Capston Design (2)

Proposal report

|  |  |
| --- | --- |
| Submit Date | 2019.03.17 |
| Team | Team 8 (LAJI) |
| Members | 20160237 Jisu An |
|  | 20133096 Hyunjae Lee |
|  | 20143954 Jinwoo Jeon |
| Prof | Sangoh Park |



Introduce

Streamers re-process broadcasted videos to extract highlights. In this process, the role of the editor becomes important, because the edited video is uploaded to YouTube, the largest video portal in the world. This process takes a lot of time and effort and small, unknown streamers can’t make such kind of highlights because of the time and effort.

We propose a service that automates this process so that highlights can be easily extracted without hiring editor to do so.

Background

Personal broadcasting services such as YouTube, Twitch, and AfreecaTV have gained popularity in recent years. Streamers communicate with viewers in real-time comments and send out a various content. One of them is **Game Broadcasting**.

Twitch is a platform where people who specialize in game broadcasting are gathered. It is loved by many viewers since they give much better experience about quality of image than other platforms like YouTube.

BM(Buisiness Model)

Streamers upload their own broadcasted video to our website. When processing is done, they should pay by ‘clip’ which is our own currency to download the re-processed video. The prices vary by size of the uploaded video(not the highlighted one).

Functions

Users need to login our website using Google or Twitch account since we need to check which video is his. We provide https connection to secure the information of the users.

We only provide game video highlighting service which is broadcasted by English. Also the chat should be English too.

When the upload is finished, users have 2 options.

1. delay: Users can set the delay time that will surround the appointed time that we suggest.
2. face detection: If streamer shows his/her face on the screen, we can give them a better highlight since we will analyze the mood(expression) of the streamer.
3. analyze: We analyze the video with chats and face of the streamer. Laugh, anger, sadness and stillness after sigh will be the benchmark of our software.

After the video is processed, user will receive the notification mail that contains the link leads to the download page. User can choose:

1. Download the video that is processed recently.
2. Re-download the video that is previously processed.
3. Directly upload the video to YouTube.

Expected UI

Login form

(Social media)

UI(1) - Login

Security(https)

Image from video

UI(2) : Main

Upload

Delay(Int)



Analyze

Face Recognition ▣

Speech Recognition▣

Chat Recognition ▣

UI(3):Download& History

Download

Rendered Video (1)

Rendered Video (2)

Rendered Video (3)

Rendered Video (4)

How we do it (Expected)

We are expecting to make highlighted video by following steps written below:

1. We make some clustering samples to analyze the frequency of word that appears in real-time comments(chats). To make better clustering sample, we accumulate the data and update the model. The chat data will be obtained by using API that Twitch provides.
2. We capture the face of the streamer to find out whether he/she is upset, happy, sad etc. These data make our model find better list of the expected time points.
3. We also analyze the sound to find the expected points since streamer shouts or say nothing after sigh when kind of funny thing happens. If there’s a lot of chat log at that time, it is highly the moment that should be included in the highlight.

Structure of our service

Operating system: ubuntu 16.04

Web server: NginX

Web Framework: Django

WSGI MiddleWare: uWSGI

Database server: PostgreSQL

Security: http over ssl (https)

Server specification :  AMD Ryzen 2400g (our own)

Weekly Plan

Week3: Setting up the server (Nginx setting, Domain Assignment(bind9), Appling https)+ Find videos for test (Twitch/Game/English)

Week 4: Social Login, User database schema setup, Encrypting DB, Connect DB and Web

Week 5 : Video upload/download implementation, study how to use twitch Chat API

Week 6 : Implement designating face camera area and UI design for it + Make algorithm for highlight automation

Week 7 : 1st Demo - Login/uploading + downloading video/Twitch CHAT API result

Week 8 (No class) : Algorithm implementation and test using chat

Week 9 : Algorithm implementation and test using face expression and sound

Week 10 : Algorithm implementation and optimization

Week 11 : Implement pay service using Toss API + Add algorithm to the service

Week 12 : Notification mail implementation + Service testing

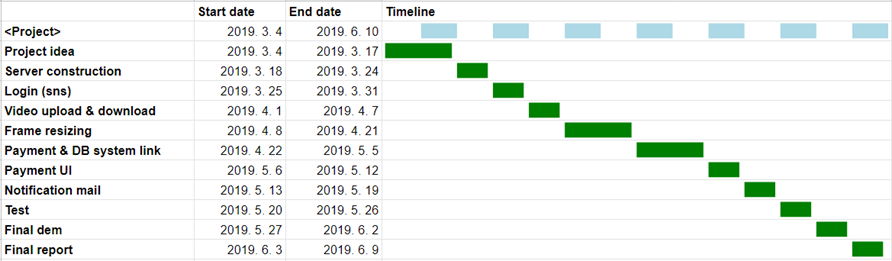
Week 13 : Refining service and error handling + Final demo prep.

Week 14 : Final Demo

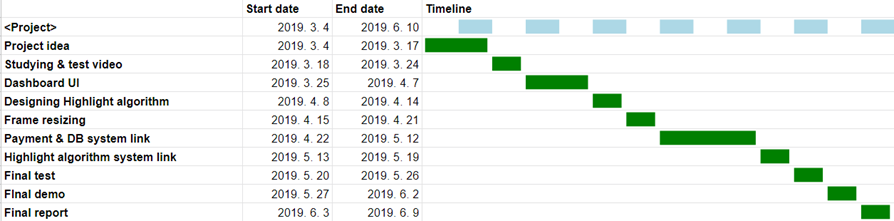
Week 15 : Report and Manual

Individual weekly plan

Jisu An,



Hyunjae Lee,



Jinwoo Jeon,

