PROJECT

유물복원 우선 순위 산출 시스템

10108 박진우

목차 LIST

01 (솔직한)동기

현재 역사 관련 학과를 지망하고 있습니다.(혼자인 이유도 여기에…)그러기에 아이디어를 다른 팀들과는 달리 보다 제한적으로 찾는데 바로 떠오른 것이 유물 복원입니다. 하지만 이미 있는 기술입니다. 그럼에도 불구하고 유물 복원이라는 주제를 포기할 수 없었습니다. 이것 만큼 코딩 동아리와 맞는 좋은 아이디어가 없다고 생각했습니다. 그래서 이미 있는 이 기술에 실현 가능한 무언가를 추가하는 것으로 시도해봤습니다.

02 아이디어 소개

원래 존재하던 유물 복원 기술에 우선 순위를 매겨 어떤 유물을 먼저 복원해야 할지 제시하는 시스템

기대효과

1)복원 대상 선정 과정에서의 객관성 확보.
2)제한된 예산과 인력을 효율적으로 활용할 수 있을 것.

03 구현방법(Python)

01 유물 사진을 입력한다.

02 이미지분석을 통해 손상 정도를 계산한다.(OpenCV)

03 손상도, 재질 난이도, 역사적 가치 점수를 합산한다.

04 최종 점수를 계산하고 우선 순위 목록을 생성한다.

05 막대그래프 형태로 결과를 시각화 한다. (Matplotlib)

예시코드(손상 정도 계산.시각화)

```
import cv2
import matplotlib.pyplot as plt
img = cv2.imread('artifact.png', 0)
edges = cv2.Canny(img, 100, 200)
damage_ratio = (edges > 0).sum() / edges.size * 100
plt.bar(['유물 A'], [damage_ratio])
plt.title('손상 정도 (%)')
plt.show()
```

평가기준		
손상정도	50%	자동계산
재질 난이도 :복원 어려운 재질	30%	사용자 입력
역사적 가치 :문헌 기록에 따른 중요도,오래된정도	20%	사용자 입력

04 한계

귀찮음이슈

사용자가 직접 입력해야 하는 부분이 꽤 있음

하드웨어 No 사용

다른 팀들은 다 사용하는데 나만 쓰지 않아 동아리 소속감이 느껴지지 않음

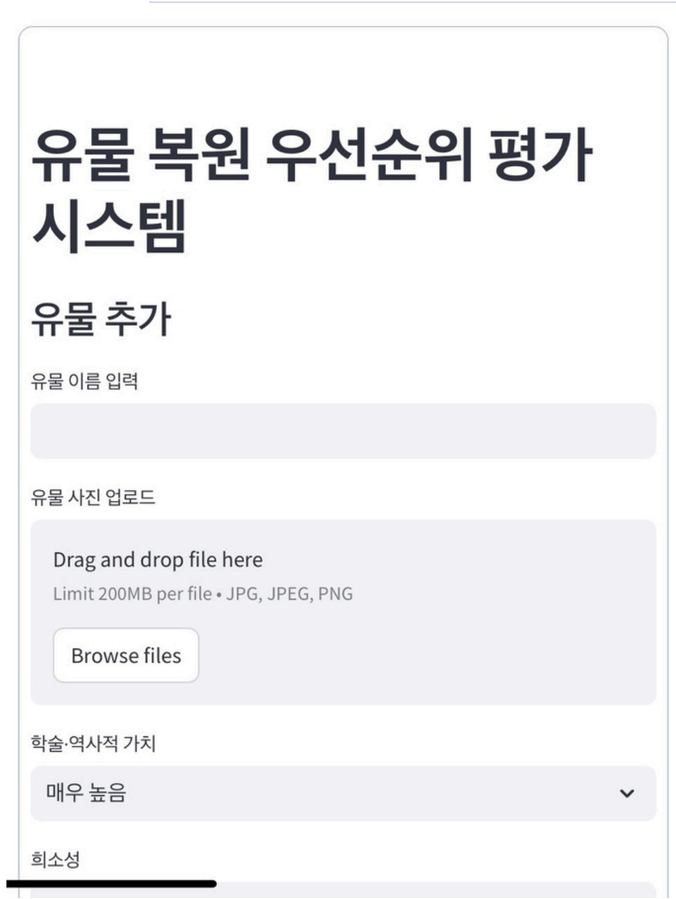
실용성

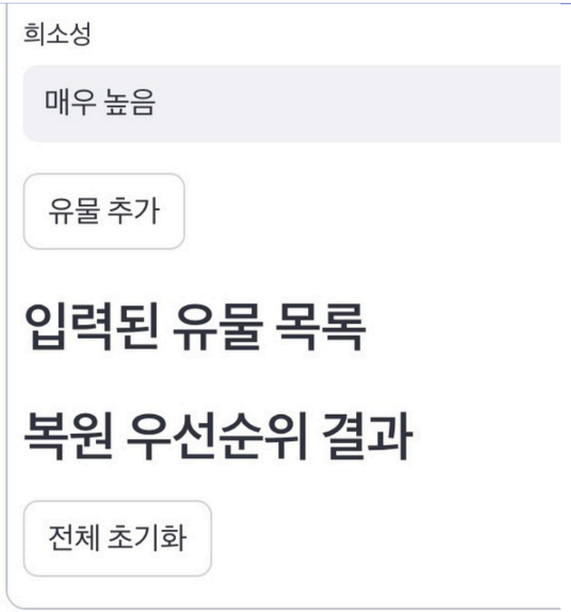
특정 분야를 제외하면 쓸 일이 거의 없음.

감사합니다

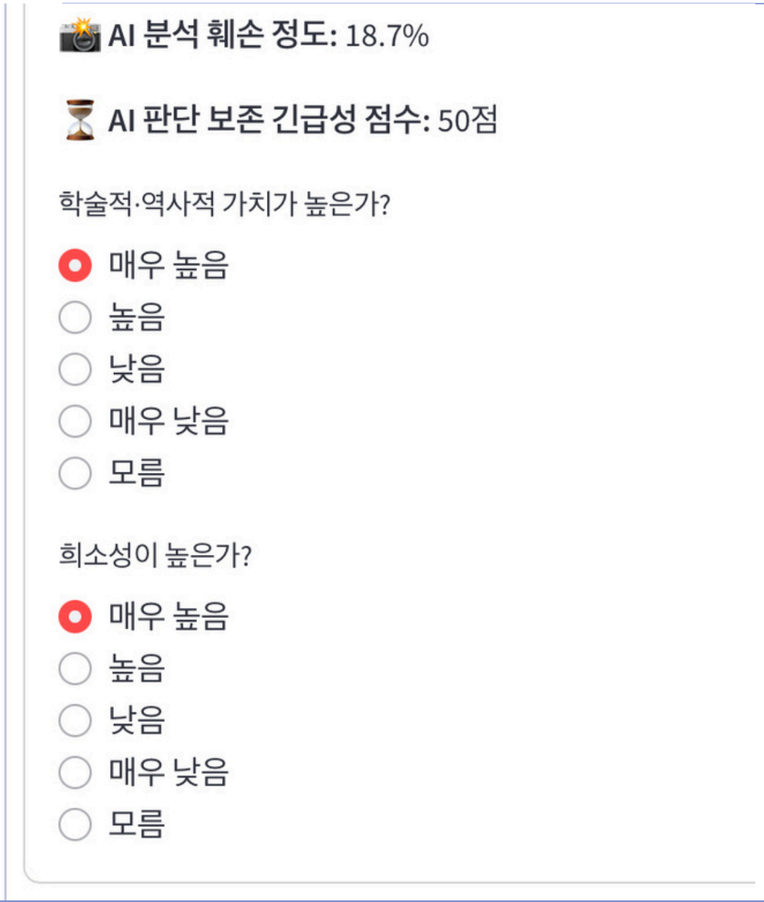
목차 LIST

05 유물 복원 우선순위 산출시스템





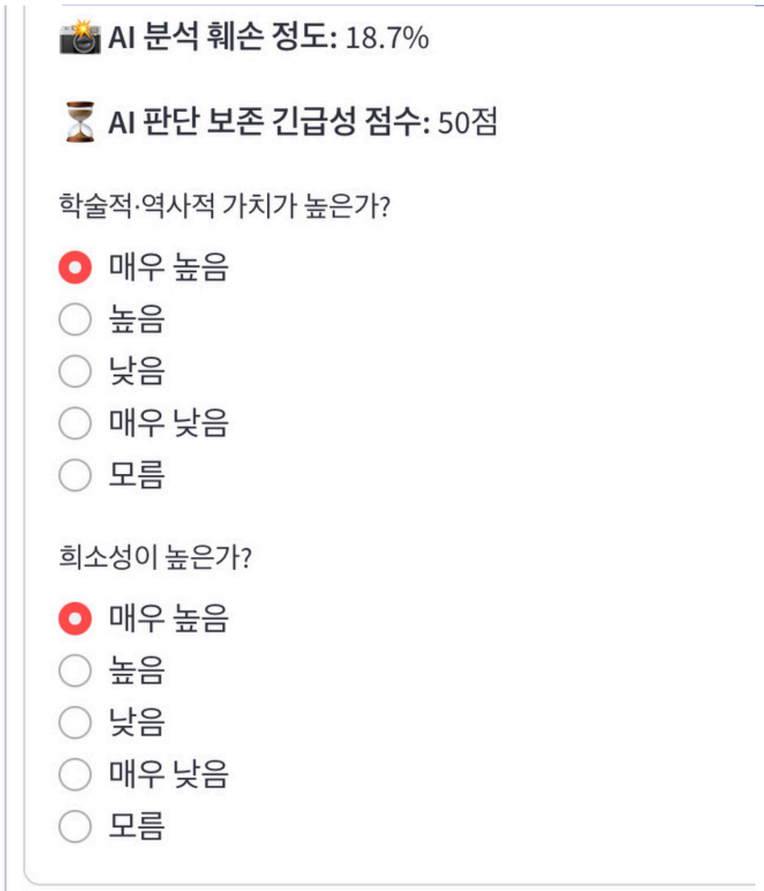
06 복원 우선순위 산출 방법



1. 훼손정도:눈에 보이는 훼손정도 30%
2. 보존 긴급성:방치 시 추가 손상 가능성 25%
3. 학술적 가치:연구 가치+전시 가치 25%
4. 희소성:동일 또는 유사 유물 존재 여부 20%

최종점수=(훼손점수*0.3)+(보존긴급성*0.25)+(학술적 가치*0.25)+(희소성*0.2)

06 복원 우선순위 산출 방법



매우 높음	100점
높음	75점
낮음	25점
매우 낮음	0점
모름	점수제외

06 복원 우선순위 산출 방법

코드

```
if st.button("📊 유물 분석 및 순위 계산"):
    calculate_and_display_ranking()

if st.button("🔄 전체 초기화"):
    st.session_state.artifacts = []
    st.experimental_rerun()
```

python

```
score = (
    damage_score * weights["damage"] +
    urgency_score * weights["urgency"] +
    academic_score * weights["academic"] +
    rarity_score * weights["rarity"]
)
```

python

```
weights = {
    "damage": 0.25,
    "urgency": 0.30,
    "academic": 0.25,
    "rarity": 0.20
}
```

python

```
score_map = {
    "매우 높음": 100,
    "높음": 70,
    "낮음": 30,
    "매우 낮음": 0,
    "모름": None
}
```

```
valid_scores = []
valid_weights = []

for key, value in scores.items():
    if value is not None:
        valid_scores.append(value)
        valid_weights.append(weights[key])

normalized_weights = [w / sum(valid_weights) for w in valid_weights]

final_score = sum(
    s * w for s, w in zip(valid_scores, normalized_weights)
)
```

07 사진 분석 AI 툴

1.사진을 흑백 이미지로 변환

2.경계선을 찾는 엣지 검출 적용

3.연속되지 않은 경계선,끊긴 윤곽선의 비율을 계산

4.전체 이미지 픽셀 대비 해당 경계선이 차지하는 비율 산출

🔍 AI 훼손 정도: 낮음 (18.8%)

07 사진 분석 AI 툴

📦 유물1

🔍 **AI 훼손 정도:** 매우 낮음 (9.6%)

한계

07 한계를 극복하려면?


아텍 에바 (Artec Eva)
24,200,000원 무료
3D프린터스토어
해외 3D 스캐너 스캐너 artec eva color handheld scanner
an body real photo studio maker
32,720,900원 10,000원
신속
3D扫描仪
彩色 人体
해외 3D 스캐너 Artec artec eva 컬러 휴대용
오 의학
36,500,900원 10,000원
찜 1
신속
LEICA BLK3D IMAGE SCANNER
라이카 BLK3D 이미지 스캐너
12,941,200원


DIST

07 사진 분석 AI 툴

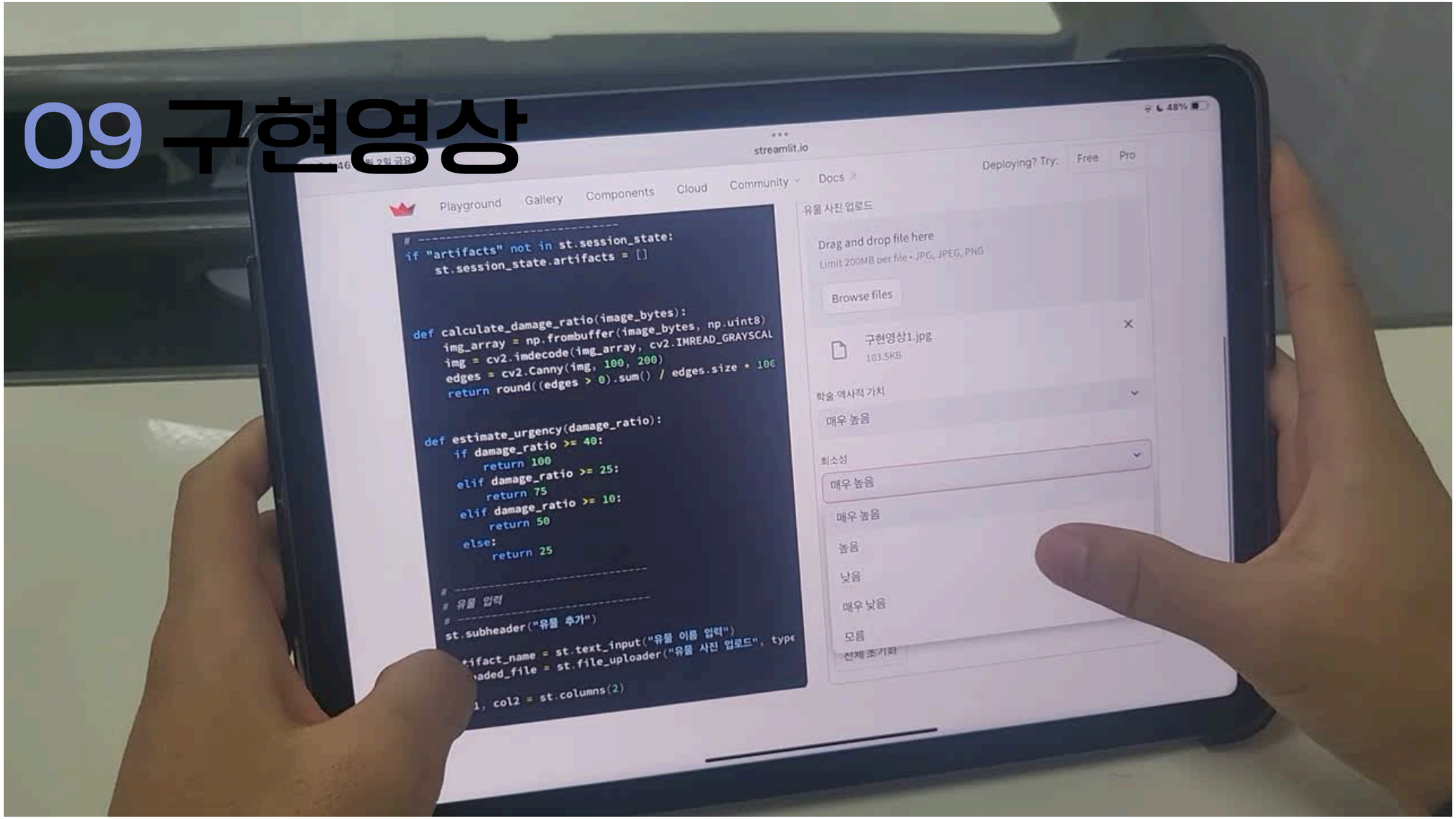
코드

```
if "artifacts" not in st.session_state:
    st.session_state.artifacts = []



def calculate_damage_ratio(image_bytes):
    img_array = np.frombuffer(image_bytes, np.uint8)
    img = cv2.imdecode(img_array, cv2.IMREAD_GRAYSCAL
    edges = cv2.Canny(img, 100, 200)
    return round((edges > 0).sum() / edges.size * 100


def estimate_urgency(damage_ratio):
    if damage_ratio >= 40:
        return 100
    elif damage_ratio >= 25:
        return 75
    elif damage_ratio >= 10:
        return 50
    else:
        return 25
```

08 추천사용자

고고학자

09 구현영상


streamlit.io
Playground
Gallery
Components
Cloud
Community
Docs
Deploying? Try:
Free
Pro
유물 사진 업로드
Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG
Browse files
구현영상1.jpg
103.5KB
학술·역사적 가치
매우 높음
최소성
매우 높음
매우 높음
높음
낮음
매우 낮음
모름
선택 초기화
if "artifacts" not in st.session_state:
st.session_state.artifacts = []
def calculate_damage_ratio(image_bytes):
img_array = np.frombuffer(image_bytes, np.uint8)
img = cv2.imdecode(img_array, cv2.IMREAD_GRAYSCAL
edges = cv2.Canny(img, 100, 200)
return round((edges > 0).sum() / edges.size * 10
def estimate_urgency(damage_ratio):
if damage_ratio >= 40:
return 100
elif damage_ratio >= 25:
return 75
elif damage_ratio >= 10:
return 50
else:
return 25
# 유물 입력
st.subheader("유물 추가")
col1, col2 = st.columns(2)

10 느낀점

처음에는 진로와 엮는데 어려움이 있었지만 어려움이 있어서 단순하지 않은 나름 특이한 주제로 하게 된 것 같습니다. 그리고 물론 ai와 함께하기는 했지만 시행착오를 겪어 결과물이 나오는 것에 대한 뿌듯함을 느끼고 이 동아리에서 제 미래를 보았습니다. 마지막으로 고고학자에 대한 관심도 생기는 개인적으로 의미있는 프로젝트였던 것 같습니다.

감사합니다