Optical Character Recognition for Running C Code

Upendra Mishra[1], Shiva Panwar[2], Deeksha Upadhyay[2], Kamal Deep Gupta[2]

Abstract:'Tesseract' is an OCR (optical character recognition) engine for various operating systems. In the following paper, using the concept of 'Tesseract', a new code runner has been proposed, named as, which can also be used to run handwritten codes. An Android application is developed which enables a person to take a picture of a piece of code by an android phone and within seconds, the output of the code is displayed on the screen. Earlier, it has been observed that the OCR engine works only for the printed text with good accuracy. But the newly developed code runner helps in detecting handwritten codes by customizing and training the system. This code runner reduces the problem and saves the time for the coders by directly clicking the picture and getting the output simultaneously on the mobile screen itself.

Index Terms: Optical Character Recognition, Tesseract, code runner, artificial intelligence, feature extraction, optical scanning, segment extraction, binarization.

1. INTRODUCTION

Over the last five decades, machine replication of human functions has successfully achieved a lot with the help of pattern recognition technology and artificial intelligence. Numbers of inventions are done in this field but still it fails to compete with the reading and recognising abilities of human being. OCR (Optical Character Recognition) is one device that helps in detection and extraction of characters but the performance of the engine is directly proportional to the quality of the input provided. OCR is a process of converting the image of the inputted text into the digitised readable form. Early implementations of conversions in this domain can be traced back to 1913. A number of OCR engine types have been constructed till date. Among these different types of OCR engines, the structural, neural network and feature type of OCRs are most commonly used. The most popular OCR engine used is the Tesseract OCR. A major drawback of thisOCR engine is that it fails in the case of poor alignment of input codes or handwritten codes given as input. The new code runner designed by our team helps us to deal with this drawback to some extent. In simple words, the new code runner allows the user to take a snap of the handwritten code with the help of an android phone and then run it accordingly, providing the output on the mobile screen itself.

The language for which this code runner is working is C language as it has simpler syntax and less vocabulary as compared to other programminglanguages. Therefore, it is easier to detect keywords and functionalities in C language. Missing declaration of any variable type or missing of commas, semicolon etc. will also be detected in compiler error while running the code with this code runner.

2. LITERATURE SURVEY

a) Brian M. Gonzalez – "Iris: A Solution for Executing Handwritten Code"[9]

- In his paper, he presented a different approach for execution of handwritten codes using a self-developed application 'Iris'.
- Iris is designed in a way that it has the ability to process an image of handwritten codes and return the result to the user.
- Various scripts are designed and incorporated in the application so that it can efficiently match it with the handwritten code presented to it.
- Further, he tries to analyse the accuracy of his application using codes in different handwritings (Figure 1).

1. *Mr. Upendra Mishra is currently working at the rank of Assistant Professor in Information Technology Department of IMS Engineering College UnderAbdul Kalam Technical University, Uttar Pradesh, India. Email id:- upendra.mishra@imsec.ac.in*
2. *Shiva Panwar, DeekshaUpadhyayand Kamal Deep Gupta are currently in their final year, pursuing Bachelors of Technology in Information Technology from IMS Engineering College under Abdul Kalam Technical University, Uttar Pradesh, India. Email id:- panwarshiva194@gmail.com.*

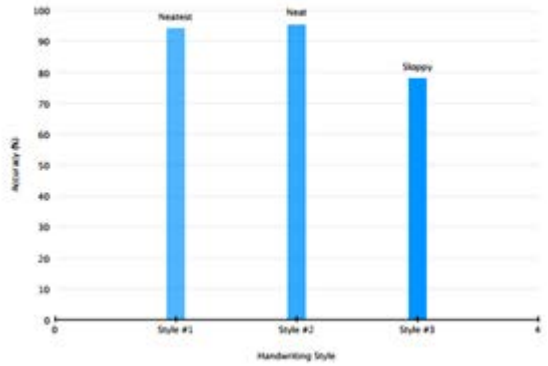


Fig.1. Tesseract accuracies from three different handwriting variations.

b) DewiNasienet. al. – "The Heuristic Extraction Algorithms for Freeman Chain Code of Handwritten Character" [13]

- Handwritten code recognition (HCR) is classified into three different stages – Pre-processing, Feature Extraction and Classification. In this paper, the author only deals with "Pre-processing" (to produce a clean character image that can be used efficiently in the next stage) and "Feature Extraction" (to remove redundancy of data for efficient use in the classification stage).
- A representation technique of pattern is required to move from pre-processing stage to feature extraction stage. In this paper, Freeman Chain Code (FCC) is selected and is used to represent the character.
- The study about FCC construction using one single route and minimizing the FCC length has been limited and therefore, 'Heuristic methods' are adopted here to minimize the length of chain code. Two algorithms name 'Recognition based' and 'Enumeration based' come under this method.
- The route length and the computation time are selected to compare the two algorithms. It has been observed that the randomized-based algorithm is faster than the enumeration-based algorithm as the latter has to consider all branches in route length, which are unvisited, visited and taboo.

c) IssamBaaziet. al. – "An Omnifont Open-Vocabulary OCR System for English and Arabic" [3]

- The paper presents an OCR system for English and Arabic languages with an omnifont, unlimited vocabulary.
- In this paper, they try to build a single model OCR system for multi-style data such as plain and italic.
- The OCR system built for the study is based on Hidden Markov Models (HMM). The paper states two of the several benefits of the system that is- there is no requirement of segmentation at word and character levels and secondly, and most importantly, the same system can be used for different languages with little or no modification i.e. the system is language independent.
- The paper showed how initial high error rates on italics while using a single system can be minimized by balancing the training between the two styles in the HMM model. Mathematical and Empirical equations are effectively used to tackle this problem.
- Their future prospects include - Incorporating other language systems such as Chinese OCR and to check the efficiency of the model using noisy/degraded data.

d) SushruthShastry – ""i" - A novel algorithm for Optical Character Recognition (OCR)"[15]

- The OCR 'i' presented in the paper is a simple, font and size independent and a high speed system. It is based on a unique segment extraction technique.
- The main catch of this methodology is that, it does not use any databases of image matrices to recognize alphabets, but it has its own unique algorithm instead.
- To check the accuracy of this algorithm, it has been implemented in MATLAB- Version 7.14.0.739 build R2012a on a test set of 500 images of text. For three font families namely Arial, Times New Roman and Courier New, an accuracy of 100% has been obtained.
- The paper states that the algorithm is the first 'no training type OCR' which has the added advantage of speed, power and memory and it can be effectively used as a kernel within a complete OCR solution for recognition and alignment of alphabets.

e) Wenxiao Du – "Code Runner – Solution for Recognition and Execution in Handwritten Code" [6]

- In his paper, he introduces an application "Code Runner" which helps in recognition and execution of handwritten codes. He states that the problem lies in the poor alignment and varying characters in the code.
- He first customized Tesseract, an OCR engine, to execute his own handwriting and further applied various image processing methods and text-level post-processing algorithms to enhance the system's capability.
- He then tabulated the accuracy percentages of the application when images of the same code are being clicked from different angles.
- Some of the issues pertaining to this application is – Poor efficiency against rotation, recognition limited to his own handwriting due to lack of training data, generalizing to other programming languages.

3. PROPOSED OCR SYSTEM

a) Basic Working of OCR:

- The main component of the typical OCR is the optical scanner. With the help of optical scanner, the digitized form of the document is achieved.
- The text region is then segmented by pre-processing and eliminating the noise in the symbols. This step is called Feature Extraction.
- The next step comprises of comparing the extracted features with the symbols obtained through a previously learning phase.
- Finally, with each character recognition, a word is detected which in turn results in the complete recognition of the code.

b) System Overview:

Since the training of OCR will require high computation, therefore, it is not feasible to run the complete code on the mobile phone. Client-server model is generally used for working on different platforms. But, in this code runner, instead of using the server, the task is performed on localhost. The client side is an android phone which is responsible for clicking and sending the image and the local host is mainly responsible for processing of the code that is captured.In OCR, a database is created at the back end. In all these efforts, recognition of only one language is attempted. The approach taken is almost similar to those of references. [8][10][11][12].

c) Working of Designed Code Runner:

1. An android phone is used to capture and upload the image (Fig. 1).
2. **Optical scanning**: Through this process a digital image of the original document is obtained(Fig. 2). This is where the pre-processing of image is done. A transport mechanism along with a sensing device is used to convert light intensity into gray scale.An appropriate algorithm is then used to threshold the gray level image to get a binary image.The resulting image may contain certain amount of noise. Therefore, methods such asbinarization, small region removal, morphological opening, filling and thinning element are used to remove the noise.

a) **Binarization**: Initially, the RGB image is converted to gray scale. Adaptive binarization is then applied to this image using Otsu's method.

b) **Small Region Removal**: Dust on camera is the main reason for producing noise on the image. To remove this noise, region labelling is done and the threshold is selected such that the punctuations are not falsely removed.

c) **Morphological Opening**: Morphological opening is done so as to smooth the boundaries of the characters on image and to make them look more realistic.

3. **Segment Extraction**: Segments are parts of image on which proposed algorithm works. The whole image is divided into number of segments.
4. **Feature extraction**:In OCR, we usually face problem of recognising the whole page of code, so there is no obvious way of defining a feature vector as a function of some independent variable [2]. At this stage, the line of text is used for training and recognition. This technique is divided into 3 parts:
   - Selecting and distributing points.
   - Use of the most appropriate algorithm.
   - Analysing characters.
5. **Training OCR**: Training of OCR is the most

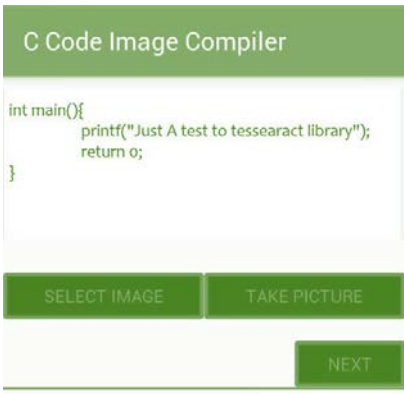


Fig.3. Code obtained after clicking a picture of the code.

important step involved for recognition of characters. Instead of using Internet, here our own handwriting is used to train the system. For each character, more than 5 set of different fonts and styles are tried. A box file is used for training image which contains all corresponding images and coordinates.

6. **Post-processing**: A heuristic algorithm is used on the retrieved data for collecting the complete code and sending it back to the user for manual adjustment as no recognition system can ever result with 100% correct data.
   After the complete correction of code, with accuracy by the user, the code is then sent back to the server (Fig. 4).Results show that there is more accuracy when there is no rotation or distortion of image.
7. **Compilation& execution**:After manually modified by user, ideally everything should be correct by now. The client then sends the text back to the server. The server compiles and executes the code and passes to the client for display.

4. FIGURES AND GRAPHS

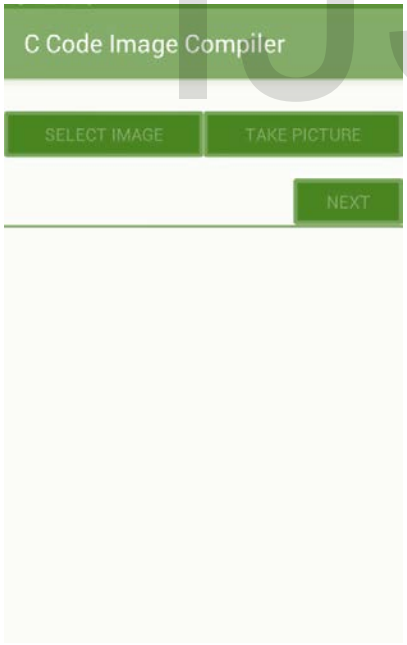
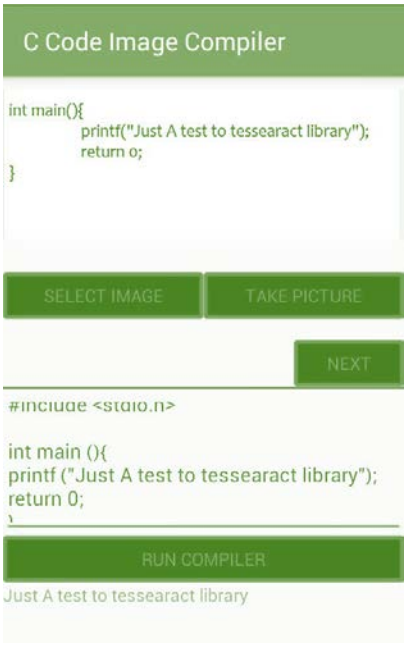


Fig. 2.How the application features on an android phone.



5. CONCLUSION AND FUTURE WORK:

Fig. 4.The code is sent back from the server for manual editing, if any.

1. In general, this code runneris efficient enough in detecting the actual code and then providing the output. Still, there are drawbacks that can be improved in the future work.
2. One way to tackle the poor robustness against the rotation of image while taking the snap is to find the centroid of the image detected and then

to use a straight line to fit them. In future, we aim at finding other ways to solve this issue.

3. We have used C language because of its easy grammar and less vocabulary. As this code runner is restricted to one language, generalising it for other programming languages is an important task that is pending. One peculiar challenge here is the recognition of indentation for languages like python.
4. Training of OCR can also be done using handwriting of arbitrary people for better accuracy.

6. REFERENCES:

[1] Al-Badr, B., Mahmoud, S., "Survey and Bibliography of Arabic Optical Text Recognition," Signal Processing, Vol.41, No. 1, 1995, pp 49-77.

[2] Bassi, I., et al, "An Omnifont Open-Vocabulary OCR System for English and Arabic", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 6, June 1999.

[3] Du, W., "Code Runner: Solution for Recognition and Execution of Handwritten Code", Department of Electrical Engineering, Stanford University.

[4] Eikvil, L., "Optical Character Recognition", Norsk Regnesentral, Oslo, Norway, Rep. 876, 1993.

[5] Elms, A.J., Illingworth, J., "Modelling Polyfont Printed Characters With HMMs and a Shift Invariant Hamming Distance," International Conference on Document Analysis and Recognition, Montreal, Canada, 1995, pp. 504-507.

[6] Gonzalez, Brian, M., "Iris: A Solution for Executing Handwritten Code.",University of Agder, July 2014.

[7] Kaltenmeier, A., Caesar, T.,et al, "Sophisticated Topology of Hidden Markov Models for Cursive Script Recognition", International Conference on Document Analysis and Recognition, Tsukuba City, Japan, 1993, pp. 139-142.

[8] Kornai, A., "Experimental HMM-Based Postal OCR System," International Conference on Acoustics, Speech, Signal Processing, Vol. 4, Munich, Germany, 1997, pp. 3,177- 3,180.

[9] Mohamed, M., Gader, P., "Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modelling And Segmentation-Based Dynamic Programming Techniques," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 5, May 1996, pp.548-554.

[10] Nasien, D., et al, "The Heuristic Extraction Algorithms for Freeman Chain Code of Handwritten Character", International Journal of Experimental Algorithms, (IJEA), Volume (1): Issue (1).

[11] Shastry, S., et al, ""i" - A novel algorithm for Optical Character Recognition (OCR)", IEEE Conference – 2013.

[12] Thong, E., "Codeable: Generating Compilable Source Code from Handwritten Source Code" EE 368 Course project.