Text Identification of handwritten using Deep Learning

Thella.Sunitha #1, Kandepi Bhargavi #2, Parvathareddy Charishma #3, Rachapudi Varun Chowdary #4, Ariveni Devaraj #5, Nasar Babu Kolli #6
#1 Associate Professor, Department of Computer Science and Engineering
#2,3,4,5,6 Student, Department of Computer Science and Engineering
QIS College of Engineering & Technology

Abstract-Handwriting Detection refers to the process by which a computer is able to recognise and make sense of handwritten input from various media, including paper documents, touch screens, photographs, and so on. To recognise handwritten text is an example of pattern recognition. The goal of pattern recognition is to assign information or an object to one of several predetermined classes. or groups. Handwriting recognition systems have historically made use of handcrafted features and extensive training data. It is difficult to train an Optical Character Recognition (OCR) system with these requirements in mind. In the last few years, deep learning-based research in the field of handwriting recognition has led to groundbreaking performance. Still, the availability of massive processing power and the exponential increase in the volume of handwritten data calls for advancements in recognition accuracy and warrants additional study. Convolutional neural networks (CNNs) are the best method for solving handwriting recognition problems because they are able to accurately perceive the structure of handwritten characters/words, which in turn aids in the automatic extraction of distinct features. The purpose of this system is to identify documents in a variety of formats. The evolution of handwriting has led to the appearance of many different types of handwritten characters, such as digits, numerals, cursive script, symbols, and scripts in languages other than English.

Keywords—Deep learning, handwritten, Text identification, Toch screen, CNN, OCR.

I. INTRODUCTION

Specifically, Deep Learning belongs to the broader field of Machine Learning, which in turn belongs to the broader field of Artificial Intelligence. Continuously evaluating data in a predetermined logical framework, deep learning algorithms aim to arrive at the same conclusions as a person would. Deep learning does this with the help of complex algorithmic structures called neural networks. The neural network's architecture mimics that of the human brain. Neural networks may be trained to recognise patterns and sort data into categories, much like our brains do for humans. The field of computer science and artificial intelligence (AI) known as natural language processing (NLP) focuses on making computers capable of comprehending written and spoken language in much the same manner as humans do. Natural language processing (NLP) blends statistical, machine learning, and deep learning models with rule-based modelling of human language as developed in computational linguistics. When combined, these tools provide computers the ability to 'understand' the whole meaning of human language, including the speaker's or writer's purpose and emotion, as conveyed via text or audio data. With the use of natural language processing (NLP), computers can translate across languages, follow spoken directions, and quickly summarise vast amounts of text, often in real time. NLP is likely already in use in our daily lives, whether it be in the shape of a voice-operated GPS system, a digital assistant, speech-to-text dictation software, a chatbot for customer support, or some other useful tool. However, natural language processing (NLP) is now playing an increasingly important role in corporate solutions that aim to improve the efficiency of businesses, boost employee productivity, and simplify crucial business procedures. The two most important processes in an OCR system are feature extraction and feature categorization (through pattern recognition). The branch of optical character recognition (OCR) that focuses on reading handwriting has recently attracted a lot of research interest. It's clear that the world could use a technology like handwriting recognition right now. As a consequence of not having this technology in place, we have been forced to rely on handwritten messages, which is not only slower but also more prone to mistakes. Storage and retrieval of physical files is inefficient. Maintenance of the data's right order requires manual work. There has been massive data loss due to the conventional method of storing data throughout history. Data can now be stored, organised, and accessed with relative ease on machines, thanks to modern technological advancements.

Using Handwritten Text Recognition software makes it simpler to archive and retrieve previously archived material. Data security is improved as a result. Google Lens is an example of software that can recognise handwriting and translate it into text. The goal of our work is to develop a programme that can read handwriting using deep learning techniques. Because of their superior accuracy on such tasks, we are considering using a CNN to solve our problem.

Handwritten A subfield of pattern recognition is text recognition. Pattern recognition is used to place information or an item into a certain category. Handwriting recognition is the process of deciphering text written in a spatial form into a corresponding symbolic form. Characters, letters, or icons, depending on the script, all share a few fundamental geometric forms.

The purpose of handwriting recognition is to accurately recognise input letters or images for further analysis by various automated process systems. In the future, this method will be used to identify documents in various file formats. The evolution of handwriting has led to the appearance of several other types of handwritten characters, such as digits, numerals, cursive script, symbols, and scripts in languages other than English. Recognizing addresses and postcodes on envelopes, deciphering monetary amounts written on bank checks, analysing documents, and verifying signatures are just some of the numerous uses for automated recognition of handwritten text. As a result, a machine's ability to "read" text is crucial for processing documents. Artificial intelligence, or AI, is a subfield of computer science that aims to mimic human intellect so that computers may carry out jobs normally reserved for human beings. AI systems may be taught to do tasks like planning, learning, reasoning, problem solving, and decision making using a computer programming language. Algorithms, including machine learning, deep learning, and rules, are the driving force of artificial intelligence systems. With the use of machine learning algorithms and statistical methods, AI systems may learn from the data they are fed. With the help of machine learning, AI systems may improve their performance on a wide variety of activities without being explicitly taught how to do so by humans.

II. RELATEDWORKS

Using a CNN, a picture may be processed and sorted into many categories (such as "Dog," "Cat," "Tiger," and "Lion"). When given a picture, a computer interprets it as a series of pixels. Depending on the image's pixel dimensions (h = height, w = width, d = dimension), it will interpret these values as h x w x d. For instance, an RGB picture with a 663 array of matrix (3 referring to RGB values) and a grayscale image with a 441 array of matrix (4 referring to the amount of colour values needed to represent a pixel) would both be considered images. The most popular types of layers in CNNs are as follows:

One of the primary components of a convolutional neural network (CNN) is the convolutional layer, a mathematical technique used to combine different types of data. A feature map is generated by applying a convolution filter to the input data.

When processing an image, convolution is the initial layer used to identify and isolate key characteristics.

Two, stride refers to the number of transformations performed on the original picture. When stride = 1, for instance, we shift the filters to operate on a single pixel at a time.

Third, non-linearity: the most common Activation function is For a non-linear operation, you need a Rectified Linear Unit, or ReLU for short. F(x) = max(O, x) is the result.

Typically, we do pooling after a convolution operation to lower the dimensionality, which is the fourth layer of our architecture. As a result, we may lower the total number of parameters, which both speeds up training and lessens the likelihood of overfitting. By sampling each feature map separately, pooling layers down may reduce the height and breadth while preserving the depth. Pooling, unlike convolution, does not need any input parameters. It overlays a sliding window over the input, and then takes the highest value inside the window. This procedure selects the greatest element in the feature map. Since several successes have been made with the MNIST dataset [1,] we have chosen to utilise it. Handwritten text recognition has been feasible even before the advent of deep learning, but previous attempts either had very poor accuracy or relied on a very limited dataset, as noted by Line Eikvil [2]. Examples of where OCR has been put to use are shown and addressed in this article. These include RF reading, vision systems, magnetic stripe reading, bar code reading, and optical mark reading. Classification of the MNIST dataset (a collection of numbers) is a common application of machine learning.

Solutions that Work Best with Convolutional Neural Networks Simard, Steinkraus, and Platt's study on the use of complex neural networks to visual document analysis is an excellent resource for doing so (CNNs). A 2-layer convolutional neural network (CNN) was utilised to feed data into a bidirectional recurrent neural network (RNN) containing Long Short-Term Memory (LSTM) cells [3] to perform word recognition.

We find that the multi-dimensional RNN model of Graves and Schmiduber to be the most effective. [4]

Also, M.J. Castro-"Handwritten Bleda's Text Recognition" work addressed a dataset with slanted words and pre-processing corrections for them. [5] Advances in the Science and Technology of Recognizing Handwritten English While working with the EMNIST dataset, Teddy Surya and Ahmad Fakhrur deploy a Deep Neural Network model consisting of two Encoding layers and one SoftMax layer. The DNNs they used had far higher accuracy than the patterned and feedforward net ANNs that had been suggested before (Artificial Neural Networks). According to studies by ChunPeng Wu and Wei Fan, [6] relaxation convolutional neural networks (R-CNN) and alternatively trained relaxation convolutional neural networks (ATRCNN) may also be used to do handwritten text recognition. Using the Keras[8] library's Convolutional Neural Networks, our model attained an accuracy of over 87% [7].

The CNN model has seen widespread use in recent years for handwritten digit recognition using the MNIST benchmark collection. Handwritten digit recognition has been estimated to be as accurate as 98% or 99% by certain studies [8]. Multiple Convolutional Neural Network models were combined into one for the ensemble model's development. Accuracy of 99.73% was reported in the MNIST digit recognition experiment [9]. The recognition accuracy for the same MNIST dataset was claimed to have increased to 99.77% when the "7-net committee" experiment was expanded to a "35-net committee." For the MNIST digit identification experiment [1], Niu and Suen combined the SVM (support vector machine) capacity of reducing the structural risk with the CNN model capability of extracting the deep features, resulting in an astonishing recognition accuracy of 99.81%. CNN was used to study the bend directional feature maps for in-air handwritten Chinese character recognition [9]. Recently, by constructing several ensembles of deep neural networks, the work of Alvear-Sandoval et al. obtained an error rate of 0.19 percent on the MNIST dataset (DNN). However, after much research, it was shown that the excellent recognition accuracy of MNIST dataset pictures is obtained only by ensemble approaches. Although ensemble approaches aid in boosting classification accuracy, doing so comes at the expense of higher testing complexity and processing cost in practical settings.

The NIST Special Database 19 is the source for the handwritten numbers that make up the IAM dataset. The images are all scaled down to 28x28, and the dataset is organised in a way that is identical to the original. There are 697932 photos of capital and lowercase letters and digits 0-9 in the training set, and 116323 images of these characters in the test set, all of which have been assigned to their respective classes. There is a list inside a list where you may access the test set and the training set. The outer list items each represent a single picture, while the inner list items each include the intensity values (from 0-255) for 784 individual pixels (given that each image has a dimension of 28 by 28). Both the test and training photos include a black backdrop and a white foreground. There is a horizontal flip and a 90 degree anticlockwise rotation applied to both test and training pictures. There are 10 digits from 0 to 9, plus 26 uppercase and lowercase letters and 10 digits, for a total of 62, which are spread between the arrays Y train and Y test. [1]

III. PROPOSED SYSTEM ARCHITECTURE

Figure 1: The basis of Handwritten Text Recognition (HTR) systems is scanned pictures of handwritten text. Here, a Neural Network (NN) trained on word-images from the IAM dataset will be constructed. The reason NN-training is achievable on the CPU is because the input layer (and therefore all the opposing layers) are often kept minimal for word-images (of course, a GPU would be better). The installation of HTR necessitates TF at the very least. Figure 2 is a block diagram depicting the suggested structure for categorising E-style handwritten characters. The suggested recognition method utilises a CNN model with a feature-mapped output layer. In order to categorise the Urdu number, our suggested model will use CNN to sort the input into one of ten categories. The same model will also categorise the provided Urdu character into one of 12 categories (see Figure 2). In the following sections, we'll go into further depth about the individual steps of our suggested approach.

A process model is a hierarchical representation of a set of related processes. In this sense, a process model is a high-level, typological description of a process. There is a one-to-one correspondence between the process model and processes, since processes are instances of the process model. Multiple "instances" (representations) of the same process model exist because it is utilised repeatedly in the production of various software programmes. In contrast to the actual process, which is what occurs, a process model may be used to dictate how things must/should/could be done. A process model is an approximation of the final product of the process.

The specifics of the procedure will be worked out as part of the system's actual development.

With any luck, a process model can help you:

I. Providing a Detailed Description

1. keep a record of the steps taken in a procedure.

Assume the role of a neutral third party whose job it is to evaluate how something has been done and decide what changes need to be made to make it more successful or efficient.

secondly, prescriptive

The first step is to specify what actions are wanted and how they may be taken.

Create norms for the process that, if adhered to, will result in the desired outcome.

They might be very rigid or very loose in their application.

3. Defining

First, you should elaborate on the reasoning for procedures.

Consider the many viable options and make an educated decision based on your findings.

Third, make sure there's a direct connection between operations and the demands that the model must meet.

Fourth, it specifies places in advance from which information may be retrieved for use in reports.

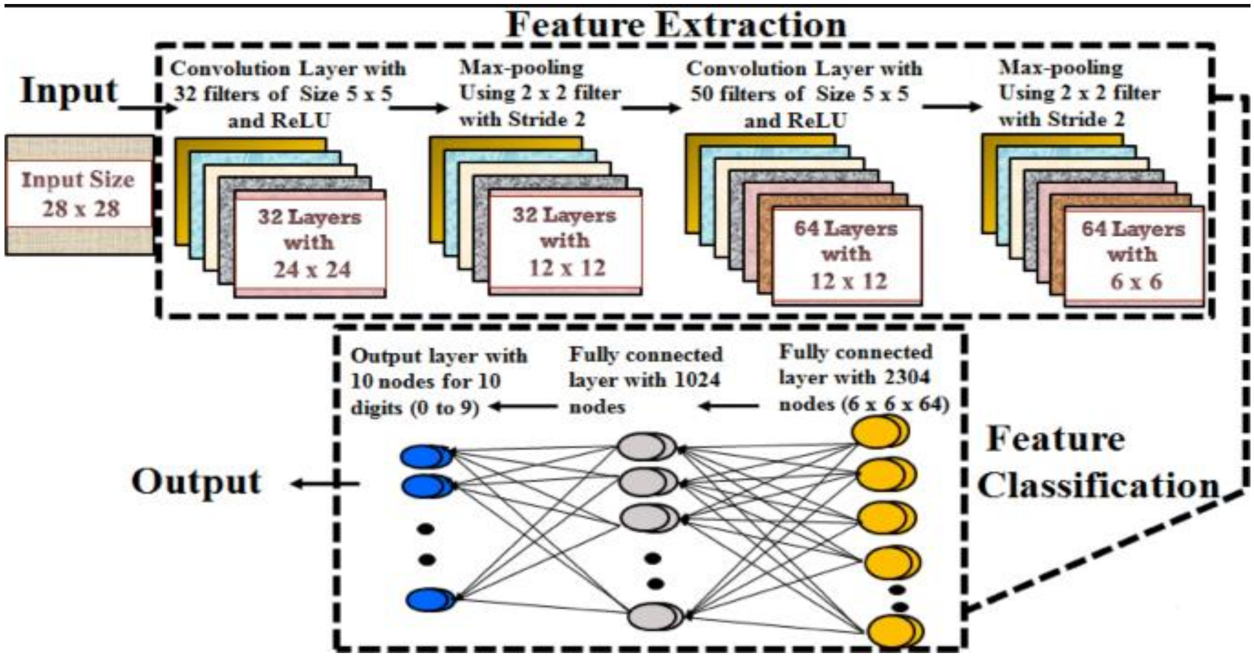

Fig.1 Proposed system architecture

IV. RESULTS AND DISCUSSION

The output screens obtained after running and executing the system are shown in Fig.2. Under the suggested Model Architecture, the input is first preprocessed before being fed into a convolutional neural network (CNN), and last into a recurrent neural network (RNN), which would provide the final result. Control flow modelling is necessary for a broad category of applications that have the following features: This refers to event-driven programmes, which are not data-driven. The programmes that generate control flow data as opposed to output formats like reports or screens. • A programme that operates on data at predetermined intervals We'll import the relevant dataset we've chosen—the MNIST dataset—and utilise a portion of it after partitioning it into a training set and a testing set. • To begin, we plan to use Python's DL libraries to create a model architecture that includes convolutional neural networks (CNNs), recurrent neural networks (RNNs), and any other layers we may find useful. Next, we plan to do any necessary pre-processing steps on the dataset before feeding it into the Training Model; finally, we'll put the Model through its paces of training and train many models by tweaking its parameters to have a variety of models to test.

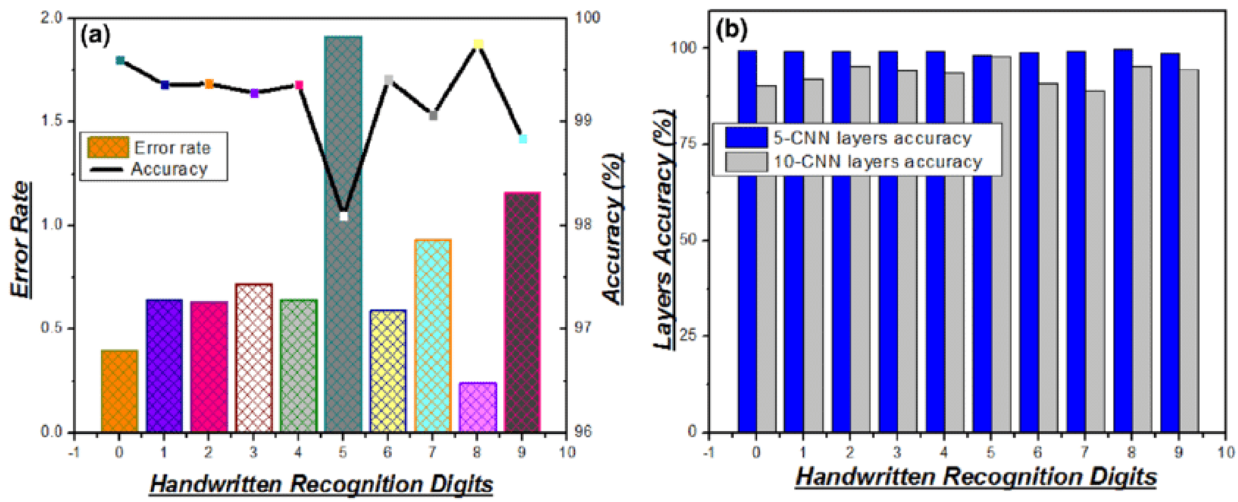

Fig.2 Accuracy graph

V. FUTURE SCOPE AND CONCLUSION

Just scraping the surface of the possibilities behind machine learning is the use of contemporary methods like neural networks to use deep learning to tackle fundamental tasks that are done in a blink of an eye by any person, such as text recognition. The potential uses for this technology are limitless. Once upon a time, OCR served as a biometric device. In order to create a database of well-defined categories, photo sensor technology was employed to collect match points of physical characteristics. Convolution neural networks and other current methods have made it possible to scan and interpret text with a precision never before achieved. By using open-source GPU resources and compute time, we want to obtain the best accuracy feasible for a small, simple model to train. The top-performing CRNN framework will be implemented in our model.

REFERENCES

1. G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "Emnist: an extension of mnist to handwritten letters," arXiv preprint arXiv:1702.05373, 2017.
2. L. Eikvil, "Optical character recognition," citeseer. ist. psu. edu/142042. html, 1993.
3. A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in Advances in neural information processing systems, 2009, pp. 545–552.
4. Fischer, A., Frinken, V., Bunke, H.: Hidden Markov models for off-line cursive handwriting recognition, in C.R. Rao (ed.): Handbook of Statistics 31, 421 – 442, Elsevier, 2013
5. Frinken, V., Bunke, H.: Continuous handwritten script recognition, in Doermann, D., Tombre, K. (eds.): Handbook of Document Image Processing and Recognition, Springer Verlag, 2014

6. S. Günter and H. Bunke. A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In Proc. 16th Int. Conf. on Pattern Recognition, volume 2, pages 332–337. IEEE, 2002.
7. U.-V. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. In Proc. 6th Int. Conf. on Document Analysis and Recognition, pages 159–163.
8. M. Liwicki and H. Bunke, "Iam-ondb - an on-line English sentence database acquired from the handwritten text on a whiteboard," in ICDAR, 2005
9. V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. IEEE, 2014, pp. 285–290.
10. S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid hmm/ann models," IEEE transactions on pattern analysis and machine intelligence, vol. 33, no. 4, pp. 767–779, 2011.
11. T. S. Gunawan, A. F. R. M. Noor, and M. Kartiwi, "Development of english handwritten recognition using deep neural network," 2018.
12. C. Wu, W. Fan, Y. He, J. Sun, and S. Naoi, "Handwritten character recognition by alternately trained relaxation convolutional neural network," 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 291–296, 2014.
13. F. Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.
14. A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in Advances in neural information processing systems, 2009, pp. 545–552.
15. 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.
16. Anita Pal and Davashankar Singh,"Handwritten English Character Recognition Using Neural Network", International Journal of Computer Science and Communication, pp: 141- 144, 2011.
17. Sandhya Arora, "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA, December 2008.
18. Om Prakash Sharma, M. K. Ghose, Krishna Bikram Shah, "An Improved Zone Based Hybrid Feature Extraction Model for Handwritten Alphabets Recognition Using Euler Number", International Journal of Soft Computing and Engineering (ISSN: 2231 - 2307), Vol. 2, Issue 2, pp. 504- 508, May 2012.