IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HAND WRITING RECOGNITION USING DEEP LEARNING ALGORITHM

M. Maha Seetha[1] ,Dr. Jai Ruby MCA, M.Phil , PhD[2]

Department of Computer Applications, Sarah Tucker College, Thirunelveli-7.

Abstract: This project seeks to classify an individual handwritten word so that handwritten text can be translated to a digital form. We used two main approaches to accomplish this task: classifying words directly and character segmentation. For the former, we use Convolution Neural Network (CNN) with various architectures to train a model that can accurately classify words. For the latter, we use Long Short Term Memory networks (LSTM) with convolution to construct bounding boxes for each character. We then pass the segmented characters to a CNN for classification, and then reconstruct each word according to the results of classification and segmentation.Then the text can be converted to three desired language of choice.

Index Terms **- Component,formatting,style,styling,insert.**

I. INTRODUCTION

Over the past decade, electronic document management systems have been of great benefit to society. Software tools such as word processors, computer aided design (CAD) packages, mark-up languages, etc, are extensively used in the generation, storage and retrieval of documents in a format understood by computers. In this format a document can be easily edited, copied on papers, or may be distributed electronically across world-wide networks. Additional processing tools like keyword, pattern search tools, etc, can be used for data extraction and further processing from the large volume of data. On the other hand, when the documents originate on paper, none of the above tasks can be accomplished by the use of computers. It has been estimated that approximately 250 billion USD per year is spent worldwide on keying information from paper documents and this is for keying only 1% of the available documents . Most of this cost is in human labor. When the process of data entry is automated, significant cost savings can be realized. In addition, the percentage of data that is brought on line can be drastically increased. In all of the applications, the major goal is to extract the information contained in the documents and to store them in a digital format. This is the motivation of electronic document analysis systems. One of the most popular natural ways of generating paper documents is handwriting. Most paper documents are handwritten because it is a natural means with convenience and speed. Wide spread acceptance of digital computers seemingly challenges the future of handwriting as the keyboard is becoming the most popular means of generating text. But the 2 convenience of paper and pen is the reason that still handwriting persists in the age of digital computer and hence seen as a more natural way to enter text, etc, into a computer, provided computers can decode the handwriting. On another perspective, the computer generation of the regional language text information is still difficult as the standard keyboards are meant for English script and need to be learned to map the multiple key combinations to produce the regional language scripts which generally have large number of characters. This is another reason why handwriting can be a more user friendly input media to computers. But till today, humans are unable to communicate to a computer using this natural means due to high handwriting variations and noise. Hence automated HCR is still an open problem. Even human beings, who possess the most efficient optical reading device (eyes), have difficulty in recognizing some cursive scripts and have an error rate of about 4% in reading tasks in the absence of context . Hence the recognition system should be insensitive to minor variations and still be able to distinguish different but sometimes very similar looking characters. In this thesis, we are specifically interested in the problem of automation of the handwriting character recognition, particularly for Indian Languages.

Optical Character Recognition

The process of converting the optically acquired images of machine printed or handwritten text (numerals, letters and symbols) into a computer processable format such as ASCII or UNICODE is called Optical character recognition (OCR). OCR generally refers to a technology that reads type written, computer printed or handwritten characters from ordinary documents and translates the images into a form that the computer can understand. The handwritten characters can be acquired using different sensors other than optical. As the basic technology of handwriting recognition is independent of the way it is acquired, in general, the recognition systems handling handwriting are called Handwriting Character Recognition (HCR) systems. As compared to handwritten characters, the typed or printed character recognition is simpler due to well defined shape and size of the characters. Here the complexity is to learn the shapes in different fonts and sizes. As the handwriting suffers huge variations, the HCR is said to be an intelligent OCR to handle high variations in the handwritten text.

The HCR is also referred as intelligent OCR (ICR), Handwritten Character Recognition (HWR), etc. In literature, OCR is commonly used to refer to automation of character recognition irrespective of the acquisition technology, and whether it is for printed text or handwritten text. In this thesis, we use OCR to refer commonly to all the domains of character recognition, PCR for Printed Character Recognition and HCR for Handwriting Character Recognition.

II. LITERATURE SURVEY

In[1] In this context, we present a new optical model architecture (Gated-CNN-BGRU), based on HTR workflow, applied to HDSR. The International Conference on Frontiers of Handwriting Recognition (ICFHR) 2014 competition on HDSR were used as baselines to evaluate the effectiveness of our proposal, whose metrics, datasets and recognition methods were adopted for fair comparison. Furthermore, we also use a private dataset (Brazilian Bank Check - Courtesy Amount Recognition), and 11 different approaches from the state-of-the-art in HDSR, as well as 2 optical models from the state-of-the-art in HTR. Finally, the proposed optical model demonstrated robustness even with low data volume (126 trainable data, for example), surpassing the results of existing methods with an average precision of 96.50%, which is equivalent to an average percentage of improvement of 3.74 points compared to the state-of-the-art in HDSR. In addition, the result stands out in the competition's CVL HDS set, where the proposed optical model achieved a precision of 93.54%, while the best result so far had been from Beijing group (from the competition itself), with 85.29%.

In [2]we propose a novel algorithm, evolved gradient direction optimizer (EVGO), updating the weights of DNNs based on the first-order gradient and a novel hyperplane we introduce. We compare the EVGO algorithm with other gradient-based algorithms, such as gradient descent, RMSProp, Adagrad, momentum, and Adam on the well-known Modified National Institute of Standards and Technology (MNIST) data set for handwritten digit recognition by implementing deep convolutional neural networks. Furthermore, we present empirical evaluations of EVGO on the CIFAR-10 and CIFAR-100 data sets by using the well-known AlexNet and ResNet architectures. Finally, we implement an empirical analysis for EVGO and other algorithms to investigate the behavior of the loss functions. The results show that EVGO outperforms all the algorithms in comparison for all experiments. We conclude that EVGO can be used effectively in the optimization of DNNs, and also, the proposed hyperplane may provide a basis for future optimization algorithms.

In[3] this paper, we develop a new efficient deep unsupervised network to learn invariant image representation from unlabeled visual data. The proposed Deep Convolutional Self-organizing Maps (DCSOM) network comprises a cascade of convolutional SOM layers trained sequentially to represent multiple levels of features. The 2D SOM grid is commonly used for either data visualization or feature extraction. However, this work employs high dimensional map size to create a new deep network. The N-Dimensional SOM (ND-SOM) grid is trained to extract abstract visual features using its classical competitive learning algorithm. The topological order of the features learned from ND-SOM helps to absorb local transformation and deformation variations exhibited in the visual data. The input image is divided into an overlapped local patches where each local patch is represented by the N-coordinates of the winner neuron in the ND-SOM grid. Each dimension of the ND-SOM can be considered as a non-linear principal component and hence it can be exploited to represent the input image using N-Feature Index Image (FII) bank. Multiple convolutional SOM layers can be cascaded to create a deep network structure. The output layer of the DCSOM network computes local histograms of each FII bank in the final convolutional SOM layer. A set of experiments using MNIST handwritten digit database and all its variants are conducted to evaluate the robust representation of the proposed DCSOM network. Experimental results reveal that the performance of DCSOM outperforms state-of-the-art methods for noisy digits and achieve a comparable performance with other complex deep learning architecture for other image variations.

In[4] To this purpose, starting from a database of on-line handwriting samples, we generated for each of them an off-line synthetic color image, where the color of each elementary trait encodes, in the three RGB channels, the dynamic information associated with that trait. To verify the role played by dynamic information, we also generated simple binary images, containing only shape information. Finally, we exploited the ability of Convolutional Neural Network (CNN) to automatically extract features on both color and binary images. The experimental results have confirmed that dynamic information allows a performance improvement with respect to the binary images.

In[5]This paper delineates the state-of-the-art recognition methods along with the user's experience in pen-centric applications for operating with handwritten mathematical expressions. Recognition methods have been categorized into classes, with a description of their merits and limitations. Particular attention is paid to end-to-end approaches based on encoder-decoder architecture and multi-modal input. Evaluation protocols and open benchmark datasets are considered as well as the comparison of the recognition performance, based on open competition results. The use of handwritten math recognition is illustrated by examples of applications for various fields and platforms. A distinctive part of the survey is that we also considered how UI design relies on the use of different recognition approaches, which is aimed at helping potential researchers improve the performance of the introduced approaches toward the best responses in practical applications. Finally, this paper presents the prospective survey of future research directions in handwritten mathematical expression recognition and their applications.

In[6] This article develops a new type of feature for handwriting using Segments Interpolation (SI) to find the best fitting line in each of the windows and build a model for finding the best operating point window size for SI features. The experimental design was done on two subsets of the Institute for Communications Technology/Ecole Nationaled'Ingénieurs de Tunis (IFN/ENIT) database. The first one contains 10 classes (C10), and the second one has 22 classes (C22). The extracted features were trained with Support Vector Machine (SVM) and Extreme Learning Machine (ELM) with different kernels and activation functions. The evaluation metrics from a classification perspective (Accuracy, Precision, Recall and F-measure) were applied. As a result, SI shows significant results with SVM 90.10% accuracy for C10 and 88.53% accuracy for C22.

In[7] In this method, circuit components are first identified using a deep neural network based on YOLO. Then, the connection among these components is recognized using a new simple boundary tracking method. Then, the binary function related to the handwritten circuit is obtained. Finally, the truth table of the logic circuit is generated. We have also created a set of various handwritten logic circuits called JSU-HWLC. The results of the experiments show the proper performance of the proposed method on the collected dataset. The experiments demonstrated that the YOLO algorithm achieved better results than other deep learning methods such as faster R-CNN, Detectron2, and RetinaNet. For this reason, YOLO has been used to identify logic gates in the proposed system.

In[8] This paper proposes a multi-stage cascading system to serve the field of offline Arabic handwriting recognition. The approach starts with applying the Hierarchical Agglomerative Clustering (HAC) technique to split the database into partially inter-related clusters. The inter-relations between the constructed clusters support representing the database as a big search tree model and help to attain a reduced complexity in matching each test image with a cluster. Cluster members are then ranked based on our new proposed ranking algorithm. This ranking algorithm starts with computing Pyramid Histogram of Oriented Gradients (PHoG), and is followed by measuring divergence by Kullback-Leiblermethod. Eventually, the classification process is applied only to the highly ranked matching classes. A comparative study is made to assess the effect of six different deep Convolution Neural Networks (DCNNs) on the final recognition rates of the proposed system. Experiments are done using the IFN/ENIT Arabic database. The proposed clustering and ranking stages lead to using only 11% of the whole database in classifying test images. Accordingly, more reduced computation complexity and more enhanced classification results are achieved compared to recent existing systems.

In[9] The recognition error was a maximum 58.28% lower in SSDCNN than in a model using the eight-directional features alone (5.13% versus 2.14%). Owing to its high accuracy (97.86%), the proposed SSDCNN reduced the recognition error by approximately 18.0% as compared with that of the winning system in the ICDAR 2013 competition. SSDCNN integrated with an adaptation mechanism, called the SSDCNN+Adapt model, and reached a new state-of-the-art (SOTA) standard with an accuracy of 97.94%. The SSDCNN exploits the stroke sequence information to learn high-quality OLHCC representations. Moreover, the learned representation and the classical eight-directional features complement each other within the SSDCNN architecture.

In[10] this paper, a methodology is proposed that covers detection, orientation prediction, and recognition of Urdu ligatures in outdoor images. As a first step, the custom FasterRCNN algorithm has been used in conjunction with well-known CNNs like Squeezenet, Googlenet, Resnet18, and Resnet50 for detection and localization purposes for images of size 320 × 240 pixels. For ligature Orientation prediction, a custom Regression Residual Neural Network (RRNN) is trained/tested on datasets containing randomly oriented ligatures. Recognition of ligatures was done using Two Stream Deep Neural Network (TSDNN). In our experiments, five-set of datasets, containing 4.2K and 51K Urdu-text-embedded synthetic images were generated using the CLE annotation text to evaluate different tasks of detection, orientation prediction, and recognition of ligatures. These synthetic images contain 132, and 1600 unique ligatures corresponding to 4.2K and 51K images respectively, with 32 variations of each ligature (4-backgrounds and font 8-color variations). Also, 1094 real-world images containing more than 12k Urdu characters were used for TSDNN's evaluation. Finally, all four detectors were evaluated and used to compare them for their ability to detect/localize Urdu-text using average-precision (AP). Resnet50 features based FasterRCNN was found to be the winner detector with AP of.98. While Squeezenet, Googlenet, Resnet18 based detectors had testing AP of.65, .88, and .87 respectively. RRNN achieved and accuracy of 79% and 99% for 4k and 51K images respectively. Similarly, for characters classification in ligatures, TSDNN attained a partial sequence recognition rate of 94.90% and 95.20% for 4k and 51K images respectively. Similarly, a partial sequence recognition rate of 76.60% attained for real world-images.

III. METHODOLOGY

The handwritten digit, English letters and Tamil Letters recognition is the ability of computers to recognize human handwritten alphanumeric and tamil letters. It is a hard task for the machine because handwritten alphanumeric and tamil letters are not perfect and can be made with many different flavors. The handwritten alphanumeric and tamil letters recognition is the solution to this problem which uses the image of a alphanumeric and tamil letters and recognizes the character present in the image.

MODULES

- ***MNIST dataset***
- ***Pre-processing***
- ***Segmentation***
- **Train the model**
- **Evaluate the model**
- ***Create GUI to predict character***

MNIST dataset

This is probably one of the most popular datasets among machine learning and deep learning enthusiasts. The MNIST dataset contains 60,000 training images of handwritten digits from zero to nine and 10,000 images for testing. So, the MNIST dataset has 10 different classes. The handwritten digits images are represented as a 28×28 matrix where each cell contains grayscale pixel value.

First, we are going to import all the modules that we are going to need for training our model. The Keras library already contains some datasets and MNIST is one of them. So we can easily import the dataset and start working with it. The mnist.load_data() method returns us the training data, its labels and also the testing data and its labels.

PRE-PROCESSING

The role of the pre-processing step is it performs various tasks on the input image. It basically upgrades the image by making it reasonable for segmentation. The fundamental motivation behind pre-processing is to take off a fascinating example from the background. For the most part, noise filtering, smoothing and standardization are to be done in this stage. The pre-processing additionally characterizes a smaller portrayal of the example. Binarization changes over a gray scale image into a binary image.

A series of operations are performed on the input image (In testing as well as training stage) during the pre-processing. It helps in enhancing the image rendering and makes the image suitable for segmentation. The main objective of pre-processing is to

remove the background noise, enhance the region of interest in image and make a clear difference between foreground and background. In order to achieve these goals: noise filtering, conversion to binary and smoothing operations are performed on the input image. Figure 3 is showing an example of image normalization.

Fig. 3: (a) Tilt input image and (b) Image after preprocessing is performed

The pre-processing also involves a compressed representation of the input image. The edge detection is also performed in order to select the region of interest. The conversion to binary insures a very good difference between foreground and background. Dilation of edges is also performed in the pre-processing step itself.

Segmentation

Segmentation of image is done in the testing stage only. In this, a complete image is decomposed into a sequence of text/subimages of individual text. The segmentation is done on the basis of edge detection and gap between the different characters. After segmentation, the sub-divided parts are labeled and then processed further one by one. This labeling is done in order to find out the number of characters in the entire image. Each sub image is then resized (70×50) and normalized with respect to itself. This helps in extracting the quality features from the image. The scanned image is identified for valid segmentation points by the help of minima or arcs locations in between the characters, which is very easy to find in handwritten texts. The segmentation points are also checked for any error point inclusion by checking all points against the average distance between two segmentation points in complete image (will be shown later).

Create the model

Now we will create our CNN model in Python data science project. A CNN model generally consists of convolutional and pooling layers. It works better for data that are represented as grid structures, this is the reason why CNN works well for image classification problems. The dropout layer is used to deactivate some of the neurons and while training, it reduces offer fitting of the model. We will then compile the model with the Adadelta optimizer.

Train the model

The model.fit() function of Keras will start the training of the model. It takes the training data, validation data, epochs, and batch size.It takes some time to train the model. After training, we save the weights and model definition in the 'mnist.h5' file.

Evaluate the model

We have 10,000 images in our dataset which will be used to evaluate how good our model works. The testing data was not involved in the training of the data therefore, it is new data for our model. The MNIST dataset is well balanced so we can get around 99% accuracy.

Create GUI to predict character

Now for the GUI, we have created a new file in which we build an interactive window to draw digits on canvas and with a button, we can recognize the digit. The Tkinter library comes in the Python standard library. We have created a function predict_digit() that takes the image as input and then uses the trained model to predict the digit.
Then we create the App class which is responsible for building the GUI for our app. We create a canvas where we can draw by capturing the mouse event and with a button, we trigger the predict_digit() function and display the results.

IV. PREPARE YOUR PAPER BEFORE STYLING

In this section, we explain the functional documentation of the project. It considers various blocks of the modules and the associated forms.

We collected data and conducted a quantitative experimental evaluation to assess the performance of the produced system. This section delves into the findings and explains them:

Segmentation: Once the pre-processing of the input images is completed, sub-images of individual digits are formed from the sequence of images. Pre-processed digit images are segmented into a sub-image of individual digits, which are assigned a number to each digit. Each individual digit is resized into pixels. In this step an edge detection technique is being used for segmentation of dataset images.

Feature Extraction: After the completion of pre-processing stage and segmentation stage, the pre-processed images are represented in the form of a matrix which contains pixels of the images that are of very large size. In this way it will be

valuable to represent the digits in the images which contain the necessary information. This activity is called feature extraction. In the feature extraction stage redundancy from the data is removed.

Classification and Recognition: In the classification and recognition step the extracted feature vectors are taken as an individual input to each of the following classifiers. In order to showcase the working system model extracted features are combined and defined using cnn classifier.

Each research work needs some estimation, to measure the accuracy and performance of handwritten digits, MNIST dataset is being used for such reasons. MNIST is the most broadly utilized standard for handwritten digit recognition. MNIST is a huge and a standard database of handwritten digits. MNIST dataset has been commonly used as a standard for testing classification algorithms in handwritten digit recognition frameworks. The initial step to be carried out is to place the dataset, which can be effectively done through the Keras programming interface. The images in the MNIST dataset are available in type of a cluster comprising of 28x28 values constituting to an image along with their labels. This is equivalent if there could be an occurrence of the testing images. The pixels are given as a variety of 784-d pixels and the range extends from 0 to 255 for example 0 implies Black and 255 implies White.

Tensorflow TensorFlow is an amazing information stream in machine learning library made by the Brain Team of Google and made open source in 2015.It is intended to ease the use and broadly relevant to both numeric and neural system issues just as different spaces. Fundamentally, TensorFlow is a lowlevel tool for doing entangled math and it targets specialists who recognize what they're doing to construct exploratory learning structures, to play around with them and to transform them into running programs. For the most, it can be considered as a programming framework in which one can entitle to calculations as graphs. Nodes in the graph speak the math activities, and the edges contain the multi-dimensional information clusters (tensors) related between them.

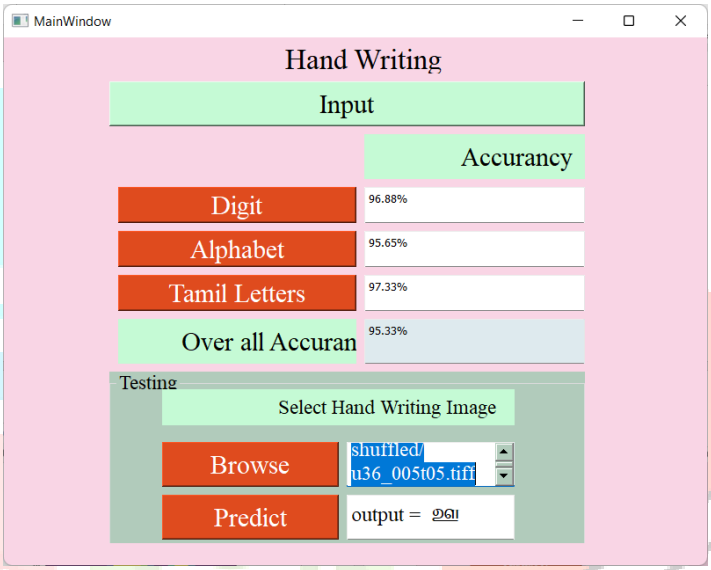

Figure 1 main window of hand writting recognition

The above figure shows the accuracy of the digit recognition, character recognition and tamil letters recognition with its accuracy.

V. CONCLUSION

A proposed handwritten character recognition system has been designed and tested. A comparison with related work has been presented. CNNs have been trained for this purpose with various types of input samples and that's why the developed program has an ability to test and classify the input character into 52 different classes with an accuracy of more than 95%. Two different learning algorithms have been used. Scaled Conjugate Gradient algorithm has been turned out to be better learning algorithm than the Resilient Back-propagation algorithm in terms of accuracy and training time, while using the same configuration. In future work, hybrid feature extraction methods will be developed in order to enhance the accuracy. Also better classification methods will be investigated in order to minimize the miss classified image. Finally, the proposed work will be extended to identify the Arabic language.

REFERENCES

1. F. De Sousa Neto, B. L. D. Bezerra, E. B. Lima and A. H. Toselli, "HDSR-Flor: A Robust End-to-End System to Solve the Handwritten Digit String Recognition Problem in Real Complex Scenarios," in IEEE Access, vol. 8, pp. 208543-208553, 2020, doi: 10.1109/ACCESS.2020.3039003.
2. I. Karabayir, O. Akbilgic and N. Tas, "A Novel Learning Algorithm to Optimize Deep Neural Networks: Evolved Gradient Direction Optimizer (EVGO)," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 2, pp. 685-694, Feb. 2021, doi: 10.1109/TNNLS.2020.2979121.
3. S. Aly and S. Almotairi, "Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition," in IEEE Access, vol. 8, pp. 107035-107045, 2020, doi: 10.1109/ACCESS.2020.3000829.
4. S. Aly and S. Almotairi, "Deep Convolutional Self-Organizing Map Network for Robust Handwritten Digit Recognition," in IEEE Access, vol. 8, pp. 107035-107045, 2020, doi: 10.1109/ACCESS.2020.3000829.
5. D. Zhelezniakov, V. Zaytsev and O. Radyvonenko, "Online Handwritten Mathematical Expression Recognition and Applications: A Survey," in IEEE Access, vol. 9, pp. 38352-38373, 2021, doi: 10.1109/ACCESS.2021.3063413.

6. H. Q. Ghadhban, M. Othman, N. Samsudin, S. Kasim, A. Mohamed and Y. Aljeroudi, "Segments Interpolation Extractor for Finding the Best Fit Line in Arabic Offline Handwriting Recognition Words," in IEEE Access, vol. 9, pp. 73482-73494, 2021, doi: 10.1109/ACCESS.2021.3080325.
7. S. Amraee, M. Chinipardaz, M. Charoosaei and M. A. Mirzaei, "Handwritten Logic Circuits Analysis Using the YOLO Network and a New Boundary Tracking Algorithm," in IEEE Access, vol. 10, pp. 76095-76104, 2022, doi: 10.1109/ACCESS.2022.3192467.
8. T. M. Ghanim, M. I. Khalil and H. M. Abbas, "Comparative Study on Deep Convolution Neural Networks DCNN-Based Offline Arabic Handwriting Recognition," in IEEE Access, vol. 8, pp. 95465-95482, 2020, doi: 10.1109/ACCESS.2020.2994290.
9. X. Liu, B. Hu, Q. Chen, X. Wu and J. You, "Stroke Sequence-Dependent Deep Convolutional Neural Network for Online Handwritten Chinese Character Recognition," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 11, pp. 4637-4648, Nov. 2020, doi: 10.1109/TNNLS.2019.2956965.
10. S. Y. Arafat and M. J. Iqbal, "Urdu-Text Detection and Recognition in Natural Scene Images Using Deep Learning," in IEEE Access, vol. 8, pp. 96787-96803, 2020, doi: 10.1109/ACCESS.2020.2994214.