A presentation on

REAL-TIME VOICE-CONTROLLED OFFLINE BASED GAME FOR PHYSICALLY CHALLENGED PEOPLE

Course Code: ECE 452

Course Title: Project and Thesis

Presented by

Student ID: 1702118
Session: 2017

Student ID: 1702127
Session: 2017

Student ID: 1702155
Session: 2017

Purpose of the project

- Implementation of VOSK for offline based games
- Only option for gamers who are physically disabled
- Online based games have greater delay
- To introduce voice command games
- Online Games are costly but offline games are free to use
- Intriguing and under appreciated
- Players can interact with the game without having the knowledge of gaming consoles
- To provide a new and better experience in video games
- Gamers can use multiple control (both physical control consoles such as keyboard , mouse, joystick and voice control)
- To avoid utilizing physical controllers which are costly

➢ What our proposed method offers:

- Better Voice command Accuracy
- Gamers can easily integrate new voice commands
- Gamers can easily modify previously registered voice commands
- Excellent Compatibility (with both systems and languages)

➢Overall System Requirements:

- Personal Computer with following configuration:
  - INTEL CORE i5 3.5 GHz
  - 8 GB RAM
  - 1 TB SSD as storage device
  - NVIDIA MX150 GPU
- Decent quality Microphone
- IDE (Pycharm , Proton , VScode)
- Python 3.9 or 3.10 (tested)

➢ Python libraries used in the project:

✓ VOSK

- KaldiRecognizer (c++ based open-source voice recognition toolkit)

✓ Pyaudio

- Speech Recognition Through Microphone
- Play and record Audio on a variety of platforms

✓ Time

- Performs a variety of time-related tasks

✓ Threading

- Threaded Objects and functions
- Parallel for loops
- Generates events that are regularly spaced in time

✓ Random

❖ Game Engine Modules:

I. Tkinter
II. Turtle
III. PyGame

FLOWCHART

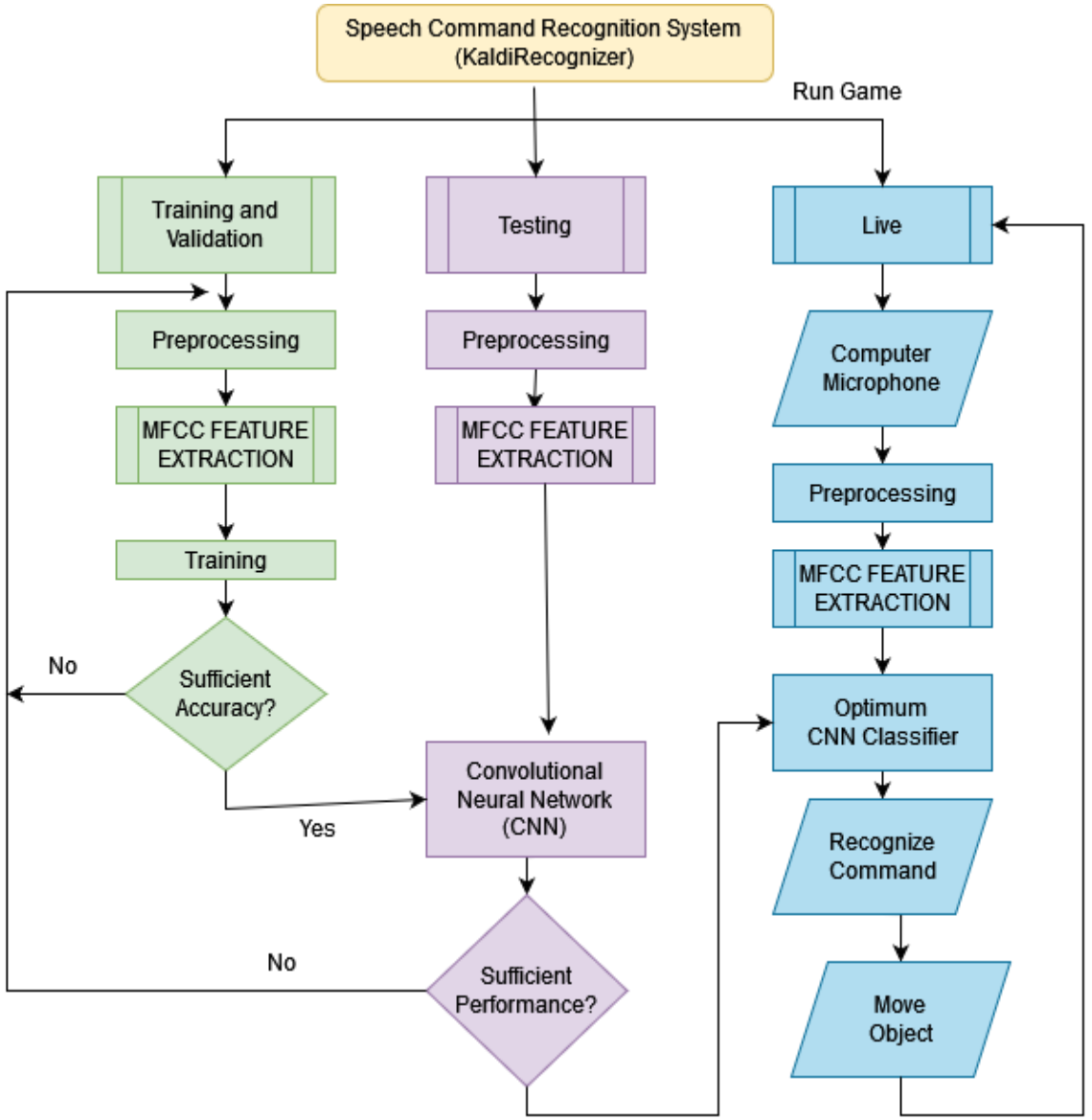


Fig_1: Overall Project Methodology

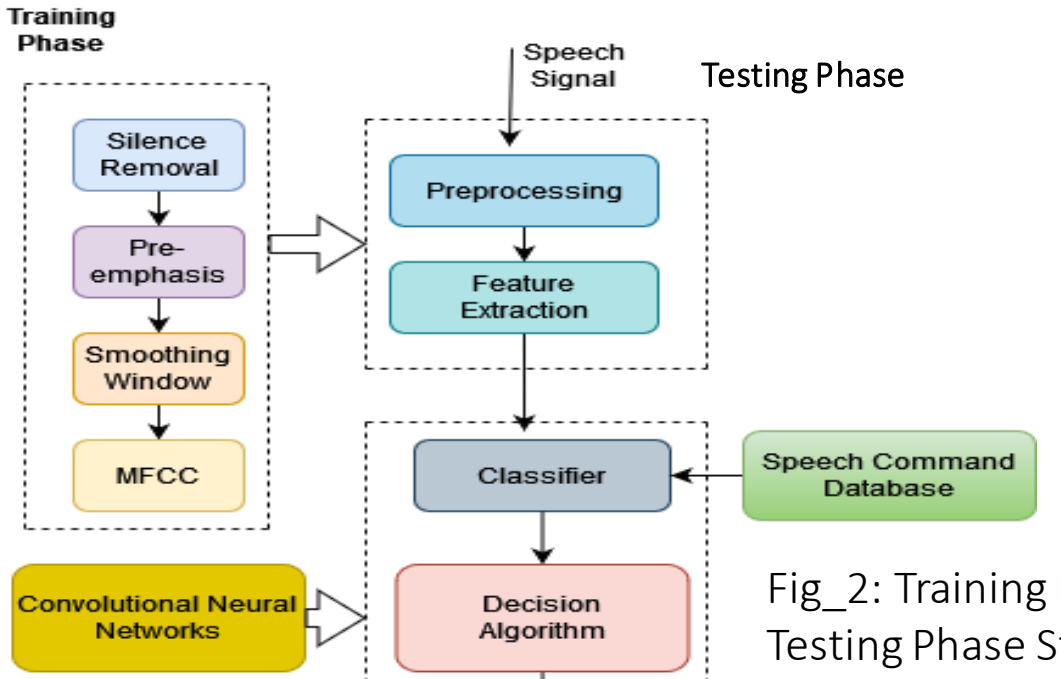


Fig_2: Training Phase & Testing Phase Steps

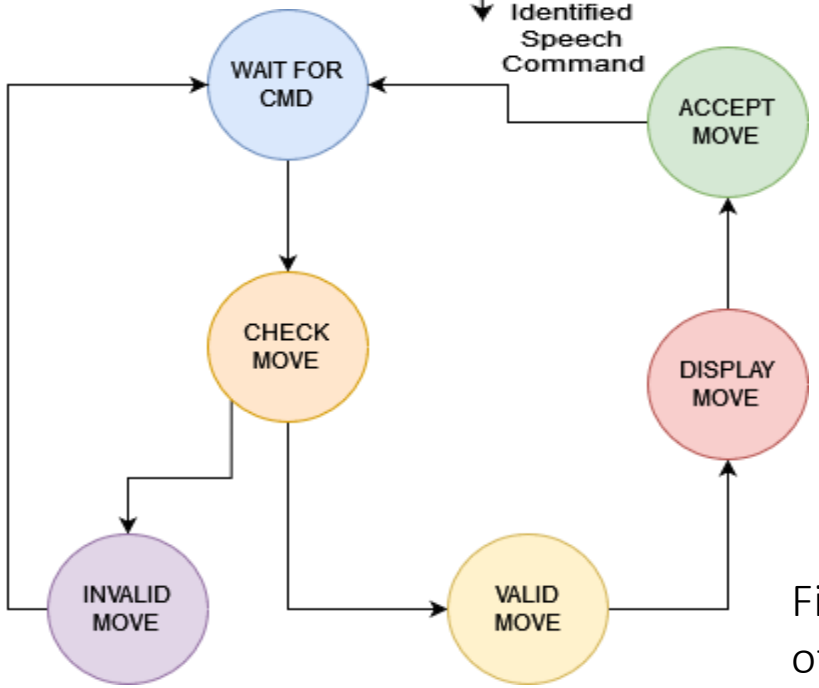
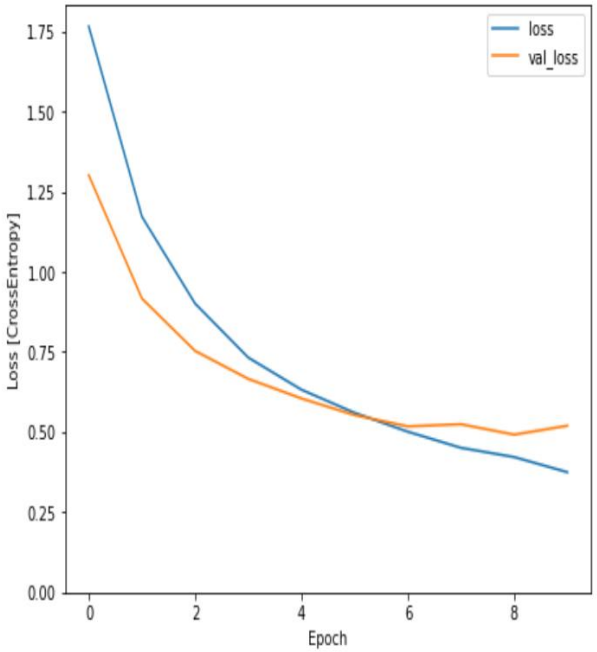
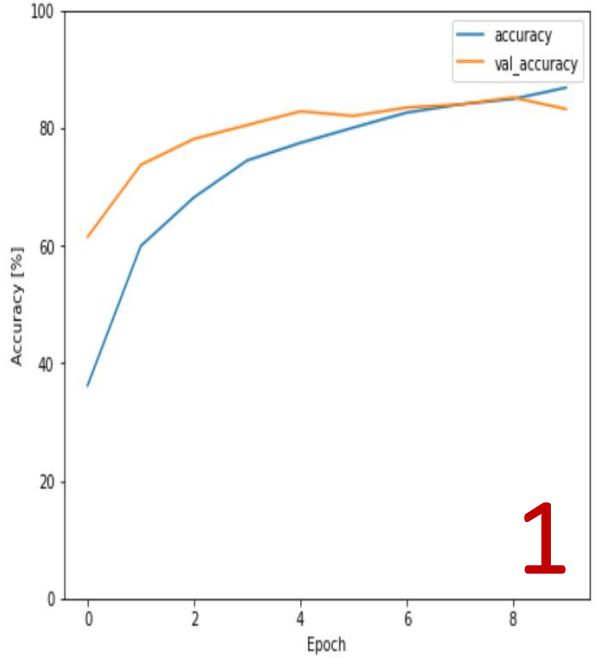


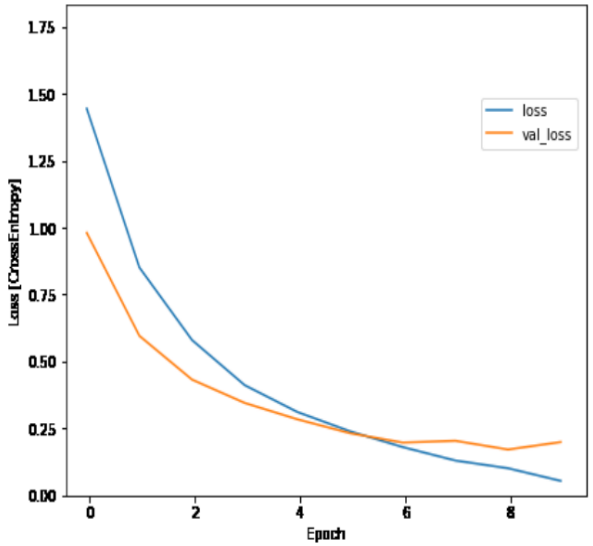
Fig_3: State diagram of game engine

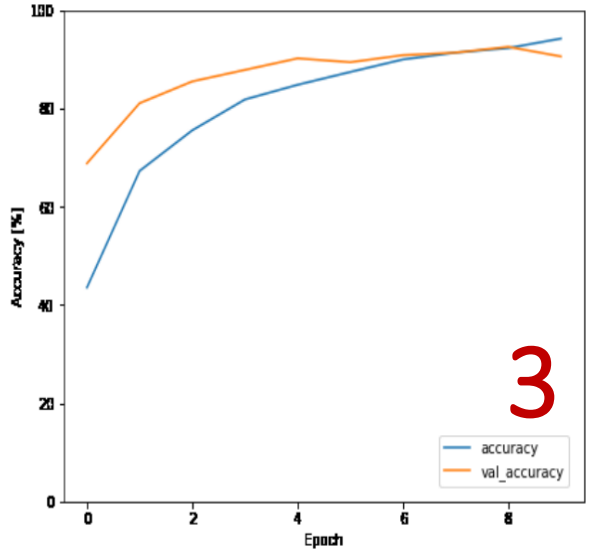
RESULT AND ANALYSIS





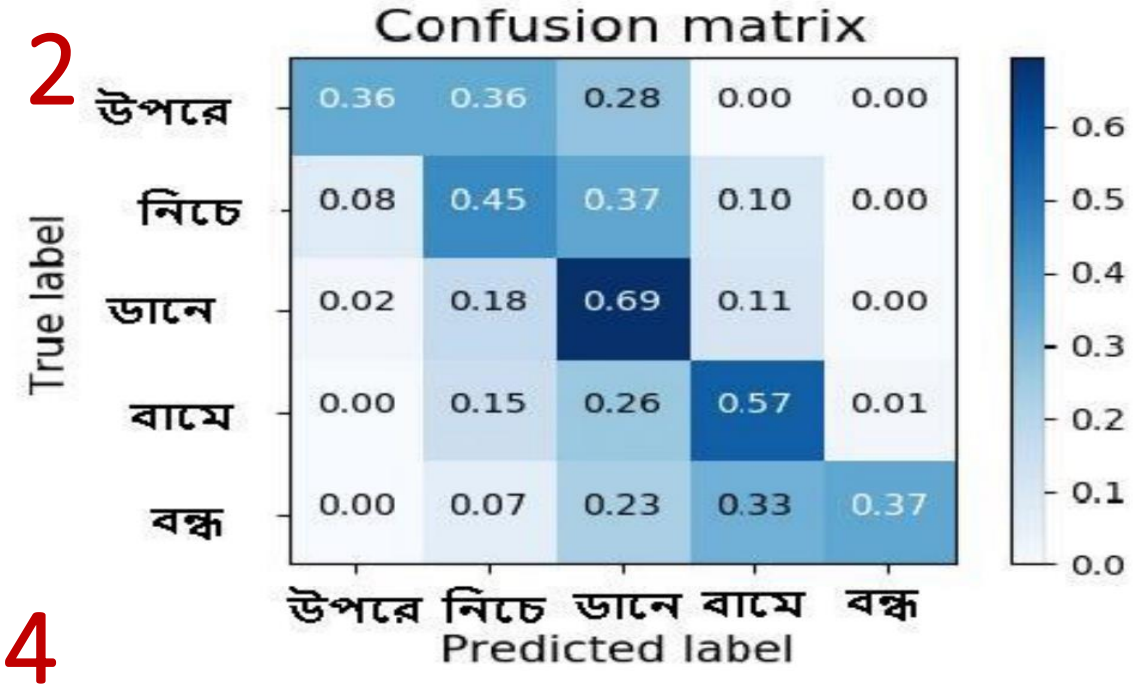
1



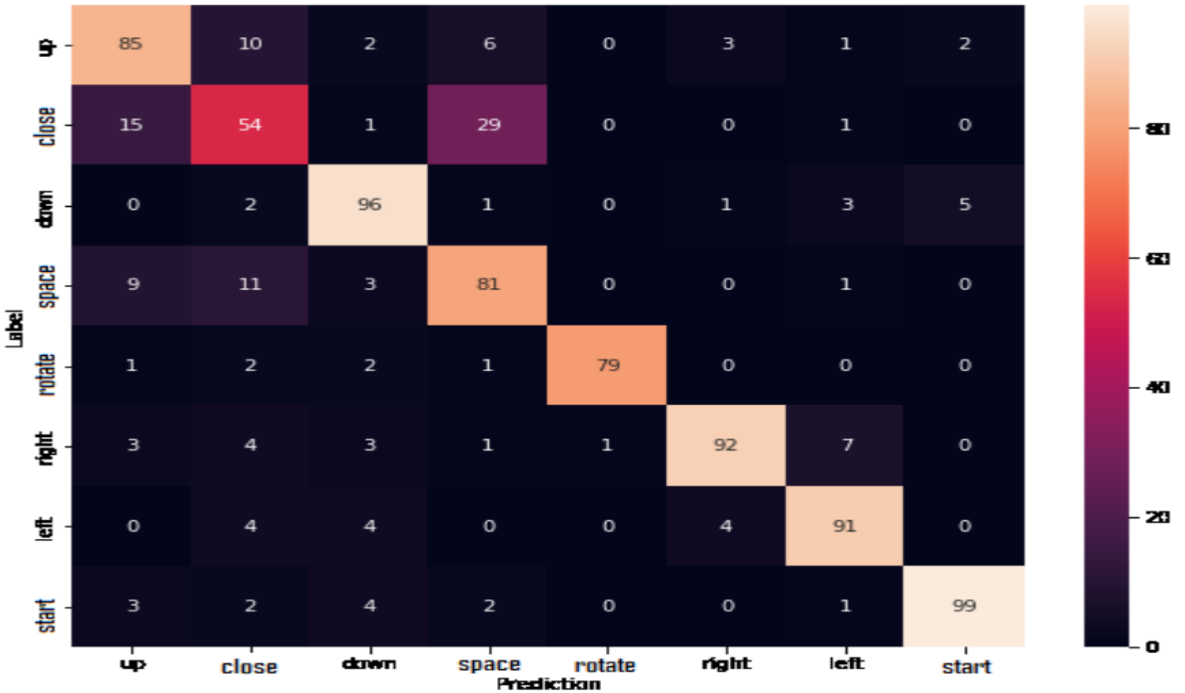


3

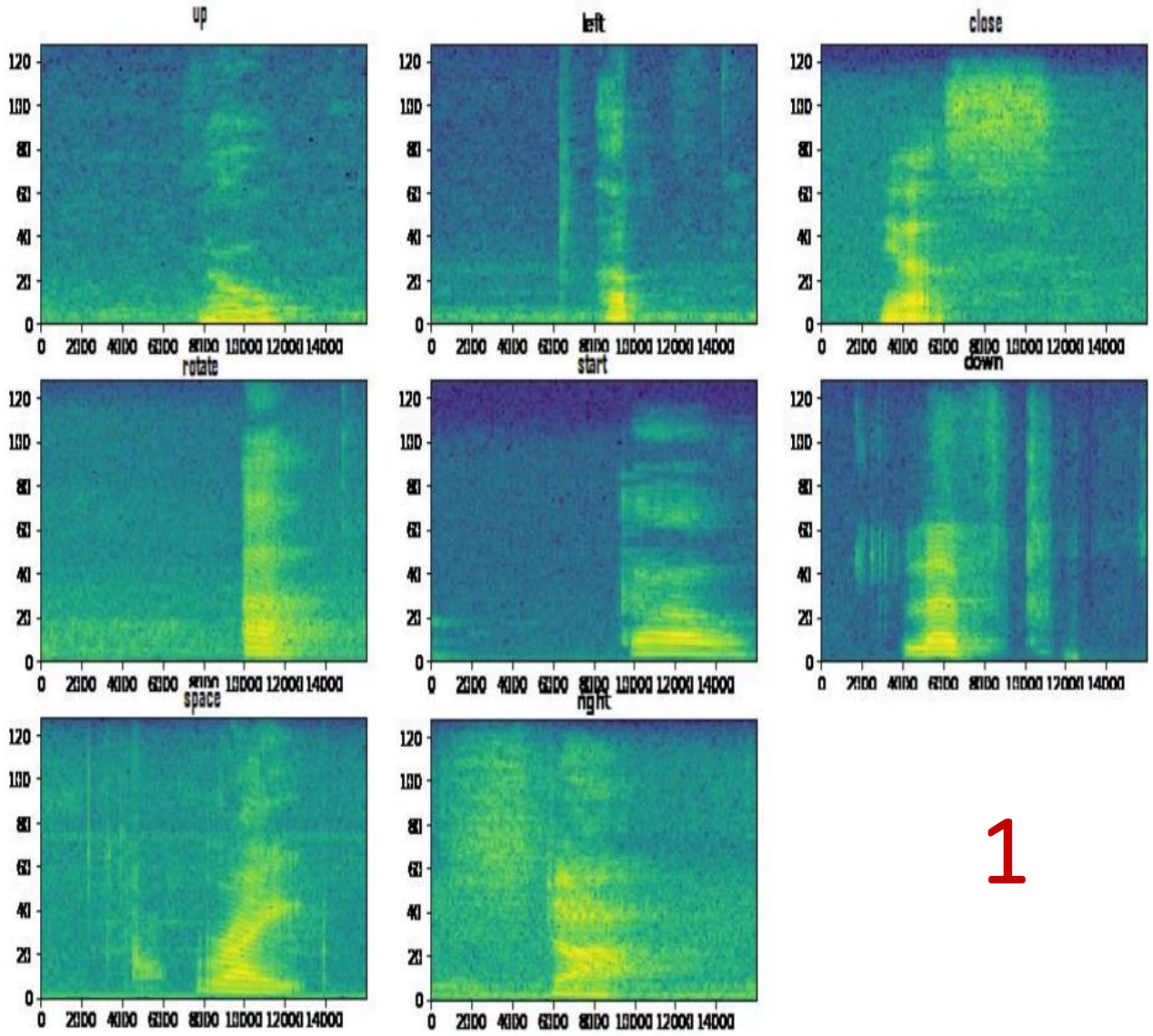
2



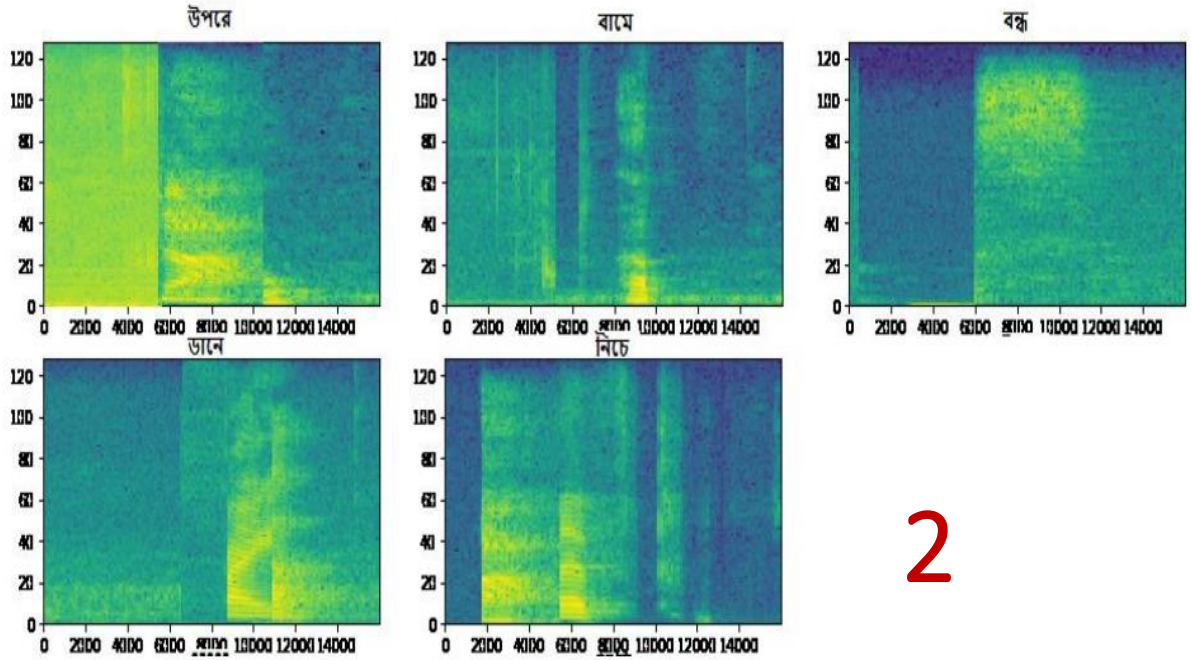
4



Spectrogram Analysis

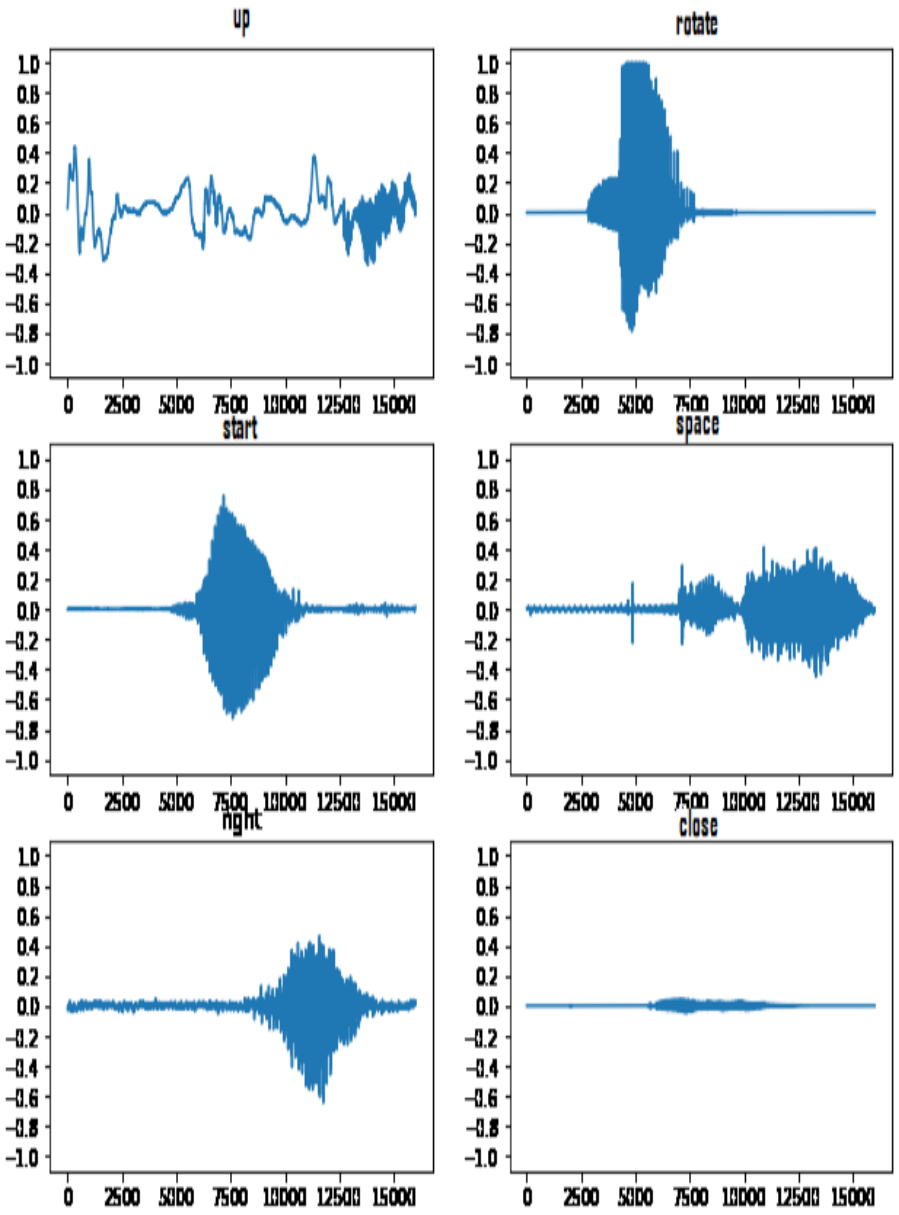


1

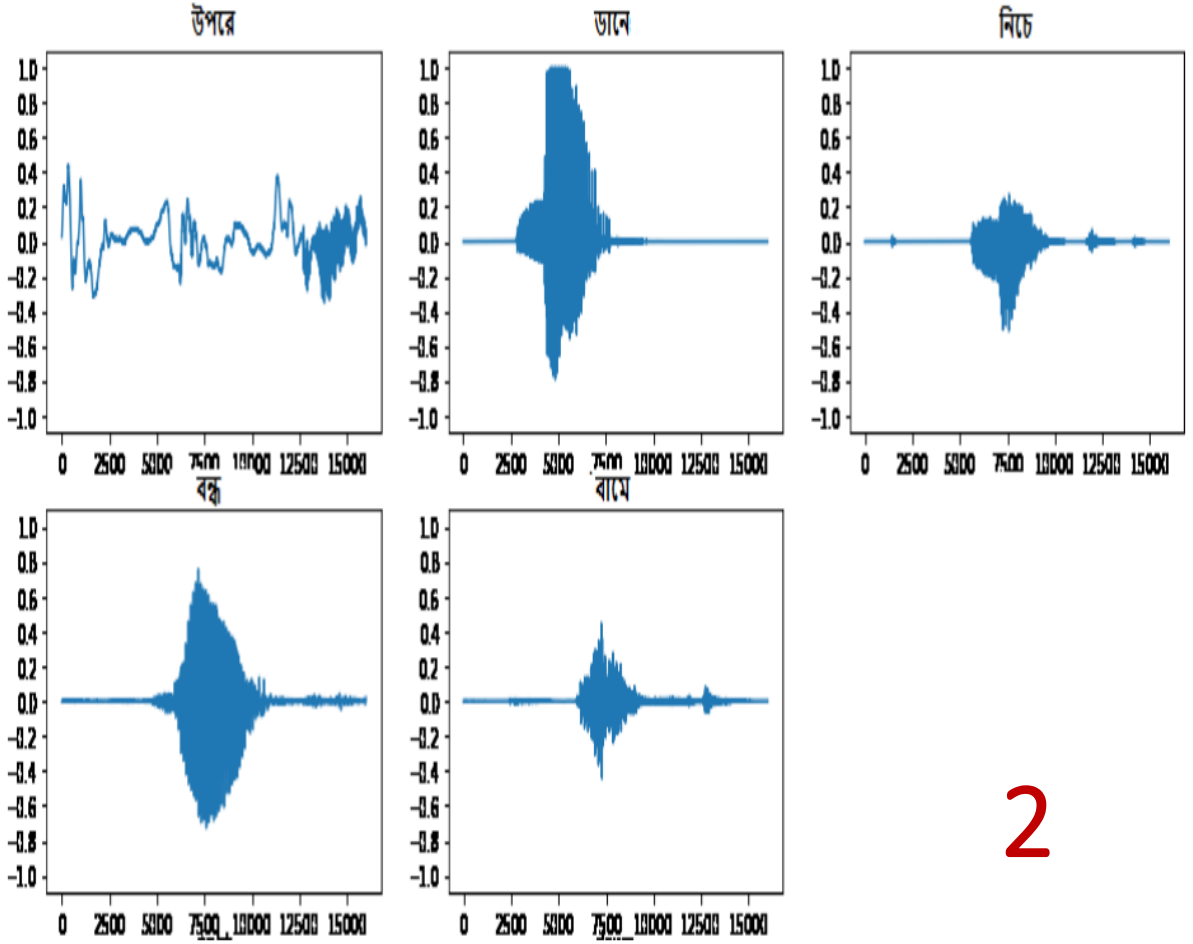


2

Waveform Analysis

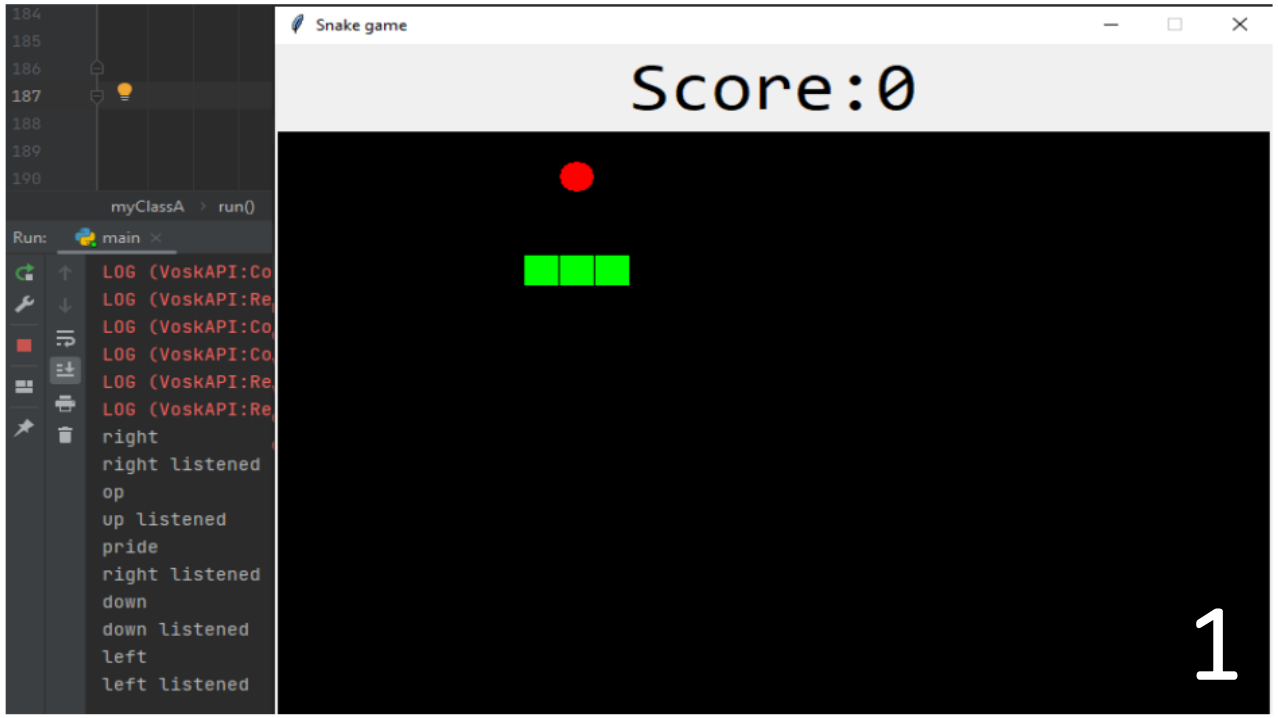


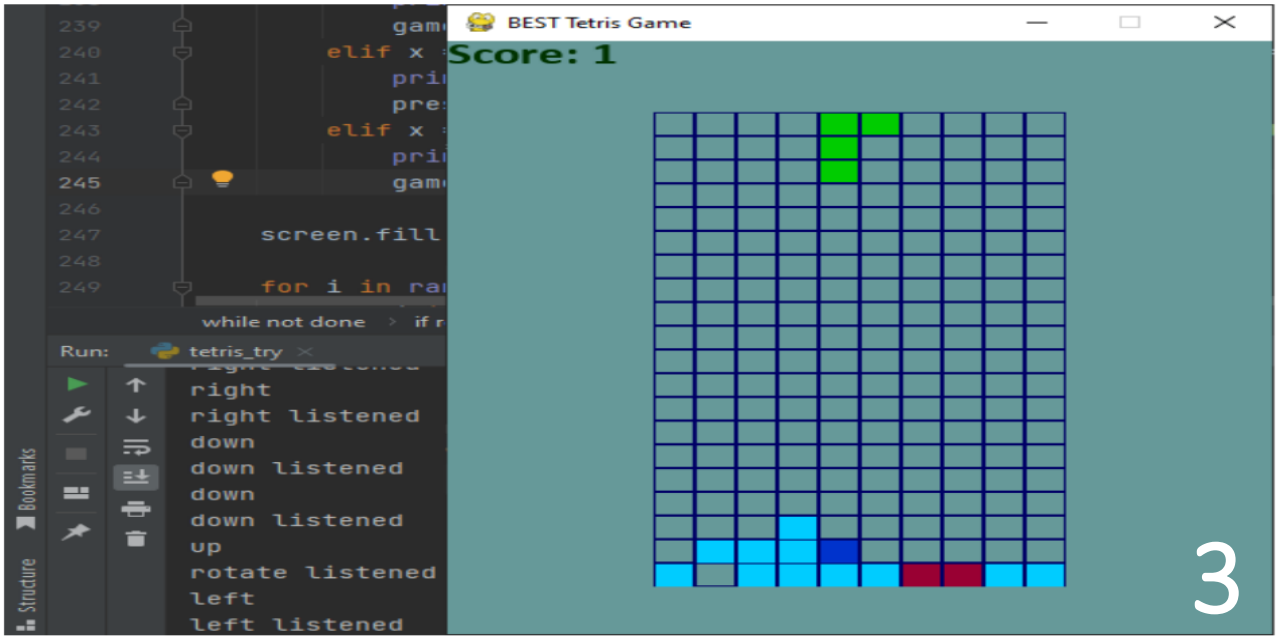
1



2

Game Engine Modules Compatibility Analysis





COMPARISON WITH OTHER METHODS

[1]Microsoft speech SDK

- Only compatible with Windows based operating system
- Environment setup is complex

VOSK model

- Compatible with both windows, linux
- Also compatible with lightweight devices - Raspberry Pi, Android, iOS
- Open source

[2] Sphinx

- Accuracy level is very poor
- Not switchable for use

- Accuracy level is decent enough

[3]Convolutional Neural Networks

- It needs training for adding or replacing new voice commands.

- Adding or replacing new commands is easy

[1] Yuan, X., & Fan, J. (2011, July). (pp. 275-278). IEEE.

[2] https://cmusphinx.github.io/wiki/tutorialbeforestart/

[3] Waqar, D. M., Gunawan, T. S., Kartiwi, M., & Ahmad, R. (2021, August).In 2021 IEEE 7th (ICSIMA) (pp. 76-81). IEEE.

LIMITATIONS

- There is a slight delay

FUTURE WORK

- We will try to integrate this module in 3D games and VR
- Multiplayer Adaptation

THANK YOU FOR YOUR KIND ATTENTION AND PATIENCE