

Voice Assisted Form Filling for the Differently Abled

Shreya Sri Ramasubramanian
Dept of Computer Science
PES University
Bengaluru, India
shreyaram22@gmail.com

Sunit Koodli
Dept of Computer Science
PES University
Bengaluru, India
koodli.sunit@gmail.com

Pranav S Nair
Dept of Computer Science
PES University
Bengaluru, India
pranavsnair2000@gmail.com

Mahah Sadique
Dept of Computer Science
PES University
Bengaluru, India mahasadique@gmail.com

Dr. Mamatha HR
Professor
Dept of Computer Science
PES University
Bengaluru, India mamathahr@pes.edu

Abstract— It is a cumbersome process for the differently abled, especially the visually impaired to fill out forms themselves. The objective of this project is to provide a voice-based medium for automated record entry in the UDID (Unique Disability ID) form, while simultaneously performing real time analysis with high efficiency and accuracy. The goal is to use this technology at kiosks in banks & government offices. This eliminates the need for a helper to fill out the form for the differently abled, giving them a sense of independence.

The technology involves a voice assistance option on the online UDID form, using which the software sends out prompts, i.e., the questions present in the UDID form. To avoid any malpractice, these prompts are sent out only after the User's identity is verified. The User's data is collected in the form of voice input and it undergoes several processes within the software. After the completion of these processes, meaningful data is stored as a dictionary. Finally, the recorded data present in the dictionary is uploaded on to the database, thus completing the process of filling the UDID form.

Keywords— *differently abled, visually impaired, form filling, text to speech, speech to text, OCR, face detection, face recognition, kiosk*

I. INTRODUCTION

Disabled, or Differently abled individuals are those who may have limited physical, mental or cognitive functionality due to a disability. These limitations negatively impact their quality of life and make day-to-day tasks more challenging for the individual.

Filling out forms independently is currently unfeasible and non-viable for the differently abled people, especially the visually impaired. The reason for this is that they are unable to:

1. Pinpoint the form field's locations,
2. Locate where to fill the responses,
3. Figure out the fields present in the lengthy forms,
4. Figure if their responses matches with what was asked in the form.

The use of hardware to assist the visually impaired has always been in use, along with software implementations like the screen-reader, but for the user to understand the context and meaning of the form in the form of audio is something new that we have achieved.

Our software mainly focusses on easing the process of filling online forms for the differently abled, especially the

visually impaired by improving navigability & accessibility. This is achieved by providing a voice based medium for automated entry in the fields present in an online form. This automation is achieved by Natural Language Processing, Computer Vision, AI and other similar techniques. Our software eliminates the need of a helper to fill out the form for the differently abled, giving the differently abled a sense of independence. This application can be used in a variety of set ups like Banks, Govt offices, etc., to aid various registration processes.

II. PREVIOUS WORK

In [1], Verma et al. discuss some of the issues and challenges the visually impaired face while navigating the internet and provide a framework to solve them. The framework comprises of a dedicated speech-based interface that may be provided to an already existing public interest website by its owner, to provide the blind users its essential services. Therefore, a blind user can independently perform important tasks on such websites, without using any assistive tool.

They make use of a Speech Enabler (web server plugin) that records the form elements along with their traversal order as a macro. A logical chain is built among form elements, and both client and server know about this. When the macro is triggered upon user request, the response page(s) are created on the fly and a unique ID is attached to each necessary element in the order of their traversal. This is done to ensure successful facilitation of the task.

In [2], Ingle et al. discuss the implementation of a Voicemail system that can be used by a visually impaired individual to access their emails easily. The proposed system makes use of Interactive Voice Response, Speech-to-text converter, mouse click events and a screen reader.

The user is first required to register himself with the app, where information is prompted using voice prompts and answered by speech. After successful registration and login, the user is taken to the main application page where he can perform two actions, view emails in the inbox and compose an email. The application uses a combination of voice prompts and click events to guide the user's movements and help them achieve their desired result.

The proposed system, though useful for the visually impaired, requires the user to make use of a mouse to move through the application, and hence is not accessible for the other differently abled who may not be able to use a mouse.

In [3], Feiz et al. discuss how it was almost impossible till now for the blind to fill out forms independently, as they are unable pinpoint the form field locations and quite often, they are unable to understand what fields are a part of the form.

They mention WiYG, a Write-it-Yourself-Guide that directs a blind user to the various fields of the form, ensuring they can fill out forms independently without the need for assistance from anyone. The WiYG makes use of a pocket sized three dimensional printed smartphone attachment which comprises of two parts: base (acts like a phone stand which keeps the phone upright) & a reflector (attaches to the top of phone to re-focus the phone-camera to the paper placed ahead of it). The system also uses well-established computer vision algorithms like Optical Character Recognition using the cameras embedded in smartphones. This dynamically generates instructions in audio format, guiding the user to the fields of the forms.

In [4], Khilari et al. give an idea about the technological perspectives and appreciation of the progress of speech to text conversion. They give the complete speech to text conversion based on Raspberry-Pi. Multiple techniques used in each step of a speech recognition process are discussed. An attempt is made to analyze an approach to efficiently design a speech recognition system.

The system is implemented using Raspbian image installed on Raspberry pi. An audio file is recorded and after that a set of commands are applied. A .txt file stores the output of speech to text conversion. Automatic Speech Recognition (ASR) is a part of the implementation, which consists of a training & recognition phase. During the training phase, the system learns the reference patterns representing the different speech sounds like phrases and words. The recognizing phase is where an unknown input pattern is identified by considering the set of references.

In [5], Singh et al. explore the possibility of face detection in the first attempt by making use of the HAAR cascade classifier trained on images with simple and complex backgrounds. The paper also details the implementation of a modified algorithm to better detect frontal faces, and images having multiple objects to be detected.

Existing face detection makes use of two main methodologies: Detection based on skin color where the classifier divides each pixel into color or non-color based on the algorithm. The other method is the Viola-Jones face detection, where the face is detected by focusing on the detection of a few integral features, that help detection with higher accuracy.

The proposed method involves converting the image to the RGB scale and making use of the Viola-Jones method to identify integral features like the eyes and nose. They then find the coordinates of these features which are then used to plot the position of the face in the image. The facial features are extracted from the image and later used for subsequent detection in other images.

The results provided by the authors show how the proposed implementation helps detecting faces in very bright or very dark lighting conditions. It also aids in the detection of multiple faces under similar conditions.

III. THE PROPOSED SYSTEM

The solution proposed is to provide an easy to use voice - based medium for automated entry in the fields present in online forms. Current systems emphasize more on user friendliness of abled users. The software we have implemented is based on increasing user friendliness of all sections of people by placing emphasis on ease of use and complete independence for the end user.

Following are the sequential steps of our proposed system:

Step 1: The presence of the user is detected and the software is prompted to initiate the form filling process.

Step 2: The User is requested to display their Aadhar Card via a voice prompt.

Step 3: The 12 digit Aadhar Number is scanned using OCR and the person's face is recognized & matched with the data present in the Aadhar Database using Face Recognition. This is implemented to ensure complete security during the form filling process.

Step 4: Each form field is read out to the user in the language of their choice (English/Kannada). The user is then prompted to respond.

Step 5: The user is asked to confirm each response for added accuracy. After the response is validated, the next relevant form field is loaded.

This process is repeated till all responses are collected. Upon the completion of this process, the form is filled with the collected responses, thus finally achieving our goal of filling out the online form.

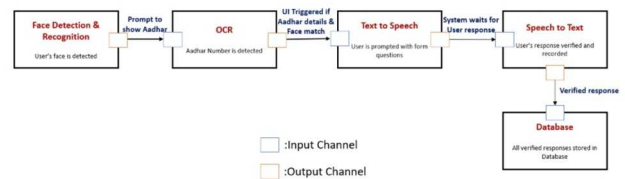


Fig 1. The Proposed System

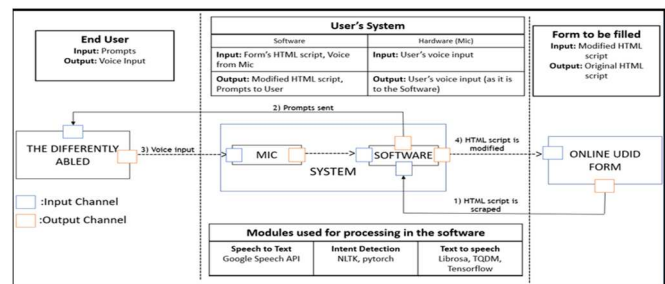


Fig 2. System Architecture

IV. EXPERIMENT AND DISCUSSIONS

Our software is split into several modules, which includes Face Detection, Face Recognition, Aadhar Detection, Speech-to-Text, Text-to-Speech, and the User Interface. Each module performs a specific activity to meet the stated requirements.

A) Database

“Notion” is used to create & manage both databases, The Aadhar database (Aadhar_db) and the Responses database (Form_Responses_db). The Aadhar database used contains sample Aadhar data, with fields like Aadhar number, name,

date of birth etc. to simulate user authentication with Aadhar data. It is stored remotely for easy access.

The Form responses database contains the collected form responses. The data is accessed by means of a POST request which contains: a secret key associated with the database, the table ID, and the names of the fields being retrieved in the form of a JSON object.

Aadhar_db

+ Add a view

Aadhar_nu...	Name	Date of ...	Gender	Address	Photo
2580 1111	Anita	August 8, 1973	Female	City	
1598	Rithik Rajendra Mali	December 9, 1999	Male	India	
2 4432	Sanjay JK	November 15, 1968	Male	Onyx	
5353	Shreya sri	December 22, 2000	Female	Apar t	

Fig 3. Aadhar Database

Form Responses

Form Responses DB

Aadhar_nu...	Name	Date of ...	Gender	Category	Address	Disabilit...
1234 5678 1901	John Doe	August 8, 1973	Female	General	Bangalore	Yes
1342 1432 2342	Tom Sue	June 23, 2000	Male	General	Bangalore	No
1342 1453 0721	Michael	May 2, 2000	Female	SC	Bangalore	Yes
4563 4563 4563	Emma	May 19, 1970	Male	General	Bangalore	Yes
3223 1643 1453	Glenn	November 15, 1968	Male	OBC	Bangalore	No
2345 6789 1534	Susan	December 9, 1999	Male	General	Bangalore	Yes
1342 5674 1980	Mary	December 22, 2000	Female	General	Bangalore	No

Fig 4. Form Responses Database

B) Face Detection

Face Detection is used to detect the presence of the user in front of the system. A HAAR feature based cascade classifier is trained on positive and negative images of different faces, and is used to detect the presence of a face in the video stream. The different features captured by the classifier include the contours of the eyes, the nose and the highlight on the cheeks. Each frame captured by the system webcam is analysed for the presence of the user. Upon successful face detection, an image of the face is captured along with a call to trigger Aadhar detection.

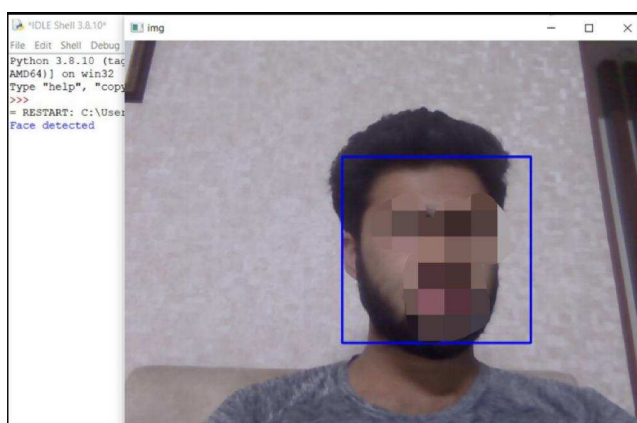


Fig 5. Face Detection

C) Aadhar Detection

The Aadhar detection module detects and returns the Aadhar number, which is then used to query and retrieve user information from the Aadhar database.

To ensure the detection is highly accurate, our software involves the use of the Google Cloud Vision API, which provides AutoML vision and other machine learning models for Optical Character Recognition. The Google Cloud Vision API offers powerful pretrained machine learning models through REST and RPC APIs.

The Aadhar database is queried with the detected Aadhar number to retrieve the users information and photograph. If retrieval is successful, Face Recognition is triggered. The application is able to detect the face through reasonable noise, though it is assumed that a proper camera setup is present which can capture video at a resolution of 720p and above.



Fig 6. Aadhar Detection

D) Face Recognition

The image captured during face detection and the user image retrieved during Aadhar detection are matched to ascertain the identity of the user. This provides an added level of security and ensures that the owner of the Aadhar card is the one filling the form. Before the software is run for the first time, the face recognition model is trained on all the faces present in the Aadhar database.

Each image is loaded, colour converted and encoded using Dlibs state-of-the-art face recognition model. These encodings are saved for later use. When face recognition is called, the image to be matched is encoded and compared with the encodings previously saved. If a match is found, corresponding Aadhar number of the user identified is returned. Successful recognition triggers the First page of the UI

E) Text to Speech

To provide a hands free experience to the end user, each prompt on the screen is read out using text to speech. Google's speech API is used for the text-to-speech conversion to ensure clear and accurate pronunciation of the text displayed on the screen. Each text to be read out is converted to an audio output and save as an mp3 file. This file is played at the appropriate intervals in the User Interface.

F) Speech to Text

To improve ease-of-use, the user responses for the form questions are captured using the microphone and converted to text using Google's speech to text API. The API makes use of complex deep-learning speech recognizer algorithms, which is out of the scope of this project. The user's response is recorded and stored as an mp3 file. This file is then read and converted to text by the API.

The API provides support for text conversion in several languages, which is useful to extend the application. The API's powerful algorithm can identify and eliminate some degree of background noise, which helps in the capture of response in slightly noisy conditions, but it is still expected that the background noise should be minimal during the form filling process.

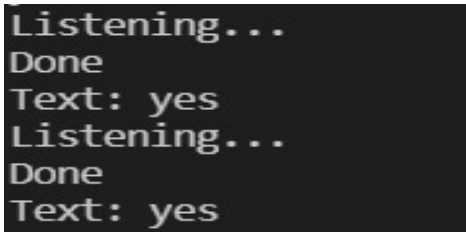


Fig 7. Speech to Text

G) User Interface

The User Interface contains the pages which are displayed to the end user. It is written in HTML & CSS with JavaScript to provide functionality & communication with the backend. Special python packages provide means to communicate between the frontend, backend & a locally hosted web server, on which the application is run.

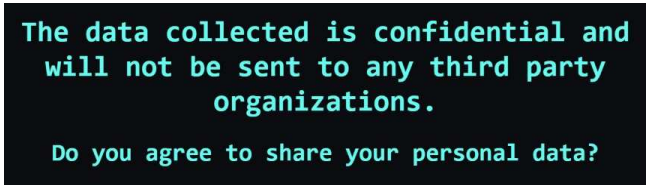


Fig 8. User Interface

H) Support for Regional Languages

To provide support for regional languages, the Google Cloud Translation API is used to translate the form questions and responses to Kannada. The User is provided with an option to use the software in either English or Kannada. The User's responses in Kannada are translated to English for verification and for saving the responses to the database.



Fig 9. Support for Kannada

The end user is allowed to respond to the form questions in the language of their choice, either manually or with voice. Aadhar detection and Face Recognition provide an added level of security and user authentication. The software is made to be adaptable and can be used for a variety of forms with minimal handling.

V. TESTING AND RESULTS

The results obtained after real-time testing of our implemented software at an academy for the blind are summarized below:

A) Face Detection

The User Interface is able to proceed to the next step of the form filling process when a human face is detected by the web cam. When no human face is detected by the web cam, the User Interface doesn't proceed to the next step and awaits the presence of a human face.

B) Text to Speech

The text provided is accurately read out and the system is ready to listen to user response for the next step. A beep is provided after the form question is read out, in order to provide a cue for the user to respond at the appropriate interval.

C) Optical Character Recognition

When a non-Aadhar document is presented, no Aadhar number is detected, thus the software restarts from Face Detection step. When a fake Aadhar is presented, Aadhar number is detected, however no Aadhar number is found in the Aadhar database. The software thus restarts from Face Detection step. When an authentic Aadhar is presented, Aadhar number is detected and found in the Aadhar database. The software proceeds to the welcome page.

D) Speech to Text

When no voice input is provided for a prompt, the Software repeatedly asks the User the same prompt, until a response is recorded. When voice input is provided before the beep, a broken response is recorded. Thus due to lack of clarity in response, the software repeatedly asks the user the same prompt, until a clear response is recorded. When voice input is provided after the beep, clear response is recorded. Due to clarity in response, the software asks the user for re confirmation, following which it transitions to the next prompt.

E) Face Recognition

When human face (not the owner of the Aadhar Card) is present in front of the web cam, the UI doesn't proceed to the next step of form filling, since the human face recognized doesn't match with the corresponding picture of the human in the Aadhar DB.



Fig 10. Faculty at the Academy for Blind using our software

When human face (owner of the Aadhar Card) is present in front of the web cam, the UI proceeds to the next step of form filling, since the human face recognized matches with the corresponding picture of the human in the Aadhar DB.

VI. CONCLUSION AND FUTURE WORK

The end goal of this project was to make it a usable product for our target audience, the differently abled. The current technologies being used and the gap between the User's requirements and the technological aids available, specifically in India, was also considered while designing and implementing our application.

Thus, we tested our application on our target audience at an Academy for the Blind. Our demo here strengthened the need to have such an application, as the faculty there mentioned how an application like this would very much be helpful. The issues we faced were:

A) The user did not know when to start speaking to initiate the automated response: This was solved by adding additional beeps.

B) The user did not know how to use overall application: This was solved by adding additional voice prompts guiding the user at every step of the process.

Since we visited both educational institutes and banks, we have acquired a firsthand idea of the lack of aid provided to the differently abled, in both domains. The future could be envisioned as kiosks to enable different processes: exam registrations, bank registration, tax filing, voting etc.

From our previous interactions it was made clear that often, the third party individuals while helping the visually impaired user might fill details incorrectly or may have malicious intents and manipulate the data.

The application removes the need for that third party, making the process more secure, convenient and giving a sense of independence to the user. The transition to make the everyday life of the visually impaired from dependent to independent, is something that needs to be worked on a large scale globally. We believe we are one step closer in contributing to that cause.

REFERENCES

- [1] Verma, Prabhat, Raghuraj Singh, and Avinash Kumar Singh. "A framework to integratespeech based interface for blind web users on the websites of public interest." *Human-Centric Computing and Information Sciences* 3.1 (2013): 1-18.
- [2] Ingle, Pranjali, Harshada Kanade, and Arti Lanke. "Voice based e-mail System for Blinds." *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)* 3.1 (2016): 25-30.
- [3] Shirin Feiz, Syed Masum Billah, Vikas Ashok, Roy Shilkrot, and IV Ramakrishnan. 2019. Towards Enabling Blind People to Independently Write on Printed Forms. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Paper 300, 1–12.
- [4] Khilari, Prachi, and V. P. Bhoje. "Implementation of speech to text conversion." *International Journal of Innovative Research in Science, Engineering and Technology* 4.7 (2015): 6441-6450.
- [5] Singh, A., H. Herunde, and F. Furtado. "Modified Haar-cascade model for face detection issues." *International journal of research in industrial engineering* 9.2 (2020): 143-171.