Plagiarism Checker X Originality Report

Similarity Found: 13%

Date: Sunday, May 08, 2022
Statistics: 497 words Plagiarized / 3947 Total words
Remarks: Low Plagiarism Detected - Your Document needs Optional Improvement.

Abstract In the current world of automation everything is getting atomized as reducing manual labor is the key to efficiency. This paper focuses on Off-line Handwriting Recognition. Handwriting Recognition is the process of extracting text from handwritten scripts, this is also known as Offline Handwriting Recognition.

The purpose of this paper is to attempt to improve the accuracy and efficiency of the system using Neural Networks and datasets used for training the model, as well as detecting and identifying the characters and exporting them in a text format. The proposed system can be used to recognize handwritten characters and convert them into text from the scanned image of a page.

Contents

. . . . 34 6.8 Text to Handwriting Example 2 . . . . 35 ii List of Abbreviations CNN: Convolutional Neural Network RNN: Recurrent Neural Network CTC: Connectionist Temporal Classification OCR: Optical Character Recognition LSTM: Long Short Term Memory iii Chapter 1 Introduction Handwriting Recognition is a subject which exists for a long period of time but the accu- racy of the process can be improved by the usage of more refined technology and better techniques which reduce the error rates and the difference between the actual text and the detected text of the system. Handwriting Recognition is the process of extracting text from handwritten scripts, this is also known as Offline Handwriting Recognition.In Offline Hand- writing Recognition the image of a script is captured using a scanner and sent forward for further processing.

This is a step into the world of automation and can help us improve efficiency in this field. Handwriting Recognition works on similar techniques of text detec- tion and recognition. Text detection includes various techniques to extract text from an image such as Texture Detection method and Connected Component method.

In texture detection the text is extracted which is detected by texture difference from the background and text is registered by further processing of the information. In the Connected Component method text is extracted using the relation between pixel connectivity and pixel intensity to detect text patterns and checks are made to clear non character elements.

The next step in extracting text from an image is Text Recognition which is done by OCR(Optical Character Recognition) which depends on identifying the character from the detected text which can be done by training the model using dataset which gives the model the basic characteris- tics of characters like the height, width, shape and size and font style. This includes all the alphanumeric characters 10 digits(0-9) and 26 alphabets with uppercase and lowercase for a total of 62 characters.

Handwritten characters are more challenging to recognize as compared to printed characters as they carry more parameters like the standard and cursive style of writing. These challenges can be overcome by using new technologies like LSTMs, RNNs, CNNs and different training methods with various datasets.RNNs can be trained for sequence generation by processing real data sequences one step at a time and predicting what comes next.

Novel sequences can be generated from a trained network by iteratively sampling from the network’s output distribution, then feeding the sample in as input at the following phase, assuming the predictions are probabilistic. To put it another way, the network treats its inventions as if they were real, much like a dreaming person.

Although the network is deterministic in nature, the stochasticity introduced by sampling results in a distribution of sequences. Because the internal state of the network, and hence its predictive distribution, is dependent on past inputs, this distribution is conditional. Chapter 2 Literature Review • In paper[1] Feature Set Evaluation for Offline Handwriting Recognition Systems: Ap- plication to the Recurrent Neural Network Model, we investigated that the purpose of the function is to remove unwanted variations in the form of individual writing styles from word image documents that are converted to symbolic strings. Design the feature anyway and keep only the information related to recognition.

• In paper [2] An Off-Line Cursive Handwriting Recognition System Enhancements, we studied that in normalization and in the detection and representation of features have led to reduced error rates. A hybrid system using recurrent neural networks and HMMs was found to perform better than a discrete probability HMM system, The system performance was increased by allowing more frames of context in the network inputs and by "untying" the output distributions, to distinguish between the different parts of each character,the error rate has been reduced to 9.8 percent • In paper[3] Diagonal Based Feature Extraction For Handwritten Character Recognition System Using Neural Network.

we learned the use of techniques such as preprocessing, segmentation, and feature extraction that are used to improve the accuracy of text recognition and identification. The first important step in any handwritten recogni- tion system is pre-processing followed by segmentation and feature extraction. • In paper[4] Text Detection and Recognition Using Enhanced MSER Detection and a Novel OCR Technique, we studied the work of different text detection methods on how they extract text from an image such as MSER and canny edge integration, stroke width variation and OCR for text recognition.

The term "extremal region" represents connected component which differentiate the higher or lower intensity of a pixel to the outer boundary pixel. It is also difficult matter that how to group character candidates into text candidates. For the purpose, generally two types of approaches are used: rule based, clustering based.

• In paper[5] Combination of CNN and LSTM showing a better performance for Offline Handwriting Recognition , we studied that it shows the benefits of combining HMM, LSTM, CNN in visual recognition and performance • In paper[6] Convolutional Neural

Network Committees For Handwritten Character Classification, we studied that how using CNNs and their training with GPUs to train 2 models can reduce character recognition error rates in offline handwriting.

We train five differently initialized nets on each preprocessed dataset as well as on the original data, for a total of 35 CNNs (Table II). Each CNN is trained for 30 epochs by on-line gradient descent. The number of epochs is limited due to the size of NIST SD 19: training a single net for the 62 class problem takes almost six hours • In paper[7] Fast and robust training of recurrent neural networks for offline handwriting recognition, we studied that LSTM-RNN with Backpropagation Through Time train- ing can be used to speed up the process while increasing the accuracy of recognition process.

When recognizing handwritten text, especially the long short-term memory recurrent neural network LSTM-RNN is used in many modern handwriting recognition systems. • In paper[8] HMM based online handwriting recognition, we studied that in handwriting recognition using hidden Markov modeling and stochastic language modeling methods originally developed for speech applications. These methods are generalized in the AEGIS architecture.

Subcharacter models called nebulous stroke models are used to model the basic units in handwriting • In paper[9] Generating Sequences With Recurrent Neural Networks, we studied that Long Short-Term Memory recurrent neural networks generate both discrete and real- valued sequences with complex, long-range structure using next-step prediction • In paper[10] The State of the Art in Online Handwriting Recognition, we studied about pre-processing, segmentation, stroke width variation, shape recognition, post- processing, optical character recognition • In paper[11] Machine Printed Text and Handwriting Identification in Noisy Document Images, we studied that the technique can be used for image enhancement to improve page segmentation accuracy of noisy documents.

After noise identification and re- moval, the zone segmentation accuracy increase from 53 percent to 78 percent using the Docstrum algorithm,segmenting and identifying text from extremely noisy docu- ment images. Instead of using simple filtering rules, we treat noise as a distinct class and use statistical classification techniques to classify each block into machine printed text, handwriting,and noise • In paper[12] A novel connectionist system for unconstrained handwriting recogni- tion,we studied the key features of the network are the BLSTM architecture, which provides access to long-range bidirectional contextual information, and the CTC out- put layer, which allows the network to be trained on unsegmented sequence data, the new approach outperformed a state-of-the-art HMM-based system and also proved more robust to changes in dictionary size • In paper [13] Online Handwriting Recognition with Support Vector Machines - A Ker- nel Approach.they used a technique that combines dynamic time warping (DTW) and support vector machines (SVM) by integrating DTW into a Gaussian SVM kernel.

The benefit of this approach is the absence of restrictive assumptions about class con- ditional densities, as made in conventional HMM based techniques. The only essential 3 assumption made is the selection of the kernel.A problem of this approach is the com- plexity. • In paper [14] full English sentence database for off-line handwriting recog-nition.

we understood that it is potentially useful for recognition of general unconstrained English text utilizing knowledge beyond the lexicon level.Their primary aim was to aid in automatic labeling of the database, but they can be integrated in any recognition system as well. • In paper[15] Methods of combining multiple classifiers and their applications to hand- writing recognition, we studied that the combination of several independent classifiers is a general problem that occurs in various application areas of pattern recognition and the experimental results on the recognition of totally unconstrained handwritten numerals have shown that the performances of individual classifiers could be improved significantly by the combination approaches.

4 Chapter 3 Project Design Proposed System Architecture In our model we have use a data set of hand writings.We have used CNN, RNN, CTC and LSTM. In our project we will push the image into the CNN layer and extract the features of that image and pass them to the RNN layer which will provide us with a matrix which we will pass to the CTC layer.

5 Activity diagram In general, handwriting recognition is classified into two types as off-line and on-line handwriting recognition methods. In the off-line recognition, the writing is usually cap- tured optically by a scanner and the completed writing is available as an image. A typical handwriting recognition system consists of pre-processing, segmentation, feature extraction, classification and recognition, and post processing stages.After the required pre-processing we will pass the handwriting into the model which will return digitised text of the handwrit- ing.If the user text as an input we will scan the text file and transfer the text to the model which will then create an object containing text styles.This object defines what style the hand writhing will be and so the model return an image containing generated handwriting.

Workflow The two most popular forms of handwriting recognition systems are offline and online hand- writing recognition. In off-line recognition, the writing is generally captured optically by a scanner, and the finished writing is provided as an image. A typical handwriting recognition system includes pre-processing, segmentation, feature extraction, classification and recogni- tion, and post-processing.

We'll feed the handwriting into the model, which will provide digitised text of the handwriting after the necessary pre-processing. If the user provides text as input, we will scan the file and send the data to the model, which will build an object with text styles. This object specifies the handwriting style, and the model returns an image of the created handwriting.

6 Handwriting to text: CNN: Input from the user is first inserted into CNN layers. CNN layers are used to extract the features from the image. Five input layers are used, of which the first two layers apply a filter of 5x5 and the last three of 3x3 for convolution operations. After the input passes through the five layers of CNN, a non-linear RELU function is applied.

Finally, a layer is used to summarise image regions, and a downsized output is generated. Even though the image is downsized by 2 in each layer, a feature map is added, which has a size of 32x256. RNN: The RNN passes the essential information to the feature sequence, which comprises 256 features each time step.

The popular Long Short-Term Memory (LSTM) RNN implementation is utilised since it can transmit information over longer distances and has more strong training features than a vanilla RNN. The RNN output sequence is mapped in a 32x38 matrix. The IAM dataset contains a total of 79 distinct characters; however, one additional character is required for the CTC process, resulting in 80 entries for each of the 32 time steps.

Connectionist Temporal Classification (CTC): The CTC is supplied to the RNN output matrix and the ground truth text during training the Neural Network, which computes the loss value. The CTC is only provided the matrix while inferring, and it decodes the final text using that information. Ground truth and recognised text can only be 32 characters long. Input data is an image of size 128x32.

Images in the dataset do not have this exact size, so we resize it to either a width of 128 or a height of 32. Then the input is converted into a target image (white) of size 128x32.Data augmentation can be implemented by arbitrarily moving or enlarging the image rather than aligning it to the left. CNN Output: Each entry in the output has 256 characteristics.

The RNN layers further process the fea- tures that we have received. Some of the features, however, already show a strong link with specific high-level aspects of the input image. Some features have a strong relationship with duplicate characters (for example, ”tt”), specific characters (for example, ”e”), or character qualities such as loops and curves (for example, in handwritten ”l”s or ”e”s). Implementation using Tensorflow: The implementation consists of 4 modules: SamplePreprocessor.py: To prepare the images from the dataset for the Neural Network. DataLoader.py: Reading sam- ples, putting them into batches, and providing an iterator-interface to go through the data. Model.py: To create the model from the given architecture, loads and saves models, manages the session simultaneously, providing an interface for training and inference. main.py: This file puts all the previously mentioned modules together. We only look at Model.py, since the other source files are concerned with basic file IO, input and output (DataLoader.py), and image processing (SamplePreprocessor.py).

We used online handwriting data to see if the prediction network could also be used to construct convincing real-valued sequences (online in this context means that the writing is recorded as a sequence of pen-tip locations, as opposed to offline handwriting, where only the page images are available). Because of its low dimensionality (two real integers per data point) and ease of visualisation, online hand-writing is a popular choice for sequence production.

The IAM online handwriting database provided all of the data for this article (IAM-OnDB). IAM-OnDB is made up of 221 distinct writers’ handwritten lines collected on a ”smart whiteboard.” The writers were invited to fill out forms from the Lancaster-Oslo-Bergen text corpus, and the position of their pen on the board was tracked using an infrared device in the corner. The training data samples are shown in.

The x and y pen coordinates, as well as the points in the sequence when the pen is taken from the whiteboard, make up the original input data. Interpolating to fill in for missing readings and deleting steps whose length exceeded a specific threshold were used to fix recording mistakes in the x, y data. Aside from that, no preprocessing was applied, and the network was trained to predict the x, y, and end-of-stroke markers one at a time.

29 Most approaches to handwriting recognition and synthesis, on the other hand, rely on com- plex preprocessing and feature extraction techniques. We avoided such strategies because they tend to limit the variety in the data that we wanted the network to simulate (e.g., by normalising the character size, slant, skew, and so on).

Predicting pen traces one at a time provides the network the most flexibility in terms of inventing new handwriting, but it also demands a lot of memory, as the typical letter takes up more than 25 timesteps and the average line takes up over 700. It’s especially difficult to predict delayed strokes (such as dots for ’i’s or crosses for ’t’s that are inserted after the remainder of the text has been typed).

IAM-OnDB is separated into three sets: a training set, two validation sets, and a test set, including 5364, 1438, 1518, and 3859 handwritten lines chosen from 775, 192, 216 and 544 forms. Each line was handled as a separate sequence in our research (meaning that possible dependencies between successive lines were ignored). We used the training set, test set, and the larger of the validation sets for training and the smaller validation set for early-stopping to maximise the amount of training data.

Because there is no independent test set, the recorded results may be overfit on the validation set; nonetheless, the validation findings are of secondary consequence because there are no benchmark results and the main purpose was to develop convincing-looking handwriting. The most difficult aspect of using the prediction network to analyse online handwriting data was determining a predictive distribution that was adequate for real-valued inputs. The following section describes how this was done. 30 Chapter 5 Testing 5.1

Unit Testing Unit Testing is the first level of testing, which is typically performed by the developers them- selves. We tested the working of both modules, text to handwriting as well as handwriting to text. It helped us understand the desired output of each module, which we had broken down into separate units. This approach came in handy for us to understand each unit.

5.2 Integration Testing Integrated tests are usually made up of a mix of automated functional and manual tests, and they can be carried out by developers or independent testers. It is the phase in software testing in which individual software modules are combined and tested as a group.

Integra- tion testing is conducted to evaluate the compliance of a system or component with specified functional requirements. The next step is to put them all together into one module. This testing is essential for establishing which devices will perform flawlessly together. 5.3 Compatibility Testing A compatibility test is an assessment used to ensure a software application is properly work- ing across different browsers, databases, operating systems (OS), mobile devices, networks and hardware. The system does not require more than 4GB RAM even when both of the modules are running.

The system can run on very low specification devices. 31 Chapter 6 Result Our model contains two parts, in the first part the user enters their own handwriting as an input,the model will perform pre-processing on the input then will return the digitised text version of the given input.

In the latter part the user enters digitised text as an input and the model will return a generated handwriting in the form of an image as an output.In our model we have implemented neural networks, we have achieved an accuracy of 64% for our model. 32 Figure 6.1: Input 1 Figure 6.2: Result 1 Figure 6.3: Input 2 Figure 6.4: Result 2 Figure 6.5: Input 3 Figure 6.6: Result 3 33 Figure 6.7: Text to Handwriting Example 1 34 Figure 6.8: Text to Handwriting Example 2 35 Chapter 7 Conclusions and Future Scope We discussed a Neural Network which is able to recognize text in images.

Our model can successfully convert the handwriting's given by the user into text and visa with an accuracy of 64%. We aim to create a more efficient handwriting recognition model. We discussed a Neural Network which is able to recognize text in images. The Neural Network consists of 5 CNN and 2 RNN layers and outputs a character-probability matrix.

This matrix is either used for CTC loss calculation or for CTC decoding. An implementation using TensorFlow is provided. The handwriting recognition model's accuracy can be enhanced. To further its versatility, the model may be trained to detect cursive and special characters. Scanning the entire page for handwriting recognition might potentially be included as a feature.

Changing the text to handwriting output file to a ruled page rather than a blank page. This model can be used to fill out forms that are cumbersome to fill out by hand. For example, a PAN card or a passport.

INTERNET SOURCES:

<1% - http://cdn.iiit.ac.in/cdn/cvit.iiit.ac.in/images/ConferencePapers/2018/offline-handwriting-recognition.pdf
<1% - https://link.springer.com/article/10.1007/s00521-021-05735-y
<1% - https://www.ijedr.org/papers/IJEDR1704192.pdf
<1% - https://d1r4g0yjvcc7lx.cloudfront.net/uploads/20190215074900/Literature-review-behaviour-change-campaigns-Jan-2019.pdf
<1% - https://1library.net/article/conclusions-conclusions-and-future-scope.zpxo0l4q
<1% - https://www.sciencedirect.com/science/article/pii/S0893608014002135
<1% - https://www.researchgate.net/publication/358194690_Deep_learning_via_LSTM_models_for_COVID-19_infection_forecasting_in_India

<1% - http://ijcsit.com/docs/Volume%207/vol7issue1/ijcsit2016070101.pdf
<1% - https://www.sciencedirect.com/science/article/pii/S1746809422002191
<1% - https://addepto.com/text-extraction-from-images-using-machine-learning/
<1% - https://link.springer.com/chapter/10.1007/978-981-16-4863-2_6
1% - https://tutorials.one/making-predictions-with-sequences/
<1% - https://www.deepdyve.com/lp/spie/hybrid-training-procedure-applied-to-recurrent-neural-networks-mPb3adsOmN
<1% - https://www.academia.edu/27677609/HANDWRITTEN_CHARACTER_RECOGNITION_USING_FEED_FORWARD_NEURAL_NETWORK_MODELS
<1% - https://link.springer.com/article/10.1007/s12652-018-1075-x
<1% - https://www.semanticscholar.org/paper/Text-detection-and-recognition-using-enhanced-MSER-Islam-Mondal/fa46ae777be8776a417a24e0b6c3f6076c5e578d/figure/2
<1% - https://issuu.com/editorijaerd/docs/scene_text_recognition_by_using_ee-
<1% - https://www.researchgate.net/publication/311754391_Text_detection_and_recognition_using_enhanced_MSER_detection_and_a_novel_OCR_technique
<1% - https://www.researchgate.net/publication/348089241_EEG-Inception_A_Novel_Deep_Convolutional_Neural_Network_for_Assistive_ERP-based_Brain-Computer_Interfaces
<1% - https://www.researchgate.net/publication/308867890_A_context-sensitive-chunk_BPTT_approach_to_training_deep_LSTMBLSTM_recurrent_neural_networks_for_offline_handwriting_recognition
1% - https://www-i6.informatik.rwth-aachen.de/publications/download/960/Doetsch-ICFHR-2014.pdf
<1% - https://www.semanticscholar.org/paper/Online-handwriting-Farsi-character-and-number-based-Sadrnezhad-Nekouie/d8c51a6f5b4edf4decbac09ba1d4e48c00319a76
<1% - https://www.sciencedirect.com/science/article/pii/S0031320399000436
<1% - https://arxiv.org/abs/1308.0850v5
<1% - https://www.researchgate.net/publication/3191825_State_of_the_art_in_on-line_handwriting_recognition
<1% - https://www.researchgate.net/publication/220861004_Text_Identification_in_Noisy_Document_Images_Using_Markov_Random_Field

<1% -
https://medium.com/analytics-vidhya/image-segmentation-techniques-using-digital-im
age-processing-machine-learning-and-deep-learning-342773fcfef5
<1% -
https://www.researchgate.net/publication/8337752_Machine_Printed_Text_and_Handwrit
ing_Identification_in_Noisy_Document_Images
<1% -
https://www.academia.edu/6608285/Using_Feature_Extraction_to_Recognize_Handwritte
n_Text_Image
<1% - https://link.springer.com/article/10.1007/s12559-010-9041-8
<1% -
https://www.researchgate.net/publication/318509381_Biometric_Signature_Verification_
Using_Recurrent_Neural_Networks
<1% -
https://www.researchgate.net/publication/220933043_A_kernel-based_approach_to_doc
ument_retrieval
<1% - https://link.springer.com/chapter/10.1007%2F978-1-84800-007-0_12
<1% -
https://www.researchgate.net/publication/308860256_Building_Handwriting_Recognizer
s_by_Leveraging_Skeletons_of_Both_Offline_and_Online_Samples
1% - https://www.scribd.com/presentation/560847379/HTR-Presentation
<1% - https://core.ac.uk/display/22965413
<1% -
https://www.researchgate.net/publication/221225548_A_New_Multiple_Classifiers_Comb
ination_Algorithm
<1% -
https://www.researchgate.net/publication/220183188_A_Method_of_Combining_Multipl
e_Experts_for_the_Recognition_of_Unconstrained_Handwritten_Numerals
<1% - http://www.jonathanleroux.org/pdf/Moritz2019ASRU12.pdf
<1% - http://utpedia.utp.edu.my/15917/1/Muhammad%20Nizarshyami_16042.pdf
<1% -
https://www.researchgate.net/publication/290454293_A_novel_approach_for_handednes
s_detection_from_off-line_handwriting_using_fuzzy_conceptual_reduction
1% - https://www.ijcaonline.org/volume8/number9/pxc3871693.pdf
<1% - http://www.mirlabs.net/his19/proceedings/22220121.pdf
<1% -
https://www.ijert.org/handwritten-text-recognition-using-deep-learning-with-tensorflow
<1% -
https://github.com/Dream-kid/Flow-Chart-Generator-From-Hand-Drawn-Chart-And-Ha
ndwriting-Recognition

<1% - https://www.geeksforgeeks.org/unit-testing-software-testing/
1% - https://www.chegg.com/homework-help/questions-and-answers/testing-phase-tests-individual-software-modules-combined-together-group-o-white-box-testin-q93791007
1% - https://en.wikipedia.org/wiki/Integration_testing
<1% - https://www.bloggersideas.com/dropfunnels-review/
1% - https://www.techtarget.com/whatis/definition/compatibility-test
<1% - https://www.researchgate.net/publication/221145928_Keyword_Spotting_Techniques_for_Sanskrit_Documents
<1% - https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5
<1% - https://co17349.medium.com/handwritten-text-recognition-system-using-tensorflow-2863b9e23782