

AI 기반 드론 제어 기초

2일차

강의 개요

1. AI 기반 드론 제어 시스템 구조
2. 음성 인식과 AI 통합
3. 드론 제어 시스템 구현
4. 웹 기반 GUI와 안전 기능 구현

1. AI 기반 드론 제어 시스템 구조

필수 라이브러리 소개

- DJITelloPy: 텔로 드론 제어
- SpeechRecognition: 음성 인식
- OpenAI/Google Gemini: AI 명령 처리
- Flask: 웹 기반 인터페이스
- gTTS: 텍스트 음성 변환

환경 설정

1. API 키 설정

```
# .env 파일 설정
OPENAI_API_KEY=your_openai_key
GOOGLE_API_KEY=your_gemini_key

# 코드에서 로드
from dotenv import load_dotenv
load_dotenv()
```

2. 기본 패키지 설치

```
pip install djitellopy opencv-python
pip install speechrecognition openai
pip install google-generativeai
pip install flask gtts pygame
```

드론 컨트롤러 기본 구조

```
class DroneController:
    def __init__(self):
        self.tello = Tello()
        self.frame_reader = None
        self.is_streaming = False

    def connect(self):
        """드론 연결 및 상태 확인"""
        print("드론에 연결 중...")
        self.tello.connect()

        battery = self.tello.get_battery()
        print(f"배터리 잔량: {battery}%")

        if battery < 20:
            raise Exception("배터리 부족")
```

2. 음성 인식과 AI 통합

음성 인식 시스템

```
import speech_recognition as sr
```

```
def setup_voice_recognition():  
    recognizer = sr.Recognizer()  
    with sr.Microphone() as source:  
        recognizer.adjust_for_ambient_noise(source) ## 주변 소음 조정  
        print("음성 인식 준비 완료")  
    return recognizer  
def recognize_speech(recognizer):  
    with sr.Microphone() as source:  
        print("명령을 말씀해주세요...")  
        audio = recognizer.listen(source)  
        return recognizer.recognize_google(audio, language='ko-KR')
```

1. OpenAI GPT 활용

```
def process_voice_command(audio_text: str) -> Dict:
    system_prompt = """
    당신은 드론 제어 시스템입니다.
    사용자의 자연어 명령을 드론 제어 명령으로 변환합니다.

    예시:
    - "위로 1미터 올라가줘" -> move(up, 100)
    - "앞으로 30센티미터" -> move(forward, 30)
    """

    response = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": system_prompt},
            {"role": "user", "content": audio_text}
        ],
        tools=[{
            "type": "function",
            "function": {
                "name": "control_drone",
                "parameters": {
                    "type": "object",
                    "properties": {
                        "command": {
                            "type": "string",
                            "enum": ["takeoff", "land", "move"]
                        }
                    }
                }
            }
        }
    ])
    return response
```

2. Google Gemini 활용

```
def process_voice_command(audio_text: str) -> Dict:
    """음성 명령을 Function calling 형식으로 변환"""
    prompt = """당신은 드론 제어 시스템입니다. 사용자의 자연어 명령을 드론 제어 명령으로 변환합니다.
```

가능한 명령어와 형식:

1. 이륙: {"command": "takeoff"}
2. 착륙: {"command": "land"}
3. 이동: {"command": "move", "parameters": {"direction": "[up/down/left/right/forward/back]", "distance": [20-500]}}
4. 회전: {"command": "rotate", "parameters": {"direction": "[clockwise/counter_clockwise]", "angle": [1-360]}}

예시:

- "위로 1미터 올라가줘" -> {"command": "move", "parameters": {"direction": "up", "distance": 100}}
- "오른쪽으로 90도 돌아" -> {"command": "rotate", "parameters": {"direction": "clockwise", "angle": 90}}

사용자 명령을 위 JSON 형식으로 변환하여 응답해주세요. 응답은 반드시 유효한 JSON 형식이어야 합니다.

사용자 명령: """ + audio_text

```
try:
    response = model.generate_content(prompt)
    # JSON 문자열을 찾아 파싱
    response_text = response.text
    # JSON 부분만 추출 (중괄호로 둘러싸인 부분)
    json_str = response_text[response_text.find("{"):response_text.rfind("}") + 1]
    command = json.loads(json_str)
    return command

except Exception as e:
    print(f"Gemini API 오류: {str(e)}")
    raise
```


function calling 이해하기

- 미리 함수를 정의해놓고 LLM이 어떤 함수를 사용할건지 판단하도록 하는 방법

```
def execute_function(self, function_name: str, parameters: Dict[str, Any] = None):
    try:
        if function_name == "takeoff":
            print("이륙!")
            return self.tello.takeoff()

        elif function_name == "land":
            print("착륙!")
            return self.tello.land()

        elif function_name == "move":
            ...생략...
        elif function_name == "rotate":
            if direction == "clockwise":
```

3. 드론 제어 시스템 구현

명령 실행 시스템

```
def execute_function(self, command: str, params: Dict = None):
    try:
        if command == "takeoff":
            self.tello.takeoff()
        elif command == "land":
            self.tello.land()
        elif command == "move":
            direction = params["direction"]
            distance = params["distance"]
            if direction == "up":
                self.tello.move_up(distance)
            elif direction == "forward":
                self.tello.move_forward(distance)
            # ... 기타 방향 처리
```

비디오 스트리밍 처리

```
def start_video_stream(self):  
    """비디오 스트리밍 시작"""  
    self.tello.streamon()  
    time.sleep(2) # 스트림 초기화 대기  
    self.frame_reader = self.tello.get_frame_read()  
    self.is_streaming = True  
  
    # 스트리밍 스레드 시작  
    self.stream_thread = threading.Thread(  
        target=self._stream_loop  
    )  
    self.stream_thread.daemon = True  
    self.stream_thread.start()
```

4. 웹 기반 GUI와 안전 기능 구현

Flask 기반 웹 인터페이스 (opencv도 필요함)

```
from flask import Flask, render_template, Response
```

```
def get_frame():  
    while True:  
        if controller and controller.frame_reader:  
            frame = controller.frame_reader.frame  
            _, buffer = cv2.imencode('.jpg', frame)  
            frame_bytes = buffer.tobytes()  
            yield (b'--frame\r\n'  
                  b'Content-Type: image/jpeg\r\n\r\n' +  
                  frame_bytes + b'\r\n')
```

통합 시스템 구현

```
def main():  
    controller = DroneController()  
    recognizer = setup_voice_recognition()  
  
    try:  
        controller.connect()  
        controller.start_video_stream()  
  
        # Flask 서버 시작  
        app.run(host='0.0.0.0', port=3000)  
  
    except Exception as e:  
        print(f"오류 발생: {str(e)}")  
        controller.emergency_land()
```

실습 과제

1. 예제코드 실행하기

- 환경설정하기
- 실행하기
- 드론 연결 및 상태 확인
- 비디오 스트리밍 설정
- 기본 명령어 처리

2. LLM 기능 사용하기

- OpenAI/Gemini API 연동
- 음성 명령 처리 구현
- 자연어 해석 시스템

3. 기능 구현하기

- 배터리 모니터링
- 비상 착륙 시스템
- 에러 처리

참고 자료

1. API 문서

- DJITelloPy: <https://djitellopy.readthedocs.io/>
- OpenAI: <https://platform.openai.com/docs>
- Gemini: <https://ai.google.dev/docs>

2. 예제 코드

- tello-scan-surroundings.py
- voice-control-tello-gemini.py
- tello-webui.py