# 2020 Spring Capstone Design Proposal

*"LOL Highlighter"*


| TEAM | MALTEMERS |
|---|---|
| **20145822** | 김영현(Yeonghyeon Kim) |
| **20143606** | 이혜성(Hyeseong Lee) |
| **20164891** | 한윤진(Yoonjin Han) |

# 1. Introduction

Since the adoption of the Asian Games E-sports event in 2018, interest in E-sports has been increasing. It is also gaining popularity in Korea as the Ministry of Culture, Sports and Tourism gives permission to set up stadiums.

Among them, League of Legends (LOL) has the largest fans worldwide, and the impact of the game is increasing globally. Most LoL matches offer replay services. The edited highlights are also has high popularity and views.

However, there are too many E-sports competitions held worldwide to meet all of these demands. Judging and editing all the highlight clips takes a lot of manpower. In addition, edited highlights cannot contain all of the scenes that the user wants. Therefore, there is the hassle of having to check the video, which is an average length of about one hour, to find the desired scene.

Accordingly, we felt that we needed a service that analyzes the game video to find the desired scene or highlight, then planned this project.

# 2. Goal of the project

We will create a web application, called LoL highlighter, that receives a League of Legends match video as input and automatically provides highlights.

When the user uploads the video URL or file of a game, the user will be provided with a timestamp of the highlights of the game.

Subsequently, we will either provide the edited highlight video to the user, or categorize the highlights as kill scenes, object team fights, etc. so that the user can pick them up.

# By Modules

## 1. Sound Analysis

1. Audio extraction: We can extract audio from any video we want (100%)
2. Sound feature detection: Get audio features we need using MFCC (100%)
3. Pattern definition: Defines a pattern that can be judged as a highlight.
4. Sound pattern matching: With defined pattern, check the highlight sections for user input. (80%)
5. Data Visualization: Visualize audio data as info graphic. (100%)

## 2. Video Analysis

1. Video-to-Image processing: The video should be completely refined into a set of available images. (100%)
2. Video frame OCR: Get text-data and detect data changes, such as KDA or match time, by using OCR on video frame. It corresponds to the red part in the figure below. (100%)
3. Image changes detection: Detect changes some important objects, such as dragon, in the game. It corresponds to the yellow part in the figure below.


**3**. User service via web application

1. Provide web service without terminal.
2. User POST video url and GET highlight clips.

# By Time

## By Mid-term Demonstration

1. Demo program will show the similarity between the existing highlight video and our derived highlight sections.

## By Final Demonstration

1. Categorize video clips more specifically by analyzing video frames.
2. Demonstrate the program on the Web

# 3. Theoretical basis

- **Audio File Analysis - MFCC**

  Using MFCC(Mel-Frequency Cepstral Coefficient) for Audio File Analysis (Beritelli, Francesco & Grasso, R.. (2009). A pattern recognition system for environmental sound classification based on MFCCs and neural networks. 1 - 4. 10.1109/ICSPCS.2008.4813723.)

  - **FFT(Fast-Fourier Transform)/ IFFT(Inverse FFT)**

    FFT is used to perform the audio signal spectralization, which is expressed by the sound pressure according to frequency. Through this, the strength of each frequency is determined and the audio height and size are analyzed..

  - **Cepstral Analysis**

    Cepstral Analysis is used to isolate the degree of synthesis of a particular frequency band called formants from the spectrum and to identify the degree of speech synthesis at that point..

- **OCR(Optical Character Recognition)**

  - **Pattern Recognition**

    OCR can recognize characters in scanned images based on text data in various fonts and formats.

  - **Feature Detection**

    OCR can detect characteristics such as the number of angular lines, intersections, or curves of text.

# 4. Project description

**Project outline**

This project is an application that analyzes LOL game videos to find highlight scenes or specific game situations, and provides them with refined contents for users to watch.

The user can receive a desired scene from the analyzed video by inputting a link of the video.

**Project detailed function**

| **Functions** | **Details** |
|---|---|
| Extract audio from video | Extract the audio that can be analyzed from the YouTube video link that is input by a user. |
| Audio Analysis | Obtain an audio metadata through MFCC.<br>Determine the highlights part features with our own algorithm. |
| Video Screen Analysis | Using OCR open source to determine whether a specific event has occurred. |
| Highlights categorizing | Based on the analyzed audio and video image data, the highlights can be categorized through filtering by specific scenes(Player's kill catch scene, team fight scene, object fight scene, etc.) |
| Time Stamp | Provide users with a clickable list of the video's viewpoint in time for highlights. |
| Selection of the Highlights | The sections that can be called the highlights of the video are comprehensively judged, selected and provided.Selection criteria are based on the following criteria. * |

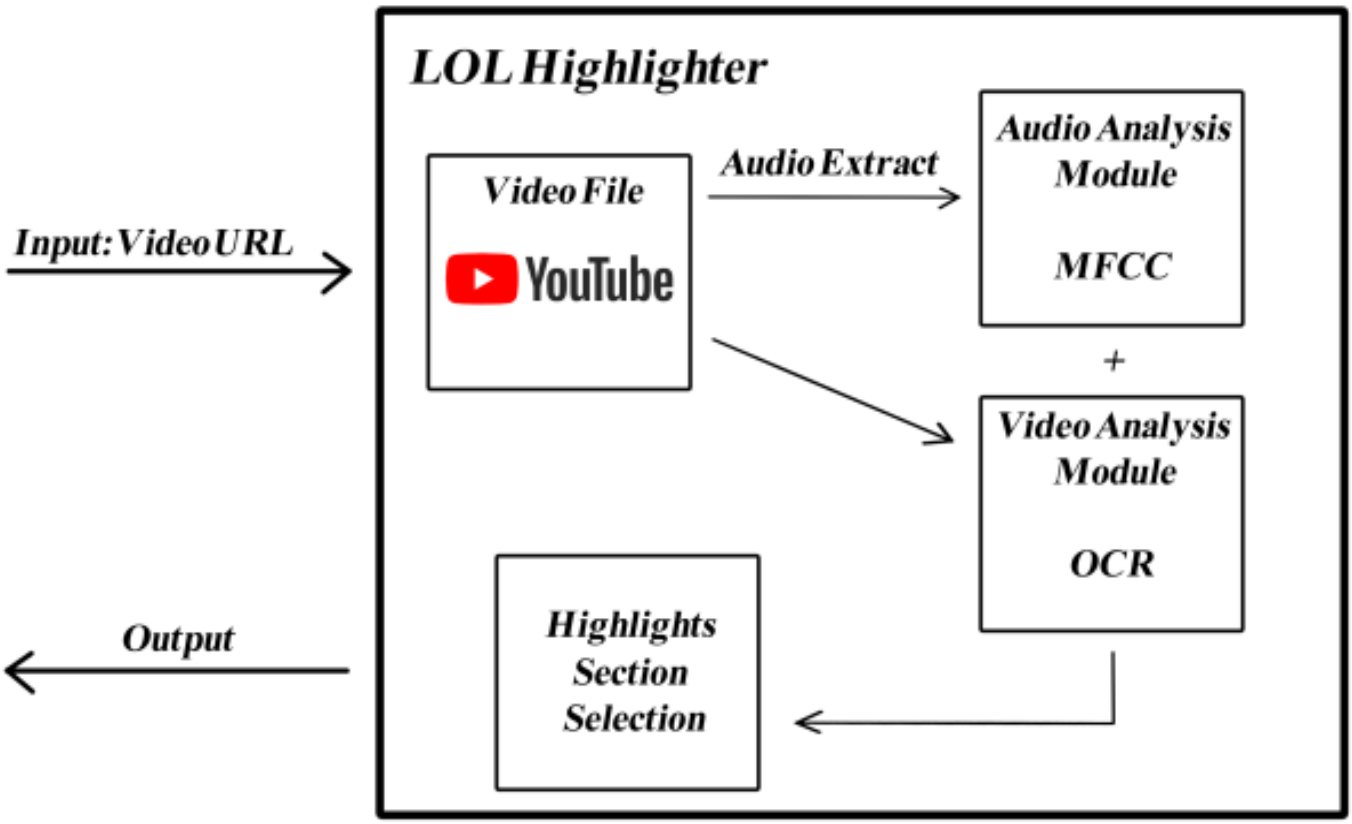
* Selection criteria for the highlights

1. Ban & Pick scene before commentator's comment at the beginning of the game.
2. The sound pattern of the commentator in the highlight scene analyzed through audio analysis.
3. Scenes of changes in key indicators(kills, objects, etc.) identified in video analysis.
4. Scene of the end of the game before the commentator's "GG" sound.

* Highlights Categories

1. Ban & Pick scene
2. Kills
3. Fight scenes for the objects such as towers, dragons, baron, etc.
4. Other Super Plays

# 5. Design and Implementation of project




- **Sound Extraction**
  - ○ Extract sound from the received link's YouTube video using *python-ffmpeg/pytube*

- **Sound Analysis**
  - ○ Audio analysis will be conducted by Sound pattern recognition algorithms. We will analysis audio extracted from already-made LCK(League of legends Champions Korea) highlight video. So, We find certain patterns of highlight videos, and check the similarity between the pattern and a input video. And then it can select the main section of input video based on the pattern.

- **Video Analysis**
  - ○ Finds the number of kills, number of tower demolition on LCK commentary description screen using OCR, and checks the change of section to find when the event occurs. And highlight sections related to objects(e.g. Dragons, Baron Nashor, etc. ) also can be found by checking changes of icons on the video UI.

- **Categorization selected highlight sections**
  - ○ Aggregate the above analysis results and classify them by category, presenting the user with a highlight timeline.

- **Implementation technology stack**: Node.js + Vue.js + Electron

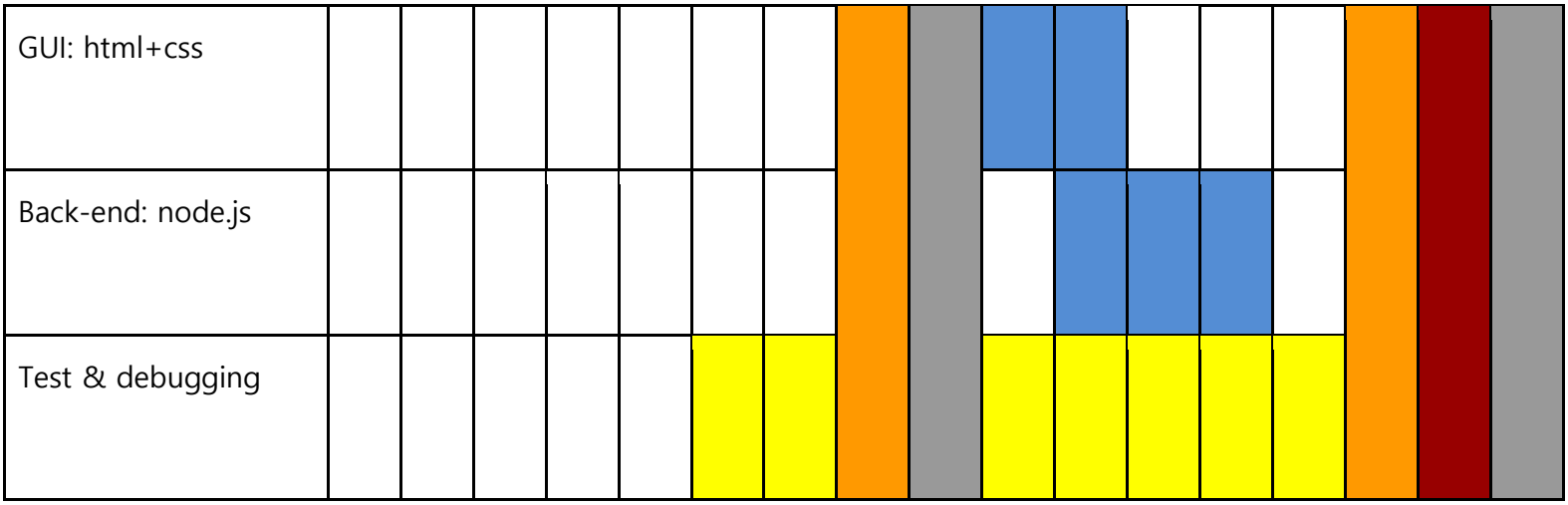
- **Github activity**: https://github.com/ycs1m1yk/LoLHighlighter
  - ○ We will commit all unit developments by detail.
  - ○ Comment and solve every issue as much as we can.

**Member role**

| | |
|---|---|
| 김영현: | OCR, Audio Extraction, MFCC Pattern Analysis |
| 이혜성: | Sound Data Visualization, Back-end, API |
| 한윤진: | Front-end, Sub-Image searching algorithm, Video-to-Image processing |
| Common: | Implementation algorithm of high-light section recognition, Research for MFCC and related theory |

# 6. Time schedule

| | | 0 Week | 1 Week | 2 Week | 3 Week | 4 Week | 5 Week | 6 Week | 7 Week | 8 Week | 9 Week | 10 Week | 11 Week | 12 Week | 13 Week | 14 Week | 15 Week | 16 Week |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Submission a proposal | | | | | | | | | 중간데모 | 중간고사 | | | | | | 최종데모 | 최종리포트제출 | 기말고사 |
| Modification of a proposal | | | | | | | | | | | | | | | | | | |
| Research | | | | | | | | | | | | | | | | | | |
| Audio processing | Audio Extraction | | | | | | | | | | | | | | | | | |
| | Data Visualization | | | | | | | | | | | | | | | | | |
| | Pattern Analysis | | | | | | | | | | | | | | | | | |
| Video processing | Screen OCR | | | | | | | | | | | | | | | | | |
| | Video-to-Image processing | | | | | | | | | | | | | | | | | |
| | Sub-image Searching | | | | | | | | | | | | | | | | | |


GUI: html+css
Back-end: node.js
Test & debugging

# 7. References

Github Open-source Softwares

1. Video Crawling
    - pytube(https://github.com/nficano/pytube)
        - Pytube is a lightweight, dependency-free Python library for downloading YouTube Videos. We will use this open-source library for downloading YouTube video and audio that is corresponding to the url given by the user.

2. Audio Analysis
    - librosa(https://github.com/librosa/librosa)
        - Librosa is a python library for audio and music analysis. This open-source library provides the functions to extract and analyze the audio such as MFCC(Mel-Frequency Cepstral Coefficient). We decided that this open-source library could be used for analyzing the pattern of MFCC to suggest the criteria for highlighting and comparing that criteria to user input for highlighting in this project.

3. Optical Character Recognition Tool
    - pytesseract(https://github.com/madmaze/pytesseract)
        - Pytesseract is a python wrapper for Google Tesseract. Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images. We can use this open-source library to recognize the number of scoreboards on the broadcast screen.

4. Image Processing Tool
    - pillow(https://github.com/python-pillow/Pillow)
        - Pillow is a PIL(Python Imaging Library). The PIL adds image processing capabilities to your Python interpreter. The PIL provides image archives, image display, image processing. In this project, we can use this library for detecting changes some important objects in the game broadcast.

Reference Documents and Academic Papers


- E-sports statistics (https://influencermarketinghub.com/esports-influencer-marketing-next-big-thing/)
- 2019 LOL WORLDS statistics (https://www.cnbc.com/2019/04/14/league-of-legends-gets-more-viewers-than-super-bowlwhats-coming-next.html)
- LoLOCR (https://github.com/teovoinea/LoLOCR)
- MFCC theory (https://brightwon.tistory.com/11)
- A pattern recognition system for environmental sound classification based on MFCCs and neural networks. 1 - 4. 10.1109/ICSPCS.2008.4813723. (Beritelli, Francesco & Grasso, R.. (2009).