# Voice Morphing Application for trainers of autistic spectrum disorder

J G Sukumar[1], Mohith Sai Ram Reddy[2], Nikhileswar Sambangi[3]
Sai Krishna Kumar[4], Remya M S[5], Prema Nedungadi[6]

[1]amenu4cse20332@am.students.amrita.edu, [2]am.en.u4cse20341@am.students.amrita.edu,
[3]amenu4cse20360@am.students.amrita.edu, [4]amenu4cse20318@am.students.amrita.edu,
[5]remyams@am.amrita.edu, [6]prema@am.amrita.edu

Department of Computer Science and Engineering, Amrita School of Computing,
Amrita Vishwa Vidyapeetham, Amritapuri, India

## Abstract

**Autism Spectrum Disorder (ASD) encompasses a range of conditions characterized by challenges with social skills, communication, and repetitive behaviors. Individuals with ASD often face difficulties in auditory processing and social interaction. Recognizing these challenges, a Voice Morphing Application has been developed specifically to aid trainers working with individuals with ASD. By leveraging deep learning technology, the application allows trainers to adjust their vocal tones, addressing the unique communication needs of ASD individuals.**

***Keywords: Autism Spectrum Disorder (ASD), Voice Morphing, Deep Learning, Auditory Processing, Social Interaction***

## 1. Introduction

Children with autism spectrum disorder (ASD) [1] often encounter significant challenges in social interactions, primarily due to difficulties in auditory processing. Traditional communication methods have struggled to effectively address the unique needs of individuals with ASD. While existing solutions exist, they may not fully meet the requirements of both trainers and students. Hence, the development of innovative tools is crucial.

This paper presents a Voice Morphing Application designed to alleviate the communication barriers faced by individuals with ASD [2]. By leveraging deep learning technology, the application empowers trainers to modify their voices, facilitating better engagement and understanding among ASD [1] students. Key challenges such as auditory information processing difficulties are addressed through this transformative tool, aiming to improve learning outcomes and overall well-being.

To address these challenges, a user-friendly interface is proposed, allowing children to communicate accurately by simply clicking on images. Additionally, trainers are provided with an interface featuring basic commands via images, facilitating seamless communication. While current approaches like assistive communication devices and speech therapy have shown effectiveness, the Voice Morphing Application, with its deep learning capabilities, offers a cutting-edge and personalized solution.

To facilitate real-world deployment and accessibility, our model was built in Python and incorporated into Flask [3][4], a popular web framework. To ensure seamless integration and optimal performance, our voice cloning, and text-to-speech models were integrated into a user-friendly web interface using the

Flask framework [3][4]. Flask, known for its simplicity and flexibility, serves as the backbone of our application, bridging the gap between our AI models and the end-users.

## 2. Related Works

### 2.1. Empowering Autistic Students in Computing

This study proffers a practical flexible framework for teachers and researchers embodying diverse computing pedagogies, to impart computing education (CE) to autistic students. The framework is based on the tenets of inclusion and personalised learning manipulating explicit CE pedagogies. Research and anecdotal evidence exhort that autistic learners tend to favour coursework and careers in the STEM domain. Specifically, autistic [1] young people excel in the fields of computing and programming. Computing is very logical and precise and seems to suit the strengths that categorize autism. Yet, the needs of autistic students are still not being met in the classroom as the education sector continues to explore inclusive models for teaching and learning. An in-depth analysis of pedagogical frameworks employed in CE for autistic students in this study underscores the imperative of practical activities in preference to theoretical discussions. Female teachers reported lower-self-assessed proficiency in semantic waves and physical computing, suggesting a need for targeted training. In comparison to their mainstream counterparts, special school teachers indicated deficit knowledge in multiple pedagogies, including unplugged computing, Parson's problems, pair programming, peer instruction, PRIMM, and projects, compared to their mainstream school counterparts, highlighting a resource and support gap.

### 2.2. Definition of Voice Cloning

Voice cloning [5] encompasses systems that generate speech imitating a specific speaker, distinguishing itself from generic speech synthesis by emphasizing the target speaker's identity. It includes technologies like text-to-speech (TTS), voice conversion (VC), and other speech synthesis systems. Evaluating a voice cloning system involves assessing its naturalness, similarity to the target speaker, memory usage, computing speed, and data efficiency. Data efficiency is crucial for scalability, enabling the system to perform well with limited data while also leveraging abundant speech data when avail- able, regardless of transcriptions. The NAUTILUS system, while adaptable to various input interfaces, primarily focuses on TTS and VC tasks due to their indispensable role in the voice cloning framework.

### 2.3. Innovative Speech Recognition

To make Sphinx-II usable in a PC environment, we need to tackle is- sues of recognition accuracy, computational efficiency, and usability simultaneously. A large amount of RAM and high-end workstation are unrealistic for today's popular PC environments where low-cost implementations are critical The system must also be speaker adaptive, because there will always be some speakers for which the recognition error rate will be much higher than average due to variation in dialect, accent, cultural background, or simply vocal tract shape. The ability of the system to reject noise is also crucial to the success of commercial speech applications. Noises include not only environmental noises such as phone rings, key clicks, air conditioning noise, etc. but also vocal noises such as coughs; ungrammatical utterances, and Out. We have significantly improved Whisper's [11] accuracy, efficiency and usability over the past two years. On a 260-word Windows continuous command-and- control task and with 800KB working memory configuration (all the RAM

required, including code and data), the average speaker-independent word recognition error rate was 1.4.

### 2.4. Vall-E: Revolutionizing Text-to-Speech Synthesis

The introduction of Vall-E [6], a neural codec language model for text-to-speech synthesis, marks a departure from previous continuous signal regression methods. Vall-E utilizes discrete codes from a pre-trained neural audio codec model, enabling training on a vast dataset of 60K hours of English speech. This extensive training data vastly surpasses existing systems and enhances Vall-E's ability for in-context learning. Notably, Vall-E [7] demonstrates superior performance over state-of-the-art zero-shot TTS systems, exhibiting enhanced speech naturalness, speaker similarity, and preservation of speaker emotion and acoustic environment. Previous TTS systems relying on cascaded models suffer from limitations in data quality and generalization, especially for unseen speakers. Existing approaches like speaker adaptation and encoding methods require additional fine-tuning and complex engineering, whereas Vall-E offers a streamlined solution with promising results.

### 2.5. Whispered Speech: Analysis and Recognition

This study explores whispered speech's acoustic properties and recognition challenges, particularly in noisy mobile phone environments. Acoustic analysis reveals formant frequency shifts and spectral energy differences between whispered [10] and normal speech. Adaptation techniques with limited whispered speech data enhance recognition accuracy. However, whispered speech still struggles in noise, prompting SNR improvement methods like mouth covering. The paper examines whispered speech recognition in a statistical ASR framework, stressing the need for extensive training data. It presents findings on acoustic characteristics and recognition performance under varied conditions. The study underscores whispered speech's importance for private mobile phone communication, where loud speaking is impractical. It discusses production mechanism differences from normal speech, attributing them to the lack of vocal cord vibration. Moreover, it addresses challenges posed by background noise, complicating whispered speech processing and recognition.

## 3. Methodology

#### 3.1. Problem Foundation

The problem foundation lies in the communication challenges faced by individuals with autism spectrum disorder (ASD) [1] due to auditory processing difficulties. Traditional methods often fail to meet their unique needs, leading to frustration and limited social interaction. Trainers struggle to engage ASD [2] individuals effectively, hindering learning and social skills development.

#### 3.2. Preliminaries

The model which is shown in [Figure 1], proposes to use multiple existing modules for various components of the workflow.

VALL-E, or Voice Augmentation with Learned Language Embeddings, is a cutting-edge model in the realm of voice morphing. It stands out for its advanced architecture and capabilities, particularly in manipulating and synthesizing speech signals with remarkable precision. VALL-E focuses on enhancing voice quality through the utilization of learned language embeddings.

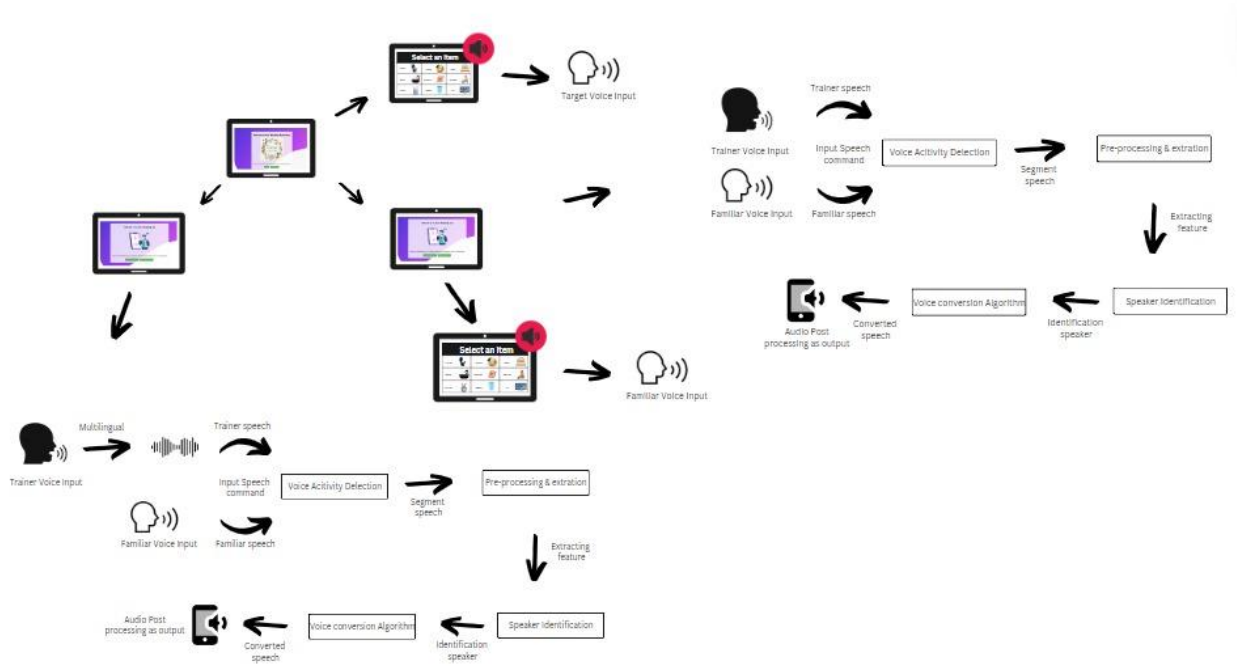


*Figure 1 - Methodology*

The VALL-E [6][7] architecture is composed of three main components: the Text Embedding and Audio Segment Quantization component, the Audio Reconstruction component, and the Quantized Token Prediction component. The processing of VALL-E [6][7] model is based on text to speech method. The audio prompt is processed by the audio codec encoder and then converted to audio quantization modules. The target text goes through phoneme conversion. This process breaks down the text into individual words, syllables, and phonemes. These every audio quantization symbol were used as a separate token. These tokens are inputs for the Text Embedding layer, and the output is used as inputs for the Quantized Token Prediction component's autoregressive (AR) [8] and non-auto-regressive (NAR) [9] blocks.

The Quantized Token Prediction section is in charge of estimating the number of quantized tokens required to generate a suitable speech that corresponds with the given text. The prediction component is divided into two blocks that use a transformer decoder Architecture: the auto-regressive (AR) [8] block and the non-autoregressive (NAR) [9] block. To calculate attention scores, the AR block only uses the quantized tokens from the prior round. This technique is

effective in predicting speech lengths for various speakers that match the given text since the initial stage (1st layer) of the quantized tokens includes auditory data like speaker information. On other side, the NAR [9] block considers the entire collection of quantized tokens

The fine-grained auditory data present in the later stages (2 to 8 layers) of the quantized tokens has a significant impact on the quality of the synthesized speech. The NAR [9] block enabling the style tokens to learn prominent features from the acoustic details, the tokens to record a number of characteristics, including background noise, speaking tempo, pitch, intonation, volume, and emotional intensity. By changing the token values, acoustic features can be controlled and scaled. In the Audio Reconstruction step, a pretrained DeCodec block is used to convert the quantized tokens that were predicted in the

Quantized Token Prediction portion back into audio waveforms. By predicting the temporal duration of the quantized tokens during this process, the AR [8] model determines the overall size of the token sequence. Next, by predicting the values of the remaining tokens, the NAR [9] model generates personalised speech tokens. The pretrained DeCodec block is used to transform the quantized tokens back into reconstructed audio waveforms.

The model incorporates a multilingual approach [10], allowing for the synthesis of speech across various languages. By leveraging adaptable components and linguistic processing techniques, system can effectively process text in different languages. Through phoneme conversion and flexible token prediction mechanisms, model accommodates linguistic nuances present in multilingual [10] contexts. The system considers diverse linguistic features, including phonetic variations, intonation patterns, and emotional nuances, to generate accurate and natural-sounding speech in multiple languages. This multilingual [10] capability enhances the versatility and accessibility of text-to-speech synthesis system, catering to a global audience with diverse linguistic backgrounds and preferences.

Whisper, on the other hand, is a neural vocoder developed by Mozilla. Like VALL-E, Whisper has gained recognition for its sophisticated architecture and advanced capabilities in voice transformation. It excels in generating natural-sounding speech, making it a valuable tool in applications requiring high-quality voice synthesis.

Whisper [11][12] stands as a breakthrough in the field of voice recognition and speech-to-text technology. With its sophisticated architecture and advanced capabilities, voice-based applications have been revolutionized, altering the way interactions occur.

At its core, Whisper [11][12] employs an encoder-decoder Transformer architecture, which is a state-of-the-art deep learning model known for its efficiency in handling sequential data. This architecture is pivotal in realizing the end-to-end Whisper system's capabilities.

The strength of Whisper [11] lies in its multitasking abilities. Not only can it convert speech to text with remarkable accuracy, but it also supports a range of other tasks. One of its standout features is speech translation, where Whisper can translate spoken words from one language to another in real-time. This feature is especially useful in multilingual settings, bridging communication gaps across different languages.

Language identification is another remarkable capability of Whisper[12]. It can recognize and identify the language being spoken, even in a multilingual environment. This feature is crucial for applications where language plays a vital role in determining the context or meaning of the speech.

Moreover, Whisper's [11] multilingual voice recognition [10] capabilities are truly impressive. It can understand and transcribe speech from various languages, making it a versatile tool for global applications. This is made possible through its extensive training on a diverse dataset comprising various languages and dialects.

The segmentation of audio into 30-second segments is a strategic approach adopted by Whisper. By breaking down the audio into manageable chunks, it ensures efficient processing and allows the system to maintain high performance even with long audio files.

The encoder component of Whisper plays a crucial role in this process. It transforms the audio segments into a log-Mel spectrum, which is a representation that captures the frequency and intensity of the sound. This transformed data serves as the input to the decoder.

The decoder, on the other hand, is trained to predict the corresponding text caption for the given audio segment. This is achieved using distinct tokens and advanced machine learning techniques. The training

process equips the decoder to understand and interpret the nuances of human speech, allowing Whisper [11] to generate accurate and contextually relevant transcriptions.

In addition to transcription, Whisper [12] also provides phrase-level timestamps, offering a detailed temporal understanding of the spoken content. This feature is valuable for applications where timing and sequence are critical, such as in legal or medical transcriptions.

Furthermore, Whisper's [11][12] ability to translate non-English speech into English is a game-changer. It opens up new possibilities for cross-lingual communication and expands the reach of voice-based applications to a global audience.

### 3.3. Algorithm

As shown in [Figure 2], we start by loading the model and deciding whether to use the computer's CPU or GPU for processing. We then check if the input and target audio files are valid and compatible. If they are, we make sure the input audio is 30 seconds long by either adding silence or cutting it short. Next, we create a picture of the audio's frequencies and intensities called a log-Mel spectrogram. This helps us understand the sound's features.

**Algorithm 1** Voice Cloning Algorithm Using Text

**function** CLONEVOICE(*input*, *target*)
  **if** CLONEVOICE(inputText, targetAudio) **then**
    **procedure** CLONEVOICE(*inputText*, *targetAudio*)
      *model* ← LoadVoiceCloningModel()
      *device* ← CPU or GPU
      **if** *inputText* and *targetAudio* are valid **then**
        *inputText* ← TranslateToEnglish(*inputText*)
        *prompt* ← MakePrompt(*targetAudio*, *inputText*)
        $tokens_{audio}, tokens_{text}$ ← Tokenize(*prompt*)
        *outputAudio* ← Generate(*model*, $tokens_{audio}, tokens_{text}$)
        *outputWav* ← SaveWAV(*outputAudio*)
        **return** *outputWav*
      **else**
        **return** error
      **end if**
    **end procedure**
  **end if**
**end function**

*Figure 2- Algorithm 1*

We then figure out the language of the input audio, which is important for accurate transcription. Using the model, we decode the audio and extract the recognized text. With the language identified, we combine elements from the target audio and recognized text to create a prompt. This helps us capture the essence of the voice we want to clone. We break down the prompts into smaller units called tokens, making them easier for the model to work with. Using the model, we generate new audio that sounds like the target voice. Finally, we save this audio as a WAV file, marking the completion of the voice cloning process.

**Algorithm 2** Voice Cloning Algorithm Using Text (Continued)

**if** CLONEVOICE(inputText, targetAudio) **then**
  **procedure** CLONEVOICE(*inputText*, *targetAudio*)
    *model* ← LoadVoiceCloningModel()
    *device* ← CPU or GPU
    **if** *inputText* and *targetAudio* are valid **then**
      *inputText* ← TranslateToEnglish(*inputText*)
      *prompt* ← MakePrompt(*targetAudio*, *inputText*)
      $tokens_{audio}, tokens_{text}$ ← Tokenize(*prompt*)
      *outputAudio* ← Generate(*model*, $tokens_{audio}, tokens_{text}$)
      *outputWav* ← SaveWAV(*outputAudio*)
      **return** *outputWav*
    **else**
      **return** error
    **end if**
  **end procedure**
**end if**

*Figure 3 – Algorithm 2*

As shown in [Figure 3], text-based voice cloning approach which allows users to input text directly, bypassing the need for audio files. This feature offers greater flexibility and convenience, enabling users to create voice clones simply by typing out the desired text. Additionally, the system automatically translates the input text into English, ensuring consistency in language processing. This capability broadens the accessibility of the voice cloning application, as users can easily input text in their preferred language without the need for pre-existing audio recordings. Overall, this unique feature streamlines the voice cloning process and enhances user experience.

As shown in below [Figure 4], the algorithm then validates the input data, ensuring that both the input image and the target audio file are valid and exist. Once the input data is validated, the algorithm extracts text content from the input image. This text could be anything present in the image, such as speech or dialogue. Subsequently, the algorithm com- bines this extracted text with the target audio file to create a prompt. This prompt serves as input for the voice cloning model. The prompt is tokenized, meaning it’s

broken down into smaller units of text or audio data that themodel can process effectively. Using the tokenized prompt, the algorithm feeds it into the voice cloning model. The model then generates synthetic audio

**Algorithm 3** Voice Cloning Algorithm Using Image for Trainers

```
if CLONEVOICE(inputImage, targetAudio) then
    procedure CLONEVOICE(inputImage, targetAudio)
        model ← LoadVoiceCloningModel()
        device ← CPU or GPU
        if inputImage and targetAudio are valid then
            text ← TextFromImageCommand(inputImage)
            prompt ← MakePrompt(targetAudio, text)
            tokens_audio, tokens_text ← Tokenize(prompt)
            outputAudio ← Generate(model, tokens_audio, tokens_text)
            outputWav ← SaveWAV(outputAudio)
            return outputWav
        else
            return error
        end if
    end procedure
end if
```

*Figure 4 – Algorithm 3*

that mimics the voice present in the target audio file. The generated audio is saved as a WAV file, making it accessible and usable for various applications as showin in Algorithm[3]. Finally, the algorithm returns the path to the generated WAV file as the result of the voice cloning process. If any issues arise during the process, such as invalid inputs or errors, the algorithm returns an error message to notify the user.

As shown in [Figure 5], it begins by checking if a child clicks on an image, which serves as the trigger for initiating the voice generation pro- cess. If the condition is met, the algorithm proceeds with the voice gener- ation procedure. Within the voice generation procedure, a function named GenerateRandomVoice is called, which takes an action parameter. This function is responsible for generating a random voice based on the provided action. Firstly, an array named voices is initialized with the action param-eter as its sole element. This array represents the available voices associated with the provided action. Next, a random index is generated within the range of the voices array length using the GenerateRandomIndex function. Th

is index is used to select a random voice from the array. Once a random index is obtained, the SelectVoice function is invoked with the voices array and the random index as arguments. This function selects the voice correspond- ing to the random index from the voicesarray.

**Algorithm 4** Voice Generation Algorithm Using JavaScript

```
if ChildClicksImage() then
    procedure GENERATERANDOMVOICE(action)
        voices ← [action]
        randomIndex ← GenerateRandomIndex(voices)
        selectedVoice ← SelectVoice(voices, randomIndex)
        DisplayVoice(selectedVoice)
        synth ← WindowSpeechSynthesis()
        utterance ← SpeechSynthesisUtterance(selectedVoice)
        synth.speak(utterance)
    end procedure
end if
```

*Figure 5 – Algorithm 4*

The selected voice is then displayed, presumably to provide visual feedback to the user regarding the chosen voice. Subsequently, the algorithm utilizes the Web Speech API by initializing a speech synthesis object (synth) using WindowSpeechSynthe- sis()— function. This object enables text-to-speech conversion within the browser environment. An utterance object is created using the SpeechSyn- thesisUtterance— constructor, with the selected voice as its parameter. This utterance object encapsulates the text content to be spoken in the selected voice. Finally, the synth.speak(utterance) statement invokes the speech synthesis engine to speak the content represented by the utterance object, effectively generating speech output using the selected voice.

## 4. Experimental Results

### 4.1. System Specifications

Software and Hardware: The voice conversion experiments utilized the Whisper and Vall-E models. The hardware setup included a microphone, speakers, a GTX 1650Ti GPU, an Intel i5 CPU, 8GB or more RAM, and an HTTP port for communication. Software components encompassed the necessary drivers for hardware and the implementation of Whisper and Vall-E models.

### 4.2. Spectral Analysis Chart

As shown in [Figure 6], The spectral analysis shows the frequency content of the original voice versus the

converted voice. Strong similarities in the locations of the formants and harmonics between the two spectra are evident. This indicates that key vocal tract characteristics like resonance frequencies are being preserved during the voice conversion process. The model also appears to be accurately capturing larger pitch and timbre (tone color/quality) of the voice. This is seen by the similarities in fundamental frequency and harmonic structure between the original and converted spectra.

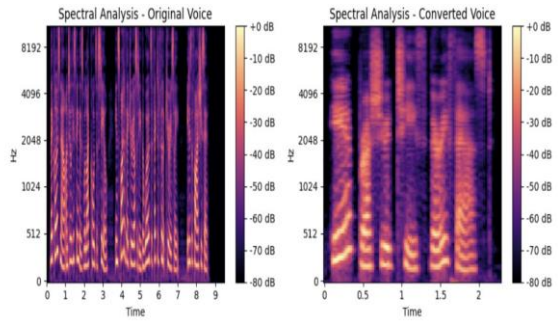


*Figure 6 – Spectral Analysis Chart*

However, there are some minor deviations visible in the higher frequency regions of the converted voice spectrum. This suggests there may still be room for improvement in how well the model captures more subtle voice attributes and details in that higher frequency range. In summary, the spectral analysis demonstrates that the voice conversion model is largely successful at maintaining the core vocal characteristics between the original and target voices. But further refinement could help it replicate even the finer nuances of the voice in the full frequency spectrum

### 4.3. Subjective Listening Test Results

The subjective listening test incorporated ratings across three metrics from 7 respondents. Key observations: • 80 percentage rated naturalness as good, indicating successful voice conversion • Similarity ratings were slightly lower, guiding refinements in target reproduction • High intelligibility suggests conversions preserved linguistic content

## 5. Output Images from the proposed Software

### 5.1. Home Page

As shown in [Figure 7] , The interface [15] features two distinct buttons: "ASD Children" and "Trainer."


Clicking on the "Trainer" button directs users to a dedicated page equipped with various tools tailored for effective communication and interaction facilitation.

*Figure 7 – Home Page*

On this page, trainers can access a range of resources and features designed to support their role in engaging individuals with Autism Spectrum Disorder (ASD). Conversely, selecting the "ASD Children" [1] button redirects users to an intuitive image-based interface. Here, ASD children can communicate and express themselves more effectively by selecting relevant images, simplifying the process and promoting meaningful interactions.

### 5.2. Trainer Main Page

As shown in [Figure 8], The trainer main page provides a brief overview of the voice cloning tool and its benefits for ASD communication training. It contains: An explanation of how the tool converts trainer voices to be more familiar for children. Testimonials highlighting success stories. Brief

instructions for getting started. Links to the voice cloning and text cloning pages.

The Voice Cloning Page [13] serves as a dedicated platform for trainers to clone their voices using audio samples. Designed with user-friendliness in mind,

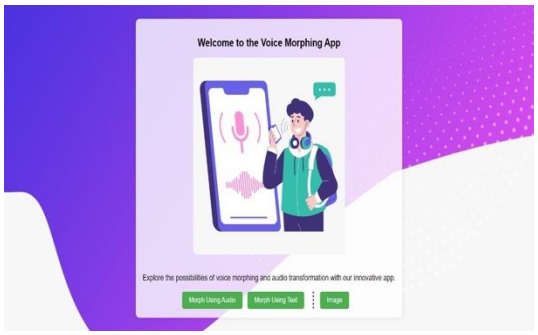


*Figure 8 – Trainer Main Page*

this page offers a seamless and intuitive interface [14] that simplifies the voice cloning process. Here's a breakdown of the key features and functionalities available on this page:

Morph Using Audio Shortcut:

From the trainer's homepage, a convenient shortcut labeled "Morph Using Audio" allows trainers to directly access the Voice Cloning Page. Clicking this shortcut facilitates a smooth transition to the Voice Cloning Page, enabling trainers to initiate the voice cloning process swiftly and effortlessly.

Voice Input Options:

As shown in Figure[9], Trainers are given the flexibility to either upload their voice files or record their voice directly using a built-in recording feature. This recording option offers a convenient alternative for trainers who prefer not to upload pre-recorded audio files, streamlining the voice input process and enhancing user convenience.

Child's Target Voice File: Another upload button allows trainers to upload the target voice file of the child. This file, too, should be in WAV format for optimal results.

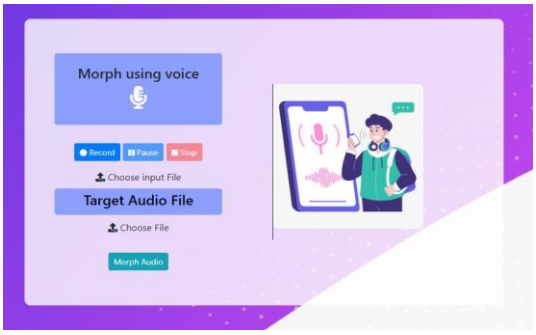


*Figure 9 – Morph using voice page*

Clone Voice Button:

Positioned prominently on the page, the "Clone Voice" button acts as the catalyst for initiating the voice cloning process. Upon clicking this button, the uploaded voice files are processed, and the voice cloning algorithm is triggered to generate the output.

Status and Error Display Area:

An area is designated on the page to display the status of the voice cloning process, keeping trainers informed about the progress and any potential issues.

In case of errors or discrepancies, this area also serves to display relevant error messages, guiding trainers on the necessary corrective actions.

Download Link for Generated Output:

Once the voice cloning process is successfully completed, a link is provided for trainers to download the generated voice cloning output file. This ensures that trainers have easy access to the cloned voice for further use or analysis.

The Text Cloning Page offers a convenient platform for trainers to clone their voice based on text input. Tailored to facilitate a straightforward and efficient cloning process, this page boasts a range of user-friendly features designed to simplify voice cloning

from text. Here's an overview of the main components and functionalities available on this page

Morph Using Text Shortcut:

As shown in [Figure 10], From the trainer's homepage, a convenient shortcut labeled "Morph Using Text" offers a streamlined pathway to the Text Cloning Page. By clicking this shortcut, trainers can swiftly navigate to the Text Cloning Page, facilitating quick and effortless initiation of the voice cloning process based on text input.

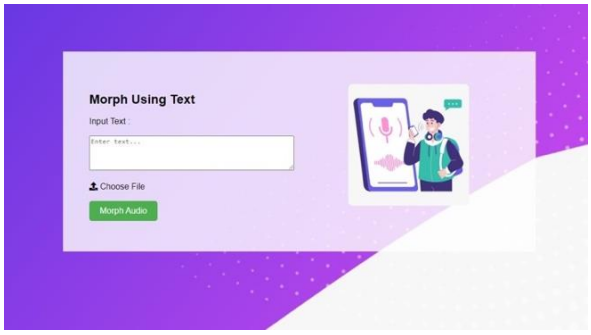


*Figure 10 – Morph using text page*

Text Input Box:

A dedicated text box is provided for trainers to enter the desired speech content. This text serves as the basis for generating the cloned voice, allowing trainers to customize the voice output according to their preferences.

Upload Button for Voice File:

Child's Target Voice File: An upload button is available for trainers to upload the target voice file of the child. This file should be in WAV format to ensure optimal compatibility and accurate voice cloning.

Clone Text Button:

Positioned prominently on the page, the "Clone Text" button acts as the trigger for initiating the voice cloning process. Upon clicking this button, the entered text and uploaded voice file are processed, and the voice cloning algorithm is activated to generate the audio output.

Status and Error Display Area:

A designated area on the page showcases the status of the voice cloning process, providing trainers with real-time updates on the progress and any potential issues that may arise.

In case of errors or discrepancies, this area also serves to display relevant error messages, guiding trainers on the necessary corrective measures.

Download Link for Generated Output:

Upon successful completion of the voice cloning process, a download link is presented to trainers, enabling them to download the generated voice cloning output file in WAV format for further use or analysis.

The "Images" button on the trainer's homepage leads to an intuitive Images Interface [15] tailored for a unique voice command generation experience. Here's a concise overview of this interface:


*Figure 11 – Trainer's Image Page*

Image Display:

As shown in [Figure 11], The interface [15] showcases a curated set of images, each representing a distinct command or comment.

Upload Button for Target Voice:

A dedicated upload button is provided for trainers to upload the target voice file in WAV format. This voice file will be used to generate the audio output corresponding to the selected image.

Command Generation:

By simply clicking on an image, trainers can instantly generate a command or comment in the target voice. The voice cloning algorithm processes the selected image and the uploaded target voice file, producing an audio output that reflects the command or comment associated with the chosen image.

Commands Like:

Sleep - It's time for bed, go get some rest.

Water - Feeling thirsty? How about a nice drink of water?

Upon clicking the "ASD Children" button on the home page, users are redirected to a specially designed Children's Interface [15]. This interface is thoughtfully crafted to cater to the unique needs and preferences of children with Autism Spectrum Disorder (ASD), aiming to facilitate their communication through innovative voice technology. Here's a detailed overview of this interface:

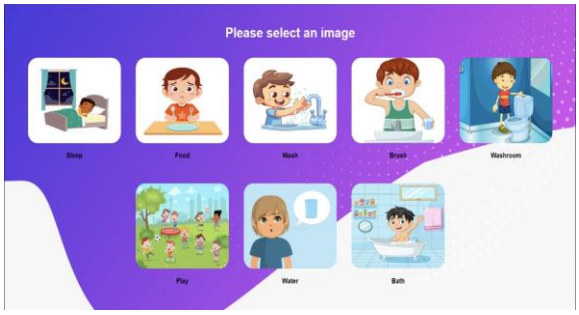


*Figure 12 – Child's Image Page*

Image Display:

As shown in [Figure 12], The interface [15] features a vibrant collection of images that are relatable and engaging for children. These images are carefully selected to represent common objects, activities, or emotions, making it easier for children to relate to and understand.

Random AI Voice Generation:

By simply clicking on an image, a random AI voice is generated. This AI voice is designed to be clear, friendly, and easy to understand, ensuring that children with ASD can easily grasp the meaning and intent behind each voice command or comment.

The AI voice serves as an interactive voice assistant, helping children to communicate their needs, express their feelings.

Requests Like:

Sleep - Can I go to bed now? I'm tired.

Water - Can I have some water? I'm thirsty.

User-Friendly Design:

The interface is designed with simplicity and ease of use in mind. Large, easily clickable images and clear instructions ensure that children can navigate the interface independently, fostering a sense of autonomy and empowerment.

## 6. Conclusion

This paper introduces a ground-breaking voice morphing tool designed to assist trainers of children with Autism Spectrum Disorder (ASD) in overcoming communication challenges. Leveraging advanced deep learning models such as Whisper and VALL-E, the application transforms a trainer's voice into a tone that is more recognizable and engaging for children with ASD, with the goal of enhancing their communication skills and educational outcomes. A key innovation lies in the utilization of state-of-the-art voice conversion techniques, which enable the modification of vocal styles while preserving linguistic content and speaker identity information.

Additionally, the tool incorporates features such as image-based communication interfaces and multilingual support, further enhancing its accessibility and effectiveness in facilitating speech and language therapy sessions for children with ASD.

## References


[1] Selina Marianna Shah; Catherine Elliott; Prema Nedungadi, Square Pegs and Round Holes: Pedagogy for Autistic Students in Computing Education, Journals & Magazines >IEEE Transactions on Education >Early Access

[2] Devika K, Venkata Ramana Murthy Oruganti, Dwarikanath Mahapatra, Ramanathan Subramanian, Outlier-based Autism Detection using Longitudinal Structural MRI, Electrical Engineering and Systems Science > Image and Video Processing

[3] Mohammad Robihul Mufid; Arif Basofi; M. Udin Harun Al Rasyid; Indhi Farhandika Rochimansyah; Abdul rokhim, Design an MVC Model using Python for Flask Framework Development, 2019 International Electronics Symposium (IES)

[4] J G Sukumar; Mohith Sai Ram Reddy; Nikhileswar Sambangi; S Abhishek; Anjali T, Enhancing salary projections: a supervised machine learning approach with flask deployment Conferences > 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA)

[5] Remya M S; Jyothiratnam; Mohith Sai Ram Reddy; Sahith Madamanchi; Prema Nedungadi, Automated Speech Correction Assistive Technology for Malayalam Articulation Errors, Conferences| 2023 Second International Conference on Informatics (ICI)

[6] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, Furu Wei, Neural Codec Language Models are Zero-Shot Text to Speech Synthesizers, Computer Science > Computation and Language

[7] Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He,Sheng Zhao, Furu Wei, Speak Foreign Languages with Your Own Voice: Cross-Lingual Neural Codec Language Modeling, Computer Science > Computation and Language

[8] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, Daan Wierstra, Deep AutoRegressive Networks, 31st International Conference on Machine Learning, PMLR 32(2):1242-1250, 2014.

[9] Yiren Wang, Fei Tian, Di He, Tao Qin, ChengXiang Zhai, Tie-Yan Liu, Non-Autoregressive Machine Translation with Auxiliary Regularization, Vol. 33 No. 01: AAAI-19, IAAI-19, EAAI-20 / AAAI Technical Track: Machine Learning

[10] Xuedong Huang; A. Acero; F. Alleva; Mei-Yuh Hwang; Li Jiang; M. Mahajan, Microsoft Windows highly intelligent speech recognizer: Whisper, Conferences > 1995 International Conference on Acoustics, Speech, and Signal Processing

[11] Taisuke Ito, Kazuya Takeda, Fumitada Itakura, Analysis and recognition of whispered speech, Speech Communication, Volume 45, Issue 2, February 2005, Pages 139-152

[12] Susmitha Vekkot, Building a generalized model for multi-lingual vocal emotion conversion, Conferences, 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)

[13] Hieu-Thi Luong; Junichi Yamagishi, NAUTILUS: A Versatile Voice Cloning System

Journals & Magazines, IEEE/ACM Transactions on Audio, Speech, and Language Processing

[14] Author PictureRyan Levering, Author PictureMichal Cutler, The portrait of a common HTML web page, DocEng '06: Proceedings of the 2006 ACM symposium on Document engineeringOctober 2006Pages 198–204

[15] Cámara, Mateo; Blanco, José Luis, Expanding the Frontiers of Web Audio With Autoencoders and JavaScript, JAES Volume 70 Issue 11 pp. 979-989; November 2022xf