DOI: https://doi.org/10.53555/nnbma.v1i1.138 **Publication URL: https://nnpub.org/index.php/BMA/article/view/138**

IMPLEMENTATION PAPER ON EFEECTIVE WAY OF MOBILE RECHARGE

Nitinkumar Sanodiya[1*], Vipul Thawre[2], Ankur Pohekar[3], Raju Gedam[4], Kunal Sarote[5], Dinesh Gawande[6]
[1,2,3,4,5,6]*Dr. Babasaheb Ambedkar College of Engineering and Research*
**Email ID – [1]gawande.dinesh@gmail.com, [2]rahul12thawre@gmail.com,* [3]nitin.san124@gmail.com, [4]rajugedam286@gmail.com, [5]kunalsarote9017@gmail.com, [6]ankurpohekar286@gmail.com

****Corresponding Author: -***
**Email ID: gawande.dinesh@gmail.com*

Abstract: -

The title of our project is Smart Scan Recharge is android application which is able to do the recharge by just only scanning the recharge voucher. This application will also display the location of the recharge vender on GPS. This way of doing a recharge is very effective because it is very efficient than the manual process.

Keywords: - *Implementation, Efeective, Mobile Recharge*

I.INTRODUCTION

In these modern days all are very use to or can say very friendly about smart phone. Mobile has the number of different applications but one its very basic use is to talk with each other for this service have to pay some money which paid by doing a mobile recharge. Traditionally recharge is done by purchasing recharge voucher scratch it and enter the number followed by the unique code. This unique code is different according to service provider. At the end of code use # symbol. After some time, our mobile get recharge.
For example: *101*<14-digit code># for AIRTEL.
In this if any symbol goes wrong or missing then have to repeat all the procedure, which can be time consuming for the user.

By implementing this able to do this process in very short time. This can be achieved as follows.
1) Choose service provider e.g., AIRTEL, IDEA etc. from app.
2) Scan the 14-digit number.
3) Recharge successful.
By this the process of recharge very simpler and user friendly also it takes very less time.

II. Architecture

The architecture is mainly divided into several modules; the major modules are for scanning the recharge voucher, display the vendor location on GPS. But the main task is converting the image into text for that we use the OCR engine and OCR engine uses a Tesseract library for converting the image into text. Another module is to display the location of the vendor on GPS, in this module first it will detect the current location of the user and according to that it will display the location of the vendor on GPS. In this the main goal is to perform the recharge by just scanning the recharge voucher and pressing the call button. In this process the main function is, while we scan the recharge voucher, we convert the scanned image into the text and OCR extract the number from the voucher for that purpose OCR uses tesseract library.
Tesseract library provide the functionality for extracting the character from the image or from the sentence.

III.The detailed description of the tesseract OCR engine is as follows.

a. Architecture of tesseract ocr engine

Since HP had independently-developed page layout analysis technology that was used in products, Tesseract never needed its own page layout analysis. Tesseract therefore assume that its input is a binary image with optional polygonal text regions defined. Processing follows a traditional step-by-step channel, but some of the stages were different in their day, and possibly stay so even now. The first step is a related component analysis in which outlines of the components are stored. This was a computationally costly design decision at the time, but had a significant advantage: by observation of the nesting of outlines, and the number of child and grandchild outlines, it is simple to find inverse text and identify it as easily as black-on-white text. Tesseract was first OCR engine which handle white-on-black text so trivially. At this stage, outlines are grouped together, purely by nesting, into *Blobs*. Blobs are prepared into text lines, and the lines and regions are observed for fixed pitch or relative text. Text lines are divided into words differently according to the kind of character arrangement. Fixed pitch text is chopped immediately by character cells. Proportional text is broken into words using definite spaces and fuzzy spaces.
Identification is a two-phase process. In the first pass, an attempt is made to find each word in turn. Each word that is acceptable is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more correctly recognize text lower down the page. Since the adaptive classifier may have educated something useful too late to make a influence near the top of the page, a second pass is run over the page, in which words that were not found well enough are found again. A final phase resolves fuzzy spaces and checks different hypotheses for the x-height to locate lowercase text.

IV.Line And Word Finding

a) Line Finding

The line searching algorithm is one of the several parts of Tesseract that has formerly been published. The line searching algorithm is planned so that a tilted page can be find without having to de-skew,
thus, saving loss of picture quality. The key parts of the process are blob filtering and line manufacture. Consideration that page design analysis has already provided text regions of a roughly uniform text size, a simple percentile height filter eliminates drop-caps and vertically moving characters. The center height similar the text size in the region, so it is safe to clean out blobs that are smaller than some fraction of the middle height, being most likely punctuation, diacritical marks and noise. The filtered blobs are more similar to fit a model of non-overlapping, similar, but sloping lines. Sorting and processing the blobs by x-coordinate makes it promising to assign blobs to a different text line, while finding the slope across the page, with greatly reduced danger of assigning to an incorrect text line in the presence of skew. Once the filtered blobs have been given to lines, a least center of squares fit is used to estimate the baselines, and the filtered-out blobs are fitted back into the correct lines. The final step of the line creation process merges blobs that overlap by at least half horizontally, giving diacritical marks organized with the correct base and correctly associating parts of few broken characters.

b. Fixed Pitch Detection and Chopping

Tesseract evaluate the text lines to search whether they are static pitch. Where it finds fixed pitch text, Tesseract chops

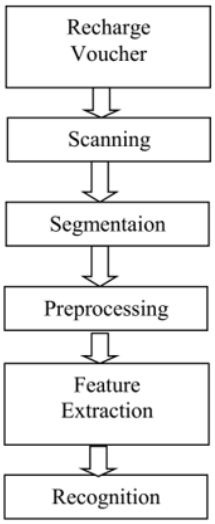
the words into characters consuming the pitch, and deactivates the cutter and associator on these words for the word recognition step, Fig shows a typical example of a fixed-pitch word.

Fig(a)A fixed-pitch chopped word.

V. Methodology

Components of OCR

Given figure(b) shows the overall functioning of optical character recognition (OCR) it contain some steps to recognize text. These steps are: scanning, segmentation, preprocessing, feature extraction, recognition. Here the input to the OCR is given as scanned recharge voucher. Firstly, it is scanned by using Android mobile. It means it digitizes the analog document. The symbols regions within the image are located, extract symbols through segmentation, preprocess each symbol, extract features and recognize them.



Figure(c) Optical Character Recognition

a) Scanning

This application uses customize Android mobile camera take code from voucher as input this process is known as scanning. Generally original input is made up of black colored. Scanning comes with a thresholding which makes the digital image as gray scale image. Thresholding is the process which converts multi-level image into bi-level image i.e., black and white image. [4] Fixed threshold level is defined in thresholding. If the gray levels are below the threshold level, identified as black. Whereas if gray
level is above the threshold level, identified as white. This results in saving memory space and computational efforts.

b. Segmentation

The process of spotting regions of printed or hand written text is segmentation. Segmentation unique text from figures and graphics. When segmentation is applied to text, it isolates characters or words. The mostly occurred problem in segmentation is: it causes confusion between text and graphics in case of joined and split characters. Usually, splits and joints in the characters causes due to scanning. If document is dark photocopy or if it scanned at low threshold, joints in characters will occur. And splits in characters will occur if document is light photocopy or scanned at high threshold.[4] OCR system also gets confused during segmentation when characters are connected to graphics.

c. Preprocessing

As we seen above, some noise may occur during scanning process. This results in poor recognition of characters. This usually occurred problem is overcomed by preprocessing. It consists of smoothing and normalization. In smoothing, certain rules are applied to the contents of image with the help of filling and thinning techniques. Normalization is responsible to handle uniform size, slant and rotation of characters.

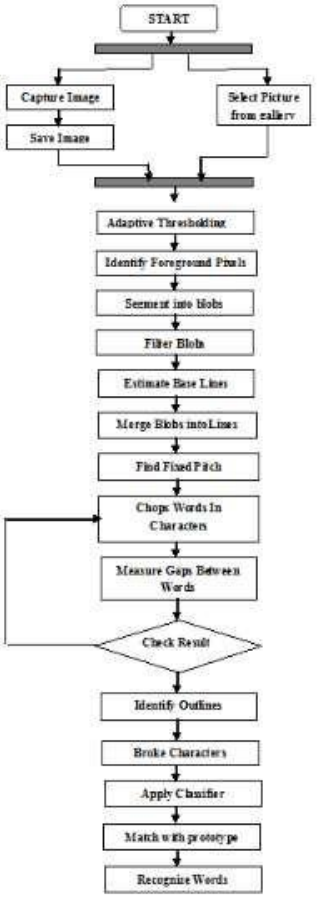
d. Feature Extraction

It extracts the features of symbols. Features are the characteristics. In this, symbols are characterized and unimportant attributes are left out. The feature extraction technique does not match concrete character patterns, but rather makes note of abstract features present in a character such as intersections, open spaces, lines, etc. [7] Tesseract algorithm is used to implement feature extraction. Feature extraction is concerned with the representation of the symbols. The character image is mapped to a higher level by extracting special characteristics of the image in the feature extraction phase.

e. Recognition

OCR system works with Tesseract algorithm which recognizes characters. Tesseract identifies characters in foreground pixels, called as blobs, and then it finds lines. Word by word recognition of characters is done throughout the lines. Recognition involves converting these images to character streams representing letters of recognized words [8]. In short, recognition extracts text from images of documents

IV. Flow Chart



VI. Implementaion

The main modules of our application are scanning the recharge voucher and display the location of vendor on GPS.

a. Home Screen

This is the home screen of our system it contain two image buttons one for scanning the recharge and other one for to view the nearest recharge shop. When click at scan recharge voucher image button then open a scanner for scanning the voucher. When we click at nearest recharge shop button it will display the location of all recharge shops where we can get a recharge.

b. Selecting Operator

In this system when click at select network then open a list of operator from which we have to select a operator. After that we have to click on proceed it will open a scanner.

c. Scanning a voucher

This snap shot contain scanning voucher through android mobile it will scan all the digits from the voucher and extract it.

d. Display Location of Recharge of Vendor on GPS

When we click at the nearest location shop button it will show a location of all the nearest shops which cell the recharge voucher.

VII. Display Location of Vendor on Gps

In this can be able to display the location of the vendor on GPS. For this use google api for finding the location and to display the location use GPS apia. Location APIs

In our system we use location APIs make it easy to build location-aware applications, without needing to focus on the details of the underlying location technology. They also let you minimize power consumption by using all of the capabilities of the device hardware. we detecting a Current Location by android GPS API, and then finding a nearest recharge shops around current location, we integrating Google maps in our app,

VIII. Future Scope

next work with this system is that it can improve the accuracy of scanner and can add more features like sending the message via this system.

IX. Conclusion

From this paper one is able to understand how tesseract engine chops the scanned image and separate the digits and process it as well as understand how location of vendor is found on GPS.

References

[1].S.V. Rice, F.R. Jenkins, T.A. Nartker, "The Fourth Annual Test of OCR Accuracy, Technical Report"95- 03, Information Science Research Institute, University of Nevada, Las Vegas, July 1995.
[2].R.W. Smith, "The Extraction and Recognition of Text from Multimedia Document Images", PhD Thesis, University of Bristol, November 1987.
[3].R. Smith, "A Simple and Efficient Skew Detection Algorithm via Text Row Accumulation", Proc. of the 3rd Int. Conf. on Document Analysis and Recognition (Vol. 2), IEEE 1995, pp. 1145-1148.
[4].P.J. Rousseeuw, A.M. Leroy, "Robust Regression and Outlier Detection", Wiley-IEEE, 2003.
[5].S.V. Rice, G. Nagy, T.A. Nartker, "Optical Character Recognition: An Illustrated Guide to the Frontier", Kluwer Academic Publishers, USA 1999, pp. 57-60.
[6].P.J. Schneider, "An Algorithm for Automatically Fitting Digitized Curves", in A.S. Glassner, Graphics Gems I, Morgan Kaufmann, 1990, pp. 612-626.
[7].R.J. Shillman, "Character Recognition Based on Phenomenological Attributes: Theory and Methods", PhD. Thesis, Massachusetts Institute of Technology. 1974.
[8].B.A. Blesser, T.T. Kuklinski, R.J. Shillman, "Empirical Tests for Feature Selection Based on a Pscychological Theory of Character Recognition", Pattern Recognition 8(2), Elsevier, New York, 1976.
[9].M. Bokser, "Omnidocument Technologies", Proc. IEEE 80(7), IEEE, USA, Jul 1992, pp. 1066-1078.
[10]. [10] H.S. Baird, R. Fossey, "A 100-Font Classifier", Proc. of the 1st Int. Conf. on Document Analysis and Recognition, IEEE, 1991, pp 332-340.
[11]. G. Nagy, "At the frontiers of OCR", Proc. IEEE 80(7), IEEE, USA, Jul 1992, pp 1093-1100.
[12]. The Tesseract open-source OCR engine http://code.google.com/p/tesseract-ocr