

개인 프로젝트 발표

박우빈

2025년 6월 20일 (금)

index

기획 배경

- 개요
- 배경

모델 소개

- 주 타겟
- 구현 기능

구현 상세

- 개발 내용
- 문제 해결
- 개발 요약

마무리

- 기타
- 참고자료

개요

프로젝트

얼굴 조작 검출 모델

영상의 조작 여부를 판별하고 이를 알려주는 프로그램입니다.



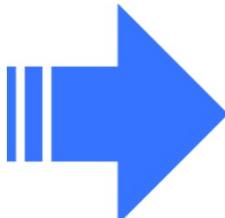
배경

"AI 협업 영상 제작 증가"

쉽고 빠르며 높은 퀄리티

딥페이크 범죄 증가

노년층, 아이 정보 취약



조작의 경각심을 늘리고
정보의 진실성 향상

▶ 뉴스 > 정책뉴스

딥페이크 범죄 집중단속 963명 검거…10월 말까지

경찰청, 피해영상을 1만 535건 삭제·차단 요청 및 피해 보호 활동 추진

2025.04.17 | 경찰청

가 목록

주 타겟

페르소나



성별 : 여성

이름 : 박지민 (29)

직업 : 유튜브 크리에이터

특징 : 소식이 빠른 유튜브를 활용하여
컨텐츠를 만들고 있음

니즈 : 영상이 믿을만한 자료였으면
좋겠음

타겟 : 일반인, 유튜브 영상 제작자, 기자

선정이유 : 부모님의 가짜뉴스를 경계하는
일반인과 자료의 신뢰도를 중요시하는 기자,
영상제작자 등을 포함



성별 : 남성

이름 : 김정훈(35)

직업 : 디지털 포렌식 수사관

특징 : 법정이나 검찰에 사용되는 증거를
검출하는 작업을 함

니즈 : 영상의 조작을 증명하는 신빙성
자료를 빠르게 얻고 싶음

타겟 : 검찰, 수사관, 법정 관계자, 피고인

선정이유 : 특정 사람 [자신 포함] 의 유죄/
무죄를 판별하기 위한 자료를 추구, 신뢰성이
있으며 빠르게 도출할 수 있으면 좋음

구현 기능

01

영상의 조작 판단 및 표시

영상을 학습하여 조작 여부를 판단하고 조작 영상이면 해당 부분을 표시해줍니다.

02

간략하고 빠르게 경량화

작동 성능과 학습 시간을 고려하여 빠르고 돌 수 있게 구성하였습니다.



구현 가능

Original



Augmented

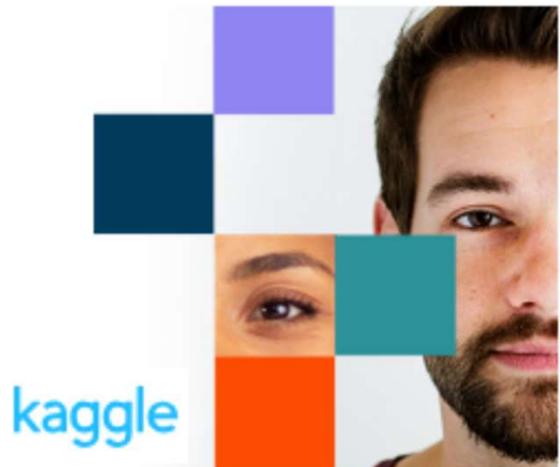


Actual: FAKE
Pred: FAKE (0.69)



개발 내용

사용 Datasets



Deepfake Detection
Challenge [Kaggle]

주요 Metrics

Accuracy

Precision, Recall, F1
score

ROC AUC

개발 내용

『주요 스택』



- 언어/런타임: Python
- 딥러닝 프레임워크: TensorFlow/Keras, PyTorch
- 전처리/증강: OpenCV, MTCNN, Albumentations
- 피처 추출: Keras Applications(Xception)
- 불균형 처리: SMOTE (imblearn)
- 분류기: scikit-learn(SVM, ROC AUC), Keras MLP
- 객체 검출: YOLOv8 (Ultralytics)
- 유틸리티: NumPy, matplotlib, joblib

모델 Architecture

분류 Classification

SVM&MLP
soft / hard Ensemble

검출 Detection

YOLOv8

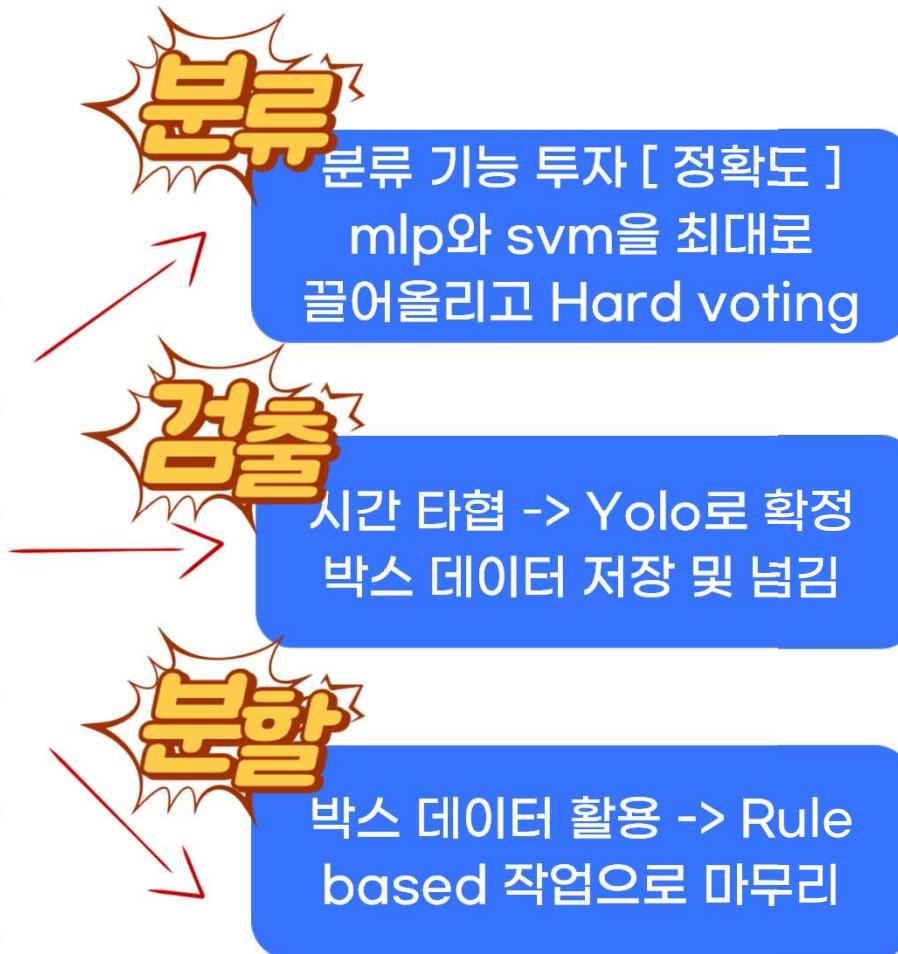
분할 Segmentation

Rule-Based Segmentation

개발 내용

주요 일정 요약

Day	작업 내용
Day 1 (6/9)	기획
Day 2 (6/10)	환경 구축 & 데이터 준비
Day 3–Day 6 (6/11 ~ 6/16)	Classification 성능 극대화 - › 각 모델 대비 및 하이퍼파라미터
Day 7(6/17)	Detection & Segmentation 구현 및 연결
Day8 ~ Day9 (6/18 ~ 6/19)	모델 구현 방향 조정 -> 분류에 투자 + 시간 절약 구축
Day10 (6/20)	발표 자료 완성 및 발표



문제 사항

문제 1

[SVM] Val Accuracy: 0.8125, Val AUC: 0.5508

	precision	recall	f1-score	support
REAL	0.50	0.20	0.29	15
FAKE	0.84	0.95	0.89	65
accuracy			0.81	80
macro avg	0.67	0.58	0.59	80
weighted avg	0.77	0.81	0.78	80

Epoch 9/10
320/320 - 0s - loss: 0.5967 - auc: 0.7395 - val_loss: 0.7640 - val_auc: 0.5723
Epoch 10/10
320/320 - 0s - loss: 0.5504 - auc: 0.7866 - val_loss: 0.7996 - val_auc: 0.5528
[MLP balanced] Val Loss: 0.6175, Val AUC: 0.5528

문제 사항

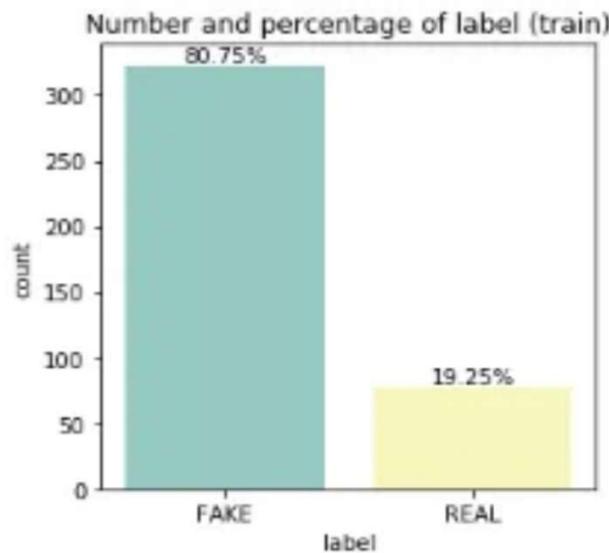
- 초반에 Xception net을 기반으로 학습
- 그 이후 성능 Test를 위해 SVM, MLP를 돌려 보았는데 성능이 엄청 떨어짐

접근 방식

- recall 지표를 보니 REAL 부분만 현저히 떨어지는 것을 확인 -> 데이터 셋을 한번 확인

문제 사항

문제 1 - 접근



문제 접근

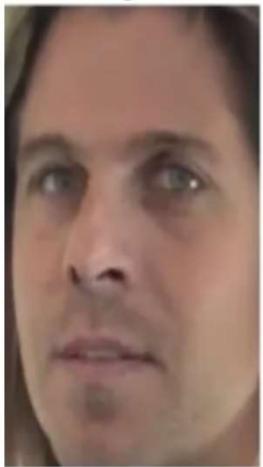
- REAL 값이 현저히 떨어짐 -> SMOTE
- 결측치 존재 -> 라벨링 보완
- 하이퍼 파라미터 적용 [Grid Search]

소폭상승

문제 사항

문제 1 - 보류

Original



MLP AUC: 0.7856410256410256
SVM AUC: 0.757948717948718
Ensemble AUC: 0.7774358974358975

	precision	recall	f1-score	support
REAL	0.67	0.27	0.38	15
FAKE	0.85	0.97	0.91	65
accuracy			0.84	80
macro avg	0.76	0.62	0.64	80
weighted avg	0.82	0.84	0.81	80

Epoch 11/15
40/40 - 1s - loss: 0.5356 - auc: 0.8166 - val_loss: 0.4376 - val_auc: 0.7774358974358975
Epoch 12/15
40/40 - 1s - loss: 0.5231 - auc: 0.8312 - val_loss: 0.5300 - val_auc: 0.7774358974358975
Epoch 13/15
...

	accuracy			0.68	80
macro avg	0.60	0.65	0.59	80	
weighted avg	0.78	0.68	0.71	80	

문제 접근

- MTCNN
- Soft Ensemble 시도
- LSTM 시도

문제 사항

문제 2



문제 사항

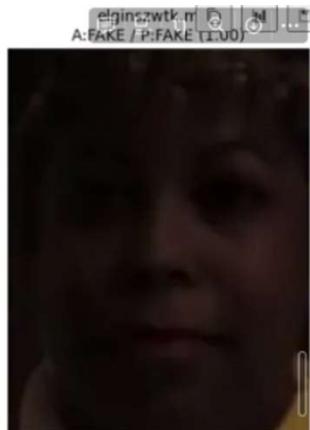
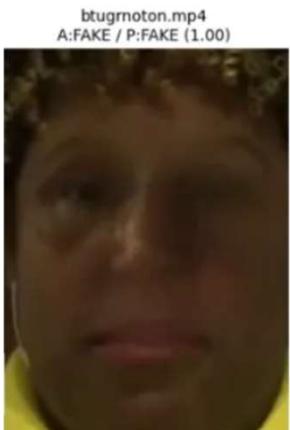
- 시간이 압도적으로 부족함
- 한번 돌리는데 기본 30~40분
- 근데 여기서 Yolo vs Fast r-cnn
- Segmentation을 개발할 시간이 없음

접근 방식

- 적당히 타협점을 찾아 접근 방식 수정

문제 사항

문제 3



문제 사항

- LSTM 의 성능 문제 발견
- 예측에 있어 잘못된 예측을 자신있게 함

접근 방식

- 분류 모델의 성능을 결국 다시 올려야 함

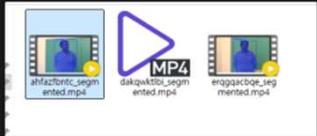
문제 사항

문제 4

→ 문제 발생

tensorflow가 기타 모듈을 설치하는 과정에서 최신 버전으로 설정 2.13.0

근데 pixellib 에서 deeplabv3는 tensorflow가 버전이 최소 2.9.0 아래여야 작동함



문제 사항

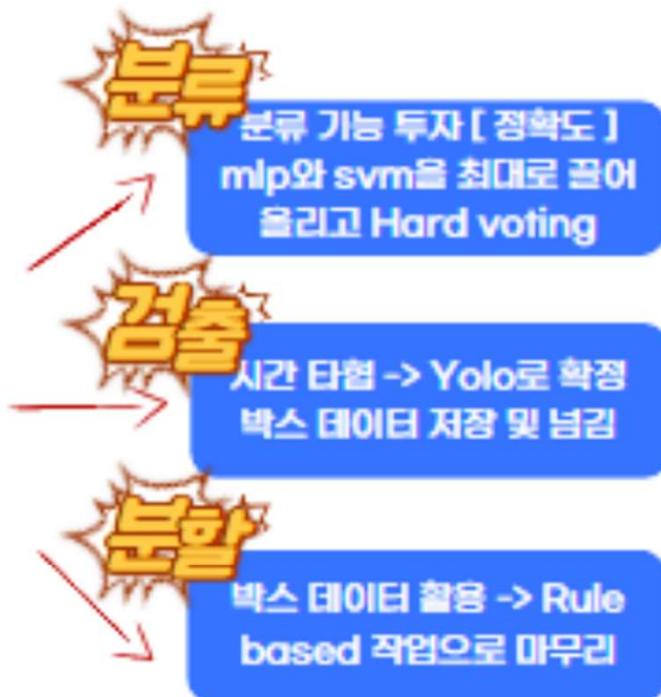
- Deeplabv3 성능 문제로 아웃
- F3net, Face_recognize 비공개
- U -net 으로 직접 모델 학습 -> 시간 부족

접근 방식

- 다른 접근 방식이 필요
- 아래는 yolov8 -seg로 임시 test

문제 해결

문제 1,2,3,4 -> 해결



문제 사항

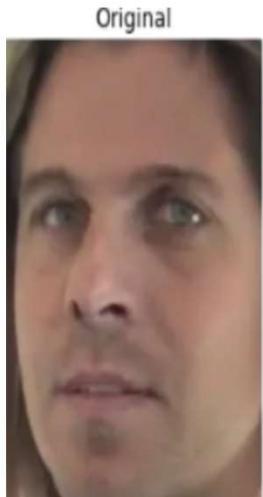
- 분류 성능 저하 + LSTM 오류
- 시간 부족 + 분할은 다른 접근이 필요

접근 방식

- 분류에 올인 -> 분류를 애초에 정확하게 하면 검출 분할에서 이중으로 하지 않아도 괜찮
- 검출은 빠른 Yolo로 통일
- 분할은 검출에서 얻은 box 데이터 활용 rule-based 모델로 전환

문제 해결

문제 1 [분류 성능], 문제 3 [LSTM 기능 문제]



```
print("Manual Hard Voting Classification Report (==2)")  
print(classification_report(y_val, strict_pred, target_names=['REAL', 'FAKE']))  
  
...  
20/20 - 0s - loss: 0.0101 - auc: 0.9967 - val_loss: 0.0890 - val_auc: 0.8774  
Epoch 100/100  
20/20 - 0s - loss: 0.0094 - auc: 0.9973 - val_loss: 0.0770 - val_auc: 0.8944
```

	precision	recall	f1-score	support
REAL	0.92	0.73	0.81	15
FAKE	0.94	0.98	0.96	65
accuracy			0.94	80
macro avg	0.93	0.86	0.89	80
weighted avg	0.94	0.94	0.93	80

문제 접근

- MTCNN with augmentation
- epoch 증가 -> 과적합이 안나오게끔만
- mlp + svm -> hard voting [mlp]

문제 해결

문제 2 [시간 부족], 문제 4 [Segmentation]

File: ahbweevwpv.mp4
Actual: FAKE
Pred: FAKE (0.69)



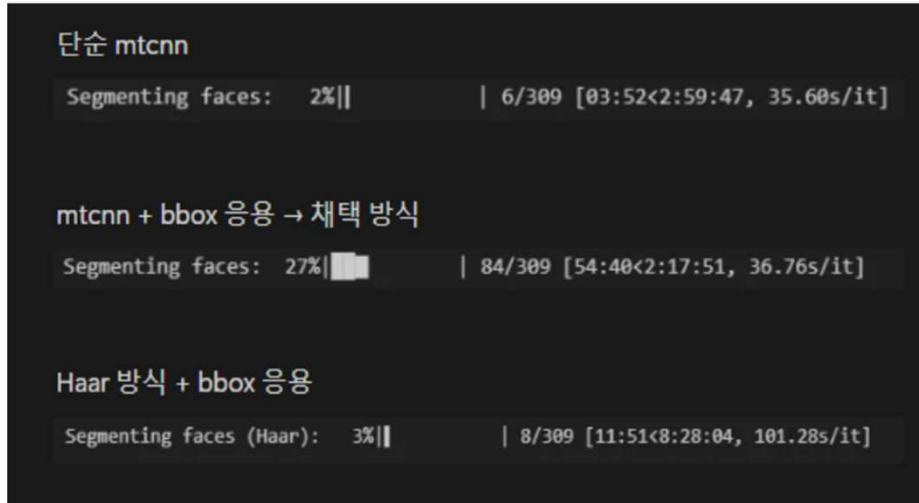
문제 접근

- YOLOv8로 빠르게 박스 처리 -> 전체 프레임 말고 평균 일부만 뽑아서 박스
- SAM -> mask 방식 [원한 의도가 아님]
- segmentation -> 박스처리 내부 얼굴 색칠



문제 사항

문제 5

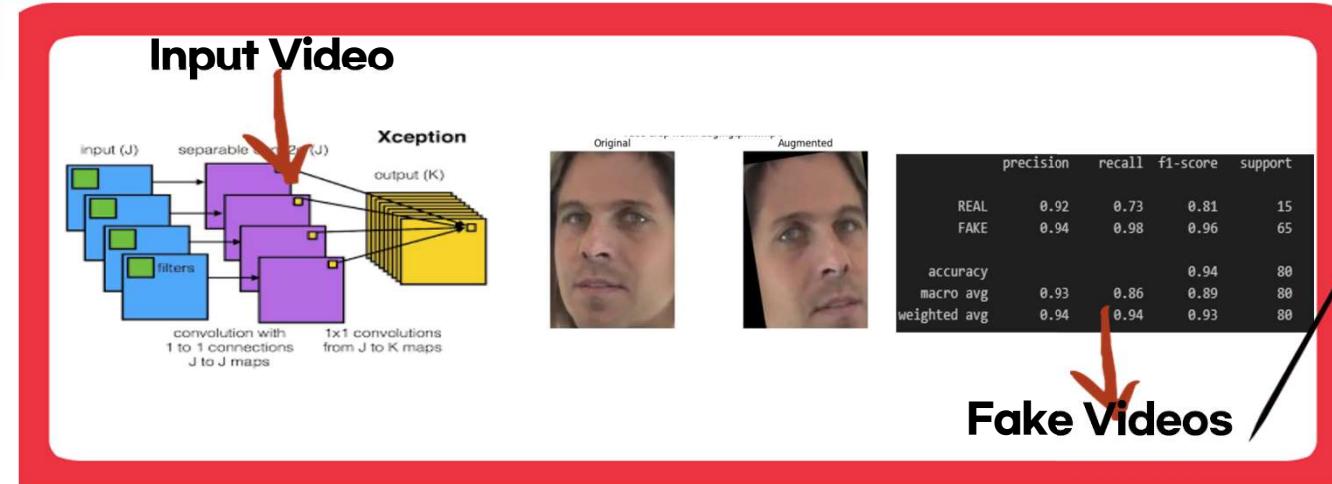


문제 사항

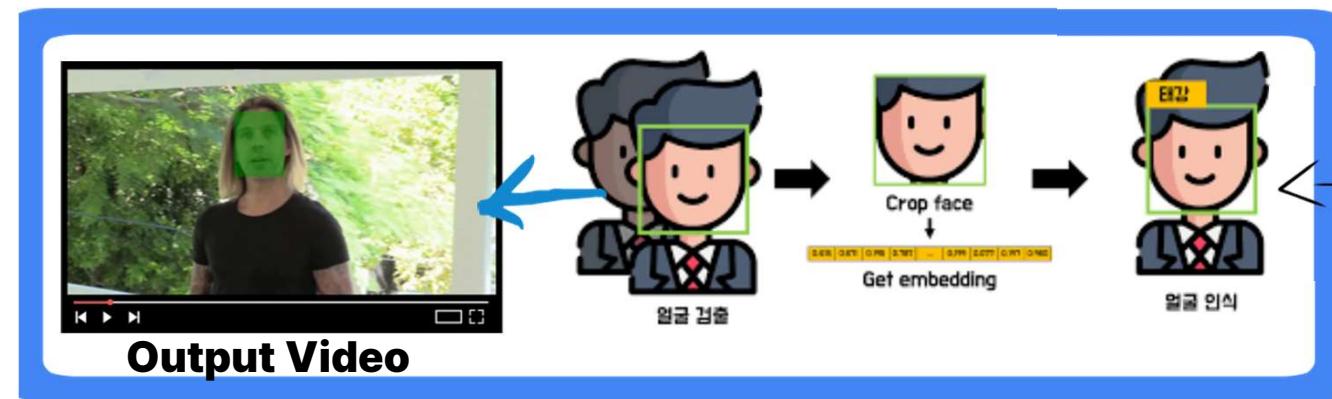
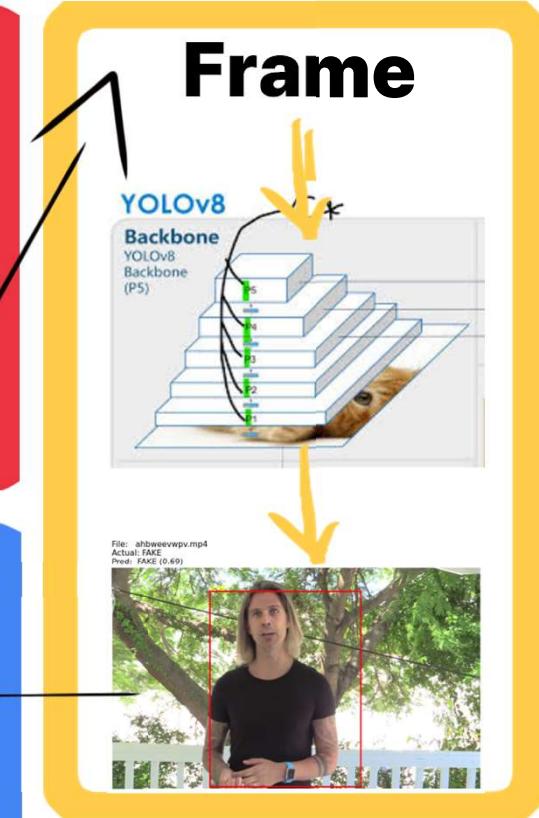
- 박스 처리 + 안에 사람 있음 -> 바로 rule-based segmentation
- 박스 안에 사람이 없음 -> mtcnn 적용 -> 얼굴 찾아서 segmentation 처리

결과 요약

1. Classification



2. Detection



3. Segmentation

기타

느낀 점과 확장 가능성

느낀 점

- 개발을 진행할 수록 **기획이 정말 중요한 단계**라는 것을 느낌 -> 확실치 않으면 **문제 발생**
- 개발 과정에서 충분히 진행 과정을 갈아 엎는 과정이 흔할 수 있겠다 생각이 들음
- 결과적으로 **문제 해결 능력 -> 대응 중요**

아쉬운 점

- 시간이 부족해서 생략한 것이 많았음 = Detection 비교, Learning Segmentation
- Kaggle 데이터 셋 제출 기한 만료 -> 만든 모델의 성능 분석 기회가 없었음

가장 기억에 남는 순간

- 시간이 부족하다 보니까 과정에서 **효율적으로 빠르게** 수행할 수 있도록 고민하고 적용하는 부분이 가장 인상깊음

추가 예상 기능

- Learning Segmentation 부여 -> 자체적으로 분류에 의존하는 것이 아니라 segmentation으로 한번더 조작 여부 판단 [픽셀 단위] -> 교차 검증도 가능
- GPT api를 이용해서 조작 검증 가능

참고자료

참고문헌

- [1] 홍진, 컴퓨터 비전과 딥러닝 강의자료, 채용연계SW캠프 인공지능과정
- [2] Preda, G., "Deepfake Starter Kit," *Kaggle Notebook*,
<https://www.kaggle.com/code/gpreda/deepfake-starter-kit>

2025/06/20 (금)

THANK YOU

박우빈