



Senior project 3

Short Track AI Highlight Editor

5분반

팀5 : 김예찬, 김동민, 박민규, 이동섭

Table of Contents



01 Preview

02 Improvements

03 Github

04 Marketing plan



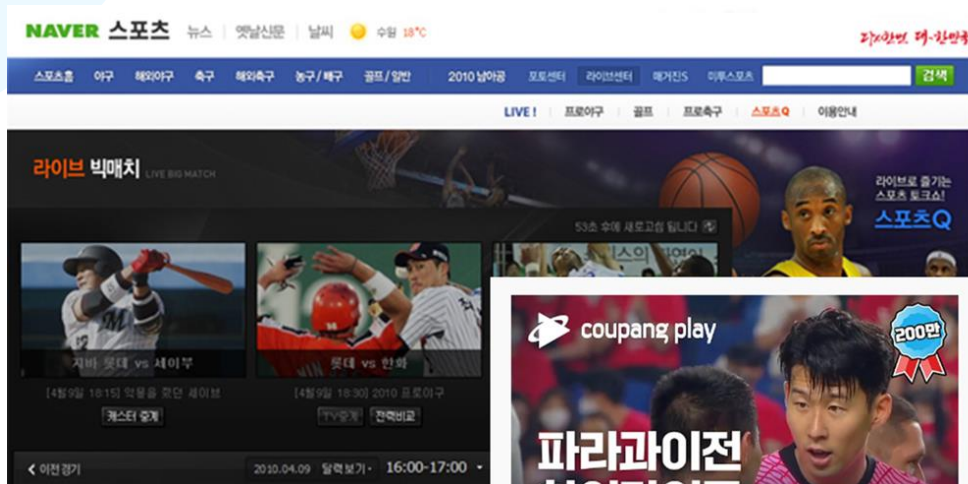


01

Preview



Our Project



파라과이 전 하이라이트 | 2022 축구 국가대표 평가전 대한민국 vs 파라과이 하이라이트 | 디지...
조회수 270만회 · 3일 전

쿠팡플레이 Coupang Play

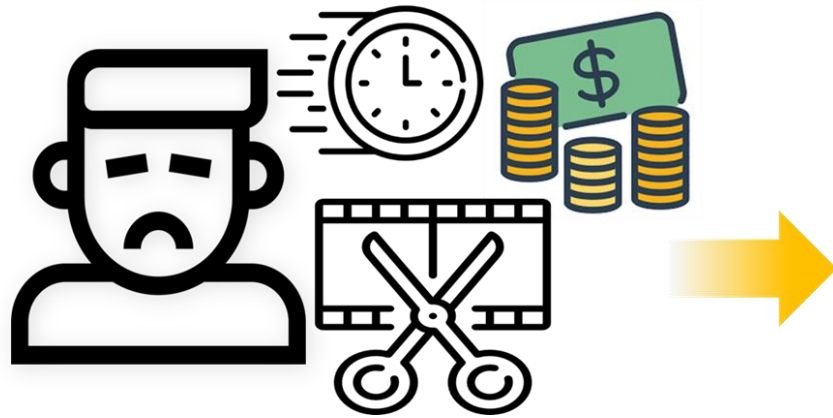
대한민국 2 - 2 파라과이 2022 축구 국가대표 평가전 하이라이트 더 보고 싶다면?
지금 쿠팡플레이에서 확인하기!

새 동영상

Demand of Sports highlight is growing

People want to see rapid highlights after games

Motivation



AI Highlight Editor

Cost a lot of time, labor, money to edit sports videos

Reduce the cost of editing highlights

Idea description

Create a highlight video by analyzing various scenes in full short track video.

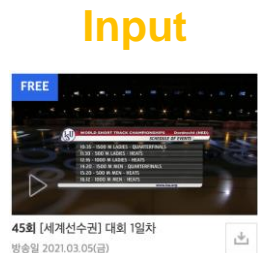
➡ Rank change, falling, passing the final/finish line, caster's rising voice tone and keywords extract.



1	KOR	52	HWANG D.
2	CAN	16	S. DUBOIS
3	ROC	3	S. ELISTRATOV
남은 바퀴 수 5		LAP 8.92	



System Architecture



Short Track
Video

MoviePy

Frame
01

Frame
02

Frame
03

Frame
04



Object Detection



Google Cloud
Vision API

OCR



Google Cloud
Speech API

STT

Score
analysis

XGBoost
classification

Output

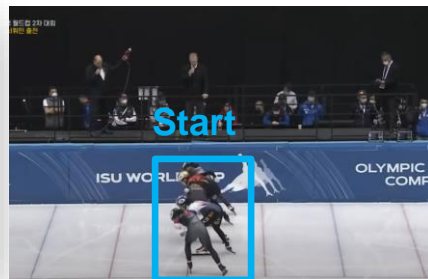
Highlight
Video

Object Detection



Yolov5 model

- A total of 2000 images labeling
- Trained 4 objects – skating, fall_down, start, finish



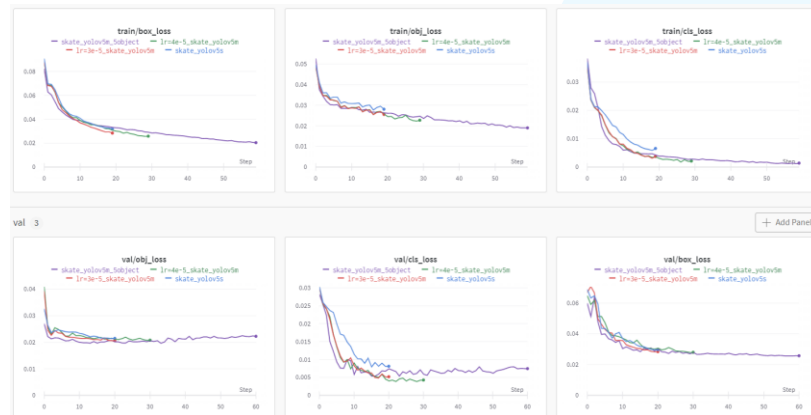


Object Detection

- Hyperparameter tuning with WandB.
- Visualized best trained model and epoch.



Weights & Biases



Validating runs/train/skate_yolov5m_5object/weights/best.pt...

Fusing layers...

YOLOv5m summary: 212 layers, 20869098 parameters, 0 gradients, 47.9 GFLOPs

Class	Images	Instances	P	R	mAP50	mAP50-95: 100% 7/7 [00:03<00:00, 1.76it/s]
all	110	281	0.93	0.939	0.949	0.673
fall_down	110	33	0.937	0.908	0.944	0.553
skating	110	187	0.89	0.947	0.949	0.604
start	110	15	0.975	0.933	0.936	0.789
finish	110	36	0.891	0.905	0.919	0.517

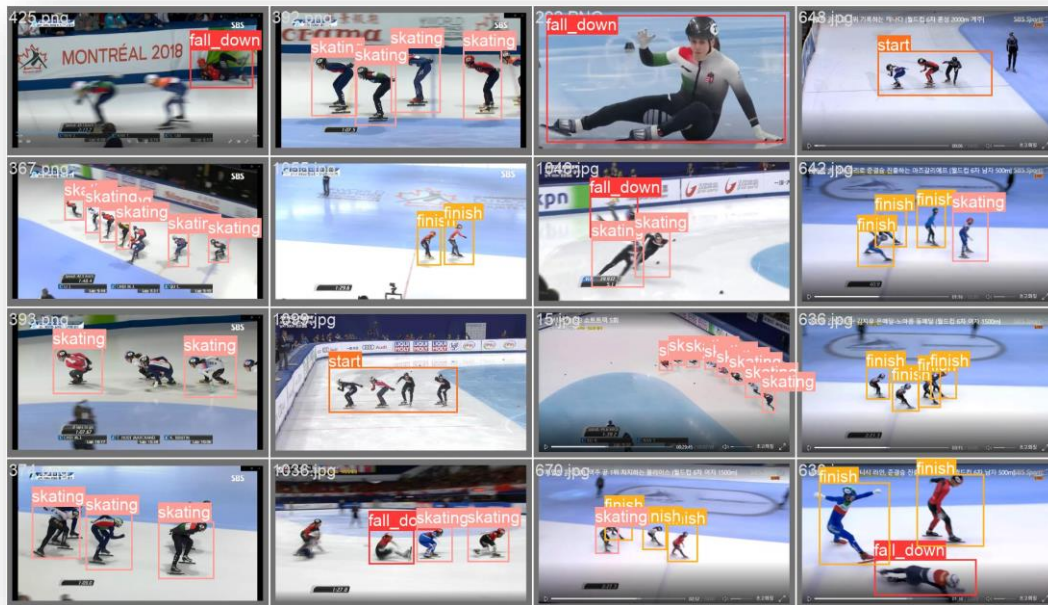
Precision on test dataset : 0.93





Object Detection

- Prediction on test dataset



OGR (Optical Character Recognition)



```
ranking: ['1. K. BOUTIN', '2. L. van RUIJVEN', '3. KIM A.L.']  
img keyword Set: { '결승', 'lap:', '1500m', '준결승', '여자', '500m' }  
lap time: ['2:05.7', 'Lap: 9.52', 'Lap: 9.68', 'Lap: 9.62']
```

OGR (Optical Character Recognition)

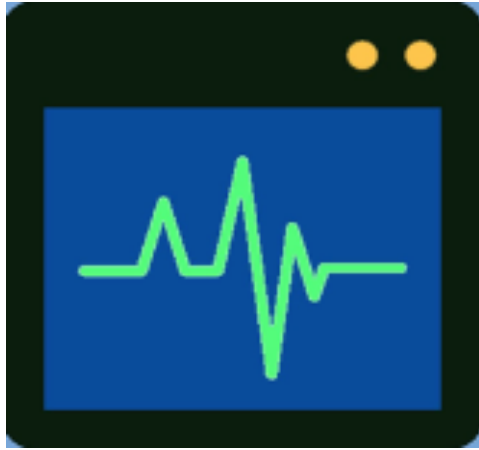


OGR (Optical Character Recognition)

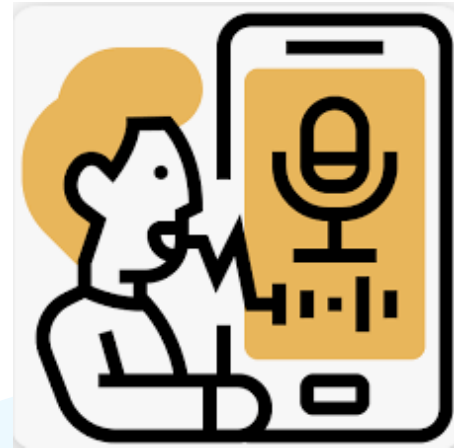


- Check the last lap with the **Unofficial Result**.
- The faster the **Lap time**, the higher the score.

voice recognition



Frequency

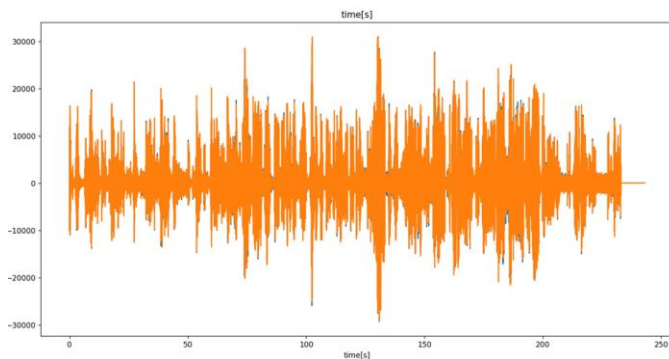


Speech-to-text

voice recognition

Measure the **amplitude** of the **frequency** to distinguish the excited sound of the caster.

➡ Extraction of highlights through the characteristics of spectator and caster sound.



Frequency extraction

```
주파수 28000이상  
[74.0, 102.0, 130.0]
```

More than 28000

voice speech-to-text



Google Cloud
Speech API



```
word_time = [] #키워드 시간초 저장
word_time2 = []
word_time3 = []

keyword = [ "캘링", "조심", "열키는", "의식", "찬스", "인코스", "자리", "연결", "첫번째 주자", "선두", "결승전", "가장먼저", "1번주자"
            "안전하게" ] #키워드 1점

keyword2 = [ "결과", "도움", "잘들어왔", "마킹", "말어", "잡아당겼", "빠르게", "에이스", "추격", "출발합니다"
            "좋아", "좋습니다", "넘겨줍니다", "타이밍", "추격", "두바퀴", "치고 나갑니다." ] #3점

keyword3 = [ "출발", "충돌", "마지막", "추월", "넘어졌습니다", "넘어짐", "파이널", "결승선 통과", "부딪", "레디", "스타트", "넘어"
            "한바퀴", "마지막 주자", "파이널랩", "결승진출", "미끄러", "마지막 코너" ] #5점
```


Video **Frame** Processing



Divided



Highlight



Merge



Criteria for selecting highlight elements



- Asked expert and ice skating organization for advice.



First lap, Finish moment



Falling down



Overtake

2015-2018	국가대표 상비군
2015	회장배 전국남녀소트랙 1500m 1위
2016	전국동계체전 1500m 3위 / 3000m relay 1위
2017	31회 중별종합대회 Super final 3위
2018	99회 동계체전 1000m 2위 / 3000m relay 1위
2019	제 33회 중별 종합 3000m Super final 2위
2019	제 36회 전국남녀 1000m 2위 / 3000m 1위
2020	단국대학교 국제스포츠 학과 졸업
2020	스포츠 건강 트레이너 1급
2020	스포츠 재활 트레이너 1급
2020	스포츠 마사지 체형관리사 1급
2020	국제 키조요가 지도자 1급
2019	영어회화능력평가시험 ESPT 자격증

**Former National Team substitute
Short track speed skating player
김윤선 선수**

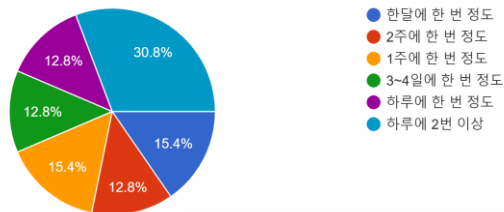


Criteria for selecting highlight elements

- Survey on short track community

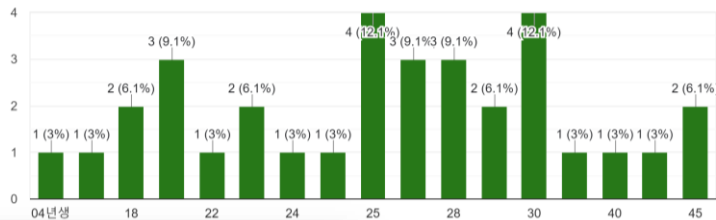
평소에 쇼트트랙을 얼마나 보시나요?(횟수)

응답 39개



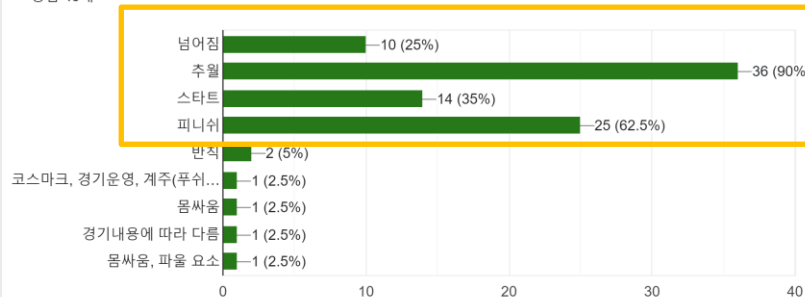
나이

응답 33개

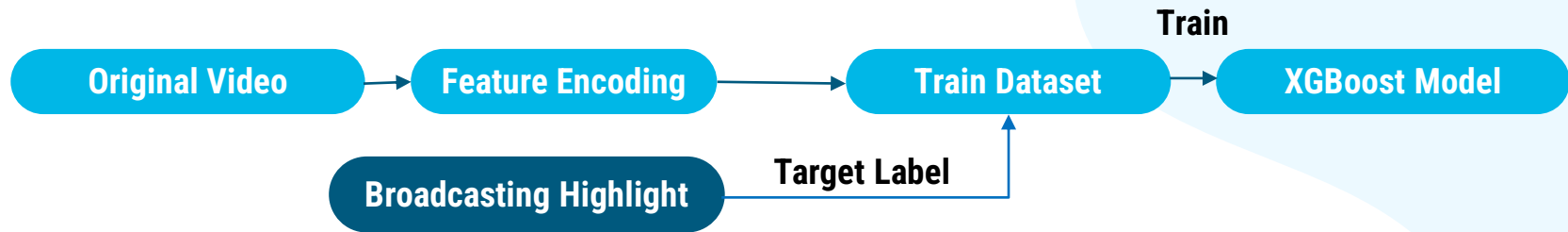


쇼트트랙 영상 제작(시청)시 중요한 점들! (하이라이트 요소)

응답 40개



XGBoost Algorithm



- **Example train dataset**

fall_down	start_finsh	rank_change	keyword	frequency	Label
0	1	0	1	0	1
1	0	0	1	0	1
0	0	1	0	0	0
0	0	1	0	1	1





02

Improvements

Improvements



STT

time: 4
Transcript: 조금씩 물어 가고 있습니다
time: 42
Transcript: 우리 일본 선수들이 우리 선수를 반박할 때마다 큰 도음이 될 수 있습니다 지금 빠져 있기 때문에 또 어떤 작용을 할지도 지켜봐야 될 것 같습니다
time: 51
Transcript: 김다경 일본
time: 58
Transcript: 강하게 FC 하는게 중 합니다
time: 65
Transcript: 일본 하게 하기 위해서는 강하게 pc를 받는게 중 합니다
time: 100
Transcript: 최하위까지 떨어져 있는 상태에서 다시 한번 기회를 꼭 봐야겠습니다 면 아래쪽부터 이제 시작을 해야 하는데 중국도 3일 밀려났습니다 나 지금 둘 두 바꿔 아직 있습니다
time: 107
Transcript: 아직 안 오네 괜찮아요 해서 박인우
time: 133
Transcript: 내가 많이 주지 않았으면 좋겠는데 이번엔 좋겠습니다 이철 때일수록 침착하게 서로를 믿으면서 경기를 풀어야 합니다 두 계단을 올라가는 것이 중요한 급하게 할 필요는 없고요 통화를 할 텐데요 그런 공간을 잘 보고 있어야 돼요 올라왔습니다
time: 136
Transcript: 박지원
time: 170
Transcript: 네덜란드 만만치 않군요 어느 쪽으로 통을 볼지 7 받아줬습니다 김다경 마지막 회 있으니 구요 중국 이들 중 있는데 강해 얼마나 빠른발걸음 버릴 올라옵니다 지금 내 스포드가 아주 좋아요
time: 173
Transcript: 이제 청소기가 들렸고
time: 208
Transcript: 네 중국 선수들이 몰라서 못 하고 있습니다 이문에서 박지원 내일 좋아요 하고 있습니다 이제 금방 들어왔습니다 그대도 지치고 있습니다 마지막 주입니다 대한민국 결석으로 정말 심장이
time: 223
Transcript: 지금 계속해서 손주가 바뀌었거든요 내일 벗어도 만만치가 않았어요 운전해 해서
time: 243

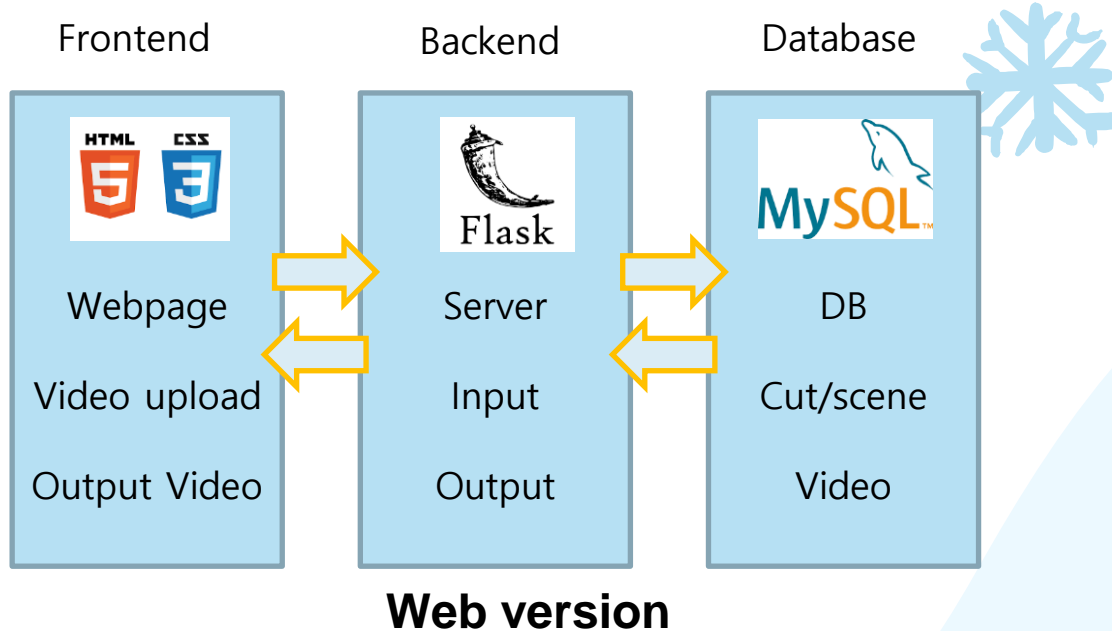
Detect terminating language
"~니다", "요 "



Keep the caster's commentary

Improvements

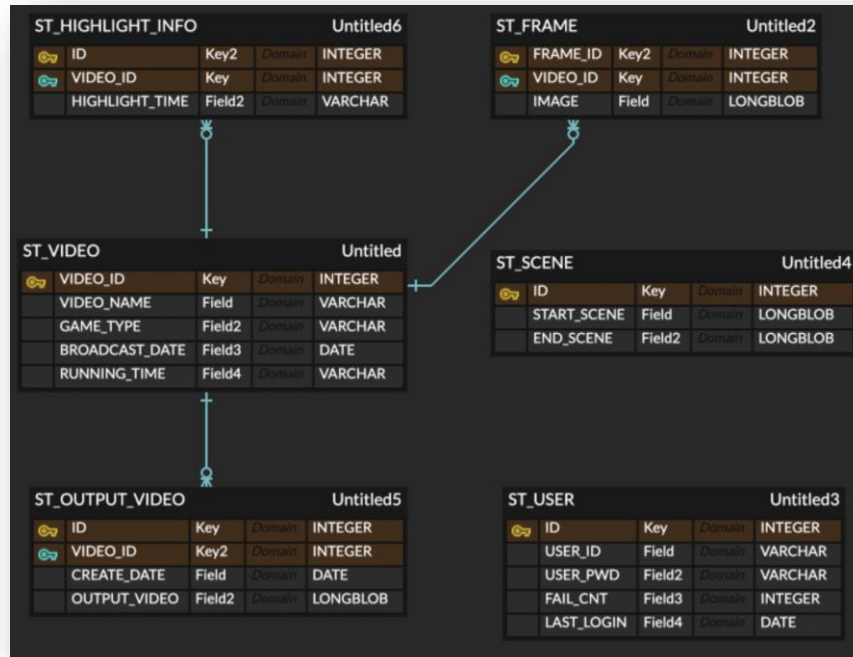
- Deploy existing python programs to the web version to improve user convenience.



Improvements



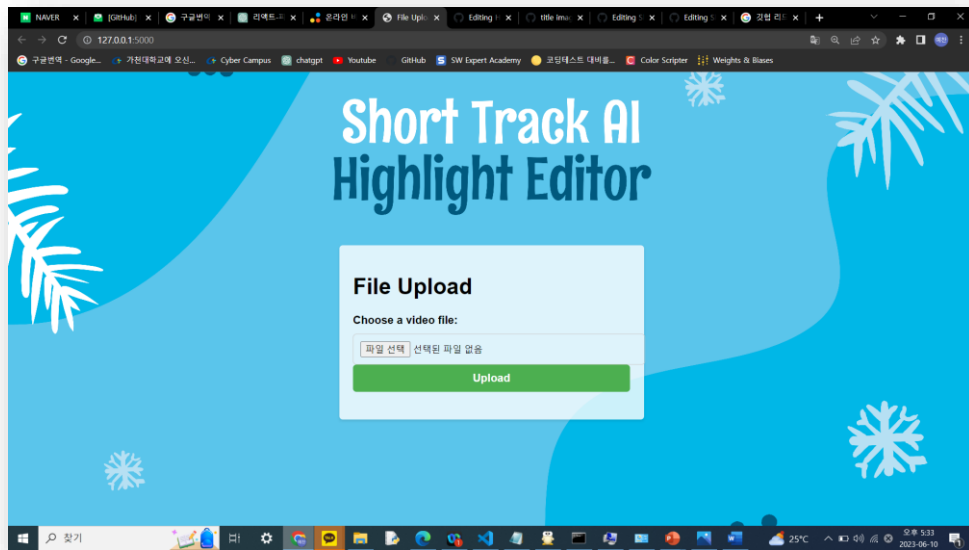
- The database was constructed using mySQL with the following structure.





Improvements

- Currently using local server due to cost.
- The web page was created in conjunction with js to configure the web page so that it could be operated on the local server.



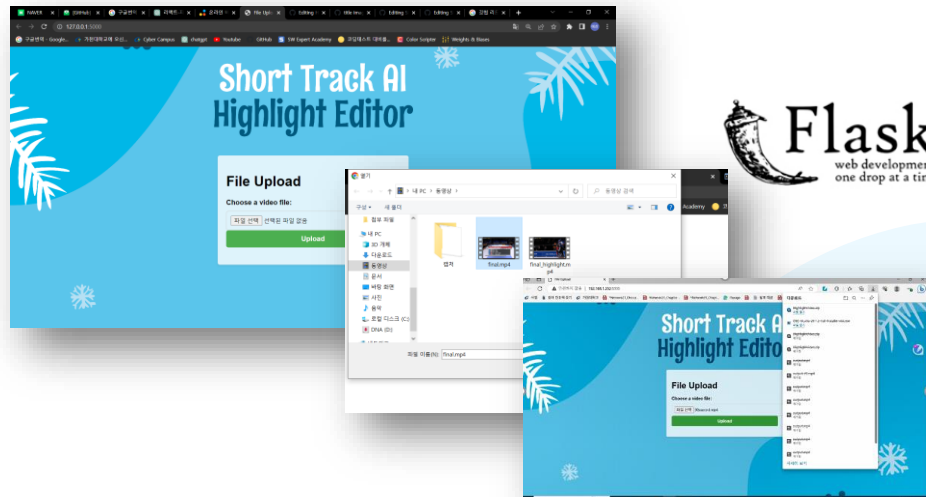


Improvements

- Two versions were created so that users could use the desired type according to their preference.



1. Python program with GUI



2. Web version

Demonstration video



1. Local Sever





03

Github

Github Readme



Link: https://github.com/GCU-Team5/ShortTrack_AI_Editor

ShortTrack_AI_Editor

This is an AI short track highlight editor using object detection, OCR, STT.

ShortTrack AI Highlight Editor

Generates highlights in short track game videos.

YouTube Link: <https://www.youtube.com/watch?v=7C35u7T2D88>

Motivation

- With the spread of smartphones and the Internet, the demand for sports highlights is growing.
- In addition, users want highlights of their favorite sports events to be uploaded more quickly right after the game.
- Immediately after the game, editing highlight videos quickly takes a lot of time, labor, and money, so we thought of developing an AI highlight editor to reduce these costs.

Installation

- \$ git clone https://github.com/GCU-Team5/ShortTrack_AI_Editor.git
- \$ python startUI.py

User Interface

Select Video Button

- The highlight video is extracted by clicking the select video button and selecting the video to be edited.

Close Button

- You can set the closing screen.

System Architecture

Short Track Video → Object Detection → OCR → STT → Highlight Video

Object Detection

ISU WORLD CUP

OCR

ISU WORLD CUP

STT

Github WIKI



Home

Kim Y.C edited this page 6 minutes ago · 15 revisions

Welcome to the ShortTrack AI Editor Wiki!

Abstract

With the spread of smartphones and the Internet, the demand for sports highlights is growing. In addition, users want highlights of their favorite sports events to be uploaded more quickly right after the game. Editing highlight videos immediately after the game takes a lot of time, labor, and money, so we thought of developing an AI highlight editor to reduce these costs. We've developed the program within 3 modules which are Object Detection, STT, OCR.

Team Structure

Total Team Size: [4 members]

Role	Name	Email
Frame edit, GUI	Dong Min Kim	thaha424@gachon.ac.kr
Object Detection	Yae Chan Kim	yckim19@naver.com
STT module	Min Gue Park	kokocom0113@gachon.ac.kr
OCR module	Dong Seop Lee	dongxieli@kakao.com

System Architecture

Edit New page

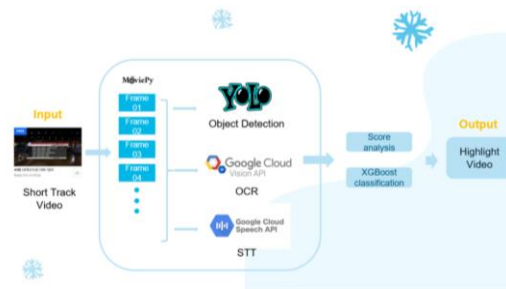
Pages 1

Find a page...

Home

- Abstract
- Team Structure
- System Architecture
- Used technologies
- 1. Object Detection
- Prerequisites
- Installation
- 2. OCR (Optical Character Recognition)
- Google Cloud Vision API
- 3. STT
- Required Libraries
- Functions
- Main Function (STT_detection)
- 4. Frame editing
- Video Division
- Video Merging
- Highlight Extraction
- How to Use

System Architecture



The structure of our implemented program is above. If you put the original short track video as an input, the video is divided into frames and then these frames are passed to the object detection, OCR, and STT modules. Afterwards, each module stores and returns highlight element scores for each frame, and the scores are combined in the score analysis part. Also, after onehot encoding the results of the module in the XGBoost part, whether it is a highlight event is classified and predicted as 0,1. Afterwards, the results of these two parts are combined to determine the final highlight frame index, and the frames are combined to make the final output video.

Github WIKI



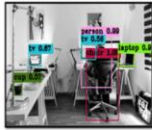
Used technologies

1. Object Detection

Object Detection



Deep learning based
object detection model



In the existing YOLO model, there is no information about the short track related object.

Trained 4 objects (Custom)
A total of 2000 images lab

The object detection is used to detect specific objects(Skating/
Torch). This code allows you to specify a target object to detect
Torch library for deep learning and image processing.

Prerequisites

- Python
- PyTorch
- TensorFlow
- YOLOv3

Installation

To use the ShortTrack object detection code, follow these steps:

1. install Python if it is not already installed.
2. install the required Python packages by running the following command:
3. Clone the YOLOv3 repository by running the following command & git clone https://github.com/yolov3/yolov3.git
4. Change the yolo3 directory to yolo3.
5. Download the pre-trained YOLOv3 model weights by running the following command:
6. Define the target object and detection threshold. I set 'target' as 'target object' & threshold.
7. interpret the results.

The scores variable contains the object detection scores for each frame in the video. Each row represents the score of the target object in the corresponding frame. You can process the scores further based on your requirements.

Notes: Make sure to have the required dependencies installed before running the code.

The code assumes that the ShortTrack videos are stored in the 'dataset' directory. You can modify the code to work with your own dataset.

The code uses a pre-trained YOLOv3 model (yolov3) for object detection. If you want to use a different model, you can modify the code accordingly.

The code currently supports the following target objects: 'skater', 'start', 'finish', 'target'. You can add more target objects by modifying the 'target' variable.

The code also supports the following target objects: 'skater', 'start', 'finish', 'target'. You can add more target objects by modifying the 'target' variable.

Feel free to experiment with different target objects and thresholds to achieve the desired results.

2. OCR (Optical Character Recognition)

OCR (Optical Character Recognition)



Google Cloud Vision API

OCR (Optical Character Recognition) using the Google Cloud Vision API. The Google Cloud Vision API is a REST API that extracts text from images and analyzes the extracted text to find specific keywords. The following libraries are imported:

- 'glob' for file handling.
- 'cv2' for image processing.

Google Cloud Vision API

OCR (Optical Character Recognition) using the Google Cloud Vision API. The Google Cloud Vision API is a REST API that extracts text from images and analyzes the extracted text to find specific keywords. The following libraries are imported:

- 'glob' for file handling.
- 'cv2' for image processing.
- 'google.cloud.vision' for the Google Cloud Vision API.

The function 'set_recognizer' is defined. It takes a parameter 'client' which represents the number of frames to process.

The Google Cloud Vision API credentials are set using the 'set_recognizer' function.

A client for the Google Cloud Vision API is created using 'vision.ImageAnnotatorClient()'. The code uses the 'glob' module to get a list of image file paths in a specific directory.

A loop is initiated to process each image.

- 'The image file is read and its content is stored in the 'image' variable.
- An instance of 'ImageAnnotator' is created using the image content.

The 'detect_text' function is called with the image to perform OCR and extract text annotations.

The extracted text is stored in the 'text' variable.

- The content is processed and relevant information such as lap times, rankings, and other keywords are extracted and stored in variables.
- A score is calculated based on the extracted information, and the score is added to a list.

After processing all the images, the function 'ranking_change_tracker' is called to calculate the score for any ranking changes that occurred during the race.

The final score list and the top 10 list indicating whether a ranking change occurred in each frame are returned.

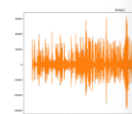
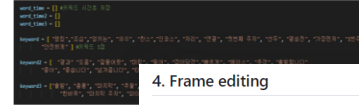
In the main block, the 'set_recognizer' function is called.

You need to have the required libraries installed and provide valid Google Cloud Vision API credentials.

Make sure to replace the placeholder 'YOUR_PATH' in the code with the actual path to your credentials file.

3. STT

voice speech-to-text



Required Libraries

The code imports the following neo

4. Frame editing

Video Frame Processing



Divided



Highlight



Merge

Video Divided, Merge, and Highlight Extraction

Frame editing part divides a video into frames, merges the frames back into a video, and extracts highlights from the video.

Video Division

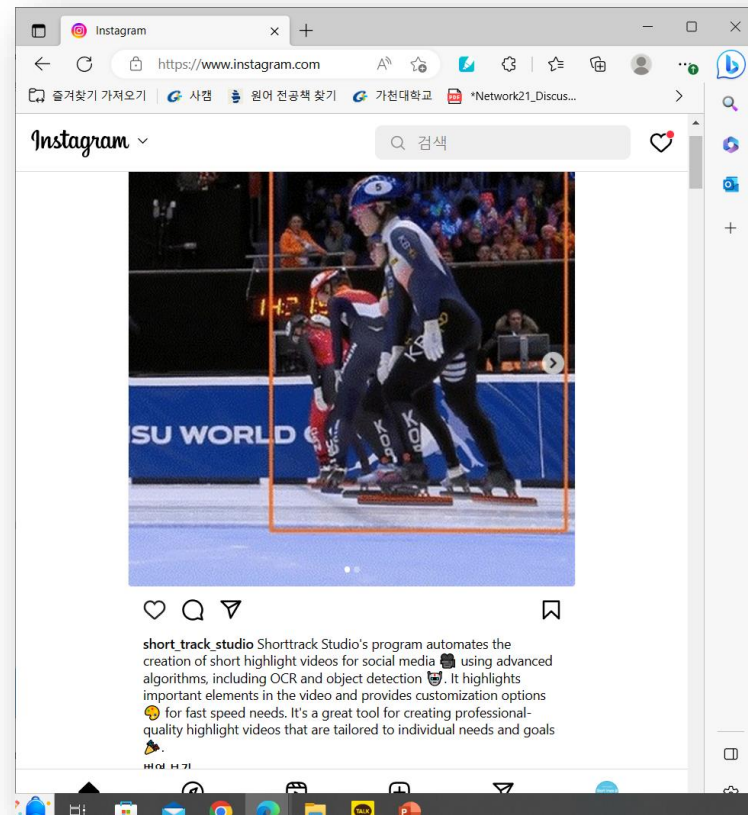
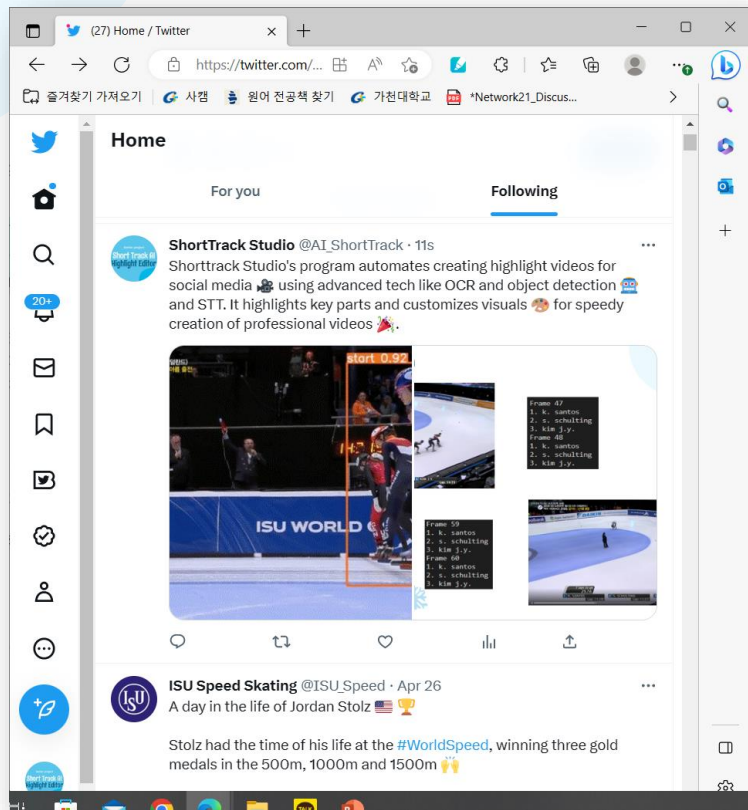
The first step is to divide the video into frames. This is done by calling the 'divide_video()' function. The 'divide_video()'

04

Marketing plan



SNS Marketing



SWOT Analysis

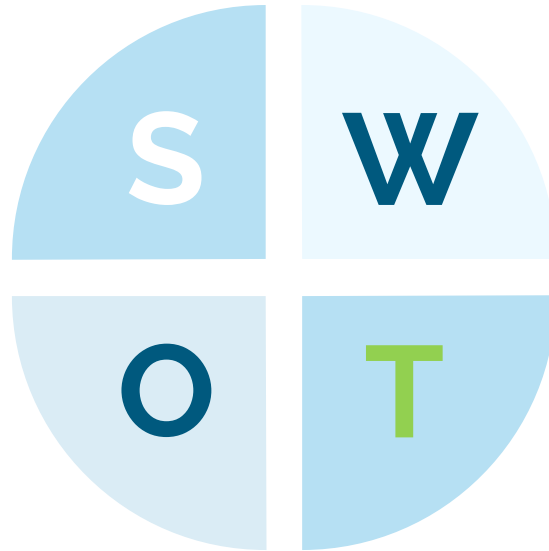
STRENGTHS

Similar program X
Low price

Monopoly on the market

Iceberg Federation Marketing

OPPORTUNITIES



WEAKNESSES

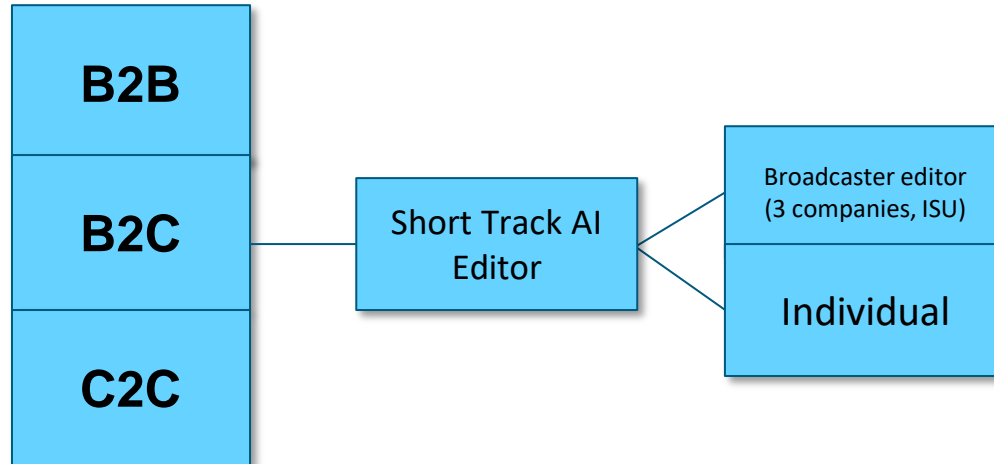
Incomplete accuracy
Highlight extraction time

Copyright infringement

illegal sharing

THREATS

STP : Segmentation



STP : Targeting



Short track editor
“Tired of too many video editing”



Athletes and fans who want
Highlight videos



STP : Positioning



Individual & Athlete

"My own video"

- Edit and extract videos that I want
- Edit and extract your own video

Broadcaster editor



YouTube editor

"Get various videos quickly and cheaply."

- Edit and extract videos quickly at an affordable price
- Editing and extracting many videos during the Olympics



Member Role



김동민

UI,
Frame split/merge
Backend
Marketing
Github-wiki



김예찬

Object Detection
Frontend
Server
Github-wiki



박민규

Speech To Text
Frontend
Marketing
Github-wiki



이동섭

OCR
Backend
Database
Github-wiki



Thank You!

Do you have any questions?

Senior project Team 5

https://github.com/GCU-Team5/ShortTrack_AI_Editor

