Handwritten text recognition and conversion into speech

Using Deep Learning and Machine Learning techniques

Tarang Sanyog Gujar, Abdul Gaffar H

Final Year B.Tech Student, Assistant Professor(Senior)

Computer Science and Engineering

Vellore Institute of Technology

***Abstract*— Nowadays everything is becoming computerized .All the time consuming processes like issuing of passport, license, banking process is now done online. All these processes involve huge amount of documentation. This sometimes can create problems like the scanned copies of handwritten documents can be difficult to understand by a computer. The way of writing anything changes from person to person and because of this, the humans also face difficulty in the recognition of different writing styles and variations, may be the patterns shifted, scaled, distorted, with some skewed and even overwritten and therefore it is hard for computers to recognize the writing of a person.So to solve this problem handwriting recognition systems are being developed. Also there is no system designed for blind people which will enable them to interpret handwritten text with precision apart from braille. And it is not feasible to provide braille in all places. Thus such a system or model was important for such people which will read handwritten text and convert into voice. Here introducing four phase: preprocessing, segmentation, feature extraction, classification. CNN Technique will be used for classification and recognition of handwritten text through training and testing.**

***Keywords*—CNN, handwriting, classification, segmentation, feature, preprocessing.**

I. INTRODUCTION

Handwriting Detection system is a system which recognizes handwritten text and successfully scans the text from the image. It uses the dataset which is defined previously and contains the fonts written by various people. The dataset contains around 700k entries for training purpose. The machine learning techniques are being recently developed and thus is discussed in this paper. The main idea of designing such a system was to provide an option to the blind. It is practically not possible to provide braille at every place and especially during our day to day activities. The speech conversion module will thus help them listen to the printed text. Even if the braille language is not available at certain places this system will definitely help them out. Also the main advantage of the system is its compactness. The system must be compact to make it available in the day to day life. The main objective of the project is to optimize the scanning of the text from the image and successfully converting it into speech or voice. Machine Learning algorithms will be used in the process of optimizing the accuracy. The basic idea will be identifying the alphabets initially and the recognizing the alphabets. After the alphabets are recognized the speech module will be implemented. The speech module is completely based on Python. Python being an open source programming language its modules are easily available and implemented. Thus the final output would be the voice generated which in a way speaks out the text identified and recognized.

The OCR(Optical Character Recognition) has been designed before. The accuracy of the OCR was very low. The motive and advantage of using Machine Learning is the high accuracy of machine learning algorithms. The concept used in the OCR involves template matching algorithm which has not proven to be efficient. The template matching algorithm involves finding small parts of the image and then comparing it with an available template. A source image is given and the templates will be made available. The templates will then be slided over the main source image. If the sliding window matches with the template then it is detected and returned. The problem in the OCR is thus that the images having a similar sub pattern will be detected and recognized as same and thus the accuracy is reduced. To overcome this less efficient model, machine learning techniques will be used.

II. LITERATURE SURVEY

Survey of the existing works-

[1] The approach used in this paper is a new approach to off-line handwritten numeral recognition based on structural and statistical features. Five different types of skeleton features: (horizontal, vertical crossings, end, branch, and cross points), number of contours in the image, Width-to-Height ratio, and distribution features are used for the recognition of numerals. We create two vectors Sample Feature Vector (SFV) is a vector which contains Structural and Statistical features of MNIST sample data base of handwritten numerals and Test Feature Vector (TFV) is a vector which contains Structural and Statistical features of MNIST test database of handwritten numerals. The performance of digit recognition system depends mainly on what kind of features are being used. The objective of this paper is to provide efficient and reliable techniques for recognition of handwritten numerals. A Euclidian minimum distance criterion is used to find minimum distances and k-nearest neighbor classifier is used to classify the numerals. MNIST database is used for both training and testing the system. A total 5000 numeral images are tested, and the overall accuracy is found to be 98.42%.

[2] The main objective was Building an effective methodology to detect hand-written characters from images with less error rate which was quite a great task. The aim was to make such an algorithm that will be able to generate error free recognition of hand written text from the given input image which will be a hand written character, and will help in document digitizing. OCR has always been an intensive research topic for more than 4 decades, it is probably one of the most time consuming, as well as labor intensive work of inputting the data through keyboard .This paper discusses about mechanical or electronic conversion of scanned images, text which contain graphics ,image captured by camera ,scanned images and the recognition of images where characters may be broken or corrupted .The optical character recognition is

the desktop based application developed using Python 3.0 and .The should be with > 97.82% accuracy when applied on different data sets, during pre-processing different techniques to remove noise from the background of the image will be used. The labelled data will be converted into gray images and then train the classifier and make generelizations on it using the validation set to reduce any kind of validation error. Finally testing the module using the test_data and see the outcome of the algorithm.

[3] In the paper by Krishna Patel and Vinit Gupta, they say that handwritten digit recognition is an important area of OCR and has continued to persist as a means of communication and recording information in day-to-day life even with the introduction of new technologies. Machine recognition of handwriting has practical significance, as in reading handwritten notes in a PDA, in postal addresses on envelopes, in amounts in bank checks, in handwritten fields in forms, etc. This overview describes the nature of handwritten language, how it is transuded into electronic data, and the basic concepts behind written language recognition algorithms. And digits or characters written which vary from person to person. Both the online case (which pertains to the availability of trajectory data during writing) and the off-line case (which pertains to scanned images) have to be considered. So this paper describes the different surveys that are done in the area of digit recognition with the different techniques means different feature extraction methods with various classifiers.

[4] In the paper written by Ishani Patel, Viraj Jagtap, Ompriya Kale, they have explained that hand written digit recognition is highly nonlinear problem. Recognition of handwritten numerals plays an active role in day to day life now days. Office automation, e-governors and many other areas, reading printed or handwritten documents and convert them to digital media is very crucial and time consuming task. So the system should be designed in such a way that it should be capable of reading handwritten numerals and provide appropriate response as humans do. However, handwritten digits vary from person to person because each one has their own style of writing, which means the same digit or character/word written by different writer will be different even in different languages. This paper presents survey on handwritten digit recognition systems with recent techniques, with three well known classifiers namely MLP, SVM and k-NN used for classification. This paper presents comparative analysis that describes recent methods and helps to find future scope. The final result of each technology is analysed and looked into.

[5] In the paper by Ivor Uhliarik, he has stated that the aim of his work is to review existing methods for the handwritten character recognition problem using machine learning algorithms and implement one of them for a user-friendly Android application. The main tasks the application provides a solution for are handwriting recognition based on touch input, handwriting recognition from live camera frames or a picture file, learning new characters, and learning interactively based on user's feedback. The recognition model that he chose is a multilayer perceptron, a feedforward artificial neural network, especially because of its high performance on nonlinearly separable problems. It has also proved powerful in OCR and ICR systems that could be seen as a further extension of this work. He had evaluated the perceptron's performance and configured its parameters in the GNU Octave programming language, after which he implemented the Android application using the same perceptron architecture, learning parameters and optimization algorithms. The application was then tested on a training set consisting of digits with the ability to learn alphabetical or different characters.

[6] The emnist paper describes the method of developing the emnist dataset. The whole procedure of how the emnist dataset was prepared is described in the paper. The various steps that are followed include Gaussian blur filter which is useful in noise removal. The next step followed is extraction of the actual digit from the picture. The digit is then placed in a square box and cropped accordingly. All these steps are essential for better accuracy. The region of interest is then packed with a border. The dataset is white coloured over black background. The final step is resizing the image into 28x28 dimension. These are the steps followed for every image in the dataset. Using these steps in designing the model is very important and so this paper was analysed.

[7] Ellahyani et al.(2015) have presented a new approach in solving the traffic sign detection and recognition problem. They have divided their approach into three parts. The first part involves image processing where they carry out segmentation based on HSI colour thresholding. The second part carries out the major step in identifying the traffic sign where it stores the traffic sign as a blob object in the database. The last step performs the recognition of the traffic sign. They have not used machine learning algorithms for the classification part, instead they have used invariant geometric moments to effectively classify the shapes. The descriptor in their algorithm has been generated by using a combination of histogram of oriented gradients (HOG) technique and local self-similarity (LSS). For the recognition part they have combined the support vector machines (SVMs) and random forest approach. They have obtained satisfactory results when compared with the already existing implementations.

III. DATA

The main source of my data is the EMNIST dataset. The EMNIST dataset has been created just like the MNIST standard dataset. The images in the dataset are of 28x28 size. Each image has a set of 784 pixels. The data has 62 classes which are not balanced. The classes are divided as 26 for small alphabets, 26 for capital alphabets and 10 for each of the digits. I would like to give a brief about the EMNIST dataset that I will be using in the analysis and prediction. The EMNIST dataset has been defined in a separate research paper which explains how the dataset was created. Various steps are followed in creating the dataset. The steps are as follows. The image of the handwritten character is first taken. The size of that image is not fixed. But the dataset should have images of same size so uniformity should be achieved. First step followed is passing the image through a Gaussian blur filter. This filter will help removing the noise from the image if any. The next step is the ROI extraction. The image does not contain the character fully covered in the given image. So it is important to remove the background of the image and ensure that image is centered into a squared image. The area around the character is thus you can say removed. After this stage the character image is padded with a border of size 2 pixel. Border around all the images is a characteristic of MNIST images. Now the final step is down-sampling of the image to 28x28 pixels. The final output is thus a gray scaled image of size 28x28. This is carried out for each image and thus the dataset is created. I will be using this dataset for my predictive analysis and system building.

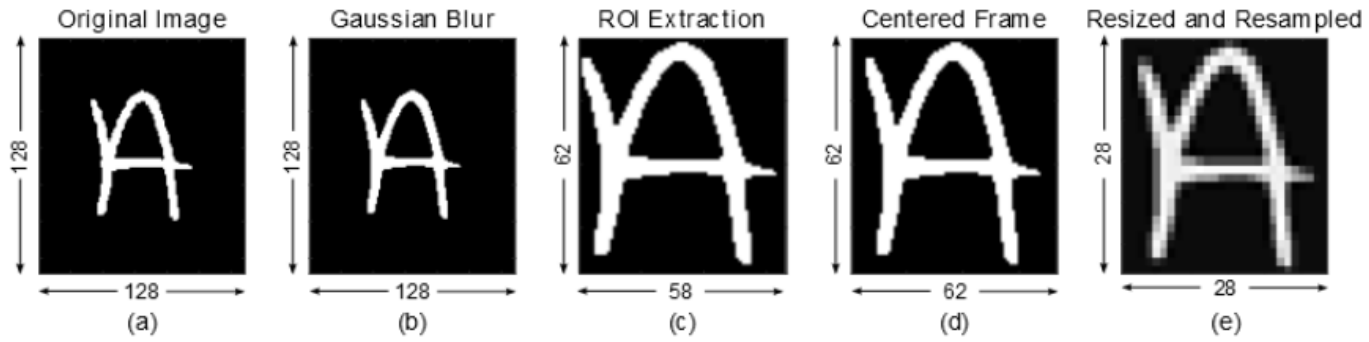


Figure 1: A sample of the EMNIST image generation and creation taken from EMNIST paper

I will be using two Machine Learning models with independent datasets. The second dataset will be a dataset with only two classes. It will be binary dataset with two classes as y=0 and y=1. The class returning 1 will be an image containing some text in it. If the block contains the alphabet completely then it will return 1. If the block contains alphabet partially then it will return 0. This dataset will be prepared by cropping the images of the alphabet from existing handwritten text. Negative class values should also be added in the dataset to increase the efficiency of the model where the dataset will be used. Equal distribution of both positive and negative values is essential for better results. This is the second dataset that will be used in the model.

IV. PROPOSED SYSTEM MODEL

The whole predictor system is built using the CNN classifier. CNN is the most basic of algorithms used in machine learning for image classification. It is basically a non-parametric method used for classification and regression both. The CNN consists of a convolution layer. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. The training examples are vectors in a multidimensional feature space, each with a class label. The number of convolution layers is user defined and more the number of layers more is the accuracy.

Framework and technologies to be-

The Python programming language supports various libraries and frameworks to perform and optimize heavy computations. The computations involved in machine learning are mostly heavy because it involves training and testing of millions of data entries. Thus Python plays an important part here. The proposed framework for the model involves the following components.

Numpy - Numpy is a library which consists of its own data types and is used to perform heavy computation on data. It optimizes the computation by applying efficient algorithms of computation.

Matplotlib - - it is a library which is used to plot graphs, charts, histograms, etc. and visualize data. This is useful in case of outlier detection, analyzing data, etc.

Cv2 - This package contains only the OpenCV core modules without the optional contrib modules. The packages contain pre-compiled OpenCV binary with Python bindings. This enables super-fast (usually < 10 seconds) OpenCV installation for Python. The main use of this module is the image processing part.

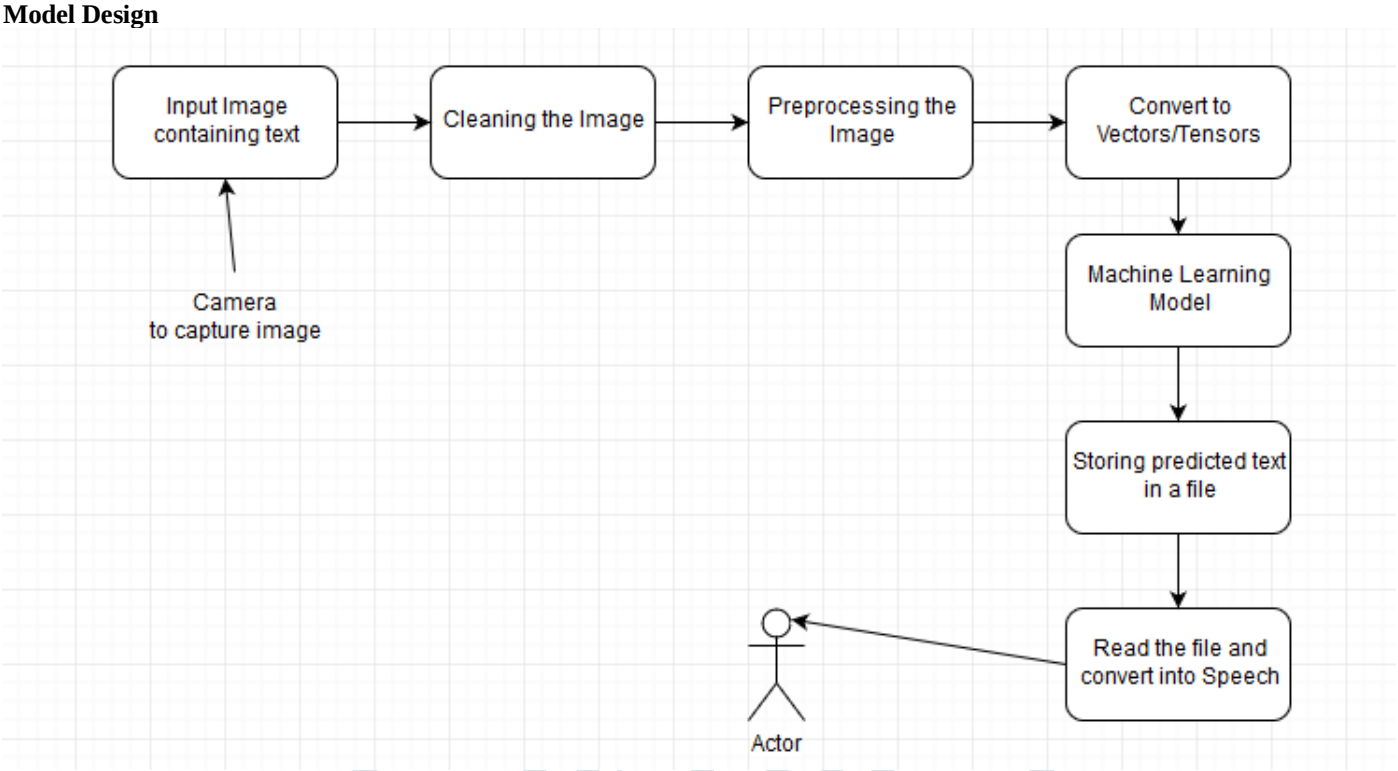
Pyttsx3 - includes drivers for the following text-to-speech synthesizers. This module is used for converting text into speech offline.

Keras: Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

TensorFlow: This is a module developed by google especially for implementing and carrying out machine learning algorithms. It is a framework which sort of provides a platform for implementing the Machine Learning and Deep Learning algorithms.

All these are the available frameworks, libraries and modules which I have incorporated in building the final system.

Model Design



V. Machine Learning Model

The Machine Learning Model is the main gist of the whole system model. I have decided to implement two machine learning models sequentially to build the whole system. The first model will be to recognize the handwritten text and the next model will be to identify the text and classify accordingly.

Part 1- Recognition:

In this sub-model, only two classes are present. It is thus a binary classification. The binary classification involves image classification which means that the image will be analyzed and then return 0 or 1 based on the type. This model will be implemented using the Keras framework. The basic CNN will be implemented in the Keras framework. Activation Function used is ReLu and sigmoidal. Three Convolutional Layers have been added in the neural network designed. Pooling of size 2x2 is done. These are the basics of any CNN which are implemented in Python using the Keras framework. All the windows which are classified as y=1 will be sent forward to the next machine learning model for identification. One epoch is enough for giving accurate output.

Part 2- Identification:

In this sub-model the major and important part of the whole model is designed and implemented. This model will be trained on the EMNIST dataset. The model will have 62 classes to choose from. The data preprocessing has been done before giving the image as input to the first model. This ensures that the image has been converted into MNIST format. This model also uses the CNN using the Keras framework in Python. Four convolutional layers are used in the network. The pooling size id 2x2 for each layer. Activation function used is the ReLu. The output layer of the network has 62 nodes, one for each class. The hidden layers contain as many as 512 nodes to increase the accuracy. I have used 10 epochs for the model to increase the accuracy of the model. The machine learning model will return the predicted class and it is then stored in a text file.

This is how the Machine Learning Model will work briefly.

VI. Training and Testing

One of the important component of any Machine Learning/Deep Learning Model is the training and testing part. The available dataset is not completely used for training but is split into training and testing. The training data is further divide into validation. Thus the whole dataset available is split into train-test-validate. This is a standard practice followed for maintaining and getting high accuracy. All the models are trained with the ADAM optimizer on a cross-entropy loss function. There are over 800k samples available for training and testing. Deep Learning Models usually require high amount of data and thus the datasets used are perfect.

Abbreviations and Acronyms

CNN – Convolution Neural Network
ReLu – Rectified Linear Unit
MNIST – Modified National Institute of Standards and Technology
EMNIST – Extended MNIST

VII. Results and Discussion

The system aims at helping the blind people listen out the written text even if it is handwritten text. This shall prove beneficial for the blind people. Till date there hasn't been any system or model which will translate the handwritten text into sound to avoid the reading part. The system will make life simpler for the blind people. They are completely dependent on braille right now but this system can help them in better understanding of the written text. The system model is built on a convolution network which involves 3 convoluting layers. On increasing the convoluting layers the accuracy becomes better. The feature extraction extracts more features which eventually helps in increasing the accuracy. Right now the trained model is giving an accuracy of 98% which is quite a high number. The accurate predicted text is being converted into speech accurately as well. The model is thus working along with the voice generation.

VIII. Conclusion

The model seems to work efficiently right now on the prepared dataset. An accuracy of 98% on the prepared dataset from the EMNIST dataset is pretty decent. The initial EMNIST dataset is unbalanced and a lot of pre-processing is required. The accuracy improves after the balancing of the dataset. So the pre-processing of the data is an integral part of the model building and deploying. Once the model is trained the predicted output is stored in a text file and then read from the text file to convert into sound. The main reason of storing in a text file is for making sure the data is backed up. Also playing the audio from the text file makes it much easier. The model can further be improved. Right now each individual letter is being recognized. A better model would be identifying the whole text. The main steps involved in that would be to first identify the text region. A machine learning algorithm would be needed for that. Fixing the window size and setting positive and negative classes as text present and text absent respectively will be the next step. Further another machine learning model needs to be trained for identifying each letter from the text detected. Again using the positive as letter present and negative as letter absent. The final stage would then be identifying each letter and then predicting it. This could be included in future scope as an upgraded model for handwritten text identification. Also the selection of a good activation function is an area which is not much explored. The whole model could be deployed on cloud which would result in better time complexity due to the advanced cloud computing technologies available.

IX. Future Scope

In this section I have identified the possible changes that can be incorporated into the existing model to make it better and efficient. They are as follows.

Making a model such that it can scan, detect and recognize a whole word using deep learning. Right now letter to letter detection is happening which can be more time consuming and inefficient. Thus if we are able to carry out and implement an algorithm for words in particular then it would make the whole model more efficient.

Making a user friendly front end for performing the task more interactively and in a better manner. Building the front end will make the whole system more dynamic. Thus building an appropriate front end is definitely one of the future scope of this project.

Amplifying the sound signal generated. The sound signal or wave generated right now is of low sound and thus amplifying the sound for voice purpose will definitely be an added advantage.

Implementing the model on a high speed processor will speed up the implementation time. Deep Learning algorithms involves computations with heavy and huge data. The local machine thus proves to be slow in such cases. Thus higher computing device is essential.

X. Acknowledgement

I would like to thank my college for giving me this opportunity to carry out the research independently and providing me with any assistance throughout the period of completion of this project. The result thus found out after implementing the model makes you feel better. My guide Prof. Abdul Gaffar H, has helped me a lot over the course of this research and thus will like to express my sincere thanks to him.

XI. References

[1] Ivor Uhliarik, "Handwritten Character Recognition Using Machine Learning Methods", 2014.

[2] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik,(2017)." EMNIST- an extension of MNIST to handwritten letters, The MARCS institute for Brain, Behaviour and Development"

[3] Batuhan Balci, Dan Sasadati and Dan Shiferew(2016)," Handwritten Text Recognition using Deep Learning", Stanford University

[4] Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing, 214*, 758-766

[5]Rasika Janrao, Mr D D Dighe(August 2016),"Handwritten English Character Recognition using LVQ and KNN", IJESRT

[6] Norhidayu binti Abdul Hamid, Nilam Nur Binti Amir Sjarif(2016),"Handwritten Recognition using SVM, KNN and Neural Network",arxiv.org

[7] Junho Yim, Jeongwoo Ju, Heechul Jung & JunmoKim, "Image Classification Using Convolutional Neural Networks with multi-stage Feature

[8] D. Cire ̧san, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. Arxiv preprint arXiv:1202.2745, 2012.

[9] Y. Le Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, et al. Hand-written digit recognition with a back-propagation network. In Advances in neural information processing systems, 1990.

[10] P.Y. Simard, D. Steinkraus, and J.C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In Proceedings of the Seventh International Conference on Document Analysis and Recognition, volume 2, pages 958–962, 2003
[11] S.C. Turaga, J.F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H.S. Seung. Convolutional networks can learn to generate affinity graphs for image segmentation.Neural Computation, 22(2):511–538, 2010.
[12] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples:An incremental bayesian approach tested on 101 object categories.Computer Vision and Image Understanding, 106(1):59–70, 2007.
[13] N. Chen and D. Blostein, “A survey of document image classification:problem statement, classifier architecture and performance evaluation,” International Journal Document Analysis Recognition, vol. 10, no. 1,pp. 1–16, 2007.
[14] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems (NIPS), 2012.
[15] Y Lecun, L Bottou, Y Bengio et al., "Gradient-based learning applied to document recognition[J]", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.