NATIONAL RESEARCH UNIVERSITY

HIGHER SCHOOL OF ECONOMICS

Graduate School of Business

Vu Minh Duc

**Intelligent Offline Voice Assistant for Daily Task Automation**

SOTA

Field of study: Business Informatics

Degree program: Business Analytics and Big Data Systems

|  | Academic Supervisor  Research Assistant  Majid Sohrabi |
| --- | --- |

**CONTENTS**

[**Introduction 4**](#_heading=h.30j0zll)

[**Business side overview 5**](#_heading=h.3znysh7)

[**Architecture side overview 7**](#_heading=h.tddmejufslsb)

[**Technical side overview 9**](#_heading=h.2dpdjvuh8545)

[3.1 The NLU architecture 9](#_heading=h.fqcqo1stdbhx)

[3.2 The wakeword architecture 10](#_heading=h.jqlkwv9eein2)

[3.3 The speech to text architecture 12](#_heading=h.jk47zectfvsn)

[3.4 The text to speech architecture 13](#_heading=h.u6xspar4qaii)

[3.5 The audio induced reaction architecture 14](#_heading=h.n0sowpk2wabs)

[**Problem statement 17**](#_heading=h.3dy6vkm)

[**References 18**](#_heading=h.1t3h5sf)

[**Appendix 20**](#_heading=h.4d34og8)

**ABBREVIATIONS**

| Abbreviation | Description |
| --- | --- |
| AI | Artificial Intelligence; |
| ASR | Automatic Speech Recognition; |
| NLP | Natural Language Processing |
| TTS | Text to Speech |
| **STT** | Speech to Text |
| **UI** | User Interface |
| **VA** | Voice Assistant |
| **NLU** | Natural Language Understanding |
| **NLG** | Natural Language Generation |
| **IoT** | Internet of things |
| **MVP** | Minimum Valuable Product |

# Introduction

Voice assistants are software agents that are designed to interpret human speech and respond via synthesized voices. These systems have been for a while and are now being integrated in smartphones, dedicated home speakers, automobiles and becoming increasingly popular. Voice assistants can perform a variety of tasks, such as answering the question, controlling the connected to the systems devices, managing some simple automations like scheduling the reminders or even have dialog conversations to support humans mentally. Some of the most common voice assistants including Apple’s Siri[1], Amazon Alexa[2], Microsoft Cortana[3], Yandex Alisa[4] and Google’s Assistant[4] are pretty competitive for daily life. However, such systems lack the offline assistant’s abilities and also restrict the amount of customizations.

This research project provides an implementation of the assistant that can be fully offline reacting to the surrounding sounds and human voices and generating various interactions feedback which includes the voice feedback and launching the subprograms. To construct and develop this voice assistant, one needs to use several customs modified and open-source technologies in combinations. These include:

1. Speech recognition system (ASR)
2. Speech to text system (STT)
3. The basement for the voice AI logic
4. Text dialog conversations feedback system
5. Text to speech system (TTS)
6. Sound induced reaction system

# Business side overview

A voice assistant is a digital program that humans can interact with through voice, allowing users to input information into a machine by talking instead of typing, making them an efficient way to enhance productivity while keeping human error to a minimum.

To understand what voice assistants are, we need to understand the purpose of such tools. These are software apps that are able to understand human language and perform tasks based on spoken context. These tools have been implemented in the market for products from big companies.  
By the end of 2024, there will be 145.1 million voice assistant users in the US, with growth expected to hover around 3% each year through the end of the forecast period in 2027. Among the top voice assistant companies in 2023, Google Assistant is the most popular with US consumers at 85.4 million users, followed by Apple’s Siri (81.1 million) and Amazon’s Alexa (73.7 million).

Consumers see voice assistants as the smarter, faster, and easier way to perform everyday activities. Yet, for more serious situations involving money (shopping, refund on an airline ticket, etc.), consumers prefer what they already know and trust — at least for now. Trust remains a barrier for voice assistant shoppers.

The voice assistant application market is projected to grow at 27.3% CAGR during the forecast period. Voice assistant applications can follow voice commands and assist in doing routine work like placing an online order, scheduling an appointment, switching on connected lights, and acting as a hassle-free facilitator for texting or calling.

In terms of application, the voice assistant market segment includes messenger bots, websites, and contact centers. In 2021, the contact centers segment had a huge market revenue share. The combination of voice assistants in contact centers, which tends to help understand customer insights better and improves employee performance, is attributed to the growth of this segment.

In terms of regional market share, in 2023, North America accounts for the largest market share in the Voice Assistant Application Market.

In the healthcare sector, voice assistants benefit greatly by delivering improved services to patients, offering remote therapy, facilitating doctor appointments through teleconferencing, observing patients remotely, and arranging clinical meetings, among others.

The drawbacks of all commercial products is the lack of customizations and as most systems require the global internet connection to get the offline tasks like turn the music in the specific folder on, scheduling the needed event, asking the timers, do simple math calculations and so on. The philosophy of the offline systems was made by MyCroft[5]. While the system is made pretty versatile, the product has several cons that require us to create our own AI assistant. The mentions cons are:

Bad and non-human speaking

Lack of the human dialog conversation

Lack of ambient audio understanding

First of all, creating a new product such as hardware or software is not an easy and fast task. Take a look at Figure 1. The business requires releasing products to be as fast, as cheap and as good as possible. But it's an unreachable task, so we need to balance between members. Using open-source projects and using the project that has been done can significantly increase the development speed.



Figure 1. The good fast cheap graph

From a business perspective, voice assistants can offer several advantages. They can help drive down customer wait times, manage meeting schedules, and improve customer service and customer experience. However, it's important to note that the development process for voice assistants requires a depth of knowledge and experience in voice recognition technology.

# Architecture side overview

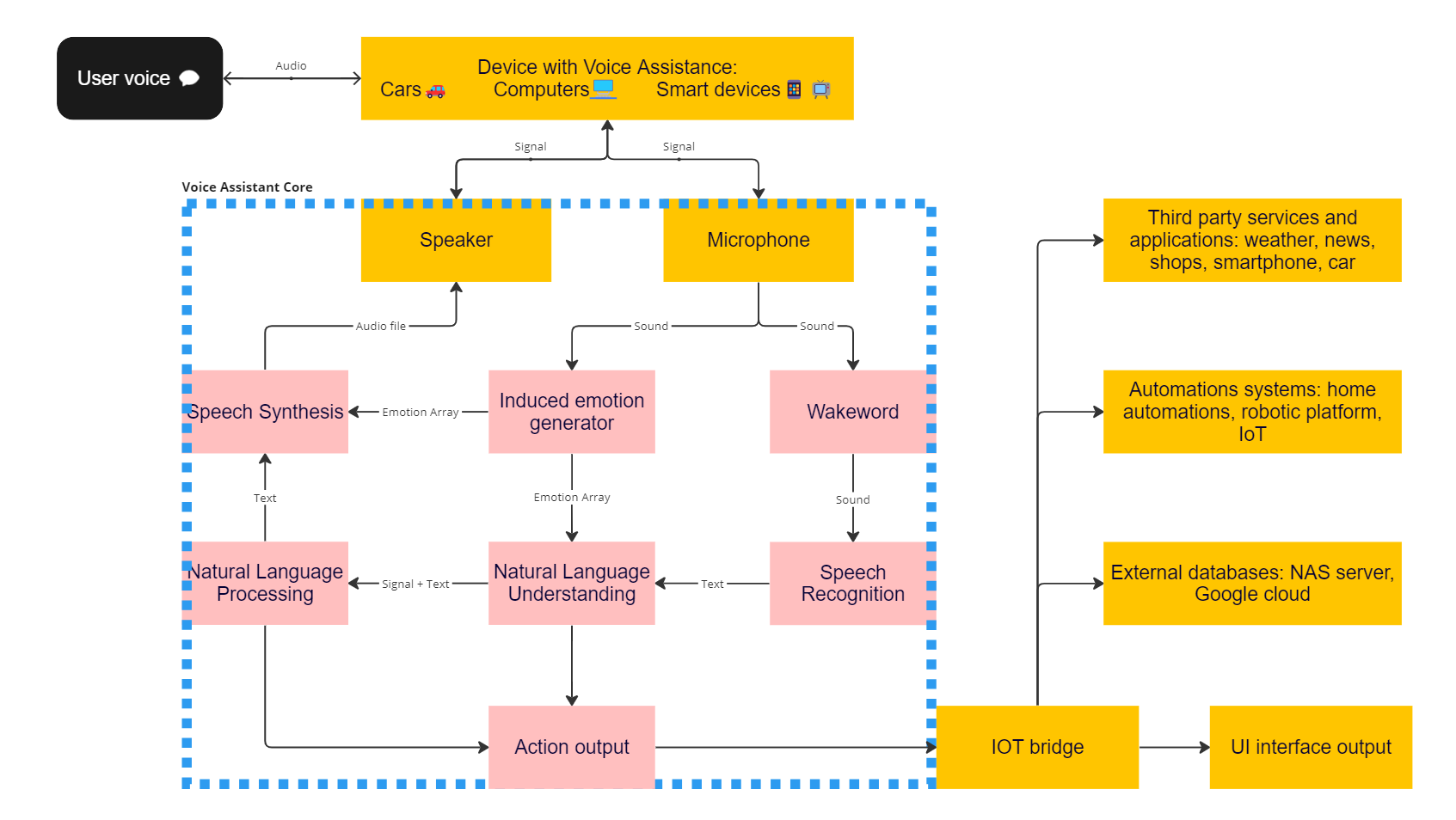


Figure 2. Voice assistant architecture

The picture above shows the architecture implementation of most of the VA systems. The communications sources could be a variety of devices. I could be a desktop computer with a background runned voice assistant or it could be a smart home gadget, or an automobile that has the computer integrated inside. The program first initiates the audio recording device. It creates an audio chunk with the fixed time frame that is remembered in the heap memory.   
The audio chunk is used for the machine feeling. This is a normal classification model of the environment. Creating the response feeling is a complicated topic, the science term of this type is induced emotion. Also the audio chunk parses the first wakeword module which is the first type of neural network system. The wake word architecture is the super lightweight model for detecting the keywords for activating the systems. “Alexa", "Siri", "Weather", "5 minutes timer” and an example of keywords. After the keyword has been detected the system will execute the Natural Language Processing(NLP) model for converting the audio data to the text data. The text data is parses to the Natural Language Understanding(NLU) then could analyze by several modules:

the action selection from the activation text like “open the music", "close the door", "close the curtain", "check the kitchen” and so on.

the neural network dialog responses for a human like conversation as the text feedback

As the response the text feedback converts back to the audio using Text to Speech(TTS) neural network module. This module uses the text input, and the sample audio as the voice intensity and creates the voice with desired emotion.

Developing a software project is a time-consuming process. There are several technical steps to create the product:

Implement a Back-End for the Assistant: The back-end of a voice assistant is the back-bone that holds everything together. It’s a core program that runs the logic interaction and implements the deep neural network architecture to get human-like behavior and also maintain stable connection with other devices.

Implement a Front-End for the Assistant: The front-end of your voice assistant is the interface that users will interact with. This could be a button that invokes the speech recognition subroutine and a text-label to provide feedback to the user.

Develop an AI Ecosystem: This involves creating a system of software that can work harmoniously together, creating a cohesive user experience. By leveraging the interconnectivity and compatibility between devices, tech ecosystems aim to simplify our lives and improve overall efficiency.

Integrate with Existing Systems: The voice assistant should be able to integrate with your existing systems. For example, if you're developing a voice assistant for a CRM system, the assistant should be able to update records related to that system.

Test and Iterate: After implementing the above steps, it's important to test the voice assistant and iterate based on the feedback received.

# Technical side overview

## 3.1 The NLU architecture

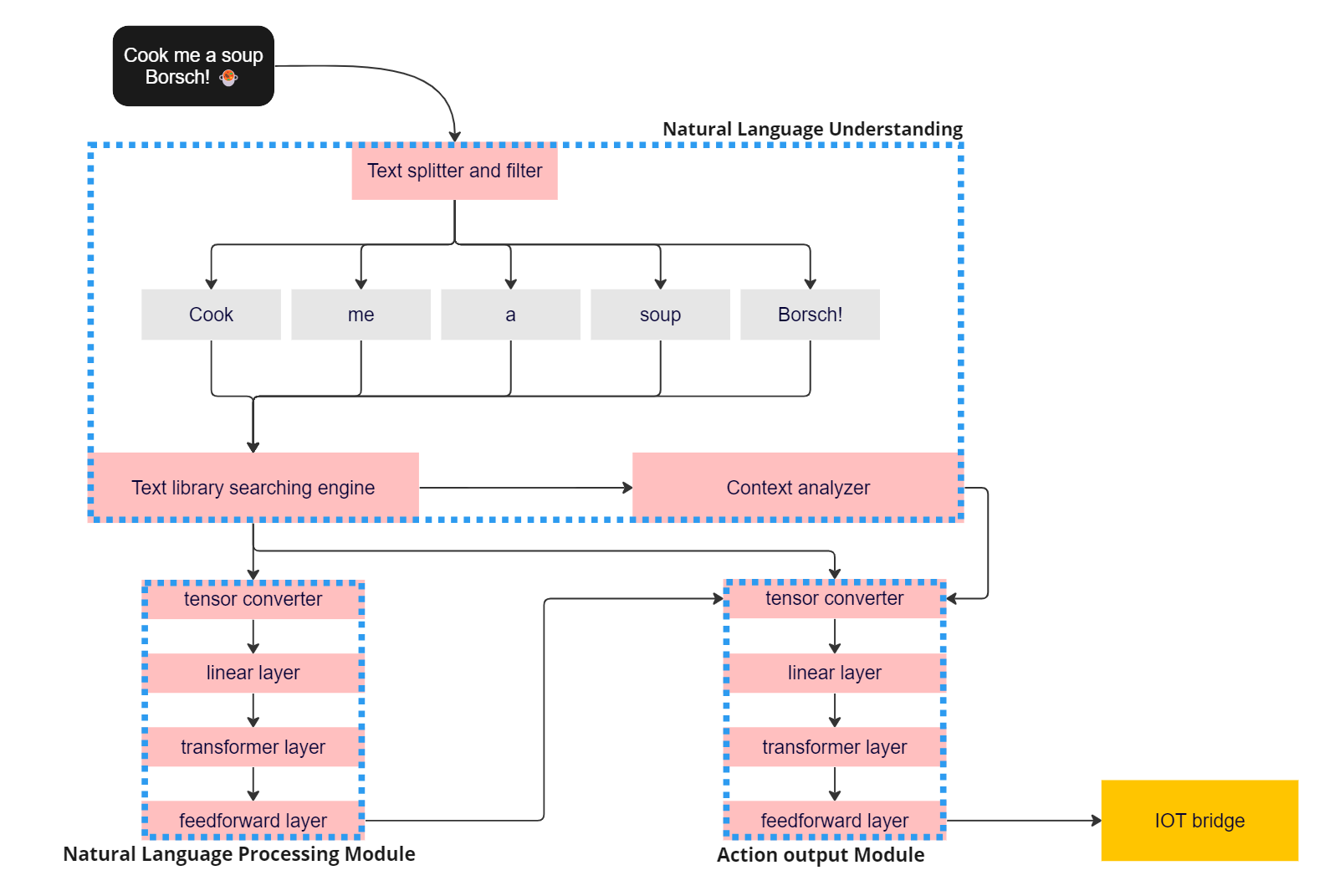


Figure 3. The NLU architecture

Deep learning models have significantly contributed to advancements in various domains such as computer vision, natural language processing, and reinforcement learning. Convolutional Neural Networks (CNNs)[6] have been the backbone of many state-of-the-art computer vision tasks, such as image classification, object detection, and image segmentation. Models like ResNet[7] and EfficientNet[8] have achieved excellent performance on benchmark datasets like ImageNet. These models are applied in various applications and continue to evolve.

In the context of Natural Language Understanding (NLU), there have been developments like GPT2[9] which uses the revolutionary attention neural network architecture. An example of this is the DialoGPT[10], a large, tunable neural conversational response generation model. It's trained on numerous conversation-like exchanges extracted from Reddit comment chains over a period spanning from 2005 through 2017. It extends the Hugging Face[11] PyTorch transformer to attain performance close to humans both in terms of automatic and human evaluation in single-turn dialogue settings. Conversational systems that leverage DialoGPT generate more relevant, context-consistent responses than baseline systems. These developments have been publicly released to facilitate research into neural response generation and the development of more intelligent open-domain dialogue systems.

In the construction of a general NLU architecture, there are two primary components: Natural Language Understanding (NLU) and dialogue management. NLU is responsible for intent classification, entity extraction, and response retrieval. It processes user utterances using an NLU model that is generated by the trained pipeline. The dialogue management component decides the next action in a conversation based on the context. This scalable architecture allows for the fine-tuning of the model to better suit specific tasks and phrases.

## 3.2 The wakeword architecture

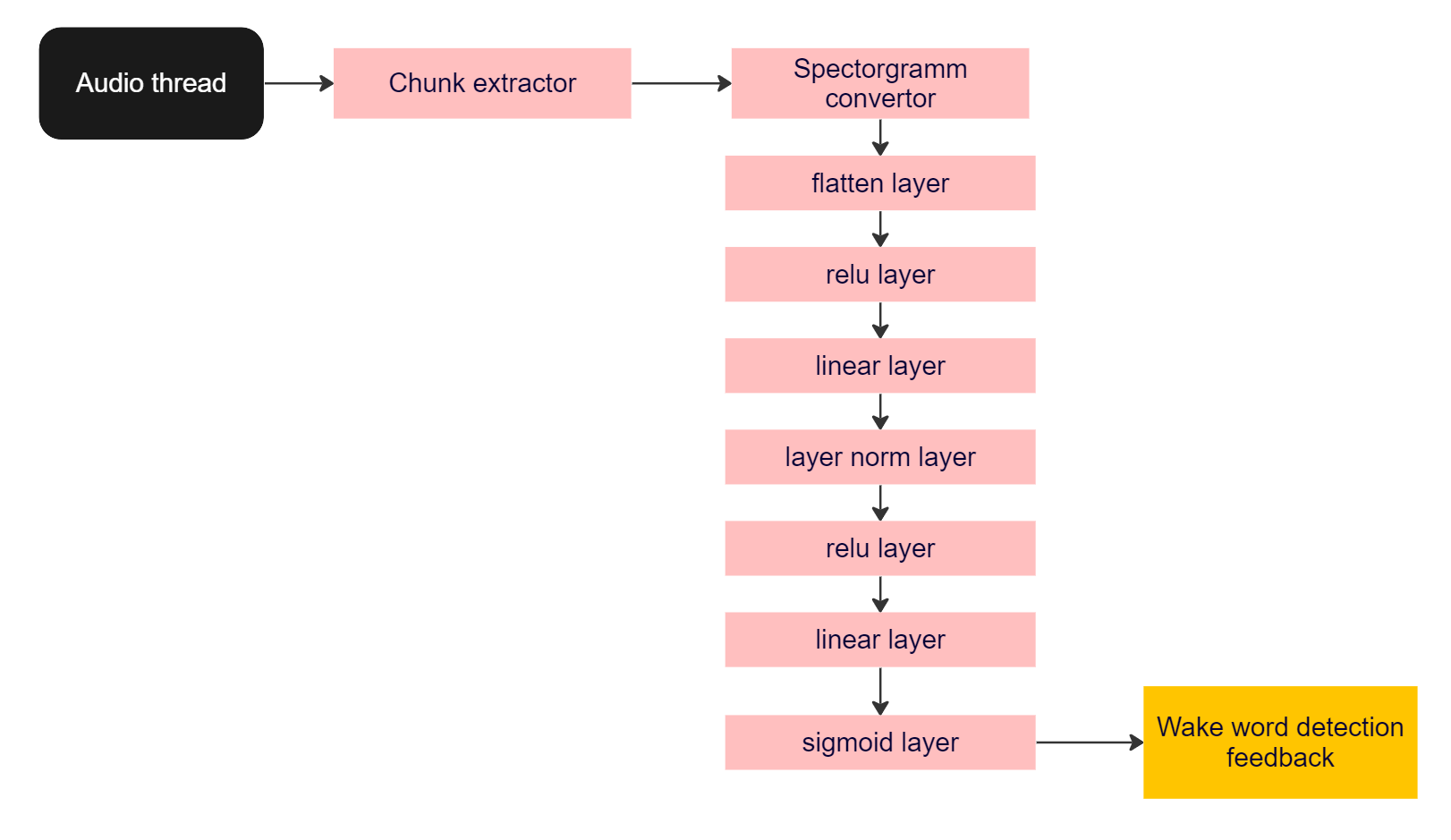


Figure 4. The wakeword architecture

The wakeword system has four high-level goals:

* Be fast enough for real-world usage, while maintaining ease of use and development. For example, running in real-time a single core of a Raspberry Pi 3[12] or even on embedded systems or micro-controllers.
* Be accurate enough for real-world usage. The included models typically have false-accept and false-reject rates below the annoyance threshold for the average user. This is obviously subjective, by a false-accept rate of <0.5 per hour and a false-reject rate of <5% is often reasonable in practice.
* Have a simple model architecture and inference process. All models also have a shared feature extraction backbone, so that each additional model only has a small impact to overall system complexity and resource requirements.
* Require little to no manual data collection to train new models. The included models were all trained with 100% synthetic speech generated from text-to-speech models. Training new models is a simple as generating new clips for the target wake word/phrase and training a small model on top of of the frozen shared feature extractor.

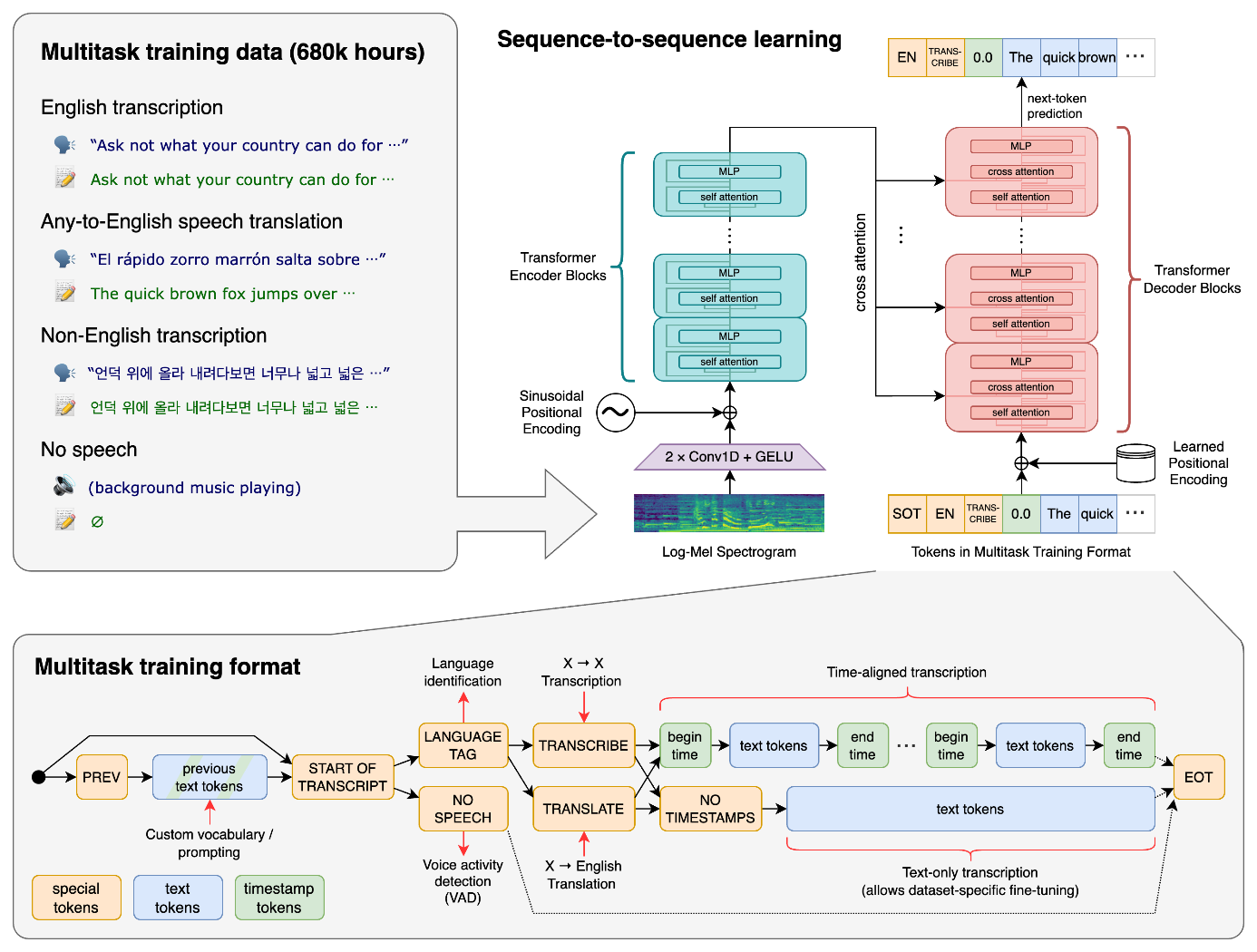
The Figure 4 shows the architecture implementation of the wake word neural network. The architecture of a wake word neural network typically involves a two-stage detection process that includes an initial detection step and a secondary confirmation step. This process is followed by a deep neural network for wake-up-word speech recognition.

In the initial detection stage, the system continuously listens to the environment for potential wake words. This stage uses a relatively smaller and less computationally intensive neural network model. Its main purpose is to efficiently process audio input and make a preliminary determination on whether a wake word has been spoken.

In the second stage, if the initial network detects a potential wake word, the audio snippet is passed to a larger, more complex neural network. This network has been trained to accurately identify the wake word with a high degree of specificity. It analyzes the audio snippet in more detail to confirm whether the wake word was indeed spoken.

The deep neural network used for wake-up-word speech recognition is typically a type of recurrent neural network (RNN)[13] known as a Long Short-Term Memory (LSTM)[14] network, which is well-suited for processing time-series data such as audio signals.

## 3.3 The speech to text architecture

 Figure 5. OpenAI Whisper STT model

Speech-to-text technology has become an essential tool for increasing productivity and saving time. It can be used to create text notes, emails, blog posts, reports, and more by converting spoken words into text in real-time. There are several tools and approaches available for implementing speech-to-text functionality, each with their pros and cons.

In this project we choose a general-purpose speech recognition model from OpenAI - Whisper[15]. Whisper is an automatic speech recognition (ASR) system trained on 680,000 hours of multilingual and multitask supervised data collected from the web. It is trained on a large dataset of diverse audio and is also a multitasking model that can perform multilingual speech recognition, speech translation, and language identification.

It is trained in a large and diverse dataset that leads to improved robustness to accents, background noise and technical language. Moreover, it enables transcription in multiple languages, as well as translation from those languages into English. Moreover, the project provides open access to models and inference code that serves as a foundation for building useful applications and for further research on robust speech processing.

## 3.4 The text to speech architecture

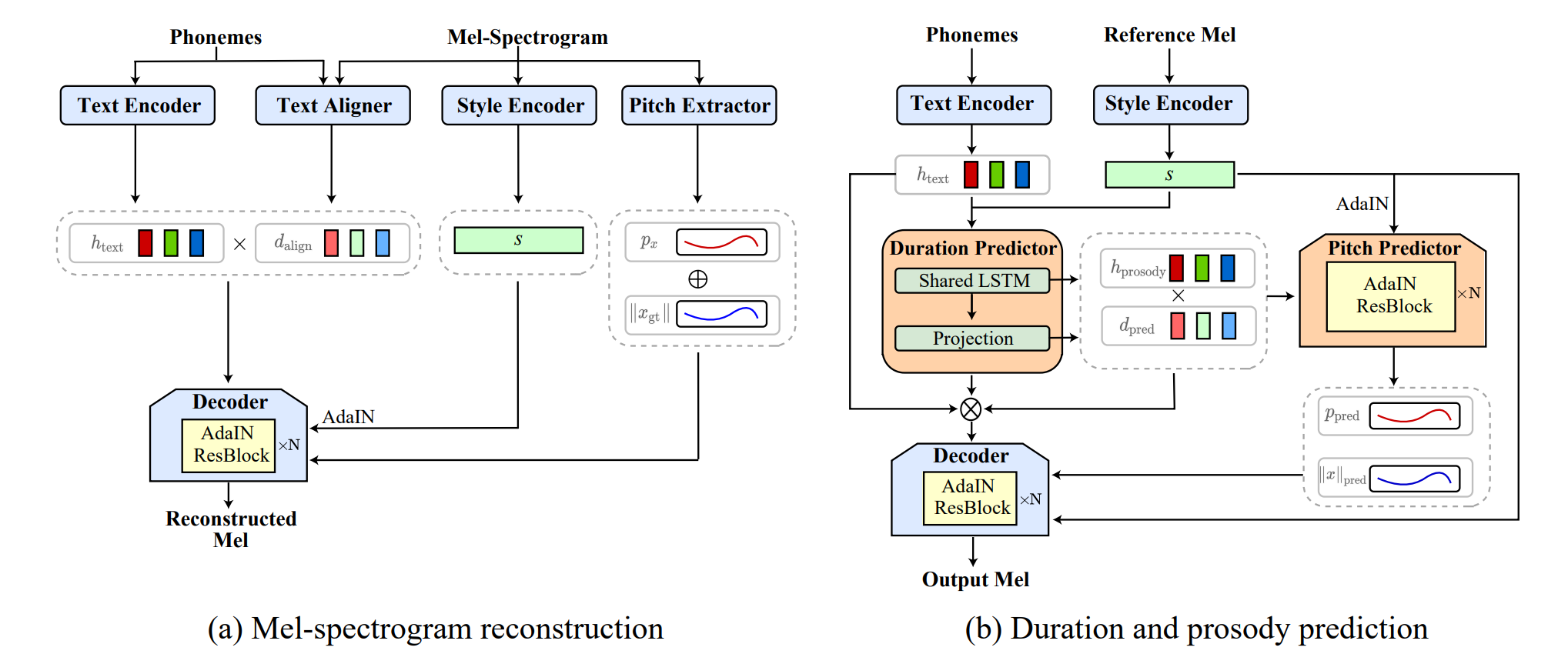


Figure 6. StyleTTS architecture model

Text-to-Speech (TTS) has recently seen great progress in synthesizing high-quality speech owing to the rapid development of parallel TTS systems, but producing speech with naturalistic prosodic variations, speaking styles and emotional tones remains challenging. Moreover, since duration and speech are generated separately, parallel TTS models still have problems finding the best monotonic alignments that are crucial for naturalistic speech synthesis. When it comes to building a voice assistant with open-source Text-to-Speech (TTS) components, there are several options to consider. The successful model could be mentioned as Tacotron2[16], FastSpeech 2[17] and the recently developed StyleTTS[18] model.

Here, the proposed StyleTTS, a style-based generative model for parallel TTS that can synthesize diverse speech with natural prosody from a reference speech utterance. With novel Transferable Monotonic Aligner (TMA) and duration-invariant data augmentation schemes, our method significantly outperforms state-of-the-art models on both single and multi-speaker datasets in subjective tests of speech naturalness and speaker similarity. Through self-supervised learning of the speaking styles, our model can synthesize speech with the same prosodic and emotional tone as any given reference speech without the need for explicitly labeling these categories.

## 3.5 The audio induced reaction architecture

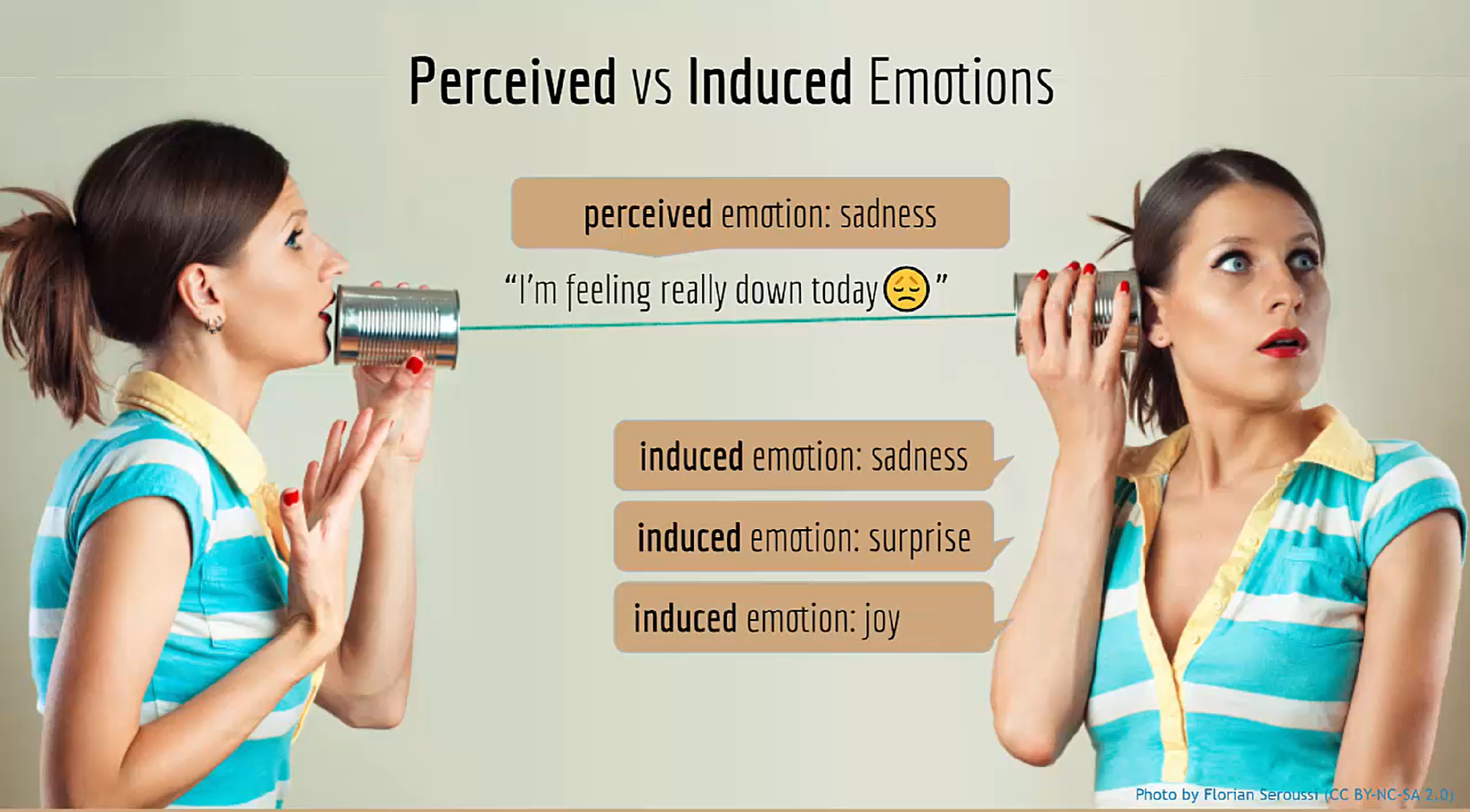


Figure 7. Perceived and induced emotions

Induced emotions refer to the emotional responses that are elicited or brought about by certain situations, experiences, or stimuli. Figure 7. shows the difference between perceived(the emotion that someone has created) and induced (the emotion that someone has reacted to). In the context of research, induced emotions are often used as a way to measure and understand human emotional responses.

Creating a dataset of induced emotions can be challenging for several reasons:

* Variability in Human Emotions: Emotions are highly individual and can vary greatly from person to person. What induces one person to feel a certain emotion might not have the same effect on another person. This variability can make it difficult to create a dataset that accurately represents a wide range of emotional responses .
* Subjectivity in Emotion Labeling: Emotions are subjective and can be interpreted differently by different people. This can make it difficult to label emotions in a consistent and objective manner. For example, what one person might perceive as "happiness" might be perceived differently by another person.
* Noise in Data: Emotions are often associated with physiological responses, such as changes in heart rate or brain activity. These physiological responses can be influenced by a variety of factors, including physical discomfort or fatigue, which can introduce noise into the data. This noise can make it difficult to accurately measure and interpret the emotional responses.
* Complexity of Emotional Responses: Emotional responses are not just about the emotion itself, but also about how that emotion is expressed and experienced. This can make it difficult to capture all aspects of an emotional response in a dataset.

Despite these challenges, researchers continue to develop methods and tools to create and analyze datasets of induced emotions, in order to better understand human emotional responses and their impact on various aspects of life.

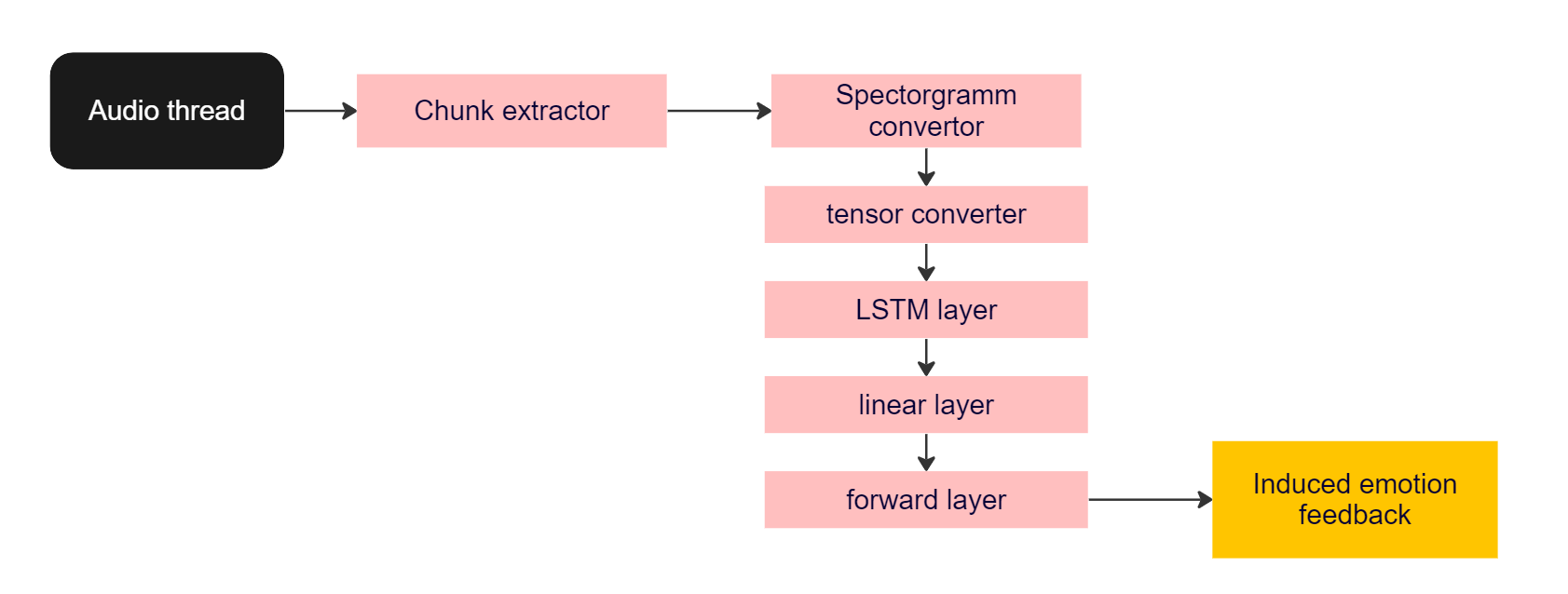


Figure 8. Induced emotions feedback architecture

Sound classification is one of the approaches for the emotion response applications in audio deep learning. It involves learning to classify sounds and predict the category of that sound. This can be applied to various practical scenarios, such as classifying music clips to identify the genre or classifying short utterances by a set of speakers to identify the speaker based on their voice.

Creating datasets for induced emotion labels are scarce but of utmost importance for the reaction task. Because of the lack of available and pre labeled datasets we decided to manually create the emotion reaction database with 26 varieties of emotions categories. These are "Admiration", "Amusement", "Anger", "Annoyance", "Approval", "Caring", "Confusion", "Curiosity", "Desire", "Disappointment", "Disapproval", "Disgust", "Embarrassment", "Fear", "Gratitude", "Grief", "Joy", "Love", "Nervousness", "Optimism", "Pride", "Realization", "Relief", "Remorse", "Sadness" and “Surprise”.

Figure 8 shows the architecture implementation of sound classification. The typical approach involves converting the chunk of sound files into spectrograms and feeding these spectrograms into a CNN model to produce predictions about the class to which the emotions should belong.

**Conclusion**

We can conclude that voice assistants are digital programs that allow users to interact with their devices through voice commands, thereby enhancing productivity and reducing human error. These assistants have a wide range of applications, from everyday tasks to more complex operations in sectors like healthcare. However, they also face challenges such as lack of customization and the need for a global internet connection for offline tasks.

The development of voice assistants involves several steps, including creating a back-end, a front-end, an AI ecosystem, integrating with existing systems, and testing and iterating based on feedback. The architecture of voice assistants involves various components like Natural Language Understanding (NLU), Wakeword detection, speech-to-text conversion, text-to-speech synthesis, and audio-induced emotion response.

In terms of market growth, the voice assistant application market is projected to grow at 27.3% CAGR during the forecast period. North America is expected to hold the largest market share in 2023. However, for more serious situations involving money, consumers still prefer traditional methods.

While voice assistants offer many advantages, it's important to note that their development requires a depth of knowledge and experience in voice recognition technology. Despite the challenges, the future of voice assistants looks promising, with increasing adoption and continuous advancements in technology.

The collection and analysis of information on existing technologies of deep learning networks for voice recognition and interaction was carried out. As a result of the analysis and testing of technologies, it was concluded that the custom VA development is most suitable for the task purpose.

The application will be tested and shown to work in real conditions of the Smart Home system, which allows the implemented application to be further used by users. In future plans to release this application on github for free download and use, it is planned to develop and add new functionality.

# 

# Problem statement

Current Limitations of Voice Assistants: While voice assistants have become increasingly popular and integrated into our daily lives, they still have limitations. For instance, they often require an internet connection to function, which can be a problem in areas with poor connectivity or during power outages.

Lack of Offline Voice Assistants: There is a notable lack of offline voice assistants in the market. Most existing voice assistants, such as Siri, Google Assistant, and Amazon Alexa, require an internet connection to function.

Daily Task Automation: The current voice assistants can perform a variety of tasks, but they often struggle with daily task automation. They may not be able to perform tasks that require a high level of precision, such as cooking or driving, without human supervision.

Privacy Concerns: There are also privacy concerns associated with voice assistants. Users are hesitant to use voice assistants for tasks that involve sensitive information, such as making purchases or accessing personal data, due to the risk of unauthorized access pwc.com.

Need for Customization: There is a need for voice assistants that can be customized to better suit the user's needs and preferences. This includes not only the ability to understand and respond to the user's voice commands but also the ability to learn from the user's habits and preferences over time.

"Despite the growing popularity and capabilities of voice assistants, there is a significant gap in the market for offline voice assistants that can perform daily task automation. These assistants should be able to function without an internet connection, be capable of performing a wide range of tasks, respect user privacy, and offer customization options to better suit the user's needs and preferences."

# References

1. Apple’s Siri. <https://www.apple.com/siri/>.
2. Amazon Alexa. <https://alexa.amazon.com/>.
3. Microsoft Cortana. <https://www.microsoft.com/en-us/cortana>.
4. Yandex Alisa. <https://yandex.ru/alice>.
5. Google’s Assistant. <https://assistant.google.com/>.
6. Keiron O'Shea, Ryan Nash. An Introduction to Convolutional Neural Networks.  
   https://arxiv.org/abs/1511.08458.
7. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun Deep Residual Learning for Image Recognition. <https://arxiv.org/abs/1512.03385>.
8. Mingxing Tan, Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. <https://arxiv.org/abs/1905.11946>.
9. Bowen Tan, Zichao Yang, Maruan Al-Shedivat, Eric P. Xing, Zhiting Hu. Progressive Generation of Long Text with Pretrained Language Models. h[ttps://arxiv.org/pdf/2006.15720.pdf](https://arxiv.org/pdf/2006.15720.pdf).
10. Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, Bill Dolan. DialoGPT: Large-Scale Generative Pre-training for Conversational Response Generation. <https://arxiv.org/abs/1911.00536>.
11. The AI community building the future. The platform where the machine learning community collaborates on models, datasets, and applications. <https://huggingface.co/>.
12. Raspberry Pi 3 Model B Single-board computer with wireless LAN and Bluetooth connectivity. <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>.
13. Robin M. Schmidt. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. <https://arxiv.org/abs/1912.05911>.
14. Christian Bakke Vennerød, Adrian Kjærran, Erling Stray Bugge. Long Short-term Memory RNN. <https://arxiv.org/abs/2105.06756>.
15. Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, Ilya Sutskever. Robust Speech Recognition via Large-Scale Weak Supervision. <https://arxiv.org/abs/2212.04356>.
16. Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, Yonghui Wu. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. <https://arxiv.org/abs/1712.05884>.
17. Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, Tie-Yan Liu. FastSpeech 2: Fast and High-Quality End-to-End Text to Speech. <https://arxiv.org/abs/2006.04558>.
18. Yinghao Aaron Li, Cong Han, Nima Mesgarani. StyleTTS: A Style-Based Generative Model for Natural and Diverse Text-to-Speech Synthesis. https://arxiv.org/abs/2205.15439.

# Appendix

Voice assistants have seen significant growth in popularity and usage. As of the end of 2024, there are expected to be 145.1 million voice assistant users in the US, with a projected growth rate of around 3% each year through 2027. Among the top voice assistant companies in 2023, Google Assistant is the most popular with US consumers at 85.4 million users, followed by Apple’s Siri (81.1 million) and Amazon’s Alexa (73.7 million).

The architecture of most voice assistants involves several components. The communications sources could be a variety of devices. The program first initiates the audio recording device, which creates an audio chunk. This audio chunk is used for the machine to feel. Also the audio chunk parses the first wakeword module, which is a type of neural network system used for detecting the keywords for activating the systems. After the keyword has been detected, the system will execute the Natural Language Processing (NLP) and Natural Language Understanding (NLU) models and could be used and processed by other applications.

The technical overview of a voice assistant involves several components:

* Natural Language Understanding (NLU) architecture
* Wakeword architecture
* Speech to Text model architecture
* Text to Speech model architecture
* Audio Induced Reaction architecture