Application of OCR to Design an Intelligent PDF Parser for Technical Documentation

Rajath Rao T N
Department of Electronics and Communication
RV College of Engineering
Bengaluru, India

S Praveen
Department of Electronics and Communication
RV College of Engineering
Bengaluru, India

***Abstract*— The recent years have seen a rise in the importance of text detection and identification. This trend is the result of developments in the fields of computer vision and machine learning, as well as a rise in the number of applications based on text detection and identification. The basic objective of text recognition, which is the basis for a wide range of applications, is to locate any text in an image and, if any, to detect, localize, and recognize it. This project aims at developing numerous methods to detect and recognize text in images using python programming language. The objectives of the project are to convert the scanned pages of the technical documents to images and perform various Optical Character Recognition (OCR) techniques on those images and store the images in a database and also to search and locate a keyword from the database and return the file structure. Hence a robust architecture is required to extract the text from the image. Results demonstrate the Character Error Rate and the Word Error Rate as the evaluation metric of the image to text conversion methods.**

Keywords—OCR; Text Recognition;Python;Segmentation

I. INTRODUCTION

Research in the field of text recognition in images attempts to create software that can automatically extract text from images and save it in the necessary formats. The need for software solutions that can identify characters in computer systems when information is scanned from paper documents is expanding today. There is a large demand for saving the data found in these paper documents so that it may later be used again through the search process.

Modern society is increasingly recognizing the need for digitization. Modern tools like artificial intelligence and natural language processing make it simpler to organize and analyze digital data for a variety of applications. Document processing is the idea of storing the information found in paper documents, reading it, and conducting searches. When processing a document, OCR methods are used to find the necessary text and print it in the appropriate format [1].

OCR, also known as optical character recognition, is the electronic or mechanical conversion of images containing printed text, handwritten text, or typed text into machine-encoded text, either from a scanned document, a scene photo (such as the text on signs and billboards in a landscape photo), a photo of a document, or from subtitle text superimposed on an image. Using OCR software, a computer can read text from static photos and turn it into editable, searchable data.

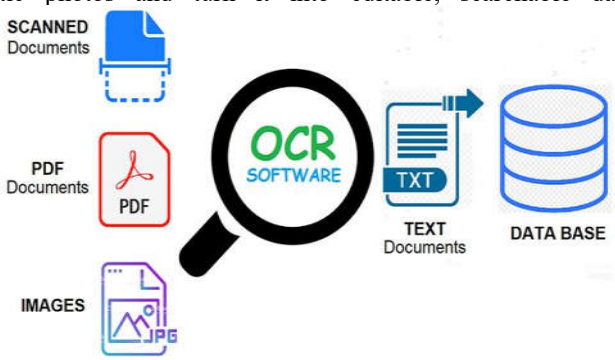

Fig. 1. Optical Character Recognition Process [2].

The various OCR software approaches can be applied to scanned documents with typed, printed, or handwritten text, PDF documents, or any images with text in them to produce the text version from the aforementioned document types. These text documents may be kept in a database and accessed later or used in other activities.

It is time-consuming to manually sift through millions of pages of scanned technical publications. Therefore, a more reliable technique of storing and retrieving the entirety of the technical documents' contents from a protected database is required. Additionally, it is necessary to give clients quick and simple access to the database-stored documents through keyword searches. This helps the teams to effectively utilize their resources and maximize their use of time, cutting costs.

This project aims at developing numerous methods to detect and recognize text in images. The objectives of the project are to convert the scanned pages of the technical documents to images and perform various Optical Character Recognition (OCR) techniques on those images and to search and locate a keyword in the given file structure. In order to achieve the mentioned objectives, three different OCR methodologies are applied using python programming language. Firstly, OCR is performed using the Pytesseract package in python. Second method used easyocr package to perform OCR of the images. Lastly, a custom OCR model is built and trained using tensorflow library in python. The results of the OCR are used in developing the Flask endpoints for addition of documents and locating keywords.

II. FUNDAMENTALS OF OPTICAL CHARACTER RECOGNITION

The OCR is the method of extracting text from the images. There are various open source pretrained models available like tesseract, by Google and easyocr, by Jaided AI.

A. Tesseract

Tesseract is a free and open source engine for text recognition (OCR) developed by Google. It can be used directly to extract text from the images. Tesseract extracts printed and typed texts and also handwritten text from the images. The generic flow of OCR process to build an Application Programming Interface (API) using the Tesseract-OCR engine is shown in the figure 2.

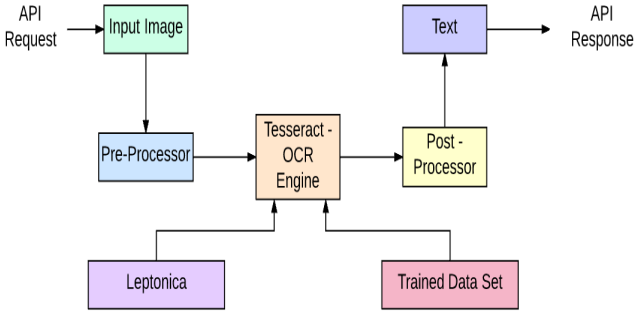

Fig. 2. OCR process flow to build API using tesseract [6].

Python-tesseract is a OCR tool to be used with the python programming language. It is used to read the texts that are embedded in the images. Python-tesseract, which is also called as pytesseract is a wrapper for Google's Tesseract OCR engine. It works as a standalone script for tesseract. Pytesseract module can read all types of images that are supported by Pillow and Leptonica imaging libraries. These image types include jpeg, gif, png, bmp, jpg, tiff and other types. Pytesseract can also be used to print the text directly through the python script instead of writing it to a text file.

1) LSTM:

The Long Short-Term Memory (LSTM) is a variant of Recurrent Neural Networks (RNN). RNN is a tool used to model sequential data, but this method fails when addressing the model for large scale data. RNN are computationally expensive compared to LSTM and the RNN suffers issues such as the vanishing gradient and the exploding gradient problems while training. Hence RNN cannot be used to learn long sequences if data very well. Hence, to solve these issues of RNN, LSTM was introduced. LSTM maintains a strong gradient over various time steps. Therefore it can be used to train with long sequences of data. The figure 3 gives the visual representation of the LSTM cell.

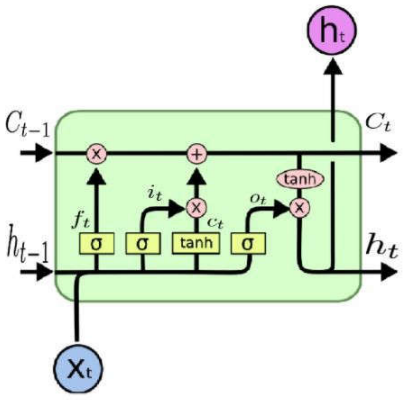

Fig. 3. Visual representation of the LSTM cell.

B. EasyOCR

EasyOCR is a python package that allows Optical Character Recognition (OCR) to be performed on images with minimal effort. EasyOCR is one of the most straight forward OCR techniques available at the moment. Configuring of easyocr is simple as it needs minimal dependency support to configure the OCR environment. EasyOCR is created and is maintained by a company called Jaided AI, which specialises in providing OCR services.EasyOCR uses PyTorch library in python. EasyOCR supports over eighty languages. The EasyOCR pipeline is as shown in the figure 4.

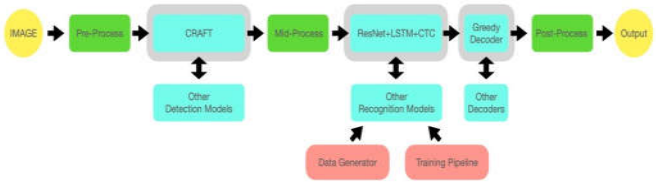

Fig. 4. EasyOCR Framework

All the deep learning mechanisms use PyTorch for the execution. Text detection execution uses the Character-Region Awareness For Text detection (CRAFT) algorithm and also the pretrained model specified in it. The recognition model is a Convolutional Recurrent Neural Networks (CRNN). It is composed of 3 main components: feature extraction (Resnet is used here) and Visual Geometry Group (VGG), LSTM is used sequence labeling and Connectionist Temporal Classification (CTC) is used for decoding.

III. DESIGN OF OPTICAL CHARACTER RECOGNITION MODELS

A. Dataset Collection

Before designing the model, it is important to have a good dataset to train the model on. Open source datasets are selected, for both printed and handwritten text.

The GoodNotes Handwriting Kollection (GNHK) is a dataset containing images with English handwritten text [13]. The dataset consists of camera captured, scanned images of English handwritten text written by people from various regions of the world. The dataset is prepared such that the users

can make use of the dataset to create models and explore on different localization and text recognition methodologies.

The Form Understanding in Noisy Scanned Documents (FUNSD) is a dataset containing images with English printed and typed text. The dataset consists of 199 scanned images of forms with full annotation [3]. The dataset can be used to perform Optical Character Recognition (OCR), text detection, spatial layout analysis and many more applications. The dataset offers a wide variety of images. The dataset contains noisy images and images with varying levels of blurring. The images also vary widely in appearance, i.e the positions of the text in the images.

B. Design Methodology

The process involves reading the PDF and converting each of the pages in the PDF to image. The images are preprocessed suitably and segmentation is done on the images to find the bounding box around the text. The image is cropped around the bounding box and is used as input to the DL model and the pytesseract and easyocr modules. The extracted text is added to the text file to store. The above process is explained in the figure 5.

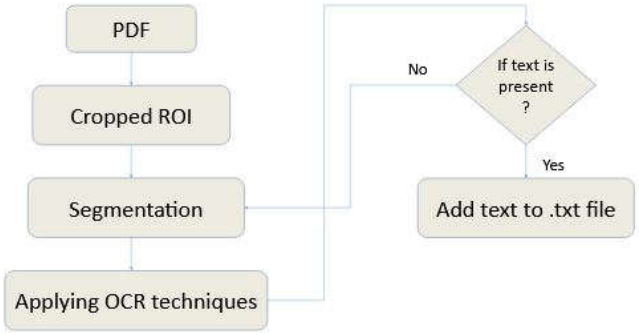

Fig. 5. Flowchart of the design

The pages of the scanned PDF file is converted to images using the PyMuPDF library in python. Each page is stored as an image. The images are preprocessed to be able to locate the text present with good accuracy using opencv, an image processing library in python. The region of interest (i.e location of the text) is marked in the images and segmentation is done on the image using opencv library. Using OCR techniques like tesseract, the characters identified and stored in a text file. The identified characters are used to make the PDF searchable and the output text file is obtained.

IV. IMPLEMENTATION OF THE MODELS

Effective design and implementation requires experimentation and improvement over past built models. Various designs are considered and compared. This chapter includes the design process and implementation of the proposed design using python language.

Python language is used throughout the project for data analysis, feature extraction and Deep Learning model implementation. Python provides robust packages and libraries that can be used in the process of building Deep Learning models. The following libraries are used in the course of this project.:

1. Numpy - This library supports the use of large and multidimensional arrays and matrices along with operations between them.

2. Matplotlib - This library provides data visualization through graphs and other plotting methods.

3. Pandas - This library provides functionalities to read and process excel and csv files.

4. OpenCV - This is the computer vision library in python. This library provides a range of functionalities that can be performed on the images for processing and analysis operations.

5. Tensorflow - This is the python library for Machine Learning. It provides a range of functionalities that can be used in developing and training Deep Learning models.

6. Keras - Keras is a high level API of tensorflow. This library is used for developing and evaluating Deep Learning models.

A. Preprocessing of the images

The image preprocessing is required to clean the images in the dataset for the input of the model. Preproceesing will lead to reduction in the model training time and also an increase in the speed of execution for predictions on the images. Image preprocessing is done by using the opencv library in python. The following functions were performed.

1. Grayscaling: Grayscaling is the process of converting the image from various color spaces like RGB to a shade of gray [8]. This is done by the numeric average of the values of the pixel in each color space. This process helps in reducing the dimension of the image and also reduces the model complexity.

2. Gaussian blurring: Gaussian blurring is the process of smoothing the image. This process is used to remove gaussian noise from the image.

3. Adaptive threshold: Adaptive threshold is the process where the threshold value is calculated for a small region. If the pixel value is above the threshold value, the pixel value is set to the maximum and if the pixel value is below the threshold, then the pixel value is set to its minimum value. This process is also called as binarization [9].

4. Morphological operations: The operations that are performed to process images based on shape are called morphological transforms. A structural element is added to the image and the output is obtained. Erosion and dilation are the basic morphological operations.

B. *Segmentation of images*

Segmentation is the process of selecting a part of the image that is required for further operations. As the images read from the PDF is of larger size, it is computationally expensive to train the data models on such large images. Hence dividing the image into smaller parts will be of importance. If the page contains a lot of empty spaces, training the model to ignore the empty spaces will result in a very inefficient algorithm with a computationally intensive architecture, which will also slow down the training of the model. Hence a good segmentation method is required. In this paper, two methods of segmentation [7], one by using the easyocr module and the other using the opencv imaging library in python.

C. *Models*

The text is extracted from the preprocessed and segmented images. Two methods are followed in this project. The first method is by using the Pytesseract library in python. The path to the executable file of tesseract has to be assigned in order to use the module. The preprocessed image is read using the opencv library and the image to string method of the pytesseract module is invoked on the image. The inputs to be provided for the method are the image and also the list of languages. The required languages are added to the method. The results are predicted based on the pretrained weights. The model is trained by using the LSTM network in combination with the CTCdecoder to obtain the results.

The second method used to perform OCR is by making use of the EasyOCR library in python. This module is written in python using pytorch. The module contains pretrained weights for over 80 languages. The pretrained model weights for the required languages are used to predict the text from the image. The Reader class is invoked to load the models for the required languages. The output text can be obtained by calling the loaded weights on the image.

The third method is to build and train the model using the datasets mentioned in the above section. The layers of the model architecture are shown below. The model is trained for over 35000 segmented images.

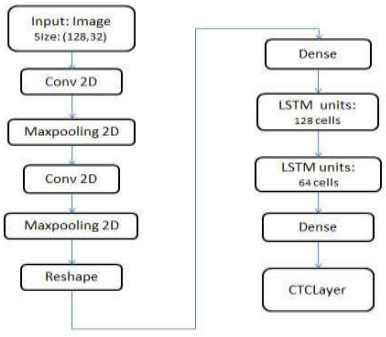

Fig. 6. Flowchart of the design

V. RESULTS

The entire dataset was divided into 80% training set and 20% validation set. The model was trained for 50 epochs. The Deep Learning model learns the features of the data using back propagation of the error. To arrive at the lowest possible error it employs different gradient descent algorithms. In this project Adam is used as the optimizer as it can vary its descent based on training.

The evaluation metric used in calculation the error rate and the accuracy are Character Error Rate (CER) and Word Error Rate (WER). The CER calculation is based on the Levenshtein distance. The error rates determine the extent to which the text extracted from the image by following the OCR methodology differs from that of the annotation of the same image. Levenshtein distance gives the minimum number of word differences, which can be insertion, deletion and substitution that is required to change one word or a sentence to other.

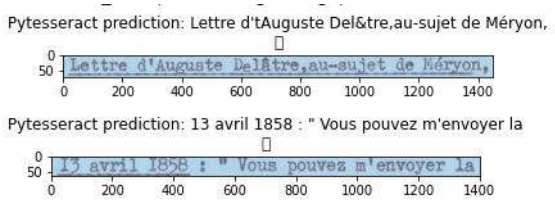

Fig. 7. Example prediction of Pytesseract.

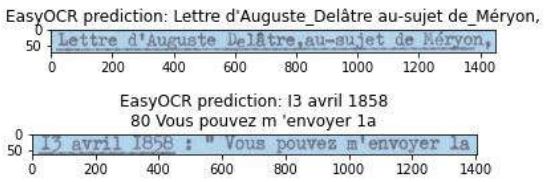

Fig. 8. Example prediction of EasyOCR.

Fig. 9. Example prediction of the DL model.

The models are evaluated by using the evaluation metric of Character Error Rate (CER) and Word Error Rate (WER). The CER and WER are the parameters used in determining which of the methodologies gives the most accurate result. The CER and WER for pytesseract is found to be 37.82% and 55.37% respectively. The CER and WER for EasyOCR is found to be 32.61% and 66.74% respectively. The CER and WER for deep learning is found to be 11.82% and 16.72% respectively. Therefore, an improvement in the performance of the OCR methodology is observed.

VI. CONCLUSION

Text detection and recognition has been one of the most important problems in the deep learning domain in recent

years. This project aims at developing numerous methods to detect and recognize text in images. Various methodologies are implemented in order to achieve the required result. This technology is part of the project to find the searched keyword in each of the added PDFs and also find the pages in which the text is present. The objectives of the paper are to convert the scanned pages of the technical documents to images and perform various Optical Character Recognition (OCR) or text extraction techniques on those images to recognize and extract the text and to save the extracted text in the required format. The paper provides a tool to enable the user to search for a keyword from any of the added PDFs and locate the keyword in the given file structure. Finally, the tables in the images is extracted and saved in the CSV format, as the values saved in the CSV file can be used to perform various operations.

REFERENCES

[1] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR)," IEEE Access, vol.8, pp. 142 642-142 668, 2020. DOI: 10.1119/access.2020.3012542.

[2] T. F. Yong, S. Azad, M. M. Rahman, K. Z. Zamli, and G. Rabby, "A highly accurate PDF-to-text conversion system for academic papers using natural language processing approach," Advanced Science Letters, vol.24, no. 10,pp. 7844-7849, Oct. 2018. DOI: 10.1166/asl.2018.13029.

[3] G. Jaume, H. K. Ekenel, and J.-P. Thiran, "FUNSD: A dataset for form understanding in noisy scanned documents," in 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), IEEE, Sep. 2019. DOI: 10.1109/icdarw.2019.10029.

[4] P. He, W. Huang, Y. Qiao, C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, no. 1, Mar. 2016. DOI: 10.1609/aaai.v30i1.10465.

[5] V. Yadav and N. Ragot, "Text extraction in document images: Highlight on using corner points," in 2016 12th IAPR Workshop on Document Analysis Systems (DAS), IEEE, Apr. 2016. DOI: 10.1109/das.2016.67.

[6] P. M. Manwatkar and S. H. Yadav, "Text recognition from images," in 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE, Mar. 2015. DOI: 10.1109/iciiecs.2015.7193210.

[7] R. Kagalkar, "A review on conversion of image to text as well as speech using edge detection and image segmentation," Nov. 2014.

[8] K. Ntirogiannis, B. Gatos, and I. Pratikakis, "Performance evaluation methodology for historical document image binarization," IEEE Transactions on Image Processing, vol. 22, no. 2, pp. 595-609, Feb. 2013. DOI: 10.1109/tip.2012.2219550.

[9] J. Yang, K. Wang, J. Li, J. Jiao, and J. Xu, "A fast adaptive binarization method for complex scene images," in 2012 19th IEEE International Conference on Image Processing, IEEE, Sep. 2012. DOI: 10.1109/icip.2012.6467253.

[10] S. Malakar, S. Halder, R. Sarkar, N. Das, S. Basu, and M. Nasipuri, "Text line extraction from handwritten document pages using spiral run length smearing algorithm," in 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS), IEEE, Dec. 2012. DOI: 10.1109/codis.2012.6422278.

[11] S C. Ramakrishnan, A. Patnia, E. Hovy, and G. A. Burns, "Layout-aware text extraction from full-text PDF of scientific articles," Source Code for Biology and Medicine, vol. 7, no. 1, May 2012. DOI: 10.1186/1751-0473-7-7.

[12] A. W. C. Lee, J. Chung, and M. Lee, "Gnhk: A dataset for english handwriting in the wild," in International Conference of Document Analysis and Recognition (ICDAR), 2021.