

개별연구

메타버스내 범죄자 회피 및 증거 기록을 위한 방안 연구

▼ 3/13 교수님과의 미팅 내용

개요에서 언급되어 있는 '모욕', '성범죄', '사기'의 경우에는 비디오, 오디오 증거가 있어야 된다고 생각하였습니다.

따라서 몇가지 아이디어를 생각해보았습니다.

1. 블랙박스 feature 구현

- 타인에게도 내가 녹화를 하고 있다는 알림을 보여줌으로서 범죄를 1차적으로 예방

2. STT (Speech To Text)를 사용하여 bad-word 감지 시 자동적으로 블랙박스 녹화

- 증거 데이터는 제3자 서버에 암호화하여 저장되어 법적 요구 시 복호화하여 증거를 조작하는 일에서 자유롭도록 하고 싶음
 - VRChat에서 욕설 감지 (STT 기반) → OBS에 신호 전송
 - OBS가 신호를 받아 자동 녹화 (이전 30초 포함)
 - 녹화된 영상이 암호화된 후 서버에 업로드
 - AES로 암호화 후 AES 키를 RSA 암호화
 - 서버에서는 RSA로 복호화 후 그 키로 AES 복호화 가능
 - 해쉬MAC (기밀성이 문제가 아니라 무결성이 문제)
 - blockchain
 -

3. 익명화 데이터 기반 사용자 행동 평가

- 악성 유저에 대해 익명으로 점수를 주어 (eg.당근마켓의 온도) 신뢰도를 확인할 수 있는 시스템 구축
- VRChat 내부에서 유저에 대한 익명 평가
 - 당근마켓 온도
 - 메이플스토리 인기도처럼?
- VRChat 내부에는 이런 정보를 저장할 수 있는 곳이 없으니 외부 서버가 필요할 듯
- API 만들어서 VRChat 내부에서 사용할 수 있는지가 관건

4. 말투, 어투, 말표현, 억양, 버릇으로 특정 가능하게끔 하는 식별화 (패턴화) 서비스

- vrc는 익명이기 때문에

5. 저작권

- 쉽지는 않을듯

기존 연구 존재 확인,

연구 contribution 무엇 확인

로블록스 < 사기, 갈취, 성폭력 등등

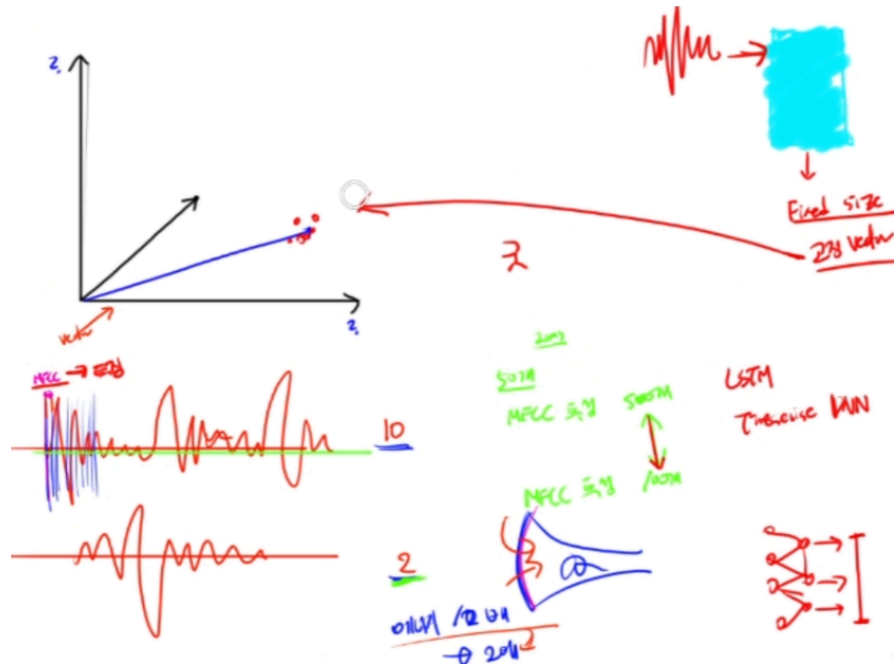
▼ 3/27 준비 내용

▼ 연구 목표 좁히기

- Whisper를 사용하여 메타버스(예: VRChat, Roblox)에서 익명 화자의 음성을 분석하고,
- 특정 패턴을 바탕으로 동일 인물 여부를 식별하여 DB에 저장하고 인식하는 방법 연구.

▼ 논문

- <https://arxiv.org/abs/2005.07143>
 - <https://github.com/TaoRuijie/ECAPA-TDNN>
- Voice Fingerprinting: A Very Important Tool against Crime
 - <https://journals.sagepub.com/doi/pdf/10.1177/0971097320120121>
 - MFCC(Mel-Frequency Cepstral Coefficients), 피치, 음성 템포, 발음 패턴 등의 특징을 추출
 - <https://brightwon.tistory.com/11>
 - <https://sanghyu.tistory.com/45>



- A Novel Model for Voice Command Fingerprinting Using Deep Learning
 - https://www.sciencedirect.com/science/article/pii/S2214212621002659?utm_source=chatgpt.com
- Spying through Your Voice Assistants: Realistic Voice Command Fingerprinting
 - https://www.usenix.org/system/files/usenixsecurity23-ahmed-dilawer.pdf?utm_source=chatgpt.com
- Advancing Audio Fingerprinting Accuracy Addressing Background Noise and Distortion Challenges
 - https://arxiv.org/abs/2402.13957?utm_source=chatgpt.com
- Fingerprinting Smart Devices Through Embedded Acoustic Components
 - https://arxiv.org/abs/1403.3366?utm_source=chatgpt.com

▼ 프로토타이핑

▼ 개요

1. 음성 데이터 수집

- VRChat / Roblox에서 **익명의 사용자가 말하는 음성 샘플** 확보
- 동일한 사람이 다른 계정을 사용하여 말할 경우도 샘플링

- Whisper로 음성을 텍스트 변환(STT) + 음성 특징 분석(voiceprint) 진행

2. 음성 특징 및 텍스트 분석

- Whisper를 이용해 **STT 변환** (영어만)
- 변환된 텍스트를 **NLP 모델**로 분석 (말투, 특정 단어 패턴 탐색)
- 음성 **fingerprint(MFCC, pitch, duration)** 등 추가 분석
- 특징을 **벡터화하여 DB에 저장** (사용자 태깅)

3. 데이터베이스 설계

- 사용자 음성 및 언어 특징을 저장하는 DB 설계
- 익명 사용자에게도 **해시 기반의 고유 태그** 할당 (예: `user_abc123`)
- 새로운 사용자가 말할 때 **기존 DB와 유사도 비교** → **동일 인물 가능성 평가**

4. 인식 시스템 구현

- 사용자가 말할 때 **Whisper + NLP 분석** → **기존 데이터와 비교**
- **유사도 점수**를 바탕으로 특정 개인일 가능성을 예측
- 80% 이상 유사하면 **동일 인물 태깅** 수행

▼ python 코드

```
import torchaudio
import whisper
import librosa
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics.pairwise import cosine_similarity
from python_speech_features import mfcc

# 1 음성 데이터 로드 (LibriSpeech 샘플)
def load_audio(file_path):
    waveform, sample_rate = librosa.load(file_path, sr=16000)
    return waveform, sample_rate

# 2 Whisper를 사용한 STT 변환
```

```

def transcribe_audio(file_path):
    model = whisper.load_model("base")
    result = model.transcribe(file_path)
    return result['text']

# 3 MFCC 기반 음성 fingerprinting
def extract_mfcc_features(audio, sample_rate):
    mfcc_features = mfcc(audio, samplerate=sample_rate, numcep=13)
    return mfcc_features

# 4 유사도 비교 (화자 식별 가능성 확인)
def compare_speakers(mfcc1, mfcc2):
    sim = cosine_similarity(mfcc1.mean(axis=0).reshape(1, -1), mfcc2.mean(
    return sim[0][0]

# 테스트 실행
audio1, sr1 = load_audio("sample1.wav")
audio2, sr2 = load_audio("sample2.wav")


text1 = transcribe_audio("sample1.wav")
text2 = transcribe_audio("sample2.wav")
print("STT 결과:", text1, "/", text2)

mfcc1 = extract_mfcc_features(audio1, sr1)
mfcc2 = extract_mfcc_features(audio2, sr2)

similarity = compare_speakers(mfcc1, mfcc2)
print("음성 유사도:", similarity)

```

1단계: 데이터셋 선택

- **LibriSpeech**: 깨끗한 음성 데이터, 문장별로 정제된 영어 데이터셋
- **Common Voice**: 다양한 억양과 잡음 포함, 사용자 기반 클라우드소싱 데이터셋
 LibriSpeech를 먼저 사용 → 필요하면 Common Voice로 확장

✓ 2단계: Whisper 기반 STT 변환

- OpenAI Whisper 모델을 사용하여 음성을 텍스트로 변환
- 음성 샘플을 다양한 크기로 자르고 정확도 확인

✅ 3단계: NLP를 활용한 언어 패턴 분석

- spaCy, NLTK, Hugging Face 등을 사용하여 특정한 말투, 단어 사용 패턴 분석
- 예시: 화자의 특정한 단어 사용 빈도, 문장 길이, 감정 분석 등

✅ 4단계: 음성 Fingerprinting 분석

- MFCC(Mel-Frequency Cepstral Coefficients) 및 기타 음향 특징 추출
- 같은 화자의 음성이 유사한지 비교 (DTW, 코사인 유사도 등 적용)