Kannada Speech Synthesis

GUI Application

By

Shashank Sharma

Developer

Preface

“Kannada Speech Synthesis, A Tutorial to use GUI Application” gives any user, with no knowledge of using the GUI Application for commercial/ non-commercial use. Kannada Speech synthesis is a project submitted by the students of Dept. of Electronics and Communication, RIT- Bangalore as a part of the final year project as specified by the department. The project is mentored by Mr. Sadashiva V Chakrasali, Assistant Professor, Dept. of Electronics and Communication Engineering, RIT. Kannada Speech synthesis is a 5 Month project which used Festival, Festvox and other voice tools to render Computer generated Kannada Voice, previously trained with a dataset of 696 Audio Files with Kannada Transcripts.

The project developed at the backend (Referred to as ‘core’ in this document) works as CLI (Command Line Interface). A series of commands with setting environmental variables and running a specific script synthesizes Kannada Text. CLI is a perfect tool for the purpose of debugging, but fails to meet the user’s requirements of working with a responsive clicks and views.

This application is developed by Shashank Sharma, one of the developer of the Core Project on QT4.8 Framework using PyQT. Starting from Scratch, every feature in this application is developed in hand. With an in-app media player, a database connectivity, theme changer and a lot other features, this application stands different in terms of its speed, responsiveness, and user satisfaction.

The application was first developed in English. Release version 2.0 rendered the application in 2 languages, English and Kannada both of which are selectable within the application. At the time of writing this report, the application is at v2.13 hosted on github and made completely open source licensed under GNU’s GPL v3.0

Application Installation and Run Instructions

The application is open source and can be cloned from Github. The application is currently tested on Ubuntu 18.04 and is expected to work well in any Linux Environments. The application is not targeted for Windows users but we intend to migrate the application and test it on Windows environment in the future releases.

The backend project is not made public for reasons of it being the project submission, but the same can be requested and will be provided to the users on certain conditions set by the developer. Once the requested file is acquired the instructions to install Backend in your Linux Machine is put in the same github repository under Core Folder.

The backend is lightweight compressed file about 50 MB and is independent of the GUI Application. The backend can be run to generate kannada voice without the GUI application. The lightweight application does not support text analysis.as unwanted features of the backend are removed in making the project lightweight.

The frontend is written in Python with a bit of Shell files that talk with the backend project in synthesizing kannada voice automatically on button click. Frontend can be downloaded by running the following command assuming that git package is installed on your Ubuntu Machine. Clone the package alongside `Main` folder (Core) which is located in the `Project` Folder

```
git clone https://github.com/shashankrnr32/KannadaTTS-Application
```

Once the application is downloaded change the directory to `KannadaTTS-Application` which contains a script called `run.sh`. The application can be run by providing executable permissions to run.sh and then executing the script.

To provide executable permissions to script, open the terminal and type

```
chmod 755 run.sh
```

To run the script,

```
./run.sh
```

Application Folder and File Structure

```
├──── App
│      ├──── Essentials.py
│      ├──── FestAPI.sh
│      ├──── Main.py
│      ├──── res
│      │      ├──── DB
│      │      └──── *.desktop
│      └──── ui
│             ├──── img
│             │      └──── <Image Files>
│             └──── *.ui , *.qrc
├──── Core
│      ├──── README.md
│      └──── scripts
│             └──── *.sh
├──── LICENSE
├──── run.sh
├──── Screenshots
│      └──── <Application Screenshots>
└──── WavFiles
       ├──── DSP
       │      └──── *.wav <Synthesized Files Processing = False>
       └──── NoDSP
              └──── *.wav <Synthesized Files Processing = True>
```

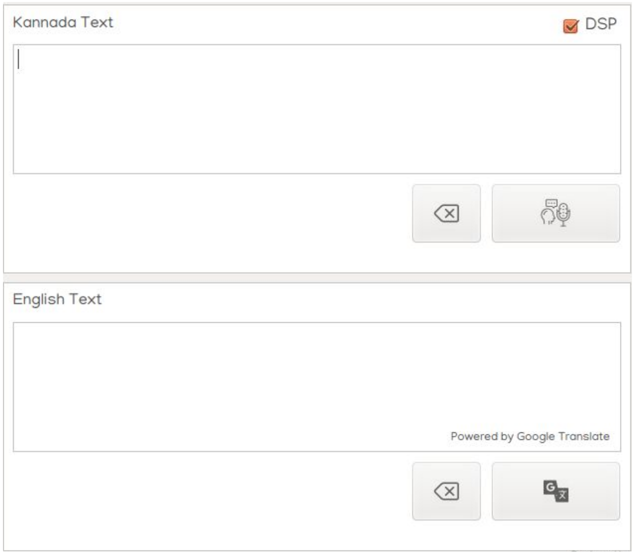
File and Folder Description

1. **App/Essentials.py** : Contains Python classes and functions for Translation, Database, Mail
2. **App/FestAPI.sh** : Shell file to connect to the core project. Also does post-synthesis operation and stores the file in WavFiles directory
3. **App/Main.py** : Application renderer Python File. Contains all the classes that handles front end views such as Main View, About Page, Table View, Plot View.
4. **App/res/DB** : SQLite Database File that contains 2 table . Kannada Synthesis and Translation Table
5. **App/res/*.desktop** : Desktop File to run the application with double click. Copy this file to desktop and double click to run the application.
6. **App/ui/*.ui , *.qrc** : User Interface Files designed in Qt-Designer. Qt Modified XML files.
7. **Core/Scripts/*.sh** : Important scripts to run the core project
8. **LICENSE** : License File. Open source licensed under GNU’s GPL v3.0
9. **run.sh** : Script to run the frontend. This script will install the required packages and will run the application
10. **WavFiles/** : contains .wav files of synthesized audio files.

Application User Interface

The application consists of a number of widgets that perform various operations that together make up the application.Some of them are, the built in Media Player, Table of details, Database View, Plot dialog etc. All of the interface components are explained in detail below.

1) **Synthesis and Translation Form** : Consists of Synthesis and Translation forms with text inputs which is accepted from the user.



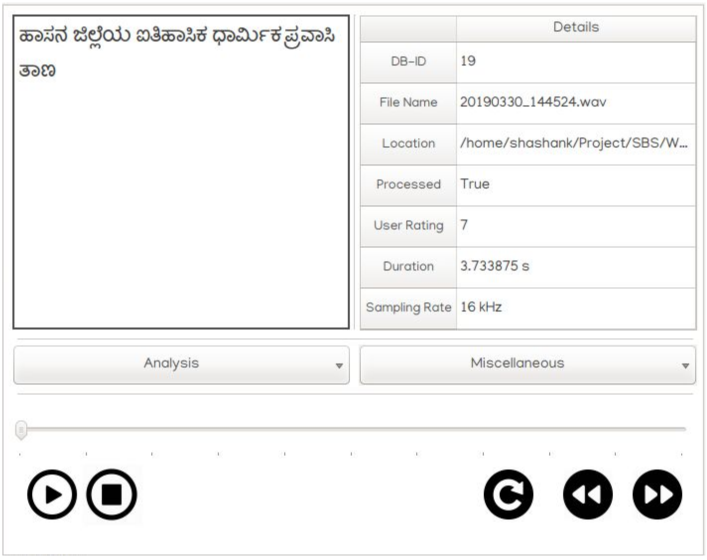
   a) **Synthesis Form** is connected to the core project through the file named FestAPI.sh. The mentioned script performs all the necessary synthesis operations and also performs post-synthesis signal processing operations based on the choice given by the user by checking/not checking the DSP checkbox. The Synthesized .wav files are saved in `WavFiles` Folder based on the choice of DSP.

      The synthesis form also validates the language of input. User cannot input English text in the Kannada Text-Input. The text that is entered by the user is scanned continuously by the code and the decision is taken to retain the text or to remove it from the input when it doesn't match with the Kannada Unicode Range.

   b) **Translation Form** is connected to Google Translate Engine API which translates English Text entered in the form and the translated text is added to the Kannada Text Input which can be synthesized. The translate API requires a private key file provided by Google Cloud.

2) **Media Player and Table of details** : Media player consists of Play/Stop, Refresh, Next/Previous that extract data from the database and display the corresponding audio file and its transcript displayed on a text browser.

   PyQT Phonon class is used to integrate audio into the application. Phonon class provides dedicated methods of play, pause, stop along with various other configurable options. A seekbar is also integrated that displays the percentage of audio played. The concept of threading is used to update the seekbar and play audio at the same time. Though there is some error in seekbar's configuration it is not perceivable by the user.

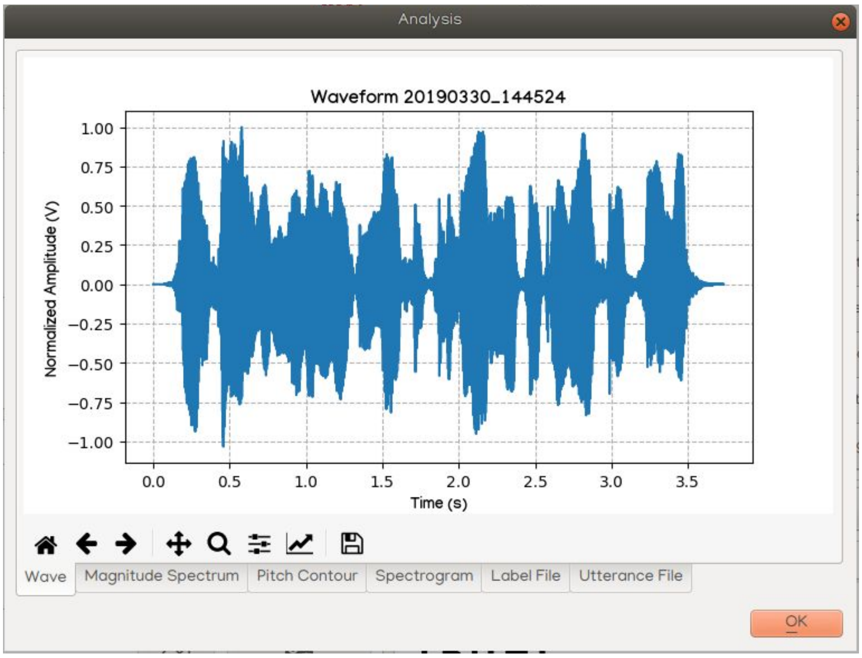


ಹಾಸನ ಜಿಲ್ಲೆಯ ಐತಿಹಾಸಿಕ ಧಾರ್ಮಿಕ ಪ್ರವಾಸಿ ತಾಣ

	Details
DB-ID	19
File Name	20190330_144524.wav
Location	/home/shashank/Project/SBS/W...
Processed	True
User Rating	7
Duration	3.733875 s
Sampling Rate	16 kHz

Table of details displays the details of the current audio active in the media player such as File Name, Location, Duration, Sampling rate and User Rating. Table of details is updated every time the active data is changed.

3) **Analysis Dialog** : The application also consists of audio and text analysis options able to display plots of different kind. These plots are rendered using Matplotlib-QT class, which is an extension of Matplotlib providing developers to integrate Graphs and plots in QT Application. The plots are integrated on a separate Dialog Box.

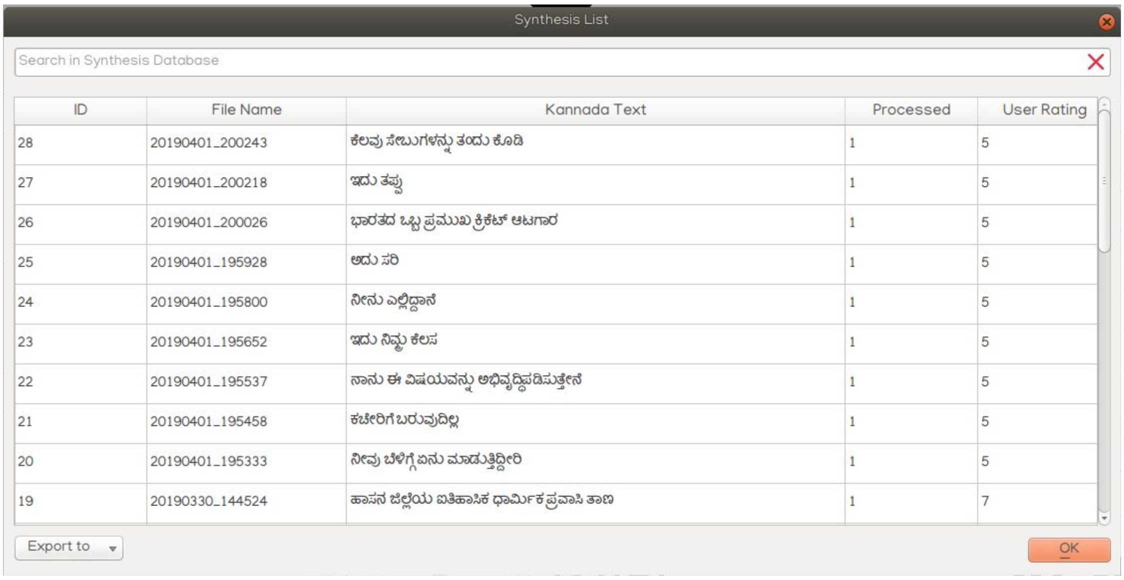


Along with the audio analysis, there is also provision of text analysis which uses Festvox to generate Label files and Utterance files. All the analysis options are displayed in the form of tabs, so that the user can change analysis type within a single window. The text analysis is a highly processor dependent task, hence the application caches the generated text files in an `ignore` folder which can be deleted at any moment.

The following plots are currently available under audio analysis.

i) Waveform
ii) Magnitude Spectrum
iii) Spectrogram
iv) Pitch Contour

4) **Database View** : All the synthesized waveforms, their details are stored in a SQLite Database with the file named `DB`. The entries of the database can be viewed with an additional database viewer but the same is designed within the application. The menu bar provides 2 options to view Database table and Translation table which extracts data from the database and displays it in the form of table.



ID	File Name	Kannada Text	Processed	User Rating
28	20190401_200243	ಕೆಲವು ಸೇಬುಗಳನ್ನು ತಂದು ಕೊಡಿ	1	5
27	20190401_200218	ಇದು ತಪ್ಪು	1	5
26	20190401_200026	ಭಾರತದ ಒಬ್ಬ ಪ್ರಮುಖ ಕ್ರಿಕೆಟ್ ಆಟಗಾರ	1	5
25	20190401_195928	ಅದು ಸರಿ	1	5
24	20190401_195800	ನೀನು ಎಲ್ಲಿದ್ದಾನೆ	1	5
23	20190401_195652	ಇದು ನಿಮ್ಮ ಕೆಲಸ	1	5
22	20190401_195537	ನಾನು ಈ ವಿಷಯವನ್ನು ಅಭಿವೃದ್ಧಿಪಡಿಸುತ್ತೇನೆ	1	5
21	20190401_195458	ಕಚೇರಿಗೆ ಬರುವುದಿಲ್ಲ	1	5
20	20190401_195333	ನೀವು ಬೆಳಿಗ್ಗೆ ಏನು ಮಾಡುತ್ತಿದ್ದೀರಿ	1	5
19	20190330_144524	ಹಾಸನ ಜಿಲ್ಲೆಯ ಐತಿಹಾಸಿಕ ಧಾರ್ಮಿಕ ಪ್ರವಾಸಿ ತಾಣ	1	7

The database view is also equipped with a search and autocomplete feature similar to search engines.

5) **About Window** : Displays the details of project, developer, mentor and license using Tabwidget similar to the plot dialog box. This is a static UI.

6) **Miscellaneous Features** : The application also consists of some additional features as mentioned below.

a) **Mail** : The active audio can be mailed automatically to any requested user. The mail id of the requesting user is entered in the input dialog. The mail id is validated and a custom mail is sent to the requestor.

b) **Theme Changer** : The application runs GTK theme by default which can be changed upon user choice. There are currently 6 themes that the application runs on.

c) **Full Screen** : The app runs only in full screen mode and is not changeable by the user. This was brought into production in v1.0 Beta.

Application Running algorithm

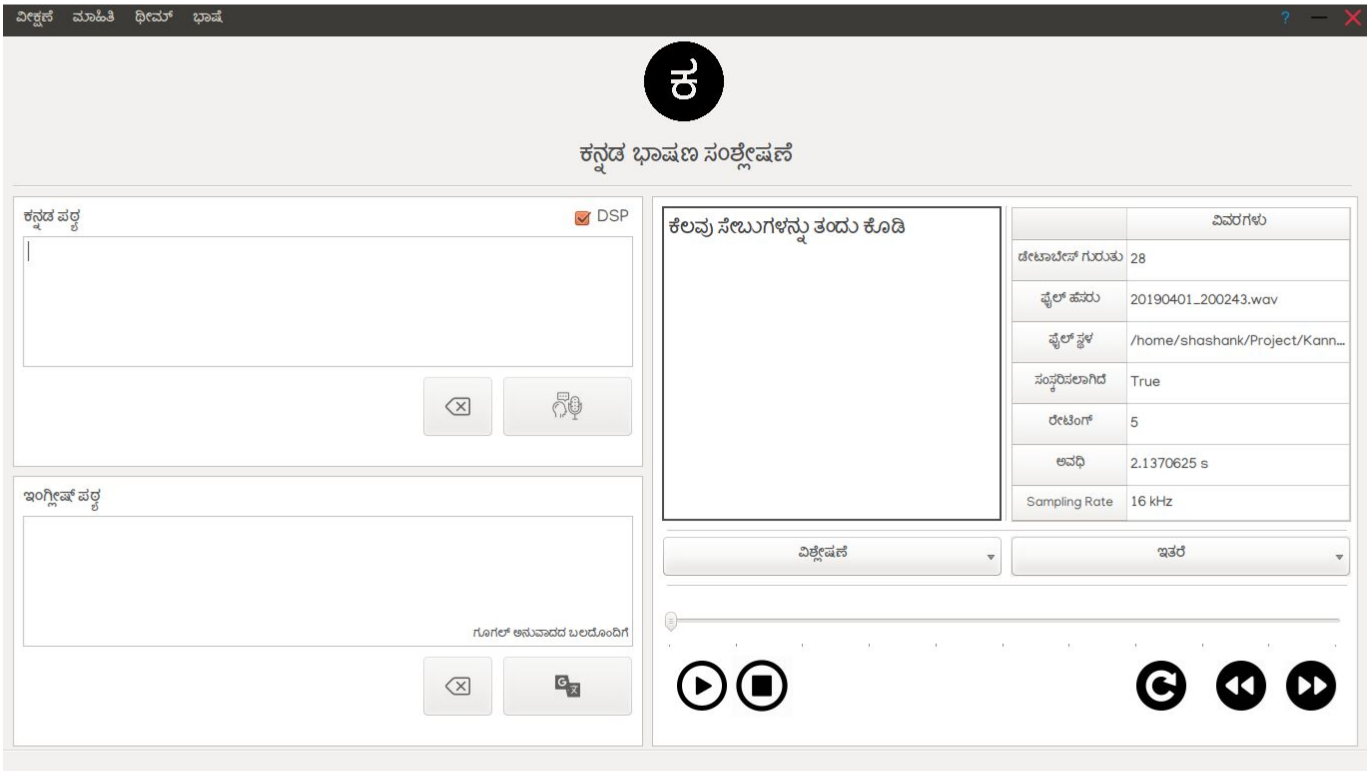
The script `run.sh` is the entry point of the application. The script follows certain algorithmic steps in rendering UI and starting GUI

1. Start running script run.sh
2. Locking Mechanism:
    a. Check for Lock File
    b. If Exists, Exit the program (Another instance of the application is running)
    c. If not, Create a lock file with the current PID (Creating new Instance)
3. Check the system if PyQt, Qt, Phonon, pysptk and other packages are installed.
4. If the packages are not installed, install them.
5. Convert `.ui` files to `.py` files by `pyuic4` (UI to py converter)
6. Convert `.qrc` files to `.py` files by `pyrcc4` (Qt Resources to py)
7. Run `Main.py`
    a. Carry out Required Operations in the Event Loop
8. On Exit/Language Change
    a. With code 0 :
        i. Remove `.py` files that were earlier converted from `.ui` file
        ii. Remove Lock File
    b. With code 10 or 20 : Language Change option
        i. Remove Lock File
        ii. Re-run the script with `-kan` argument (No argument if English)

Language Choice

The application was primarily developed in English with English titles, tooltips, status messages. The application was then ported to Kannada in the github release version 2.0. All the user interface designed was same as that of the English counterpart, but the Language was changed to Kannada to provide a Kannada support to the users. All the features that work in English version also works in Kannada Version.

The Kannada ui can be rendered by passing `-kan` argument to run.sh or the user can manually change it in the application which closes the current application and renders a new window with changed language. The user can revert back to English the same way. Dynamic Language translation is not enabled as of this project.



ಕೆಲವು ಸೇಬುಗಳನ್ನು ತಂದು ಕೊಡಿ

	ವಿವರಗಳು
ಡೇಟಾಬೇಸ್ ಗುರುತು	28
ಫೈಲ್ ಹೆಸರು	20190401_200243.wav
ಫೈಲ್ ಸ್ಥಳ	/home/shashank/Project/Kann...
ಸಂಸ್ಕರಿಸಲಾಗಿದೆ	True
ರೇಟಿಂಗ್	5
ಅವಧಿ	2.1370625 s
Sampling Rate	16 kHz