**MOOCREC: LEARNING STYLES-ORIENTED MOOCS RECOMMENDER AND SEARCH ENGINE**

Saugat Aryal, Porawagama A.S, Hasith M.G.S, Thoradeniya S.D

IT 14 146602, IT 14 142024, IT 14 140280, IT 14 138232

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

Department of Software Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September 2018

**MOOCREC: LEARNING STYLES-ORIENTED MOOCS RECOMMENDER AND SEARCH ENGINE**

Saugat Aryal, Porawagama A.S, Hasith M.G.S, Thoradeniya S.D

IT 14 146602, IT 14 142024, IT 14 140280, IT 14 138232

Dissertation submitted in partial fulfillment of the requirements for the degree Bachelor of Science Special (Honors) in Information Technology Specializing in Software Engineering

Department of Software Engineering

Sri Lanka Institute of Information Technology

September 2018

**DECLARATION**

We declare that this is our own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, We hereby grant to Sri Lanka Institute of Information Technology the nonexclusive right to reproduce and distribute our dissertation, in whole or in part in print, electronic or other medium. We retain the right to use this content in whole or part in future works (such as articles or books).

Date:2018-09-03

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Signature |
| Saugat Aryal | IT 14 146602 |  |
| Porawagama A.S | IT 14 142024 |  |
| Hasith M.G.S | IT 14 140280 |  |
| Thoradeniya S.D | IT 14 138232 |  |

The above candidate has carried out research for the B.Sc Dissertation under my supervision.

Supervisor: Mr. Nuwan Kodagoda

Co-Supervisor: Ms. Kushnara Suriyawansa

Signature:Date:2018-09-03

**ABSTRACT**

Massive Open Online Courses (MOOCs) are the new revolution in the field of e-learning, providing a large number of courses in different domains to a wide range of learners. Because of the several MOOC platforms (like, edX, Coursera, Udacity, Futurlearn etc.) available today, it is a tedious manual task for a learner to browse through different courses in these platforms before he finds the best course that meets his learning style and current knowledge level in that subject matter. MoocRec is a unique learning styles-oriented system that aims to recommend the most appropriate courses for a learner from different MOOC platforms based on his learning styles and individual needs. The standard video styles used in MOOCs (such as talking head, tutorial/demonstration, slides, animation etc.) are considered as the key parameter when mapping a course with the learning styles. MoocRec also allows the learners to search for courses using, specific topics and English accent of the speaker to provide them with a better- personalized learning environment.

**ACKNOWLEDGEMENT**

The work described in this research paper was carried out as our 4th year research project for the subject Comprehensive Design Analysis Project. The completed final project is the result of combining all the hard work of the group members and the encouragement, support and guidance given by many others. Therefore, it is our duty to express our gratitude to all who gave us the support to complete this major task.

We are deeply indebted to our supervisor Mr. Nuwan Kodagoda and our co-supervisor Ms. Kushnara Suriyawansa, lecturers of Sri Lanka Institute of Information Technology whose suggestions, constant encouragement and support in the development of this research, particularly for the many stimulating and instructive discussions. We are also extremely grateful to Mr. Jayantha Amararachchi, Senior Lecturer/ Head-SLIIT Centre for Research who gave and confirmed the permission to carry out this research and for all the encouragement and guidance given.

We also wish to thank all our colleagues and friends for all their help, support, interest and valuable advices. Finally, we would like to thank all others whose names are not listed particularly but have given their support in many ways and encouraged us to make this a success.

Table of Contents

[1 INTRODUCTION 8](#_Toc523782610)

[1.1 Background Context 8](#_Toc523782611)

[1.2 Research Gap 8](#_Toc523782612)

[1.3 Research Questions 9](#_Toc523782613)

[2 Methodology 10](#_Toc523782614)

[2.1 Web Crawling and Scraping 10](#_Toc523782615)

[2.2 Video Styles Classification 11](#_Toc523782616)

[2.3 Topic Modeling 13](#_Toc523782617)

[2.4 Linguistic Complexity 14](#_Toc523782618)

[2.5 Mapping of MOOCs with Learning Styles 14](#_Toc523782619)

[2.6 Web Application 16](#_Toc523782620)

[2.7 Tools and Technologies 16](#_Toc523782621)

[2.8 Research Findings 17](#_Toc523782622)

[3 Results and Discussion 19](#_Toc523782623)

[3.1 Evidence 19](#_Toc523782624)

[3.2 Discussion 22](#_Toc523782625)

[4 Conclusion 22](#_Toc523782626)

[5 References 23](#_Toc523782627)

[6 Appendices 25](#_Toc523782628)

**LIST OF FIGURES**

[Figure 1: High-Level Architecture Diagram 10](#_Toc523782577)

[Figure 2: Frontend Web Application 16](#_Toc523782578)

[Figure 3: Determine the Learning Styles 20](#_Toc523782579)

[Figure 4: Recommend MOOCs to Learner 20](#_Toc523782580)

[Figure 5: Searching for Specific Course 21](#_Toc523782581)

[Figure 6: Filter MOOCs based on Topics 21](#_Toc523782582)

[Figure 7: Class Diagram 25](#_Toc523782583)

[Figure 8: Activity Diagram 26](file:///F:\Year%204\Research\18-036_ThesisDraft.docx#_Toc523782584)

[Figure 9: Use-Case Diagram 27](file:///F:\Year%204\Research\18-036_ThesisDraft.docx#_Toc523782585)

**LIST OF TABLES**

[Table 1: Felder and Silverman Learning Style Model 15](#_Toc523782592)

[Table 2: Mapping of MOOCs with FSLSM 15](#_Toc523782593)

[Table 3: Web Scrapper Performance for edX 17](#_Toc523782594)

[Table 4: Web Scrapper Performance for Coursera 18](#_Toc523782595)

[Table 5: Topic Modeling Sample 1 19](#_Toc523782596)

[Table 6: Topic Modeling Sample 2 19](#_Toc523782597)

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| MOOCs | Massive Open Online Courses |
| FSLSM | Felder and Silverman Learning Style Model |
| ILS | Index of Learning Style |
| CPU | Central Processing Unit |
| RAM | Random Access Memory |
| CNN | Convolutional Neural Network |
| OS | Operating System |

# INTRODUCTION

## Background Context

MOOCs have emerged as big players in the field of online learning and its unique characteristics make it more effective in the modern era of education. Over the years, various platforms such as edX, Coursera, Udacity, Futurelearn etc. have emerged with the intent of providing massive educational resources to any learners. According to the survey from Class Central [1], until 2017, around 81 million students are registered worldwide, where 23 million were new learners that registered only in 2017. Similarly, the total number of MOOCs surpassed 9400 contributed from over 800 universities in 2017 [1]. Hence, we can see the exponential growth of MOOCs.

In the recent years, there has been increasing attention towards the characteristics of learners such as learning styles. Different learners have the ability to learn in different ways and hence poses their own style of learning. Because of this behavior, the learner mostly explores through different MOOCs platform to find the most suitable learning resources that best fit their needs, preferences and learning styles. It has been also stated from studies that providing learners with learning materials and resources that suit their preferences and learning style makes learning easier for them [2][3].

Several MOOC search engines are currently available, such as, Class Central [4], My Mooc [5], MoocLab [6] and Coursetalk [7] that serves as a unified platform for MOOC platforms. However, none of the existing systems takes the learner’s learning styles into consideration when searching for courses. Also, the learner can only search for courses based on basic filters, like provider, category, duration, language etc.

## Research Gap

Several studies have proposed to integrate the concept of learning styles into the open learning environment (MOOCs), to provide adaptive and personalized support for learning [3], [8], [9]. Other studies reveal the use of data mining and machine learning algorithms to automatically identify the learner’s learning styles. However, there has been no significant research to support the direct mapping of learning styles with MOOCs.

Researches regarding MOOCs search engine and recommenders are also being carried out at a rapid rate. A recommender system using Case Based Reasoning (CBR) approach is proposed in [10]. User’s query is described by five attributes where each attribute is assigned a weight value based on the user’s preference. “Courducate” is another system proposed for a personalized search engine with two functionalities: multi-site search and multi-filed search [11]. Besides using the BM25 ranking function, a noble ranking function is used to rank the sites upon query. A different approach is taken in [12] where the authors propose to associate MOOCs with learning outcomes. Hence, allowing learners to discover the most suitable MOOCs for their learning objectives. [13] proposes two contributions: Using attribute and attribute value weight of resources to get specific user preferences; A new algorithm to overcome the shortcomings of the Collaborative Filtering (CF) and provide more accurate personalized recommendations on MOOCs. In the similar context, Content-based and collaborative filtering recommendation approaches are used to accommodate several undergraduate characteristics when recommending MOOCs [14].

Finally, we can conclude that, until now, no system has been implemented that recommends courses to a learner from different MOOC platforms based on his learning styles and other personalized needs and requirements.

## Research Questions

The relationship between video styles used in MOOCs and learning styles of the learner needs to addressed. Questions regarding the affect of a specific video style to learner, browsing of MOOCs from different platforms etc. needs to answered. This research study attempts to answer the questions relating to these factors.

# Methodology

The high-level architecture diagram of MoocRec is shown in Figure 1. The web scrappers will first extract MOOC contents (videos, transcripts, description etc.) from three different platforms: edX, Coursera, and Futurelearn. After scraping the data, it calls the video classification, topic modeling, linguistic complexity and accent detection functions one after another. The video classification process uses a deep learning approach to automatically classify standard video styles used in MOOC platforms. These video styles are considered as a key parameter when mapping a course with the learning styles. The topic modeling component will extract abstract topics from the transcripts which are used when the learner wants to filter courses using specific keywords/topics. The linguistic complexity process determines the complexity level of the transcripts. The accent detection component classifies the English accent of the speaker as native or non-native. Finally, the output of each process is stored in the non-relational MongoDB database hosted in the cloud. All the backend processes are also hosted in the Amazon Cloud Server which is connected to a frontend web application through API. More details about each process are described in this section:

A screenshot of a social media post

Description generated with very high confidence

Figure 1: High-Level Architecture Diagram

## Web Crawling and Scraping

Web scraping is a software program that extracts information from websites. These programs are able to simulate human web surfing behavior by implementing either low-level Hypertext Transfer Protocol (HTTP) or embedding a web browser. It concentrates on transforming unstructured data found online, into structured data that can then be stored and analyzed into a database or internal storage. The web crawler that we have implemented executes five different functions. First, it undergoes the task of web interaction. This will happen when a new course is detected in one of our MOOC sites (edX, Coursera or Futurelearn). Second, programs known as wrappers need support for generation and execution. At this point, we collect overall information about the contents to be downloaded and their size. Scheduling is the third function that allows wrappers to be applied repeatedly to their respective target pages. Next, data transformation occurs which involves the filtering, mapping, refining and integrating of data from one or more sources. The result is then structured to create the desired output format. Finally, different contents of a course, like course videos, transcripts, description, pdf documents that are available are downloaded and stored temporarily to perform further analysis.

## Video Styles Classification

Video lectures are the fundamental and significant component of MOOCs. They are produced in various standard styles that are used across different MOOCs platforms. Usually, a single video is the composition of these different styles. Guo et al. [15] mentions 6 different types of production styles: Slides (PowerPoint presentation with voice-over), Code (video screencast of writing code), Khan-style (full-screen video of instructor drawing free-hand), Classroom (video captured in live classroom), Studio (recording in a studio with no audience) and Office Desk (close-up shots of instructor at an office desk). Similarly, Hansch et al. [16] also presents similar video styles but using different names. Other additional styles like, Animation, Conversation, Text-Overlay, Picture-in-Picture are also described. However, for the system only, talking head (labeled as Office Desk by Guo), slides, code, conversation and animation are considered as they are most common in computer programming related courses, which is the focus of MoocRec.

The videos are first decomposed into a set of image frames using OpenCV [17] library in python. Then, we perform image classification of these frames. To build a powerful classification model and achieve better accuracy, we leverage the features of various existing deep Convolutional Neural Network (CNN) architectures called, VGGNet [18], GoogLeNet [19], ResNet [20], that are pre-trained on large ImageNet dataset [21]. Only the convolutional part of the deep architectures are instantiated to record the “bottleneck features” from the training and test dataset. And finally, a small fully connected model is trained on top of these stored features to build a new model. Different classifier models are built on top of each architecture and based on their accuracy level, the most appropriate model is selected for classification and further processing.

After the image frames have been classified by the neural network model, we then calculate the composition of each video style type for a single video and as well as for the overall course. While all the image frames can be extracted from a video, one frame in every two minutes is considered. This is because adjacent frames in high-frame-rate videos do not change significantly.

If t is the time period of a single MOOC video in seconds, the number of frames n considered for classification is given by:

n = t / 120 (1)

If h, s, and c is the number of frames predicted as talking head, slide and code by the model, then the composition of each style in percentage (%) in a single video is given by:

head = (h / n) \* 100 (2)

slide = (s / n) \* 100 (3)

code = (c / n) \* 100 (4)

Finally, if there are total of v number of videos in a MOOC course, where h1, h2, h3…hv represent the amount of talking head, s1, s2, s3…sv represent the amount of slide and c1, c2, c3…cv are the amount of code for each video, then the average composition of each style in percentage(%) for the overall course is given by:

headaverage = (h1 + h2 + h3 +…+hv / v) \* 100 (5)

slideaverage = (s1 + s2 + s3 +…+sv / v) \* 100 (6)

codeaverage = (c1 + c2 + c3 +…+cv / v) \* 100 (7)

These average values calculated for each video styles are used while mapping a MOOC with the learning styles.

## Topic Modeling

Topic modeling is a frequently used text-mining tool for discovery of hidden semantic structures in a text body for discovering the abstract “topics” that occur in a collection of documents. In MOOCs platform, transcripts are available for each video lecture. Specific topics can be extracted from the document which is useful in the scenario where the learner wants to search for MOOC with specific topics.

Latent Dirichlet allocation (LDA), first introduced by Blei, Ng, and Jordan in 2003 [22], is one of the most popular methods in topic modeling. LDA represents topics by word probabilities. The words with the highest probabilities in each topic usually give a good idea of what the topic is can word probabilities from LDA.

The transcripts of the MOOC courses undergoes Data Pre-processing and following steps are performed before applying LDA models [23]:

* Tokenization- Split the text into sentences and the sentences into words. Lowercase the words and remove punctuation.
* Words that have fewer than 3 characters are removed.
* All stop words are removed. (Stop Words are words which do not contain important significance to be used in Search Queries)
* Words are lemmatized — words in third person are changed to first person and verbs in past and future tenses are changed into present.

LDA, an unsupervised generative probabilistic method for modeling a corpus, is the most commonly used topic modeling method. Given a corpus *D* consisting of *M* documents, with document *d* having Nd words (*d* {1,..., M}), LDA models *D* according to the following generative process [22].

1. Choose a multinomial distribution *φ**t*for topic *t* (*t* ∈{1,..., *T*}) from a Dirichlet distribution with parameter *β*.
2. Choose a multinomial distribution *θ**d*for document *d* (*d* ∈{1,..., *M*}) from a Dirichlet distribution with parameter *α*.
3. For a word *wn*(*n* ∈{1,..., *N**d*}) in document *d*,
4. Select a topic *zn*from *θ**d*
5. Select a word *wn*from *φ**zn*

LDA is a distinguished tool for latent topic distribution for a large corpus. Therefore, it has the ability to identify sub-topics for a technology area composed of many patents and represent each of the patents in an array of topic distributions. The output topics for each video lectures are then stored in the database.

## Linguistic Complexity

The Linguistic complexity of the transcripts is measured using two main variables, that are calculated separately [23].

1. *Syntactic Complexity* [24]: There are 14 different measures of syntactic complexity. Some of them are, Mean Length of Clause (MLC), Mean Length of Sentence (MLS), Mean Length of T-Unit (MLT) etc. To calculate the syntactic complexity of the transcripts, first, they are fed as input which is parsed using Stanford parser. Then we count the occurrences of words, sentences, clause, t-unit etc. Then finally we calculate the complexity using those 14 measures and store the syntactic complexity indices.
2. *Semantic Complexity:* Semantic complexity correlates with the number of ways meaning can be derived and interpreted from an utterance. It is also associated with the types of syntactical structures necessary for it to be an intelligible utterance, and the number of different pathways meaning can be retrieved from.

## Mapping of MOOCs with Learning Styles

Various Learning Style models have been proposed over the years, however, we have adopted the widely used Felder and Silverman Learning Style Model (FSLSM) based on literature [25]. It has been proven from studies that FSLSM is the most appropriate model in personalized online learning environments. The FSLSM classifies learning styles into four different dimensions and identifies two types of learners for each dimension. The four dimensions are perception, input, processing, and understanding. A brief description of the characteristics of each learning styles is shown in Table 1.

Table 1: Felder and Silverman Learning Style Model

|  |  |  |
| --- | --- | --- |
| Dimension | Learner Styles and their Characteristics | |
| Perception | Sensory – Concrete or Practical Information | Intuitive – Conceptual or Theoretical Information |
| Input | Visual – Graphs/ Pictures / Diagrams | Verbal – Hear or Read Information |
| Processing | Active – Experiments / Participation | Reflective – Individual Learning or Thinking |
| Understanding | Sequential – Information in linear order | Global – Think about the big picture |

In order to identify the learning styles of the learner, Index of Learning Styles (ILS) questionnaire [26] is embedded in the system. It contains 44 questions that determine a learner with learning style from each dimension. For example, a learner is sensory, visual, reflective and sequential or intuitive, verbal, active and global so on. Considering the properties of these learning styles and the characteristics of MOOC video styles and literature [27], a two-dimensional mapping was generated. The detail about the mapping is shown in Table 2.

Table 2: Mapping of MOOCs with FSLSM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Talking Head | Code / Tutorial | Slide | Conversation | Animation |
| Sensory |  | Yes | Yes |  |  |
| Intuitive | Yes |  |  | Yes | Yes |
| Visual |  |  |  |  | Yes |
| Verbal | Yes | Yes | Yes |  |  |
| Active |  | Yes |  | Yes |  |
| Reflective |  |  | Yes |  |  |
| Sequential |  | Yes | Yes |  |  |
| Global | Yes |  |  | Yes | Yes |

Using the Equation (5), (6) and (7), if the percentage of average talking head video style is more in a course, then based on the mapping of Table 2, it is more suitable to an intuitive, verbal and global learner. Similarly, if the coding video style dominates a video, it is more favorable to a sensory, verbal, active and global learner and so on.

## Web Application

The frontend client application of MoocRec is a web application developed using Angular5, HTML5, CSS3, and JavaScript. Any learner who uses the system takes the ILS questionnaire to identify his learning styles. He can also take the standardized tests for programming languages, like Java, Python, PHP to determine their current knowledge level in that domain. Finally, the system recommends the most appropriate MOOCs from different platforms based on the mapping data. The learner can also further filter the courses. Figure 2 describes the user interface for searching recommended MOOCs based on different parameters.

A screenshot of a cell phone

Description generated with very high confidence

Figure 2: Frontend Web Application

## Tools and Technologies

**Tools**

* Anaconda IDE
* Google Colaboratory
* JetBrains WebStorm
* MongoDB Compass

**Technologies**

* Python
* Keras Library
* OpenCV
* Tensorflow Backend
* Matplotlib
* Natural Language Toolkit
* Scrapy
* Beautiful Soup
* HTML5, CSS3, JavaScript, Angular5
* MongoDB

## Research Findings

“MoocRec” is still in the implementation phase and hence, only three different processes, including web scraping, video styles classification and topic modeling were evaluated based on the tasks completed. The results are presented in this section.

The performance of the web scrapper for downloading a sample course in two MOOC platforms are shown in Table 3 and 4.

Table 3: Web Scrapper Performance for edX

|  |  |  |
| --- | --- | --- |
| Platform - edX | | |
| Course – Java Fundamentals for Android Development | | |
| Contents | Available in the Course | Downloaded from Scrapper |
| Videos | 109 | 109 |
| Video Transcripts (txt) | 0 | 0 |
| Documents (Pdf) | 35 | 35 |

Table 4: Web Scrapper Performance for Coursera

|  |  |  |
| --- | --- | --- |
| Platform - Coursera | | |
| Course – Java Programming: Solving Problems with Software | | |
| Contents | Available in the Course | Downloaded from Scrapper |
| Videos | 61 | 61 |
| Video Transcripts (txt) | 61 | 61 |
| Documents (Pdf) | 15 | 15 |

The scrapper was evaluated by taking the average time taken for downloading the same course five times. The average time taken to download a course from edX with 109 videos, where each video has an average time of 8 minutes and 35 seconds long was 120 minutes and 15 seconds. Similarly, for a course in Coursera with 61 videos, where each video has an average time of 5 minutes and 10 seconds long, took 30 minutes and 15 seconds. The web scraping time solely depends on the Internet connection being used. These results were obtained from a 4G connection with a speed of around 7 Mbps.

For video styles classification, 500 image frames of three different video styles (talking head, slides, and code) each were used as training set and 100 images as testing set. The VGG-16 [28] based classifier model was trained on a laptop with 6 GB RAM and Intel Core i5 2.20 GHz CPU processor, where it took around 120 minutes. The test set accuracy obtained is 90 percent classification accuracy for the three classes.

The topic modeling component is evaluated by comparing the relevant number of topics extracted from the course with the ones obtained from the algorithm. The results of the evaluation for a sample course in two platforms are shown in Table 5 and 6.

Table 5: Topic Modeling Sample 1

|  |  |  |
| --- | --- | --- |
| Platform - Coursera | | |
| Course – Object-Oriented Programming in Java | | |
| Content | Available in the Course | Using Topic Modeling Algorithm |
| Sub-Topics | 8 | 5 |

Table 6: Topic Modeling Sample 2

|  |  |  |
| --- | --- | --- |
| Platform - edX | | |
| Course – Introduction to Java Programming – Part I | | |
| Content | Available in the Course | Using Topic Modeling Algorithm |
| Sub-Topics | 14 | 20 |

It takes approx.1 minute for the topic modeling algorithm to extract the topics from a collection of course transcripts having more than 23,000 words on a laptop with 8 GB RAM and Intel Core i5 1.60 GHz CPU processor.

The results shown in this section are for the backend processes for indexing a new course which is a one-time process

# Results and Discussion

## Evidence

After the learner takes the ILS questionnaire, he/she can identify their learning styles. MoocRec works perfectly to correctly identify the learning styles based on the questions answered as shown in Figure 3.

A screenshot of a social media post

Description generated with very high confidence

Figure 3: Determine the Learning Styles

The system then recommends the most suitable courses to the learner based on his learning styles. At the same time, it also responds correctly when the learner searches for courses using specific keywords and topics as shown in Figure 4, 5 and 6.

A screenshot of a cell phone

Description generated with very high confidence

Figure 4: Recommend MOOCs to Learner

A screenshot of a cell phone

Description generated with very high confidence

Figure 5: Searching for Specific Course

A screenshot of a cell phone

Description generated with very high confidence

Figure 6: Filter MOOCs based on Topics

## Discussion

To validate the overall system, an experiment is setup including a few group of students with different learning styles. First, they will be asked to browse through multiple MOOC platforms and select the best course for them. Next, they will be using the system which will recommend them the courses based on their learning styles. The performance and accuracy of the overall system can hence be calculated.

The discussions for other components will be updated as soon as the overall implementation of the system is complete.

# Conclusion

In today’s world, MOOCs have grown as a popular platform for learning attracting learners of different learning styles. However, because of the different learning styles of the learners and the availability of similar courses in different MOOC platforms, it is overwhelming for a learner to explore through different resources before they finally find the course that is most suitable for them. This research paper proposes a practically usable solution called, MoocRec to overcome this widely faced problem. MoocRec is a first approach towards the objective of providing personalized MOOCs learning on the basis of the learner’s preferences and needs.

Although MoocRec is designed only for computer programming related courses in three MOOC platforms (edX, Coursera, and Futurelearn), it can be expanded to cover other domains of study and platforms as well.

# References

[1] “By The Numbers: MOOCS in 2017 — Class Central.” [Online]. Available: https://www.class-central.com/report/mooc-stats-2017/. [Accessed: 26-Mar-2018].

[2] S. Graf and T.-C. Liu, “Supporting Teachers in Identifying Students’ Learning Styles in Learning Management Systems: An Automatic Student Modelling Approach,” *Educ. Technol. Soc.*, vol. 12, no. 4, pp. 3–14, 2009.

[3] C. Li and H. Zhou, “Enhancing the Efficiency of Massive Online Learning by Integrating Intelligent Analysis into MOOCs with an Application to Education of Sustainability,” *Sustainability*, vol. 10, no. 2, p. 468, Feb. 2018.

[4] “Class Central • #1 Search Engine for Free Online Courses &amp; MOOCs.” [Online]. Available: https://www.class-central.com/. [Accessed: 03-Apr-2018].

[5] “Browse the best MOOC and free online trainings in the world - My Mooc.” [Online]. Available: https://www.my-mooc.com/en/. [Accessed: 03-Apr-2018].

[6] “Home | MoocLab - Connecting People to Online Learning.” [Online]. Available: https://www.mooclab.club/. [Accessed: 03-Apr-2018].

[7] “Student reviews of online courses on Computer Science, Business, Design, Data Science, Humanities and more | CourseTalk.” [Online]. Available: https://www.coursetalk.com/. [Accessed: 03-Apr-2018].

[8] H. A. Fasihuddin, G. D. Skinner, and R. I. Athauda, “Boosting the Opportunities of Open Learning (MOOCs) through Learning Theories,” *GSTF J. Comput.*, vol. 3, no. 3, p. 31, Dec. 2013.

[9] H. A. Fasihuddin, G. D. Skinner, and R. I. Athauda, “Personalizing Open Learning Environments through the adaptation to Learning Styles.”

[10] F. Bousbahi and H. Chorfi, “MOOC-Rec: A Case Based Recommender System for MOOCs,” *Procedia - Soc. Behav. Sci.*, vol. 195, pp. 1813–1822, Jul. 2015.

[11] Q. Cheng and Y. Gao, “Courducate --An MOOC Search and Recommendation System.”

[12] C. D.-K. Israel Gutiérrez-Rojas, Derick Leony, Carlos Alario-Hoyos, Mar Pérez-Sanagustín, “Towards an Outcome-based Discovery and Recommendation of MOOCs using moocrank,” in *European MOOCs Stakeholders Summit*, 2014.

[13] Y. Wang, B. Liang, W. Ji, S. Wang, and Y. Chen, “An improved algorithm for personalized recommendation on MOOCs,” *Int. J. Crowd Sci.*, vol. 1, no. 3, pp. 186–196, Sep. 2017.

[14] D. Fu, Q. Liu, S. Zhang, and J. Wang, “The Undergraduate-Oriented Framework of MOOCs Recommender System,” in *2015 International Symposium on Educational Technology (ISET)*, 2015, pp. 115–119.

[15] P. J. Guo, J. Kim, and R. Rubin, “How video production affects student engagement,” in *Proceedings of the first ACM conference on Learning @ scale conference - L@S ’14*, 2014, pp. 41–50.

[16] A. Hansch, L. Hillers, K. McConachie, C. Newman, T. Schildhauer, and P. Schmidt, “Video and Online Learning: Critical Reflections and Findings from the Field,” *SSRN Electron. J.*, Mar. 2015.

[17] “OpenCV library.” [Online]. Available: https://opencv.org/. [Accessed: 13-Aug-2018].

[18] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Sep. 2014.

[19] C. Szegedy *et al.*, “Going Deeper with Convolutions,” Sep. 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” Dec. 2015.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[22] D. M. Blei, A. Y. Ng, and J. B. Edu, “Latent Dirichlet Allocation Michael I. Jordan,” 2003.

[23] “Topic Modeling and Latent Dirichlet Allocation (LDA) in Python.” [Online]. Available: https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24. [Accessed: 12-Jun-2018].

# Appendices

**A screenshot of a social media post

Description generated with very high confidence**

Figure 7: Class Diagram

A screenshot of a cell phone

Description generated with very high confidence

Figure 8: Activity Diagram

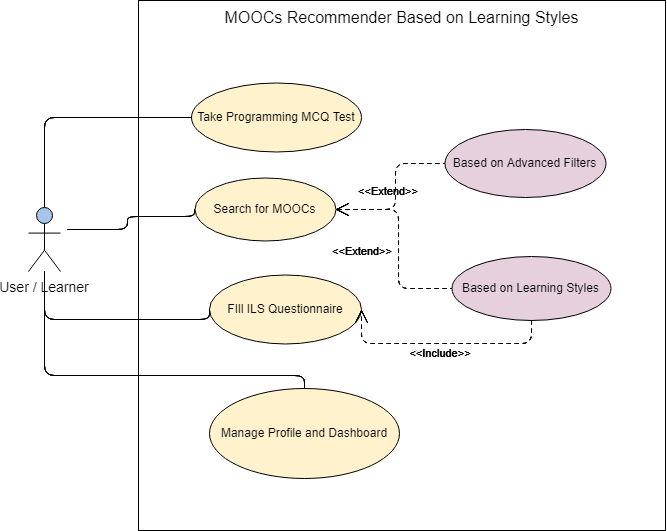


Figure 9: Use-Case Diagram