**Handwritten Text Recognition**

**Group Name: The Pirates**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **First name** | **Last Name** | **Student number** |
| **Muralidhar Reddy** | **Gogireddy** | **C0821216** |
| **Suraj** | **Alamoni** | **C0825558** |
| **Seetha Ram** | **Dasari** | **C0830481** |
| **S.V. Aasheesh** | **Sharma** | **C0825100** |

**Submission date: *23rd April 2022***

Contents

[Abstract 3](#_Toc101628360)

[Introduction 4](#_Toc101628361)

[Literature Survey1 5](#_Toc101628362)

[Literature Survey2 5](#_Toc101628363)

[Literature Survey3 5](#_Toc101628364)

[Datasets 6](#_Toc101628365)

[Methods 7](#_Toc101628366)

[Pre Processing: 7](#_Toc101628367)

[Modeling 8](#_Toc101628368)

[Classification Layers 9](#_Toc101628369)

[Convolution Neural Network (CNN) 9](#_Toc101628370)

[Recurrent Neural network (RNN) 9](#_Toc101628371)

[Connectionist Temporal Classification (CTC) 9](#_Toc101628372)

[Final Model Summary 10](#_Toc101628373)

[Evaluation Metric 10](#_Toc101628374)

[Implementation 11](#_Toc101628375)

[Hardware and Software Components Used 12](#_Toc101628376)

[Results 12](#_Toc101628377)

[Pre-Processing 12](#_Toc101628378)

[Text Recognition 13](#_Toc101628379)

[Output 14](#_Toc101628380)

[Conclusions and Future Work 16](#_Toc101628381)

[References 16](#_Toc101628382)

# **Abstract**

In today’s world, various fields of technology are pushing further the boundaries of human reach due to advanced scientific techniques. A field such as character recognition falls into this category. The digitization of printed documents and the direct preservation of information in digital form has become the norm in this fast-paced world. The goal of this project is to develop an algorithm for recognizing handwritten characters, also known as HTR (Handwritten Text Recognition). By classifying handwritten words, handwritten text can be translated to digital form. Several studies on Handwritten Text Recognition have focused on building advanced models for line recognition on small data sets. In this paper, two main datasets are used to develop a deep learning model with quality, data efficiency, and integration. First, we have the IAM dataset, which is quite famous. One of the biggest challenges is retrieving enough high-quality training data from the data set which has huge numbers of JPG files. IAM dataset contains contributions from 657 authors, totaling 1,539 handwritten pages, containing 115,320 words. On the other hand, we have the GNHK abbreviated as Good Notes Handwritten collection dataset which has a vast array of camera-captured images of English handwritten text sourced from various regions around the globe. In brief, GNHK dataset has 39026 texts extracted from camera captured images. After a thorough data cleaning and preprocessing on both datasets, the resultant data is fed into a deep learning model utilizing Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and the Connectionist Temporal Classification algorithm (CTC) to tackle both classifications and sequence issues with handwriting data. Considering the size of the datasets, multiple methods will be used to train the model on the IAM dataset first and after that, on both datasets together. In this paper, we will devise efficient digital image processing algorithms followed by a comparison processing step and an accuracy estimation is given at the end in percentage terms.

# **Introduction**

Handwritten text recognition (HTR) has gathered extensive research interest due to its potential benefits as a way to simplify digitizing handwritten content. Texts in handwritten form can be found in many formats, including notes, memos, whiteboards, medical records, historic documents, and stylus-entered texts. To offer a predicted solution, machine needs to recognize handwritten text in images. This points to the importance of investigating how to build large-scale handwriting recognition systems that can handle a wide range of scripts and languages. Human operators used to convert various documents into electronic format before the development of handwriting recognition began in the 1950s, so the process was quite time-consuming and prone to errors. By using image preprocessing techniques, automatic text recognition aims to reduce these errors and increase the speed and precision of the recognition process. Over the past few years, handwriting recognition has become one of the most fascinating and challenging areas of research in image processing and pattern recognition. It greatly improves the interface between man and machine in a wide range of applications and contributes greatly to the advancement of automation.

## Literature Survey1

‘Grimsdale’ made an early noteworthy contribution to character recognition research in 1959. Much of the work conducted in the early 1960s was based on the so-called analysis-by-synthesis method, proposed by Eden in 1968. Eden's work was important because he asserted formally that all handwritten characters can be formed by a finite number of schematic features, a point which had previously been implicitly acknowledged. In later attempts to recognize characters, this theory was applied to all methods in syntactic (structural) approaches.

## Literature Survey2

K. Gaurav, Bhatia P. K., this paper describes pre-processing techniques involved in the recognition of characters from different images that are created from simple handwriting to complex images containing colored background and high variation in intensity. In this publication, preprocessing techniques such as skew detection and correction are covered, as well as contrast stretching, noise reduction techniques, normalization, and segmentation. Based on our analysis, we concluded that we cannot completely process an image with a single preprocessing technique. Nevertheless, even after implementing all the above techniques, it is sometimes not possible to get the full accuracy from a preprocessing system. (Purohit, n.d.)

## Literature Survey3

Salvador España-Boquera, as part of this study, different techniques are applied to remove slope and slant from handwritten text and to normalize the size of the text images using supervised learning. Its main goal was to develop a system with high preprocessing and recognition accuracy. Many techniques are applied to analyze an image file's shape and distinguish its characteristics. (Purohit, n.d.)

## Datasets

This paper discusses two main data sets that are used in Deep Learning models, those being IAM dataset and GNHK dataset. The IAM Handwriting Dataset contains passages written by several authors in handwriting. This data is typically used to classify writers based on their writing styles. Such a problem may be solved by extracting features such as spacing between letters, curvatures, etc. and feeding them into Support Vector Machines. To train the system, we don't need the full IAM Handwriting Dataset, but we use words data file instead and feed it to a deep learning model which uses TensorFlow and keras. In this dataset, images of handwritten sentences are provided with a dash-separated file name. First, we have the test code, then we have the author ID, third the passage ID, and finally we have the sentence ID. In total, there are 115320 labeled words and 13353 text lines in the words data file. (IAM On-Line Handwriting Database, n.d.)

In addition, we present the Good Notes Handwriting Kollection (GNHK) dataset, which includes camera-captured images of English handwritten text collected from around the world. This dataset is based on scene text datasets, allowing researchers to explore new methods for localization and text recognition. The dataset contains colored image files having different dimensions, with sentences and lines. This dataset consists of 39026 words and 9363 lines of text. GNHK dataset, after undergoing all the preprocessing techniques, is combined with the IAM dataset and provided to a deep learning model using RNN, CNN, and CTC techniques, which provides optimum results during classification. (Lee, n.d.)

# **Methods**

## Pre-Processing:

Since the length of words varies, the image dimensions differed across both datasets. To give these images to the Deep Neural Network, we had to ensure that the input image size and the output character length were same for all the images and labels. To make the images into the same tensor shapes for the input layer, we have resized them by fixing their aspect ratio (adding necessary padding on the top, right, bottom, and left). The optimal width and height for the image resizing were 200 pixels by width and 32 pixels by height. The preprocessed images are shown in Fig.1.

Company name

Description automatically generated

Fig.1- Pre-Processed Images

First, we had to convert the characters into numeric values for the labels. We have used TensorFlow's StringLookup method to vectorize our target labels. To ensure that the target labels were also of the same length, we padded zeros on the right side of the vectorized array to match the size of the maximum word length.

For the GNHK Dataset, the images were not rectangular. To ensure all the images are rectangular, we have used the bounding box technique of Open CV library to get the images in a rectangle format.

## Modeling

Once our data is Pre-Processed, it is passed to the Model for trapower and block diagram Fig. 2 depicts each step of the modeling process.

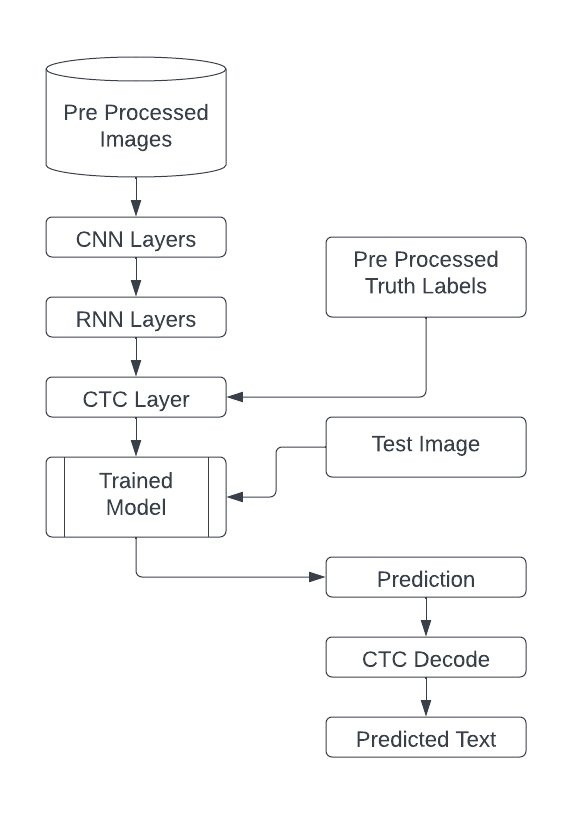


Fig. 2- Block Diagram of Model

## Classification Layers

Convolution Neural Network (CNN)

Whenever we have images as inputs to a neural network, we think of Convolution Neural Network. It mainly has three properties, feature extraction being the most important; by applying various kernel filters across the image, it can capture different set of features and also the correlation between adjacent pixels. It also performs dimension reduction with the help of pooling technique and helps capture non-linear relationships using the activation function such as ReLU at the end.

In our Model, we have used 2 CNN Layers with a kernel size of 3x3, with first layer having 32 channels and the second layer having 64 channels. Relu was used as an activation function in both the layers, and MaxPooling was done after each layer.

Recurrent Neural network (RNN)

Widely used in text and speech recognition tasks to capture the temporal dynamic behavior in sequence data. In our Model, we have used Bi directional Long short-term memory (BiLSTM). It reads the input from both forward and backward directions. We have used 2 BiLSTM layers with 128 units in the first layer and 64 units in the second layer. We also added the dropout layer between CNN layers and RNN layers to reduce the chances of overfitting.

Connectionist Temporal Classification (CTC)

CTC is a scoring function to train RNN layers. It is specially used in handwritten text recognition tasks and speech recognition. As users' handwriting varies, the number of timestamps for the same word can differ for different users. To encounter this problem of having repeating characters, we use CTC loss with a break character ('-').

## Final Model Summary



## Evaluation Metric

We have used Character Error Rate (CER) as an evaluation metric to compare the results of our models. This is calculated using Levenshtein distance between the predicted label and the truth label. For example the distance between “hello” and “hallu” is 2 as it needs two character edits to transform one into the other string. The CER would be 2/6 = 0.333. Which means that for every 3 charters we need to edit one character to make both the strings equal.

## Implementation

Initially, we trained the Model using IAM Dataset word images and got an accuracy measure of**80%**for 50 epochs. We later applied the same model structure to train on the GNHK Dataset. However, even after training the Model for 70 epochs the accuracy was not improving (remained at 55%). We did trial and error experimentation to increase our Model's performance on GNHK Data set.

Experimentations:

* Increased number of Convolution layers from 2 to 5.
* Changed the parameters of Convolution layers such as number of filter channels, kernel size and stride.
* Increased the LSTM layers from 2 to 3.
* Changed unit sizes of LSTM layers.
* Increased the learning rate and used RMSprop optimizer for faster convergence of the loss function.

The Final Model was trained using both datasets, and we have achieved an accuracy of 65%. Even though we initially thought that increasing the training data would improve the Model's performance, this did stand true in our case. The Model's performance depends on various parameters such as the type of input data, the complexity of the Model, the loss function, etc.

## Hardware and Software Components Used

Google's TensorFlow was the main library that we used to train our Model. We have also used other support libraries such as Open CV, Numpy, Matplotlib, and glob, which helped in preprocessing steps, visualization, and working with file reading.

We have used Google Drive as our primary source of data storage. Initially, we used our own machines to train the Model. Understanding the limitations of our devices, we decided to go with Google Colab, which provides access to GPU and TPU (Tensor Processing Unit) for faster training. Even the free tier Colab had its limitations, so we used the Colab Pro Plus tier to run our models and try different experiments on the Model.

# **Results**

Our handwritten recognition model is built with only one sole intention in mind it should recognize text from the source and depict the text digitally right from the processing and training stage.

Pre-Processing  
 While pre-processing our word Images, we have followed several steps to process the data. Those Include the first being extracting all the words from their respective folders and naming them exclusively according to their image path, the second being cropping all the images by their dimensions and the third being adding padding for them to make all the cropped images homogenized so that the model performance is not hindered, and these steps are done for all the word images in both IAM and GNHK datasets.

Training and Validation  
 We are following the same approach for the datasets; we have divided each dataset into three parts where 90 percent of the data is for training, 5 percent of the data is for validation, and finally, the remaining 5 percent of the data is for testing our model.

## Text Recognition

At first, we have implemented our model on both the datasets exclusively and came to a conclusion that our model's performance is good and model's performance is not as good for the GNHK dataset images we came to a conclusion that because the data present in in IAM dataset was simple as compared to GNHK dataset as it has contained only words with black and white images with no noise in them, but where as in GNHK dataset the word images present in the are having different coloured, sized and styled texts and on top of that they are containing additional labels and other printed information which are irrelevant to text in some instances, presence of such noises have affected the model's performance on GNHK dataset, On IAM dataset we have run the model till 50 epochs and where on the other hand we have run our model till and saw that both train and test loss were converging with each other and we have stopped and where as for GNHK dataset we have ran our model till 75 epochs that observed that we have our desired loss at 75th epoch for train data but there is a considerable distance between the test loss and train loss.

A picture containing shape

Description automatically generatedChart

Description automatically generated with medium confidence  
 Fig. 3- Loss curve for IAM dataset. Fig. 4- Loss curve for GNHK dataset.

Since there is this discrepancy between the two datasets before merging them, we have tuned our model by employing hyperparameter tuning techniques and added and deleted a few layers in our model on a trail and error basis, hoping that it would improve our model's performance. Still, it didn't, and finally, we merged both the datasets as one and subjected this combined dataset to our model and ran it for 50 epochs. We have seen the loss curves of both train and test data of the merged dataset and observed that the loss values of Train data have reduced as there is an increase in epochs, and there is some consistent distance between train and test data's loss value.

A picture containing graphical user interface

Description automatically generated

Fig. 5- Loss Curve of merged dataset.

# Output

The model takes images of the containing the texts to depict what is written in them, here is in the image below there is prediction of texts according to to their input images after passed them through the model.

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 6- Predicted text for the respective word images

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Dataset** | **Method** | **Accuracy** |
| [ (J. Sueiras, 2018)] | IAM, RIMES | CNN, LSTM | 95% |
| [ (R.Vaidya, 2018)] | NIST | Deep neural networks, OpenCV, Neural Network is trained using Tensorflow | 94% |
| Our Model | IAM and Custom Handwritten | CNN, RNN, CTC | 80% (IAM DS)  65% (Merged DS) |

Table.1- Comparison the results of our model with the likes of other Depp learning models.

Our proposed model is giving us accuracy getting us an accuracy score of 80% foe IAM dataset alone and for whereas on the combined dataset the accuracy stands at 65% with the error rates of 20% and 35% on the IAM and Merged datasets respectively and the comparison of our model and the other models which were published in some reputed journals is shown above in a tabular format.

# **Conclusions and Future Work**

We have tried improving the model's performance on actual-world data by training the model on two datasets IAM On-Line Handwriting Database (IAM-OnDB) and GoodNotes Handwriting Kollection (GNHK). We achieved an accuracy of 80 percent for 50 epochs for the IAM dataset, and an accuracy of 65 percent for the combined dataset IAM and GNHK. We have used Convolution Neural Network (CNN), Recurrent Neural Network (RNN), and Connectionist Temporal Classification (CTC) in our model. Some of the team's impediments were related to the computational power, and researching the solution and resolving those impediments had cost us some time. From this project, our team has understood the importance of the Software Development Lifecycle, Methodologies and how vital each phase in it is. We also learned how crucial Team management is to make any project a success and how one should proactively look out for the challenges that could hamper the deliverables deadline and think about their solutions and mitigate their impact.

In the future we plan to improve on the model to take paragraphs as inputs, and the word segmentation from the paragraph image. We plan to deploy the model to be accessed through web using REST API or UI application as well.

# **References**

[1] IAM On-Line Handwriting Database (n.d.). In Computer Vision and Artificial Intelligence. Retrieved from [https://fki.tic.heia-fr.ch/databases/iam-on-line-handwriting-database#:~:text=The%20IAM%20On%2DLine%20Handwriting,03%5D%20at%20the%20ICDAR%202005](https://fki.tic.heia-fr.ch/databases/iam-on-line-handwriting-database#:~:text=The%20IAM%20On%2DLine%20Handwriting,03%5D%20at%20the%20ICDAR%202005.)

[2] IAM Handwriting Top50 (n.d.). In Kaggle. Retrieved from <https://www.kaggle.com/datasets/tejasreddy/iam-handwriting-top50>

[3] Lee, A. W. C., Chung, J., & Lee, M. (n.d.). GNHK: A dataset for English handwriting in the wild. GoodNotes. Retrieved April 22, 2022, from https://webflow.goodnotes.com/gnhk

[4] Baek, Y., Nam, D., Park, S., Lee, J., Shin, S., Baek, J., Lee, C. Y., & Lee, H. (2020, June 11). Cleval: Character-level evaluation for text detection and recognition tasks. arXiv.org. Retrieved April 22, 2022, from https://arxiv.org/abs/2006.06244

[5] Graves, A., & Schmidhuber, J. (1970, January 1). Offline handwriting recognition with multidimensional recurrent neural networks. Advances in Neural Information Processing Systems. Retrieved April 22, 2022, from https://papers.nips.cc/paper/2008/hash/66368270ffd51418ec58bd793f2d9b1b-Abstract.html

[6] Purohit, A., & Chauhan, S. S. (n.d.). A literature survey on handwritten character ... - ijcsit.com. Retrieved April 23, 2022, from http://ijcsit.com/docs/Volume%207/vol7issue1/ijcsit2016070101.pdf

[7] J. Sueiras, V. Ruiz, A. Sanchez and J.F. Velez, “Offline Continuous Handwriting Recognition using Sequence to Sequence Neural Networks”, Neurocomputing, Vol. 289, pp. 119-128, 2018.

[8] R.Vaidya, D.Trivedi, S.Satra and P. M. Pimpale, "Handwritten Character Recognition Using Deep-Learning," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).