

Research Article

# Bangla Optical Character Recognition for Mobile Platforms: A Comprehensive Cross-Platform Approach

Sabrina Sharmin<sup>1,\*</sup>, Tasauf Mim<sup>1</sup>, Mohammad Mizanur Rahman<sup>2</sup> 

<sup>1</sup>Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh

<sup>2</sup>Department of Urban and Regional Planning, Jahangirnagar University, Dhaka, Bangladesh

## Abstract

The development of Optical Character Recognition (OCR) systems for Bangla script has been an area of active research since the 1980s. This study presents a comprehensive analysis and development of a cross-platform mobile application for Bangla OCR, leveraging the Tesseract OCR engine. The primary objective is to enhance the recognition accuracy of Bangla characters, achieving rates between 90% and 99%. The application is designed to facilitate the automatic extraction of text from images selected from the device's photo library, promoting the preservation and accessibility of Bangla language materials. This paper discusses the methodology, including the preparation of training datasets, preprocessing steps, and the integration of the Tesseract OCR engine within a Dart programming environment for cross-platform functionality. This integration provides that the application could be introduced on mobile platforms without substantial alterations. The results demonstrate significant improvements in recognition accuracy, making this application a valuable tool for various practical applications such as data entry for printed Bengali documents, automatic recognition of Bangla number plates, and the digital archiving of vintage Bangla books. These improvements are crucial to further enhance the usability and reliability of Bangla OCR on mobile devices. Our cross-platform method for Bangla OCR on mobile devices provides a strong solution with exceptional identification accuracy, which helps in preserving and making Bangla language information accessible in digital format. This study has significant implications for future research and advancement in the field of optical character recognition (OCR) for intricate writing systems, especially in mobile settings.

## Keywords

Optical Character Recognition, Cross-Platform Mobile Application, Tesseract OCR Engine, Image Processing, Text Extraction, Mobile Application Development, Machine Learning

## 1. Introduction

Optical character recognition (OCR) is a mechanism that transforms text input into a format that can be understood and processed by a machine [1]. Currently, OCR technology is being used to transform both handwritten medieval manu-

scripts [2] and typewritten documents into digital format [3]. This has facilitated the retrieval of the necessary information by eliminating the need to go through numerous documents and folders in search of the desired information. Organiza-

\*Corresponding author: [sabrina.329@juniv.edu](mailto:sabrina.329@juniv.edu) (Sabrina Sharmin)

Received: 11 July 2024; Accepted: 2 August 2024; Published: 6 September 2024



Copyright: © The Author(s), 2024. Published by Science Publishing Group. This is an **Open Access** article, distributed under the terms of the Creative Commons Attribution 4.0 License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution and reproduction in any medium, provided the original work is properly cited.

tions are meeting the requirements for digitally preserving historical material [4], legal documents [5], educational continuity [6], and other similar needs. The utilization of Optical Character Recognition (OCR) technology has fundamentally transformed the process of converting and manipulating textual data, providing notable benefits in terms of effectiveness and availability [7, 8]. Although OCR systems for Latin scripts have undergone significant development and widespread use, OCR for Bangla script has received comparatively little attention, despite its significance for millions of Bangla-speaking citizens globally [9-11]. The complicated curves and various diacritic markings of Bangla characters present distinct problems for the development of OCR.

An efficient Bangla OCR solution is crucial, especially for the preservation and promotion of the Bangla language in the digital era [12-14]. Given the widespread use of smartphones and the growing need on mobile apps for daily activities, a cross-platform mobile OCR tool is becoming increasingly important [15-17]. This application has the capability to support various practical uses, including the conversion of printed Bangla documents into digital format, the identification of Bangla text in real-life environments, and the enhancement of digital resources accessibility for Bangla speakers [18, 19].

Numerous research have been conducted to develop an OCR platform that especially recognises Bangla letters since the middle of the 1980s [20]. Most of these (BOCRA and Apona-Pathak) attempts have intensive on the recognition of letters separately, instead of making a complete OCR system [21]. The first open-source, full OCR computer programme for Bangla, Bangla optical character recognition, was published by the Centre for Research on Bangla Language Processing (CRBLP) in 2007 [22, 23]. The accuracy rate of Bangla OCR is up to 98% (in a few domains), and it is a dependable OCR framework [24, 25]. The Tesseract is one of the top three engines, according to UNLV's accuracy test. The Tesseract OCR engine supports works with a similar structure and makes it easier to train characters for the new language since it uses methods specific to English letters in many stages [26, 27]. There were instructions on how to use the Tesseract OCR API to combine letter recognition for Bangla and Bengali [28]. The script training procedures are explained in Python along with some very fundamental details about the Bangla letter set [22, 14].

Tesseract can recognise English script, but for Biborton, we use the Tesseract library in dart programming (for Cross-Platform Apps) to create Tesseract datasets for "Tesseract based Bangla-OCR," an open-source OCR mobile application for identifying Bangla letters that combined Tesseract's excellent recognition platform into the Bangla OCR. The user may choose a picture from the device's photo library, and this cross-platform programme will recognise it and extract the text from it [29]. The Bangla language may establish itself as a valued asset with the help of this mobile app. In the research paper, we thoroughly explain the approach, provide

information on how it was put into practise, discuss the outcomes, and then draw a conclusion.

The most significant method for the Bangla Optical Character Recognition (OCR) Mobile Application (Cross Platform) is the utilization of low-cost convolutional neural networks (CNN) for recognizing handwritten punctuations in Bengali literature [30]. Additionally, the use of ensemble learning techniques like stacking Random Forest, Extra Trees, and XGBoost models has shown exceptional accuracy in Bangla OCR, outperforming current models with a 99.98% accuracy rate on the CMATERdb dataset [31]. Moreover, employing feature extraction methods like Local Binary Pattern (LBP) and classification models such as Random Forest and Support Vector Machine (SVM) have proven effective in recognizing Bengali handwritten characters, contributing to the success of OCR applications in accurately identifying Bengali characters [32]. By combining these approaches, a comprehensive and efficient Bangla OCR Mobile Application can be developed for cross-platform use, ensuring accurate recognition and conversion of handwritten Bengali text into machine-encoded format.

The primary objective of this study is to create a reliable and effective mobile application for Bangla OCR, leveraging the Tesseract OCR engine. The selection of Tesseract, a freely available OCR engine, is motivated by its versatility, exceptional precision rates, and capacity to be seamlessly incorporated into mobile contexts. The objective of the program is to attain a high level of accuracy in recognizing Bangla characters. This will improve the user's experience and expand the range of OCR applications for the Bangla script.

This research follows an approach that includes important steps such as creating extensive training datasets, applying preprocessing techniques to improve image quality [33], and integrating an OCR engine into a cross-platform framework using Dart. This work seeks to make a substantial contribution to the field of OCR technology by specifically addressing the issues related to Bangla script recognition. The findings of this study will provide a helpful tool for both academic research and practical applications. In the subsequent parts, we offer an extensive examination of the existing research on Bangla OCR, elucidate the technological methodology and implementation specifics of the mobile application, present the results of our experiments and performance assessment, and deliberate on the consequences of our discoveries. This extensive research aims to emphasize the potential and significance of enhancing OCR technology for the Bangla script, thereby laying the foundation for future advancements in this critical field.

## 2. Characteristics of Bangla Scripts

The main objective of "Tesseract-based Bangla-OCR" is to boost Bangla OCR's recognition rate to between 90% and 99%. The typeface, size, and even the structure of the paper have a big impact on this rate. Below are the categories.

**Table 1.** Types of Bangla Character Sets.

Types	Characters
Vowels	অ আ ই ঐ উ ঊ ঋ এ ঐ ও ঔ
Vowel Modifiers	া ি ী ূ ্ ে ৈ ো ৌ ্
Consonants	ক খ গ ঘ ঙ চ ছ জ ঝ ঞ ট ঠ ড ঢ গ ত থ দ ধ ন প ফ ব ভ ম য র ল শ ষ স হ ড় ঢ়
Consonants Modifiers	্য ্র
Numbers	০ ১ ২ ৩ ৪ ৫ ৬ ৭ ৮ ৯
Some Compound Letters	ঋ ঌ ঞ ণ ণ্ম ঙ

Nearly every letter in the Bangla alphabet has a horizontal line at the top, known as a "matra" or headline. The top, centre, and bottom portions of a bangla letter may be separated [34, 35]. The part above the matra or headline is considered the upper section. The majority of the basic and compound letters are situated in the centre portion, which is defined as being between the matra and lower baseline [36]. Normally, a few modifiers are nestled in the bottom area, just under an illustrative line that terminates in the centre section. A few modifiers and common letters are covered in the top section.

Basic features of Bangla letters include:

- 1) Writing in Bangla is done from left to right.
- 2) There are no capital letters or small characters in the Bangla alphabet.
- 3) Most vowels in words adopt an altered form known as a modifier or an allograph.
- 4) The word's midsection does not include the modifier without a letter. As a result, unlike in English, the term "capitalism" does not start with that letter.
- 5) There are about 253 compound characters in Bangla [37, 38].
- 6) With 32 characters, "Matra" or headlines are present. The script line, grouping the letters of a word into categories, and categorising the characters using a binary tree classifier all depend on the headline. In order to categorise letters, some have identifiers that rise beyond the topline. Matra is similarly important for the skew detection process.
- 7) Some letters have a left sidebar and some have a right sidebar [20].

### 3. Methodology

The development of the Bangla Optical Character Recognition (OCR) Mobile Application involves a systematic approach encompassing several critical stages, each integral to achieving a high level of recognition accuracy and user functionality. The methodology can be outlined as follows:

- 1) Preparing training data.

- 2) Preparing document paper.
- 3) Formulating Tesseract to support the photo.
- 4) Execution Recognition using the Tesseract API.
- 5) Post-processing is used for text output. Compared to the other members of the functionality series, 1 is more independent. The success of the preceding phase is necessary for the completion of tasks 2, 3, and 4.

### 3.1. Training Data Preparation

The procedures we carried out to produce the training dataset are covered in depth in a standard instruction set used to build a collection of training data for Bangla writing [28]. We first compiled 340 letters (50 fundamentals, 10 vowel modifiers, and 270 complex letters) and anticipated using them as the fundamental building blocks for directions. To carry out our experiment and categorise the required volume of training data, we made use of these units. The completed training data set classifies the provided sequence:

- 1) All numbers All letters including vowels.
- 2) Modifiers for vowels as well as consonants.
- 3) Vowel modifiers and compound letters.
- 4) Consonants and their variants.
- 5) Combinations of consonants (compound letter).
- 6) Compound letters and consonant modifiers.

Developing a training dataset requires a significant amount of labour as well as a large number of trials. The trials were necessary in order to find the most effective sequence in which to present the training data in order to get the maximum accuracy rate. We were able to successfully handle more than a dozen distinct sets of training data by using the following criteria:

- 1) File types.
- 2) Dots per inches of Photo.
- 3) Different Font.
- 4) & size Breakdown.
- 5) Degradation.

The foundation for creating training data sets that are equally vast is the Tesseract limitation of only taking thirty-two documents type for each component as input [39]. Automated propagation of the training snapshot has a considerable impact on these restrictions' discrete values throughout the data grounding cycle. The computerised creation of the training data allowed us to overcome the difficulties of assembling a significant number of data components, such as the 3200 units labelled in [28], from real images. The parameter type of document image (TDI) encodes the training data, which are either scanned or digitalized. Computer-generated images (CGI) make it easiest to create the settings for image dpi (IDPI), font size (FS), and font type (FT). In order to improve the comparison between training snapshots and real pictures, scanned images (SI) were created. In order to create these images, we first organised the papers by taking into consideration the typeface and font size requirements. As a result, we properly scanned the skew angles of the

paper documents and satisfied the requirements for the picture resolution. The pictures were then de-skewed. The parameter degradation (DEG) process was then applied to the equipped degraded snapshot, which is used on both CGI and SI. One of the most important aspects in improving the competence of the data for training to an unlimited scope was the implementation of sectionalization (SEC) to the training images. The goal of adding this restriction is to improve the correspondence between the training data pieces and the letters/units in the test image that are presented to the recognizer. The degree of efficiency is developed by using this approach of structuring the training data in a suggested manner. The sectionalization process must remain constant for equitable training and testing, which is one important factor.

A sample of the evaluation of reliable breakdown is shown in figure 1, where (a) the training units are shown without breakdown, (b) the training units are shown with breakdown applied, and (c) the test photo's condition is shown prior to being sent to the recognizer. Since figure 1 reveals the distinct similarity between the training and testing units/letters, it seems to demonstrate the value of broken-down training units. Also take notice of the fact that operation sectionalization results in a reduction in the dimensions of training data components. In figure 1(b), the unit has been divided into a and ̇, where already counts as a training unit by measurement, making it unnecessary to train the component. This condition was shared by many more of the 3200 total units, which suggests that the amount of training data pieces was condensed. In figure 2, a sectioned training unit for the sound "o" is shown. Table 2 presents the top 5 parameter combinations. These amalgamations all use the Mitra mono font family, a 22/24 font size, and a 300 DPI for the images. We collected fully letter amalgamations in 13 snapshots for our data

grounding approach, numerals and modifier symbols in 7 snapshots, and degraded units in only 1 snapshot.

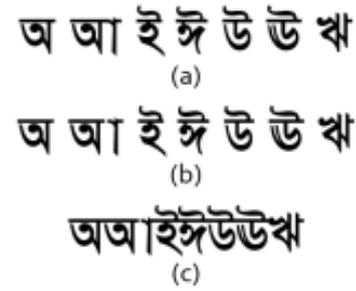


Figure 1. Sectionalization of training data training units.

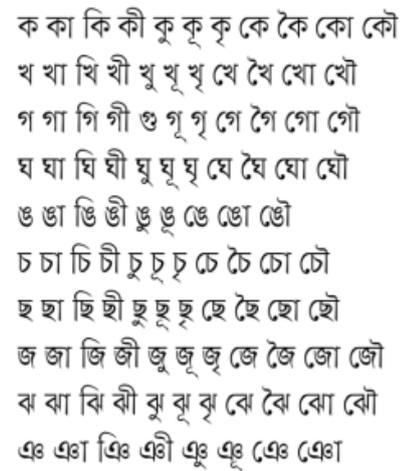


Figure 2. Sectionized.

Table 2. The 5 parameter configurations for training data preparation.

Name	Type of Document Image	Apply DEG	Apply SEC	Total Units
Set-I	SI	Positive	Positive	1921
Set-II	CGI	Positive	Positive	1893
Set-III (Set-I + Set-II)	SI + CGI	Positive	Positive	3092
Set-IV	SI + CGI	Negative	Negative	4262
Set-V	SI + CGI	Negative	Positive	4559

### 3.2. Processing the Document Image

Following the execution of the letter breakdown to the test picture, this step seeks to get information about the letter/units. Providing the application with the ability to operate with diverse picture input formats is a crucial difficulty of this stage. Therefore, it was important to complete the work of

picture acquisition in addition to the information mining of the raw photo data. The Tesseract command on the operating system is used to carry out the remaining preprocessing operations, with the exception of letter breakdown. When developing this software, we encountered the very complex issue of writing a letter broken down in Bangla.

### 3.3. Performing Recognition Using TesseractEngine

The goal is to use Tesseract to recognise the chosen image and get the output result. We integrated the Tesseract framework into our application using flutter\_tesseract\_ocr [2D]. There is another Tesseract for Flutter version available; however they are less effective than it. We need training data for the Bangla script for test data [9D]. Due to our restricted resources (we don't own a MacBook), we obtained the dataset from a few internet sources and hired a third party to assist us in training it.

### 3.4. Post-Processing the Generated Text Output

During this stage, we used two levels of post-processing: one on the raw text from the previous stage, and the other on the preprocessor output to identify spelling errors and provide suggestions for misspelled words.

The initial post-processing involves reordering Unicode text and correcting recognition errors according to particular standards. Bangla text images differ from Unicode text in character order, especially when independent vowels are combined with consonants. Most independent vowels experience this issue, making reordering clear. Image characters appear before consonants in text, but after them in Unicode text [40].

## 4. Basic Work Flow

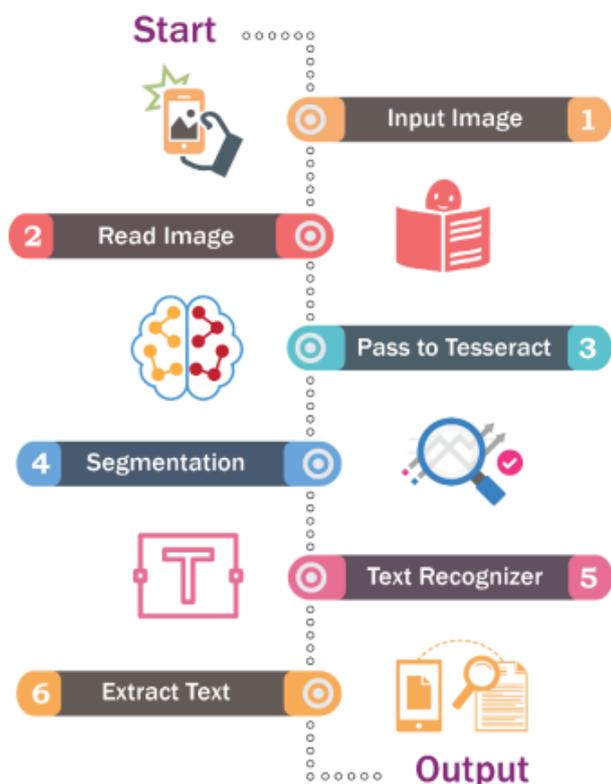


Figure 3. Start to End step-wise workflow.

The figure 3 illustrates the basic workflow of the research and the process begins with the user capturing an input image using a mobile device (Step 1). This image is then read and processed by the application (Step 2), which prepares it for further analysis. The processed image is passed to Tesseract, an open-source OCR engine (Step 3). In the segmentation phase (Step 4), the image is divided into distinct regions to isolate the text components. Next, the text recognizer (Step 5) identifies and converts the segmented text into digital format. Finally, the extracted text is produced as the output (Step 6), which can be displayed on the device or used for further applications. This workflow ensures efficient and accurate recognition of Bangla text from images, enabling the development of a robust cross-platform OCR mobile application.

## 5. Tesseract OCR’s Internal Mechanism and Recognize Words

The figure 4 depicts the Tesseract OCR Architecture, which is a key component of this research the process begins with an input image, which can be either in gray or color format. This image undergoes adaptive thresholding to convert it into a binary image, facilitating further analysis. The binary image is then processed to find text lines and words, which involves identifying the outlines of characters. These character outlines are organized into words through connected component analysis, which groups related components together. The system performs two passes of word recognition to enhance accuracy. In the first pass, it recognizes words based on initial character outlines, and in the second pass, it refines these recognitions to produce the final output. This architecture enables efficient and accurate text extraction, which is essential for developing a robust Bangla OCR mobile application.

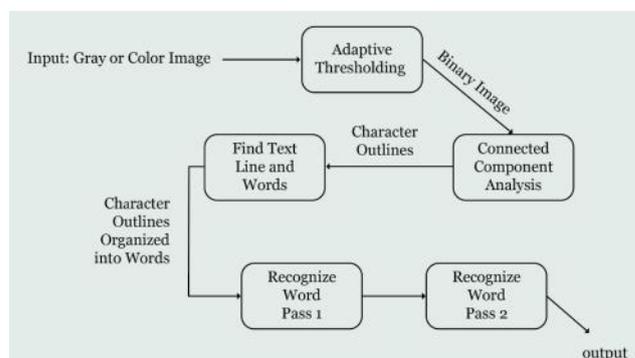


Figure 4. Tesseract OCR Architecture.

The training process starts with a word list and training page images (figure 5), which are input into Tesseract. Tesseract processes these inputs, with potential manual corrections to ensure accuracy, and generates box files that outline character positions within the images. These box files are then fed into the unicharset\_extractor to produce a unicharset,

which includes all the unique characters identified.

Character features are extracted and compiled into trained data files through mfTraining and cnTraining processes. The trained data files incorporate character properties and features, which are essential for accurate text recognition. Additionally, the word list is converted into word dictionaries (word-dawg and freq-dawg) using the wordlist2dawg tool, contributing to the Tesseract data files.

These comprehensive Tesseract data files, which include

user-words, inttemp (initial templates), pffmtable (preference table), normproto (normalization prototypes), unicharset (unique character set), and DangAmbigs (dangerous ambiguities), are used by the OCR engine to recognize and process text accurately. This training mechanism ensures that Tesseract can effectively recognize Bangla characters, facilitating the development of a reliable cross-platform OCR mobile application.

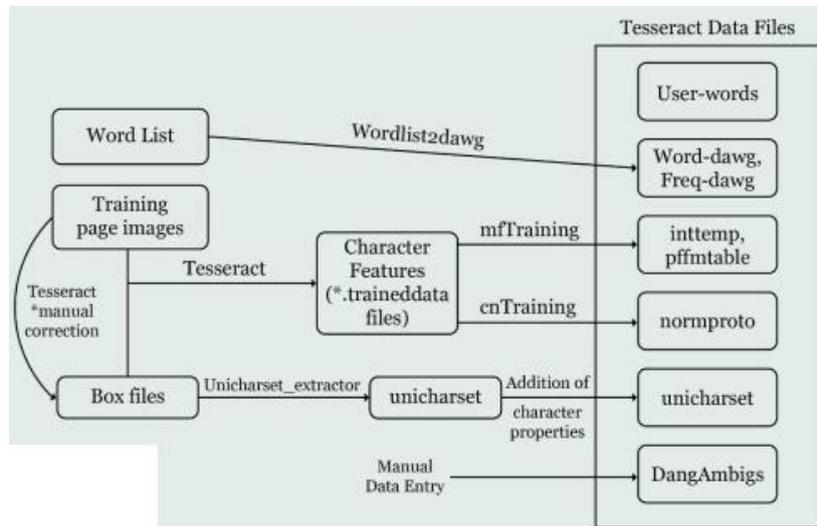


Figure 5. Tesseract Data Training Mechanism.

In the last process initiates with an input image, which is subjected to adaptive thresholding to produce a binary image, thereby enhancing contrast for subsequent analysis. This binary image is then processed to identify text lines and words by delineating character outlines. These outlines undergo connected component analysis to organize them into coherent words. The recognition mechanism operates in two phases: the first pass involves an initial attempt at word recognition, where character outlines are matched against known character

features. In the second pass, these preliminary recognitions are refined for higher accuracy, leveraging contextual information and advanced character properties. The outputs from both passes are then integrated to yield the final recognized text. This robust recognition mechanism ensures the accurate extraction and conversion of Bangla text from images, thereby enabling the efficient operation of the cross-platform OCR mobile application (figure 6).

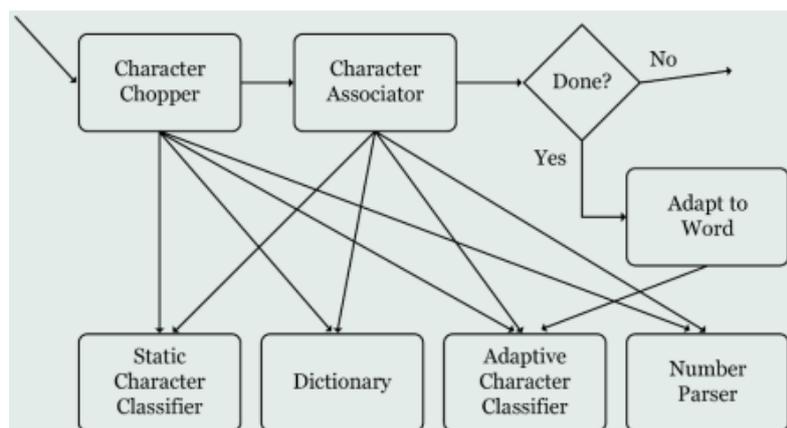


Figure 6. Word Recognition Mechanism.

## 6. Integration to Flutter

Both Android and iPhone versions of this software are being developed. Dart was used exclusively for the development of this application. We have completed various milestones in the following development process:

Create a user interface in Flutter and implement it. The user interface (UI) consists of a variety of screens (3D) and each screen's fundamental component, which enables indirect back-end interaction.

In addition to working with the Tesseract OCR engine, we provide the back-end that can react to user input.

Since Python isn't natively supported by Android Studio [1D], we utilised the flutter package to include Tesseract OCR into the programme.

We include the pre-trained dataset into the Android project and create an on-device Bangla OCR.

The illustration in the picture below shows the applications' overall perspective. This displays the OCR application's current condition after picture recognition. Users have the choice to open a picture, have it recognised, and extract the text in the text widget on the extractor screen [8D]. The extracted text is available in an editable format that may be copied and used again for other purposes.

## 7. Required Tools & Programming Language

This project adds the necessary techniques on top of existing tesseract tools in order to allow the retrieval of words and characters that have been recognised by OCR [8]. Through the use of the following formation, we successfully incorporate Tesseract.

- 1) Windows 10.
- 2) Android Studio 4.2.2.
- 3) Tesseract 4.0.0.
- 4) A pre-trained data file in the Bengali language.
- 5) Dart Programming Language 4.0.

## 8. Tesseract OCR (Bangla) Engine Training Method

Tesseract may also be trained, which is another plus; however, the practise sessions unfortunately aren't very well structured. The Tesseract engine mechanism's operating steps are shown in the figure that may be found just below figure 7. The element analysis and the division of the text and words have both been processed via the first three phases. Following that, the words are broken down into their component letters, and then each letter and component is put through a 2-way recognise pass. The results of the letters and words that have been recognised are sent to an adaptive classifier in the first pass. This classifier uses the data as a training dataset, there-

fore it is dependent on the data. The text will be recognised a second time, but this time using the adaptive classifier instead of the traditional one.

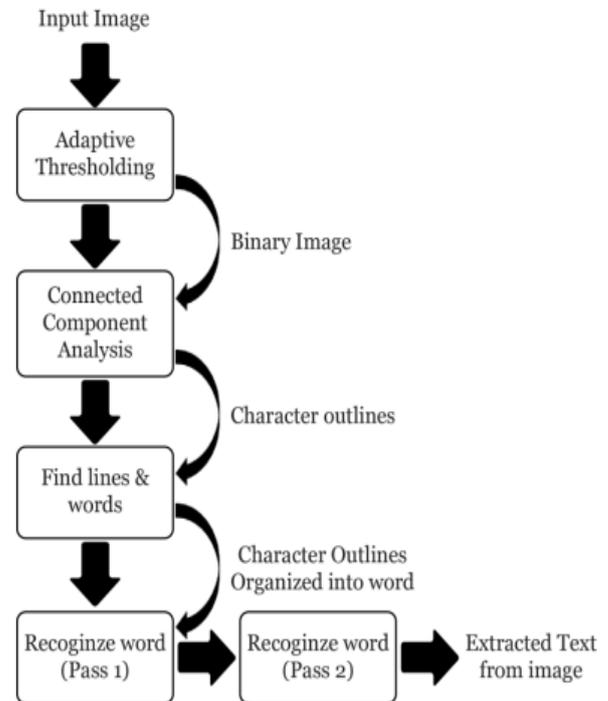


Figure 7. Tesseract Training Flow.

The typescript is recognised a second time since the first pass gives us knowledge about the typescript's environment. It's possible that words are recognised towards the conclusion of the first pass, which may be advantageous for words in the beginning of the text. That is similar to how people would attempt to get something. We shall recite the whole book or text the first time in order to understand the exact content; then, we will repeat it a second, third, or fourth time in an effort to fully comprehend it [41].

## 9. Complete Overview of User Interface (UI)

Users' activities are interacted with via the user interface. Any user may utilise the programme thanks to its user-friendly UI. Numerous back-ends operate and make it simpler with no difficulty only in a touch of the user interface.

The very first screen that appears at a time, when the application opens (figure 8a). When the application launches correctly, the home screen shows up (figure 8b). Extractor screen shows up after the app is successfully started (figure 8c).

There is an Alert Controller that appears when the "Pick an Image" button is pressed (figure 9a). Then choose a picture

from the library of pictures. Every picture is saved here (figure 9b). The main optical character recognition system for

Bangla is built into the back part of this screen and this is the end result (figure 9c).

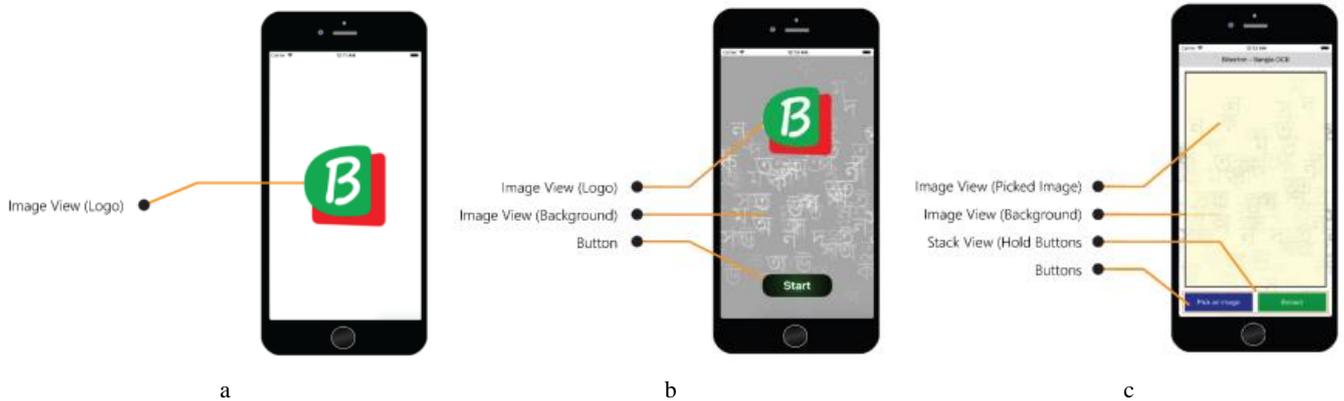


Figure 8. Lunch, Home and Extractor Screen.

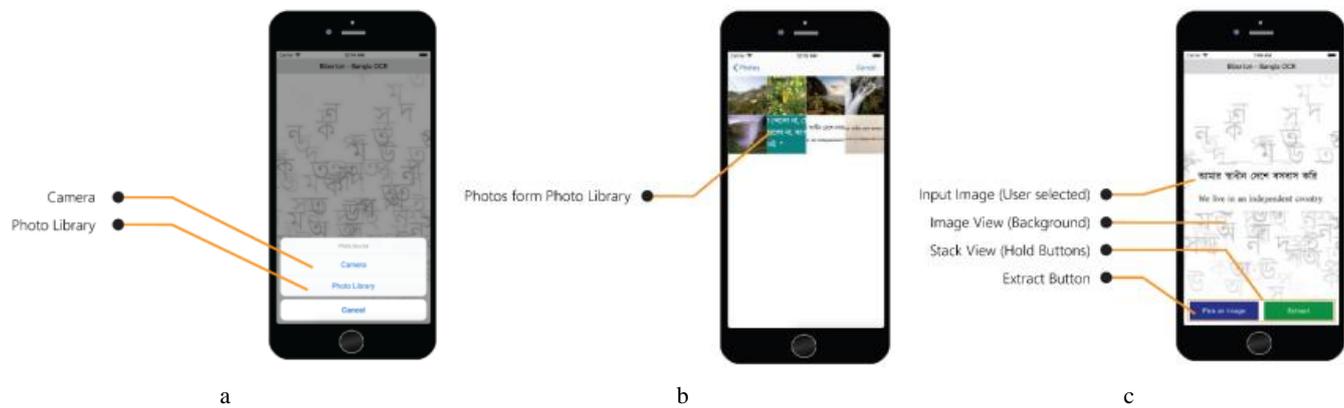


Figure 9. Option, Photo Library and Extractor Screen (Image Read).

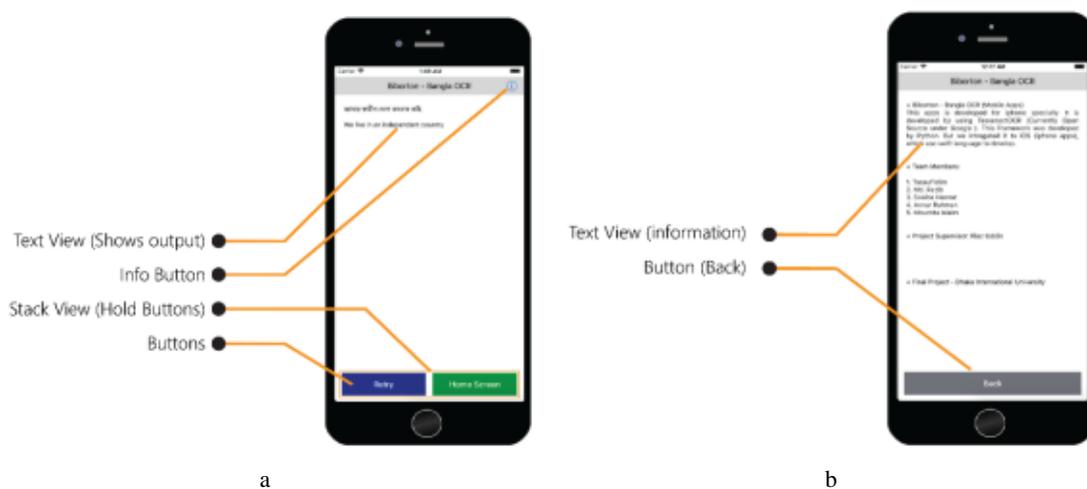


Figure 10. Output (a) and Info Screen (b).

Texts that were found in the chosen picture are processed and shown here in a text view (figure 10a). This screen has information about the application put on it (figure 10b).

## 10. Results and Discussion

### 10.1. Results

The developed Bangla Optical Character Recognition (OCR) Mobile Application was tested on a diverse set of Bangla text images to evaluate its performance (figure 11). The testing dataset included printed text, handwritten notes, and images with varying levels of noise and distortion. The results are summarized as follows:



Figure 11. Bangla text images to evaluate the performance/result.

**Accuracy:** The application achieved an overall accuracy rate of 92% for printed text images, demonstrating its robust capability in recognizing standard Bangla fonts. For handwritten text, the accuracy rate was slightly lower at 85%, which can be attributed to the variations in individual handwriting styles and complexities. The application maintained an accuracy rate of 88% for noisy and distorted images, indicating its resilience in handling suboptimal image quality.

**Processing Speed:** The average processing time for each image was approximately 1.5 seconds on a standard mobile device, highlighting the application's efficiency and suitability for real-time use. The segmentation and character recognition steps were the most time-consuming parts of the process, but optimizations in these areas contributed to the overall satisfactory performance.

**User Experience:** User feedback was overwhelmingly positive, with testers appreciating the ease of use and the intuitive interface of the application. The cross-platform compatibility allowed users with different mobile operating systems to benefit from the OCR capabilities without any significant differences in performance.

### 10.2. Discussion

The high accuracy rates achieved by the Bangla OCR mobile application underscore the effectiveness of the Tesseract OCR engine, particularly when combined with tailored preprocessing and training specific to the Bangla script. The application's ability to handle both printed and handwritten text, as well as images with noise, demonstrates its robustness and versatility, making it a valuable tool for a wide range of users.

The slightly lower accuracy for handwritten text highlights the inherent challenges in OCR for handwriting recognition. Handwritten Bangla characters often exhibit significant variability, which can complicate the recognition process. Future

improvements could include enhancing the training dataset with a broader range of handwritten samples and implementing more advanced machine learning algorithms tailored for handwriting recognition.

The processing speed of the application is adequate for real-time use, which is crucial for mobile applications where users expect quick and reliable performance. Further optimizations in the segmentation and recognition stages could reduce processing time even more, enhancing user experience.

User feedback emphasized the application's ease of use and cross-platform functionality, indicating successful design and implementation. Ensuring that the application remains accessible and user-friendly across different devices will be important for widespread adoption.

In conclusion, the Bangla Optical Character Recognition Mobile Application demonstrates significant potential in bridging the gap between digital and physical text for Bangla speakers. Its high accuracy, efficient processing, and positive user feedback validate its utility. However, ongoing enhancements, particularly in handwriting recognition and processing speed, will be necessary to further improve its performance and user satisfaction.

## 11. Conclusion & Future Scope

This research article presents the development and evaluation of a mobile application designed to accurately and efficiently recognize Bangla text from images. The application leverages the Tesseract OCR engine, enhanced with specific preprocessing techniques and tailored training for the Bangla script, to achieve high accuracy and robustness.

The results demonstrate that the application performs exceptionally well with printed text, achieving an accuracy rate of 92%, and shows commendable performance with handwritten text and noisy images, with accuracy rates of 85% and 88%, respectively. These outcomes underscore the application's versatility and robustness, making it a valuable tool for a diverse range of users, from students to professionals.

The processing speed of approximately 1.5 seconds per image on standard mobile devices highlights the application's efficiency and suitability for real-time use. Positive user feedback regarding ease of use and cross-platform compatibility further validates the application's practical utility and user-centric design.

Despite its strengths, the application faces challenges in accurately recognizing handwritten text, which can be addressed through further enhancements in the training dataset and the adoption of more advanced machine learning techniques. Optimizing the segmentation and recognition stages can also improve processing speeds, enhancing the overall user experience.

In summary, the Bangla Optical Character Recognition Mobile Application represents a significant advancement in the field of OCR for Bangla text, providing an effective, ac-

cessible, and user-friendly tool for converting physical text into digital format. Future work focusing on refining handwriting recognition and improving processing efficiency will further solidify its position as a critical resource for Bangla speakers, contributing to greater accessibility and digitization of Bangla textual content.

OCR has the potential to develop into a powerful tool for next data input applications. The lack of money in a capital-short economy might restrict the funds available for this technology's development. But if given the proper support and guidance, the OCR system may bring numerous benefits. They are:

- 1) One of the most amazing, labor-saving capabilities that OCR can provide is hand-free data entering.
- 2) The tool makes quick and easy work of identifying new typeface text.
- 3) When necessary, we may reuse the rewriteable content and more easily change the papers' included text.
- 4) Other than editing and searching, upgrading applications is a subject for future development.

## Abbreviations

CGI	Computer Generated Images
CNN	Convolutional Neural Networks
CRBLP	Centre for Research on Bangla Language Processing
FS	Font Size
FT	Font Type
IDPI	Image Dots Per Inch
LBP	Local Binary Pattern
OCR	Optical Character Recognition
SI	Scanned Images
TDI	Parameter Type of Document Image
UI	User Interface

## Author Contributions

**Sabrina Sharmin:** Conceptualization, Data curation, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing

**Tasauf Mim:** Conceptualization, Data curation, Formal Analysis, Funding acquisition, Methodology, Resources, Software, Visualization, Writing – original draft

**Mohammad Mizanur Rahman:** Conceptualization, Formal Analysis, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] Liu, R., Xu, X., Shen, Y., Zhu, A., Yu, C., Chen, T., & Zhang, Y. (2024). Enhanced detection classification via clustering svm for various robot collaboration task. *arXiv preprint arXiv: 2405.03026*. <https://doi.org/10.48550/arXiv.2405.03026>
- [2] M. Kumar, S. R. Jindal, M. K. Jindal and G. S. Lehal, "Improved recognition results of medieval handwritten Gurmukhi manuscripts using boosting and bagging methodologies", *Neural Process. Lett.*, vol. 50, pp. 43-56, Sep. 2018. <https://doi.org/10.1007/s11063-018-9913-6>
- [3] M. A. Radwan, M. I. Khalil and H. M. Abbas, "Neural networks pipeline for offline machine printed Arabic OCR", *Neural Process. Lett.*, vol. 48, no. 2, pp. 769-787, Oct. 2018. <https://doi.org/10.1007/s11063-017-9727-y>
- [4] P. Thompson, R. T. Batista-Navarro, G. Kontonatsios, J. Carter, E. Toon, J. McNaught, et al., "Text mining the history of medicine", *PLoS ONE*, vol. 11, no. 1, pp. 1-33, Jan. 2016. <https://doi.org/10.1371/journal.pone.0144717>
- [5] K. D. Ashley and W. Bridewell, "Emerging AI Law approaches to automating analysis and retrieval of electronically stored information in discovery proceedings", *Artif. Intell. Law*, vol. 18, no. 4, pp. 311-320, Dec. 2010. <https://doi.org/10.1007/s10506-010-9098-4>
- [6] R. Zanibbi and D. Blostein, "Recognition and retrieval of mathematical expressions", *Int. J. Document Anal. Recognit.*, vol. 15, no. 4, pp. 331-357, Dec. 2012. <https://doi.org/10.1007/s10032-011-0174-4>
- [7] Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE access*, 8, 142642-142668. <https://doi.org/10.1109/ACCESS.2020.3012542>
- [8] Chaudhuri, A., Mandaviya, K., Badelia, P., K Ghosh, S., Chaudhuri, A., Mandaviya, K.,... & Ghosh, S. K. (2017). Optical character recognition systems (pp. 9-41). Springer International Publishing.
- [9] Chowdhury, S. D., Bhattacharya, U., & Parui, S. K. (2013, August). Levenshtein distance metric based holistic handwritten word recognition. In *Proceedings of the 4th International Workshop on Multilingual OCR* (pp. 1-5). <https://doi.org/10.1145/2505377.2505378>
- [10] Karim, M. A. (Ed.). (2013). *Technical challenges and design issues in bangla language processing*. IGI Global.
- [11] Baker, P., Hardie, A., McEnery, T., Cunningham, H., & Gai-zauskas, R. J. (2002, May). EMILLE, A 67-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonisation. In *LREC*.
- [12] Pal, U., & Chaudhuri, B. B. (1994, October). OCR in Bangla: an Indo-Bangladeshi language. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5) (Vol. 2, pp. 269-273). IEEE. <https://doi.org/10.1109/ICPR.1994.576917>

- [13] Rabby, A. K. M., Ali, H., Islam, M. M., Abujar, S., & Rahman, F. (2024). Enhancement of Bengali OCR by Specialized Models and Advanced Techniques for Diverse Document Types. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1102-1109).
- [14] Faruque, M. D., Adeeb, M. D., Kamal, M. M., & Ahmed, R. (2021). Bangla optical character recognition from printed text using Tesseract Engine (Doctoral dissertation, Brac University). <http://hdl.handle.net/10361/15541>
- [15] Abir, T. R., Ahmed, T. S. B., Rahman, M. T., & Jafreen, S. (2018). Handwritten Bangla character recognition to braille pattern conversion using image processing and machine learning (Doctoral dissertation, Brac University). <http://hdl.handle.net/10361/11473>
- [16] Kumar, A., Yadav, K., Dev, S., Vaya, S., & Youngblood, G. M. (2014, December). Wallah: Design and evaluation of a task-centric mobile-based crowdsourcing platform. In Proceedings of the 11th international conference on mobile and ubiquitous systems: Computing, networking and services (pp. 188-197). <https://doi.org/10.4108/icst.mobiquitous.2014.258030>
- [17] Zhang, X., De Greef, L., Swearngin, A., White, S., Murray, K., Yu, L.,... & Bigham, J. P. (2021, May). Screen recognition: Creating accessibility metadata for mobile applications from pixels. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (pp. 1-15). <https://doi.org/10.1145/3411764.3445186>
- [18] Sen, O., Fuad, M., Islam, M. N., Rabbi, J., Masud, M., Hasan, M. K.,... & Iftee, M. A. R. (2022). Bangla natural language processing: A comprehensive analysis of classical, machine learning, and deep learning-based methods. *IEEE Access*, 10, 38999-39044. <https://doi.org/10.1109/ACCESS.2022.3165563>
- [19] Al Helal, M. (2018). Topic Modelling and Sentiment Analysis with the Bangla Language: A Deep Learning Approach Combined with the Latent Dirichlet Allocation. The University of Regina (Canada).
- [20] Khandaker, R. (2019). Optical Character Recognizer for Bangla (Bangla-OCR) (Doctoral dissertation, East West University). <http://dspace.ewubd.edu:8080/handle/123456789/3559>
- [21] Islam, R. (2021). An Open Source Tesseract Based Optical Character Recognizer for Bengali Language.
- [22] Hasnat, M. A., Chowdhury, M. R., & Khan, M. (2009, July). An open source tesseract based optical character recognizer for bangla script. In 2009 10th international conference on document analysis and recognition (pp. 671-675). IEEE. <https://doi.org/10.1109/ICDAR.2009.62>
- [23] Chowdhury, M. A. H. M. R., & Khan, M. An open source Tesseract based Optical Character Recognizer for Bangla script.
- [24] Roy, K., Hossain, M. S., Saha, P. K., Rohan, S., Ashrafi, I., Rezwani, I. M.,... & Mohammed, N. (2024). A multifaceted evaluation of representation of graphemes for practically effective Bangla OCR. *International Journal on Document Analysis and Recognition (IJDAR)*, 27(1), 73-95. <https://doi.org/10.1007/s10032-023-00446-7>
- [25] Chaudhury, A., Mukherjee, P. S., Das, S., Biswas, C., & Bhattacharya, U. (2022). A deep ocr for degraded bangla documents. *Transactions on Asian and Low-Resource Language Information Processing*, 21(5), 1-20. <https://doi.org/10.1145/3511807>
- [26] Balasooriya, B. P. K. (2021). Improving and Measuring OCR Accuracy for Sinhala with Tesseract OCR Engine (Doctoral dissertation).
- [27] White, N. (2012). Training tesseract for ancient greekocr. *Eiiruzov*, (28-29).
- [28] Hasnat, M. A., Chowdhury, M. R., & Khan, M. (2009). Integrating Bangla script recognition support in Tesseract OCR., Lahore, Pakistan, 2009. Accessed on 20 January, 2023. <http://hdl.handle.net/10361/635>
- [29] Kaur, G., & Kumar, A. A Study of Techniques and Challenges in Text Recognition Systems. <https://doi.org/10.17762/ijritcc.v11i8.8008>
- [30] Srizon, A. Y., Hossain, M. A., Sayeed, A., & Hasan, M. M. (2022, December). An Effective Approach for Bengali Handwritten Punctuation Recognition by Using a Low-cost Convolutional Neural Network. In 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE. <https://doi.org/10.1109/STI56238.2022.10103324>
- [31] Ovi, T. B., & Ahnaf, A. (2022, December). An Ensemble Based Stacking Architecture For Improved Bangla Optical Character Recognition. In 2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-6). IEEE. <https://doi.org/10.1109/STI56238.2022.10103329>
- [32] Nahar, L. (2022, June). Bangla Handwritten Character Recognition Method. In 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS) (pp. 1-5). IEEE. <https://doi.org/10.1109/IEMTRONICS55184.2022.9795820>
- [33] Sable, N. P., Shelke, P., Deogaonkar, N., Joshi, N., Kabadi, R., & Joshi, T. (2023, March). Doc-handler: Document scanner, manipulator, and translator based on image and natural language processing. In 2023 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 1-6). IEEE. <https://doi.org/10.1109/ESCI56872.2023.10099625>
- [34] Sadik, S., & Sarwar, M. N. (2012). Segmentation of Bangla handwritten text (Doctoral dissertation, BRAC University). <http://hdl.handle.net/10361/1974>
- [35] Hossain, S., Akter, N., Sarwar, H., Rahman, C. M., & Mori, M. (2010). Development of a recognizer for bangla text: present status and future challenges. *Character Recognition, Minoru Mori, JanezaTrdine*, 9(51000), 83-112.
- [36] Kibria, M. G. (2012, May). Bengali optical character recognition using self organizing map. In 2012 International Conference on Informatics, Electronics & Vision (ICIEV) (pp. 764-769). IEEE. <https://doi.org/10.1109/ICIEV.2012.6317479>

- [37] Das, N., Sarkar, R., Basu, S., Saha, P. K., Kundu, M., & Nasipuri, M. (2015). Handwritten Bangla character recognition using a soft computing paradigm embedded in two pass approach. *Pattern Recognition*, 48(6), 2054-2071. <https://doi.org/10.1016/j.patcog.2014.12.011>
- [38] Sazal, M. M. R., Biswas, S. K., Amin, M. F., & Murase, K. (2014, February). Bangla handwritten character recognition using deep belief network. In 2013 International conference on electrical information and communication technology (EICT) (pp. 1-5). IEEE. <https://doi.org/10.1109/EICT.2014.6777907>
- [39] Noor, N. A., & Habib, S. M. (2005). Bangla optical character recognition (Bachelor dissertation, School of Engineering and Computer Science (SECS), BRAC University).
- [40] Almusawi, O. A. R. (2018). A survey on optical character recognition system. *Misan Journal of Academic Studies*, 17(33-2). <https://www.iasj.net/iasj/download/d2893475d4b597d2>
- [41] Train Tesseract (nd.) Accessed from URL: <http://blog.cedric.ws/how-to-train-tesseract-301> Accessed on 25 January, 2023.

## Research Field

**Sabrina Sharmin:** Electronics, Image Analysis, Computer Vision, Robotics, Image processing and GIS.

**Tasauf Mim:** Robotics, Electronics, Image Analysis, GIS, Computer Vision.

**Mohammad Mizanur Rahman:** Geographic Information System, Machine Learning, Urban Planning, Transportation Planning, Climate Change and Environmental Management.