iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VI **Month of publication:** June 2023

DOI: https://doi.org/10.22214/ijraset.2023.54317

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VI Jun 2023- Available at www.ijraset.com

Handwritten Text Recognition and Conversion to Speech

Akash Anand[1], Akshay Anand Rastogi[2], Rohit A Chadichal[3], Anshul Surana[4], Dr. Shyamala G[5], Dr. Latha N.R[6]
[1, 2, 3, 4]*Undergraduate Student,* [5, 6]*Assistant Professor, Department of Computer Science Engineering BMS College of Engineering Bangalore, India*

Abstract: ***Handwritten text recognition and conversion to speech is a complex task that involves multiple stages and technologies. The process begins with image processing, where the handwritten text is captured and pre-processed to enhance its quality and remove any noise. The next step is to perform optical character recognition (OCR), which involves recognizing individual characters in the text and converting them into a digital form that can be processed by a computer. Once the text has been digitized, it is processed by natural language processing (NLP) algorithms to identify and extract relevant information, such as dates, names, and numbers. The final step is to convert the digitized text into speech using text-to-speech (TTS) technology. This involves synthesizing a spoken language representation of the text, typically using machine learning algorithms to model the pronunciation and rhythm of human speech. Overall, handwritten text recognition and conversion to speech is a challenging task that requires a combination of image processing, OCR, NLP, and TTS technologies. However, advances in these fields have made it possible to create systems that can accurately recognize and convert handwritten text into speech with high levels of accuracy***

Keywords: Optical character recognition, Text-to-Speech, Machine Learning, Natural Language Processing

I. INTRODUCTION

Handwriting recognition has been one of the most fascinating and grueling exploration areas in the field of image processing and pattern recognition in recent times. It contributes immensely to the advancement of an robotization process and can ameliorate the interface between man and machine in multitudinous operations. Several exploration workshop have been fastening on new ways and styles that would reduce the processing time while furnishing advanced recognition delicacy. Handwriting recognition (HWR), also known as Handwritten Text Recognition (HTR), is the capability of a computer to admit and interpret comprehensible handwritten input from sources similar as paper documents, photos, touch defenses and other bias. The input is generally in the form of an image similar as a picture of handwritten textbook that's fed to a pattern- recognition software, or as real- time recognition using a camera for optic scanning. The image of the written textbook may be tasted "off line " from a piece of paper by optic scanning(optic character recognition) or intelligent word recognition. Alternately, the movements of the pen tip may be tasted "on line ", for illustration by a pen- grounded computer screen face, a generally easier task as there are further suggestions available. A handwriting recognition system handles formatting, performs correct segmentation into characters, and finds the most presumptive words. In general, handwriting recognition is classified into two types as out- line and on- line handwriting recognition styles. In the off- line recognition, the jotting is generally captured optically by a scanner and the completed jotting is available as an image. optic character recognition (OCR) is the most mainstream fashion used for handwriting recognition. This is done by surveying a handwritten document and also converting it into a introductory textbook document. This is done by scanning a handwritten document and then converting it into a basic text document. This also works by taking a picture of a handwritten text. OCR is basically a form of image recognitionthat is meant to recognize handwriting instead of faces or shapes such as landmarks.

II. OBJECTIVES

1) *To Perceive And Transform Manually Written Content Into Computerized Text To Create A Uniform And Consistent Output To Assist The End User*

The process of perceiving and transforming manually written content into computerized text involves converting physical documents or handwritten notes into digital format, such as electronic text documents or digital images. This can be done using various technologies and tools, including scanners, optical character recognition (OCR) software, and transcription services.

©IJRASET: All Rights are Reserved

Once the handwritten content has been digitized, the next step is to create a uniform and consistent output that is easily accessible and usable for the end user. This may involve editing and formatting the text, correcting any errors or inconsistencies, and organizing the content into a logical structure.

2) To Be Able To Successfully Upload A Handwritten Document And To Scan Handwritten Text Into Computerized Text

Prepare the handwritten document: Ensure that the document is clear and legible. If necessary, make any necessary adjustments, such as straightening pages or removing any obstructions that may hinder scanning. you have access to upload handwritten document on a digital file (such as a PDF, JPEG file) on your computer.

Once you have the scanned image of the handwritten document, you can use Optical Character Recognition (OCR) software to convert the image into computerized text. OCR software analyzes the scanned image and attempts to recognize and extract the text from it.

3) To Convert The Covert The Computerized Text To Speech

To convert computerized text into speech, you can use Text-to-Speech (TTS) technology. TTS software processes text input and generates synthesized speech output. Copy the computerized text that you want to convert into speech. Make sure the text is correctly formatted and ready for synthesis. Ensure that any special characters or formatting instructions are removed or adjusted, as they may affect the quality of the speech output.

Once the TTS software has processed the computerized text, it will generate the speech output. Depending on the tool, the output can be played directly on the website or saved as an audio file. Listen to the synthesized speech and ensure it meets your requirements.

4) To Allow The User To Download The Speech In Mp3 Format

To allow the user to download the synthesized speech in MP3 format. Select MP3 as the download format select MP3 as the desired download format. This ensures that the synthesized speech will be saved as an MP3 file.

III. LITERATURE SURVEY

A. Handwritten Form Recognition Using Artificial Neural Network

Due to increased operation of digital technologies in all sectors and in nearly all day to day conditioning to store and pass information, Handwriting character recognition has come a popular subject of exploration. Handwriting remains applicable, but people still want to have Handwriting clones converted into electronic clones that can be communicated and stored electronically. Handwriting character recognition refers to the computer's capability to descry and interpret comprehensible Handwriting input from Handwriting sources similar as touch defenses, photos, paper documents, and other sources. Handwriting characters remain complex since different individualities have different handwriting styles. This paper aims to report the development of a Handwriting character recognition system that will be used to read scholars and lectures Handwriting notes. The development is grounded on an artificial neural network, which is a field of study in artificial intelligence. Different ways and styles are used to develop a Handwriting character recognition system. still, many of them concentrate on neural networks. The use of neural networks for feting Handwriting characters is more effective and robust compared with other computing ways. The paper also outlines the methodology, design, and armature of the Handwriting character recognition system and testing and results of the system development. The end is to demonstrate the effectiveness of neural networks for Handwriting character recognition.

Handwriting integers and character recognitions have come decreasingly important in moment's digitized world due to their practical operations in colorful day to day conditioning.

It can be proven by the fact that in recent times, different recognition systems have been developed or proposed to be used in different fields where high bracket effectiveness is demanded. Systems that are used to fete Handwriting letters, characters, and integers help people to break more complex tasks that else would be time- consuming and expensive.

A good illustration is the use of automatic processing systems used in banks to reuse bank cheques. Without automated bank cheque processing systems, the bank would be needed to employ numerous workers who may not be as effective as the computerized processing system. The handwriting recognition systems can be inspired by natural neural networks, which allow humans and creatures to learn and modelnon-linear and complex connections. That means they can be developed from the artificial neural network

The main ideal of this exploration is to design an expert system for Handwriting character recognition using neural network approach. Other objects include- To address the issue of delicacy in Handwriting character recognition systems by developing a system that will use effective technology for feting Handwriting characters and words from image media. - To probe and demonstrate the utility of neural network technology in development of effective Handwriting character recognition systems. This exploration is aimed to answer the following questions • What are the different ways and styles used in Handwriting character recognition? • How can the performance of Handwriting recognition systems be bettered using artificial neural networks? This paper will be targeting university scholars and preceptors who want to convert their Handwriting notes and papers into electronic format. Despite the increased relinquishment of digital technology in institutions of advanced education, handwriting remains part of scholars' and preceptors' diurnal lives. scholars take Handwriting notes while harkening to their lectures and take notes while reading from different sources. Some also note down their studies, plans, and ideas on their notes. Likewise, speakers have Handwriting notes that they would want to communicate to scholars. Hence, this paper will be targeting scholars and speakers to develop a system that will allow them to convert their Handwriting works into electronic workshop that can be stored and communicated electronically. The central supposition of this paper is that scholars and speakers need to have clones of their workshop that are stored electronically in their particular computers. Further, handwriting with pen and paper can not be entirely replaced by digital technology.

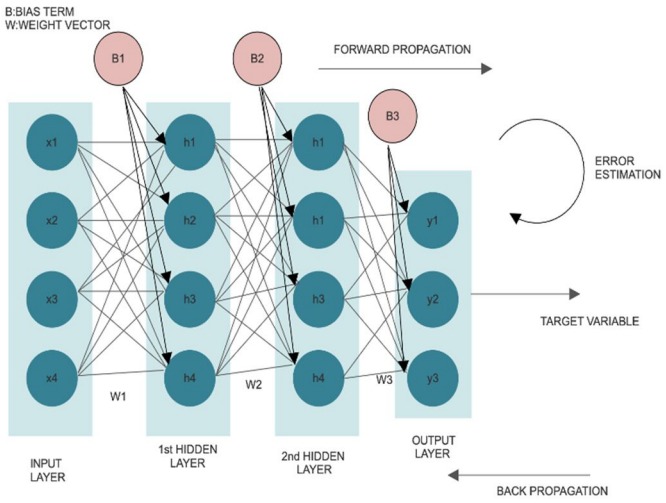


Figure1:Artificial Neural Network architecture which shows the non linear transformation in each hidden layer

B. Handwritten Text Recognition using Deep Learning (CNN,RNN)

In computing, Handwritten Text Recognition refers to the capability and system of a computer to honor and understand comprehendible handwriting acquired from a variety of sources, including prints, scrutinized documents, and other sources. Machine literacy ways like handwritten textbook recognition are used to produce patterns. To achieve pattern realisation, input or objects are organised or categorised into a small number of groups or orders from a vast number of possible options.historically, handwriting identification styles depended on handwrought characteristics and a significant volume of preexisting knowledge. Training an Handwritten Text Recognition system utilising the standard styles and a varied collection of rules and criteria is a delicate job. multitudinous exploration on handwriting recognition have been conducted in recent times, with a special focus on deep literacy styles that have produced advance results. Despite this, the adding quantum of handwritten data and the new strides in calculating power are veritably important need to increase the factual delicacy achieved in duly recognising written textbook, which necessitates farther study. Consonant with the fact that convolutional neural networks (CONVNETs) algorithms are extremely effective at rooting structure from images, they're also extremely able of relating handwritten textual characters words in ways that grease automatic recognition of unique characteristics. As a result, CONVNETs algorithms are the most applicable system for addressing handwritten textbook recognition challenges. A Handwritten Text Recognition system is primarily concerned with feature extraction and feature discrimination/classification (based on patterns). Handwritten Text Recognition is a much-needed technique today. Prior to the effective application of this technology, we depended on handwriting texts, which is prone to mistake. Physical data is notoriously inefficient to store and access.

To keep the data organized properly, manual effort is needed. Throughout history, significant data loss has occurred because of conventional data storage methods. Modern technology enables individuals to store data on computers, which facilitates data storage, organization, and access.It is much simpler to save and retrieve data that was previously stored using Handwritten Text Recognition software. Additionally, it strengthens the data's security. Google Lens is an example of such software for handwritten text recognition. Our project's objective is to develop a deep learning-based application capable of recognizing handwriting. We believe that by addressing our issue via the lens of CONVNET, we may get a higher degree of accuracy. Handwritten Textual Identification is a subset of pattern identification and is denoted by the phrase "textual recognition." In pattern recognition, the goal is to organise or categorise data or objects into one of a wide number of groups or categories, which may then be used to make decisions. A Handwritten Text Recognition system is primarily concerned with point birth and point demarcation/ bracket (grounded on patterns). Handwritten Text Recognition is a important- demanded fashion moment. Prior to the effective operation of this technology, we depended on handwriting textbooks, which is prone to mistake. Physical data is notoriously hamstrung to store and access. To keep the data organized duly, homemade trouble is demanded. Throughout history, significant data loss has passed because of conventional data storehouse styles. ultramodern technology enables individualities to store data on computers, which facilitates data storehouse, association, andaccess.It's important simpler to save and recoup data that was preliminarily stored using Handwritten Text Recognition software. also, it strengthens the data's security. Google Lens is an illustration of similar software for handwritten textbook recognition. Our design's ideal is to develop a deep literacy- grounded operation able of feting handwriting. We believe that by addressing our issue via the lens of CONVNET, we may get a advanced degree of accuracy. Handwritten Textual Identification is a subset of pattern identification and is denoted by the expression" textual recognition." In pattern recognition, the thing is to organise or categorise data or objects into one of a wide number of groups or orders, which may also be used to make opinions. Handwriting As described by the scientific community, textual recognition is the process of converting a spatially pronounced expression/ language to its emblematic fellow. Scripts are made up of a collection of symbols known as characters or letters, which are organised according to a set of abecedarian forms.The ideal of handwriting is to directly fete input letters or images, which are latterly estimated by a variety of automatic process systems. The system used to fete colorful types of jotting is relatively analogous. Handwriting has advanced in sophistication, as shown by the presence of numerous types of handwritten characters, including integers, numbers, cursive script, emblematic expressions, and characters in English and alternate languages. Mechanized identification of handwritten textbook could be utmost beneficial in a variety of operations that bear humongous quantities of handwritten data to be reused, in the likes of addresses and postcode recognition on envelopes, quantum interpretation on banking instruments, analysis of colorful documents, and hand verification. therefore, a machine must be able of reading documents or data to facilitate document processing. CONVNET image groups analyses and categorises an input image (E.g., Alligator, Cat Family, Tiger, Lion). The entering image is interpreted by the computer as a collection of pixels. For illustration, each picture is comprised of a 8x 8x 3 matrix of RGB values (3 denotes RGB values) and a 8x 8x 1 matrix of image having grayscale values, here3 and 1 denote the number of colour values needed to represent each image pixel, independently.

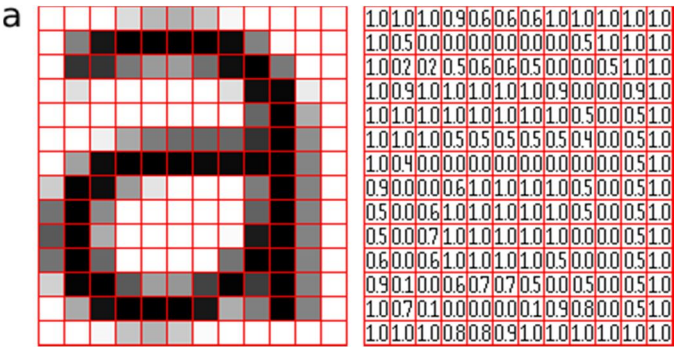


Figure 2: Representation of image as a grid of pixels

C. A Handwriting Recognition Using Eccentricity and Metric Feature Extraction Based on K-Nearest Neighbours

This paper proposes a new approach for handwriting recognition that incorporates eccentricity and metric features into the feature extraction process. Eccentricity is calculated as the distance between the center of the bounding box of the handwritten sample and the farthest point in the sample. Metric features, on the other hand, are calculated using the distances between pairs of points in the handwritten sample. The extracted features are then used to train a k-nearest neighbor classifier, which is used to recognize the handwritten sample.

The performance of the proposed method is evaluated using a standard dataset and is compared with traditional methods of handwriting recognition. The results show that the proposed method outperforms traditional methods in terms of recognition accuracy, demonstrating the effectiveness of incorporating eccentricity and metric features into the feature extraction process. Handwriting recognition is an important area of research in the field of pattern recognitionand has numerous applications in areas such as document analysis, signature verification, and character recognition. The traditional approach to handwriting recognition involves extracting features from the handwritten sample and then matching the sample with the closest reference in the database. However, this method has some limitations and the need for improved feature extraction techniques has been recognized. In this study, the authors propose a new approach for handwriting recognition that incorporates eccentricity and metric features into the feature extraction process. Eccentricity is calculated as the distance between the center of the bounding box of the handwritten sample and the farthest point in the sample. Metric features, on the other hand, are calculated using the distances between pairs of points in the handwritten sample. The extracted features are then used to train a knearest neighbor classifier, which is used to recognize the handwritten sample. The objective of the paper is to propose a new approach for handwriting recognition that incorporates eccentricity and metric features into the feature extraction process and to evaluate its performance. The authors aim to demonstrate the effectiveness of incorporating these features in improving the accuracy of handwriting recognition systems. The authors also aim to compare the performance of the proposed method with traditional methods of handwriting recognition. The study aims to contribute to the field of pattern recognition by presenting a novel method for handwriting recognition that can provide improved results compared to existing methods.

D. Handwritten Pattern Recognition and Natural Language Processing

State of the Art Development of information technologies is growing steadily. With the rearmost software technologies development and operation of the styles of artificial intelligence and machine literacy intelligence embededs in computers, the prospects are that in near future computers will be suitable to break problems hemselves like people do. Artificial intelligenc emulates mortal geste on computers. Rather than executing instructions one by one, as they are programmed, machine literacy employs previous experience/ data that's used in the process of system's training. In this state of the art paper, common styles in AI, similar as machine literacy, pattern recognition and the natural language processing (NLP) are bandied. Also are given standard armature of NLP processing system and the position that's demanded for understanding NLP. Incipiently the statistical NLP processing and multi-word expressions are described. Depending of the compass of operation, there are numerous delineations for the artificial intilligance.According to artificial intelligence maps mortal geste on computers. Anyhow whether the mortal geste is emulated or not, the thing of AI is to produce intelligence. Without any mistrustfulness, the yet to come challenge in AI is to emulate fully or near- impeccably general intelligence. For different purposes, AI combines different styles from the linguistics, statistics and computational intelligence. AI is an interdisciplinary branch of computer wisdom that has connections to other lores similar as neuroscience, gospel, linguistics and psychology. Despite its operation in assiduity, currently-prophetic styles in AI are also generally used in social lores, similar as economics.

There are several areas of specialization of artificial intelligence, similar as

1) games playing, i.e. computers are programmed to oppose gamers
2) Expert systems computers are programmed to make opinions about situations in real- life (Mycin is a typical expert AI system that was developed in the 1970's and it has been used for bacteria identification and ecommendation of specifics and medicines grounded on given symptoms).
3) Natural language computers are programmed to reuse rulings from spoken languages, analysing the morphology, lexicography and indeed the semantics of a whole judgment .
4) Neural networks combination of artificial neurons designed upon the neuron of a mortal being, primarily used for recognition purpose.
5) Robotics computers are programmed to admit girding signals and to generate intelligent responses upon them

Patterns are a form of language.pattern recognition is studied in numerous fields, including psychology, psychiatry, ethnology, cognitive lores, and computer wisdom andtraffic flow.Pattern recognition is a field in machine literacy, but may also relate to pattern recognition(psychology), identification of faces, objects, words, warbles,etc. Since he scopes of machine literacy, knowledge discovery, pattern recognition and data boobytrapping largely lap, they're hard to separate. Most frequently, machine literacy refers to styles grounded on supervised literacy, while unsupervise literacy is primarily explored by data mining and KDD- knowledge discovery.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Unlike machine literacy which is concentrated to maximize the rate of recognition pattern recognition models patterns and discrepancies set up in data. For our exploration, pattern recognition is important as a field in machine literacy. Supervised literacy employs training data set, which is used to identify patterns that match or act formerly annotated discrepancies. Unlike supervised literacy, unsupervised literacy doesn't calculate on training data and it can be applied to descry strange discrepancies in data. By assaying training samples, supervised literacy styles always produce an inferreduction.Applying these functions, an affair for any valid input object can be fluently prognosticated.

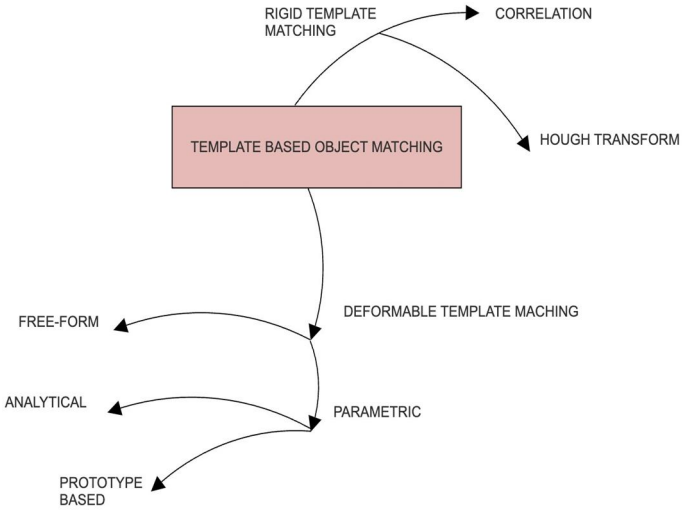


Figure 3: An Overview of template matching techniques

IV. DATASET

A. *IAM Dataset*

The IAM Handwriting Database contains forms of handwritten English textbook which can be used to train and test handwritten text that can recognizers and to perform pen identification and verification trials.

The database was first published in at the ICDAR 1999. Using this database an HMM grounded recognition system for handwritten rulings was developed and published in at the ICPR 2000. The segmentation scheme used in the alternate interpretation of the database is proved in and has been published in the ICPR 2002. The IAM- database as of August 2002 is described in. We use the database considerably in our own exploration.

The database contains forms of unconstrained handwritten textbook, which were scrutinized at a resolution of 300dpi and saved as PNG and JPG images with 256 argentine situations. The figure below provides samples of a complete form, a textbook line and some uprooted words. All forms and also all uprooted textbook lines, words and rulings are available for download as PNG lines, with corresponding XML meta- information included into the image lines. All textbooks in the IAM database are erected using rulings handed by the LOB Corpus The words have been pulled from runners of scrutinized textbook using an automatic segmentation scheme and were vindicated manually. The segmentation scheme has been developed at our institute.

All form, line and word images are handed as PNG lines and the corresponding form marker lines, including segmentation information and variety of estimated parameters from the pre processing way described in are included in the image lines as meta-information in XML format which is described in XML train and XML train format.

V. METHODOLOGY

A. *Training*

We trained each of our models on across- entropy loss function using the ADAM Optimizer followed by softmax activation activation function with the exception of changing the last completely connected affair estate to collude to the number of classes word/ character vocabulary), we kept the identical architectures for the Visual figure Group (VGG) and Residual Networks (RESNET) models. We started from scrape while training the word and character classifier. As it can be seen in the Data section of our paper, we had a large number of unique words in our dataset. still, some word images appeared in our dataset only a multiple times, which made it truly hard for us to train on these images.

This issue, along with the fact that our dataset formerly had a large vocabulary, encouraged us to cut off some of the words in our dataset and not include those words in our training/ substantiation/ test dataset that we were going to use with our models.
We thus limited our data with word images that appeared at least 20 times in our dataset (before splitting into train and substantiation sets). still, still also we had around 4000 unique words with at least 20 circumstances in our dataset. To speed our training process over, we decided to limit our vocabulary size significantly down to 50 words, which still took 5- 6 hours for training and substantiation. Our models and algorithms aren't dependent on hardcoded number of images and would therefore have worked with any number of samples, but we decided to constrict the number of words down for effectiveness purposes

B. Word-Level Classification

For our word- position bracket model, we first constructed a vocabulary grounded on aimlessly opting 50 words with at least 20 circumstances in our dataset. We trained our word classifier with multiple CNN infrastructures VGG- 19, RESNET- 18, and RESNET- 34. The VGG convolutional network armature was one of the first veritably deep neural nets to achieve state- of- the- art results on crucial deep literacy tasks. Defying the standard practice at the time, VGG employed much lower convolutional pollutants (3 X 3) and a smaller number of open field channels in exchange for adding depth in their networks to balance computational costs for the ImageNet 2014 Challenge. By moving from the traditional 3- 7 layers of former CNNs to 16- 19 layers for different duplications of their model, their model not only attained first and the alternate places in the localization and bracket tracks, independently, but also was set up to generalize well to other computer vision tasks still, soon after, Residual Networks (RESNET) outgunned VGG in first place at the ImageNet challenge. Feting that veritably deep literacy networks were delicate to train, in part because of the recession of grade inflow to earlier layers in the network. He et. al developed the notion of the residual subcaste, which rather learned difference functions with respect to the inputs to the layers of the function. thus, a subcaste in a veritably deep residual network would have the option of learning a zero residual (this tendency can be incrementally executed with regularization) and therefore conserving the input and, during backpropagation, conserving the grade to earlier layers. This expression allowed RESNETs with 100 layers to train comparably in terms of effectiveness and parametrization as former deep literacy models and with numerous further layers. RESNETs are as of the date of this paper

C. Character-Level Classification

We also move on to letter segmentation and word reconstruction by classifying each character independently to ameliorate the direct word bracket results. The Tesseract LSTM model with complication was originally downloaded and pre-trained on the forenamed English datasets before it being meliorated onto our dataset. We also had to modify each of our input data so that it included not only input image, but also bounding box markers of each particular character in that image and the closest approximation in the four top left, nethermostop left wing, top right, and top left wing of each personality character, which we uprooted from the XML data. Given the material data, the Tesseract contains programs to automatically construct this dataset. Following that, we created a character vocabulary made up of single- number numbers, uppercase, and lowercase letters. Adam, the optimization fashion we eventually used in our final model was also used, but we used a different kind of loss that was more applicable for the issue CTC loss. In a nutshell, CTC (Connectionist Temporal Bracket) is a fashion/ loss function created by Graves etal. for tutoring intermittent neural networks to marker affair sequences (markers for characters in our illustration) from unsegmented input data the input word picture in our case. RNNs with CTC loss have been shown to be more effective than further conventional styles like CRFs (tentative Random Fields) and HMMs for tasks like labelling speech signal data with word- position abstracts because of their automated literacy process, need for only an input/ affair data representation as opposed to substantial hand- finagled features, and capability to further generally prisoner environment over time and space in their retired state. CTC can separate and label each of the unshaped input data available by assessing network labors as a probability distribution over all possible marker sequences, conditioned on a given input sequence ” created by considering segmentation of the input sequence given a certain minimal member size and starting training of the network from there. We handed these final segmented character prints to our model after honing and completing our segmentation model, as preliminarily described. With the same model designs for bracket and indeed with thepossibility of segmentation error, we discovered that character- position bracket was more successful, as seen in the delicacy graph for charactervs. word- position bracket. These findings corroborated our suppositions that the performance was bettered by the model's original point representation problem, which for characters was significantly lower in compass than for words, and its final labelling problem. Due to a lack of data that was enough for the extent of our issue and excrescencies in the segmentation model, we believed that our model didn't perform any better.

VI. IMPLEMENTATION

The process of converting handwritten text to speech involves several steps, and there is no single "evolution matrix" that describes the entire process. However, here is a general overview of the steps involved:

Optical Character Recognition (OCR): The first step is to use OCR technology to recognize the text in the handwritten document and convert it into digital text. OCR software uses various algorithms to identify the shape and structure of each character and match it to a corresponding digital character.

Natural Language Processing (NLP): Once the text has been digitized, NLP algorithms can be used to analyze the text and identify the meaning of the words and sentences. NLP can help to identify the grammatical structure of the text, as well as any idioms, colloquialisms, or other linguistic features that may affect the way the text is read.

Implementation Using Neural Network, The main models used here are CNN and RNN. We use a NN for our task. It consists of convolutional NN (CNN) layers, recurrent NN (RNN) layers and a final Connectionist Temporal Classification (CTC) layer. We can also view the NN in a more formal way as a function which maps an image (or matrix) M of size W×H to a character sequence (c1, c2, . . .) with a length between 0 and L.

As you can see, the text is recognized on character-level, therefore words or texts not contained in the training data can be recognized too (as long as the individual characters get correctly classified).

The input image is fed into the CNN layers. These layers are trained to extract relevant features from the image. Each layer consists of three operations.

i. First, the convolution operation, which applies a filter kernel of size 5×5 in the first two layers and 3×3 in the last three layers to the input. Then, the non-linear RELU function is applied.

ii. Finally, a pooling layer summarizes image regions and outputs a downsized version of the input.

iii. While the image height is downsized by 2 in each layer, feature maps (channels) are added, so that the output feature map (or sequence) has a size of 32×256

Text-to-Speech (TTS) Conversion: After the text has been analyzed, it can be fed into a TTS system that converts the written text into speech. TTS systems use a combination of recorded voice samples and synthetic voice generation algorithms to create speech that sounds natural and fluent.

Audio Processing: Once the speech has been generated, it can be further processed to improve the quality and clarity of the audio. This may involve noise reduction, equalization, or other audio enhancement techniques.

Here are some snippet code used to training, ,testing and implementation of this project

A. Batch Training

```
def trainBatch(self, batch, batchNum):
""" Feed a batch into the NN to train it """
sparse = self.toSpare (batch.gtTexts)
rate = 0.001 # if you use the pretrained model to continue train
#rate= 0.01 if self.batches Trained < 10 else (# 0.001 if self.batches Trained < 2750 else 0.001) # variable learning_rate is used from trained from scratch evalList = [self.merge, self.optimizer, self.loss]
feedDict= {self.inputImgs: batch.imgs, self.gtTexts: sparse, self.seqLen: [Model.maxText Len] * Model.batchSize, self.learningRate: rate} (loss_summary, _, lossVal) = self.sess.run(evalList, feedDict)
# Tensorboard: Add loss_summary to writer self writer add summary (loss_summary, batchNum)
self.batches Trained += 1
return lossVal
```

B. CNN Model layers

```
def setupCNN (self):
""" Create CNN layers and return output of these layers"""
cnnIn4d = tf.expand_dims (input=self.inputImgs, axis=3)
# First Layer: Conv (5x5) + Pool (2x2) Output size: 400 x 32 x 64 with tf.name_scope ('Conv_Pool_1'): kernel = tf.Variable (tf.truncated_normal( [5, 5, 1, 64], stddev=0.1)) conv = tf.nn.conv2d( cnnIn4d, kernel, padding='SAME', strides=(1, 1, 1, 1))
learelu = tf.nn.leaky_relu (conv, alpha=0.01)
pool = tf.nn.max_pool (lea relu, (1, 2, 2, 1), (1, 2, 2, 1), 'VALID')
```

```
# Second Layer: Conv (5x5) + Pool (1x2) - Output size: 400 x 16 x 128 with tf.name_scope ('Conv_Pool_2'):
kernel = tf.Variable(tf.truncated_normal([5, 5, 64, 128], stddev=0.1))
conv = tf.nn.conv2d(pool, kernel, padding='SAME', strides=(1, 1, 1, 1))
learelu = tf.nn.leaky_relu (conv, alpha=0.01)
pool = tf.nn.max_pool(lea relu, (1, 1, 2, 1), (1, 1, 2, 1), 'VALID')
# Third Layer: Conv (3x3) + Pool (2x2) + Simple Batch Norm Output size: 200 x 8 x 128 with tf.name_scope ('Conv_Pool_BN_3'):
kernel = tf.Variable(tf.truncated_normal(
[3, 3, 128, 128], stddev=0.1))
conv = tf.nn.conv2d(
pool, kernel, padding='SAME', strides=(1, 1, 1, 1))
mean, variance = tf.nn.moments (conv, axes=[0])
batch_norm =btf.nn.batch_normalization (conv, mean, variance, offset=None, scale=None, variance_epsilon=0.001) learelu = tf.nn.leaky_relu (batch_norm, alpha=0.01)
pool = tf.nn.max_pool(lea relu, (1, 2, 2, 1), (1, 2, 2, 1), 'VALID')
```

C. Validation of error rate and accuracy

```
# Validate
Char Error Rate, address Accuracy, word Error Rate = validate
(model,loader)cer_summary=tf.Summary(value=[tf.Summary.Value(tag='charErrorRate', simple_value_charErrorRate)])
# Tensorboard: Add cer_summary to writer model writer add summary (cer_summary, epoch)
address_summary = tf.Summary(value=[tf.Summary.Value(
# Tensorboard: Track charErrorRate tag='addressAccuracy', simple_value-addressAccuracy)]) # Tensorboard: Track addressAccuracy
#Tensorboard:Add address address summary to writer model.writer.add_summary(address_summary, epoch)
wer_summary = tf.Summary(value=[tf.Summary.Value( tag='wordErrorRate', simple_value=wordErrorRate)])
# Tensorboard: Add wer_summary to writer model.writer.add_summary (wer_summary, epoch)
# Tensorboard: Track wordErrorRate
# If best validation accuracy so far, save model parameters
if charErrorRate < bestCharErrorRate:
else:
print('Character error rate improved, save model') bestCharErrorRate = charErrorRate
noImprovementSince = 0
model.save()
open (FilePaths.fnAccuracy, 'w').write(
'Validation character error rate of saved model: %f%% %% (charErrorRate*100.0))
print('Character error rate not improved')
noImprovementSince += 1
# Stop training if no more improvement in the last x epochs
if noImprovementSince >= earlyStopping:
print('No more improvement since %d epochs. Training stopped.' % earlyStopping)
break
```

D. Results

We gave a picture as an input as show in figure 4 and and the model predicts the output as shown in figure 5 using the trained knowledge.



Figure 4: Sample input text

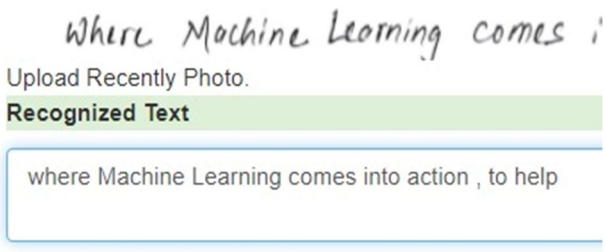


Figure 5: Sample Output for input Image

E. Accuracy Rate

Our model was able to predict with an accuracy of 92 percent.

This accuracy can be increased further in future research work by pre-processing the dataset even more and by adding new parameters to the dataset model and increasing the number of max pooling layer in CNN archetecture. The lower the loss, the better the model (unless the model has overfitted to the training data).

The following table shows the comparison of accuracy with other published models models

PUBLISHED MODEL	AUTHORS	ACCURACY RATE
Hand Witten Text Recognition using Deep Learning	Authors: Makineni Surya Tej, Tungala Veerendra Saradhi, Mallempati Spandana, Vundru Savya	87.3%
HANDWRITTEN TEXT RECOGNITION: with Deep Learning and Android	Shubham Sanjay Mor, Shivam Solanki, Saransh Gupta, Sayam Dhingra, Monika Jain, Rahul Saxena	87.1%
Handwritten Text Recognition using Deep Learning	Batuhan Balci ,Saadati, Dan Shiferaw	81%
Handwritten Text Recognition System Based on Neural Network	Ahmed Mahi Obaid, Hazem M. El Bakry, M.A. Eldosuky, A.I. Shehab	83.22%

Overall, the process of converting handwritten text to speech has evolved significantly over the years, with advances in OCR, NLP, and TTS technologies leading to more accurate and natural-sounding speech synthesis. However, there is still much room for improvement, particularly in the areas of context recognition and emotional tone detection, which could help to make the resulting speech sound more human-like and expressive.

VII. CONCLUSION

We bandied a NN which is suitable to fete textbook in images. The NN consists of 5 CNN and 2 RNN layers and labors a character-probability matrix. This matrix is either used for CTC loss computation or for CTC decoding. The use of CNN and RNN gave emotional delicacy in utmost of the models.

©IJRASET: All Rights are Reserved

As with numerous data wisdom systems, different features could be used and or finagled.

unborn work

1) unborn work on this model can be done to simply ameliorate this model's effectiveness and accuracy. Some styles through which this can be done include

2) Data addition increase dataset- size by applying further(arbitrary) metamorphoses tothe input images.

3) Remove cursive jotting style in the input images.

4) Increase input size if input of NN is large enough, complete textbook- lines can be used Add further CNN layers.

5) Replace LSTM by 2D- LSTM

6) Decoder use token end or word ray hunt decoding to constrain the affair to dictionary words.

7) Text correction if the honored word isn't contained in a wordbook, hunt for the most analogous one.

REFERENCES

[1] N. Darapaneni et al., "Handwritten Form Recognition Using Artificial Neural Network," 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), 2020, pp. 420-424, doi: 10.1109/ICIIS51140.2020.9342638.

[2] Jamshed Memon, Maira Sami, Rizwan Ahmed Khan, Mueen Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review(SLR)", 2020 IEEE Access, Vol.8, 2020, doi:10.1109/ACCESS.2020.3012542

[3] HPattern Recognition and Natural Language Processing: State of the Art, BYMirjana Kocaleva, Done Stojanov, Igor Stojanovik, Zoran Zdravev ,Published On:Elearning Center – University "Goce Delcev", Krste Misirkov bb, Shtip, R.Macedonia Faculty of Computer Science – University "Goce Delcev", Krste Misirkov bb, Shtip, R.Macedonia

[4] A Handwriting Recognition Using Eccentricity and Metric Feature Extraction Based on K-Nearest Neighbors, BY: E. Hari Rachmawanto, G. Rambu Anarqi, D. R. I. Moses Setiadi and C. Atika Sari Published on : International Seminar on Application for Technology of Information and Communication, 2018, pp. 411-416

[5] Handwritten Text Recognition using Deep Learning (CNN,RNN) BY- Rohini G. Khalkar, Adarsh Singh Dikhi, Anirudh Goel3, Manisha Gupta PUBLISHED ON :IARJSET International Advanced Research Journal in Science, Engine Vol. 8, Issue 6, June 2021

[6] España-Boquera, S.; Castro-Bleda, M.J.; Gorbe-Moya, J.; Zamora-Martinez, F. (2011). Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. , 33(4), 0–779. doi:10.1109/tpami.2010.141 Dept.ofCSE,BMSCE2022-23 34

[7] Gyeonghwan Kim1, Venu Govindaraju2, Sargur N. Srihari2 Department of , oul 100- 611, Korea; e-mail: gkim@ccs.sogang.ac.kr 2 CEDAR, State University of New York at Buffalo, 520 Lee Entrance, Amherst, NY 14228–2567, USA

[8] Hull, J.J. (1994). A database for handwritten text recognition research. , 16(5), 0–554. doi:10.1109/34.291440

[9] Read, J.C., S.J. MacFarlane, and C. Casey. Measuring the Usability of Text Input Methods for Children. in HCI2012. 2012. Lille, France: Springer Verlag.

[10] Read, J.C., S.J. MacFarlane, and C. Casey. Designing a Handwriting Recognition Based Writing Environment for Children. in 8th International EARLI SIG Writing Conference. 2015. Staffordshire, England.

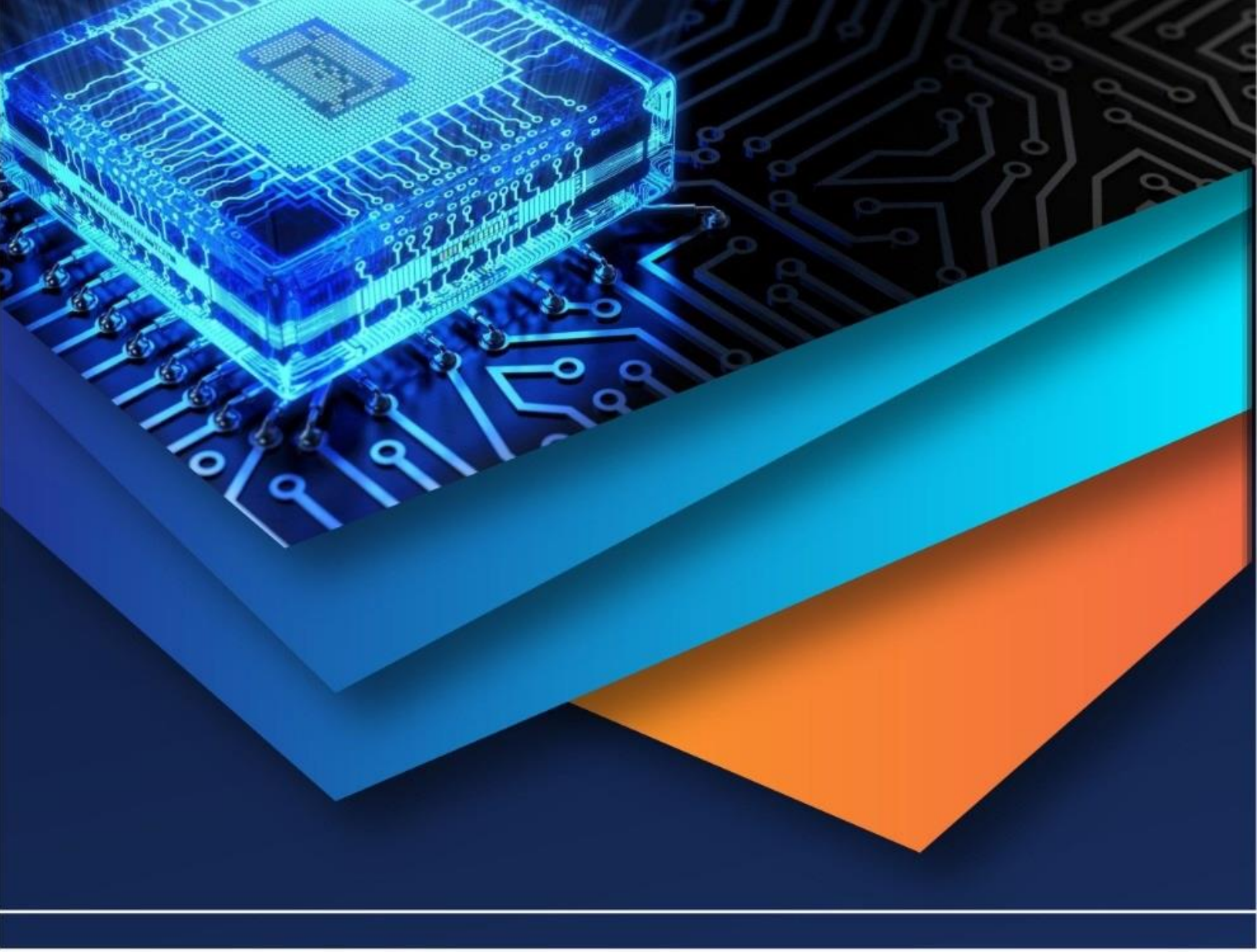
[11] Hanna, L., K. Risden, and K. Alexander, J, Guidelines for usability testing with children. Interactions, 1997.

[12] Schuler, D. and A. Namioka, eds. Participatory Design: Principles and Practices. 1993, Lawrence Erlbaum: Hillsdale, NJ.

[13] Druin, A. Cooperative inquiry: Developing new technologies for children with children. in CHI99. 2014: ACM Press.

[14] Scaife, M., et al. Designing For or Designing With? Informant Design for Interactive Learning Environments. in CHI 97. 2009. Atlanta.

[15] Mankoff, J., G.D. Abowd, and S.E. Hudson, OOPS: a toolkit supporting mediation techniques for resolving ambiguity in recognition-based interfaces. Computers and Graphics, 2008.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)