Accuracy Evaluation of Arabic Optical Character Recognition Voting Technique: Experimental Study

Yusof A. Batawi and Osama A. Abulnaja

Abstract — In earlier work we have introduced a high accurate Arabic Optical Character Recognition (AOCR) technique. The proposed algorithm is based on the N-Version Programming (NVP) technique and AOCR software. The proposed technique is called Arabic Optical Character Recognition Voting (AOCRV) scheme.

This work studies the effect of the AOCRV technique on AOCR accuracy. The research evaluates four accuracy types: Accuracy\_ALL, Accuracy\_!, Accuracy\_!, and Accuracy\_!.

Key Words — Arabic Optical Character Recognition, N-Version Programming Technique, Arabic Optical Character Recognition Software Accuracy, Accuracy Evaluation.

1. INTRODUCTION

Optical Character Recognition (OCR) programs are used to transform the image of a scanned printed text into an editable electronic file. There are many commercially available OCR software. The accuracy of the English OCR has become very high, but unfortunately, Arabic OCR (AOCR) outputs accuracy are not satisfying [1, 2].

OCR contains the recognition for the printed text and the handwritten text. But to divide the two types of the character recognition researchers specify the OCR software for the printed text recognition. For the recognition of the handwritten text the software called Intelligent Character Recognition (ICR).

1.1. ISRI Voting Algorithm

In 1992, researchers at the Information Science Research Institute (ISRI) at the University of Nevada Las Vegas (UNLV) conducted an experiment to determine the accuracy of six commercially available Optical Character Recognition (OCR) devices. In this experiment other than the six OCR devices, the ISRI used a voting algorithm that used all of the six programs. The result of this experiment was that the ISRI voting algorithm has the most accurate result among the other devices [3].

Also, in 1994 the researchers at ISRI preformed another experiment of OCR accuracy with 12 vendors participated in this test. The result of this experiment was that the ISRI voting algorithm has the most accurate result among the other devices. Moreover, when the accuracy of the results obtained from the OCR programs is low, it will be difficult to get an agreement among the outputs of the participated OCR programs which is hard for a voting system to resolve [4].

Furthermore, in 1996, the researchers at ISRI performed another experiment of OCR accuracy with five vendors participated and the ISRI voting machine. The researchers study the character accuracy between the vendors, the study included the effect of resolution and page quality. The highest performance was by ISRI voting algorithm for all selected samples except for magazine sample it came the second [5].

1.2. Arabic Optical Character Recognition (AOCR) Software

Arabic characters contains 28 characters, each character have a different shape according to its position. Most Arabic characters have four shapes, except some of them have two shapes. This issue extends the character shape from 28 to 100. The difference of writing the character happens when the character been isolated, at the beginning, in the middle and at the ending. Arabic is a cursive language. This is one of the differences between Arabic character and Latin character. Basically this is the main problem for recognition the Arabic characters. Other differences between the Arabic character and the Latin character is that the Arabic characters are written from right to left unlike the Latin character which is written from left to right. Another problem in recognizing the Arabic character is the diacritics which are special samples that change the meaning of a word. Also, in Arabic printed text we can use wrapping between two characters. Those are the main reasons for the difficulties for the Arabic Optical Character Recognition development [6].

There have been several studies on Arabic Optical Character Recognition (AOCR) [7] – [12]. Due to the difficulty in dealing with the Arabic characters there are just few AOCR software commercially available. The most common AOCR software are:

1. Rediris Ver.11 Middle East: supports more than 120 languages including the Arabic language. It can convert PDF files into editable text. Also, it can recognize barcodes and capture hand printed notes. The company claims that this software has the most accuracy in recognition the Arabic characters [13].

2. Sakher Gold V.8: supports the main Arabic based characters languages: Arabic, Farsi, Urdu, Jawi, Pashto,
as well as eighteen other international languages (English, Czech, Danish, Dutch, Finnish, French, German, Greek, Hungarian, Indonesian, Italian, Norwegian, Polish, Portuguese, Russian, Spanish, Swedish, and Turkish). It combines two main technologies: Omni Technology, which depends on highly advanced research in artificial intelligence, and Training Technology, which increases the accuracy of character recognition [14].

3. Vajeh Shenas 1st Edition: is a multi-language optical character recognition that supports Farsi, Arabic, English, Pashtu, Urdu, French and 15 other languages. The company claims that the accuracy for Persian /Arabic language is more than 95% [15].

4. VERUS™ Professional: the company claims that this program is the most accurate Middle Eastern language optical character recognition in the world. It recognizes Arabic, Farsi (Persian), Dari, Pashto, English and French. It automatically detects and cleans degraded and skewed documents, automatically identifies page primary language, and recognizes page fonts without manual intervention [16].

1.3. N-Version Programming (NVP) Approach

One of the most widely used software fault-tolerant technique is the N-Version Programming (NVP) technique. In the N-Version Programming (NVP) technique every program or a program version is made up of \( N \) independently (diversely) designed program versions \( (V_1, V_2, ..., V_{\lfloor N/2 \rfloor+1}, ..., V_{N-1}, V_N) \) performing the same task and a voter (decision function) which generates the program output by taking the output with the majority votes. A program output is considered reliable if at least \( \lfloor N/2 \rfloor + 1 \) versions agree on the same output [17, 18], see Figure 1.

![Fig. 1. NVP Technique](image1)

2. Accuracy Types

In our experiment three types of errors are considered:

1. **Missing Character Error:** occurs when software could not recognize a character and no character is generated.
2. **Added Character Error:** occurs when software does not recognize a character correctly and generates another character instead of the original one.
3. **Marked Character Error:** occurs when software could not recognize a character and generates a mark character. The marked characters are used in all Optical Character Recognition (OCR) software when it could not recognize a character. However, OCR software use different marked character to indicate that error. For example, Rediris Ver.11 Middle East version uses the tilde mark (~), and Sakher Gold V.8 and Vajeh Shenas 1st Edition use the exclamation mark (!).

From above we obtain the following:

\[
\text{Error} = \frac{\text{Total Number of Missing Characters} + \text{Total Number of Added Characters} + \text{Total Number of Marked Characters}}{\text{Total Number of Characters}}
\]

Thus;

![Fig. 2. AOCRV Technique](image2)
Accuracy = \frac{\text{Total Number of Characters} - \text{Error}}{\text{Total Number of Characters}} \times 100 \quad (2)

Also, in our experiment four types of accuracy are considered: \text{Accuracy}_{\text{ALL}}, \text{Accuracy}_\sim, \text{Accuracy}_!\!, \text{and} \text{Accuracy}_{\sim!}. \text{In the following the four types are explained.}

2.1. Accuracy_{\text{ALL}}

In this type of accuracy, generation of tilde mark or exclamation mark for unrecognized character is considered an error. In other words, total number of marked characters is calculated as follows.

\text{Total Number of Marked Characters} = \text{Total Number of Tilde Marks} + \text{Total Number of Exclamation Marks} \quad (3)

From eq.(1) and eq.(3), we get:

\text{Error}_{\text{ALL}} = \text{Total Number of Missin g Characters} + \text{Total Number of Added Characters} - \text{Total Number of Tilde Marks} + \text{Total Number of Exclamation Marks} \quad (4)

From eq.(2) and eq.(4), we get:

\text{Accuracy}_{\text{ALL}} = \frac{\text{Total Number of Characters} - \text{Error}_{\text{ALL}}}{\text{Total Number of Characters}} \times 100 \quad (5)

2.2. Accuracy_{\!}

In this type of accuracy, generation of exclamation mark for unrecognized character is considered an error. On the other hand, generation of tilde mark for unrecognized character is considered correct character. In other words, total number of marked characters is calculated as follows.

\text{Total Number of Marked Characters} = \text{Total Number of Exclamation Marks} \quad (6)

From eq.(1) and eq.(6), we get:

\text{Error}_{\!} = \text{Total Number of Missin g Characters} + \text{Total Number of Added Characters} - \text{Total Number of Exclamation Marks} \quad (7)

From eq.(2) and eq.(7), we get:

\text{Accuracy}_{\!} = \frac{\text{Total Number of Characters} - \text{Error}_{\!}}{\text{Total Number of Characters}} \times 100 \quad (8)

2.3. Accuracy_{\sim}

In this type of accuracy, generation of tilde mark for unrecognized character is considered an error. On the other hand, generation of exclamation mark for unrecognized character is considered correct character. In other words, total number of marked characters is calculated as follows.

\text{Total Number of Marked Characters} = \text{Total Number of Tilde Marks} \quad (9)

From eq.(1) and eq.(9), we get:

\text{Error}_\sim = \text{Total Number of Missin g Characters} + \text{Total Number of Added Characters} - \text{Total Number of Tilde Marks} \quad (10)

From eq.(2) and eq.(10), we get:

\text{Accuracy}_\sim = \frac{\text{Total Number of Characters} - \text{Error}_\sim}{\text{Total Number of Characters}} \times 100 \quad (11)

2.4. Accuracy_{\sim!}

In this type of accuracy, generation of tilde mark or exclamation mark for unrecognized character is considered correct character. In other words, total number of marked characters always equal to zero.

From eq.(1), we get:

\text{Error}_{\sim!} = \text{Total Number of Missin g Characters} + \text{Total Number of Added Characters} \quad (12)

From eq.(2) and eq.(12), we get:

\text{Accuracy}_{\sim!} = \frac{\text{Total Number of Characters} - \text{Error}_{\sim!}}{\text{Total Number of Characters}} \times 100 \quad (13)

3. EXPERIMENT AND RESULTS

Since we could not find an Arabic scanned images database, in our experiment we selected 35 printed text samples from newspaper and magazines. The selected samples contain 20829 characters. All samples are scanned and converted to images at resolution of 300-dot peer inch (dpi) using genies scanner. Three Arabic Optical Character Recognition (AOCR) software are selected (i.e., N = 3): Rediris Ver.11 Middle East, Sakher Gold V.8, and Vajeh Shenas 1st Edition. The outputs of the three selected AOCR software are passed to the Arabic Optical Character Recognition Voting (AOCRV) technique voter, which generates the output with the majority votes. The voting is done for each character generated by the AOCR software. If there is a character with a majority, the character is released; otherwise, a marked character (\sim) is produced.

Tables 1, 2, 3, and 4 show the ranking of the selected three AOCR software and the AOCRV technique (based on the generated outputs by the selected AOCR software for the selected 35 samples) for the \text{Accuracy}_{\text{ALL}}, \text{Accuracy}_{\!}, \text{Accuracy}_{\sim}, \text{and} \text{Accuracy}_{\sim!}, \text{respectively. From these tables we can see that the number of times the AOCRV technique ranked the 1st highest accuracy always the highest.}

Tables 5, 6, 7, and 8 show the accuracy percentage of the selected three AOCR software and the AOCRV technique (based on the generated outputs by the selected AOCR software for the selected 35 samples) for the \text{Accuracy}_{\text{ALL}}, \text{Accuracy}_{\!}, \text{Accuracy}_{\sim}, \text{and} \text{Accuracy}_{\sim!}, \text{respectively. From
these tables we can see that the AOCRV technique outperforms the selected AOCR software.

4. CONCLUSION

In this work we studied the effect of the Arabic Optical Character Recognition Voting (AOCRV) technique on the Arabic Optical Character Recognition (AOCR) accuracy. The AOCRV technique is based on the N-Version Programming (NVP) technique and AOCR software.

In our experiment we selected three AOCR software: Rediris Ver.11 Middle East, Sakher Gold V.8, and Vajeh Shenas 1st Edition. Furthermore, we evaluated four accuracy types: AccuracyALL, Accuracy1, Accuracy2, and Accuracy3.

Our study showed that under the conditions experimented here, the accuracy of the AOCRV technique outperforms the selected AOCR software.

The work accomplished for this research will serve as the foundation for future research work including:

- Study the effect of taking the vote word by word, instead of character by character, on AOCR technique accuracy.
- Study the effect of adding a dictionary to the AOCRV technique on accuracy.
- Study the effect of the type and the size of letter fonts on AOCR software accuracy, and hence on the AOCRV technique accuracy.

REFERENCES


TABLE 1

Software Accuracy All Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>1st highest accuracy</th>
<th>2nd highest accuracy</th>
<th>3rd highest accuracy</th>
<th>4th highest accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCR</td>
<td>26</td>
<td>8</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>10</td>
<td>18</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>Rediris Ver.11</td>
<td>2</td>
<td>3</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>0</td>
<td>3</td>
<td>14</td>
<td>18</td>
</tr>
</tbody>
</table>

TABLE 2

Software Accuracy Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>1st highest accuracy</th>
<th>2nd highest accuracy</th>
<th>3rd highest accuracy</th>
<th>4th highest accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCR</td>
<td>34</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>2</td>
<td>25</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Rediris Ver.11</td>
<td>1</td>
<td>4</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>0</td>
<td>3</td>
<td>14</td>
<td>19</td>
</tr>
</tbody>
</table>
TABLE 3
Software Accuracy, Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>1st highest accuracy</th>
<th>2nd highest accuracy</th>
<th>3rd highest accuracy</th>
<th>4th highest accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCRV</td>
<td>24</td>
<td>10</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>13</td>
<td>17</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>2</td>
<td>2</td>
<td>22</td>
<td>9</td>
</tr>
<tr>
<td>Rediris Ver.11 Middle East</td>
<td>2</td>
<td>1</td>
<td>10</td>
<td>22</td>
</tr>
</tbody>
</table>

TABLE 4
Software Accuracy− Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>1st highest accuracy</th>
<th>2nd highest accuracy</th>
<th>3rd highest accuracy</th>
<th>4th highest accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCRV</td>
<td>34</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>2</td>
<td>27</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Rediris Ver.11 Middle East</td>
<td>1</td>
<td>2</td>
<td>10</td>
<td>22</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>0</td>
<td>4</td>
<td>22</td>
<td>9</td>
</tr>
</tbody>
</table>

TABLE 5
Software Accuracy− Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>Error#</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCRV</td>
<td>460</td>
<td>97.788</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>575</td>
<td>97.235</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>1017</td>
<td>95.1096</td>
</tr>
<tr>
<td>Rediris Ver.11 Middle East</td>
<td>1072</td>
<td>94.8452</td>
</tr>
</tbody>
</table>

TABLE 6
Software Accuracy− Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>Error#</th>
<th>~ Mark</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCRV</td>
<td>310</td>
<td>150</td>
<td>98.5093</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>575</td>
<td>0</td>
<td>97.235</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>1017</td>
<td>0</td>
<td>95.1096</td>
</tr>
<tr>
<td>Rediris Ver.11 Middle East</td>
<td>1043</td>
<td>29</td>
<td>94.9846</td>
</tr>
</tbody>
</table>

TABLE 7
Software Accuracy− Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>Error#</th>
<th>! Mark</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCRV</td>
<td>451</td>
<td>9</td>
<td>97.8347</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>532</td>
<td>43</td>
<td>97.4418</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>841</td>
<td>176</td>
<td>95.956</td>
</tr>
<tr>
<td>Rediris Ver.11 Middle East</td>
<td>1072</td>
<td>0</td>
<td>94.8533</td>
</tr>
</tbody>
</table>

TABLE 8
Software Accuracy− Ranking

<table>
<thead>
<tr>
<th>Software</th>
<th>Error#</th>
<th>~ Mark</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOCRV</td>
<td>301</td>
<td>159</td>
<td>98.5526</td>
</tr>
<tr>
<td>Sakher Gold V.8</td>
<td>532</td>
<td>43</td>
<td>97.4418</td>
</tr>
<tr>
<td>Vajeh Shenas 1st Edition</td>
<td>841</td>
<td>176</td>
<td>95.956</td>
</tr>
<tr>
<td>Rediris Ver.11 Middle East</td>
<td>1043</td>
<td>29</td>
<td>94.9846</td>
</tr>
</tbody>
</table>

BIOGRAPHIES

Yusof A. Batawi received the BS degree in computer science and the MS degree in computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 2005 and 2011, respectively. Currently he is the head of the Academic Affairs Department in Faculty of Computing and Information Technology at King Abdulaziz University. His research interests are Fault Tolerance and Software Engineering.

Osama A. Abulnaja received the BS degree in computer science from King Abdulaziz University, Jeddah, Saudi Arabia, in 1986. He received the MS degree in computer science and the PhD degree in engineering (computer science) from University of Wisconsin-Milwaukee, Milwaukee, Wisconsin, USA, in 1990 and 1996, respectively. Currently, he is a professor of computer science at King Abdulaziz University. His research interests are Fault Tolerance, Systems Programming, Software Engineering, Parallel and Distributed Processing, and Systems Performance.