Journal of Applied Information Science
12 (2) 2024, 01-12
http://www.publishingindia.com/jais

Handwritten Text Recognition System using Deep Learning Techniques

Yakuta Tayyebi[1*], Nabiya Seikh[2] and Priyanshee Parmar[2]

[1]Assistant Professor (CSE), Prestige Institute of Engineering Management & Research, Indore, Madhya Pradesh, India. Email: tayyebiyakuta@gmail.com

[2]Scholar, B.Tech (CSE), Prestige Institute of Engineering Management & Research, Indore, Madhya Pradesh, India.

[*]Corresponding Author

Abstract: Handwritten Text Recognition (HTR) is for converting handwritten data into digital formats. Handwritten Text Recognition is a critical area of research due to the increasing need to convert vast amounts of handwritten data into digital formats. This paper aims to explore the challenges, techniques and advancements in developing efficient handwritten text recognition systems. Various studies have focused on languages and scripts utilizing approaches like deep learning, feature extraction, and segmentation methods. The research landscape includes the use of artificial intelligence, convolutional neural networks, and pattern recognition to enhance the accuracy and applicability of handwritten text recognition systems. By synthesizing these diverse methodologies and findings, this paper contributes to the ongoing efforts to improve the recognition of handwritten text across various domains. Through a comprehensive examination of current research and technological approaches, this paper seeks to provide insights that will drive the continuous improvement of HTR systems, ultimately facilitating better digital transformation of handwritten information.

Keywords: Deep learning, Handwritten Text Recognition (HTR), Optical Character Recognition (OCR).

I. Introduction

Converting handwritten text into a machine-readable text is known as handwritten text recognition, or HTR. The primary challenge facing Handwritten Text Recognition (HTR) systems is the wide range of handwriting styles, which can differ greatly amongst authors. The goal of a handwritten text recognition system is to translate handwritten text into digitally, machine-readable text by implementing character recognition system. This will enable effective extraction of text from handwritten documents.

In recent times, there has been a surge in interest in handwritten text recognition within the domain of pattern recognition. Numerous researchers have proposed various techniques aimed at facilitating the transcription of an array of documents, including general document forms, contemporary documents, medical documents and doctor’s prescriptions and historical archives through both offline and online approaches [1-3]. Fig. 1 illustrates the category of the handwritten text recognition systems based on strategies used. Transcription refers to process of automatically converting digital image of handwritten text into its machine recognizable data. Optical Character Recognition (OCR) [4] represents the benchmark technique of this field. Optical Character Recognition (OCR) stands out as a fundamental technique in this domain. OCR typically comprises two primary phases: text detection, which involves segmenting the text into smaller patches, and content recognition, where the contents of these patches are recognized and translated into machine-readable text. The initial and most basic OCR system, developed specifically for recognizing Latin numerals, laid the groundwork for subsequent advancements in the field [5].

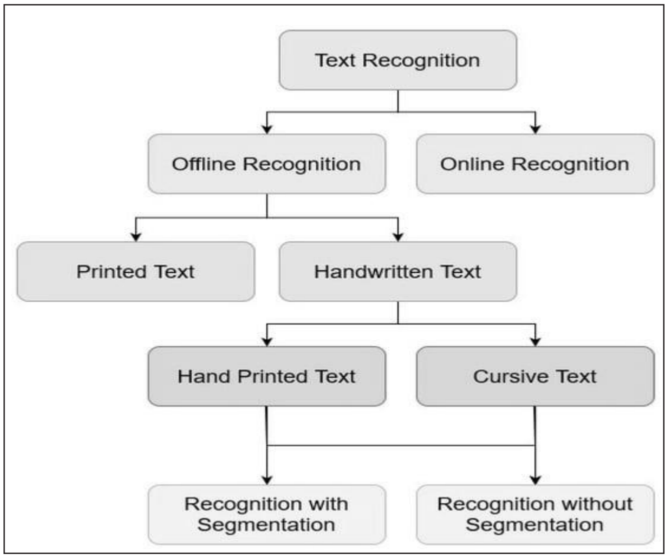

Fig. 1: Handwritten Text Recognition Systems

II. Literature Review

Handwritten text recognition is the process of recognizing characters in documents, photos, and other sources and converting them into formats that computers can understand. It is also known as HTR. Accurately recognizing complexly constructed compound handwritten characters remains a serious challenge.

Among the methods used to extract text from a picture are text detection, text segmentation, and text recognition. Python is used to code IAM OCR and OpenCV for this. An optical text recognition system with several algorithms is required for this. HP developed IAM, the most accurate optical text recognition engine on the market at the moment. OpenCV (Open-Source Computer Vision) is a library used for image processing and performing computer vision tasks.

Following section presents work done on handwritten text recognition of cursive words.

In this study, H. Bunke *et al.,* employ a holistic approach to cursive word recognition. This methodology involves the successive transformation of words through various phases, including letters, complete words, points and other features. Connection between letters and features are leveraged to generate a unique feature vector from the image, and word candidates are matched against a lexicon to identify partially recognized terms. Only a limited subset of the lexicon's 130 terms is actually recognized. Unlike conventional approaches employing classifiers, word recognition in this method involves assigning ratings to pre-segmented segments using letter hypotheses, with the segments then identified based on the highest rating [6].

Moudgil, Singh, and Gautam provide a comprehensive overview of recent trends in Optical Character Recognition (OCR) systems, focusing on their application to historical manuscripts. The paper discusses challenges in recognizing various scripts, degradation of documents, and the need for pre-processing techniques. It highlights advancements in deep learning and neural networks for OCR accuracy, as well as the role of large annotated datasets in enhancing system performance for diverse languages and scripts [7].

In S**.** Sudhakar work, a Hidden Markov Model (HMM) is utilized to recognize isolated and cursive handwritten letters. To enhance the effectiveness of the HMM, a hybrid approach is adopted. The median values of black runs in each scan line serve as features for text recognition. Four different scan directions are employed to extract features from a character image, resulting in a sparse directional representation of the character. For recognition, a discrete density left-to-right HMM is employed. Researches have proposed a recognition-based segmentation strategy for cursive handwriting. Higher-order HMMs utilize features to validate the segmentation path. The graph search approach is employed to identify the correct segmentation points by determining the shortest path at the lowest cost. In recognition, the HMM's probability of the observation sequence is utilized [8].

Researchers have adopted a recognition-based segmentation approach for handwritten cursive word identification. This study compares two approaches. The first approach utilizes a combination of neural networks and Hidden Markov Models (HMMs) for object recognition. In contrast, the second approach employs a discrete HMM. In the latter approach, segmentation graphs are employed for initial word segmentation. The likelihood of each word in the lexicon is computed by the HMM by summing the probabilities along each potential path in the graph. Subsequently, a neural network is used to calculate probability for each letter hypothesis in the graph. In the second technique, 140 geometric characteristics are derived from each pre-segmented segment. These characteristics are then transformed into a single symbol using vector quantization (VQ) [9].

Y. Wu, X. Chen and Z. Wang, utilized a segmentation-based technique to recognize cursive handwritten words. This method initially segments cursive words into individual characters, which are then combined and recognized to form meaningful words by cross-referencing them with dictionaries. The lexicon used in this work comprises 26 terms, limiting the scope of the paper to recognition of these 26 words. The handwritten text recognition in the following paper is done for individual characters [10].

Using neural networks, researchers created a recognition system. For the same training dataset, they employed three distinct Neural Network (NN) topologies: radial basis function network, closest neighbor network, and back propagation neural network. They evaluated each network's performance and tuned the hidden layer's number of neurons so that it is independent of the starting value. They came to the conclusion that combining feed forward back propagation with a typical feature extraction approach produced the best results [11].

This work introduces the Euclidean Distance Measure (EDM) method, achieving a recognition accuracy of 90.77%. The learning rule via Artificial Neural Networks (ANN) increases precision to 95.38% in cases of misclassification. Further enhancement to 98.46% is attained by multiplying recognition scores by Euclidean distances [12].

In their 2021 paper, J. Kang, D. Kim and J. Hwang introduce a novel approach for handwritten document recognition that combines a Convolutional Neural Network (CNN) with a graph search strategy. The CNN is employed to extract features from handwritten text images, while the graph search method is used to find optimal segmentation points in the handwritten text. This hybrid approach aims to improve the accuracy and efficiency of recognizing handwritten characters and words, addressing challenges such as the variability in handwriting styles and the complexity of cursive script. The proposed method achieves significant performance improvements in handwritten text recognition tasks [13].

Researcher, S. Stiehr, M. Diem and R. Sablatnig proposed a hybrid approach combining Long Short-Term Memory (LSTM) networks and Hidden Markov Models (HMMs) for the recognition of historical handwritten documents. The LSTM networks are employed to capture the sequential nature of handwritten text, effectively handling long-term dependencies and variability in handwriting. The HMMs are used to model the underlying structure of the text and improve segmentation accuracy. This combination leverages the strengths of both techniques, enhancing the recognition performance on challenging historical documents. The approach demonstrates significant improvements in accurately transcribing historical handwritten texts compared to traditional methods [14].

B. Shi, X. Bais and C. Yao present a novel Convolutional Long Short-Term Memory (ConvLSTM) network for handwritten text recognition. This network combines the strengths of Convolutional Neural Networks (CNNs) for feature extraction and LSTM networks for sequence modeling. The ConvLSTM framework effectively captures spatial dependencies in handwritten text images and temporal dependencies in sequences of characters. This hybrid approach improves the recognition of handwritten text by leveraging the ability of CNNs to handle complex image features and the proficiency of LSTMs in managing sequential data. The proposed model achieves significant improvements in accuracy and efficiency over traditional methods in handwritten text recognition tasks [15].

F. Ahmed, S. S. I. Abedin, and M. F. Bari provide an extensive review of handwritten text recognition (HTR) systems and methodologies in their 2019 paper. They cover a broad spectrum of techniques, including traditional machine learning approaches and modern deep learning methods. The paper discusses various feature extraction techniques, segmentation methods, and classification algorithms employed in HTR. Additionally, it evaluates the performance of these methods on different datasets and highlights the challenges and future directions in the field. The review serves as a comprehensive resource for researchers, offering insights into the evolution and advancements in HTR technologies [16].

HTR systems have substantially improved since the time of utilizing the Hidden Markov Model (HMM) and handcrafted features for text recognition [17]. However, the efficiency of HMMs is low because of drawbacks like memory lessens [18] and the limitation of manual feature selection process. To address these issues, researchers proposed hybrid systems that integrate some advance technique with HMM. Example include HMM combined with Gaussian mixture emission distributions [19], HMM deployed with Convolutional Neural Network (CNN), or HMM integrated with Recurrent Neural Network (RNN) [20], all of which have remarkably improved recognition accuracy.

Today, systems are capable of analyzing document layouts and recognize text, paragraphs, and entire documents. These advanced systems can identify various handwritten styles in languages such as Chinese, French, Arabic, Latin and more. This progress is largely due to the use of machine learning techniques, including Convolutional Neural Networks [21], Recurrent Neural Networks (RNN), Convolutional Recurrent Neural Networks (CRNN) [22], Gated-CNN [23], Multi-Dimensional Long Short-Term Memory Recurrent Neural Networks (MDLSTM-RNNs) [24]. With emergence of large language model, HTR systems can be integrated to generate understandable natural language [27]. Despite these remarkable advancements in the last decade, there still exist some challenges that need to be focused.

III. Proposed Methodology

A. Deep Learning

It is easy to define two or two artificial neural networks as deep learning architectures (ANNs). Prediction accuracy is being increased by adding more hidden layers. Comparing DL to typical artificial neural networks, additional hidden layers are used. Traditional deep neural networks (DNN) employ weighting. The preprocessed input is sent through a nonlinear activation function, to produce the output.

Consequently, the aim of DNN training is to minimize the loss function by optimizing the network weights. Every lower-level trait is used to define the higher-level characteristic.

B. CNN - Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a supervised deep learning architecture. Software for image analysis uses it most frequently. The three layers are a fully connected layer, a pooling layer, and a convolutional layer that comprise a CNN. To create different feature maps, the input picture is processed through the convolution layer's filters or kernel.

Each feature map is shrunk by the pooling layer in order to maintain a minimal number of weights. The method known as down sampling or subsampling is employed. Three distinct types of pooling processes are maximum pooling, average pooling, and global pooling. Following these layers, a completely linked layer is used to classify the 2D feature map once it has been converted into a 1D vector. In Fig. 2, the CNN technique for image classification is displayed.

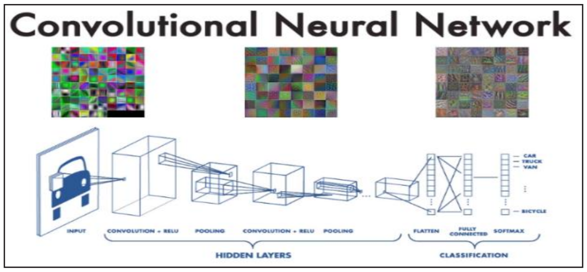

Fig. 2: CNN

Fig. 2 shows the architecture of a CNN with two folding layers. Every convolution layer is followed by a pooling or subsampling layer. The output of the last pooling layer is fully linked. Provided the last output layer is Layer 35. In 36, Deeper—a novel end-to-end deep learning system—was presented as a means of identifying patterns and obtaining critical information from medical records. Using a collection of discrete components, convolutional neural networks are utilized to anticipate unplanned restart following a discharge. Subsequently, Cheng examined the temporal features of the patient's electronic health record using DL technology. In the second layer of the suggested DL, the convolution operator was applied in the temporal dimension of the patient's EHR matrix.

C. RNN - Recurrent Neural Networks

The function of Recurrent Neural Networks (RNNs) in a Handwritten Text Recognition (HTR) system is to analyze sequential data and capture temporal dependencies inherent in handwritten text. RNNs process sequential inputs, such as individual characters or strokes in handwriting, by maintaining an internal state that enables them to remember past information. This capability allows RNNs to effectively model the structure and context of handwritten text, making them well-suited for tasks like handwriting recognition. In an HTR system, RNNs can be used to encode sequential input data, extract relevant features, and generate output predictions, contributing to the overall accuracy and efficiency of the recognition process.

D. Our Approch

i) Architecture of HTR System

Overall architecture and the components of handwritten text recognition system are denoted in Fig. 3, through a block diagram. The dataset is randomly divided into two parts, training data and testing data. Training data are used to train the system and this trained model is then employed to recognize handwritten test data.

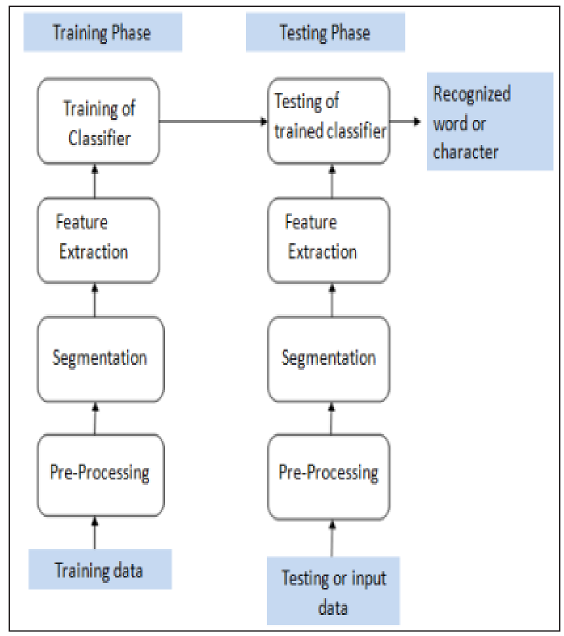

Fig. 3: Block Diagram of HTR System

ii) Pre-Processing

Pre-processing involves a series of operations applied to the scanned input image to enhance its quality and make it suitable for subsequent processing. This stage includes tasks such as noise removal, binarization, and skew correction, among others.

iii) Noise Removal

Noise removal is crucial in digital image processing as it eliminates unwanted pixels that can distort the original information. Several methods have been developed for noise reduction. These include sophisticated approaches such as non-local means

[25] and anisotropic diffusion [26], as well as traditional methods like Gaussian, Mean, and Median filters. Each method aims to enhance image quality by effectively reducing noise while preserving important details and structures in the image.

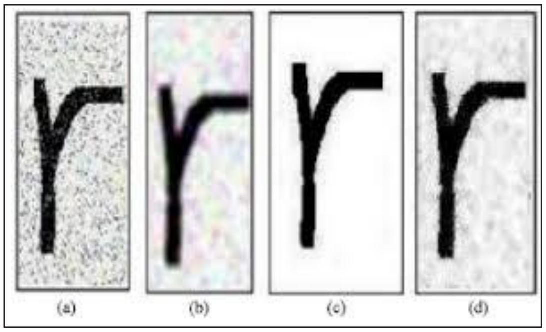

Fig. 4: Noise Removal

iv) Binarization and Skew Correction

Binarization is the process of converting a grey scale image into a binary image using thresholding technique like Otsu's method of thresholding. Fig. 5 demonstrates the output of binarization [28].

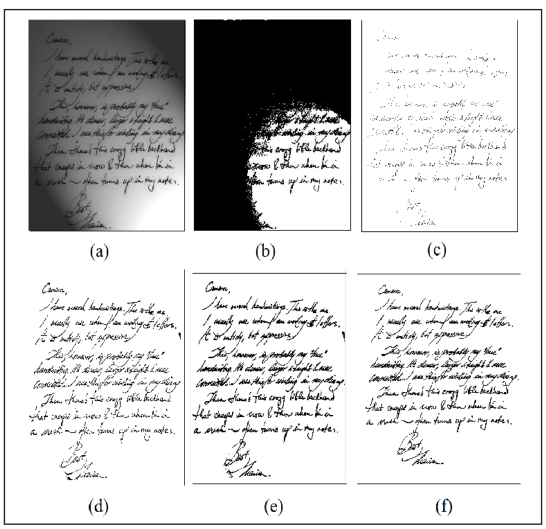

Fig. 5: Binarization

Skew correction is elimination of skewness in input data for its proper processing. Handwritten documents are not absolutely horizontally aligned. Skew correction methods are therefore necessary before any other processing. Fig. 6 illustrates the implementation of skew correction.

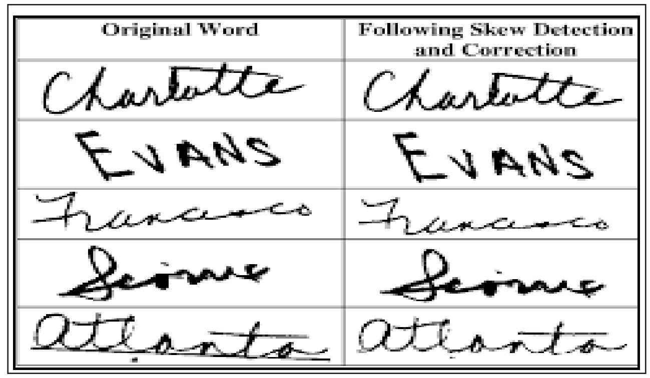

Fig. 6: Skew Correction

v) Segmentation

The process of segmenting handwritten text images involves categorizing the text into individual characters, words, lines, and paragraphs. This is done by often analyzing the pixel properties within the image. Various methods have been deployed for this purpose, including thresholding, region-based, edge-based, watershed, and clustering-based techniques. This segmentation stage is crucial, as it can remarkably enhance the accuracy of Handwritten Text Recognition (HTR) models [29].

vi) Feature Extraction

During this stage, the essential features of characters are extracted to facilitate their classification in the recognition phase. This is a critical step, as effective feature extraction can significantly enhance the recognition rate and minimize misclassification. Features such as binary and directional attributes are identified and compiled into a feature vector. Feature extraction methods generally fall into these categories. Fig. 7 demonstrates the process of feature extraction.

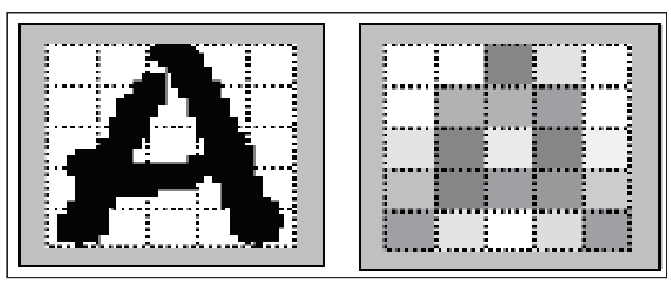

Fig. 7: Feature Extraction

Text recognition in handwritten cursive is done using four different approaches. The holistic approach recognizes words as a whole without breaking them apart by taking away their constituent aspects. *Segmentation-Based Approach:* Words are divided into segments. *Recognition-Based Segmentation Method:* Character classification and segmentation are carried out concurrently with the use of the suitable learning technique. *Mixed Approach:* This system combines the approaches mentioned before.

The following paper provides a brief summary of the HTR that is currently available for English. The benefits and drawbacks of HTR approaches are highlighted. To categorize the input characters, several characteristics are retrieved and classifiers of various kinds are employed. The current work focuses on exploring potential methods for creating an offline HTR system for the English language that can recognize both cursive and distinct characters.

E. Dataset

The dataset used IAM Handwriting, is a comprehensive resource for research and development in handwritten text recognition. This dataset has evolved as a standard reference point in the field of Optical Character Recognition (OCR) for handwritten text, as it includes a broad range of handwriting styles, offering diversity in terms of slant, spacing, and letter formation. The IAM Handwriting Database contains over 1,500 pages of handwritten text, each scanned in high-resolution TIFF format, providing a detailed visual representation of the written content. These pages were transcribed from original forms filled out by writers of varying backgrounds, reflecting a wide array of handwriting characteristics. Each form contains several lines of text, and the data is segmented into lines, words, and individual characters to facilitate detailed analysis and training of recognition models.

To ensure effective use of this dataset, researchers can access detailed annotations and transcriptions that map to specific segments within the scanned forms. These annotations identify the beginning and end of lines, delineate individual words, and provide detailed character-level information.

Data partitioning in the IAM Handwriting Database is structured to support robust training and evaluation of recognition models. It includes predefined sets for training, validation, and testing, allowing researchers to train models on a specific subset and then evaluate their performance on another, ensuring a rigorous assessment of model generalization. Typically, the training set comprises about 6,161 lines of text, the validation set includes 900 lines, and the test set contains 2,915 lines. This partitioning structure is crucial for evaluating the accuracy and robustness of OCR models and for comparing different approaches and algorithms.

To access the IAM Handwriting Database, you must register with the University of Bern in Switzerland, which manages the dataset. Registration requires agreeing to terms of use, primarily restricting the dataset to academic and non-commercial applications. Once registered and approved, users can download the dataset for research and development purposes. The Table I below provides the details of the dataset.

Table I: Details of Training Samples

	Printed			Handwritten		
Model	Perc.	Recall	Acu.	Perc.	Recall	Acu.
OCR	0.701	0.704	0.591	0.562	0.597	0.448
HTR	0.671	0.644	0.541	0.583	0.601	0.475
Dual	0.705	0.702	0.509	0.587	0.613	0.473

F. OCR

Optical Text Recognition (OCR) plays a critical role in Handwritten Text Recognition (HTR) systems by enabling the conversion of handwritten text into digital formats. OCR technology involves the recognition and interpretation of printed or handwritten characters from images, documents, or scanned text. In the context of HTR systems, OCR is employed to preprocess handwritten images, extract textual information, and facilitate subsequent recognition processes.

The use of OCR in HTR systems typically involves several key steps. Firstly, handwritten documents or images containing text are scanned or captured using digital imaging devices, such as cameras or scanners. These images are then preprocessed to enhance quality, remove noise, and improve readability, which may involve techniques such as image binarization, noise reduction, and edge detection.

Once preprocessed, the handwritten images are fed into the OCR engine, where the actual recognition process takes place. The OCR engine analyzes the input images, segmenting them into individual characters, words, or lines, and then extracts features from these segments to identify and classify the corresponding textual content. This feature extraction process may involve various advance techniques.

After feature extraction, the OCR engine utilizes pattern recognition algorithms to match the extracted features against a predefined set of character or language models, enabling the recognition and interpretation of handwritten text. These models may be trained using machine learning algorithms on labeled datasets to improve recognition accuracy and adaptability to different handwriting styles and languages. Once the handwritten text is recognized, it is converted into digital format, typically in the form of machine-readable text or structured data, which can be further processed, analyzed, or stored electronically. This digitized text can then be used for various purposes, such as indexing, searching, or analysis, facilitating the integration of handwritten documents into digital workflows and systems.

Overall, OCR technology plays a critical role in HTR systems by enabling the efficient and accurate recognition of handwritten text, thereby bridging the gap between analog and digital formats and facilitating the digitization and utilization of handwritten documents in various domains and applications.

G. TensorFlow

TensorFlow, is a well recognized open-source machine learning framework developed by Google, used in Handwritten Text Recognition (HTR) systems to implement OCR functionalities. TensorFlow provides a wide variety of libraries and tools for training neural networks, making it suitable for various OCR tasks, including preprocessing, feature extraction, and pattern recognition.

In a TensorFlow-based HTR system, handwritten images are first preprocessed using TensorFlow's image processing capabilities. This may involve tasks such as resizing, normalization, and noise reduction to enhance image quality and improve OCR accuracy. TensorFlow's extensive collection of image processing functions and pre-trained models, such as those available in the TensorFlow Image Processing library, can be leveraged for this purpose.

Once preprocessed, the handwritten images are fed into TensorFlow's neural network architecture for OCR, which typically involves the use of Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs). TensorFlow provides high-level APIs, such as Keras, for training deep learning models, allowing developers to easily construct complex neural network architectures for OCR tasks.

For feature extraction, TensorFlow provides a range of built-in functions and layers for convolution, pooling, and normalization, which can be used to extract meaningful features from handwritten images. Additionally, TensorFlow's extensive collection of pre-trained models, such as those available in the TensorFlow Model Zoo, can be fine-tuned or adapted for OCR tasks, reducing the need for extensive manual feature engineering.

For pattern recognition and text classification, TensorFlow provides various optimization algorithms, loss functions, and evaluation metrics to train and evaluate OCR models effectively. TensorFlow's flexibility and scalability enable programmers to experiment with different neural network architectures, hyperparameters, and training strategies to improve OCR performance. Once trained, the TensorFlow-based OCR model can be deployed in HTR systems to recognize handwritten text in real-time or batch processing scenarios.

TensorFlow's compatibility with various hardware platforms, including CPUs, GPUs, and TPUs, allows for efficient deployment and execution of OCR models across different environments.

TensorFlow serves as a powerful tool for implementing OCR functionalities in HTR systems, offering a comprehensive set of tools and libraries for preprocessing, feature extraction, pattern recognition, and model deployment. Its flexibility, scalability, and ease of use make it an ideal choice for building robust and efficient OCR solutions for handwritten text recognition.

H. CTC

Connectionist Temporal Classification (CTC) is a technique commonly used in Handwritten Text Recognition (HTR) systems to train neural networks for sequence labeling tasks. In the context of HTR, CTC allows the network to learn to recognize and transcribe sequences of characters directly from images of handwritten text, without the need for explicit alignment between input and output sequences. This makes CTC well-suited for handling variable-length sequences and diverse handwriting styles, making it a popular choice for training HTR models.

I. Gradio

Gradio is a Python library specifically designed for developing user interfaces for machine learning and data science models. It allows developers to create interactive demos for their models with minimal code. In this script, Gradio is used to create a web-based user interface that accepts an image input and various configuration parameters, processes the image, and outputs both the recognized text and a visual representation of the detected words.

J. OpenCV

OpenCV is Python library used for image processing and computer vision related tasks. In this script, OpenCV is used to read images (`cv2.imread`), draw rectangles on detected text regions (`cv2.rectangle`), and overlay text on the image (`cv2.putText`).

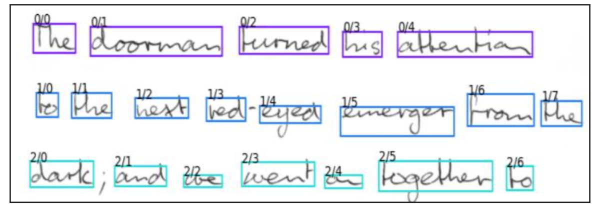

Fig. 8: Demonstration

K. Custom HTR Pipeline

The code imports functions and classes from a custom module called `htr_pipeline`. This module contains the `read_page` function for detecting and recognizing text from a page, and configurations for text detection and clustering (`DetectorConfig`, `LineClusteringConfig`, `ReaderConfig`). It also includes a `PrefixTree` class for handling a list of words, likely used for word-based recognition with a dictionary.

L. JSON

JSON (JavaScript Object Notation) is a lightweight data interchange format. The script uses it to read configuration settings from a `config.json` file. This configuration data is used to set up the examples for the Gradio interface, defining parameters like image scale, margin, and text size.

M. Python Standard Library

The code uses Python standard libraries such as `json` for parsing JSON files and basic file handling, as well as other typical Python functionalities like list comprehension and looping. These technologies work together to create a gradio-based interface for a handwritten text recognition system that reads an image, recognizes text, and provides visualization and output for the user.

IV. Results and Discussion

The output of the Handwritten Text Recognition (HTR) system using IAM and OpenCV depends on several factors, such as the accuracy of the OCR engine, the quantity and diversity of the training dataset, and the quality of the input images. In general, the accuracy of the IAM OCR engine's recognition of printed text is rather excellent. However, it may be more challenging to interpret handwritten content due to the variety of handwriting styles and typefaces. Deskewing, noise reduction, and binarization are a few picture pre-processing techniques that can improve the quality of the input photographs. This might lead to increased accuracy in text recognition.

Training the IAM OCR engine with a vast and diverse collection of handwritten photos can help improve accuracy. If the OCR engine is exposed to a range of handwriting styles and typefaces, it may pick up on different patterns and characteristics of handwritten text. It is important to verify the correctness of the OCR engine after training it with an alternative set of test images. This can demonstrate the accuracy of the OCR engine as well as areas for improvement. The detected text can be extracted and stored in a text file for additional analysis or processing at a later time. This has significant implications for applications such as document analysis, postal automation, and the digitalization of historical manuscripts.

All things considered, a handwritten text recognition system that transcribes handwritten text from pictures can be made using IAM and OpenCV. It's critical to correctly modify the training settings and pre-processing techniques to achieve the optimum outcomes.

V. Conclusion

The Handwritten Text Recognition (HTR) use of IAM and OpenCV is a challenging yet fascinating application of computer vision technology. When a large and diverse collection of handwritten photographs is used to train an OCR engine, it is feasible to detect and transcribe handwritten text from photos with a reasonable degree of accuracy.

Digitalizing historical manuscripts, automating the postal service, and document analysis are just a few of the many applications for HTR systems. It may also be used by students to assist them in translating their handwritten assignments and notes.

However, the accuracy of HTR systems can be influenced by several factors, including the OCR engine's precision, the quantity and variety of the training dataset, and the caliber of the input photographs. It's critical to correctly modify the pre-processing steps and training settings to achieve the best outcomes.

All things considered, an accurate and efficient HTR system that can identify and transcribe handwritten text from pictures may be constructed using IAM and OpenCV. Numerous areas may find applications for this technology.

References

[1] A. Graves, and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," *Advances in Neural Information Processing Systems (NIPS)*, vol. 21, pp. 545-552, 2008.

[2] A. Sarkar, U. Pal, and S. Mollah, "Recognition of handwritten indic scripts: A comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1332-1353, Jun. 2018.

[3] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298-2304, Nov. 2017.

[4] "AttentionHTR: Handwritten text recognition based on attention encoder-decoder networks," *arXiv preprint arXiv:2201.09390.*

[5] R. Smith, "An overview of the tesseract OCR engine," *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, Curitiba, Brazil, 2007, pp. 629-633.

[6] H. Bunke, M. Roth, and E. G. Schukat-Talamazzini, "Offline cursive handwriting recognition using Hidden Markov Models," *Pattern Recognition*, vol. 28, no. 9, Elsevier Science Ltd., 1995.

[7] A. Moudgil, S. Singh, and V. Gautam, "An overview of recent trends in OCR systems

for manuscripts," in *Cyber Intelligence and Information Retrieval: Proceedings of CIIR 2021*, 2022, pp. 525-533.

[8] S. Sudhakar, S. S. Sukhija, and N. Gupta, "Handwritten text recognition using deep learning approaches: A comprehensive review," *IEEE Access*, vol. 9, pp. 114840-114867, 2021, doi: https://doi.org/10.1109/ACCESS.2021.3105061.

[9] T. Ahmed, M. F. Ayub, and A. Mehmood, "A survey on automated recognition of handwritten text from images," *IEEE Access*, vol. 8, pp. 150964-150987, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3017460.

[10] Y. Wu, X. Chen, and Z. Wang, "Offline handwritten text recognition using graph convolutional networks and recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 2890-2906, Jun. 2022, doi: https://doi.org/10.1109/TPAMI.2021.3094992.

[11] M. Chowdhury, and M. Z. Hossain, "Handwriting recognition of Bangla text using a deep neural network," *IEEE Access*, vol. 8, pp. 98258-98267, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2996527.

[12] X. Liu, Z. Chen, and C. Wang, "A novel deep learning framework for recognizing handwritten Chinese characters," *IEEE Access*, vol. 7, pp. 102945-102953, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2928535.

[13] J. Kang, D. Kim, and J. Hwang, "Handwritten document recognition using a new convolutional neural network and graph search approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5420-5431, Dec. 2021, doi: https://doi.org/10.1109/TNNLS.2021.3063820.

[14] S. Stiehr, M. Diem, and R. Sablatnig, "Combining LSTM networks and HMMs for handwriting recognition of historical documents," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1796-1806, Jun. 2019, doi: https://doi.org/10.1109/TNNLS.2018.2877260.

[15] B. Shi, X. Bai, and C. Yao, "A convolutional LSTM network for handwritten text recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 1455-1467, 2020.

[16] F. Ahmed, S. S. I. Abedin, and M. F. Bari, "A comprehensive review of handwritten text recognition systems and methods," *IEEE Access*, vol. 7, pp. 135987-136004, 2019.

[17] G. Pozzebon, L. Rigazio, and F. Roli, "A review of deep learning techniques for handwritten text recognition," *IEEE Access*, vol. 8, pp. 142642-142657, 2020.

[18] A. Chowdhury et al., "An efficient end-to-end neural model for handwritten text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1587-1600, May 2021.

[19] H. Yılmaz, and B. Yanikoglu, "A hybrid HMM and Gaussian mixture model approach for handwritten text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2714-2727, Nov. 2019.

[20] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 6, pp. 1235-1248, Jun. 2020.

[21] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006, pp. 369-376.

[22] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298-2304, Nov. 2017.

[23] D. Karatzas et al., "ICDAR 2015 competition on robust reading," in *Proceedings of the 13th International Conference on Document*

Analysis and Recognition (ICDAR), 2015, pp. 1156-1160.

[24] Z. Zhang et al., "Multi-oriented text detection with fully convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4159-4167.

[25] D. L. Donoho, and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425-455, 1994, doi: https://doi.org/10.1093/biomet/81.3.425

[26] P. Perona, and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, Jul. 1990.

[27] A. Negi, Aradhana, C. V. Verma, and Y. Tayyebi, "Artificial intelligence empowered language models: A review," I*nternational Conference on Advances in Data-driven Computing and Intelligent Systems*, Singapore: Springer Nature Singapore, 2023.

[28] Y. Tayyebi, and D. S. Bhilare, "A comparative study of open source network based intrusion detection systems," *Int. J. Comput. Sci. Inf. Technol. (IJCSIT)*, vol. 9, no. 2, pp. 23-26, 2018.

[29] Y. Tayyebi, and D. S. Bhilare, "Cloud security through intrusion detection system (IDS): Review of existing solutions," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 4, no. 6, pp. 213-215, 2015.