



오픈AI를 통해 AI사진  
구분하기

AISW

20251599 김시우

# 목차

- 문제점
- 추진배경
- 생성형 AI활용
- 기대효과





# 문제점

- 최근 몇 년간 AI가 큰 발전을 해오다 보니 그럴 듯한 AI사진들이 유튜브나 인터넷에 공공연하게 퍼지고 있습니다.

# 추진배경

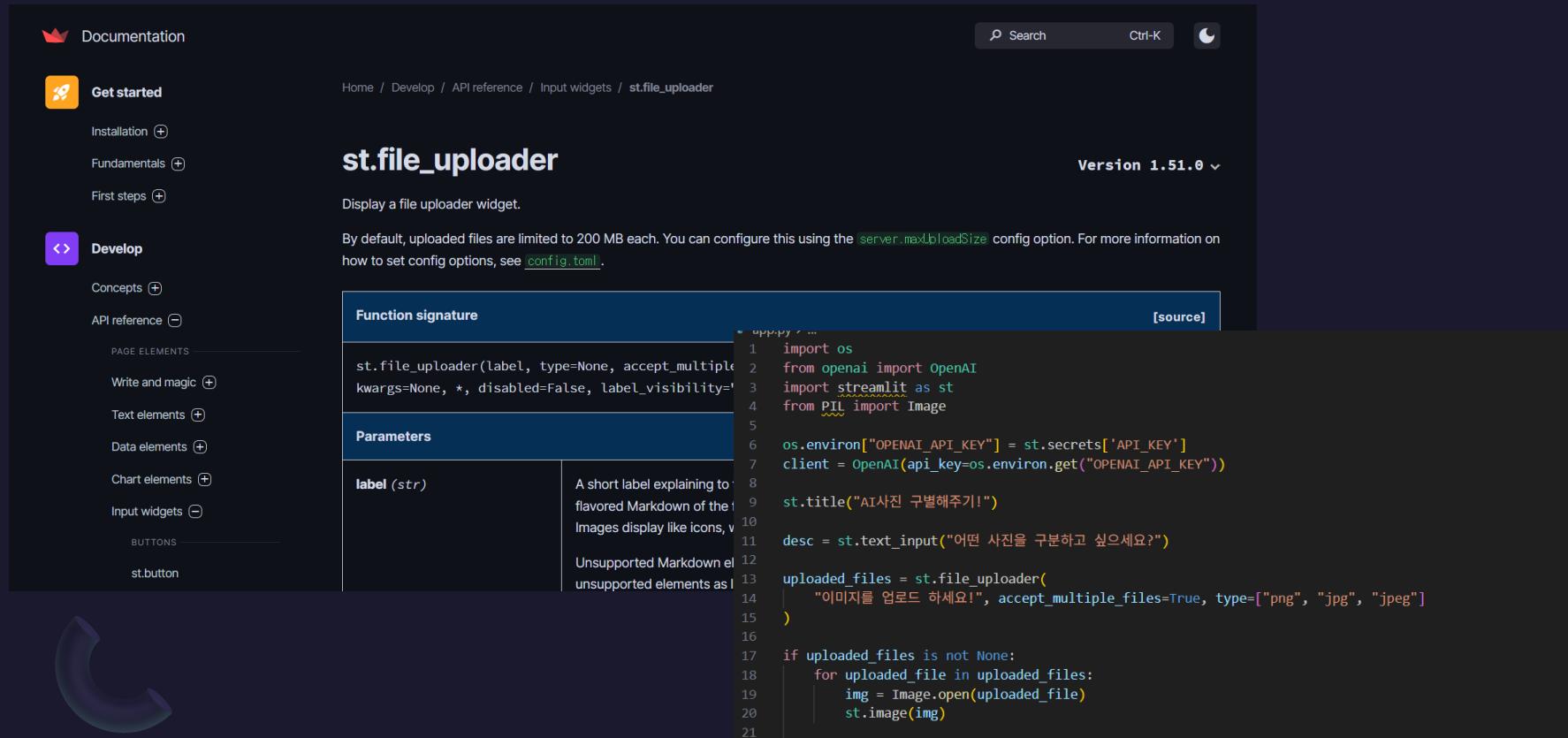
앞서 나온 문제점들로 인해 ai로 인한  
피해가 나오고 있어 프로젝트를 통해 어  
느정도 기초를 잡아 해결을 해보고자 함

# 활용 기술

The image is a collage of three screenshots illustrating various technologies:

- Streamlit Input widgets documentation:** A screenshot of the Streamlit documentation website under the "Input widgets" section. It shows a "Button elements" example with a "Click me" button and its corresponding code: `clicked = st.button("Click me")`.
- Streamlit Button example:** A screenshot of a Streamlit application showing a "Button" element with the text "Click me".
- Github AI interface:** A screenshot of the GitHub AI interface, showing a "Welcome to GitHub" message and a search bar. Below it, a "Playlist" for "GitHub for beginners on YouTube" is displayed, featuring a video thumbnail for "What is Git?".

# 과정



The screenshot shows the Streamlit API reference for the `st.file_uploader` widget. The page is titled "st.file\_uploader" and indicates "Version 1.51.0". The left sidebar is the "Develop" section, which includes "Get started", "Installation", "Fundamentals", "First steps", and "API reference". The "API reference" section is expanded, showing categories like "PAGE ELEMENTS", "Text elements", "Data elements", "Chart elements", "Input widgets", and "BUTTONS". The "Input widgets" section is expanded, showing the `st.button` component. The main content area shows the `st.file_uploader` function signature and its parameters. The function signature is:st.file\_uploader(label, type=None, accept\_multiple\_files=False, key=None, \*\*kwargs) → FileUploaderThe parameters are:label (str) A short label explaining to the user what to upload. It can be styled with flavored Markdown of the form ``. Images display like icons, while unsupported elements are displayed as text.  
type (str) The type of file to accept. It can be a single file type (e.g., "image/png") or a list of file types (e.g., ["image/png", "image/jpeg"]).The "Parameters" table also includes a note: "Unsupported Markdown elements are displayed as text." Below the parameters, the source code for the `st.file_uploader` function is provided:import os  
from openai import OpenAI  
import streamlit as st  
from PIL import Image  
  
os.environ["OPENAI\_API\_KEY"] = st.secrets['API\_KEY']  
client = OpenAI(api\_key=os.environ.get("OPENAI\_API\_KEY"))  
  
st.title("AI 사진 구별해주기!")  
  
desc = st.text\_input("어떤 사진을 구분하고 싶으세요?")  
  
uploaded\_files = st.file\_uploader(  
 "이미지를 업로드 하세요!", accept\_multiple\_files=True, type=["png", "jpg", "jpeg"]  
)  
  
if uploaded\_files is not None:  
 for uploaded\_file in uploaded\_files:  
 img = Image.open(uploaded\_file)  
 st.image(img)

# 웹페이지

<https://aiimagedetect11.steamlit.app/>



# AI사진 구분 하는 open ai



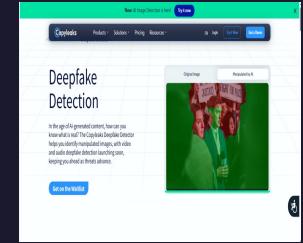
Vastav AI



Isogen.ai



TruthScan



Copyleaks  
Deepfake  
Detector

# 기대효과

안 좋은 쪽으로의 ai의 사용을 막을 수 있다는 점이고 구분 기술을 통해 진실과 거짓을 구분 할 수 있는 힘을 기를 수 있다.

편리함 보다는 이런 기술이 있다는 것을 알려줌으로써 이 분야에 대해 더 잘 알게 된다는 것이고 이런 ai사진을 구분하는 기술을 통해 현대사회에 적응했다는 느낌을 들게 할 수 있다는 것이다.

시장 규모를 확대하려면 이 기술이 필요한 언론이나 법률 및 저작권 쪽, 소셜미디어 쪽에 어필을 할 수 있다는 것이다.

문제해결

사용자 및 기업

편리함 및 만족도

비용절감, 시간단축

시장규모 확대

사용자에게는 간편하게 거짓과 진실을 구분할 수 있는 기술을 사용할 수 있다는 것이고 기업은 무분별한 ai로 인한 가짜뉴스를 어느정도 방지할 수 있다는 것이다.

이런 프로젝트를 크게 벌리려면 사이트 운영비와 더 많은 관리자가 필요하고 더욱더 많은 ai데이터를 모으면 시간이 단축이 될 수도 있다.



# 결과(총정리)

준비하면서

- 웹페이지를 만드는 것 자체가 처음이라 많이 힘들었음
- 평생 한번도 접하지 않은 파이선을 가지고 과제를 하다 보니 뇌에 과부하가 일어났음
- 맞게 한 것 같은데 에러가 나서 고치는 것에 상당히 애를 먹었음

끝 마치면서

- 코드를 작성하거나 웹페이지를 만드는 것이 생소하였지만 이 또한 좋은 경험이라는 것이라는 것을 알게 되었다.



# 요약

A I 이미지를 구분하는 웹페이지를 만들고 실험을 해보면서 다른 a i 이미지를 구분하는 웹도 다양하게 알아보게 되었음 .

# 감사합니다.

20251599 김시우

