融保工

딥페이크 오디오 감지 프로그램 만들기

-기획부-

목차

- 활동 일정

딥페이크 오디오 감지: 이론부터 실전까지

- 개요

❒ 학습 목표:

- 딥페이크 오디오 기술의 원리와 영향 이해
- Hugging Face 플랫폼과 Transformers 라이브러리 활용법 습득
- 실제 딥페이크 오디오 감지 모델 개발 및 구현

❒ 오늘 배워야 할 것:

- 모델 검색해서 활용하는 법

* 모든 의문과 오류는 GPT와 함께 물어보며 진행합니다

딥페이크 오디오의 정의와 중요성

- 딥페이크 오디오 정의:
  - "인공지능 기술을 사용하여 생성 또는 조작된 음성 콘텐츠"
- 주요 기술:
  - 음성 합성 (Text-to-Speech)
  - 음성 변환 (Voice Conversion)
  - 음성 클로닝 (Voice Cloning)
- 응용 분야:
  - 엔터테인먼트: 영화 더빙, 게임 음성
  - 접근성: 개인화된 음성 어시스턴트
  - 언어 학습: 발음 교정, 외국어 연습

딥페이크 오디오의 정의와 중요성

- 딥페이크 오디오 정의:
  - "인공지능 기술을 사용하여 생성 또는 조작된 음성 콘텐츠"
- 주요 기술:
  - 음성 합성 (Text-to-Speech)
  - 음성 변환 (Voice Conversion)
  - 음성 클로닝 (Voice Cloning)
- 응용 분야:
  - 엔터테인먼트: 영화 더빙, 게임 음성
  - 접근성: 개인화된 음성 어시스턴트
  - 언어 학습: 발음 교정, 외국어 연습
- 잠재적 위험:
  - 사기와 신원 도용
  - 허위 정보 유포
  - 개인 프라이버시 침해
- 중요성:
  - 기술 발전에 따른 오디오 조작의 정교화
  - 진실성 검증의 어려움 증가
  - 법적, 윤리적 문제 대두
  - 감지 기술의 필요성 증대

딥페이크 활용한 사이버 사기 공격, 한 기업으로부터 3500만 달러 빼앗아

입력 : 2021-10-21 16:42

3500만 달러가 한 UAE 기업에서 사라졌다. 공격자들이 한 임원의 목소리로 직원에게 전화를 걸어 송금을 지시했기 때문이다. 임원의 목소리를 흉내 낸 건 다름 아니라 딥페이크라는 인공지능 기술. 딥페이크가 BEC 공격에 활용된 사례가 벌써 두 건이나 누적됐다.

[보안뉴스 문가용 기자] 한 사기단이 딥페이크 오디오를 사용해 UAE의 한 기업으로부터 3500만 달러를 가져가는 데 성공했다. 이들은 회사의 임원으로 가장하여 회사 인수에 필요한 돈을 송금하라고 직원들을 속였고, 가짜 목소리에 직원들은 깜빡 속았다고 한다.

딥페이크 오디오 기술의 핵심 구성 요소

- **음성 합성 (Text-to-Speech, TTS)**
  - 정의: 텍스트를 자연스러운 음성으로 변환하는 기술
- 주요 기술:
  - WaveNet (DeepMind): 고품질 음성 생성을 위한 딥러닝 모델
  - Tacotron 2 (Google): 자연스러운 억양과 강세를 생성하는 end-to-end 모델
- 최신 동향:
  - FastSpeech 2 (Microsoft): 병렬 처리로 빠른 음성 생성
  - VITS (Kakao Enterprise): 텍스트에서 직접 파형을 생성하는 end-to-end 모델

- **음성 변환 (Voice Conversion)**
  - 정의: 한 화자의 음성을 다른 화자의 음성으로 변환하는 기술
- 주요 기술:
  - CycleGAN-VC: 비병렬 데이터를 사용한 음성 변환
  - StarGAN-VC: 다중 화자 간 음성 변환
- 최신 동향:
  - AGAIN-VC: 적대적 생성 네트워크를 이용한 고품질 음성 변환

- **음성 클로닝 (Voice Cloning)**
  - 정의: 적은 양의 음성 샘플로 특정 화자의 음성을 복제하는 기술
- 주요 기술:
  - SV2TTS (Transfer Learning): 화자 인증, 텍스트 인코딩, 스펙트로그램 생성의 3단계 과정
  - AutoVC: 자기지도 학습을 통한 화자 독립적 콘텐츠 인코딩
- 최신 동향:
  - VALL-E (Microsoft): 3초 음성 샘플로 고품질 음성 클로닝
  - YourTTS (Coqui AI): 다국어 지원 및 감정 제어가 가능한 음성 클로닝
- 적은 데이터로 가능한 음성 복제 기술

Hugging Face란?

- 정의: "자연어 처리(NLP) 분야의 오픈소스 기술을 선도하는 AI 커뮤니티 및 플랫폼"
  - 협업 중심: 전 세계 개발자, 연구자들의 지식 공유
  - 모델 허브: 150,000개 이상의 사전 훈련된 모델 제공
  - 데이터셋: 20,000개 이상의 공개 데이터셋
  - 문서화: 상세한 튜토리얼과 API 문서 제공

- 대표 제품: Transformers 라이브러리
  - NLP, 컴퓨터 비전, 음성 처리 등 다양한 AI 작업 지원
  - 영향력:
  - GitHub 스타 수: 70,000+ (Transformers 라이브러리)
  - 월간 다운로드: 1,000만 회 이상

Hugging Face에서 제공하는 서비스와 리소스

- 모델 허브 (Model Hub):
  - 150,000+ 사전 훈련된 모델
  - 다양한 작업 (NLP, 비전, 오디오 등) 지원
  - 커스텀 모델 업로드 및 공유 가능
- 데이터셋 (Datasets):
  - 20,000+ 공개 데이터셋
  - 효율적인 데이터 로딩 및 전처리 기능
- Spaces
  - 데모 애플리케이션 호스팅 플랫폼
  - 모델의 실시간 시연 및 공유 가능

Hugging Face에서 제공하는 서비스와 리소스

- AutoTrain:
  - 코드 없이 모델 훈련 자동화
  - 사용자 정의 데이터로 모델 fine-tuning
- Inference API:
  - 클라우드 기반 모델 추론 서비스
  - 확장 가능한 배포 솔루션
- 문서 및 튜토리얼:
  - 상세한 API 문서
  - 단계별 가이드 및 예제 코드
- 커뮤니티 포럼:
  - 기술 논의 및 문제 해결 플랫폼

모델 및 데이터셋 탐색하기

- huggingface.co/models
- 모델 허브 탐색
  - 필터 옵션:
  - • 태스크 (예: 음성 인식, 텍스트 분류)
  - • 언어
  - • 라이브러리 (예: Transformers, TensorFlow)
  - 정렬 옵션: 다운로드 수, 좋아요 수 등
- 모델 페이지 구성:
  - 모델 카드: 설명, 사용법, 성능 지표
  - 파일 브라우저: 모델 가중치, 설정 파일
  - 커뮤니티 기여: 댓글, 이슈 트래커
- 데이터셋 허브:
  - URL: huggingface.co/datasets
  - 필터 옵션: 태스크, 언어, 라이선스 등
  - 데이터셋 카드: 설명, 통계, 사용 예시
- 실습: 오디오 관련 모델 찾기
  - 검색어: "audio classification"
  - 인기 모델 예시: facebook/wav2vec2-base
  - 모델 사용법 확인 및 코드 스니펫 분석

- 데이터셋 사용 예시:
  - 코드

```python
from datasets import load_dataset
dataset = load_dataset("common_voice", "en")
print(dataset["train"][0])
```

Hugging Face 모델 사용 개요

- Hugging Face 모델: 쉽고 빠른 AI 모델 사용

- Hugging Face: AI 모델의 GitHub라고 생각하면 쉽습니다
  - 다양한 사전 훈련된 모델 제공 (텍스트, 이미지, 오디오 등)
  - 간단한 코드로 고성능 모델 사용 가능
- Hugging Face 모델 사용하기: 3단계 과정
  - 필요한 라이브러리 설치
  - 모델과 토크나이저 불러오기
  - 모델 사용하여 작업 수행

Hugging Face 모델 사용하기

- 라이브러리 설치

❒ 1단계: 필요한 라이브러리 설치하기

□ pip를 사용하여 transformers 라이브러리 설치

```
pip install transformers
```

```
pip install torch  # PyTorch 설치
```

❒ 2단계: 모델과 토크나이저 불러오기

```
from transformers import AutoTokenizer, AutoModel

# 모델 이름은 Hugging Face 모델 허브에서 확인
model_name = "bert-base-uncased"  # 예시 모델

# 토크나이저와 모델 불러오기
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)
```

라이브러리 설명

- 프로젝트에 사용되는 주요 라이브러리

- PyQt5
- librosa
- torch
- transformers

라이브러리 설명

- PyQt5 소개

- PyQt5: Python용 GUI 프레임워크
  - 정의: Qt 프레임워크의 Python 바인딩 • 주요 특징: 크로스 플랫폼 지원 (Windows, macOS, Linux)
  - 풍부한 UI 컴포넌트 제공
  - 시그널-슬롯 메커니즘으로 이벤트 처리
- 프로젝트에서의 역할: 사용자 인터페이스 구현
- PyQt 문법

```python
from PyQt5.QtWidgets import QApplication, QMainWindow,

app = QApplication([])
window = QMainWindow()
button = QPushButton("Click me!", window)
window.show()
app.exec_()
```

라이브러리 설명

- librosa 소개

- librosa: 음악 및 오디오 분석 라이브러리
- 정의: 음악 및 오디오 분석을 위한 Python 패키지
- 주요 기능:
  - 오디오 파일 로딩 및 재생
  - 스펙트로그램 생성
  - 음높이 및 박자 추출
  - 음색 분석
- 프로젝트에서의 역할: 오디오 파일 처리 및 특성 추출

```python
import librosa

# 오디오 파일 로드
y, sr = librosa.load('audio_file.mp3')

# 스펙트로그램 생성
spectrogram = librosa.stft(y)
```

라이브러리 설명

- torch (PyTorch) 소개

- PyTorch: 유연한 딥러닝 프레임워크
- 정의: Facebook AI Research Lab에서 개발한 오픈소스 머신러닝 라이브러리
- 주요 기능:
  - 동적 계산 그래프
  - GPU 가속 지원
  - 풍부한 생태계와 도구
- 프로젝트에서의 역할: 딥러닝 모델 운용 및 계산 수행

```python
import torch

# 텐서 생성
x = torch.tensor([1, 2, 3])

# GPU 사용 (가능한 경우)
device = torch.device("cuda" if torch.cuda.is_available()
x = x.to(device)
```

라이브러리 설명

- transformers 소개

- Hugging Face Transformers: NLP 모델의 Swiss Army Knife
- 정의: 최신 자연어 처리 모델을 쉽게 사용할 수 있게 해주는 라이브러리
- 주요 기능:
  - 다양한 사전 훈련 모델 지원(BART, GPT)
  - 간편한 모델 로딩 및 사용
  - 파인튜닝 기능 지원
- 프로젝트에서의 역할: 오디오 분류 모델 로딩 및 추론

```
from transformers import AutoModelForAudioClassification, AutoFeatureExtractor


model = AutoModelForAudioClassification.from_pretrained("facebook/wav2vec2-base")
feature_extractor = AutoFeatureExtractor.from_pretrained("facebook/wav2vec2-base")
```

Hugging Face 모델 사용하기

- 라이브러리 설치

❒ 3단계: 모델로 작업 수행하기

- 텍스트 분류 작업

```
# 입력 텍스트 토큰화
inputs = tokenizer("이것은 예시 문장입니다.", return_tensors="pt")

# 모델로 추론
outputs = model(**inputs)

# 결과 처리 (예: 분류 작업의 경우)
import torch
predictions = torch.nn.functional.softmax(outputs.logits, dim=-1)
print(predictions)
```

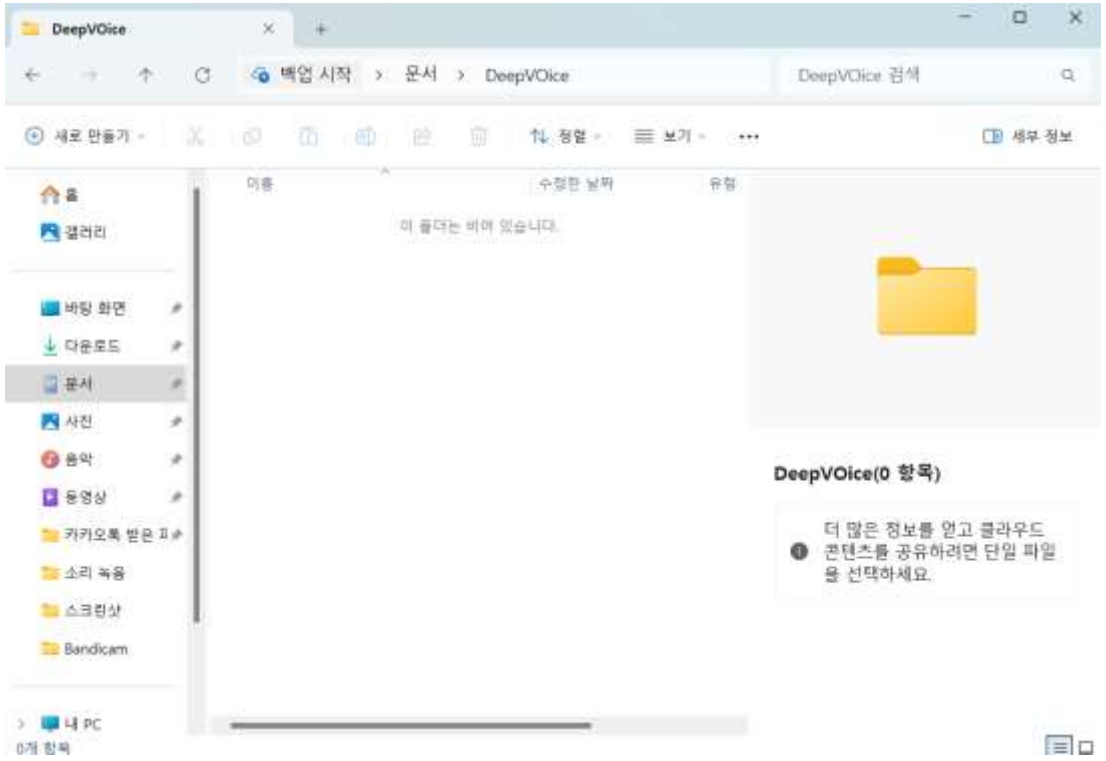
❒ 실제 사용 예시: 감성 분석하기

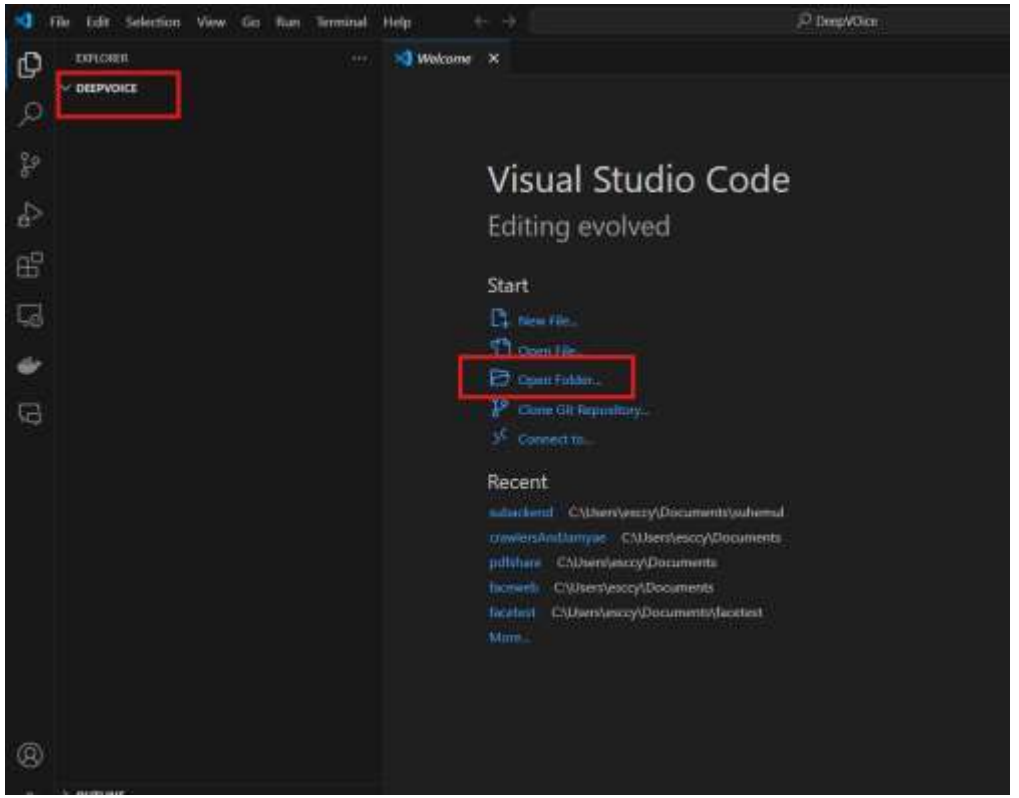
```
from transformers import pipeline

# 감성 분석을 위한 파이프라인 생성
sentiment_analyzer = pipeline("sentiment-analysis")

# 텍스트 분석
text = "I love using Hugging Face models!"
result = sentiment_analyzer(text)

print(result)  # [{'label': 'POSITIVE', 'score': 0.9998}]
```

프로젝트: 딥페이크 오디오 감지기 만들기

- 폴더 만들기

❒ 내 문서(혹은 다른위치)에 DeepVoice라는 이름의 폴더를 만듭니다. VSC로 열어주세요.





프로젝트: 딥페이크 오디오 감지기 만들기

- requirements.txt 쓰기

- 이번 프로젝트에는 PyQt5, librosa, torch, transformers 라이브러리가 필요
- pip로 다운받는 게 일반적이지만, 나중에 기억해뒀다가 다시 받을 수 있도록 requirements.txt를 작성
- 예시 requirements.txt:

```
PyQt5==5.15.6
librosa==0.9.1
torch==1.10.0
transformers==4.15.0
```

- 한 번에 라이브러리 설치하는 명령어:
  - pip install -r requirements.txt

프로젝트: 딥페이크 오디오 감지기 만들기

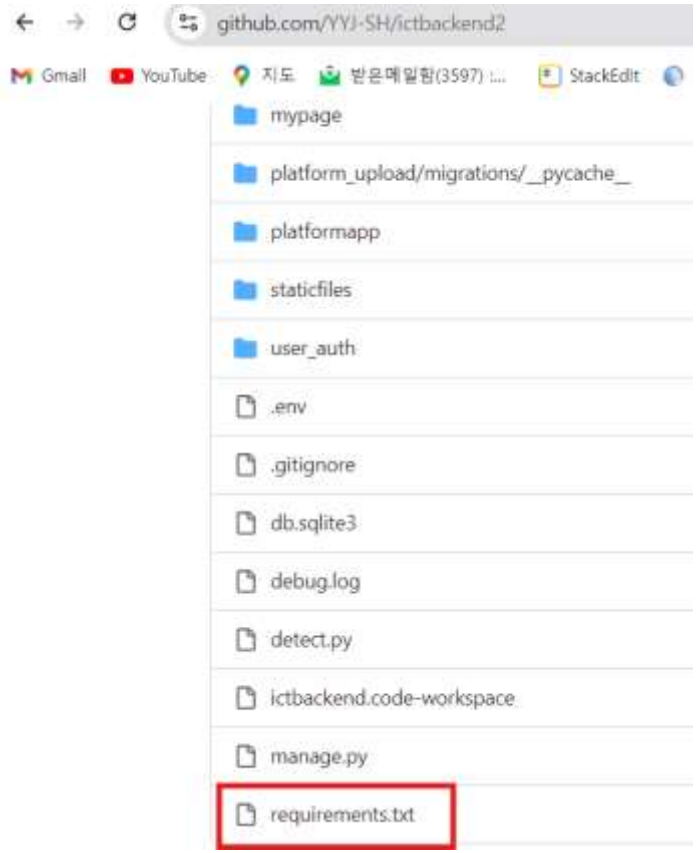
requirements.txt: 프로젝트 의존성 관리

1. requirements.txt란?
- Python 프로젝트의 외부 라이브러리 목록
- 라이브러리 이름과 버전 정보 포함

2. 작성 방법
- 수동: 직접 라이브러리와 버전 입력
  예) numpy==1.21.0
  pandas>=1.3.0
- 자동: pip freeze > requirements.txt

3. 사용 방법
- 패키지 설치: pip install -r requirements.txt

4. 장점
- 프로젝트 의존성 일괄 관리
- 개발 환경 재현 용이
- 협업 시 환경 통일 가능



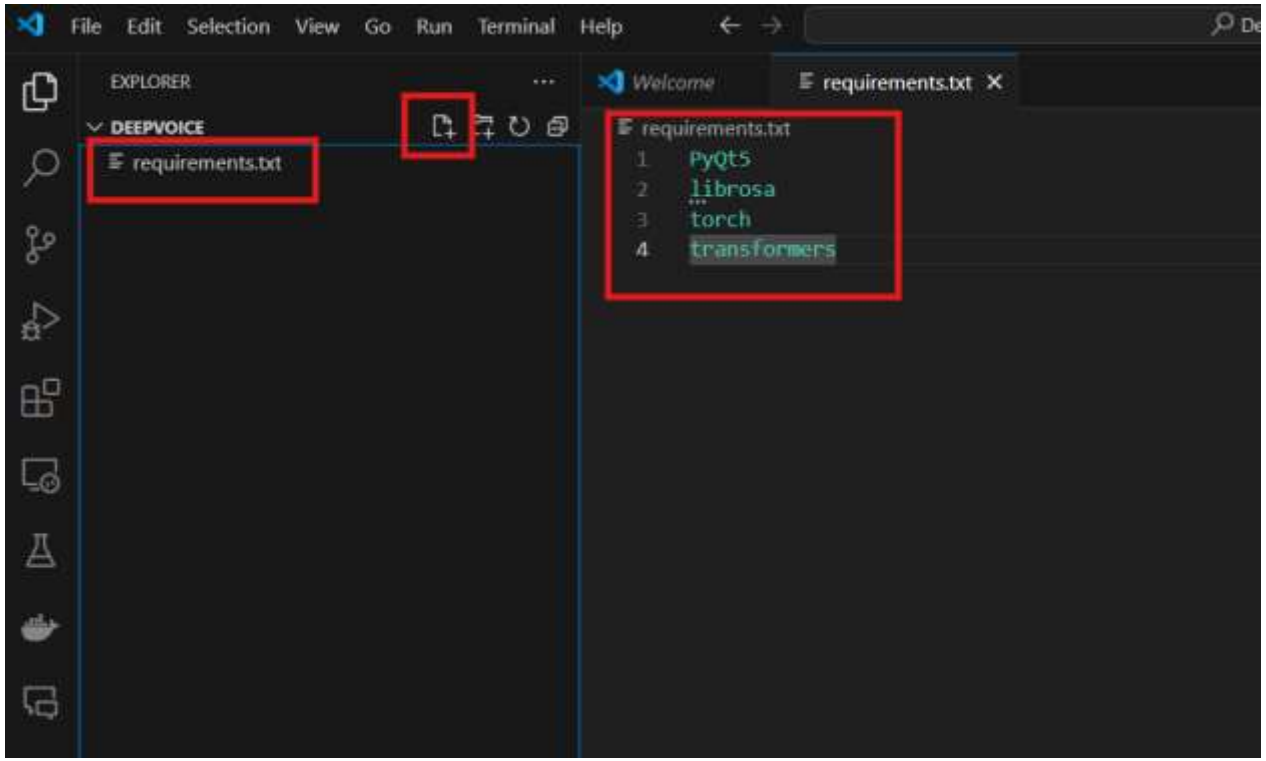
프로젝트: 딥페이크 오디오 감지기 만들기

- requirements.txt 작성

❒ new file(새 파일) > requirements.txt

❒ 하단내용 작성:

```
PyQt5
librosa
torch
transformers
```



프로젝트: 딥페이크 오디오 감지기 만들기

- venv로 가상환경 설정하기

1. venv란?
   - Python 3.3+ 버전에 기본 내장된 가상 환경 생성 도구
   - Python 프로젝트별 독립 환경 제공 도구
   - 시스템 Python과 분리된 환경 생성

2. venv가 필요한 이유
   - 프로젝트별 패키지 버전 관리 용이
   - 시스템 Python 환경 보호
   - 프로젝트 의존성 정확한 재현 가능

3. venv 설정 과정
   a. 가상 환경 생성:
      python -m venv myenv

   b. 가상 환경 활성화:
      - Windows: myenv\Scripts\activate
      - macOS/Linux: source myenv/bin/activate

   c. 패키지 설치:
      pip install -r requirements.txt

4. 가상 환경 사용 팁
   - 프로젝트마다 새로운 가상 환경 생성
   - 활성화 상태 확인 (프롬프트 변경)
   - 작업 완료 후 비활성화: deactivate

프로젝트: 딥페이크 오디오 감지기 만들기

- venv로 가상환경 설정하기

1. venv란?
   - Python 3.3+ 버전에 기본 내장된 가상 환경 생성 도구
   - Python 프로젝트별 독립 환경 제공 도구
   - 시스템 Python과 분리된 환경 생성

2. venv가 필요한 이유
   - 프로젝트별 패키지 버전 관리 용이
   - 시스템 Python 환경 보호
   - 프로젝트 의존성 정확한 재현 가능

3. venv 설정 과정
   a. 가상 환경 생성:
      python -m venv myenv

   b. 가상 환경 활성화:
      - Windows: myenv\Scripts\activate
      - macOS/Linux: source myenv/bin/activate

   c. 패키지 설치:
      pip install -r requirements.txt

4. 가상 환경 사용 팁
   - 프로젝트마다 새로운 가상 환경 생성
   - 활성화 상태 확인 (프롬프트 변경)
   - 작업 완료 후 비활성화: deactivate

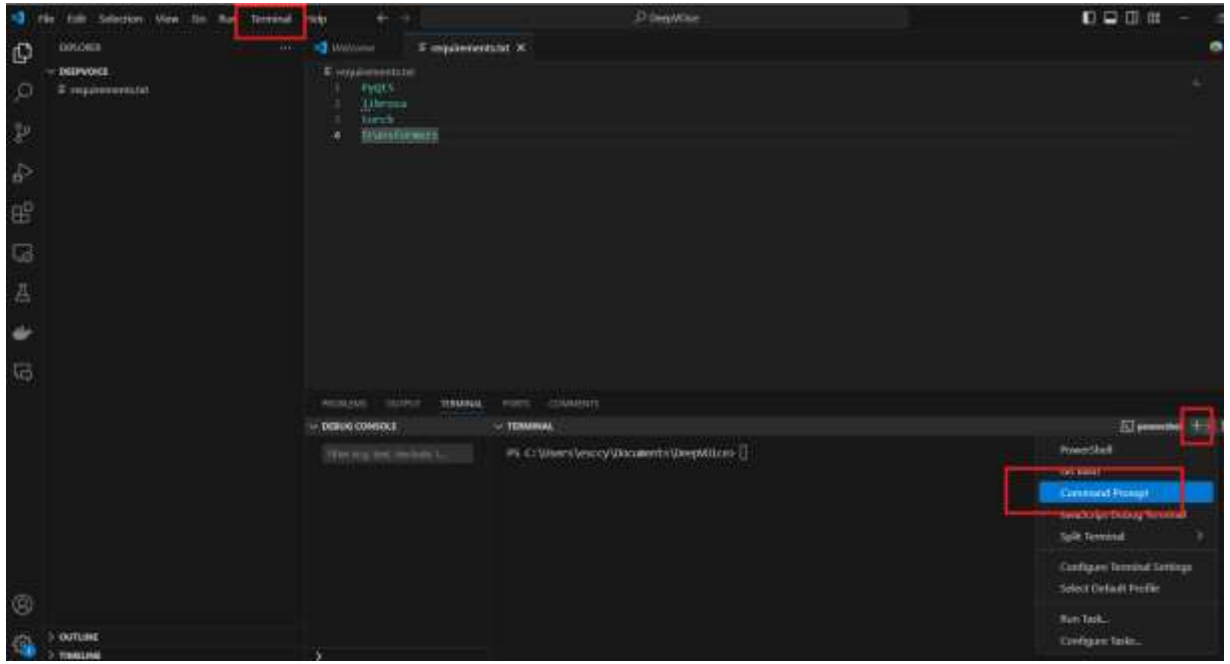
프로젝트: 딥페이크 오디오 감지기 만들기

❒ 터미널> 새 터미널 > Comand Prompt

python -m venv myenv

myenv\Scripts\activate

pip install -r requirements.txt

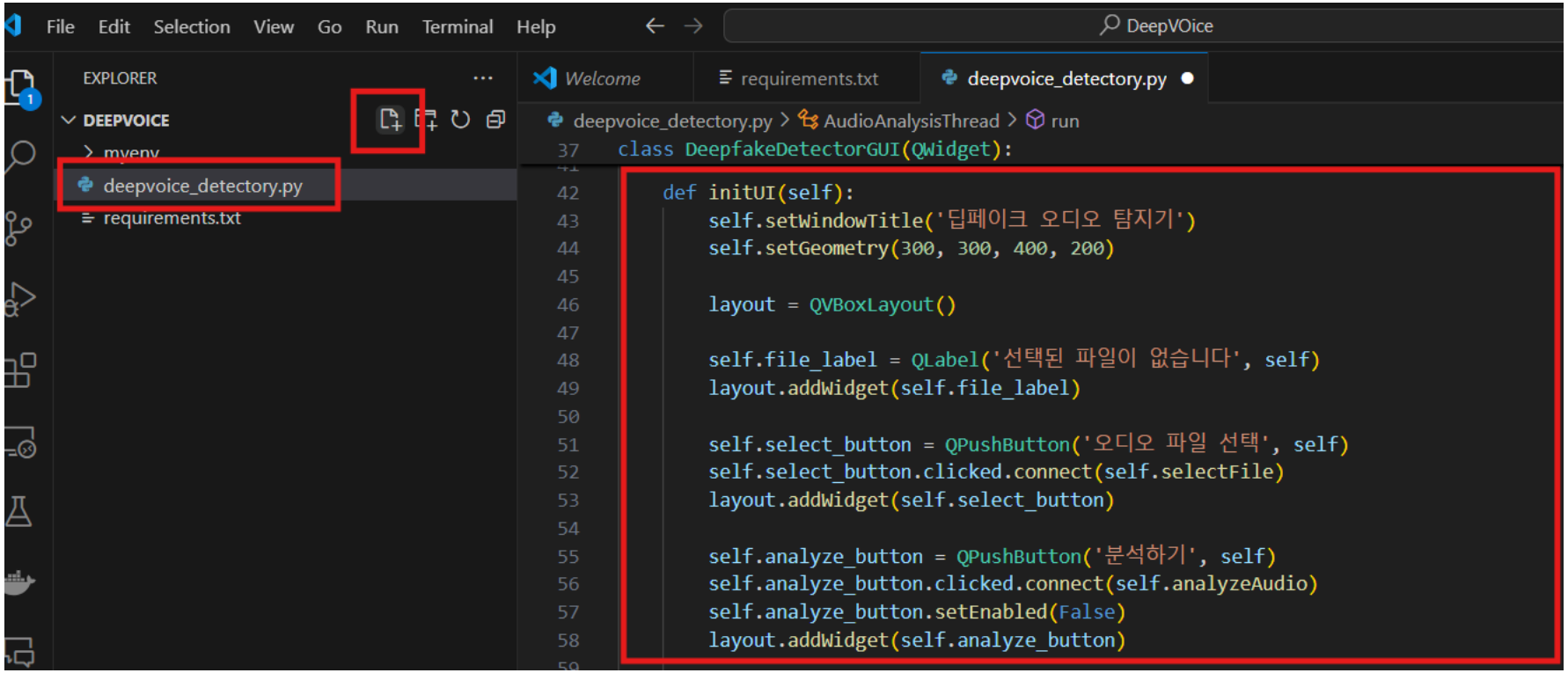


```
TERMINAL
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

(base) C:\Users\esccy\Documents\DeepVOice>python -m venv myenv

(base) C:\Users\esccy\Documents\DeepVOice>myenv\Scripts\activate
```

```
(myenv) (base) C:\Users\esccy\Documents\DeepVOice>pip install -r requirements.txt
```

프로젝트: 딥페이크 오디오 감지기 만들기

❒ new file > deepvoice_detector.py > 코드 붙여넣기 (코드 다음페이지에)



코드

- Ctrl+A로 복사하고 Ctrl+C, Ctrl+v

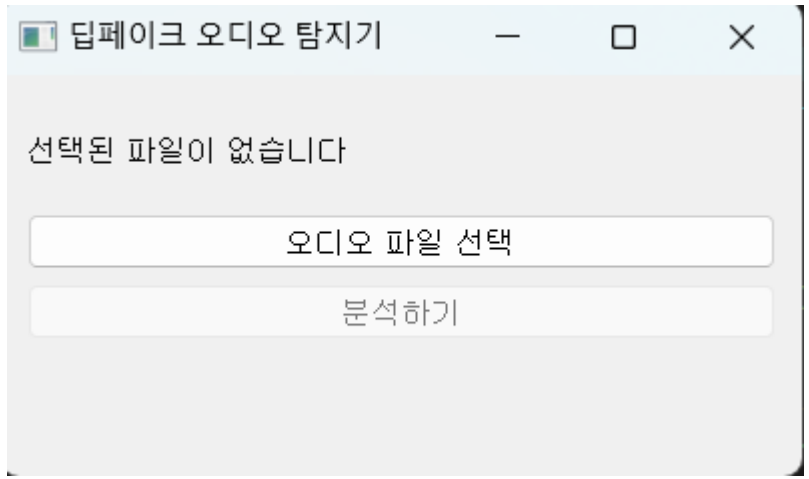
```
import sys
import os
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QVBoxLayout, QLabel, QFileDialog,
QProgressBar
from PyQt5.QtCore import Qt, QThread, pyqtSignal
import librosa  # 오디오 파일을 처리하는 라이브러리
import torch  # 인공지능 모델을 실행하는 데 사용하는 라이브러리
from transformers import AutoFeatureExtractor, AutoModelForAudioClassification  # 사전 훈련된 AI 모델을 로드하
기 위한 모듈

# 오디오 파일 분석을 위한 별도의 스레드 클래스
class AudioAnalysisThread(QThread):
    # 결과가 나왔을 때 이를 GUI로 보내기 위한 신호 (prediction, confidence를 전달함)
    result_signal = pyqtSignal(str, float)
    # 에러가 발생했을 때 이를 GUI로 보내기 위한 신호
    error_signal = pyqtSignal(str)

    # 초기화 메서드, 분석할 파일 경로를 받아온다.
    def __init__(self, file_path):
        super().__init__() # 부모 클래스의 초기화 메서드를 호출
```

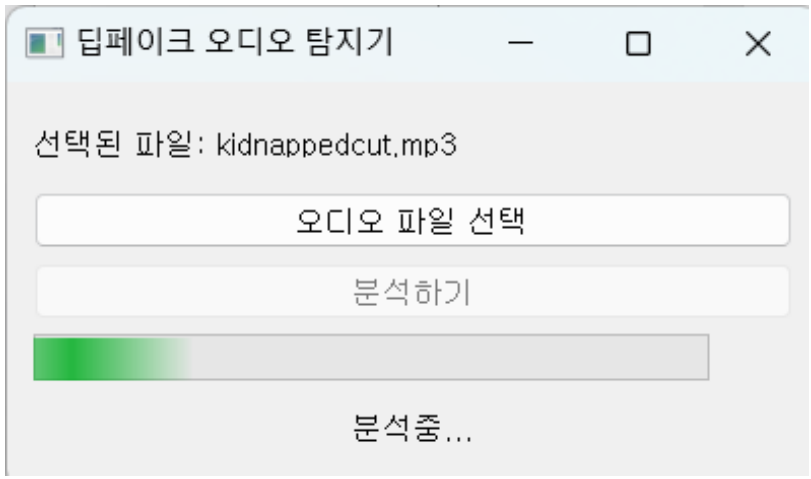
코드 실행

- 터미널에서 명령어 입력

python deepvoice_detector.py
(python (내 파이썬 파일 이름))

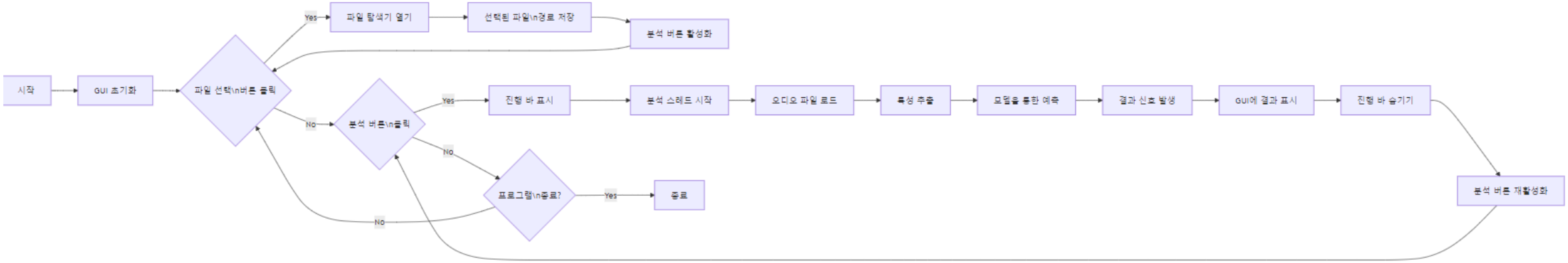
```
(myenv) (base) C:\Users\esccy\Documents\DeepVOice>python deepvoice_detectory.py
```





코드 설명

- 흐름도


시작
GUI 초기화
파일 선택\n버튼 클릭
Yes
파일 탐색기 열기
선택된 파일\n경로 저장
분석 버튼 활성화
No
분석 버튼\n클릭
Yes
진행 바 표시
분석 스레드 시작
오디오 파일 로드
특성 추출
모델을 통한 예측
결과 신호 발생
GUI에 결과 표시
진행 바 숨기기
분석 버튼 재활성화
No
프로그램\n종료?
Yes
종료
No

실습

- 그런데 deep voice를 어디서 구하지?

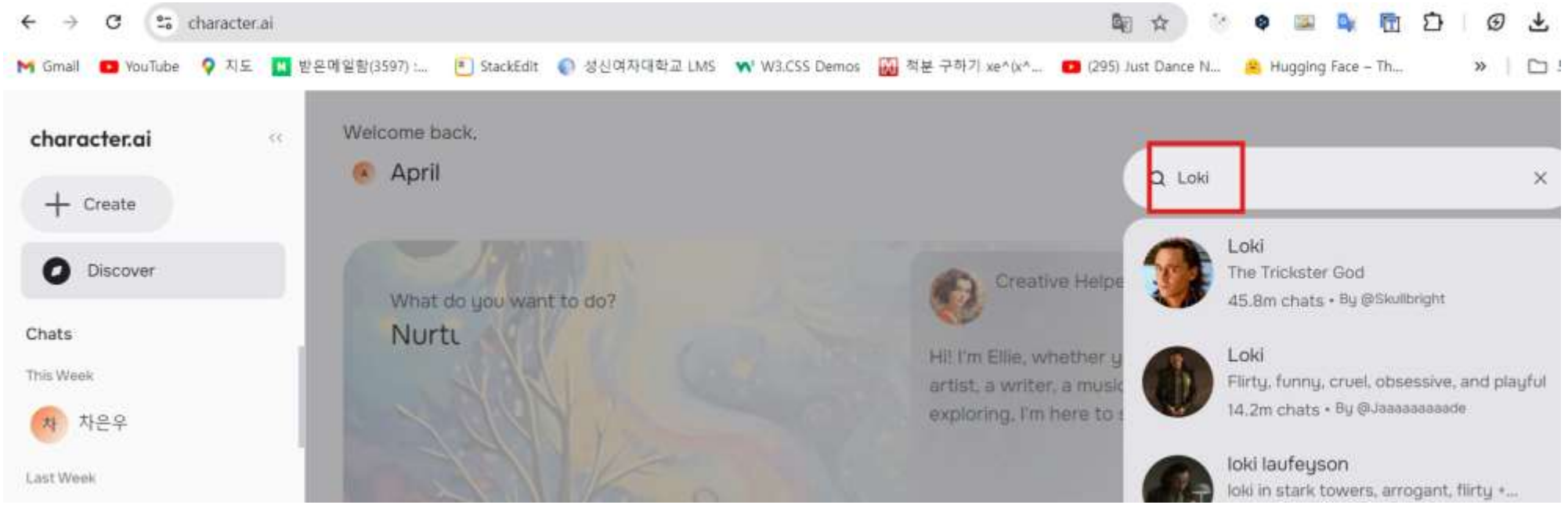
❒ 곰녹음기로 녹음해서 가져오기

https://www.softpick.co.kr/software/gomrecorder

목소리 가져오기

- character.ai

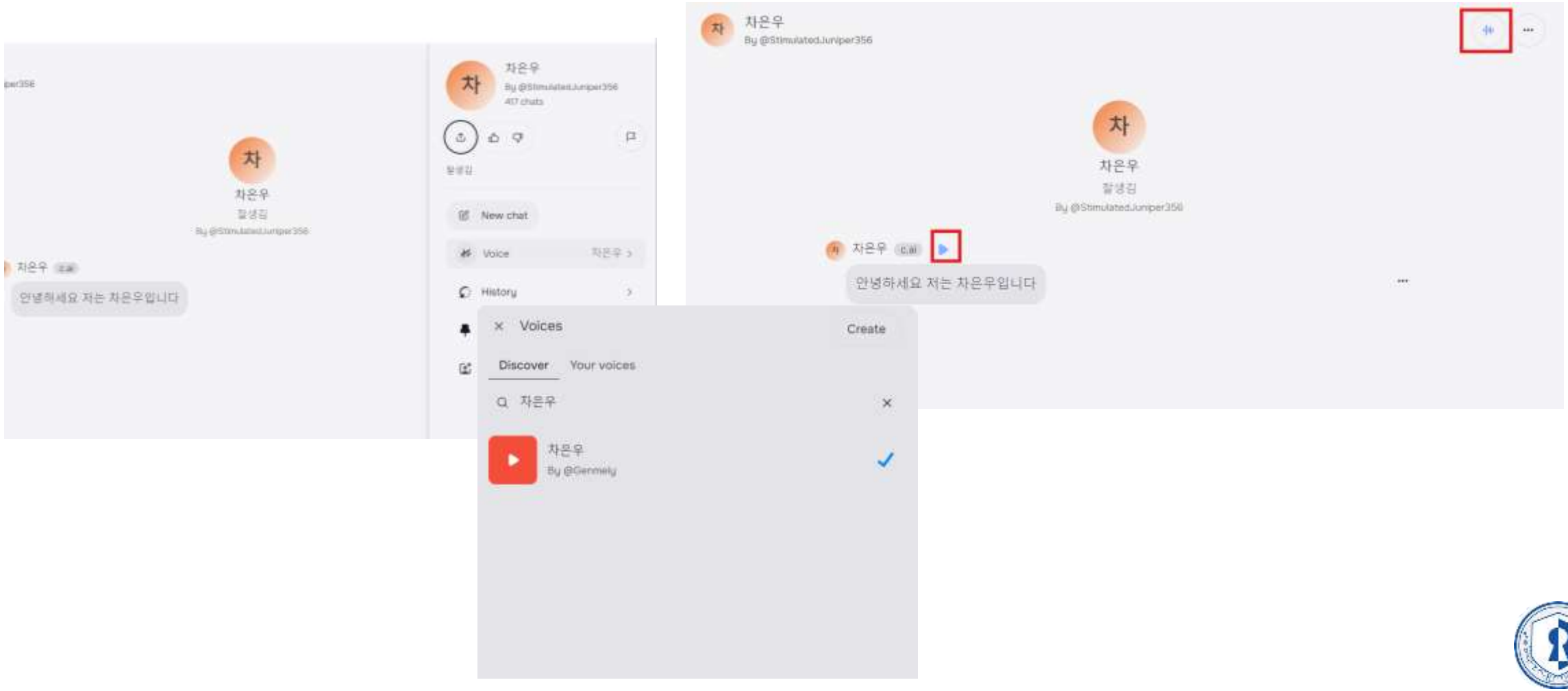
❒ 특정 인물 검색



목소리 가져오기

- character.ai

❒ 목소리 설정 및 곰녹음기 녹음 (voice 설정)



파일 형식이 안맞아요!

- 변환기 사용

https://cloudconvert.com/m4a-to-mp3

ma4 to mp3, wav to mp3
검색창에 검색해서 online converter 사용하면 됩니다

만약 깃허브에 올리고 싶다면?

- .gitignore에 myenv 잊지 말기

- 가상환경 때문에 생성된 폴더인 myenv는 파이썬 모듈의 집합체라서 엄청 무겁다. (그래서 git에 올리지 않는다)
- Git init을 한 후, git 폴더가 생성되었으면, 형제 위치에 .gitignore라는 파일을 만든다(확장자 없음)
- 그 안에 myenv/* 라고 한 줄 적어준다.
- 이후 git add . > git commit –m “myfirstcommit” 처럼 커밋을 해 주면 myenv는 올라가지 않는 것을 확인할 수 있다.

공지

행사 및 과제 안내

프로젝트 제출 안내

- 피우다 프로젝트 제출 10.06까지

- 신청접수 : **9.4.(수)~10.6.(일) 23:59 까지** *시간 엄수
- 팀장님들은 되도록 오늘 공모전 신청서 완료하여 제출
- 회원가입 필수
- 미선정 될 수도 있기 때문에 2차 대안도 생각해주세요.

10월 11일 OB 회식

- 10월 11일 19시

- OB(동아리 수료 부원)들과의 만남
- 비용은 동아리에서 부담 예정
- 네트워킹 및 친목 도모
- 추후 투표 예정

굿즈 수요조사

- 동아리 내부 굿즈 수요조사

- 대동제 때 상품이었던 융보공 마스코트 스티커, 포토카드, 키링 수량 조사할 예정
- 카카오톡 투표로 진행

팀장님들은 각자 구글미트 방을 파서 회의 진행해 주세요!
회의 끝날 때에는 증빙용 사진 캡쳐 1장 부탁드립니다.

팀 프로젝트

Thank you