

# Smart Control of Expenses Using Mobile Devices

Giovani R. F. Junior<sup>1</sup>, Vicente J. P. Amorim<sup>1</sup>, Thiago L. Gomes<sup>2</sup>, Igor M. Pereira<sup>1</sup>

<sup>1</sup>iMobilis Lab – Department of Computer and Systems(DECSI)

ICEA - Federal University of Ouro Preto (UFOP) - João Monlevade - MG - Brazil

gr.ferregetti@gmail.com, vjpmorim@icea.ufop.br, igormuzetti@gmail.com

<sup>2</sup>Department of Computer and Systems(DECSI)

ICEA - Federal University of Ouro Preto (UFOP) - João Monlevade - MG - Brazil

thiagoluange@gmail.com

**Abstract.** *It is estimated that only 3.1% of the Brazilian population controls their expenses through digital applications. While 8.9% does not use a digital platform due to a lack of knowledge, 8.1% do not have time. Considering the current usage levels, applications providing a more automated control of expenditures would simplify this task for an average user, making mobile applications a more attractive option. Using digital image processing techniques, optical character recognition (OCR), and post-processing a novel software was developed in Android for receipts automatic recognition allowing mobile users to monitor expenses using photos. Information recognized by the application replicated a real receipt with an acceptable level of similarity.*

## 1. Introduction

According to the study conducted by the Brazilian Credit Protection Service (SPC Brasil) [Bruno and Miret 2016], only two in five Brazilians properly take care of their expenses. Furthermore, only 3.1% of Brazilians control their spending using digital applications. Compared to the last annual study conducted by SPC Brasil, [Bruno and Miret 2016] and [Gandra 2015], the percentage of people that use digital applications to control spendings reduced from 4% to 3.1%. Also, the percentage of individuals who used paper for controlling spending increased from 23% to 29.8%.

In a separate study conducted through Connect, the Brazilian Institute of Public Opinion and Statistics (IBOPE) digital platform [Administradores 2013], 69% of Brazilians believe that mobile technologies, such as tablets and smartphones, positively contribute to improving population's quality of life. Additionally, 66% of respondents believed that mobile devices are essential to becoming informed consumers. Although that, consumers that are controlling their expenses are abandoning the use of these electronic devices. One hypothesis blames currently available applications, which fail to provide a satisfactory performance or function to meet consumer needs: difficult use, or require excessive amounts of time and effort to use[Zhel 2016][Google 2016c]. Besides that, from the user's point of view, inserting data from each receipt might become very annoying and time consuming.

To improve consumer use of technology to monitor their expenses, this paper presents the development of a novel way to control financial data through a mobile application. Using digital image processing techniques and optical character recognition

(OCR), the approach proposed here analyzes a photograph taken by a mobile device to retrieve receipt information. Thus, providing a simple system for consumers to digitally keep track of all expenses.

Some work has been done to improve Digital Image Processing, Optical Character Recognition and Post-Processing techniques such as the proposed by Hang[Heng 2013], Hasnat et. al.[Hasnat et al. 2009] and Bassil et. al.[Bassil and Alwani 2012]. Next section makes a comparison between these works.

The remaining of this paper is divided into the following sections: Section 2 present the works that are close to the one described here. Section 3 depict main motivations that led to this work. Section 4.1 explains the two sets of receipts created and their purpose. Section 4.2 describes how the image processing techniques were selected. Section 5 show how a test application was built to validate our idea. By its turn, Section 6 present the results while Section 7 conclude this work and describe possible future works.

## 2. Related Works

Application developed by Hang (2013) [Heng 2013] used user images to determine how a bill will be split amongst people at a table in a restaurant or bar. After taking the picture, the user must then manually set regions containing items and prices in the image. The application recognizes the data and lists each contribution value based on items a person has consumed. This study mentions the use of an adaptive algorithm for binarization, despite no specific technique is specified. After binarization, resulting image is recognized using the Tesseract library[Library 2016] without any training. After that, the text is processed, and items contained in the pictures are identified. Unlike the work presented here, this proposal trusts on a user intervention to select specific parts of a picture where the most important part of information is.

Hasnat et al. [Hasnat et al. 2009] designed an OCR process software for the Bengali language in combination with the Tesseract library, which was called BanglaOCR. This paper focused mainly on Tesseract training and post-processing techniques. Initially, the authors created fourteen different training sessions to determine what combination of training produced optimum results. Therefore, it developed a post-processing on two levels: correction recognition problems based on specific language's rules, for instance, the characters order; and the use of eighty thousand dictionary words, whose function is to check words spelling in the text. Final result of recognition was 93 % for high-quality images. This paper obtains the document picture using a scanner with a specific angle, different from what we proposed here, which is a receipt picture obtained by a mobile camera, with the angle determined by the user through the guidelines.

Bassil and Alwani [Bassil and Alwani 2012] proposed a post-processing algorithm to correct syntax and semantic mistakes based on their context. Presented algorithm uses an online word correction service maintained by Google. It divides resulting text of optical character recognition in five words blocks, which was judged to be enough to produce a context. These blocks are sent to Google's correction service. When a response did not contain the phrase "you mean ...", the five words block was considered as a correct block; and when the answer did contain that phrase, the old block is replaced by the suggested one. According to the authors, proposed algorithm improved error correction rate by 690% for a document in English and 403% for a text in Arabic. Although this tech-

nique can correct syntactic and semantic errors, it depends on the Google's service and an internet connection. Here, we propose a post-processing mechanism using a framework which is built inside Android and does not need a permanent internet connection.

Other projects have very similar proposals. As an example, there is a Brazilian project called NFSCAN [Application 2015], which uses digital image processing and optical character recognition to identify data from receipts to ease credits donation to Non-Governmental Organizations. However, the lack of a specific documentation about its implementation and results prevents a further discussion.

### **3. Motivations and Main Objectives**

As presented before, expenditures control via mobile applications still not popular. Reasons for this unpopularity are on mobile software complexity, lack of usability or even due to the lengthy time required to manage provided information. Low user adoption of digital methods to control expenditures may also be caused by the lack of specific functionalities. To provide an easy way of taking control of expenses and prevent users from having to enter all the receipt's relevant information would reverse that digital abandonment. Therefore, a tool that can interpret receipt's data and perform necessary calculations automatically is essential. Indeed, the user can simply use a smartphone to photograph the receipt and thus keep track of expenses.

To implement an application based on photographic data recognition to control expenses, digital image processing methods and optical character recognition must obtain relevant information from the receipt. However, depending on image quality and character font used to write the receipt, these two techniques may not be enough. As a way to circumvent this fact, previous works [Hasnat et al. 2009] [Bassil and Alwani 2012] proposed a step called post-processing, known as a way to enhance information obtained by OCR through the use of online or offline spelling checkers. Since internet access is not available all the time for the users, any application that performs digital image processing, OCR and post-processing shall let it into account.

Considering the points raised above, the solution presented by this work covers the creation of an application to automatically identify data from a generic receipt allowing the users to control his expenses in a more fashionable way.

### **4. Data Sampling and Test Set**

This section describes the selection process of receipts and techniques considered by this work.

#### **4.1. Selection of Receipts**

To apply those methods described in the next section and get final results from the method proposed here, we created two sets that include all types of receipts: new, old, crumpled, with different fonts and layouts.

The first set has 50 receipts which are used to compare image processing methods and create 53 training regiments. The second set has 30 receipts which are used to apply the post-processing and obtain the final recognition rate.

## 4.2. Selection of Techniques

This section describes main methods used during development of this project. Digital image processing methods, such as binarization and segmentation, were used to improve OCR results. Training methods are commonly used when is necessary to apply OCR over a non-standard fonts and idioms. By its turn, post-processing is used to correct spelling mistakes in the results coming from OCR tool.

## 4.3. Digital Image Processing

Several tests were performed involving filters and digital image processing techniques, for example: binarization, median filter, Gaussian filter, erosion and dilation, and transformed perspective. These techniques were taken into account as a way to improve the recognition rate of information retrieved from the receipts. However, some processes degraded the picture quality and its characters, even though it results in a greater recognition rate, that occurs owing to the noise reduction.

### 4.3.1. Binarization

Binarization algorithms compare each pixel value to a threshold value, that can be fixed, calculated using the whole image, or even calculated using different regions of a picture. So, if the pixel value is greater than a threshold value, the pixel is set to foreground. Otherwise, the pixel value is set to background[Gonzalez and Woods 2010].

Binarization method with a fixed threshold value proved to be the least efficient technique because it can not process lighting differences, such as shadows. Indeed, in the work presented here, binarization function provided by Tesseract library produced worst results comparatively to the top 4 algorithms, as shown by Figure 5.

In contrast, best results were obtained by both local and global adaptive techniques, which adjust threshold value to the image being binarized. Still, two best results were produced by Sauvola's and Wolf's algorithm, which represent the local adaptive binarization technique. Unlike global adaptive binarization method, the threshold of each region of an image adapts such region in a way that it will have a specific threshold value. Therefore, binarization algorithm chosen during application development was the one proposed by Sauvola due to its better results.

### 4.3.2. Segmentation

Segmentation is the process of dividing one image into different regions so that they can be treated separately later[Gonzalez and Woods 2010]. Tesseract library[Google 2016b] provides a function that allows image segmentation into lines, words and symbols, for images that contain text. However, our results showed that recognition was higher without segmenting the receipt, as shown by Figure 6. Consequently, a segmentation method was not considered by the application.

## 4.4. OCR Tool Training

Tesseract library can recognize common fonts from different languages. It is necessary to tell the library how to recognize new fonts or new languages, through a process called

training [White 2012].

Training occurs by representing characters in an image with a text symbol contained in that picture. For instance, an alphabet letter is represented by a particular image segment. Furthermore, it can be taught how this new “language” works by combining its symbols, forming a dictionary [White 2012]. This work have used the Tesseract library to recognize receipts text.

#### 4.4.1. Training Files

Initially, 53 training regiments were created and tested to determine the success of a training file. From this set of training regiments, 50 were created using each test file, 1 was created using the whole set and the last 2 were obtained from Google’s page. Specifically at the initial phase, 50 receipts were considered for the first training set. These training files were obtained in two steps: Qt-box-editor[Qt 2016] tool was used to create box files, which was in charge of teaching Tesseract[Library 2016] what means each character symbol in the image, creating a binary file with a “tr” extension; moreover, the Tesseract library was used to generate the training file. Figure 7 presents 13 most significant training files.

Analyzing Figure 8, some of the training files produced better results recognizing numbers than characters. However, post-processing phase explained in section 4.5, correct words that are incorrectly spelled and cannot make any correction to the segments that contain numbers. Then, to perform a comparative analysis of the post processing impact, two training files were chosen to be tested with the second set that contains 30 receipts: the one with the highest percentage of numbers recognition, called “por” training file; and the one with second highest average recognition including all types of symbols, called “t16” training file.

#### 4.5. Post-processing

Post-processing main objective is to detect and correct spelling mistakes created by optical character recognition tool.

As explained in detail by [Bassil and Alwani 2012], there are two types of errors: semantic and syntactic. Therefore, there are also three approaches of correction: **Manual error correction**: a group of people is hired to correct the errors manually. Although it is the easiest way, it requires continuous human intervention; **Dictionary-based correction**: corrects the mistakes based on a dictionary; and **Context-based correction**: correct the mistakes based on grammatical error and semantic context. In this paper, we implemented a dictionary-based correction method using Android’s Spelling Checker Framework[Google 2016a].

#### 4.6. Levenshtein-Distance Algorithm

This algorithm calculates the minimum number of operations needed to transform one string of characters to another. Allowed operations are: insertion, substitution, and removal[Levenshtein 2016]. Therefore, from the number of required operations, is possible to calculate a similarity level between two characters sequence. Here, the Levenshtein

algorithm was used to calculate the similarity level between text automatically recognized and the original text – previously loaded to text files by a human developer.

## 5. Application Development

Bellow there is a description of methods used to implement the application main parts.

### 5.1. Implementation

As discussed in section 4.2, methods implemented by the application were: **Binarization Method**: Sauvola method provided best results; **Segmentation Method**: Omitted because recognition percentage worsened; **Training file**: “por” training file provided best number of recognitions among the training files explored with satisfactory character recognition; and **Post-processing**: Android’s Spelling Checker Framework was chosen because it provided an offline way to access the data.

### 5.2. Camera

This work proposes a test application called “NotaFiscal”, that has an initial screen similar to a common camera application. When the application starts, it is possible to see camera image, as shown in Figure 1. A green line, as presented by Figure 1, aims to assist the user in aligning the camera and the receipt. In Figure 2 there are two lines which also serve as a guide. Additionally, operation mode described above is complemented with an edge detection system to allow a better visualization and alignment.

With a screen tap, a receipt picture is taken, which will result in image saving and character recognition. To enable edge detection function, the user must execute a screen long tap for a few seconds.

### 5.3. Storage

To allow a later usage application saves data obtained from optical character recognizer right after it is processed by post-processing step. A new file is created in “recAppNotaFiscal” folder as shown by Figure 3.

Images, in a binarized form, are stored in picture’s folder, as shown in Figure 4. Storing the pictures in a binarized form is justified by its size, which is much lower when compared to the colored version.

Indeed, training file used in this project, “por” is stored inside “tessdata” folder as shown in Figure 3.

## 6. Evaluation and Results

This section discusses results of optical character recognition (OCR) and post-processing techniques using the two best training files, “por” and “t16”.

It is important to emphasize that OCR results only involve methods with best results in our tests. Those methods are Sauvola’s Algorithm for Binarization, without image segmentation, and the best training files. Furthermore, results of proposed post-processing method are based on Android’s internal dictionary.

As shown in Figure 9, the average result of OCR using “por” training file was 61.28%, with the best individual result being 78.0%. Post-processing enhanced similarity to 63.34% on average without changing best similarity in individual results.



Figure 1. Application initial screen presenting the receipt and guideline.



Figure 2. Application screen showing guide-lines and edge detection.

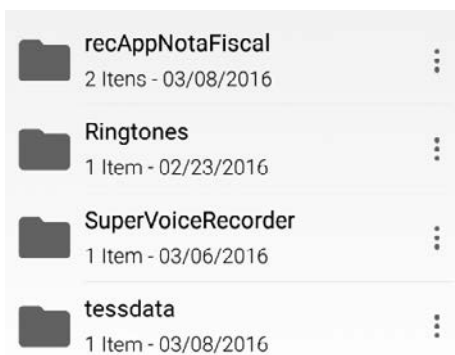
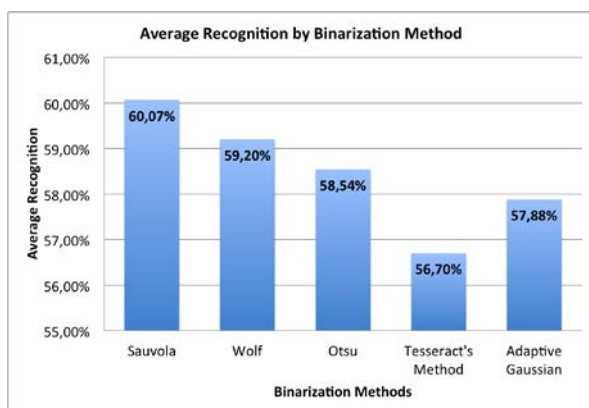


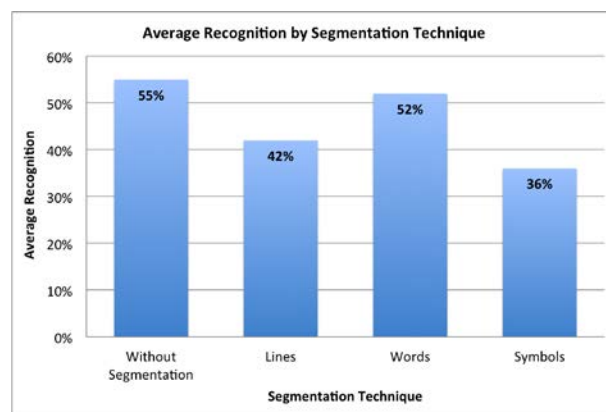
Figure 3. Application folders to store training files and post-processing results.



Figure 4. Album containing receipt photos taken by the user.



**Figure 5. Tests results evaluating the best binarization method. For each set, 50 receipts were taken into account.**



**Figure 6. Tests results evaluating the best segmentation method. For each set, 50 receipts were taken into account.**

Average result for “t16” training file was 63.18% (Figure 10). Post-processing enhanced similarity to 67.07%. In addition to that, best OCR result had its similarity decreased from 78.0% to 66.86%, so the best result after post-processing was 75.0%.

It was observed that OCR results can be affected by receipt’s characteristics. For instance, the first factor is receipt font non-standardization. Other influences include printing quality and how old receipt is. Thus, the receipt should be photographed as soon as possible to provide the best OCR results. Moreover, receipt physical state, e.g. scratched or cropped, also had a large impact on its recognition rate.

Hence, OCR results had the greatest impact on post-processing technique, specifically the exchange of letters, numbers and special characters. Techniques discussed here were not applied to correct sections of text that have numbers and special characters. This fact is mainly because it can have particular information about purchases. Indeed, the focus of this work was on syntactic and semantic errors correction.

This work also implemented an algorithm to split OCR results into sections of text through the recognition of blank spaces. If these sections only contain letters, the algorithm looks and replaces old section by the word(s) that has the biggest similarity level. Table 1 shows information regarding each receipt after post-processing. This information is divided into a total number of sections, number of words, number of correct words, and number of words that were replaced by a similar word.

It is possible to conclude that post-processing enhanced recognition rate for both training files, considering the average result. Interestingly, post-processing applied to OCR result of “t16” training file led to an enhancement up to 12.69% for one receipt. However, it also resulted in a decrease of 16.01% of similarity for another receipt. Regarding the “por” training file, post-processing led to an improvement up to 7.57%, while the similarity level decreases up to 0.27%, which is almost 60 times smaller than in training file “t16”.



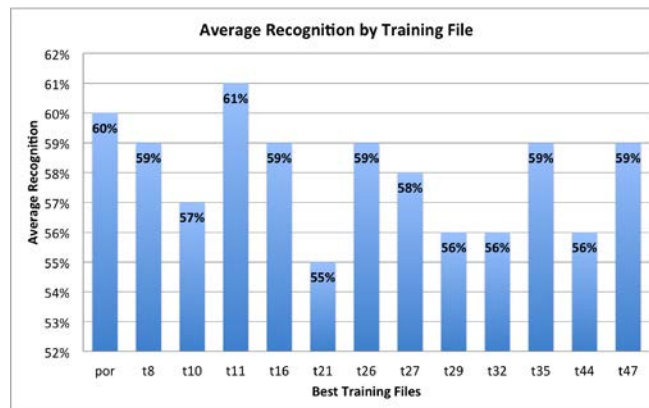


Figure 7. Tests results evaluating the best 13 training files from the set of 50 receipts.

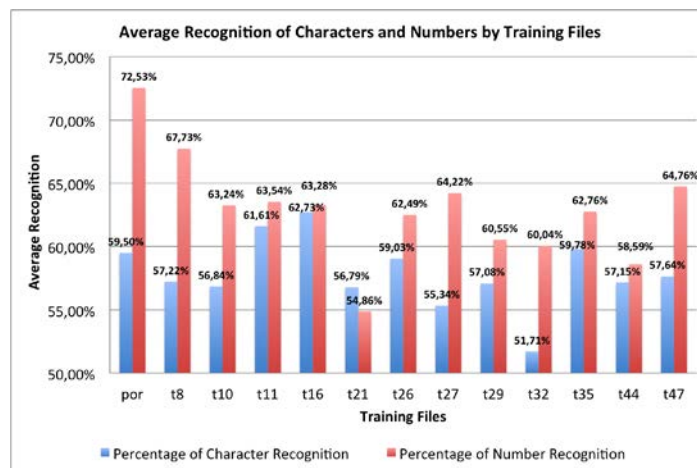


Figure 8. Tests results figuring out the percentage of character recognition and number recognition from the set of 50 receipts.

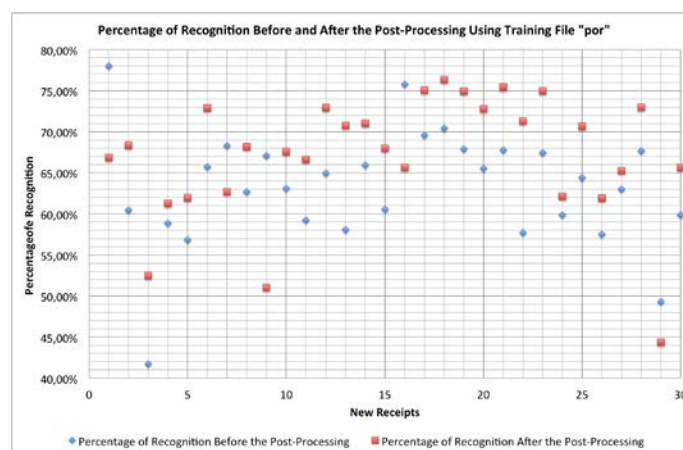
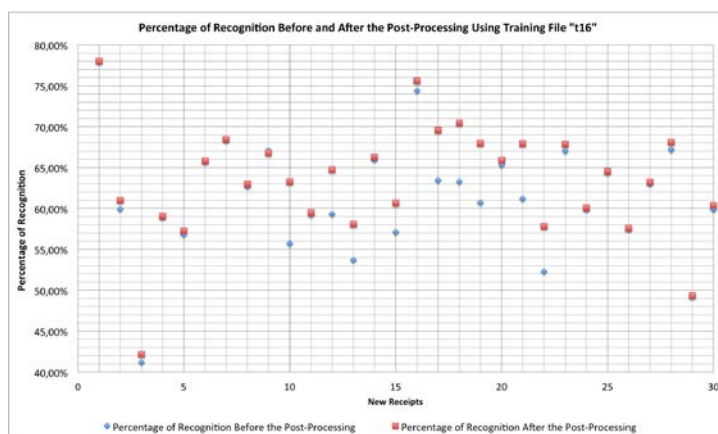


Figure 9. Results considering a new set of 30 receipts. Post-processing was applied on OCR results with “por” training file.



**Figure 10. Results considering a new set of 30 receipts. Post-processing was applied on OCR results with “t16” training file.**

## 7. Conclusions and Future Works

Recently, technology has evolved to provide a better quality of life for the population by simplifying everyday tasks. Activities that previously demanded hours to be completed can now be executed in few seconds with the aid of technology. This paper proposes a development of an application that can help consumers to take control of their expenses. Arduous task of creating spreadsheets and performing math can be exchanged by a simple task of taking a photo of the receipt in conjunction with the convenience of a smartphone.

When compared to similar works, application proposed here achieved good results. Our work specifically considers characters automatic recognition in the mobile environment without any user intervention. Furthermore, recognition rate obtained here encloses different formats of documents, which also use different sets of fonts.

This application development was made possible by the combination of best digital image processing, optical character recognition, and post-processing techniques. Results revealed a significant impact when considering a specific set of documents – receipts. In summary, for this specific case, best binarization method was proved to be the one proposed by Sauvola. Interestingly, image segmentation into lines, words, and characters resulted in a worse recognition rate. Also, the training file here called “por” provided by the Tesseract’s library web page resulted in the best percentage of characters recognition. Finally, the post-processing was demonstrated to be very effective on the similarity enhancement in a comparison between the recognized words and the real words held on receipt.

Each technique considered by application final version was chosen by decision making based on real data. Therefore, during the evaluation of each technique, it was possible to visualize an increased similarity between the original receipt and recognized data. Regarding techniques used in this study, it was observed that Sauvola’s algorithm results in better information recognition. In addition, the training file available on Google’s page, “por” training, has the best average character recognition rate, having a great importance in approach proposed by this work. It was also observed that no receipts segmentation technique resulted in better recognition rates. Finally, the post-processing anchored in Android’s platform native spelling checker produced a significant improvement in the

| "por" Training |                    |                 |             |                | "t16" Training |                    |                 |             |                |
|----------------|--------------------|-----------------|-------------|----------------|----------------|--------------------|-----------------|-------------|----------------|
| Receipt        | Number of Segments | Number of Words | Exact Words | Replaced Words | Receipt        | Number of Segments | Number of Words | Exact Words | Replaced Words |
| 1              | 114                | 44              | 0           | 3              | 1              | 115                | 42              | 0           | 2              |
| 2              | 88                 | 25              | 0           | 5              | 2              | 91                 | 25              | 1           | 6              |
| 3              | 83                 | 29              | 0           | 7              | 3              | 81                 | 22              | 5           | 2              |
| 4              | 164                | 62              | 0           | 4              | 4              | 165                | 53              | 0           | 6              |
| 5              | 164                | 57              | 0           | 8              | 5              | 163                | 55              | 6           | 7              |
| 6              | 163                | 52              | 0           | 3              | 6              | 163                | 56              | 14          | 7              |
| 7              | 95                 | 36              | 0           | 5              | 7              | 97                 | 33              | 0           | 5              |
| 8              | 155                | 54              | 0           | 6              | 8              | 154                | 50              | 1           | 5              |
| 9              | 112                | 34              | 0           | 5              | 9              | 111                | 39              | 6           | 5              |
| 10             | 231                | 90              | 0           | 4              | 10             | 231                | 84              | 2           | 6              |
| 11             | 110                | 36              | 0           | 4              | 11             | 108                | 34              | 0           | 5              |
| 12             | 157                | 53              | 0           | 4              | 12             | 156                | 48              | 6           | 7              |
| 13             | 150                | 57              | 0           | 6              | 13             | 148                | 47              | 11          | 7              |
| 14             | 132                | 50              | 0           | 4              | 14             | 132                | 47              | 1           | 4              |
| 15             | 164                | 61              | 0           | 3              | 15             | 163                | 54              | 2           | 5              |
| 16             | 173                | 58              | 0           | 4              | 16             | 175                | 50              | 0           | 11             |
| 17             | 201                | 64              | 0           | 7              | 17             | 202                | 62              | 0           | 10             |
| 18             | 209                | 77              | 0           | 4              | 18             | 209                | 73              | 0           | 7              |
| 19             | 216                | 71              | 0           | 4              | 19             | 221                | 74              | 11          | 5              |
| 20             | 113                | 37              | 0           | 7              | 20             | 113                | 37              | 1           | 6              |
| 21             | 175                | 65              | 0           | 6              | 21             | 175                | 64              | 0           | 4              |
| 22             | 175                | 62              | 0           | 1              | 22             | 177                | 56              | 9           | 3              |
| 23             | 143                | 51              | 0           | 8              | 23             | 145                | 49              | 3           | 5              |
| 24             | 139                | 40              | 0           | 2              | 24             | 140                | 39              | 0           | 6              |
| 25             | 131                | 45              | 0           | 3              | 25             | 132                | 44              | 1           | 6              |
| 26             | 123                | 44              | 0           | 6              | 26             | 122                | 46              | 0           | 4              |
| 27             | 149                | 49              | 0           | 4              | 27             | 147                | 50              | 0           | 6              |
| 28             | 160                | 50              | 0           | 6              | 28             | 157                | 51              | 1           | 5              |
| 29             | 94                 | 29              | 0           | 3              | 29             | 94                 | 28              | 1           | 2              |
| 30             | 140                | 43              | 0           | 6              | 30             | 142                | 41              | 0           | 4              |
| <b>Mean</b>    | <b>147.43</b>      | <b>50.83</b>    | <b>0</b>    | <b>4.73</b>    | <b>Mean</b>    | <b>147.63</b>      | <b>48.43</b>    | <b>2.73</b> | <b>5.43</b>    |

**Table 1. Results of OCR using "por" and "t16" training files, number of segments that represent words and number of replaced words.**

final recognition rate.

In addition to evaluation data, this research proposed an initial method of post-processing, however, this area has great potential for growth. Several other techniques can be implemented to enhance similarity of the recognized text further, considering the same mobile environment. Therefore, next possible step of the post-processing could use machine learning and/or another technique to correct recurring mistakes that arise from OCR. Future works could also analyze the impact of an automatic correction method based on machine learning. As proposed in [Hasnat et al. 2009], some correction methods can be implemented to correct spelling mistakes based on information that has a specific format and predefined rules, such as, date, hour, total amount and etc.

Finally, the application proposed here has a great potential to improve expenses management. In addition to providing an alternative and much less laborious method, this method provides organized data storage, which traditionally is made by several folders or spreadsheets. Therefore, it is expected that with some improvements made by future studies, this application may have a significant effect on the users' life quality boosting mobile digital applications to financial managements.

## References

- Administradores (2013). Para brasileiros, smartphone e tablet melhoram a qualidade de vida. <http://www.administradores.com.br/noticias/tecnologia/pesquisa-para-brasileiros-smartphone-e-tablet-melhoram-a-qualidade-de-vida/75469/>. Last accessed 2016, March 09.
- Application, N. (2015). <http://www.nfscan.cc/?lang=pt>. Last accessed 2016, February 29.

- Bassil, Y. and Alwani, M. (2012). Ocr post-processing error correction algorithm using google's online spelling suggestion. *Journal of Emerging Trends in Computing and Information Sciences*.
- Bruno, V. and Miret, R. (2016). 46% dos brasileiros não controlam seu orçamento, revela pesquisa do spc brasil. <https://www.spcbrasil.org.br/imprensa/noticia/971-46dosbrasileirosnaocontrolamseuorcamentorevelapesquisadospcbrasil>. Last accessed 2016, March 09.
- Gandra, A. (2015). Spc: falta de disciplina é maior dificuldade para controle de gastos. <http://agenciabrasil.ebc.com.br/economia/noticia/2015-01/falta-de-disciplina-e-principal-dificuldade-dos-brasileiros-para-nao>. Last accessed 2016, February 02.
- Gonzalez, R. C. and Woods, R. E. (2010). *Processamento Digital de Imagens*. Pearson Education, 3 edition.
- Google (2016a). Android spelling checker framework. <http://developer.android.com/intl/pt-br/guide/topics/text/spell-checker-framework.html>. Last accessed 2016, February 29.
- Google (2016b). How to use the tools provided to train tesseract 3.0x for a new language accessed 08 mar. 2016. <https://github.com/tesseract-ocr/tesseract/wiki/TrainingTesseract>. Last accessed 2016, March 08.
- Google (2016c). Mobile app marketing insights: How consumers really find and use your apps. <https://think.storage.googleapis.com/docs/mobile-app-marketing-insights.pdf>. Last accessed 2016, April 01.
- Hasnat, M. A., Rahman, M., and Khan, C. M. (2009). An open source tesseract based optical character recognizer for bangla script. *10th International Conference on Document Analysis and Recognition*.
- Heng, A. (2013). L'addition: Splitting the check, made easy. [https://stacks.stanford.edu/file/druid:yt916dh6570/Heng\\_LAddition\\_Restaurant\\_Check\\_Splitting.pdf](https://stacks.stanford.edu/file/druid:yt916dh6570/Heng_LAddition_Restaurant_Check_Splitting.pdf). Last accessed 2016, March 09.
- Levenshtein (2016). The levenshtein algorithm. <http://www.levenshtein.net/>. Last accessed 2016, March 02.
- Library, T. O. (2016). Tesseract ocr library source code. <https://github.com/tesseract-ocr>. Last accessed 2016, march 01.
- Qt (2016). Qt box editor tool. <https://zdenop.github.io/qt-box-editor/>. Last accessed 2016, march 08.
- White, N. (2012). *Training Tesseract for Ancient Greek OCR*. Department of Classics and Ancient History 38 North Bailey Durham, DH1 3EU, UK, Email: [nick.white@durham.ac.uk](mailto:nick.white@durham.ac.uk).
- Zhel, M. (2016). 10 reasons why people abandon your app. <https://mofluid.com/blog/10-reasons-why-people-abandon-your-app>. Last accessed 2016, April 01.