Embedded Deep Learning to Support Hearing Loss Mobility: In-House Speaking Assistant

Nikola Stanojkovski[0000−0001−9555−963X], Tashko Pavlov[0000−0001−7689−4475], Mario Stojchevski[0000−0002−0662−6421], Kostadin Mishev[0000−0003−3982−3330], and Monika Simjanoska[0000−0002−5028−3841]

Ss. Cyril and Methodius University
Faculty of Computer Science and Engineering,
Rugjer Boshkovikj, 16, 1000 Skopje, N. Macedonia
{nikola.stanojkovski, tashko.pavlov,
mario.stojchevski}@students.finki.ukim.mk,
{kostadin.mishev,monika.simjanoska}@finki.ukim.mk
http://www.finki.ukim.mk

Abstract. Hearing-impaired people encounter significant mobility issues in their everyday life as the desire to communicate what they think and need in an increasingly crowded urban society grows, all for social, physiological, safety, and self-esteem reasons. Because of the rapid advancement of technology in the twenty-first century, there are several tools available to aid these individuals with their difficulties and to enable them to clarify their thoughts through a number of devices. All of these technologies, however, are prohibitively expensive and difficult to obtain for these individuals. Many of these assistants require an Internet connection to function properly since they involve extensive data processing on certain Cloud services, which can be a limiting factor because Internet connections are not available everywhere, and not everyone has easy access to them. In this study, we offer a methodology for providing a hearing-impaired person with a speaking assistant that is entirely integrated within the device and does not require an Internet connection, making it a highly economical and portable solution for anyone. Our solution embeds one intelligent Deep Learning model, a text-to-speech model, on a single smart mobile device. This model generates an audio file from the text entered by the user and plays it back to the device's available output speaker. Our work brings us one step closer to fully self-contained embedded intelligent models that use cutting-edge AI to assist hearing-impaired people with communication.

Keywords: Intelligent embedded systems · Deep learning · Disabilities · Assistive technology · Speech technologies · Mobile technologies.

1 Introduction

According to the World Health Organization (WHO), 1.5 billion people are estimated to be hearing impaired in 2022[1], accounting for approximately 20% of

[1] https://www.who.int/health-topics/hearing-loss

the global population. One billion of these people are young and between the ages of 12 and 35 years old. By 2050, the predicted number of people with some degree of hearing loss would be approximately 2.5 billion[2], with 700 million of them requiring rehabilitation. This indicates that one out of every ten people will suffer from hearing loss that is incapacitating. All of these statistics combined paint a clear picture of how serious the problem of hearing loss is. As a result, the demand for low-cost assistive technology has risen to the top of the priority list.

Hearing impaired people still face considerable challenges in their daily lives due to a lack of accessible and suitable adaptive equipment. In the absence of assistive aids, a never-ending cycle of ineffective foundational education is perpetuated, followed by substantial impairment in social relationships, resulting in dissatisfaction, low confidence, and limited autonomy. Bullying has harmed almost one-third of deaf youth's family ties, living circumstances, communication, judgment and physical health [5]. For guiding hearing-impaired people, there are a variety of methods and special tools, each with its own set of benefits and drawbacks:

- Cochlear implant [16]: A medical procedure that uses electric stimulation of the remaining auditory nerve to restore some hearing to a completely deafened person. It's utilized to break through the "inhuman stillness that divides and estranges," as Helen Keller, a prominent blind-deaf writer, lecturer, and social crusader, described it. The history of the cochlear implant is long, illustrious, and fascinating, with dynamic interactions between engineers and clinicians, as well as a delicate balance between experimentation and ethics [15]. As indicated by the exponential rise in both the patient population and scholarly publishing, cochlear-implant research has matured as a field. Many deaf people can now hear effectively thanks to cochlear implants, but it is clear that full hearing recovery is not possible and that this technology can only be utilized in a limited number of circumstances and situations with a certain degree of precision [14].
- Alerting devices [1]: Devices that help hearing impaired people be aware of sounds, such as the doorbell or a ringing phone. Additionally, they can inform them of local events like fires, intruders, or their child's behavior. These gadgets emit a signal that can be identified. A horn, a flashing light, or a vibration can all serve as signals. The specific solutions for this cause include tactile devices, strobe lights, low frequency alarms, and audible smoke detectors. The demands of people with hearing loss at any point can be met by all of these technologies, but because of the limited ways in which information can be delivered to them, they cannot possibly meet all of their needs.
- Telecommunication relay services [13]: Text and video-based services that have gained popularity in recent years. Typically, video-based services are referred to as either video remote interpretation (VRI) or video relay services (VRS). In the USA, VRS and VRI are thought to serve different purposes,

[2] https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

although in Europe, VI services are discussed more broadly. The first text relay service (TRS) used experienced operators to read and write captions in order to support live telephone conversations. Deaf VI callers worry about the interpreter handling video calls accurately and effectively, much like with text-relay services. The accuracy of the message being interpreted is the main worry. 40% still have concerns about the lack of speed, and 40% are also concerned about the privacy of their conversation.

2 Related Work

The paper 'Deep learning based assistive technology on audio visual speech recognition for hearing impaired' [4] uses deep learning based assistive technology to conduct an audio-visual speech recognition solution for the mute and hearing impaired people. It is a system that utilizes artificial intelligence, signal processing and natural language processing (NLP). The idea is to use two modules, one for audio speech recognition and another one for visual recognition of lip movement. Both outputs from the modules are fed into a third module leading to a multi-modal fusion process that analyzes this data and outputs a generated text. The software solution is an example of practical usage of the hidden Markov Model [7] and Convolutional Neural Networks (CNN).

'The BridgeApp' [11] is a software solution (application) for a mobile device that is completely used offline. The application is made up of four modules: text-to-speech, speech-to-text, finger spelling and scenarios. The first two are simply converting text to audio, and, audio to text respectively. The finger spelling module converts text to sign language. The scenarios module consists of 4 situational modules (work, school, store, church) that help the user communicate in those particular situations or surroundings. Although it claims it is fully offline, there is not much information about the back-end processing of the gathered data and the database itself.

'An Assistive Technology for Hearing-Impaired Persons: Analysis, Requirements and Architecture' [6] presents ways of analyzing people's needs that have hearing impairment and a solution with a clear advantage over the mentioned papers they have researched. The presented solution is a software embedded assistive technology that can be used indoors and outdoors. In the paper they have illustrated a diagram where they explain the expected use case. Typically, a user would have the device (smartphone) with them, which will be capable of recording any audio in their surroundings using built in sensors. Thereafter, pattern recognition algorithms are used to extract useful features from the sound waves that are fed into a neural network. The neural network is capable of discovering the context behind this sound and outputting this to a tactile interface. They have not discussed the type of interface that they want to use to communicate this back to the user, but it is said that any kind of vibrating watch, belt or bracelet is sufficient.

Many papers such as 'Hearing technologies' [2] by Blazer and the co-authors research about the various solutions to this problem. This paper focuses on re-

searching the past solutions that were offered to people with hearing impairments and the improvements of the very same solutions. Such are traditional hearing aids that were mainly treated as medical devices that later transitioned to consumer products. These hearing aids most commonly include a microphone, some kind of an analog-to-digital converter to process signals, battery for the device and an output device. In the paper the important thing mentioned is that hearing aids cannot restore hearing, nor correct it to an extend as glasses can correct vision for example.

The paper 'Personal sound amplification products vs conventional hearing aid' [8] explains that purchasing hearing aids in the United States can only be done with a prescription from a licensed professional with a mean cost of $4700 for 2 hearing devices. PSAPs (Personal Sound Amplification Products) [3] on the other hand are thought as over the counter and less expensive devices and are not specifically labeled as hearing loss treatment. These devices are accessible more to the public and are technologically comparable to the hearing aids that the medical industry presents to the people. The paper continues to test and provide results for the PSAP. The results point out to over 70% improvement in speech understanding while the candidates were using the PSAP.

'Mobile applications to detect hearing impairment: opportunities and challenges' [12] is a paper that describes how doctors or patients can detect hearing loss early in life. It should not be neglected that many people suffer from hearing loss, which is frequently untreated. The paper's main point is that hearing problems can be detected and treated early on. This is accomplished through the use of various software applications, mostly on iOS, that a regular person can use to determine whether he or she has a problem.

3 Methodology

This section presents the key components of our methodology described in the following subsections. The suggested methodology was created exclusively for Android mobile devices and can be applicable to a wide range of use cases and applications.

3.1 Text-to-speech Model

We provide an implementation of the FastSpeech 2 [9] model for Text-to-Speech (TTS) inference in English language on mobile device. In this research we present the possibilities of FastSpeech 2 model on a limited processing power, specifically on Android mobile device. Even though this paper presents an implementation only on a FastSpeech 2 TTS model it can be further expanded using different models of the same category. The novelty presented in this research paper is the ability of executing such complex model on a mobile device which is rather limited in processing power.

[3] https://en.wikipedia.org/wiki/Personal_sound_amplification_product

FastSpeech 2 was selected as the TTS model because it is a non-auto regressive model that greatly speeds up and produces comparable quality speech synthesis. The FastSpeech 2 model outperforms the FastSpeech [10] model in terms of the quality of the generated speech and quicker training and inference times. This non-auto regressive model expects only text as input. The model is composed of a number of layers, including: Phoneme Embedding, Encoder, Variance Adaptor, Mel-spectrogram Decoder and Waveform Decoder. The encoder block transforms the phoneme embedding sequence into a phoneme hidden sequence which is then the input to the variance adaptor. The variance adaptor has a key function in this architecture since it applies different voice specific information such as duration, pitch and energy to the hidden sequence. Finally the mel-spectrogram decoder model converts the transformed hidden sequence into mel-spectrogram sequence in parallel.

3.2 Model Training

For the purpose of our implementation we used the pre-trained version of the FastSpeech 2 model. The model has been trained on the LJSpeech dataset [3], which is a public domain speech dataset made up of 13,100 short audio samples of a single speaker reading from seven different books. A transcription is provided for each audio clip. Each audio clip length is varying from 1 to 10 seconds and has a combined length of 24 hours. Table 1 presents the statistics of the dataset used for training, evaluating and testing the FastSpeech 2 model.

Table 1: LJSpeech Dataset Statistics

Metric	**Value**
Total Clips	13,100
Total Words	225,715
Total Characters	1,308,678
Total Duration	23:55:17
Mean Clip Duration	6.57 sec
Min Clip Duration	1.11 sec
Max Clip Duration	10.10 sec
Mean Words per Clip	17.23
Distinct Words	13,821

Table 2 shows the dataset splits and the corresponding number of samples in each set.

4 The Architecture

All demographics, including those who have hearing loss, find mobile devices to be the easiest tools to use. Android currently has the highest popularity among

Table 2: FastSpeech 2 Dataset Splits

Dataset	**No. of samples**
Training	12,228
Validation	349
Testing	523

all platforms[4]. As a result, the decision was made to build a solution that offers an application for people with hearing loss to comfortably handle the gadget.

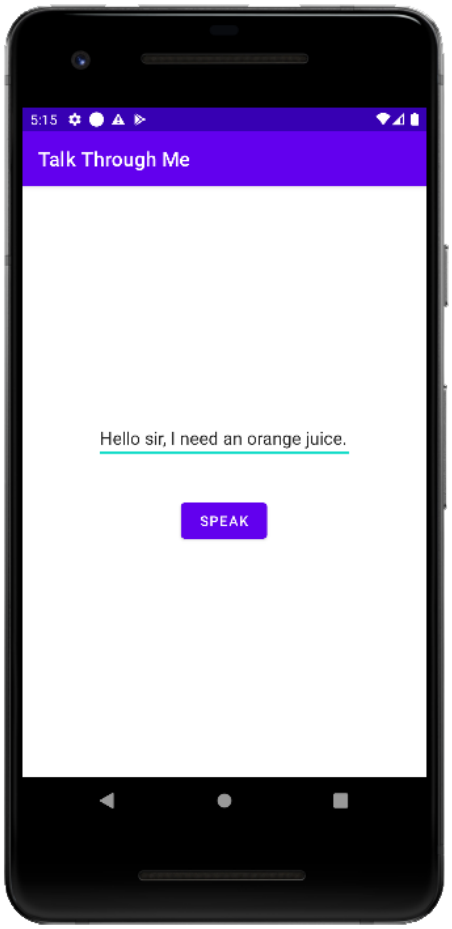

Fig. 1: Application's Main Activity

[4] https://www.businessofapps.com/data/android-statistics/

4.1 Android components

As for the application's primary architecture for the design and implementation, the solution makes use of a few Android components. Other crucial aspects of Android, such as the built-in APIs, were also utilized.

In an Android-based application, an activity is a single, narrowly defined task that the user can complete. There is only one Android activity in the application: **MainActivity**. It's the primary screen of the application, which appears immediately after it is launched. It has a simple appearance with an edit-text widget for entering and modifying text and a button for submitting the data and starting model inference with the presently input text. The design of this activity is shown in Figure 1.

Services are components that carry out time-consuming activities in the background. There is one service that was created to simplify communication with Python models and scripts: **TTSService**. It's the service that launches the Text-to-Speech model, it's Vocoder, and calls the model's inference function when necessary. If the inference is successful, the audio is played through the device's built-in output speaker.

Coroutines are concurrency design patterns that make asynchronous code easier to understand and run. They were useful in our application because of few tasks, such as model inference and loading, as well as information exchange between components, which required multi-threaded parallel processing in the background in separate threads, in order to keep user's attention while interacting with the interface.

Notifications are messages that Android displays outside the application's user interface to provide the user with reminders. We employed this feature to provide the user with real-time updates on the Text-to-Speech model's progress. While the application is running, the notification bar is not dismissive and remains active at all times.

Broadcast Receivers (Receivers) [5] are components that allow system and/or application events to be registered. These components served as a communication and data transmission method between different types of components in our system. It was primarily used to exchange information across the **TTSService** and the **MainActivity**.

4.2 Modules integration

Chaquopy[6] is the name of the Python library that was used to run and execute Python programs in the Android application. It is a very potent tool because of its user-friendly features and the possibilities it opens up for any form of Python-related task, particularly the implementation of Machine Learning models on a mobile device. The location, name, and exact method that must be called, together with the required parameters of any appropriate type, are all

[5] https://developer.android.com/guide/components/broadcasts
[6] https://chaquo.com/chaquopy/

specified by the Android application when calling scripts. The Text-to-Speech model FastSpeech 2, which was developed using Python scripts and APIs that are compatible with it, was loaded and inferred using this tool.

4.3 Application flow

Application's design objective was to enable a person with hearing loss to utilize it to fulfill his tasks with the least amount of interaction and in the quickest possible time, taking into account his need for verbal expression. The straightforward design was chosen in order to prevent any distractions, such as irrelevant graphic animations and objects, from interfering with the user's speaking needs at any given time while he is using the tool.

A single edit-text field is used as a tool for the user to enter the text that he wants the device to speak for him, and a button is located below it for submitting the data and informing the user whether it is available by toggling the availability with changing the statuses of it from disabled to enabled, and vice versa. As was already indicated, the design consists of one simple activity that loads with these few straightforward components when the program launches.

After the user submits the text, it is transmitted to the Text-to-Speech model: FastSpeech 2, for inference, synthesis and audio production with the necessary information. The ".wav" file that is being placed in the device's temporary storage is immediately played out on the phone's speaker after this task is completed. The user is prompted to provide these rights if the program does not already have them for storing data on the phone's device.

Only the initial launch of the program on the user's mobile device requires an Internet connection. This is because all required dependencies must be downloaded for the Text-to-Speech model to perform. The remaining processing is done internally on the device with no additional use of any form of Cloud services.

Figure 2 illustrates the user flow as well as the interaction between the components of the program and how they operate together.

The source code is available on GitHub[7].

5 Experiments and Results

Since the novelty of this research is the capacity to run complicated, performance-intensive, and massive Deep learning models on a low resource mobile device, we evaluated the performance of the Text-to-speech model on several mobile devices and compared the outcomes. To compare these results, we run 5 comparable tests on 4 various tasks related to our application. Three distinct devices underwent the tests.

If the installation time is brief, the user experience will be considerably enhanced and there will be less waiting before use. This is so that the user doesn't

[7] https://github.com/nikolaStanojkovski/Talk_Through_Me

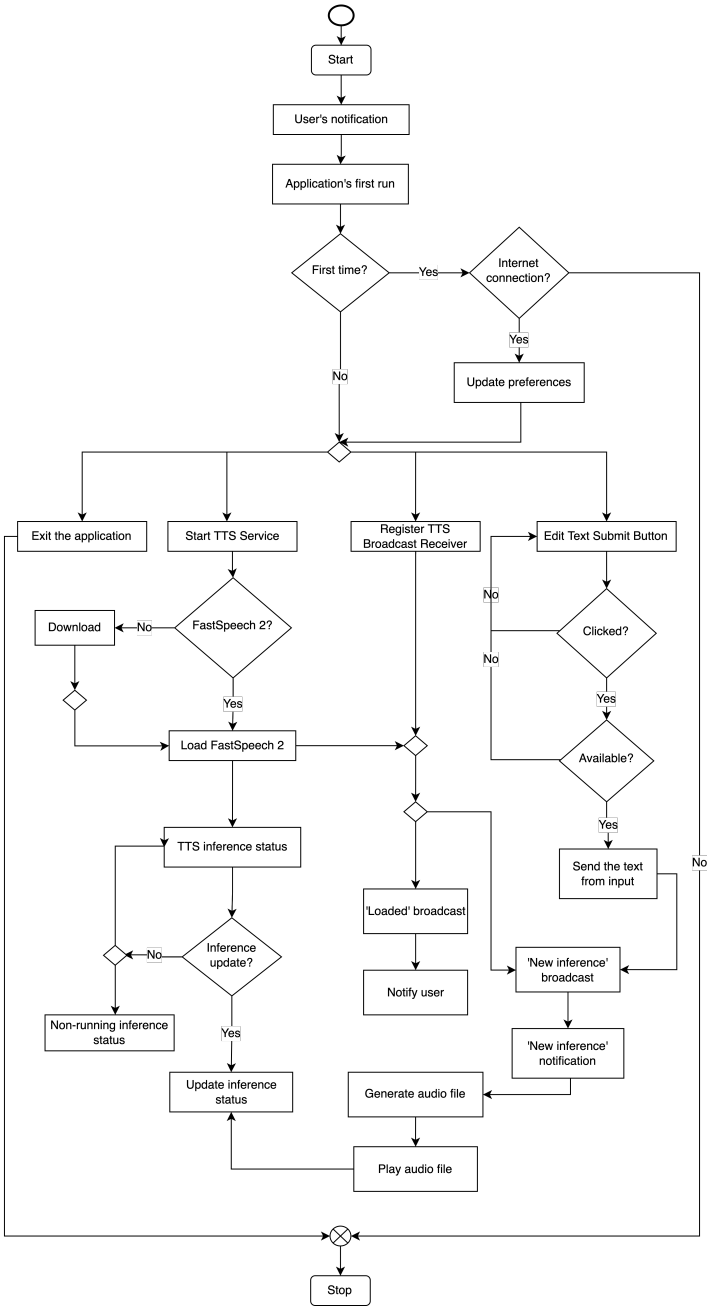

Fig. 2: Application's flow diagram

get the impression that the program is consuming excessive amounts of storage space or processing unnecessary device data. How efficiently the application generates audio files, or simply how long the user must wait for the device to speak the text he has requested, depends on the FastSpeech 2 Text-To-Speech engine, which is utilized as the main text processing engine. The loading time varies on the first launch because the model is loaded from its source. The amount of time the user must wait for the application to load before using it is determined by the Text-To-Speech engine's loading time, which is why it is very crucial. These are the reasons for the chosen metrics in the experiments.

The results from the experiments are shown in Table 3, Table 4 and Table 5.

Table 3: App performance on 'Pixel 2' Emulator

Metric	**1st try**	**2nd try**	**3rd try**	**4th try**	**5th try**	**Average**
App installation time	2m20s	2m05s	1m45s	1m50s	2m30s	**2m6s**
TTS model loading time (first time)	23.3s	22s	25.3s	21.2s	21.9s	**22.74s**
TTS model loading time (every other time)	7.5s	9.2s	9.5s	8.1s	7s	**8.26s**
TTS model inference time	6s	5.2s	4.8s	5.5s	5.1s	**5.32s**
Final app size	0.85GB					

As stated at the beginning, we are aware of how crucial fast inference time is for this particular problem. The performance of the suggested solution when used on an emulator device is displayed in Table 3. To assess the effectiveness of all the above parameters, we conducted 5 experiments. The results demonstrate that such execution time may be reached without using any external services or running such complex models on the Cloud. The generation of an audio file for it to be played on the output speaker requires quick inference time on the Text-to-speech model. We crammed the entire system onto a portable device with fast execution speeds. The average inference time for the entire system is ∼6s-7s. The Text-to-Speech model typically takes ∼5s-6s of inference time in average from the moment the text is input in order to produce an audio file.

The Xiaomi Poco X3 Pro smartphone has cutting-edge mid-to-high-end hardware specifications. Table 4 summarizes the results, as well as information about the metrics used during the tests. It is obvious that on a more powerful device, the results are much better, and the application is utilized more quickly and effectively.

Table 5 presents the execution times of the 4 tests for the application. The mobile device that these tests were executed on is Samsung Galaxy A41, which has low to average performance according to its specifications.

The devices ordered by performance (from highest to lowest) on which the application was tested is:

Table 4: App performance on 'Xiaomi Poco X3 Pro'

Metric	**1st try**	**2nd try**	**3rd try**	**4th try**	**5th try**	**Average**
App installation time	22s	20s	21.2s	18s	20.8s	**20.4s**
TTS model loading time (first time)	18s	18.8s	18.5s	19s	16.9s	**18.24s**
TTS model loading time (every other time)	7.3s	7.7s	6.4s	6.9s	7.1s	**7.08s**
TTS model inference time	10s	10.5s	9.5s	9.8s	7.5s	**9.46s**

Table 5: App performance on 'Samsung Galaxy A41'

Metric	**1st try**	**2nd try**	**3rd try**	**4th try**	**5th try**	**Average**
App installation time	48s	50s	58s	38s	46s	**48s**
TTS model loading time (first time)	1m21s	1m07s	1m21s	1m23s	1m08s	**1m16s**
TTS model loading time (every other time)	51s	28s	27s	28s	27s	**32.2s**
TTS model inference time	23s	24s	25s	24s	23s	**23.8s**

– 'Xiaomi Poco X3 Pro'[8]
– 'Pixel 2'[9]
– 'Samsung Galaxy A41'[10]

The outcomes of the experiments demonstrate that the inference time and model loading time are influenced heavily by the device's processing speed. The state of the battery, how many processes are running simultaneously on the device, how much RAM is set aside, and other software settings are just a few of the many variables affecting all of these trial outcomes. Performance of the devices varies and is greatly influenced by their characteristics. The user experience is improved by the device's greater specifications. Given all of this, it is recommended that a user owns a more contemporary mobile device with the minimum hardware specifications listed below in order to obtain the best user experience:

– CPU: Quad-core (x2.0 GHz)
– RAM Memory: 4GB
– Storage: 40GB
– Loud-speaker: Any
– Operating System: Android 10

[8] https://www.po.co/global/product/poco-x3-pro
[9] https://www.android.com/intl/en$_u k/phones/google - pixel - 2/$
[10] https://www.samsung.com/ie/smartphones/galaxy-a/galaxy-a41-black-64gb-sm-a415fzkdeua/

6 Conclusion and Future Work

This paper is focused on attaining a solution within a mobile device for the hearing impaired people. The goal is that a user would type a text input in his/her mobile device. The device is to read out the text to the person that hearing-impaired is trying to communicate with. The mobile device is to serve as interface for communication between the two individuals and help the impaired to feel more secure and confident in their day to day life. The solution provided in this research encourages that deep learning models are not to be executed on the Cloud and constant internet connection through which the data is streamed, but to implement such models on a poor processing unit device which does not require internet connection, yet it does the processing in-house and can be used in various places. Until now no such solution was provided where the model was implemented in-house on a daily used widget such as mobile device. In order to enable users with reasonably priced devices to set up, install, and make inferences more quickly and effectively, our future work will concentrate on speeding the model's text input processing and loading.

References

1. Ashley, E.M.: Waking effectiveness of emergency alerting devices for the hearing able, hard of hearing, and deaf populations. University of Maryland, College Park (2007)
2. Blazer, D.G., Domnitz, S., Liverman, C.T., for Adults, A.H.H.C., National Academies of Sciences, E., Medicine, et al.: Hearing technologies: Expanding options. In: Hearing Health Care for Adults: Priorities for Improving Access and Affordability. National Academies Press (US) (2016)
3. Ito, K., Johnson, L.: The lj speech dataset. https://keithito.com/LJ-Speech-Dataset/ (2017)
4. Kumar, L.A., Renuka, D.K., Rose, S.L., Shunmuga priya, M.C., Wartana, I.M.: Deep learning based assistive technology on audio visual speech recognition for hearing impaired. International Journal of Cognitive Computing in Engineering **3**, 24–30 (2022). https://doi.org/https://doi.org/10.1016/j.ijcce.2022.01.003, https://www.sciencedirect.com/science/article/pii/S2666307422000031
5. Landsberger, S.A., Diaz, D.R., Spring, N.Z., Sheward, J., Sculley, C.: Psychiatric diagnoses and psychosocial needs of outpatient deaf children and adolescents. Child Psychiatry & Human Development **45**(1), 42–51 (2014)
6. Mielke, M., Grünewald, A., Brück, R.: An assistive technology for hearing-impaired persons: Analysis, requirements and architecture. In: 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). pp. 4702–4705 (2013). https://doi.org/10.1109/EMBC.2013.6610597
7. Rabiner, L.R.: Mathematical foundations of hidden markov models. In: Niemann, H., Lang, M., Sagerer, G. (eds.) Recent Advances in Speech Understanding and Dialog Systems. pp. 183–205. Springer Berlin Heidelberg, Berlin, Heidelberg (1988)
8. Reed, N.S., Betz, J., Kendig, N., Korczak, M., Lin, F.R.: Personal sound amplification products vs a conventional hearing aid for speech understanding in noise. Jama **318**(1), 89–90 (2017)

9. Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.: Fastspeech 2: Fast and high-quality end-to-end text to speech (2020). https://doi.org/10.48550/ARXIV.2006.04558, https://arxiv.org/abs/2006.04558
10. Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.: Fastspeech: Fast, robust and controllable text to speech (2019). https://doi.org/10.48550/ARXIV.1905.09263, https://arxiv.org/abs/1905.09263
11. Samonte, M.J.C., Gazmin, R.A., Soriano, J.D.S., Valencia, M.N.O.: Bridgeapp: An assistive mobile communication application for the deaf and mute. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC). pp. 1310–1315 (2019). https://doi.org/10.1109/ICTC46691.2019.8939866
12. Swanepoel, D.W., De Sousa, K.C., Smits, C., Moore, D.R.: Mobile applications to detect hearing impairment: opportunities and challenges. Bulletin of the World Health Organization **97**(10), 717 (2019)
13. Turner, G.H., Napier, J., Skinner, R., Wheatley, M.: Telecommunication relay services as a tool for deaf political participation and citizenship. Information, Communication & Society **20**(10), 1521–1538 (2017)
14. Tyler, R.S.: Advantages of disadvantages expected and reported by cochlear implant patients. The American journal of otology **15**(4), 523–531 (1994)
15. Zeng, F.G.: Trends in cochlear implants. Trends in amplification **8**(1), 1–34 (2004)
16. Zeng, F.G., Rebscher, S., Harrison, W., Sun, X., Feng, H.: Cochlear implants: system design, integration, and evaluation. IEEE reviews in biomedical engineering **1**, 115–142 (2008)