MOOCS RECOMMENDER BASED ON LEARNING STYLES

COMPLEX VIDEO PRODUCTION STYLE CLASSIFICATION

Software Requirement Specification
Project ID: 19-089

Pathirana P.H.P.S.L.

IT16 0043 82

Bachelor of Science Special (Honors) in Information Technology
Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology
Sri Lanka

Date of Submission: 2019-05-13

MOOCS RECOMMENDER BASED ON LEARNING STYLES

COMPLEX VIDEO PRODUCTION STYLE CLASSIFICATION

Project ID: 19-089

Hilmy S.B.M.

Supervisor: Mr. Nuwan Kodagoda

Co-Supervisor: Ms. Kushnara Suriyawansa

Date of Submission: 2019-05-13

DECLARATION

Name	ID	Signature
Pathirana P.H.P.S.L.	IT16004382	

Supervisor: Mr. Nuwan Kodagoda Signature:

Co-supervisor: Ms. Kushnara Suriyawansa Signature:

1 Introduction

1.1 Purpose

The purpose of this Software Requirement Specification is to provide a detailed overview of requirements for "analysis and classification of complex Massive Open Online Course (MOOC) video production styles" component of our MOOCREC V2 research project. This document is intended to the customers of the product and the developers who take part in developing, testing and maintaining the software. It describes about the project's target audience and its features, user interfaces, hardware and software requirements. It defines how the clients, team and other stakeholders see the product and functionality.

1.2 Scope

Primarily, the scope of this document pertains to the analysis and classification of complex MOOC video production styles. It focuses on the customers of the product and the developers who take part in developing, testing and maintaining the software.

1.3 Definitions, Acronyms and Abbreviations

MOOC	Massive Open Online Course
SRS	Software Requirement Specification
GPU	Graphic Processing Unit
RAM	Random Access Memory
CNN	Convolutional Neural Network

1.4 Overview

The main goal of the application to find the user's learning style and suggest the most appropriate and suitable MOOC course based on the learning style and the goal of "analysis and classification of complex MOOC video production styles" sub component is to classify different types of video styles used in the course and to calculate the composition of each video production style when there are more than one video production styles. [1]

This SRS will cover the functional and non-functional requirements of the analysis and classification of complex types of MOOC video production styles of the MOOC Recommender application. This is described in the document by three chapters.

The first chapter contains the purpose, scope and the overall description of the component. The second chapter provides an overall description of the software product, a comparison of the product with similar existing systems and also contains the interfaces, constraints and the operations performed by the system.

2 Overall Descriptions

MOOC resources include multiple modalities such as lecture videos, audio transcriptions, slides, textbooks, forum discussions and clickstream log data. Among them, lecture videos are arguably the central and omnipresent component for knowledge transfer, to which other data modalities support.

Generally, the types of MOOC videos include: talking head, slides, coding, animations, writing (khan academy) etc. Some videos may contain multiple production styles, e.g.: switching from a talking head to a slide or switching from a slide to an animation. When a MOOC video contain more than one video production style, composition of each production style will be calculated as percentages. Then the production style of the video will be determined by the average value of those percentages.

2.1 Product Perspective

Class Central, My MOOC and MOOCREC V1 are some applications in the same domain as MOOCREC V2. None of these applications contain a MOOC recommender based on video production style except MOOCREC v1. But it only works only for slides, coding and talking head video production styles. In MOOCREC V2, we are going to implement a MOOC video classifier which can identify more complex and mixed video production styles of MOOC videos.

2.1.1 System Interfaces

- Database connectivity Interface
- Web Scrapper Interface

2.1.2 User Interfaces

This component doesn't contain any User Interface since it is a backend process.

2.1.3 Hardware Interfaces

- A computer with a GPU with a graphic memory of 2GB or greater.

2.1.4 Software Interfaces

- Python 3.7
- Keras (Deep learning library)
- Google Tensorflow

2.1.5 Communication Interfaces

- An Internet connection with at least 60Mb/s speed.

2.1.6 Memory Constraints

- Computer should contain at least 6GB of RAM.

2.1.7 Operations

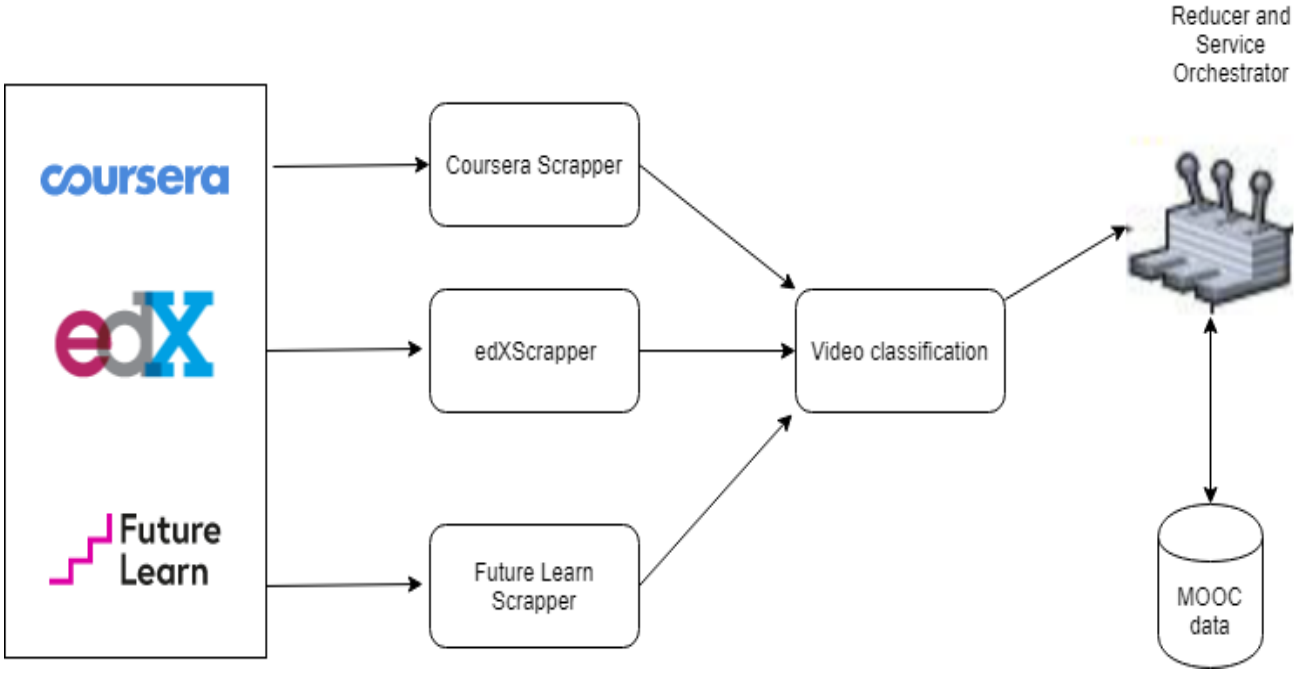
No operations are required by the user because video classification is carried out in the backend.

2.1.8 Site Adaption Requirements

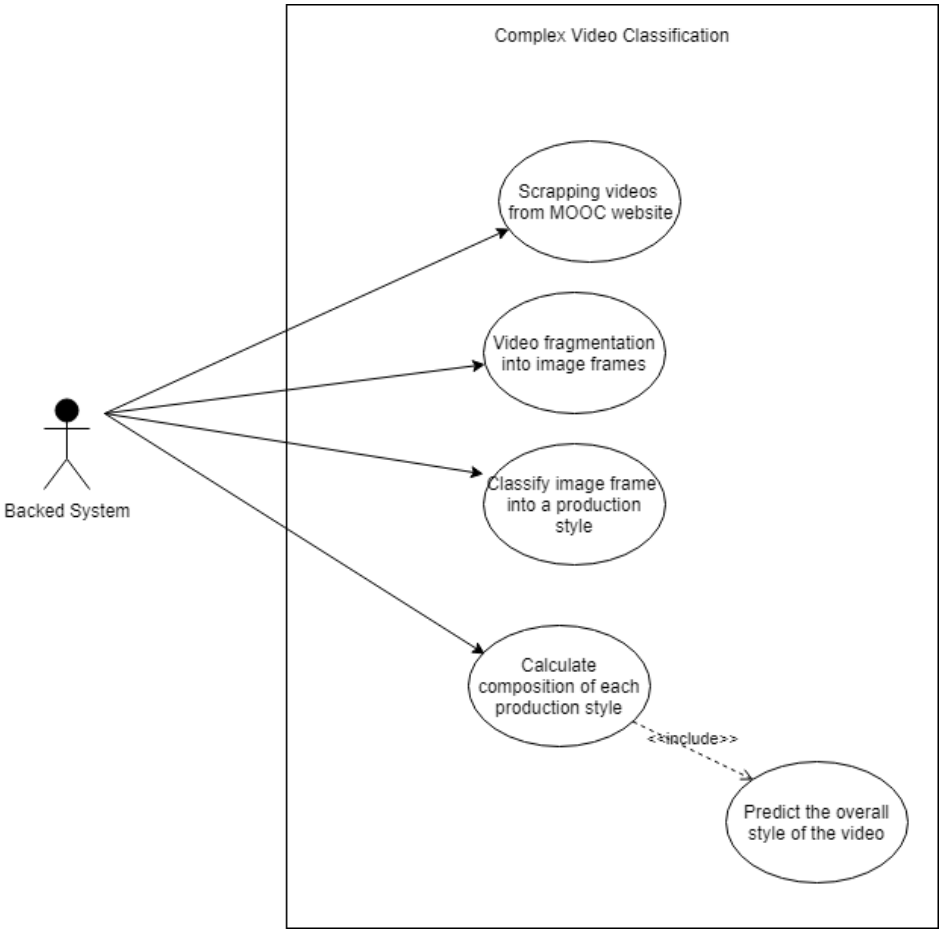
No site adaption requirements are required since this is a backend process, and this doesn't directly connect with the end user.

Product Functions

2.1.9 Overall Design



First videos will be gathered from MOOC websites (Coursera, edX, Future Learn) and then those videos are fragmented into image frames. Those image frames/keyframes are passed into trained CNN model. Then neural network will automatically learn the features of the image and prediction will be given.



2.1.10 **Use Case Scenarios**

Use Case Name	Extracting Videos from the source
Pre-Condition	Videos should be available in the source
Post-Condition	Extracted Video should be available in the database
Actor	Web Crawler - Backend System
Main Success Scenarios	1. Go to desired webpage. 2. Login to the page using credentials. 3. Gather video content. 4. Save content in the database.
Extension	1a. Connection Issues. 1b. Invalid content accessed.

Use Case Name	Splitting video files in to Image frames
Pre-Condition	Videos should be available in the database
Post-Condition	Image frames should be available after splitting
Actor	Video splitter – Backend System,
Main Success Scenarios	1. Select the desired video. 2. Split the video into consecutive image frames from the beginning to the end of the video.
Extension	1a. Processing an invalid video

Use Case Name	Classification of Image fragment into a style
Pre-Condition	Image frames should be available after fragmentation
Post-Condition	Image frames should be categorized into the correct video production style
Actor	Image classifier – Backend System,
Main Success Scenarios	1. Select the image frame. 2. Pass the image through CNN 3. CNN automatically classifies the image into a video production style
Extension	1a. Processing an invalid image frame 1b. Processing a corrupted file

2.2 User Characteristics

A user of this component should be a professional in software development and he should have experience in python coding, deep learning algorithms and tools, CNN etc.

2.3 Constraints

Video style classification is a heavy computational task and it requires heavy computational power.

- 6GB of RAM
- Python is the implementation language
- Windows 8 or higher
- TensorFlow as the computational framework
- Keras – Neural network library written in python

2.4 Assumptions and Dependencies

- An established Internet connection with at least 60Mb/s speed.
- Hard disk storage space of 500GB.
- A computer having Nvidia or AMD GPU with 2GB of GPU memory.

2.5 Apportioning of Requirements

2.5.1 Essentials Requirements

- Collecting videos from MOOC platforms Coursera, edX and Future Learn.
- Split selected video into image frames.
- Pass image frames into CNN model and classify each frame into video production style.
- Calculate the composition of each production style and then determine the composition of whole video.

2.5.2 Desirable Requirements

- Implementation of additional web crawlers to gather MOOC videos from wide number of MOOC sources.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

Video classification is a backend process and it doesn't contain any User Interfaces.

3.1.2 Hardware Interfaces

- At least 6GB of RAM
- An Internet connection with at least 60Mb/s
- A computer with a GPU would be advantageous when training the neural network rather than using a normal CPU. Mathematical heavy functions can be carried out fast by GPU's than CPU. It might take a week to complete a task by a CPU and GPU only might only take a day for same task.

3.1.3 Software Interfaces

TensorFlow: It is an open source library for numerical and large-scale machine learning. TensorFlow bundles together a slew of machines and deep learning (aka neural networking) models and algorithms and makes them useful by way of a common metaphor. It uses python to provide a convenient front-end API for building applications with the framework, while executing those applications in high-performance C++.

Keras: It is library written in python used in building machine learning models. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML.

3.1.4 Communication Interfaces

An Internet connection with at least 60 Mb/s is required for the web crawlers to gather data and to use cloud services.

3.2 Functions

3.2.1 Extracting videos from MOOC websites

Description	Extracting videos from MOOC sources using web crawler.
Sequence of Operations	1. Initialization of web crawler 2. Extract video from the source 3. Store extracted video in the database.
Validity Checks	Desired video should be downloaded.
Input	MOOC Video Details (URL, video title, …)
Output	Extracted MOOC video
Error Handling	If extraction of the video from the source fails, application will keep retrying for several times and if it fails, video source is considered as inaccessible or removed from the source.

3.2.2 **Splitting video into frames**

Description	Splitting video into frames
Sequence of Operations	1. Select the video 2. Split the video into smaller parts 3. Split small videos into image frames
Validity Checks	Correct video selected
Input	Video (Size, Format)
Output	Image Frames
Error Handling	If the fragmentation of video fails retry for several times and log the error so the developers can identify the bug and debug it.

3.2.3 **Classification of fragmented Image into a production style**

Description	Classification of a fragmented image into a production style
Sequence of Operations	1. Select the image fragment to classify 2. Pass the selected image frame into CNN model 3. CNN determines the production style of the image frame
Validity Checks	Correct Image frame selected, and correct style of the image determined by the CNN model
Input	Image
Output	Video production style

3.2.4 **Determine the overall composition of the MOOC video**

Description	Determine the overall composition of the MOOC video
Sequence of Operations	1. Get the addition of each video style in image frames. 2. Calculate the percentage of each production style in the video. 3. Production style with the highest percentage is considered as the production style of the video
Input	Production style of each image frame
Output	Production style of the video

3.3 Performance Requirements

- This component should be able to gather a 5-minute video from MOOC source by taking less than 2 minutes.
- A 5-minute video should be able to split into frames form less than 20 seconds.
- Prediction of composition of 5-minute MOOC video should not take more than 30 seconds.

3.4 Logical Database Requirements

3.4.1 Data Format

MongoDB is used as the database and JSON format is used to represent a style of a video.

MOOC Video

1. Video Id
2. Video Name
3. URL
4. Video Style
5. Created Date

3.4.2 Data Accessibility

End user will not be able to access the data directly.

3.5 Design Constraints

3.5.1 Standards Compliance

The web crawlers should comply to the web crawling policies of the MOOC platforms.

3.5.2 Data Constraints

All data must be represented in JSON format so that all modules of the system can interpolate.

3.6 Software System Attributes

3.6.1 Reliability

The system should be available 24X7 with reliability around 99% to perform the functions without any failure.

3.6.2 Security

The CNN model can be easily affected by outside effects. So some encryptions mechanisms are taken into account to prevent the loss of data during the process.

3.6.3 Maintainability

The neural network is designed and developed so that it can re-trained with new data. It will lead easily maintainable system.

3.6.4 Scalability

This system uses a service orchestrator which can distribute the load when the load increases. Work will be distributed horizontally.

3.6.5 Availability

System will have a high availability of 99.99%. The system is designed to carry out tasks simultaneously and the system will be up and running all the time.

4 References

[1] X. a. L. C. a. L. S.-W. a. Z. V. Zhang, *Automated segmentation of MOOC lectures towards customized learning*, IEEE, 2016.