Review-2 Report

Improving Text Extraction Accuracy with Image Preprocessing

By,

KOMALI BEERAM (160118733067)

SOUMYA VEMURI (160118733081)

Department of Computer Science and Engineering,

Chaitanya Bharathi Institute of Technology (Autonomous),

(Affiliated to Osmania University, Hyderabad)

Hyderabad, TELANGANA (INDIA) –500 075

[2021-2022]

ABSTRACT

Digitization allows us to immortalize a physical entity by creating a digital representation of it on our devices. It saves us time in manually sifting through physical storage units such as albums and notebooks and provides us with programs to manage and secure our data. We often take images of Receipts or Invoices, Identity Cards, and nutritional labels to save a copy of their details. This can be taken a step further by automating the process of information extraction and documentation.

Advancements in computer vision have provided us with the expertise to create tools for text detection and extraction. But it is still an ongoing challenge because documents with unstructured layouts, poor image quality, and noise around the text yield very low accuracy in text extraction results. Conquering this challenge would require the image to be highly enhanced through pre-processing techniques such as Brightness Correction, Contour Detection, Skewness Correction, Morphology, and Binarization. A mechanism made from the best combination of image pre-processing techniques prior to text extraction can improve text accuracy to a large extent.

Table of Contents

1. INTRODUCTION

In today's day and age, an increase in demand for digitization has fueled a massive growth in technology and communication and the use of printed materials such as books and papers has significantly reduced. Also, it is easier to organize digitized data and analyze them for various purposed with many advanced techniques like artificial intelligence etc. To translate physical and handwritten documents into digital copies, optical character recognition (OCR) has come into the sight of researchers and since its first advent it has undergone significant changes in methodology and made considerable progress towards its goal.

1.1. Problem Statement

Historical OCR engines have their accuracy lying between 70-80% for a high-quality image at page level. That means in a page of 100 words 70-80 words are accurate. This will lead to significant inaccuracies if used on a large volume of sensitive documents.

Through this project we aim to improve the accuracy of the result by implementing an ideal combination of image preprocessing techniques prior to text extraction.

1.2. Methodology

I. Document Scan and File Upload

User scans their document and uploads it to the system interface. Documents include invoices, receipts, nutrition labels, book covers etc.

II. Image Preprocessing Pipeline

Scanned documents undergo the best combination of image preprocessing techniques for maximum image text enhancement. Image Preprocessing Techniques include

a. Lighting Correction
   Lighting correction corrects light variation produced by surface relief or document curvature. When a thick book is scanned, the shadow of the binding will appear on the image. This technique allows obtaining light uniformity and eliminate such shadows, whether vertical or horizontal.
b. Contour Detection
   Contours can be explained simply as a curve joining all the continuous points along the boundary, having the same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition. For better accuracy, binary images are used.
c. Scaling of Image

Ensure that the images are scaled to the right size which usually is of at least 300 Dots Per Inch. Keeping DPI lower than 200 will give unclear and incomprehensible results while keeping the DPI above 600 will unnecessarily increase the size of the output file without improving the quality of the file. Thus, a DPI of 300 works best for this purpose.

d. Skewness Correction
   When the optical axis of the camera is not perpendicular to the text plane, Perspective distortion occurs. Text boundaries lose rectangular shapes and characters distort, decreasing the performance of recognition models trained on undistorted samples.
e. Noise Removal
   Noise can drastically reduce the overall quality of the OCR process. It can be present in the background or foreground and can result from poor scanning or the poor original quality of the data.
f. Binarization
   This step converts a multicolored image (RGB) to a black and white image. There are several algorithms to convert a color image to monochrome image, ranging from simple thresholding to more sophisticated zonal analysis.
g. Contrast Correction
   Low contrast can result in poor OCR. Increase the contrast and density before carrying out the OCR process. Increasing the contrast between the text/image and its background brings out more clarity in the output.

III. Text Extraction

The image goes through an Optical Character Recognition (OCR) engine that's been built to recognize printed text in paper documents, handwritten characters, and text elements in the image.

IV. Information Extraction

Information Extraction involves extracting meaningful information from raw text data into a structured format.

1.3. Outline of Report

For performance of recognition, we will simulate the proposed text recognition system to various characters in English language and calculate the recognition rate or accuracy which is expected to improve from the current OCR engines.

1.4. Scope of Project

The scope of our project on a grid infrastructure is to provide an efficient and enhanced software tool for the users to perform document image analysis, document processing by reading and recognizing the characters in research, academic, governmental, and business organizations that are having large pool of documented, scanned images. Irrespective of the size of documents and the type of characters in documents, the product is recognizing them, searching them, and processing them faster according to the needs of the environment.

1.5. Organization of Report

The report is divided into two parts. Each part deals with different aspects of our project. Each part has various chapters explaining in detail.

- Part 1: Introduction
  This part summarizes about the project in brief. It includes the problem statement, methodology, and outline of our results and the scope of the project.
- Part 2: Literature Survey
  This part briefly describes the overall architecture of text recognition system and provide a brief overview of the existing work carried out in the field of image preprocessing and text recognition.
- Part 3: References
  This part has a consolidated list of papers that we referred to develop the project.

Although the document may be read from front to back for a complete understanding of the project, it was written in sections and hence can be read as such. For an overview of the document and the project itself, refer to Introduction.

2. LITERATURE SURVEY

In this section we briefly describe the overall architecture of text recognition system and provide a brief overview of the existing work carried out in the field of image preprocessing and text recognition.

2.1 Introduction to the problem domain terminology

A text recognition has gained a lot of prominence in recent years as it has entered into a large arena of applications, and it is a field which is driven by the need to preserve and have access to the information containing documents in an easier and quicker way. One of the convenient ways to transfer the information from the paper or books is to scan them which convert the information into an image thus preventing reuse of the scanned information in the form of a text. One of the popular techniques used for text recognition is Optical Character Recognition. It converts scanned images of text into editable format.

The process of text recognition starts with capturing the image of the required document, preprocessing it to acquire the desired portion and then segmenting it to extract the text content present in it. The text recognition system can be divided into three modules:

A. Pre-processing Module

A document is generally scanned and converted in to the form of a picture. A picture is the combinations of picture elements which are also known as pixels. At this stage we have the data in the form of image and this image can be further analyzed so that the important information can be retrieved. So to improve quality of the input image, few operations are performed for enhancement of image such as noise removal, binarization, skew correction etc.

B. Text Recognition Module

this module can be used for text recognition in output image of pre-processing model and give output data which are in computer understandable form. Hence in this module following techniques are used.

- Segmentation

  In text recognition module, the segmentation is the most important process. It is done to make the separation between the individual characters of an image.
- Feature Extraction

  It is the process to retrieve the most important data from the raw data. To store the different features of a character, different classes are made.
- Classification

Classification is the process of identifying each character and assigning to it the correct character class, so that texts in images are converted into computer understandable form. This process used extracted feature of text image for classification i.e. input to this stage is output of the feature with stored pattern and find out best matching class for input.

C. Post-processing Module

The output of text recognition module is in the form of text data which is understandable by the computer. So there is a need to store it into a proper format such as text or word for further use such as editing or searching in that data.

2.2. Existing Solutions

Many researchers had contributed and proposed their concepts on text extraction from an image and retrieving the information. Each solution has its own pros and cons that are discussed in this subsection.

Rishabh Mittal and Anchal Garg [1] introduced and explained the concept of OCR and the process of extraction by grouping it into majorly six steps: image acquisition, pre-processing, segmentation, feature extraction, classification, and post-processing. The paper reveals that the modern ocr system's preprocessing pipeline is restricted to spatial image filtering, thresholding, noise-removal, and skew detection/correction. Improving components like Scan goals, filtered picture quality, type of printer utilized whether inkjet or laser, the nature of the paper, phonetic complexities, the lopsided brightening, and watermarks can impact the precision of OCR. Hence work can be done on improving the precision of OCR.

Sanjeev Kumar, Mahika Sharma, Kritika Handa, Rishika Jaiswal [2] proposed a novel adaptive algorithm to improve ocr accuracy with advanced image preprocessing using machine learning. Their focus was to reduce the noise of the image solely by scaling the original source image to around 300 DPI which has helped to eliminate the single biggest obstacle of the Tesseract, i.e., Tesseract's computation time of reading images with the highest character dimensions above 20 pixels. However, their algorithm does not cover images with uneven brightness, watermarks, or different fonts.

Sahana K Adyanthaya put forward a paper [3] that presents the various steps taken to recognize text from images. The steps addressed in this paper were Image Preprocessing, Segmentation, Feature Extraction and Classification. The author highlights that the noise present in an image has a major role to play in successful text recognition and that noise removal increases the probability of accurate text recognition and generates more

accurate output. The paper mentions that Gaussian filter and mean filter can be used for noise removal, that normalization should be done to ensure uniformity followed by binarization to convert the gray image into a binary image.

Naveen Sankaran and C.V Jawahar [4] proposed a neural network-based framework that operates based on BLSTM-Bidirectional Long Short-Term Memory that allows OCR to work at the word level. It leads to over 20% better results when compared to a regular OCR framework. It uses a method that does not require segmentation, that is one amongst the foremost common reasons for the error. Also, it found an over 9% decrease in character error compared to the more widely available OCR framework.

Work by S. Akopyan, O.V. Belyaeva, T.P. Plechov and D.Y. Turdakov [5] is based on a text extraction pipeline which is used to extract text from varied quality of images obtained from social media. Their work mainly focuses on dividing the input images into various classes and then preprocessing is done depending on the classes. This is followed by text recognition using the OCR engine. The dataset collected from social media is made use of in this work.

Dan Sporici, Elena Cușnir and Costin-Anton Boiangiu [6] underlined Tesseract 4.0 flaws, highly related to the segmentation procedure and proposed an adaptive image preprocessing step guided by a reinforcement learning model, which attempts to minimize the edit distance between the recognized text and the ground truth. This approach has boosted the character-level accuracy of Tesseract 4.0 from 0.134 to 0.616 and the F1 score from 0.163 to 0.729. The model adjusts samples with the purpose of maximizing the overall recognition efficiency without requiring external guidance or knowledge which has a direct benefit of including kernels, which can generate samples that might look unnatural. From a qualitative point of view, the changes are substantial yet not optimal since a reinforcement learning approach does not guarantee that local optimums will be avoided each time and hence the algorithm can get stuck on kernel configurations which will provide inferior results if not enough exploration is performed.

Dr. PL Chitra, and P Bhavani [7], in this paper have studied various images to remove unwanted noise and performed enhancement techniques such as contrast limited adaptive histogram equalization, Laplacian and Harr filtering, unsharp masking, sharpening, high boost filtering and color models then the Clustering algorithms are useful for data logically and extract pattern-analysis, grouping, decision-making, and machine-learning techniques and Segment the regions using binary, K-means and OTSU segmentation algorithm. It classifies the images with the help of SVM and K-Nearest Neighbors (KNN) Classifier to produce good results for those images.

Anupriya Shrivastava, Amudha J.Deepa Gupta and Kshitij Sharma [8] in their work have developed a system based on Convolutional Neural Network and Long ShortTerm

Memory. The developed model identifies the texts from images which are horizontal, curved or oriented style. The model has four components. The first component performs feature extraction at the low level. The second component uses a shared convolution approach to extract high level features. Irrelevant features are ignored by the third component. The fourth component predicts the character sequences.

K. Karthick, K.B. Ravindrakumar, R. Francis and S.Ilankannan [9] have discussed the various steps in text detection in detail highlighting the different techniques used for the same. They have also emphasized on handwritten text recognition which is one of the complex fields. From their study it has been found that best results can be had with reduced computation time, and it is possible to segment multilingual characters and enhance the character recognition rate.

Sai Abhishikth Ayyadevara, P N V Sai Ram Teja, Rajesh Kumar M [10],this paper deals with two different proposals of machine learning techniques. The first one was a new feature extraction technique, including the feature of three different existing feature extraction techniques. While the second one includes the analysis of the performance of three different neural networks for two different feature techniques- geometric and gradient. After doing all the survey, they concluded that the Convolutional neural network is most efficiently absorbed through the Levenberg-Marquardt algorithm.

Kukich[11] suggested using a n-gram dictionary or method based on the errors and returning the possible word to the dictionary using mathematical steps. These methods may reduce the total number of OCR errors in standard language names, but it is possible that the words may be correctly identified that are not in the dictionary of geographical names.

2.3. Related Work

This subsection of the paper deals with the efforts carried out by various developers towards the core of recognizing text from different images.

Yang Zhang and Hao Zhang HaoranLi [12] described in their paper, an information extraction pipeline used for event flyers. The major steps in the pipeline include image capture and upload, image preprocessing, text detection, OCR and NLP information extraction. The paper lists several situations where a raw image could cause inaccurate results. The OCR engine would assume the picture is taken from a perpendicular upright view, but images taken from a handheld camera could contain distortions. The illumination of the image not being uniform throughout and an image containing multiple blocks of text of different sizes and colors could also affect the output. The image preprocessing methods included were edge detection, geometric correction(transformation), and Binarization.

Lavanya Bhaskar and R Ranjit [13] discuss an event planner for the brochure images, that implements text extraction by convolution followed by MSER feature extraction and Stroke width method. The event planner then directly links the event text to the google calendar for scheduling the events. However, the algorithm is not tested for event information taken from handwritten images and complex font text present in the images.

Brijesh Kumar Y. Panchal, and Gaurang Chauhan [14] proposed an implementation on the Android Application to extract using Tesseract OCR in which the following concepts are used, which are Adaptive Thresholding, Connected Component, Fine Lines, and Recognize Word. Using this Optical Character Recognition (OCR) Technology. The Application generates text, which is printed on a clean, B/W or colorful background and then can be converted into a computer readable form ASCII. With the help of this Android Application using Tesseract OCR, the system has two ways for Text Extraction. The first one is to capture a photo while the second one uploads an image from the gallery. After that the system can proceed as per the user requirement which portion of the image they want to crop or edit. After editing the picture, it converts into the text. This Android Application is for two languages, English and Hindi.

Salvador España-Boquera, Maria J. C. B., Jorge G. M., and Francisco Z. M. [15], this paper outlines the hybrid Hidden Markov Model (HMM) is used to conceive the unconstrained offline handwritten texts. The main characteristics of the recognition systems is to produce a new way in the form of preprocessing and recognition which are both based on ANNs. The preprocessing is used to clean the images and to enhance the non-uniform slant and slope correction. Whereas the recognition is used to estimate the emission probabilities.

K.Gaurav, Bhatia P. K. [16] , this paper deals with assorted pre-processing techniques used for handwritten recognition which consists of different images starting from a simple handwritten document and extending its radius to complex background and diverse image intensities. The pre-processing techniques that were included are contrast stretching, noise removal techniques, normalization, and segmentation, binarization, morphological processing techniques. They concluded that no technique for preprocessing can single handedly be used to produce an image. All the techniques go hand in hand. Even though after applying all the said techniques, the accuracy of the image is not up to the mark.

2.4. Tools/Technologies Required

The following technologies and tools are identified to develop our proposed solution-

A. Tesseract

Tesseract is an open-source text recognition (OCR) Engine, available under the Apache 2.0 license. It can be used directly, or (for programmers)

using an API to extract printed text from images. It supports a wide variety of languages.

B. Python

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

C. OpenCV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

D. Pillow

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating, and saving images.

E. NLTK

NLTK (Natural Language Toolkit) Library is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.

3. REFERENCES

[1]. Mittal, Rishabh; Garg, Anchal, "Text extraction using OCR: A Systematic Review", 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp 357–362.

[2]. Sanjeev Kumar, Mahika Sharma, Kritika Handa, Rishika Jaiswal, "Improve OCR Accuracy with Advanced Image Preprocessing using Machine Learning with Python", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-7, May 2020.

[3]. Sahana K Adyanthaya, "Text Recognition from Images: A Study", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Volume-8 Issue-13, 2020.

[4]. Naveen Sankaran and C.V Jawahar, "Recognition of Printed Devanagari Text Using BLSTM Neural Network", 21st International Conference on Pattern Recognition (ICPR), November 11-15, 2012. Tsukuba, Japan.

[5]. M.S. Akopyan, O.V. Belyaeva, T.P. Plechov and D.Y. Turdakov, "Text recognition on images from social media", Ivannikov Memorial Workshop (IVMEM), 2019.

[6]. Dr. PL Chithra, P Bhavani, P., "A Study on Various Image Processing Techniques", International Journal of Emerging Technology and Innovative Engineering (ISSN (print): 2394 – 6598) Volume 5, Issue 5, May 2019.

[7]. Anupriya Shrivastava, Amudha J., Deepa Gupta, Kshitij Sharma, "Deep Learning Model for Text Recognition in Images", 10th ICCCNT 2019 July 6-8, 2019, IIT - Kanpur, Kanpur, India.

[8]. K. Karthick, K.B. Ravindrakumar, R. Francis, S. Ilankannan, "Steps Involved in Text Recognition and Recent Research in OCR; A Study", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1, May 2019.

[9]. Sai Abhishikth Ayyadevara, P N V Sai Ram Teja, Bharath K P Rajesh Kumar M, "Handwritten Character Recognition Using Unique Feature Extraction Technique", International Research Journal of Modernization in Engineering Technology and Science, Volume-3 Issue-10, Jan 2020.

[10]. Karen Kukich, “System that automatically converts word into text”, ACM Computing Surveys, Vol. 24, No. 4, December 1992.

[11]. Yang Zhang, Hao Zhang, HaoranLi, “Event Info Extraction from Flyers”, 2021.

[12]. Bhaskar, L., & Ranjith, R., “Robust Text Extraction in Images for Personal Event Planner”, 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) July 1-3, 2020, IIT-Kharagpur, Kharagpur, India.

[13]. Brijeshkumar Y. Panchal and Gaurang Chauhan, “Design and implementation of android application to extract text from images by using tesseract for English and Hindi”, 3rd International Scientific Conference of Engineering Sciences and Advances Technologies (IICESAT), Journal of Physics: Conference Series, Volume 1973, 4-5 June 2021.

[14]. Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., “Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.

[15]. K. Gaurav and Bhatia P. K., “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”, 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.

[16]. K. Gaurav and Bhatia P. K., “Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition”, 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.