Text Restoration Model Using OCR and Spelling Correction Algorithms

Anna Liza Ramos, Toblerone Mina, Jeremy Mangilaya
Donnalyn T. Lapasa, Elizabeth Manaig, John Ace Luiz Bagacina

Institute of Computer Studies, Saint Michael’s College of Laguna, Binan City, Philippines.

AnnaLiza.Ramos@smcl.edu.ph; t.mina@student.smcl.edu.ph; j.escueta@student.smcl.edu.ph
d.lapasa@student.smcl.edu.ph; e manaig@student.smcl.edu.ph; j.bagacina@student.smcl.edu.ph

Abstract - Printed documents are valuable assets of information and must be preserved for future reference. This study aims to restore text in different types of degradation, apply a post-processing techniques to increase accuracy and to address the limitation of OCR in misspelled words. The model consists of 500 words written in Times New Roman, Calibri, Helvetica and Aarial. Results showed wet documents achieved the highest accuracy rate among other types of degradation with a score of 99.2% while the ones written over with pens achieved 91.2% for most of the conjoined words caused by the degradation were not recognized by the model. Aarial fonts had the best results while Times New Roman acquired the lowest performance rate. In terms of spelling corrections, the model showed impressive results, correcting more than half of the misspelled words.

Keywords - optical character recognition, text restoration, spelling, document degradation

I. INTRODUCTION

Preservation of document is very important for it is becoming common that some useful contents printed in documents may be lost or degraded accidentally [1] due to aging and human manipulations. These defects are usually bleed-through inks, folding marks, ink fading, holes, spots, [2] aging of the paper, poor typesetting, low resolution/bandwidth in the case of faxed documents, and the like. [3] Not only documents, but also books and other printed materials go through the process of degradation over such spans of time [4]. The aforementioned documents might bare very important information but due to the different types of degradations, it may no longer be in readable forms [5]. Nowadays, with the application of technology, documents are being digitized [6]. It is mainly performed to bring all information into the digital world to be accessed easier through online means such as web media. The digitization of documents is obtained by scanning those pages and preserving them as images. Then, images of the text documents are processed using Image Processing techniques [7]. The text is extracted from the images to make them convert into a digital form [8]. Optical Character Recognition (OCR) is an image processing technique that can specifically perform the said task [9] and many companies are using these applications such as sorting mails, reading bank cheques, and verifying digital signatures, bills processing, automatic recognizing of plate numbers, validating passports, [10] recognizing of objects on digitalized maps, understanding of hand-written office forms [11] and used by impaired person in reading printed text [12]. However, the OCR output still have an issue on accuracy [13] [14] because document images with different types of degradation makes the process even more challenging [15] like faded inks, small image resolution, specific language requirement, image noises, [16] [17] discoloured materials and old age papers [18]. These errors lead to misspellings and linguistics errors in the output text [19][20].

A. Background of the Study

Several studies were reviewed related to the topic and found that several studies about OCR are now running on mobile phones. In fact, a certain study proposed an optical character reader for smart phones using Tesseract and Mezzofanti which achieved 82.7% accuracy when tested in only 20 characters, but the lighting and skew were hindering the text extraction [35]. Another study also utilized Tesseract as the OCR engine and it managed to recognize characters with 90% accuracy, however, it was only tested on images with under 500 characters only and not on degraded documents [14]. A separate experiment also proposed a mobile application that extracted text from degraded document images using Tesseract too as the OCR engine with Adaptive Document Image Binarization to improve the performance using eight (8) documents with different types of degradations for testing which included background stains, background noise, misaligned text, heavy prints, and broken characters. The results showed an average character accuracy of 93.17% and word accuracy of 85.82%. The best results in terms of both character and word accuracy were achieved from documents with background stains only [13]. All the previous studies did not further improve the performance of Tesseract OCR engine like the application of the post-processing technique. Furthermore, they only tested it on a few numbers of characters and not in softly degraded documents only. To address these issues, the
study aims to add a post-processing technique that corrects misspellings acquired from the image to text conversion in order to improve its accuracy. It would focus only on printed letters in the English alphabet and formal fonts – not supporting handwritten texts, especially cursive.

B. Research Objectives

1. To gather 16 documents with 4 common types of degradations and 4 common types of fonts as training datasets.

2. To apply Tesseract as the OCR engine in detecting, extracting, and recognizing the text within the degraded documents.

3. To apply Norvig’s spelling correction algorithm for the OCR post-processing.

4. To implement the model in Android OS.

C. Conceptual Framework

Fig. 1 shows the captured images of degraded documents using Samsung Galaxy J7 Core with 18 megapixels. The images were used as the training data sets in this study. These images were loaded from the device’s camera or gallery application as an input, then would be processed through optical character recognition using Tesseract. The captured texts would be extracted using Norvig’s Algorithm in correcting spelling errors. The text extracted by Tesseract would be analyzed and put into a token list. The tokens would be matched into the prototypes from the dictionary. If a token did not match, it would be considered a misspelling and resolve with the correct word. The misspelled words would be replaced by the correct words nearest to them before it being displayed in digital form.

D. Theoretical Framework

D1. Optical Character Recognition: Optical Character Recognition or OCR is one of the emerging research fields in artificial intelligence, [39] and pattern recognition] [25] It has become one of the most successful applications in the said fields. [15] It is the digital translation of printed [23] text-based images into machine readable text [11] such that the text can be edited, modified, stored and shared with ease. [48] It is getting a valuable demand [47] as it prevents
keyboarding, which is the most common way of inputting text into computers and is probably the most time-consuming operation. [35] The real value of OCR is the effort and time that it reduces [3] in manual retyping the text. [53] Before, the OCR process images in flatbed desktop scanners [37] however, the disadvantage the portability and the speed in capturing an images [41] but with the advancement of technology OCR can now be run on mobile devices. [14] [17] which is portable, [29] and easier to use [28]. The accuracy is depends on the specification of the devices being used [35].

**D2. Tesseract OCR Engine:** Tesseract is an open source OCR engine available for public use [46] and is considered the most accurate OCR engine[30]. Several studies utilized the engine and it significantly achieved impressive results, compared to OCRAD and GOCR which are OCR engines available for the public as well [60]. Most of the developer used Tesseract because of its flexibility [33]. Moreover, Tesseract was tested on 236 images captured by the prototype. It correctly identified 227 numbers and only had nine (9) errors. This gave a total accuracy rate of 96.18% for Tesseract. Most errors were recognizing “0” as a number “6” or “9.” Results for single network approach for OCR is 77.81%.

**D3. Norvig’s Spelling Correction Algorithm:** The algorithm is a spell checker proposed by Peter Norvig which is modified to include numbers within text as a spell correction candidate. [57]. It does not require complex cascading grammar rules or API calls. Moreover, this algorithm creates increasing alternations for each of the words against the large corpus of correctly spelled words (dictionary) until it finds a match. [19] which generates all possible terms using Levenshtein Distance[22] [43]. Norvig’s Spelling Correction Algorithm achieves a 90% accuracy rate at a processing speed of at least 10 words per second [31].

II. METHODOLOGY

**A. Collecting Degraded Documents**

![Types of Degradations](image.png)

(a) Page Crumple  
(b) Background Stains  
(c) Wet  
(d) Written Over

Figure 2: Types of Degradations
Sample images of papers with the most common types of degradations were gathered. The images were captured using an Android phone (Galaxy j7 Core) with 18 mega pixels in a distance of 20 cm. The images were used as training data sets. The degradations included crumpled pages, background stains, wet papers, written over words, and all that have broken characters. To replicate the degradation, the experiment printed documents with 500 words in the fonts Times New Roman, Calibri, Helvetica and Arial which are the most common fonts used in printed documents. In Sample A, the pages were crumpled and were flatten again by hand. In Sample B, the experiment stained the papers using soy sauce for the degradation. In Sample C, the papers were soaked in the water for about 30 seconds to make the ink spread on the paper and were then sprinkled with acetone to spread the ink even more. And in Sample D, pens were drawn all over the pages and some letters were re-written to cover some letters or make the letters thicker.

B. Image Pre-Processing

The black pixels were separated from the white, it was assumed that those are the characters. The outlines of the characters were then detected to separate them as blobs. These blobs were organized into a baseline to analyze their distances from each other. Tesseract then identified the group of characters that formed a word by testing their pitch and distance to one another.

C. Separation of Words

Tesseract analyzed the document image by detecting the pattern of the pixels indicated as black. If it would detect a wide area of white pixels, it would assume that it was a space in between two words.

D. Word Recognition

Every character was recognized using the polygonal approximation for extracting their features then matching it with the prototype.

E. Spelling Correction

<table>
<thead>
<tr>
<th>Key</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>The</td>
<td>1</td>
</tr>
<tr>
<td>governmet</td>
<td>2</td>
</tr>
<tr>
<td>is</td>
<td>3</td>
</tr>
<tr>
<td>in</td>
<td>4</td>
</tr>
<tr>
<td>favr</td>
<td>5</td>
</tr>
</tbody>
</table>
The study built a dictionary in the database by entering correctly spelled words in a hash table. After the text was extracted from the image, it was split up into strings (tokens containing only characters and not letters). These tokens were organized into a list without repetitions.

**F. Matching to the Dictionary**

All words from a dictionary were passed into a hash table with each word having a key. A token was fetched and passed to the hash function \( H(x) \). If the function returned a key, it means that the word existed in the hash table, hence it was properly spelled. Otherwise, it was either not a word or spelled incorrectly.

**G. Solving the Closest Word from the Misspelling**

If there would be misspelled word detected, the closest correctly spelled word from it from the dictionary would be computed to replace it by computing the Levenshtein distance of the two input strings [2].

**H. Prototype Model**

If there would be misspelled word detected, the closest correctly spelled word from it from the dictionary would be computed to replace it by computing the Levenshtein distance of the two input strings [2].
The model was built using Apache Cordova, an open source software for creating mobile applications and the language Tesseract.js in implementing the model into the Android device. The mobile application took a 9.31 MB memory with a minimum of 10 seconds in processing the image.

I. Metrics of Evaluation

In getting the accuracy of the model, the study counted all the recognized words and divide it by the total number of words in the text (500) then multiplied by 100 to get the percentage. The same formula was applied for the computation of the spelling correction’s accuracy.

No. of Recognized Words \( \frac{\text{Total No. of Words}}{\times 100} \)

III. RESULTS AND DISCUSSION

The experiment was tested on degraded documents with 500 words in font size12, which is the most commonly used font size in printed documents. The individual results of the degraded documents for optical character recognition and spelling correction were listed in the tables below.

### TABLE I. OCR RESULTS FOR PAGE CRUMPLE

<table>
<thead>
<tr>
<th>Fonts</th>
<th>No. of Recognized Words</th>
<th>No. of Misrecognized Words</th>
<th>Total Number of Words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time New Roman</td>
<td>457</td>
<td>43</td>
<td>500</td>
<td>91.4%</td>
</tr>
<tr>
<td>Arial</td>
<td>469</td>
<td>31</td>
<td>500</td>
<td>93.8%</td>
</tr>
<tr>
<td>Helvetica</td>
<td>460</td>
<td>40</td>
<td>500</td>
<td>92%</td>
</tr>
<tr>
<td>Calibri</td>
<td>464</td>
<td>36</td>
<td>500</td>
<td>92.8%</td>
</tr>
</tbody>
</table>

In the crumpled documents, among the different types of font, Arial got the highest accuracy as it has the largest characters hence, the deformity of the paper did not affect the words that much. On the other hand, Times New Roman has the least accuracy because of the appearance of the letters tend to have tails and smaller.

### TABLE II. OCR RESULTS FOR BACKGROUND STAINS

<table>
<thead>
<tr>
<th>Fonts</th>
<th>No. of Recognized Words</th>
<th>No. of Misrecognized Words</th>
<th>Total Number of Words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time New Roman</td>
<td>425</td>
<td>75</td>
<td>500</td>
<td>85%</td>
</tr>
<tr>
<td>Arial</td>
<td>494</td>
<td>6</td>
<td>500</td>
<td>98.8%</td>
</tr>
<tr>
<td>Helvetica</td>
<td>487</td>
<td>13</td>
<td>500</td>
<td>97.39%</td>
</tr>
<tr>
<td>Calibri</td>
<td>434</td>
<td>66</td>
<td>500</td>
<td>86.8%</td>
</tr>
</tbody>
</table>

While on the background stains degradation, Arial still acquired the highest accuracy while Times New Roman marked the least accuracy due to the same reasons. The sizes and spacing of the characters in Times New Roman became harder to recognize as they were affected by the dark background caused by the stains.

### TABLE III. OCR RESULTS FOR WET

<table>
<thead>
<tr>
<th>Fonts</th>
<th>No. of Recognized Words</th>
<th>No. of Misrecognized Words</th>
<th>Total Number of Words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time New Roman</td>
<td>438</td>
<td>62</td>
<td>500</td>
<td>87.6%</td>
</tr>
<tr>
<td>Arial</td>
<td>456</td>
<td>44</td>
<td>500</td>
<td>91.2%</td>
</tr>
<tr>
<td>Helvetica</td>
<td>444</td>
<td>36</td>
<td>500</td>
<td>88.8%</td>
</tr>
</tbody>
</table>

The documents with pens written over them had the highest error rate among the four degradations. This was caused by the man-made writings that conjoined some of the words together. The model was not able to chop the conjoined words and therefore, not recognized. The colors of the pen also affected the extraction as they made the background darker as well.

For the spelling correction results, only the misspelled words were tested which were not all the misrecognized words from the OCR results. Some of the misrecognized words in the results above were those that tend to be completely unrecognized which implied that the model failed to recognize even a single letter from the words.

### TABLE V. SPELLING CORRECTION RESULTS FOR PAGE CRUMPLE

<table>
<thead>
<tr>
<th>Fonts</th>
<th>No. of Misspelled Words</th>
<th>No. of Corrected Words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time New Roman</td>
<td>25</td>
<td>18</td>
<td>72%</td>
</tr>
<tr>
<td>Arial</td>
<td>14</td>
<td>10</td>
<td>71.42%</td>
</tr>
<tr>
<td>Helvetica</td>
<td>24</td>
<td>16</td>
<td>66.66%</td>
</tr>
<tr>
<td>Calibri</td>
<td>19</td>
<td>10</td>
<td>52.63%</td>
</tr>
</tbody>
</table>

In the crumpled documents, Times New Roman had the highest error rate in spelling. However, the model corrected most of the misspellings and thus, making its accuracy percentage the highest. The spelling errors from the crumpled documents were not too complicated which was why the model was able to correct them.
TABLE VI: SPELLING CORRECTION RESULTS FOR BACKGROUND STAINS

<table>
<thead>
<tr>
<th>Fonts</th>
<th>No. of Misspelled Words</th>
<th>No. of Corrected Words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time New Roman</td>
<td>47</td>
<td>32</td>
<td>68.09%</td>
</tr>
<tr>
<td>Arial</td>
<td>4</td>
<td>3</td>
<td>75%</td>
</tr>
<tr>
<td>Helvetica</td>
<td>6</td>
<td>5</td>
<td>83.33%</td>
</tr>
<tr>
<td>Calibri</td>
<td>42</td>
<td>31</td>
<td>73.80%</td>
</tr>
</tbody>
</table>

Still, the Times New Roman had the highest amount of misspellings in background stains and also had the lowest accuracy because of its least balanced ratio in misspelled and corrected words. The dark background affected the words too much that there were no words from the dictionary close to them.

The spelling correction accuracy achieved the highest in the wet documents since they have the least spelling errors among the other degradations. All the misspellings in Arial and Helvetica were corrected and reached a 100% accuracy. Only one word was not corrected in both Times New Roman and Calibri, but the accuracy of Times New Roman was higher because the number of errors it had is more than Calibri. This degradation did not affect the extraction even though the words were deformed. However, it had lesser deformation than what were identified in the crumpled documents.

TABLE VIII. SPELLING CORRECTION RESULTS FOR WRITTEN OVER

<table>
<thead>
<tr>
<th>Fonts</th>
<th>No. of Misspelled Words</th>
<th>No. of Corrected Words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time New Roman</td>
<td>24</td>
<td>20</td>
<td>83.33%</td>
</tr>
<tr>
<td>Arial</td>
<td>17</td>
<td>13</td>
<td>76.47%</td>
</tr>
<tr>
<td>Helvetica</td>
<td>19</td>
<td>15</td>
<td>78.94%</td>
</tr>
<tr>
<td>Calibri</td>
<td>22</td>
<td>17</td>
<td>77.27%</td>
</tr>
</tbody>
</table>

The Times New Roman still had the highest accuracy as it had the most spelling errors. Even though the missrecognized words in this degradation had the largest amount, only few were to be corrected as the other words were not recognized at all. However, the recognized misspelled ones were corrected fairly.

In summary, having the best OCR results, the degraded documents in Arial also had the least misspellings, while Times New Roman had the most since it achieved the least accuracy in the OCR. In all four degradations, most of the misspellings were corrected. Those which were not corrected were too complicated for the model and did not have close words from the dictionary (e.g. “recognition” became “macafeiniicri”). Moreover, some of the misspelled words were converted into different correctly spelled words from the dictionary (e.g. “weather” instead of “whether”). Still, the model showed impressive results in the spelling correction as more than half of the number of misspelled words were revised which made the overall accuracy higher.

IV. CONCLUSION

This study presented a model that extracts text from degraded documents using Optical Character Recognition (OCR) and spelling correction algorithms. Tesseract was chosen as the OCR engine and dictionary-based spelling correction to be implemented in this study. The performance of the model was tested on different degradations including page crumples, background stains, wet documents, and documents written over with pens – in Times New Roman, Arial, Calibri, and Helvetica fonts. The wet documents achieved the highest accuracy in all four fonts while the documents that were written over by pens achieved the least. The thresholding of the pages was believed to be the cause and the background greatly affected the conversion. As for the fonts, Arial achieved the highest accuracy while Times New Roman achieved the least due to the sizes and the spacing of the fonts. The model corrected more than half of the misspelled words because the other words were already too complicated and were not close to any word from the recorded dictionary.

ACKNOWLEDGMENT

The authors wish to thanks the Dr. Lorena W. Rabago, Dr. Maryli Rosas, Prof. Maria Art Antonette Clarino, and Prof. Raquel Bermudez for the valuable inputs to make this study possible. Likewise to Mr. Adrian Evanculla for helping with the coding difficulties and Ms. Jazmin Tan for reviewing the grammar and sentence structures of this manuscript.

REFERENCES
