefits of handwritten character recognition data entry methods. Therefore, HCR eliminates the cost of misplaced or lost documents.

3. High Accuracy

One of the major challenges of data entry is inaccuracy. Automated data entry tools such as handwritten character recognition data entry result in reduced errors and inaccuracies, resulting in efficient data entry. Besides, problems like data loss can also be successfully tackled by character recognition data entry.

4. Increased Storage Space

Handwritten character recognition can scan, document, and catalogue information from enterprise-wide paper documents. This simply means that the data can now be stored in an electronic format in servers, eradicating the need for maintaining huge paper files. In this way, handwritten character recognition data entry serves as one of the best tools to implement "Paperless" approach across the organization.

5. Superior Data Security

Data security is of utmost importance for any organization. Paper documents are easily prone to loss or destruction. Papers can be misplaced, stolen, or destroyed by natural elements such as moisture, pests, and fire. However, this is not the case with data that is scanned, analysed, and stored in digital formats. Furthermore, the access to these digital documents can also be minimized to prevent mishandling of the digitized data.

6. Makes Documents Editable

Scanned documents need to be edited most of the time, particularly when some information must be updated. OCR converts data to any preferred formats such as Word, etc., which can be easily edited. This can be of great help when there are contents which have to be constantly updated or regularly changed.

7. Disaster Recovery

Disaster recovery is one of the major benefits of using character recognition for data entry. When data is stored electronically in secure servers and distributed systems, it remains safe even under emergency situations. When there are sudden fire breakouts or natural calamity, the digitized data can be quickly retrieved to ensure business continuation.

## 1.5 Disadvantages

• Handwritten character recognition text works efficiently with the written text only and not with printed text. Handwriting must be learnt by the pc.

• Handwritten character recognition systems are expensive.

• There is the need of lot of space required by the image produced.

• The quality of the image can be lost during this process.

• All the documents got to be checked over carefully then manually corrected.

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

• Not worth doing for little amounts of text.

## 1.6 Applications

Handwritten character recognition has various applications across various sectors:

• **Banking**: in banking sector HCR can be used for reading various forms used in banks like reading bank cheques, bank forms that are handwritten

• **Postal**: Here letters that are handwritten can be interpreted into digital text that makes work simple. From postal address to information written in the post can be digitalised.

• **Digital libraries**: HCR plays an important role for digital libraries, allowing the entry of image textual information into computers by digitization, image restoration, and recognition methods. Many handwritten ancient scripts can be digitalised by using HCR. which helps us to understand and visualise ancient knowledge.

• **Medical sector**: There are many scripts in ayurveda that are handwritten, those scripts can be retrieved. Handwritten doctor prescriptions can be digitalised that makes understanding and passing information easy.

## 1.7 Hardware software and software requirements

• Hardware requirements:

o System requirements

o Windows 10 operating System

o I3 processor

o 128 MB RAM

• Software requirements:

o Environment: Jupyter notebook

o Packages:

• Numpy: it provides an efficient interface to store and operate on dense data buffers

• Pyplot: pyplot is a collection of functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

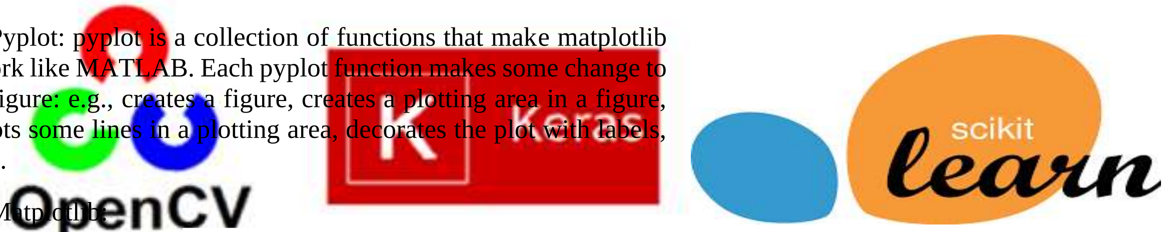



Fig 1.1: software requirements and packages used

• Matplotlib: Matplotlib supports various types of graphical representations like Bar Graphs, Histograms, Line Graph, Scatter Plot, Stem Plots, etc.

• Keras: Keras is a powerful and easy-to-use free open-source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

• Opencv: OpenCV is a Python open-source library, which is used for image processing and computer vision in Artificial intelligence, Machine Learning, face recognition, etc.

• 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 modelling including classification, regression, clustering and dimensionality reduction via a consistence interface.

• Pandas: Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is used to load data.

## *2. LITERATURE SURVEY*

K. Gaurav, Bhatia P. K. Et al[1], this paper deals with the various pre-processing techniques involved in the character recognition with different kind of images ranges from a simple handwritten form based documents and documents containing colored and complex background and varied intensities. In this, different preprocessing techniques like skew detection and correction, image enhancement techniques of contrast stretching, binarization, noise removal techniques, normalization and segmentation, morphological processing techniques are discussed. It was concluded that using a single technique for preprocessing, we can't completely process the image. However, even after applying all the said techniques might not possible to achieve the full accuracy in a preprocessing system.

Sandhya Arora[2], used four feature extraction techniques namely, intersection, shadow feature, chain code histogram and straight-line fitting features. Shadow features are computed globally for character image while intersection features, chain code histogram features and line fitting features are computed by dividing the character image into different segments. On experimentation with a dataset of 4900 samples the overall recognition rate observed was 92.80% for Devanagari characters.

Nafiz Arica at al.[3] proposed a method which avoids most of the pre-processing operations, which causes loss of important information. One of the major contributions of the method is to development of a powerful segmentation algorithm. Utilization of the character boundaries, local maxima and minima, slant angle, upper and lower baselines, stroke height and width, and ascenders and descenders improve the search algorithm of the optimal segmentation path, applied on a gray-scale image. This approach decreases the over-segmentation.

J.P.Premi a, R.Madhumithab, N.R.Raajan[4] proposed a method in which OCR is implemented using artificial neural networks. Optical Character Recognition for text recognition from the images which could be in any of the forms machine learning approach to recognize the characters using CNN and the accuracy is found to be 73% approximately within a fraction of second. Later the recognized images are converted into the text file and then get translated into the preferable languages of handwritten text files as well as from the ancient manuscript (Language-English).

Chooi Shir Ley1, Aimi Syammi Binti Ab Ghafar[5] proposed a method to develop a model base on CNN that can identify the handwritten character from the EMNIST with high accuracy. The NIST dataset consists of English alphabets and digits that use to train the neural network. NIST datasets are downloaded from the Kaggle. They improvised the previous model by implementing an optimization technique that can increase the handwritten character recognition accuracy.

Monika, Monika Ingole, Khemutai Tighare[6] gave a brief review on implementation of neural networks on handwritten characters. In this paper they gave a method that has to be implemented to construct neural networks ad how can image be processed to fetch predicted output.

S. Anandh Kishan, J. Clinton David, P.Sharon Femi[7], In this Paper they have given a brief introduction on how to test and implement EMNIST dataset to train the model. They have mentioned about the image processing using dataset and their dimensionality have been mentioned.This paper helped me understand image preprocessing and segementation.

Junlei Song, Qi Liu, Shengnan Tian, Yi Wei, Fang Jin, Wenqui Mo, Kaifeng Dong[8], In this paper they introduced mixed programming for visualization of image using matlab. It also provides the brief understanding of the matlab in visualizing the character using grayscale.

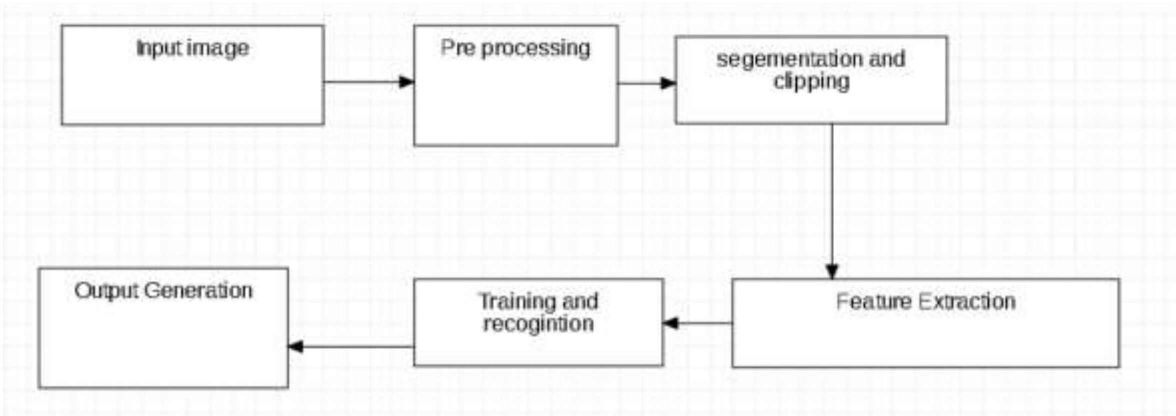
Mr.Narendrasing.B Rajput, Prof.S.M Rajput, Prof.S.M.Badave,[9] in this paper they have proposed character recognition techniques for identifying the character based on the dataset. They have proposed preprocessing, segmentation, image processing techniques in this paper.

Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir[10] in this paper they proposed convolutional neural network (CNN) have made great progress in HCR by learning distinct characteristics from large amounts of raw data. They have also proposed the conceptual understanding of CNN and its capability to recognize the image from dataset which is used to determine the accuracy with help of testing and training the dataset.

Thatikonda Somashekar[11], in this paper he presented different methods used for feature extraction, and classification and their results. This paper gave a clear idea on all concepts used and their respective accuracy rate. Our view of process behind handwritten character recognition is explain in a brief way

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

## 2.1 Existing System




• In the present world scenario as different persons have different hand writing styles. When the alphabets are hand written it becomes difficult for the evaluator to recognise the alphabets because of variance in size, translation, stroke thickness, rotation.

• There must also be a person to enter the details like name, amount numbers written in words on cheque written by users which is a time taking process as the person must evaluate the alphabets.

• There are also chances that the alphabets may get entered in incorrectly as there is similarity between numbers such as g and y, i and l, u and v when written through cursive handwriting.

• There exists system which can scan printed bold text and there even exists system where we can write our handwritten character directly onto touch pad but there doesn't exist a system which can interpret handwritten character.

## 2.2 Proposed System

• Character recognition system is the working of a machine to train itself for recognizing the alphabets which is solution for the existing problem.

• Recognizing alphabets from different sources like bank cheque, papers, images, written using tablets, ipad, postal cards, doctor prescriptions.

• In this project we use machine learning algorithm. We used CNN algorithm to interpret images of handwritten character.

• The uniqueness and variety in the handwriting of different individuals also influence the formation and appearance of the alphabets.

• The aim of a handwriting recognition system is to convert handwritten characters into machine understandable format using CNN with accuracy.

## *3. PROPOSED ARCHITECTURE*

- **Image as input**

The image acquisition step involves acquiring an input image containing handwriting. In this case, the image should be in a specific format, such as PNG. The data and images of handwritten character can obtain from the MNIST database. EMNIST database can download from the Kaggle. The Kaggle database has stored several EMNIST dataset, such as PNG, CSV and TXT. The image for predicting the output can acquire through a digital camera, scanner, or any other suitable input device.

- **Pre-processing**

Fig 3.1: Proposed architecture

Pre-processing of the input data acquired pre-treatment. The Pre-processing improves the input data quality to make the input sample more suitable for the next stage of the recognition system. The pre-processing has the functions of noise removing, binarization, edge detection, segmentation, dilation, and fill in. Pre-processing can normalize the strokes. It can also eliminate changes that reduce accuracy.

The pre-processing mainly deals with various deformation. For instance, sawtooth size, missing pen point during movement, the space around the curved and uneven.

• **Noise removing:** Uniform filtering and non-uniform filtering methods can use to eliminate unnecessary or unwanted patterns. Due to the influence of noise, there may be separate line segments, such as large gaps between lines. Therefore, it is significant to remove all these errors. It can make the information retrieval in the best way. There are many noises in the image. One type of additive noise is called "salt and pepper noise." The black and white dots scattered on the image usually look like salt and pepper and can be observed in almost all documents. Noise reduction techniques split into two categories, namely filtering and morphological operations.

• **Filtering**: Filtering intends to eliminate noise and reduce stray points usually caused by uneven writing surfaces and a low sampling rate of data acquisition equipment and various design and spatial frequency domain filter for filtering

• **Binarization:** Binarization is the process of converting character images into binary 0 and 1 forms. In this case, all types of characters will interpret as grayscale pictures. After converting the grayscale image into a binary matrix, all image characters will be captured vertically. Similarly, the grey image may work to reduce the overall complexity of the system.

• **Edge detection**: Edges define object boundaries, so these object boundaries are useful in phase lie recognition and segmentation. Edge detection can be improved for all exception filter and reduce the unnecessary amount of information. On the other hand, it can retain all the significant features of the image. There are many methods of performing edge detection, usually divided into gradient or Laplacian operators. Observe the highest and smallest values of the first derivative of the image, and use the gradient method to detect edges. Then, search for zero crossings in the second derivative to identify image edges.

• **Thresholding**: Colour or grey images will be represented as numbers 0 or 1 to reduce storage requirements and increase processing speed. It is a binary image, meaning that every value above the selected threshold is 1 and 0. When the image is converting to grayscale, the handwritten characters are darker than their background. Threshold processing means can separate darker and lighter areas from the image.

• **Dilation and fill in**: The document may be skewing during scanning. There are several commonly used methods to detect page tilt. One is to identify the connected components and find the average angle of the centroids that connect them. Skew should eliminate as this will reduce the accuracy of the document. After that, with the help of calculating the skew angle, to horizontal the skewed line.

- **Segmentation**

Segmentation is dividing the input text data image into several lines and then into individual characters. It removes unnecessary parts from the data image. Segmentation is subdivided into two segments, including external and internal subdivision. The external segment is sub-divided into paragraphs, lines, and words. On the other hand, internal segmentation is to segment a single character from the input text data. In this project, the technology uses words, lines, and character segmentation. It usually performs by segmenting a single character from the word picture. Besides, content is processing in a tree-like way. In the initial scheme, the line histogram would use to segment the line. Then, at each level, the characters are retrieved through a histogram technique and finally retrieved.

- **Feature Extraction**

Feature extraction is a significant substantive data retrieved from the original data content method. The essential materialist data is an accurate and efficient embodiment of the character image. Feature extraction points to recognize the extraction of patterns that is most important for the classification. Extracted from the original data set of features to maximize character recognition rate (includingminimal elements) is called feature extraction. The feature extraction process is to collect the difference. Gradient-based feature extraction is extracting features from necessary and useful data. These features are very significant for any training system that performs pattern recognition tasks. The capacities used in feature extraction include indexing and labeling, boxing and cropping, and reshaping and resizing.

• **Indexing and Labeling**: Through this process, different characters in the index image are marked. Thus, it helps in the character classification in the image and make feature extraction of character extraction.

• **Boxing and Cropping**: It is the process of creating a border around the characters represented in the image. It can make cropping of characters easier. After boxing, cut out characters to store them as input variables for recognition.

• **Reshaping and Resizing**: Reshape the image to change the size of the received characters to the aspired contour. Resize can reduce the character size to the minimum level.

- **Training and Recognition**

After feature extraction, the next step is classification. The classification process is to make decisions for a new input character like this to suit the classes or appearance. The stage of categorizing characters is recognized and assigned labels. Good performance depends on the classification and feature extraction selection. When the input image uses as the input sample for a handwritten character recognition system, all significant features will be retrieved and input into a trained classifier, such as an artificial neural network. Preparation of the input feature is compared with the stored patterns to find a matching category of the input image. It would complete with the help of the classifier. The correct label training data is needed to analyze the test samples. Then, train the convolutional neural network model in Python. Perform training with weight decay and exit on each fully connected layer. The recognition process is done using three layers of CNN. CNN uses filters on the pixels of any image to learn detailed patterns compared to global patterns with a traditional neural network. To create CNN, we have to define:

1. **A convolutional Layer**: Apply the number of filters to the feature map. After convolution, we need to use a relay activation function to add non-linearity to the network.

2. **Pooling Layer**: The next step after the Convention is to down sampling the maximum facility. The objective is to reduce the mobility of the feature map to prevent overfitting and improve the computation speed. Max pooling is a traditional technique, which splits feature maps into subfields and only holds maximum values.

3. **Fully connected Layers**: All neurons from the past layers are associated with the other next layers. The CNN has classified the label according to the features from convolutional layers and reduced with any pooling layer.

This proposed approach employs convolutional layer, max pooling layer and dropout layer, flatten layer and dense layer. The image is firstly passed into a convolutional layer for two times which apply certain extracted features on the filter. The max pooling layer was used to reduce the spatial size of given image which further transferred into a dropout layer which prevents overfitting.

- **Output Recognition**

We use opencv to read image and after undergoing all the processes mentioned above final image is used to make predictions. The output screen contains image and its predicted value. Here the outline of character is highlighted by representing it in RGB shade and the predicted value is displayed below. After execution screen is displayed which contains result.

## *4. IMPLEMENTATION*

### 4.1 Algorithm

Convolution Neural Network (CNN) is a deep learning algorithm that is widely used for image recognition and classification. It is a class of deep neural networks that require minimum pre-processing. It inputs the image in the form of small chunks rather than inputting a single pixel at a time, so the network can detect uncertain patterns (edges) in the image more efficiently.

CNN contains 3 layers namely, an input layer, an output layer, and multiple hidden layers which include Convolutional layers, Pooling layers (Max and Average pooling), Fully connected layers (FC), and normalization layers. CNN uses a filter (kernel) which is an array of weights to extract features from the input image.

Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

• The first hidden layer is a convolutional layer called a Convolution2D and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a has feature maps, which with the size of 5×5 and a rectifier activation function.

• Next, we define a pooling layer that takes the max called MaxPooling2D.

It is configured with a pool size of 2×2. Pooling layers, also known as down sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights.

Max pooling: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

• The next layer is a regularization layer using dropout called Dropout. It is configured to randomly exclude 20% of neurons in the layer in order to reduce overfitting.

• Next is a layer that converts the 2D matrix data to a vector called Flatten. It allows the output to be processed by standard fully connected layers.

• Next a fully connected layer with neurons This layer has node in the output layer connects directly to a node in the previous layer.

• This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use Relu functions, FC layers usually leverage a SoftMax activation function to classify inputs appropriately, producing a probability from 0 to 1.

• Finally, the output layer that makes the prediction for the given input.

CNN employs different activation functions at each layer to add some non-linearity. As we move into the CNN, we observe the height and width decrease while the number of channels increases. Finally, the generated column matrix is used to predict the output.

Convolutional neural networks now provide a more scalable approach to image classification and object recognition tasks, leveraging principles from linear algebra, specifically matrix multiplication, to identify patterns within an image.

CNNs eliminate the need for manual feature extraction. The features are learned directly by the CNN and produce highly accurate recognition results.

This algorithm can be retrained for new recognition tasks, enabling you to build on pre-existing networks.

The ability of convolution neural networks is to automatically learn a huge number of filters in parallel and which is specific to a training dataset under the constraints of a specific predictive modelling problem, such as image classification.

Training a network is a process of finding kernels in convolution layers and weights in fully connected layers which minimize differences between output predictions and given input values on a training dataset.

Overfitting refers to a situation where a model learns statistical data specific to the training set and ends up memorizing the irrelevant noise instead of learning the signal, and, therefore, performs less well on a subsequent new dataset.

## 4.2 Code

**#importing packages**

from keras.datasets import mnist

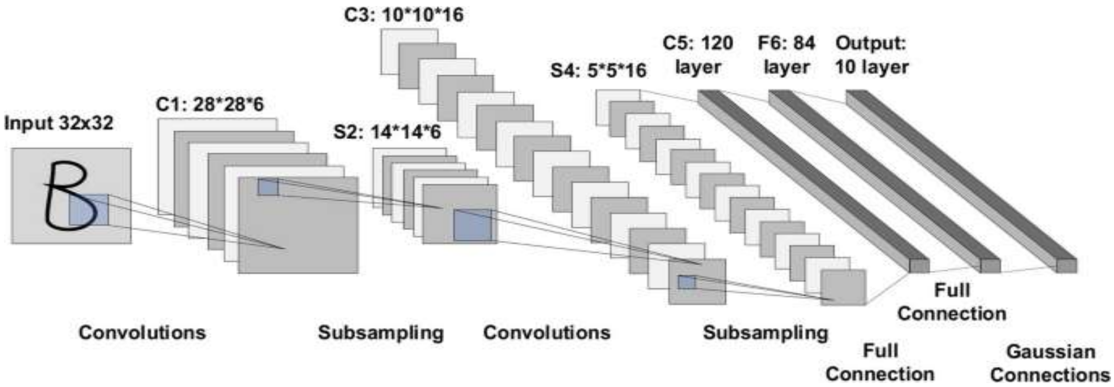



Fig 4.1: Convolutional Neural Network model

import matplotlib.pyplot as plt

import cv2

import numpy as np

from keras.models import Sequential

from keras.layers import Dense, Flatten, Conv2D, MaxPool2D, Dropout

from keras.optimizers import SGD, Adam

from keras.callbacks import ReduceLROnPlateau, EarlyStopping

from keras.utils import to_categorical

import pandas as pd

import numpy as np

from sklearn.model_selection import train_test_split

from keras.utils import np_utils

import matplotlib.pyplot as plt

from tqdm import tqdm_notebook

from sklearn.utils import shuffle

**# Read the data...**

data = pd.read_csv(r"C:\Users\mohan\OneDrive\Desktop\char_data.csv").astype(float)

**# Split data the X - Our data , and y - the predict label**

X = data.drop('0',axis = 1)

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

```
y = data['0']
train_x,test_x,train_y,test_y=train_test_split(X,y,test_size= 0.2)
train_x=np.reshape(train_x.values,(train_x.shape[0],28,2))
test_x = np.reshape(test_x.values, (test_x.shape[0],28,28))
print("Train data shape: ", train_x.shape)
print("Test data shape: ", test_x.shape)
```

**# Dictionary for getting characters from index values...**

```
word_dict =
{0:'A',1:'B',2:'C',3:'D',4:'E',5:'F',6:'G',7:'H',8:'I',9:'J',10:'K',11:'L',12:'M',13:'N',14:'O',15:'P',16:'Q',17:'R',18:'S',19:'T',20:'U',21:'V',22:'W',23:'X', 24:'Y',25:'Z'}
train_yint = np.int0(y)
count = np.zeros(26, dtype='int')
for i in train_yint:
    count[i] +=1
alphabets = []
for i in word_dict.values():
    alphabets.append(i)
fig, ax = plt.subplots(1,1, figsize=(10,10))
ax.barh(alphabets, count)
plt.xlabel("Number of elements ")
plt.ylabel("Alphabets")
plt.grid()
plt.show()
```

**#Shuffling the data ...**

```
shuff = shuffle(train_x[:100])
fig, ax = plt.subplots(3,3, figsize = (10,10))
axes = ax.flatten()
for i in range(9):
    axes[i].imshow(np.reshape(shuff[i], (28,28)), cmap="Greys")
plt.show()
#Reshaping the training & test dataset so that it can be put in the model...
train_X = train_x.reshape(train_x.shape[0],train_x.shape[1],train_x.shape[2],1)
print("New shape of train data: ", train_X.shape)
test_X = test_x.reshape(test_x.shape[0], test_x.shape[1], test_x.shape[2],1)
print("New shape of train data: ", test_X.shape)
```

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

**# Converting the labels to categorical values...**

```
train_yOHE = to_categorical(train_y, num_classes = 26, dtype='int')
print("New shape of train labels: ", train_yOHE.shape)
test_yOHE = to_categorical(test_y, num_classes = 26, dtype='int')
print("New shape of test labels: ", test_yOHE.shape)
```

**# CNN model...**

```
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(3, 3), activation='relu', input_shape=(28,28,1)))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu', padding = 'same'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu', padding = 'valid'))
model.add(MaxPool2D(pool_size=(2, 2), strides=2))
model.add(Flatten())
model.add(Dense(64,activation ="relu"))
model.add(Dense(128,activation ="relu"))
model.add(Dense(26,activation ="softmax"))
model.compile(optimizer = Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=1, min_lr=0.0001)
early_stop = EarlyStopping(monitor='val_loss', min_delta=0, patience=2, verbose=0, mode='auto')
history = model.fit(train_X, train_yOHE, epochs=1, callbacks=[reduce_lr, early_stop],  validation_data = (test_X,test_yOHE))
model.summary()
model.save(r'model_hand.h5')
```

**# Displaying the accuracies & losses for train & validation set...**

```
print("The validation accuracy is :", history.history['val_accuracy'])
print("The training accuracy is :", history.history['accuracy'])
print("The validation loss is :", history.history['val_loss'])
print("The training loss is :", history.history['loss'])
```

**#Making model predictions...**

```
pred = model.predict(test_X[:9])
print(test_X.shape)
```

**# Displaying some of the test images & their predicted labels...**

```
fig, axes = plt.subplots(3,3, figsize=(8,9))
```

```
axes = axes.flatten()
for i,ax in enumerate(axes):
    img = np.reshape(test_X[i], (28,28))
    ax.imshow(img, cmap="Greys")
    pred = word_dict[np.argmax(test_yOHE[i])]
    ax.set_title("Prediction: "+pred)
    ax.grid()
```

**# Prediction on external image...**

```
from PIL import Image
import cv2
import numpy as np
img = cv2.imread(r'C:\Users\mohan\OneDrive\Desktop\project\handwritten character recognition\images\image001.jpeg')
img_copy = img.copy()
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (400,440))
img_copy = cv2.GaussianBlur(img_copy, (7,7), 0)
img_gray = cv2.cvtColor(img_copy, cv2.COLOR_BGR2GRAY)
_, img_thresh = cv2.threshold(img_gray, 100, 255, cv2.THRESH_BINARY_INV)
img_final = cv2.resize(img_thresh, (28,28))
img_final =np.reshape(img_final, (1,28,28,1))
img_pred = word_dict[np.argmax(model.predict(img_final))]
cv2.putText(img, "csea012 ", (20,25), cv2.FONT_HERSHEY_TRIPLEX, 0.7, color = (0,0,230))
cv2.putText(img, "Prediction: " + img_pred, (20,410), cv2.FONT_HERSHEY_DUPLEX, 1.3, color = (255,0,30))
cv2.imshow(' handwritten character recognition _ _ _ ', img)
while (1):
    k = cv2.waitKey(1) & 0xFF
    if k == 27:
        break
cv2.destroyAllWindows()
```

International Journal for Innovative
Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

## 5. *RESULT*

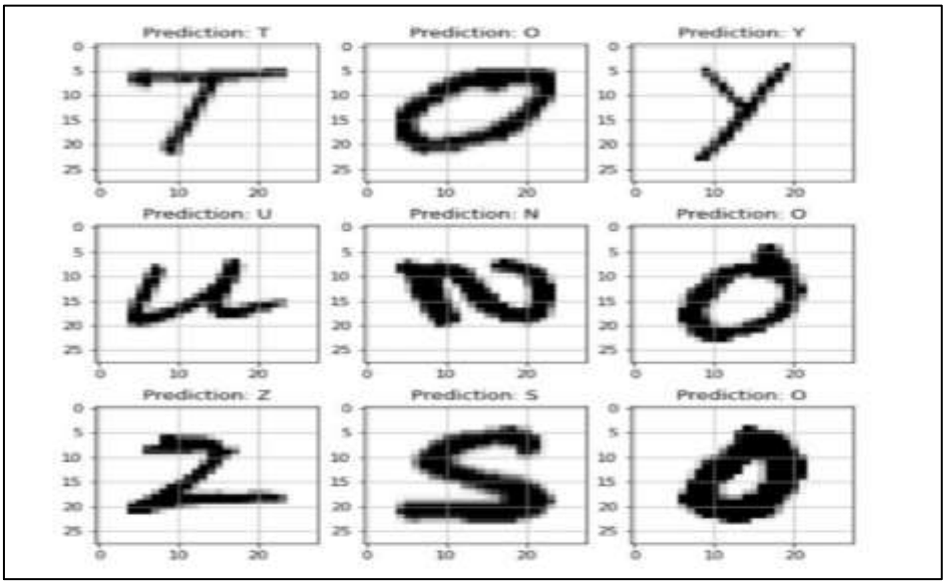



Fig 5a: Test images and their predictions

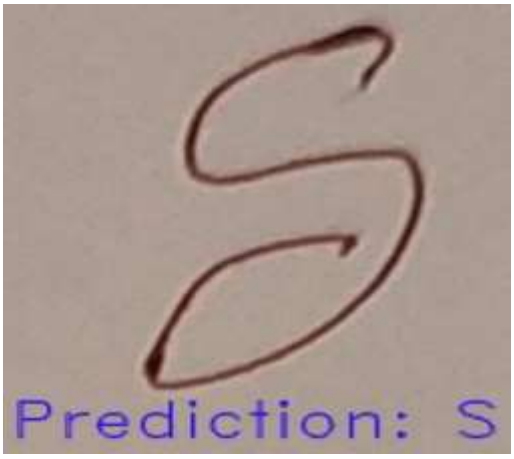



Fig 5b: Character prediction-1

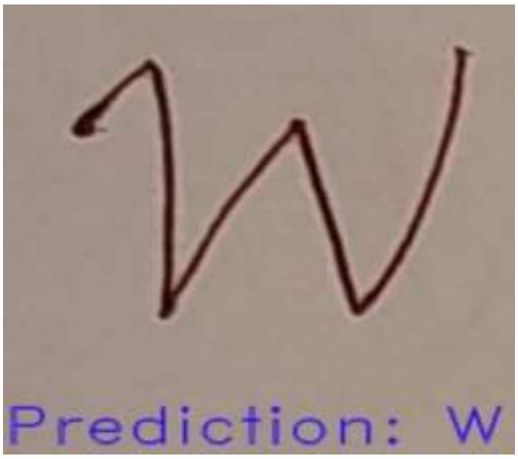



Fig 5c: Character prediction-2

## 6. *CONCLUSION*

The machine learning has become a dominant method in a variety of complex tasks such as image classification and object detection, During development of the project, we learned how the system recognises handwritten characters using Keras, sklearn, and OpenCV, NumPy, Matplotlib, Pyplot. This handwriting recognition system utilized basic image processing algorithms to segment characters from an input image and then we passed each individual character through our trained handwriting recognition model to recognize each character.

This algorithm first it reads the text, binarizes, and extracts features of handwritten character images, and setups the database of character. A recognition model based on machine learning algorithm was setup in matplot which is trained by sample images in the database. After all the training samples have been studied, the learning effects of the model would be tested by test samples. This proposed approach has very low wrong recognition accuracy but has high speed of performance. Performing with proper pre-processing tasks and valid database will leads to the successful detection of English alphabets from the images with better efficiency. Handwriting recognition based on machine learning algorithm provides an effective way to improve the overall recognition rate of data in the field of handwritten letters.

## 7. FUTURE SCOPE

Handwriting recognition can be both offline and online. When text is first written on the paper and then fed into a computer, it is called offline handwriting recognition. But on the other hand, when the computer identifies the handwritten text while writing on the screen of a digital device, it is called online handwriting recognition.

Here are few of the applications and interpretations of handwritten character recognition.

- **To interpret**

Banking system suffers from huge dependencies upon manpower and written documents thus making conventional banking processes would be effective and time-consuming. The current methods for processing transactions made through cheques causes a delay in the processing as the details have to be manually entered. Handwritten Character Recognition finds usage in various fields of data entry and identification purposes this can be achieved through automatic cheque processing.

Application Form Handwriting style of an individual person also varies time to time and is inconsistent, so the application forms filled manually will have different styles and it becomes difficult to detect the character by the person who enters the details in the system. With the help of this system, we will be able to recognise the character by directly scanning the documents and can detect the details of the person.

Old scripts This offline handwritten character manual identification and analysis is a time consuming and lengthy process. During manual recognition process, there might have possibilities of human errors. This human error might change the meaning of existing ancient document content accurate analysis and transliteration/translation is an essential requirement for keeping ancient document alive. In future we are expecting to explore the Back-propagation algorithm to make recognition of characters faster and more efficient and improve the overall performance.

- **Biometrics and forensics**

The handwriting character recognition is also applicable in writer identification, where it is used in biometric and forensic.so one of the applications of handwriting recognition is solving ancient manuscripts disputes, where we will be able to determine the actual writer of the manuscript which can be identified with the help of some of features of writers handwriting.

- **Automatic conversions of prescription to typed form**

As it is one of the challenging tasks to understand the doctors handwriting. This problem can be solved by using online handwriting recognition technique which is used when automatic conversion of prescription to typed form has to employed.

- Not only the application mentioned above are applicable but can also can be used in future for recognizing the alphabets by scanning of the paper documents or images directly through system.

- This project provides the precise alphabet along with the accuracy up to which the character is determined.

- In recent days the advancement in technologies has pushed the limits further for man to get rid of older equipment which posed inconvenience in using.

- The future is completely based on technology where no one will use the paper and pen for writing. In this regard they will use touch pads to write, so the inbuilt software which can automatically detect character.

• Several applications including address recognition in post office, bank cheque processing, document reading require handwriting recognition systems.

## 8. REFERENCES


1. K. Gaurav and Bhatia P. K., “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”, 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.
2. Sandhya Arora, “Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition”, IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.
3. Nafiz Arica, and Fatos T. Yarman-Vural, ―Optical Character Recognition for Cursive Handwriting,‖ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113, June 2002.
4. J.P.Premi a ,R.Madhumithab , N.R.Raajan, “CNN based Digital alphanumeric archaeolinguistics apprehension for ancient script detection”, Turkish Journal of Computer and Mathematics Education Vol.12 No.6 (2021), 5320-5326
5. Chooi Shir Ley1, Aimi Syammi Binti Ab Ghafar1*, “Handwritten Character Recognition Using Convolutional Neural Network”, Progress in Engineering Application and Technology Vol. 2 No. 1 (2021) p. 593-611
6. Monika*1, Monika Ingole2, Khemutai Tighare2, “Handwritten Character Recognition Using Neural Network”, International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 7, Issue 4, July-August-2021
7. S. Anandh Kishan, J. Clinton David, P.Sharon Femi, “Handwritten Character Recognition Using Cnn”, IJRAR September 2018, Volume 5, Issue 3
8. Junlei Song, Qi Liu, Shengnan Tian, Yi Wei, Fang Jin, Wenqui Mo, Kaifeng Dong, “Research on handwritten alphabet recognition system based on extreme learning machine”, Proceedings of the 37th Chinese Control Conference July 25-27, 2018
9. Mr.Narendrasing.B Rajput, Prof.S.M Rajput, Prof.S.M.Badave, “Handwritten Character Recognition - A Review”, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 8, October - 2012
10. Khandokar, Md M Hasan, F Ernawan, Md S Islam, M N Kabir, “Handwritten character recognition using convolutional neural network”, Journal of Physics: Conference Series, ICMSE 2020
11. Thatikonda Somashekar, “A Survey on Handwritten Character Recognition using Deep Learning Technique”, Journal of University of Shanghai for Science and Technology, Volume 23, Issue 6, June – 2021.