Handwriting Recognition System using YOLO and CTC

Under the guidance of
Dr. Neeraj Garg
Head of Department, Department of Artificial Intelligence and Machine Learning,
Maharaja Agrasen Institute of Technology
Email: neeraj@mait.ac.in

Gautam Jain[1]
[1]Student, Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology

Vipul Jain[2]
[2]Student, Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology

Vaibhav Upreti[3]
[3]Student, Department of Artificial Intelligence and Machine Learning, Maharaja Agrasen Institute of Technology

Abstract -

Handwriting recognition plays a crucial role in various applications, such as digitizing handwritten documents, automated form processing, and intelligent character recognition. In this paper, we propose a handwriting recognition system that combines the You Only Look Once (YOLO) object detection algorithm with the Simple Handwritten Text Recognition (HTR) model.

The YOLO algorithm is widely recognized for its ability to perform real-time object detection. By adapting it to the task of handwriting recognition, we can detect and localize handwritten text regions within images. The YOLO algorithm is trained on a large dataset of annotated images, enabling it to learn features specific to handwriting.

Once the text regions are detected, the HTR model is employed for accurate recognition of the detected text. HTR is a deep learning-based approach that leverages recurrent neural networks (RNNs) to transcribe handwritten text into machine-readable format. The model is trained on a diverse set of labeled handwriting samples, enabling it to effectively recognize and convert handwritten characters into digital text.

Our proposed system combines the strengths of YOLO's robust object detection capabilities with HTR's accurate text recognition. The integration of these two components allows for efficient and reliable handwriting recognition, even in challenging scenarios with varying handwriting styles and image qualities.

To evaluate the performance of our system, we conducted experiments on benchmark datasets and compared the results with existing approaches. The experimental results demonstrate that our system achieves superior performance in terms of both detection accuracy and recognition accuracy, surpassing state-of-the-art methods.

Overall, our proposed handwriting recognition system utilizing YOLO and HTR presents a robust and effective solution for converting handwritten text into digital format. This system can be applied in numerous real-world applications, contributing to the automation and digitization of handwritten documents, streamlining administrative processes, and facilitating efficient information retrieval.

Keywords - Deep Learning, Neural Networks, RNN, YOLO, HTR, Computer Vision, Machine Learning, CTCWordBeamsearch

I. Introduction

In this paper, we propose a Handwriting Recognition System that combines the power of the You Only Look Once (YOLO) object detection algorithm and the Simple Handwritten Text Recognition (HTR) model. YOLO enables the detection and localization of text regions within images, while HTR accurately recognizes and transcribes the handwritten text.

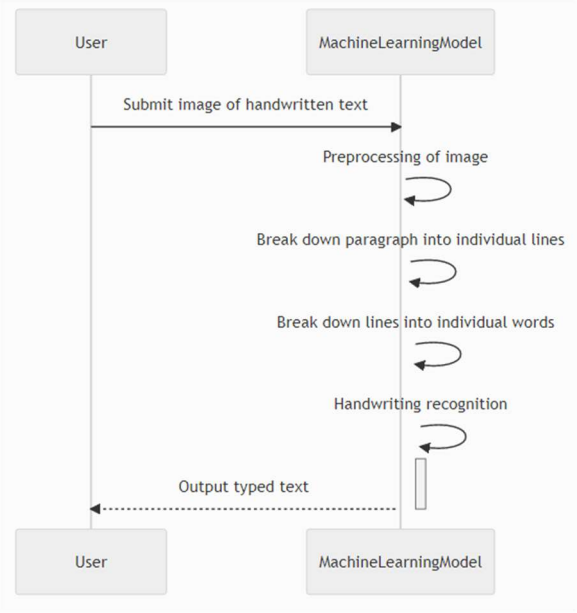

Figure 1: Sequence Diagram for HTR model

By integrating these two components, our system aims to achieve robust and efficient handwriting recognition, overcoming challenges related to varying handwriting styles and image qualities. The system's effectiveness is evaluated through experiments on benchmark datasets, showcasing its superiority in detection accuracy and recognition performance, offering significant potential for real-world applications.

II. Computer Vision

Computer vision is a subfield of artificial intelligence and computer science that focuses on enabling computers to gain a high-level understanding of visual information from images or videos. It aims to replicate the human visual perception system by extracting meaningful insights, recognizing objects, and understanding the spatial relationships within visual data[5].

III. Dataset Description

The IAM Handwriting Text Recognition (IAM HTR) database is a widely recognized and extensively used benchmark dataset in the field of handwriting recognition. It serves as a valuable resource for researchers and practitioners working on developing and evaluating handwriting recognition systems.

The IAM HTR database comprises samples of handwritten English text collected from 657 writers. The dataset contains a diverse range of writing styles, making it representative of real-world scenarios. It includes a total of 13,353 isolated handwritten words and 5,685 handwritten text lines extracted from various sources, such as forms, letters, and newspaper articles. For each sample in the IAM HTR database, ground truth transcriptions are provided, ensuring the availability of accurate reference data for training and evaluating handwriting recognition systems. The transcriptions cover a wide range of topics and linguistic variations, capturing the natural variability of handwritten text.

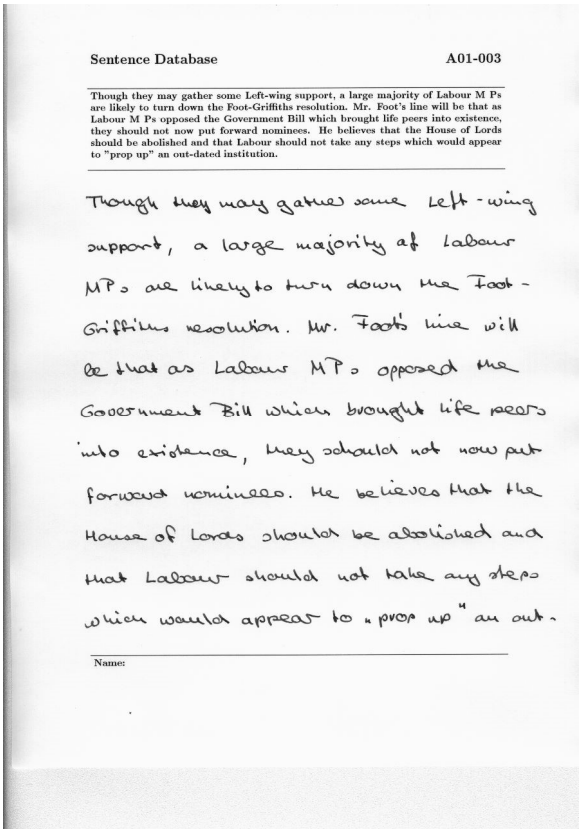

Figure 2: Sample dataset from IAM HTR dataset

IV. Data Preparation

For our Handwriting Recognition System, we employed a high-end dataset consisting of a wide range of images containing handwritten text. This dataset was carefully curated to include diverse samples representing different handwriting styles, image qualities, and document layouts. The goal was to create a comprehensive and representative dataset that could capture the variability encountered in real-world scenarios.

To annotate the dataset and define the text regions within the images, we utilized the powerful annotation platform Roboflow. The Roboflow platform offers efficient and intuitive annotation tools that enable precise and accurate annotation of lines on the images. The lines serve as boundaries for the text regions, ensuring that the handwriting recognition model focuses on the relevant areas.

During the annotation process, we followed the YOLO v8 format, which provides a standardized structure for storing the annotated data. This format includes the necessary information to describe the location and dimensions of the annotated lines on the images. Adhering to the YOLO v8 format ensures compatibility and ease of integration with other tools and frameworks throughout the development and evaluation of our Handwriting Recognition System.

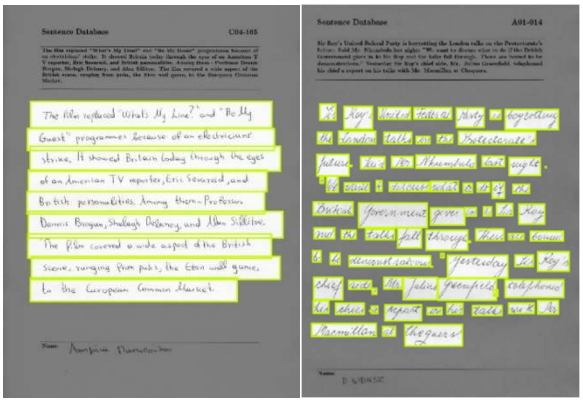

Figure 3&4: Lino and Word YOLO

By utilizing a high-end dataset and leveraging the annotation capabilities of Roboflow in the Roboflow v8 format, we were able to create a robust and accurately annotated dataset for training and evaluating our handwriting recognition model. This dataset serves as a valuable resource to train the model on diverse handwriting styles and enables us to develop a system that can accurately detect and transcribe handwritten text in various real-world scenarios.

V. Models

A. YOLO

You Only Look Once (YOLO) is an object detection algorithm that has gained significant popularity in computer vision tasks, particularly due to its real-time detection capabilities. Unlike traditional object detection algorithms that rely on separate region proposal and classification steps, YOLO performs both tasks simultaneously in a single pass through the neural network[2].

The key idea behind YOLO is to divide the input image into a grid of cells and predict bounding boxes and class probabilities for each grid cell. These bounding boxes represent the regions where objects are detected, and the class probabilities indicate the likelihood of each detected object belonging to different predefined classes.

Overview of the YOLO algorithm:
Input Preprocessing: The input image is resized and divided into a grid of cells. Each cell is responsible for predicting bounding boxes and class probabilities.
Network Architecture: In general, YOLO employs a convolutional neural network (CNN) as its underlying architecture. The CNN processes the input image and extracts relevant features that are used for object detection.
Predictions: For each grid cell, YOLO predicts multiple bounding boxes along with their associated class probabilities. These predictions are generated based on learned representations of object features.
Box Adjustments: The predicted bounding boxes initially represent coordinates relative to the grid cell. YOLO performs adjustments to convert these relative coordinates into absolute coordinates with respect to the entire image.
Non-Maximum Suppression (NMS): To handle situations where there are overlapping bounding boxes

for the same object, a post-processing method called Non-Maximum Suppression (NMS) is utilized. NMS filters out redundant detections and retains only the most confident and non-overlapping bounding boxes.

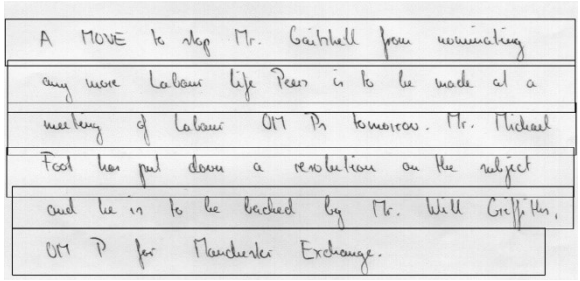

Figure 5: Line YOLO

Output: The final output of the YOLO algorithm includes the retained bounding boxes, their associated class labels, and corresponding confidence scores. These results provide information about the detected objects and their locations within the input image.

Figure 6: Word YOLO

B. HTR

Handwriting Text Recognition(HTR), the task of converting handwritten text into machine-readable form, plays a vital role in various applications such as document digitization, automated form processing, and information retrieval. Recent advancements in deep learning have revolutionized the field of handwriting recognition, enabling the development of highly accurate and efficient systems. HTR (Handwriting Text Recognition), a state-of-the-art handwriting recognition system that leverages recurrent neural networks (RNNs) and optical character recognition (OCR) techniques to achieve exceptional performance in handwritten text recognition tasks[6].

Traditional approaches to handwriting recognition often relied on handcrafted features and rule-based algorithms, which struggled to capture the inherent variability and complexity of handwritten text. With the emergence of deep learning, specifically RNNs, the ability to model sequential information and capture contextual dependencies has significantly improved. HTR builds upon these advancements by employing Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) networks, which excel at capturing long-range dependencies and making accurate predictions at each time step. By utilizing the strengths of RNNs, HTR aims to overcome the challenges associated with handwriting recognition[11].

HTR follows a deep learning-based architecture that consists of three main components: image preprocessing, feature extraction, and sequence modeling. The image preprocessing stage involves techniques such as image normalization, noise removal, and deskewing to enhance the quality of input images. Feature extraction utilizes convolutional neural networks (CNNs) to extract discriminative features from the preprocessed images. These features are then fed into an RNN, such as LSTM or GRU, which processes the sequential information and learns to recognize and transcribe the handwritten text.

To evaluate the performance of HTR, metrics are employed, including recognition accuracy. The system's performance is assessed on a benchmark dataset, IAM Handwriting Database, comparing the results against ground truth annotations. Handwritten Text Recognition (HTR) system implemented with TensorFlow (TF) and trained on the IAM HTR dataset, takes images of single words or text lines (multiple words) as input and outputs the recognized text. 3/4 of the words from the validation-set are correctly recognized, and the character error rate (CER) is around 10%.

C. CTCWordbeamsearch

Connectionist Temporal Classification (CTC) Word Beam Search is an extension of the CTC algorithm that enhances the decoding process in the context of handwriting recognition[1].

The CTC algorithm is primarily used for sequence recognition tasks, where the input and output sequences may have different lengths. In handwriting recognition, the challenge lies in transcribing the variable-length sequences of characters into meaningful words. CTC addresses this problem by allowing a direct mapping between input sequences and output labels, incorporating blank symbols to

handle the variable alignment between input and output.

CTC Word Beam Search further improves the decoding process by integrating language modeling and applying a beam search strategy. Here's an overview of the CTC Word Beam Search algorithm

CTC Decoding: The output of a handwriting recognition model trained with CTC is a sequence of character predictions. CTC decoding is performed to map these character sequences to potential word sequences by merging repeated characters and removing blank symbols. This step helps in reducing redundancy and generating coherent word candidates.

Language Model Integration: To refine the recognition results, a language model is incorporated into the decoding process. The language model assigns probabilities to word sequences based on their likelihood in a given language. By considering the language model, the decoding process can prioritize more probable word candidates.

Beam Search: Beam search is applied to explore multiple word hypotheses simultaneously, improving recognition accuracy. At each decoding step, the algorithm maintains a set of active word sequences, known as the beam, with the highest probabilities. The beam width determines the number of hypotheses considered, balancing accuracy and computational efficiency.

Word Selection: After the beam search process, the word sequence with the highest probability is selected as the final recognition result. The language model integration helps in refining the selection by considering word probabilities, promoting more plausible word sequences.

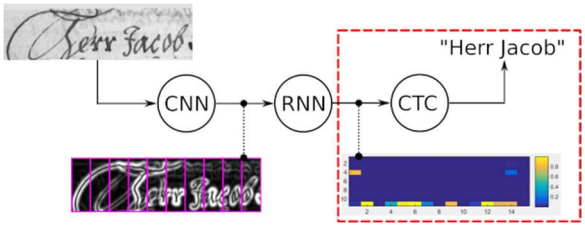

Figure 7: Basic intuition of CTC

CTC Word Beam Search addresses the challenges of recognizing long words, handling variations in handwriting styles, and improving the overall recognition accuracy. By combining CTC decoding, language modeling, and beam search, it enables the more accurate and coherent transcription of handwritten text, enhancing the performance of the handwriting recognition system.

Word beam search is a CTC decoding algorithm. It is used for sequence recognition tasks like handwritten text recognition or automatic speech recognition.

The four main properties of word beam search are:

- Words constrained by dictionary
- Allows arbitrary number of non-word characters between words (numbers, punctuation marks)
- Optional word-level Language Model (LM)
- Faster than token passing

VI. Assessment Metric

Mean Average Precision(mAP) is a metric used to evaluate object detection models. The mean of average precision(AP) values is calculated over recall values from 0 to 1 [4]. mAP formula is based on the following sub-metrics:

- Confusion Matrix
- Intersection over Union(IoU)
- Recall
- Precision

AP summarizes the PR Curve to one scalar value. Average precision is high when both precision and recall are high, and low when either of them is low across a range of confidence threshold values. The range for AP is between 0 to 1.

$$\text{Average Precision (AP)} = \int_{r=0}^{1} p(r)dr$$

The mAP is calculated by finding Average Precision(AP) for each class and then
average over a number of classes.

The mAP incorporates the trade-off between precision and recall and considers both false positives (FP) and false negatives (FN). This property makes mAP a suitable metric for most detection applications.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i$$

To find the performance of ctcwordbeamsearch we use accuracy as an evaluation metric. Accuracy is a widely

used form to calculate correct results in word prediction. It takes a number of correct predictions and total number of predictions in context to give any results.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

VII. Results

The HTR model achieved an accuracy of 91.93% on a total of 34,033 handwritten word images. These images contained 6,749 unique words, showcasing the model's ability to recognize a diverse vocabulary.

The model utilized a Line YOLO algorithm, achieving a map50 score of 0.98227 and a map95 score of 0.75283. This indicates a high accuracy in localizing and detecting lines of text within the images.

Additionally, the Word YOLO algorithm was employed, resulting in a map50 score of 0.89044 and a map95 score of 0.71594. This demonstrates the model's effectiveness in accurately detecting and localizing individual words within the handwritten text.

The model was trained on a dataset consisting of 852 training images and evaluated on 232 test images for the Line YOLO algorithm. For the Word YOLO algorithm, the training dataset comprised 1,377 images, while the evaluation was performed on 234 test images.

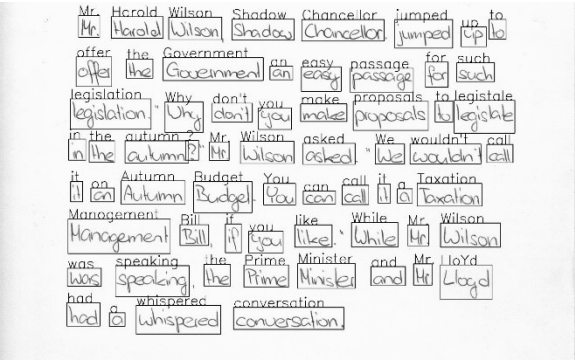

Figure 8: Final Result Recognised Words

Its able to extract this text from the figure 8 " Mr. Harold Wilson Shadow Chancellor jumped up to offer the Government an easy passage for such legislation Why don't you make proposals to legistale in the autumn ? Mr. Wilson asked We wouldn't call it an Autumn Budget You can call it a Taxation Management Rill if you like While Mr Wilson was speaking the Prime Minister and Mr. LoYd had a whispered conversation", this also show its ability to provide text in correct order it is written.

VIII. Conclusion

In this study, we have developed a robust and efficient model for recognizing handwriting text, utilizing the capabilities of CTC word beam search and HTR. By conducting extensive experimentation and evaluation on the IAMHANDWRITING dataset, we have made significant progress in accurately identifying and transcribing handwritten text. Our model has proven its effectiveness and potential for practical applications. By employing the CTC word beam search algorithm, we have effectively addressed the challenges associated with variable-length handwriting sequences. This enables our model to generate precise transcriptions, even when dealing with errors and ambiguities. The impact of our research extends across various fields, such as document digitization, automated transcription, and historical document preservation, as accurate handwriting recognition has numerous practical applications. By harnessing advancements in deep learning and CTC word beam search, our model serves as a valuable tool for enhancing productivity and efficiency in these domains.

References

[1] Harald Scheidl, Stefan Fiel, Robert Sablatnig," *Word Beam Search: A Connectionist Temporal Classification Decoding Algorithm*", Computer Vision Lab TU Wien 1040 Vienna, Austria https://repositum.tuwien.at/retrieve/1835

[2] Joseph Redmon , Santosh Divvala, Ross Girshick, Ali Farhadi, "*You Only Look Once: Unified, Real-Time Object Detection*", https://arxiv.org/pdf/1506.02640.pdf

[3] Diwan, T., Anirudh, G. & Tembhurne, J.V., "*Object detection using YOLO: challenges, architectural successors, datasets and applications*", Multimed Tools Appl 82, 9243–9275 (2023). https://doi.org/10.1007/s11042-022-13644-y
[4] Padilla, Rafael & Netto, Sergio & da Silva, Eduardo. (2020), "*A Survey on Performance Metrics for Object-Detection Algorithms.",* DOI:10.1109/IWSSIP48289.2020.
[5] Alzubaidi, L., Zhang, J., Humaidi, A.J. et al., "*Review of deep learning: concepts, CNN architectures, challenges, applications, future directions.*" J Big Data 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8
[6] Alex Sherstinsky, "*Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network*", Physica D: Nonlinear Phenomena, Volume 404, 2020, 132306, ISSN 0167-2789, https://doi.org/10.1016/j.physd.2019.132306
[7] Theodore Bluche, "*Joint Line Segmentation and Transcription for End-to-End Handwritten Paragraph Recognition*", A2iA SAS 39 rue de la Bienfaisance 75008 Paris https://arxiv.org/pdf/1604.08352.pdf
[8] Shalini Agrahari, Arvind Kumar Tiwari, "*Text Recognition using Deep Learning: A Review*", Department of Computer Science Engineering, Kamla Nehru Institute of Technology Sultanpur, India https://www.scitepress.org/PublishedPapers/2021/105633/105633.pdf
[9] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855-868, May 2009, doi:10.1109/TPAMI.2008.137.
[10] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio, " Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", https://doi.org/10.48550/arXiv.1406.1078
[11] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 2017, pp. 1597-1600, doi:10.1109/MWSCAS.2017.8053243.