MoocRec: Learning Styles-Oriented MOOC Recommender and Search Engine

Saugat Aryal
Department of Software Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
saugat.aryl@gmail.com

Anjana Shriwantha Porawagama
Department of Software Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
axewilledge@gmail.com

Makulle Gedara Sajeeva Hasith
Department of Software Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
sa.hasith@gmail.com

Sachini Chloe Thoradeniya
Department of Software Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka.
chloethora@gmail.com

Nuwan Kodagoda
Department of Software Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
nuwan.k@sliit.lk

Kushnara Suriyawansa
Department of Information Technology
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
kushnara.s@sliit.lk

***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. Due to the availability of several MOOC providers (including edX, Coursera, Udacity, FutureLearn), a specific domain has multiple courses spread across these platforms that confuses a learner on selecting the most suitable course for him. It is a tedious manual task for the learner to browse through various courses before he finds the best course that meets his learning requirements and objectives. MoocRec is a unique learning styles-oriented system that recommends the most suitable courses to a learner from different MOOC platforms based on their learning styles and individual needs. The courses are recommended based on the mapping of Felder and Silverman learning styles with the standard video styles used in MOOC videos (including talking head, slide, tutorial/demonstration). MoocRec also allows the learners to search for courses using specific topics to provide an enhanced personalized learning environment. Results show that MoocRec is strongly reliable and can be used for personalized learning.**

Keywords–MOOCs, Learning Styles, Video Styles Classification, Web Scraping, Topic Modeling

I. INTRODUCTION

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 9,400 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 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 also been stated from studies that learning becomes easier, effective and efficient when learners are provided with learning materials and resources that match their preferences and learning style. [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 recommending courses. Also, the learner can only search for courses based on basic filters, like provider, category, duration, language etc.

MoocRec is a sophisticated educational platform that works by integrating two domains: MOOCs video styles and learning styles. The primary purpose of the system is to recommend the most suitable learning resources to a learner based on his learning styles. It works by performing content analysis of MOOCs data (videos and transcripts). In addition to recommending courses, it also allows the learner to filter courses using specific topics parameter. Although any individual learner who is interested to learn through MOOCs can use MoocRec, it currently works for freely available computer programming related courses from two different platforms: edX and Coursera.

II. LITERATURE REVIEW

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.

Research 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 by Bousbahi and Chorfi [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. Wang et al. [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 [13]. In the similar context, Content-based and collaborative filtering recommendation approaches are used to accommodate several undergraduate characteristics when recommending MOOCs [14].

A thorough study of literature reveals that, no system has been implemented yet that recommends courses to a learner from different MOOC platforms based on his learning styles and other personalized needs and requirements.

III. METHODOLOGY

The high-level architecture diagram of MoocRec is shown in Fig. 1. The backend of MoocRec consists of several modules. The web scrappers first extract MOOC contents (videos and transcripts) of free courses from two different platforms: edX and Coursera. After scraping the data, it invokes the video classification and topic modeling services consecutively to perform the content analysis. In general, a single MOOC video is composed of multiple styles. The video classification component uses an image-based deep learning approach to automatically classify the standard video styles available in the video and calculate their individual composition level. The computed values are used as a reference when mapping a course with the learning styles of the learner. The topic modeling component extracts abstract topics from the transcripts that are used when the learner wants to filter courses using specific keywords/topics. More details about each process are described in this section:

1) *Web Crawling and Scraping*

Web scraping is the process of automatically extracting information or data from websites using a script. The process involves parsing the unstructured web content and then transforming it to get structured data that can be analyzed and stored into a database or internal storage. Web crawling,

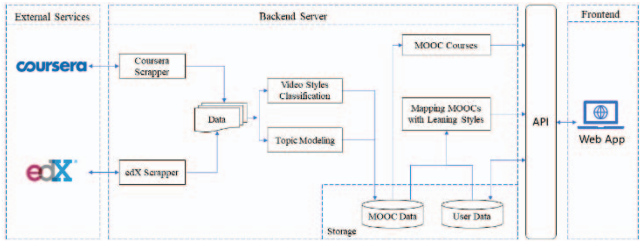

Figure 1: High-Level Architecture of MoocRec

on the other hand, involves the process of indexing the hyperlinks of a website. Both the process of crawling and scraping goes together when extracting the contents found online. The web crawler implemented to download the MOOCs content works with five different functions. First, it undergoes the task of web interaction. This happens when a new course is detected in one of the MOOC platforms (edX or Coursera). Next, programs called wrappers need support for generation and execution. At this stage, the overall information about the contents to be downloaded and their size is gathered. Consequently, scheduling is carried out that allows the wrappers to be applied repeatedly to their respective target pages. Next, data transformation occurs which involves 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 content analysis.

2) *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 in the background), Code (tutorial or demonstration by writing code), Khan-style (full-screen video of instructor demonstrating using free-hand), Classroom (video captured in live classroom), Studio (video recording in a studio) 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, slides and code video styles are considered as they are most common in computer programming related courses [17], which is the focus of MoocRec. The description of the styles is shown in Table I.

Table I. Video Styles Description

Video Style Category	Description
Talking Head	Shows the instructor's head
Slide	Powerpoint slide presentation with educational content
Code	Software demonstration, tutorial or full-screen code-writing


Figure 2: Talking Head

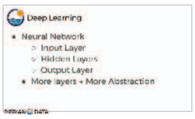

Figure 3: Slide

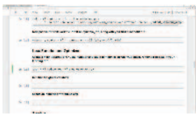
Figure 4: Code

Image-based classification approach is adopted to classify the video styles where a single video is split into multiple image frames using OpenCV [18] library in python and each frame is classified into one of the video style classes. To build a powerful classification model and achieve better accuracy [19], we leverage the features of several existing state-of-the-art deep neural architectures, including, VGG16 [20], InceptionV3 [21], and ResNet50 [22]. Only the convolutional part of the models was instantiated to record the "bottleneck features" and finally, a custom fully connected layer with three classes was trained on top of these stored features to build a new model. The VGG16-based model showed the highest accuracy of 92.5% among the three and hence was selected to perform the classification.

After the image frames have been classified by the 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:

$$n = t / 120 \tag{1}$$

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:

$$head = (h / n) * 100 \tag{2}$$

$$slide = (s / n) * 100 \tag{3}$$

$$code = (c / n) * 100 \tag{4}$$

Finally, if there are total of v number of videos in a MOOC course, where $h_1, h_2, h_3 \ldots h_v$ represent the amount of talking head, $s_1, s_2, s_3 \ldots s_v$ represent the amount of slide and $c_1, c_2, c_3 \ldots c_v$ are the amount of code for each video, then the average composition of each style in percentage(%) for the overall course is given by:

$$head_{average} = (h_1 + h_2 + h_3 + \ldots + h_v / v) * 100 \tag{5}$$

$$slide_{average} = (s_1 + s_2 + s_3 + \ldots + s_v / v) * 100 \tag{6}$$

$$code_{average} = (c_1 + c_2 + c_3 + \ldots + c_v / v) * 100 \tag{7}$$

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

3) Mapping 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 [23]. It has been proven from studies that FSLSM is the most recommended model in personalized online learning environments. The FSLSM categorizes learning styles into four different dimensions and defines two types of learning styles 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 II.

Table II. Felder and Silverman Learning Style Model

Dimension	Learner Styles and their Characteristics	
Perception	Sensory – Concrete or Practical Information. These types of learners are oriented towards in detail procedures and facts rather than abstract information.	Intuitive – Conceptual or Theoretical Information. Prefers to learn by referring to abstract information.
Input	Visual – Prefers to learn from materials with more graphs, pictures and diagrams.	Verbal – Prefers to learn by listening and reading.
Processing	Active – These learners learn better by practically applying and having discussions.	Reflective – These learners prefer to learn individually thinking.
Understanding	Sequential – Prefers to get information in a sequential manner.	Global – Prefers to learn by looking at an overall view.

In order to identify the learning styles of the learner, Index of Learning Styles (ILS) questionnaire [24] is embedded in the system. It contains 44 questions that determines the overall learning style as a combination of one style from each dimension. For example, a learner can be sensory, visual, reflective and sequential or intuitive, verbal, active and global and so on. The relationship between video styles in MOOCs and learning styles shown in Table II is determined based on the characteristics of FSLSM, properties of video styles and literature [25]. MoocRec works on the basis of this correlation table using which most suitable and appropriate MOOCs are recommended to a learner.

Table III. Mapping of MOOCs with FSLSM Model

	Talki-ng Head	Code / Tutorial	Slide	Conver-sation	Animati-on
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

Characteristics of sensory learners are mapped with only code/tutorial and slide video styles as they prefer to learn from procedures and in detail information. Talking head and conversation video styles will not have an impact on these learners as they do not match with any special feature of sensory learners. Animations in a technical domain instructional video may mostly present an abstract view of the information. Thus, it is considered under intuitive learners. Only animation video style is mapped with visual learners since it is the only video style which will assuredly contain graphics. Using the equations (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 III, 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.

4) Topic Modeling

Topic modeling is a commonly used text-mining technique for identifying hidden semantic structures or abstract topics in a text document. In MOOCs platform, transcripts are available for each video lecture. The extracted topics from the transcripts is useful in the scenario when the learner wants to search for MOOC with specific topics which allows for more personalized environment for learning.

Latent Dirichlet allocation (LDA) is one of the most widely used method for topic modeling which was first introduced by Blei, Ng, and Jordan in 2003 [26]. LDA is based on the probabilities of word. The higher probability of the words in the documents usually give an idea for what the hidden topics are.

The transcripts of the MOOC courses undergoes data pre-processing and following steps are performed before applying LDA model [27]:

- Tokenization - Get the lowercase words without punctuations of the transcripts by splitting the text into sentences and then into words.
- Remove the words that have characters less than 3.
- Remove all the Stopwords (Words which do not affect for search queries).
- Lemmatize the words - Past and future tense verbs are changed into present. Third person words are changed into first person.

Modeling a corpus using the LDA, an unsupervised generative probabilistic method is one of the most commonly used topic modeling algorithm. Given a corpus *D* consisting of *M* documents, with document *d* having Nd words ($d \in \{1,..., M\}$), LDA models *D* according to the following generative process [28].

1) From a Dirichlet distribution with parameter β, find a multinomial distribution φ t for topic t ($t \in \{1,..., T\}$)

2) From a Dirichlet distribution with parameter α, find a multinomial distribution θ d for document d ($d \in \{1,..., M\}$)

3) For a word w_n ($n \in \{1,..., N_d\}$) in document *d*,
   a) Select a topic z_n from θ_d
   b) Select a word w_n from φ_{z_n}

For large corpus of latent topic distributions, LDA is a suitable tool that has the ability to discover abstract topics in a text document which is composed of many discoverable patents. It represents an array of topic distribution for each of the patents. Hence, the output topics for each video lectures transcripts obtained are then stored in the database.

5) Web Application

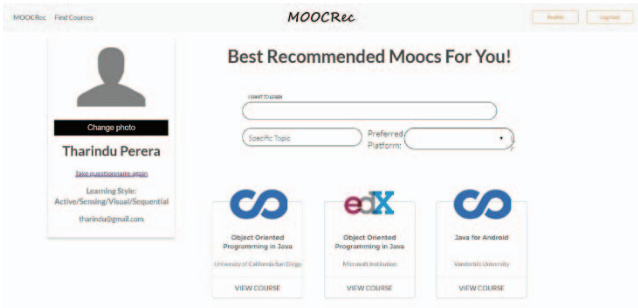

Figure 5: Recommended courses based on learning styles

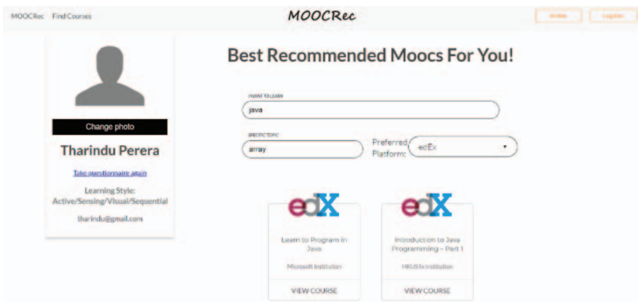

Figure 6: Search recommended courses using parameters

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. Finally, the system recommends the most appropriate MOOCs from different platforms based on the mapping data. The learner can also further filter the courses. Fig. 5 describes the user interface for displaying the list of all recommended MOOCs based on the learner's learning styles. Fig. 6 shows the filtered courses based on learner's search queries. In this case, all Java courses in edX platform which contains "array" as topic are shown.

IV. RESEARCH FINDINGS & RESULTS

The driving principle behind MoocRec is the relationship shown in Table III which shows the mapping of Felder and Silverman learning styles with the video styles used in MOOC videos. The validity and reliablity of the correlation between those two domains has been derived from the literature. However, the accuracy of video styles classification as shown in Table VI and VII is over 93% which accounts for the overall performance of the system. The backend processes of MoocRec system including web scraping, video styles classification and topic modeling were evaluated for performance and their 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 IV and V.

Table IV. Web Scrapper Sample Evaluation 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 V. Web Scrapper Sample Evaluation 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, 200 image frames of three different video styles (talking head, slides, and code) each were used as training set and 40 images as testing set. The VGG16-based classifier model was selected based on the accuracy level (92.5%) as compared to other pre-trained based models. The model was trained with an epoch size of 30 and a batch size of 20. Since the dataset used is small and the new model was built on the top of pre-trained model instead of creating from scratch, the training process was very fast even on CPU. It took an average of 4 seconds per epoch to train and test the given data for each model on a system with 6 GB RAM and Intel Core i5 2.20 GHz processor. The video composition algorithm (after classification) was tested for two sample videos from two different courses. In order to have the same measurable units, the length of videos in minutes was converted to a percentage value. The videos were then passed to the pipeline: splitting into image frames, prediction or classification and composition calculation. It took 11 seconds for both videos with a length of 4 minutes and 54 seconds to complete the pipeline. The actual composition of each video styles contained in the videos was calculated manually through observation. The results of composition computation are shown in Table VI and Table VII.

Table VI. Video Styles Composition Evaluation Sample 1

Platform - Coursera			
Course – HTML, CSS, and Javascript for Web Developers			
Week - 2			
Lecture 12 – Anatomy of a CSS Rule			
Video Style	**Actual Composition (%)**	**Using Algorithm (%)**	**Error (%)**
Talking Head	0	0	0
Code	49.3	52.7	6.9
Slide	50.7	47.3	6.7

Table VII. Video Styles Composition Evaluation Sample 2

Platform - Coursera			
Course – Introduction to Data Science in Python			
Week - 2			
Basic Data Processing with Pandas – Missing Values			
Video Style	**Actual Composition (%)**	**Using Algorithm (%)**	**Error (%)**
Talking Head	62.2	63.5	2
Code	37.8	35.1	7.1
Slide	0	1.3	1.3

The topic modeling component was 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 VIII and IX.

Table VIII. Topic Modeling Evaluation Sample 1

Platform – Coursera			
Course – Object-Oriented Programming in Java			
Content	**Available in the Course**	**Using Topic Modeling Algorithm**	**Error (%)**
Sub-Topics	8	5	37.5

Table IX. Topic Modeling Evaluation Sample 2

Platform - edX			
Course – Introduction to Java Programming – Part I			
Content	**Available in the Course**	**Using Topic Modeling Algorithm**	**Error (%)**
Sub-Topics	14	20	42.85

It took 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.

V. CONCLUSION & FUTURE WORKS

In the world of e-learning, MOOCs have grown as a popular platform for learning, attracting learners of different learning styles. However, because of their different learning characteristics and requirements and the availability of similar abundant 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 provides a new dimension in the field of MOOCs by mapping the standard video styles with the learning styles to provide a unique personalization approach based on which MOOCs are recommended to a learner. Also, it allows the learner to search using abstract topics based on their needs and preferences. Overall, it serves as an easy, effective and efficient learning medium.

The current version of MoocRec is only limited to the Felder and Silverman learning style model. Similarly, MoocRec currently indexes only computer science courses from edX and Coursera platform that. The mapping shown in Table III is only based on literature and needs to be empirically validated.

In future works, the overall MoocRec system can be tested and validated with actual learners based on the MOOCs recommended by the system and how relevant the learner finds it. Furthermore, various other parameters to provide personalized learning experience to the learners in MOOCs domain can be explored. Similarly, other video styles such as, conversation, animation and whiteboard can also be incorporated to include different domains. In addition, other learning style models apart from FSLSM can also be taken into consideration.

ACKNOWLEDGMENT

We would like to convey our sincere appreciation to the administration of Sri Lanka Institute of Information Technology (SLIIT) for providing us with suitable environment and prerequisites to complete this project. We also want to express our gratitude to our kith and kin for their persistent support and understanding upon in making this project successful.

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 & 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. Fasihuddin, G. Skinner, and R. Athauda, "Personalizing Open Learning Environments through the adaptation to Learning Styles," 2014.

[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] J. Reutemann, "Differences and Commonalities – A comparative report of video styles and course descriptions on edX, Coursera, Futurelearn and Iversity," in *European MOOCs Stakeholders Summit*, 2016.

[18] "OpenCV library." [Online]. Available: https://opencv.org/. [Accessed: 13-Aug-2018].

[19] "Building powerful image classification models using very little data." [Online]. Available: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html. [Accessed: 13-Aug-2018].

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv Prepr. arXiv1409.1556*, Sep. 2014.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[23] R. M. Felder, R. M. Felder, and L. K. Silverman, "Learning and teaching styles in engineering education," *Eng. Educ.*, vol. 78, no. 7, pp. 674--681, 1988.

[24] "Index of Learning Styles Questionnaire." [Online]. Available: https://www.webtools.ncsu.edu/learningstyles/. [Accessed: 04-Jun-2018].

[25] B. Ciloglugil and M. M. Inceoglu, "A Felder and Silverman Learning Styles Model Based Personalization Approach to Recommend Learning Objects," Springer, Cham, 2016, pp. 386–397.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[27] "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].

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