Journal of Computing and Communication Vol.1, No.1, PP. 48-64, 2022

An Efficient Content-Based Video Recommendation

Walaa H. Elashmawi* [a,b], Youssef Roshdy* [b], Foad Osama [b], and MennatAllah Hassan[b]

[a]Department of Computer Science, Faculty of Computer Science & Informatics, Suez Canal University, Cairo, Egypt

[b]Department of Computer Science, Misr International University, Cairo, Egypt

*Corresponding authors: Walaa H Elashmawi , Youssef Roshdy [w.hashmawi@cu.edu.eg, youssefroshdy2@gmail.com]

ARTICLE DATA

Article history:
Received 20 Jan 2022
Revised 08 Feb 2022
Accepted 08 Feb 2022
Available online

Keywords:
Feature extraction
Video recommendation
Video streaming
Cold-Start
Sound detection
Dynamic time warping algorithm.

ABSTRACT

In a world full of online videos, it is really hard to find relevant content as the data is simply too much. A recommendation system was created to refine this experience, to match relevant content to an interested user. Most recommending systems use algorithms, calculations, and implicit feedback. These methods are effective unless the video does not have implicit feedback in which the algorithms will mostly fail to get relevant content. This is known as cold-start that affects newly uploaded videos, since they start without any data or user comments. Another problem facing users every day is finding the content they want, because it is dependent on videos having labels or having many user views. Since the search engine's mechanism uses the tags and keywords inserted for the video rather than the actual content in it. In this paper, a recommendation system by content is proposed, the system detects the objects and sounds inside the video, and also adds the feature to search using uploaded scenes or filter scenes based on keyword inputted. More experimental results have been done with various scenarios to demonstrate the effectiveness of the proposed system in terms of video recommendation by content.

1. Introduction

Recently, video consumption is essential for a generation heavily reliant on media as entertainment, a way to keep users entertained for longer periods of time through a new method for video recommendation is required. Two basic aspects are considered by the current recommendation systems, which are the users and items. The system starts making decisions as soon as the items are ranked and depending on their rankings different results are given. A ranking is a relationship between items and each other. Each item will be item will be higher ranked or lower ranked or even equal to the opposing item being compared to it.

That being said, the higher-ranked item is much more considered than the lower-ranked ones. Depending on the total user's behavior and information gained from users on a certain streaming service such as country, age, and viewing history, a recommendation for a given video. This is known by the Collaborative Filtering (CF) [1][2]. It is about top selections of the users that were given a rating.

These ratings are put in comparison with different users using the similarity method for giving out the most suitable recommendations to the user. A classical way of recommending videos is as shown in Figure 1. According to Figure 1, the recommended method relies mainly on user data and video watching history.

One of the most iconic similarity measurements is the cosine similarity. It surpasses most measurements where the angle between the items is measured. Instead of, the straight distance just like when using Euclidean distance measurement, it successfully makes the similarity measurement results much more meaningful and accurate. Also, its accuracy as the common attributes number is used over the possible attributes, and is compared to Jaccard's intersection divided by their union. Concluding similarities, the most suitable one for this kind of comparison is the similarity Cosine similarity. CF's application in real-world systems have been very plenty, for example, Netflix and Amazon [3] use it because of how simple and effective it is. Another method for recommendation content-based filtering, it analyzes the video's content[4] such as consistency in textual information. What differs between both recommendation types is

that CF only uses user rated information in its system, while content-based uses the content inside the videos to give a recommendation based on it.

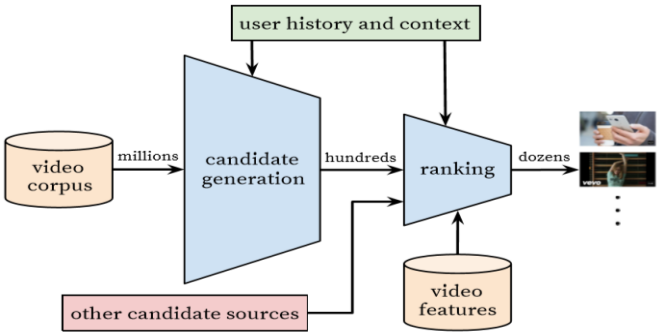

Figure 1: A classical video recommendation method [1]

In [5], a discussion and summarizing for different methods of recommendation and filtering was made and are shown in Figure 2.

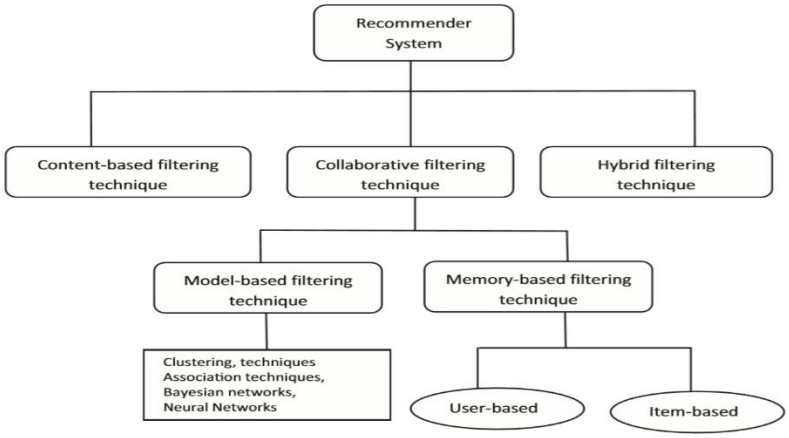

Figure 2: The principles of different recommendation systems [5]

Feature extraction requires analyzing a large number of objects directly from the video frames that were extracted. Briefly, feature extraction algorithms vary based on usage, there are basic requirements such as edge detection algorithms, on the other hand, this can be very complicated and advanced applying it to achieve computer vision. This benefits the system in locating and separating many wanted sections or features an inserted video (frames).

The main need for feature extraction is to get features from the videos and save it as meaning-full data that can be used for multiple applications Tariq et al. [7] discussed different similarity measurements and listed in Table 1.

Table 1: Similarity Measurement Comparison

Similarity Measurement	Description
Cosine Similarity: $cos(\theta) = \frac{A.B}{AB}$	Measure Angle between point A and point B
Euclidean Distance: $d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$	Measure the distance between point p and point q
Manhattan Distance: $d(M,P) = \lvert Mx - Mp\rvert + \lvert My - P\,y\rvert$	Measure the sum of the X-axis and Yaxis from point M to point P

Jaccrad Similarity: $\frac{A \cap B}{A \cup B} = \frac{A \cap B}{\lvert A \rvert + \lvert B \rvert - A \cap B}$	Use the Intersection and Union between two datasets (A and B) to determine the similarity

Most video platforms services use algorithms associated with various calculations and a way to find a video in relation to the user's current one with high content, accuracy isn't available.

In these current times, making the user experience better and raising the watching times are the market's main incentive, especially for video streaming platforms or video search engines.

The purpose of the paper is the enhancement the recommendation of video and improvement of the video classification precision using the cosine similarity measurement to achieve better recommendations when it comes to searching for specific scenes.

This paper is organized as follows. Section 2 discusses the literature review about multiple solutions for video recommendation achieved by other researchers. The proposed recommendation system is illustrated in section 3. Section 4 lists some of the experiments and results achieved for evaluating the proposed recommendation system. The conclusion is drawn in section 5.

2 Literature Review

Nowadays, recommendation systems assume an urgent job in clients' choices. Plenty of studies have been worked on with systems of recommendation as general and on video as special. Subsequently, the point of this area is picking up data about the video proposal techniques.

2.1 Content-based detection review

Li et al. [8] proposed a recommendation system based on using content. This uses a formula to rate objects by measuring video properties that provide just about all the necessary information such as audios, pixels, meta-data, and subtitles. Flawlessly, this data acquired is sufficient to create a relevant table that can compare the user's interesting videos with other videos on different platforms. This is confirmed that the content is relevant to the user's interests.

Yoshida et al. [9] proposed collecting tags and visual and audio features from videos and mixes the semantic and effective information gathered to suggest videos to the user. The tag-based similarity is determined by counting the number of specific tags exchanged between the two different videos. The audio-visual one is used to capture video valence by using color information, which is significantly outperforming that method.

A proposal by Jain et al. [10] for a video recommendation system provides information to the users using WebCrawler (i.e., search engine) and Rating Factor with Neural Network. Content-based Filtering and Collaborative Filtering methods are used to find similar interests between users. Based on the viewers' watching history, the system can suggest videos to the watchers.

Kumar et al.[11] introduced multiple content-based video prediction techniques using multiple content-based video prediction data-set architectures. To make use of the given frame and video-level features to predict similarity to other videos. Using ways like Deep Neural Network and Random Forest Regression on video pairs is solving the cold start problem. The paper concluded that video recommendation results were enhanced by a large margin.

Bhabad et al.[12] proposed a method for lecture video analysis based on the quality of a recording. The method splits a given video to retrieve a frame, and then it uses the Optical Character Recognition (OCR) technology to retrieve keywords from segmented frames.

At the same time, the Automated Speech Recognition (ASR) technique extracts textual metadata from the video's audio track. The system delivers highly accurate ends up in less time of computation.

Zongxian et al. [13] found a solution to the cold start problem. The presented system is based on a Siamese network compared with methods such as collaborative filtering, which is one of the most commonly used methods of video streaming.
They also noted that content-based platforms are also using meta-data (e.g. actors and directors), which presents a challenge for new videos entering the cold-start wall by themselves.

Seko et al.[14] proposed an alternative algorithm to suggest videos for groups rather than individuals, since viewing history and viewer preference take The similarity of the new content to the viewed content is calculated if it exceeds the threshold set for its utility, the video is considered useful to the group.

For a quick recommendation [16] Bviskar et al.[15] used M-distance and Collaborative filtering algorithms. Thus, the aim is to provide users with tailored suggestions to promote the discovery of videos frequently linked to their interests. While the issue triggered the advice to avoid customer interface enhancement and precomputation recommendations production. This is attributable to suggestions for pre-computation that do not represent the recent activities of the consumer. However, they solved the problem by automatically measuring new app preferences for each app on request, and using a program that allows users to watch corresponding content for their interest. This approach is successful in getting the highest satisfaction of the customer. They also considered attributes such as video watch time, mouse hover, and other attributes. So the recommended videos are based on those attributes.

Li [17] surveyed the definition of video retrieval. This proposed a simple scheme for retrieving videos of certain attributes defined on iterated strings. The Hidden Markov Model (HMM) was used to match videos and audio extractions as the primary content-based method. In [18], a method for combining a database with indexes is provided to establish an index structure for getting videos from their content [19-22]. When searching in a broad video database, users can read and understand the content of the video database, as they browse through the correct indexes. Users can search for a particular video by navigating through the relevant indexes. This can be used to further boost the database retrieval system, providing better results for the customer.

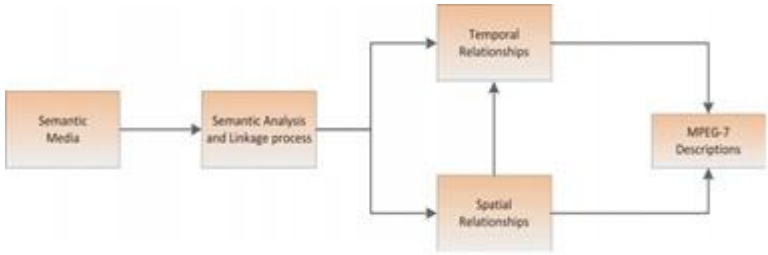

Figure 3: MAC-REALM Stages

A Multimedia Ontology language (M-OWL)[23] was used by Ghosh et al.[24] to map user searches to domain-based concepts. By retreating their search history while using a search engine, the user's preferences are learned and stored. The videos are drawn using content-based choices based on the descriptors for MPEG-7 as shown in Fig. 3. Similar to the current search algorithm present on YouTube. With this approach, the users can be shown better search results by showing them similar videos relative to what they've been watching recently.

Covington et al.[25] tried to solve the cold-start problem by using the advantages of deep convolution neural networks to boost the content-based video recommendations. As a result, the frame-level-vision system declared its improvement over the system of audio information and textual meta-data. And that's going to help users get more content appropriate. The method for retrieving information that already exists is described in [26]. RGB2GRAY and others are used as color space-data for an efficient approach to animation videos. In addition, Euclidean Distance, City Block Distance[27], and Canberra Distance methods are used for comparing the likeness of the searched part of a video with a part of other videos in the database at the time of searching. Their results showed that the RGB2YCBCR algorithm works well and that it is the most efficient compared to the other two.

Parmal et al.[28] applied a MAC-REALM scheme to extract characteristics of syntactic and semantic content[29] which include few user interactions. According to Figure 3, the scheme has four main stages, all of them conclude to serialized content descriptions. A conclusion was drawn that using MAC-REALM, raw media can be transformed into a content model using a method for the extraction and modeling characteristic of content.

Covington et al.[25] tried to solve the cold-start problem by using the advantages of deep convolution neural networks to boost the content-based video recommendations. As a result, the frame-level-vision system declared its improvement over the system of audio information and textual meta-data. And that's going to help users get more content appropriate. The method for retrieving information that already exists is described in [26]. RGB2GRAY and others are used as color space-data for an efficient approach to animation videos. In addition, Euclidean Distance, City Block Distance[27], and Canberra Distance methods are used for comparing the likeness of the searched part of a video with a part of other videos in the database at the time of searching. Their results showed that the RGB2YCBCR algorithm works well and that it is the most efficient compared to the other two.

Parmal et al.[28] applied a MAC-REALM scheme to extract characteristics of syntactic and semantic content[29] which include few user interactions. According to Figure 3, the scheme has four main stages, all of them conclude to serialized content descriptions. A conclusion was drawn that using MAC-REALM, raw media can be transformed into a content model using a method for the extraction and modeling characteristic of content.

Yarmohammadi et al.[30] 's motivation for the work is to automate the multimedia image processing. The main issue in this work is the analysis of the video content. The researchers contributed to the problem by applying systems consisting of three main components including shot boundary detection, hierarchical video summary, retrieval, and target index video.

Deldjoo et al [31] expanded the use of automated methods to extract visual features from images. A new system is developed that is a content-based recommendation to automatically evaluate video content and also extract some stylistic features such as movement, color, and lighting. This research can be extended to fix issues resulting from videos that don't have meta-data. Considering attributes as a video type, the accuracy of the recommendation system will eventually increase. Also, it provides the opportunity to analyze videos in full length.

Bai et al.[32]'s main purpose is to recommend new read articles that meet or match the interests of the user. This is related to video recommendation, as it also suffers from problems of cold start. It may use some algorithms and methods that include several algorithms, such as content-based, collaborative filtering, graph-based, and other hybrid methods. Additionally, some important issues related to

recommendation systems are discussed, such as scalability, sparsity, serendipity, privacy, unified scholarly data standards, and also the issue of cold start.

Chaudhary et al.[33] addressed that most of the algorithms can not manage the cold-start problem because they are freshly uploaded, users who imply circumstances that social media platforms are neglecting to continue to recommend new items to those users. Therefore, users can not easily have similar and relevant data from this report. Thus, using bi-clustering and fusion as a model scheme, they set out a recommendation framework for dealing with the cold-start problem. The device procedures were also oriented to be under the computational environment so that the rating matrix could be simulated. The framework utilizes the process of bi-clustering to combat ranking values and knowledge loss so it uses the technique of fusion and smoothing.

2.2 Audio detection review

Feroze and Maud [34] were looking into the problem of the detection of audio events from scenes gathered from real life. Sound event detection can be summarized into two sections which are monophonic and polyphonic. The problem with monophonic sounds is the removal of the concurrent events from other sound sources. One of the works made on monophonic sounds event detection is the detection of sounds made by firearms[35]. In polyphonic sounds, sound events aren't restricted to just one sound. As in public places, more than one sound event can be heard from multiple sources. So distinguishing between them is still challenging for machines. The researchers of this paper used Perceptual Linear Predictive (PLP) [36] in place of Mel-frequency cepstral coefficients. Comparing between them for the sound event detection problem, the PLP is concluded to give a better performance in comparison to other detection systems. If the suggested feature is used, results are most likely going to get better.

Samireddy et al. [37] implemented a gunshot detection algorithm by using the General Cross Correlation (GCC). They also studied the Sound Pressure Level (SPL) and how far the gunshots were using a diverse selection of guns to see which range is considered acceptable to detect gunshots. An algorithm for shotgun shots detection has been implemented with the usage of the GCC method. It was proven that the detection of gunshots of a diverse selection of guns is possible using the GCC method through the muzzle blast[38] signature of the guns.

Ozdes and Severoglu [39]studied the spectrum detection in further detail by using deep learning. Spectrum detection's goal is to regularly observe a specific frequency band, and report back if a signal exists or not. For spectrum detection, examples of used techniques are energy and matched filter detection [40]. For the deep learning part, they experimented using the CNN, a common architecture for deep neural networks. In their architecture, they had five convolutions' layers referencing from another paper [40]. After experimenting with many parameters, they found the optimal model that efficiently classifies sound spectrum detection. This model has higher performance and scored better accuracies. It was faster in comparison to the older methods being used.

Yue et al. [42] suggested using CNN for audio source detection and estimating the Time Delay of Arrival (TODA) in 3D space, unlike other algorithms that identify the angle of the source in a 2D space. The algorithm the researchers experimented with is founded on the generalized cross-correlation method

with phase transform (GCC-PHAT) [43][44], and by modifying its formula, more than one source of audio can be detected. In conclusion, using mathematical calculations, accuracy could be proved for single sound source direction detection and compatibility of multiple sound sources.

Matsumoto et al [45] there is a problem faces the music which is videos recommendation problem. As most users can't get the music videos, they want without putting suitable queries to find it. They proposed a network construction to deliver the users some music videos similar to their preferences. They do it by using the features and the sounds in the video. As well as social metadata is taken from analysis data [46–49]. Their method proved to be accurate and results obtained from their experiments using real-world datasets proved to be efficient.

Yoshida and Hayashi [50] inserted a music background to videos that haven't already included music, since adding the correct background music will make it the video much more entertaining to watch. They present OtoPittan, a system that suggests music to put in the background for the video. The system uses the desired impression and the expected impression taken from the features. These features are taken from the video and the background music to be suggested. The system appears to give the users satisfying results.

Although the research papers gathered all discuss different techniques for video recommendation, further improvements to the quality of content-based video recommendation are still possible. In summary, these papers used some algorithms to achieve certain goals, by manipulating some ideas and algorithms, we can achieve the recommendation system. Table 2 summarizes the most existing algorithms used for recommending videos.

Table 2: Related work summary table

Used Algorithm	Ref.	year
Using History and Preference from users as Parameters	10	2019
Siamese Network technique	13	2019
Random forest Regression and Deep Neural Network (DNN) on Video Pairs	11	2018
OCR technology and ASR technique	12	2017
M-Distance and Collaborative Filtering (CF) Algorithm	16	2016
Deep Convolutional Neural Network	25	2016
A scheme for extracting syntactic and semantic content (MAC-REALM)	28	2015
Audio-Visual Algorithm Tag-based Similarity Algorithm	9	2013
RGB2Gray and RGB2YCBCR as color space data	26	2012
Shoot boundary detection, Hierarchical video summarization	30	2012
Euclidean Distance, Canberra Distance and City Block Distance	27	2008
Collaborative Filtering and Content-Based Filtering algorithm	6	2008
Web history obtained from the users search using a search engine	23	2007

3 Proposed video recommendation system

The system which we are proposing, mainly contains three phases as described below. The overall system is shown in Figure 4.

3.1 Input phase

First of all, this system is implemented as a plugin tool for ease of use, as it will fit on any chromium-based browser. This plugin has a small GUI menu in which the user can use all of the system's features. The plugin does not require any accounts to use, but more features will be available if the user is logged

in. This system is composed of three major phases. The first phase is how the video is inserted into the system. Simply by using the GUI, the user can easily select and upload any video with the format of MP4. The system also allows the insertion of online links. This menu also can allow the user to see the filtered video and the full version of the non-filtered video. This is one of the features that the user get access while he is logged on. Logging in is also possible using google API which grants google account to be used as the user's plugin account.

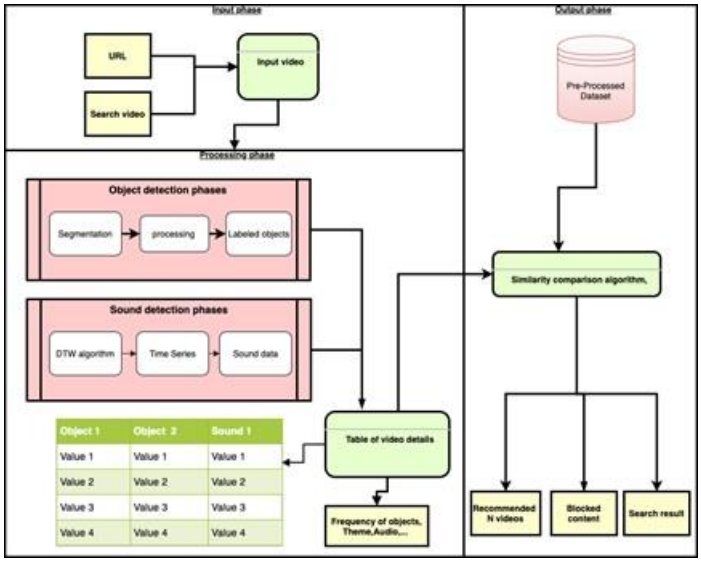

Figure 4: The proposed system overview

3.2 Processing phase

After the video is inserted and uploaded successfully, processing can take place. The first process is passing the video by the object detection algorithms using YOLO implementation. The YOLO implementation is great for object detection as it uses only 5 to 10 percent of the frames in the video, which saves both processing power and time. This phase will result in labeling each object found in the video, these objects will be later on. Sound detection also takes place in the same phase, in which the audio is extracted from the video. The extracted video uses Dynamic Time Warping "DTW" algorithm which compares the two-time series (i.e., the extracted audio), this will compare the temporal distortions between them. By calculation of the distance matrix between time series, the audio file is extracted from the original video inserted. Then it is selected and classified along with multiple classes. While both audio and objects data are being extracted from the videos respectively, The sheet of data will be created. This sheet of data acts, as an ID for the video content as it will be used in the comparison in the third and final phase. In this phase also filtration process takes place, as the objects detected the user can define some objects to be removed for age and safety restrictions. This is crucial as users can enjoy more and worry less about their displayed content and the safety of younger audiences. Table 3 shows more information about the videos ID and how it is used to be compared along with other videos to achieve similarity and relevancy. Based on the Table 3, the video consists of a set of objects. Each object's frequency represents the number of presence of such an object in the video. While the sound's value is either -1 or 0 where -1 means that the object has no related sound and the value 0 indicates that this object has sound detected.

While processing takes place in phase 2, training also takes place. For the training, Open-Label is used to put labels on the images extracted with the preset objects. Then those images are extracted creating a file which has all the names of the objects intended to train the system on. Darknet framework weights

are used to extract a file with the weights from our existing files which gets the results of the training. For testing, weights are taken from the model, the file which has the objects names and the videos we want to pull out the objects from it and its frequencies to use them in the YOLO Object Detection script. The training process can be shown in Figure 5.

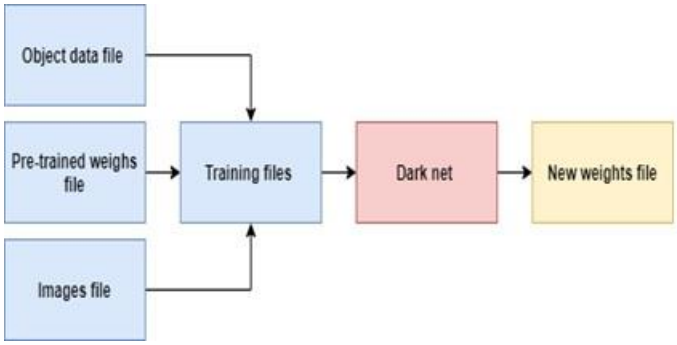

Figure 5: The training process

Table 3: Video Sheet

Video-ID	Object	Frequency	Sound Data
12	person	4990	-1
12	dog	11	-1
12	cell phone	5	-1
12	wine glass	4	-1
12	cat	1	-1
12	bus	2	-1
12	bear	3	-1
12	cow	1	-1
12	gun fire	232132	0
12	violin	2341345	0
12	boat	7	-1
12	truck	1742	-1

3.3 Output phase

The final phase in which the output takes place. To be more specific, the finished processing of phase 2 is being transformed into a proper result in this phase. The sheet representing each video will be used to compare the similarity between the videos according to their content extracted, by Cosine similarity measurement. The comparison of the sheet of data will take place alongside the sheets of data in the database. This database has videos. Each is with their sheet of data also referred to as Video ID. The database is also clustered into categories to makes searching in the database much more efficient. Similar videos are clustered together, so when an inserted video starts getting compared, this video should be compared with a cluster full of similar videos in the same cluster rather than less relevant content outside the cluster. After similarity takes place, the system should output similar recommended videos for the user, the output can be the form of recommended videos, search result and filtered version of the input video. By using the equation (1): A and B are two arrays of dimension n where A represents the objects frequencies of uploaded video and B represents the objects' frequencies of the preprocessed video in the database.

$$Similarity = cos\theta = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i.B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2}\sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad (1)$$

where A_i and B_i are the i-th elements (i.e., frequency of object) of array A and B, respectively. n is the identical number of objects in the video. Table 4 has listed some objects associated with their frequencies in two videos as an illustrative example. Based on equation (1), the relevancy between the two videos is 0.78215.

Table 4: Relevancy Table.

Video 1		Video 2	
Object Name	**Frequency**	**Object Name**	**Frequency**
person	4990	person	9891
truck	1742	truck	468
car	3138	car	4533
motorbike	171	motorbike	17
handbag	4	handbag	62
backpack	13	backpack	24
bicycle	50	bicycle	4
skateboard	17	skateboard	1
train	3	train	12
parking meter	5	parking meter	105
tie	5	tie	1
dog	11	dog	4
cell phone	5	cell phone	33
cat	1	cat	3
bus	3138	bus	798
bear	3	bear	3
cup	1	cup	10
stop sign	4	traffic light	43
umbrella	13	snowboard	7
boat	7	bottle	156
spoon	2	fire hydrant	990
cow	1	tvmonitor	17
wine glass	4	fork	1
-	-	refrigerator	15
-	-	chair	7
-	-	bird	9
-	-	clock	40

The object-frequency table, which is shown below, is an example for how the data might look like. Currently, it is showing some objects and their frequencies respectively.

Table 5: Object-frequency table

Object	Frequency	sound data
Car	1295	0.9
Person	667	0.8
Clock	239	0.67
Cell Phone	2	0.24
Gun	3	0.6667
Glass	13	0.874

4 Experimental results and performance analysis

This section contains some experiments (i.e., scenarios) to ensure the system works as intended. In addition, the proposed system is compared against YouTube recommendation in terms of the relevancy. Also, we investigate the performance of the proposed content-based video recommendation on YouTube-8M benchmark dataset [51]. This dataset was created to make working with computer vision and using popular YouTube content easier. It consists of many categories. These categories are used as labels, to Mark each category with its video contents to make the huge number of videos easier to deal with and easier to navigate through. Also, the dataset being from YouTube making it a realistic example as its one of the most used video Platforms.

4.1 Experiment setup

All experiments are conducted using a YOLO library to process the scenes in each video for object detection. Then, a python script is used to get the objects and accuracy out of the processed videos. For sound detection, the DTW algorithm is used which returns the value of the audio extracted from each audio class (e.g., Gunshots, Laughter, and Telephones).

All experiments are done on Windows 10 operating system and Google chrome. As well, most of the processing load was done on Google Colab for training the dataset. AWS was used as a host server for the whole system and its processing. For creating users and logging into the system (i.e., plugin), a Google sign-in API is used.

In the following subsection, we highlight some scenarios for the proposed working system in addition to screenshots to the GUI of the system.

4.2 The GUI of proposed system

The main menu that will appear upon using the plugin, is shown in Figure 6. The menu gives access to upload video or to enter URL for a specific video. In case of selecting uploading video as shown in Figure 6, a menu appears upon clicking on "upload video". The OS upload window will pop up on the click of the button "choose file", then the upload process will begin, or the user can simply insert the URL for the video. Then, the flow of the system will take place in the background so that the experience is seamless to the user. Whether a user can upload video or insert a URL, the results will appear in Figure 7 (i.e., the recommended Top-N videos).

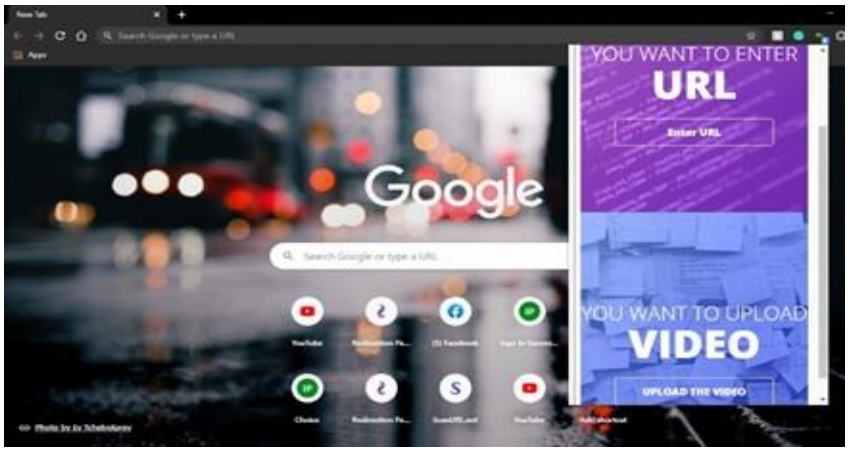

Figure 6: The main interface

4.2.1 Scenario 1

According to the first phase, the scene uploaded by the user had a person and a car, the processing of the video is done according to the diagram. After the phase of detecting objects, the properties of the aforementioned objects (car and person) will be put down by the table of data. Subsequently, similarity

will be found and used, in the previous instance, content that is satisfactory and similar to the scene uploaded containing the car and person should be provided.

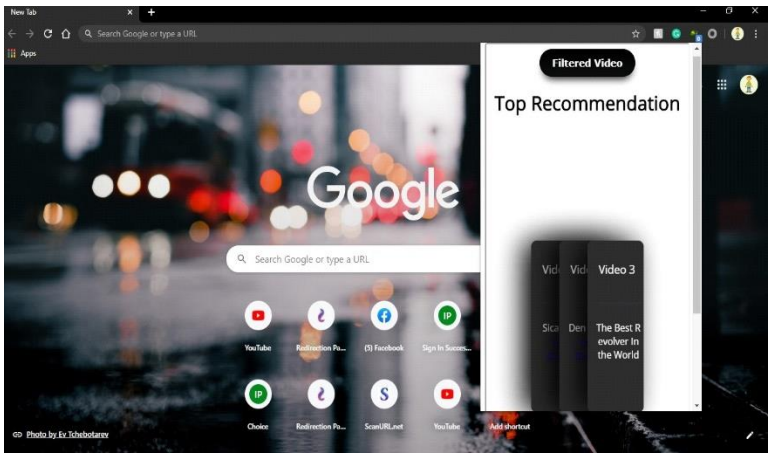

Figure 7: Top N-Recommended

4.2.2 Scenario 2

In phase two, if a scenario is made in which a scene has three cars and two trucks in it, the object-frequency table should contain the data output for the current phase. When looking at the object car, the object truck's frequency will be lower than the object car's. Also, for phase two but this time considering audio, if the audio extracted from a gunshot scene, the audio file should contain the audio sample of a gunshot. this sample will be compared with the audio class of gunshots returning a value for its relevancy

4.2.3 Scenario 3

In phase three, let's take this instance, a video with a person, chair, and a table, the detection algorithm should see the three mentioned things as objects. In the table of data, the objects will appear as an outcome and give you the videos that achieve the highest similarity when compared to existing data. Using another example, we can use a video that has a watch, a computer, and a person. The object computer was inputted in the filtering list, any scene that has the object computer will be obstructed, while the other scenes will be viewed normally.

4.2.4 Scenario 4

In the case of filtering as shown in Figure 8, in this scene, a man was shot with a gun, so this scene should be removed for age restrictions, so the frames which contain the dead man Figure are removed and the video is reconstructed without this scene as shown in Figure 8 below.

Figure 8: the video after

Here is an example shown in Figure 9 that was provided from an earlier demo for the current proposed system. In this case, the frame was taken from a video that has some car in it. Labelling the cars was successful as seen in the figure.

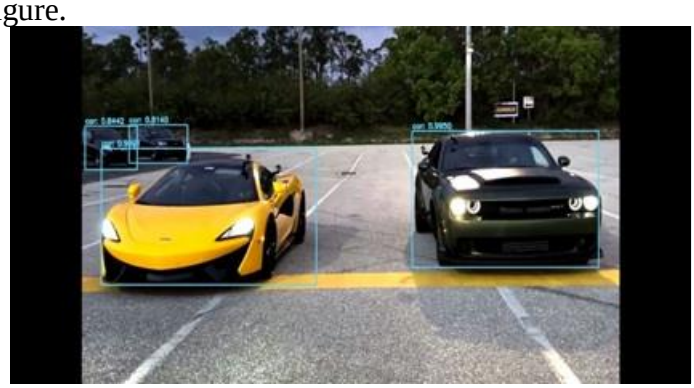

Figure 9: Object labeling

4.4 Performance measure and analysis

In order to validate the accuracy of the proposed recommendation system and compare the accuracy to YouTube recommendation accuracy, Formula (2) can be used to determine how the relevancy results are achieved.

$$Relevancy(\%) = \frac{\sum_{i=1}^{n}((O(i) * f(i)) + SV(i))}{\sum_{j=1}^{x}\sum_{k=1}^{m}((Oj(k) * fj(k)) + SVj(k))} \quad (2)$$

Where n is the total number of objects in the input video or test video, $O(i)$ is the object i extracted from the input video, $f(i)$ is the number of occurrences of object i and for how long it appeared. $SV(i)$ is the attribute value of object i extracted from the sound detection algorithm which reflect the sound relevancy. x is the number of similar videos recommended by the proposed system or the YouTube recommendation and m is the total number of objects in the recommended video j. $Oj(k)$ represents the object k of the recommended video that the algorithm matched with recommended video j along with its frequency value $fj(k)$, and $SVj(k)$ is the matching sound value of object k for video j.

Figure 10 shows the relevancy results of 30 various genre videos that tested on the proposed system and achieved an average accuracy 74.2\%. The Figure illustrated that the accuracy of the recommended results is ranged from 64\% to 84\% with respect to various videos.

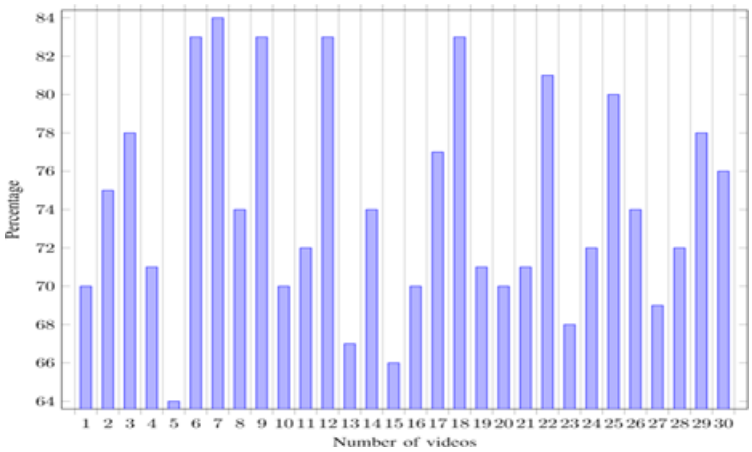

Figure 10: The relevancy of the proposed recommended system

For a fair comparison between the proposed recommended system and the recommended YouTube system, a sample of 10 videos are tested and uploaded to the system as well as watched on YouTube. For a seek of ensuring fair recommendations on both platforms and comparing how relevant are the videos, we recommend to users with the videos that the YouTube system recommends.

Each selected video from YouTube is imported to the system, taking all processing normally and giving eventually some recommendations. Also, while browsing YouTube, we viewed the same video and took its top recommendations. All of the recommendation videos from YouTube and from the proposed system were analyzed as if they are being analyzed for creating the sheet of data. Hence, a sheet of data is created representing their objects respectively. These data can be used in formula (2) for computing the accuracy of recommendation for both systems. Figure 11 shows the results of proposed recommended system and the YouTube system over the 10 tested videos. It has been proved that the proposed system has achieved an average accuracy 69.4 \% while the YouTube overall recommendation system has achieved relevancy of content 62 \%.

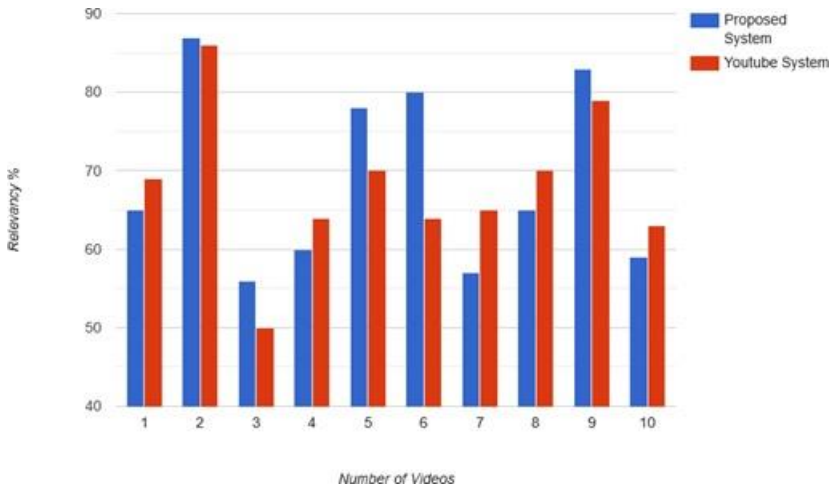

Figure 11: Proposed recommendation system vs YouTube recommendation system

From the above results, the proposed content-based recommendation system has obtained an efficient result in the recommendation process based on the content (objects and sounds).

5 Conclusion and future directions

This paper proposes an efficient content-based video recommendation system. This system is built around extracting content from the video instead of general features such as genre and reviews. While maintaining the simplicity for the user without complications, the system is designed as a plugin for browsers. The system has three major phases, starting by inserting a video using URL or Direct Upload, object, and sound detection to a recommended video and filter videos for being the watching companion as it is designed to be. The proposed system has two-fold. The first is helping the freshly uploaded videos and resolving the cold start problem. And the second is getting a recommendation to a scene from another scene. Some future headings which are proposed, consist of location of more complex objects, more features to be extracted from a given video, and using other similarity measurements. In addition, a large dataset can be tested to ensure accuracy.

References

[1] 1. Xiaoyuan Su and Taghi M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article 493 ID 421425, 19 pages, 2009. https://doi.org/10.1155/2009/421425

[2] Shi, Y., Larson, M., Hanjalic, A. (2014). Collaborative Filtering beyond the User-Item Matrix. ACM Computing Surveys, 47(1), 1–45. 496 doi:10.1145/2556270

[3] Wang, W., Zhang, G., Lu, J. (2015). Collaborative Filtering with EntropyDriven User Similarity in Recommender Systems. International Journal of Intelligent Systems 30(8), pp. 854-870.

[4] 4. Pazzani, M. and Billsus, D. 2007 Content-Based Recommendation Systems. The Adaptive Web. (May 2007), 325-341.

[5] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, Recommendation systems: Principles, methods and evaluation, Egyptian Informatics Journal, Volume 16, Issue 3, 2015, Pages 261-273\

[6] Imani, M., Ghassemian, H. (2014). Feature extraction using partitioning of feature space for hyperspectral images conference om intelligent systems (ICIS).

[7] Tariq, S., Saleem, M., Shahbaz, M. (2019). User Similarity Determination in Social Networks. Technologies, 7(2), 36.

[8] Li, Y., Wang, H., Liu, H., & Chen, B. (2017). A study on content-based video recommendation. 2017 IEEE International Conference on Image Processing (ICIP).

[9] Yoshida, T., Irie, G., Arai, H., & Taniguchi, Y. (2013). Towards semantic and affective content-based video recommendation. 2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW).

[10] Jain, S., Pawar, T., Shah, H., Morye, O., & Patil, B. (2019). Video Recommendation System Based on Human Interest. 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT).

[11] Kumar, Y., Sharma, A., Khaund, A., Kumar, A., Kumaraguru, P., Shah, R. R., & Zimmermann, R. (2018). IceBreaker: Solving Cold Start Problem for Video Recommendation Engines

[12] Bhabad, D. T., Therese, S., & Gedam, M. (2017). Multimedia based Information Retrieval Approach based on ASR and OCR and Video Recommendation System

[13] Zongxian Li1,2∗, Sheng Li1∗, Lantian Xue1,2, Yonghong Tian1,2† 1 National Engineering Laboratory for Video Technology, School of EE&CS, Peking University, Beijing, China 2 Pengcheng Laboratory, Shenzhen, China

[14] Seko, S., Motegi, M., Yagi, T., & Muto, S. (2011). Video content recommendation for group based on viewing history and viewer preference. 2011 IEEE International Conference on Consumer Electronics (ICCE).

[15] Baviskar, P., Gunjal, P., Sirohiya, R., M anwar, S. (2017). A Survey on "User Search Recommendation System for Videos".International Journal of Innovative Research in Science, Engineering and Technology.

[16] Zheng, L., Min, F., Zhang, H., Chen, W. (2016) Fast Recommendations with the M-Distance, IEEE Conference 2016.

[17] N.A., L. (2009). Hidden Markov Model for Content-Based Video Retrieval. 2009 Third Asia International Conference on Modelling & Simulation

[18] Hiwatari, Y., Fushikida, K., & Waki, H. (n.d.). An index structure for content-based retrieval from a video database. Proceedings Third International Conference on Computational Intelligence and Multimedia Applications. ICCIMA'99

[19] William I. Grosky, Multimedia Information Systems, IEEE Multimedia, Spring 1994.

[20] Rune Hjelsvold, Roger Midtstraum and Olav Sandatå, Searching and Browsing a Shared Video Database, Multimedia database systems, Kluwer Academic Publishers 298-317, 1996.

[21] M. Flicker. Query by image and video content: the qbic system. IEEE computer, 23-32, 1995.

[22] Eitetsu Oomoto and Katsumi Tanaka, OVID: Design and Implementation of a Video-Object Database System, A Guided Tour of Multimedia Systems and Applications, IEEE Computer Society Press, 1995.

[23] Mallik, A., Chaudhury, S., Jain, A., Matela, M., & Poornachander, P. (2007). Content Based Re-ranking Scheme for Video Queries on the Web. 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops.

[24] H. Ghosh, S. Chaudhury, K. Kashyap, and B. Maiti. Ontology specification and integration for multimedia applications. In Ontologies, volume 14 of Integrated Series in Information Systems, pages 265–296. Springer, 2007.

[25] Covington, P., Adams, J., Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16.

[26] Kosamkar, P., & Potey, M. (2012). Feature based retrieval for animation video. Proceedings of the International Conference on Advances in Computing, Communications and Informatics - ICACCI '12.

[27] P. Geetha, Vasumathi Narayanan, "A Survey of Content-Based Video Retrieval," IEEE 2008.

[28] }Parmar, M., Angelides, M. C. (2015). MAC-REALM: A Video Content Feature Extraction and Modelling Framework. The Computer Journal, 58(9),2135–2171.

[29] Moens, M.F., Poulisse, G.J. and Vrt, M.M. (2012) State of the art on semantic retrieval of AV content beyond text resources.

[30] Yarmohammadi, H., Rahmati, M., & Khadivi, S. (2013). Content based video retrieval using information theory. 2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP).

[31] Deldjoo, Y., Elahi, M., Cremonesi, P., Garzotto, F., Piazzolla, P., & Quadrana, M. (2016). Content-Based Video Recommendation System Based on Stylistic Visual Features. Journal on Data Semantics, 5(2), 99–113.

[32] Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., & Xia, F. (2019). Scientific Paper Recommendation: A Survey. IEEE Access, 1–1.

[33] Chaudhary, Pankaj and Deshmukh,A.(2015) "A Survey of Content Aware Video based Social Recommendation System." 2015.

[34] Feroze, K., & Maud, A. R. (2018). Sound event detection in real life audio using perceptual linear predictive feature with neural network. 2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST).

[35] C. Clavel, T. Ehrette and G. Richard, "Events Detection for an Audio-Based Surveillance System," 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, 2005, pp.1306-1309.

[36] H. Hermansky and L. A. Cox, "Perceptual Linear Predictive (PLP) Analysis-Resynthesis Technique," Final Program and Paper Summaries 1991 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY,USA, 1991, pp. {0_37-0_38}.

[37] H. Hermansky and L. A. Cox, "Perceptual Linear Predictive (PLP) Analysis-Resynthesis Technique," Final Program and Paper Summaries 1991 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY,USA, 1991, pp. {0_37-0_38}.

[38] R. C. Maher, “Modeling and signal processing of acoustic gunshot recordings,” Proc. IEEE 12th Digital Signal Processing Workshop, Jackson Lake Lodge, USA, 2006, pp. 257-261

[39] Ozdes, M., & Severoglu, B. M. (2019). Sound Spectrum Detection Using Deep Learning. 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT).

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, page 2012.

[41] Tevfik Yucek and Huseyin Arslan. A survey of spectrum sensing algorithms for cognitive radio applications. IEEE communications surveys & tutorials, 11(1):116–130, 2009.

[42] Yue, X., Qu, G., Liu, B., & Liu, A. (2018). Detection Sound Source Direction in 3D Space Using Convolutional Neural Networks. 2018 First International Conference on Artificial Intelligence for Industries (AI4I).

[43] C. H. Knapp and G. C. Carter, “The generalized correlation method for estimation of time delay,” IEEE Trans. Acoustic., Speech, Signal Process., vol. 24, August 1976

[44] B. Kwon, Y. Park and Y. s. Park, "Analysis of the GCC-PHAT technique for multiple sources," ICCAS 2010, Gyeonggi-do, 2010, pp. 2070-2073.

[45] Matsumoto, Y., Harakawa, R., Ogawa, T., & Haseyama, M. (2017). Construction of network using heterogeneous social metadata for music video recommendation. 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE).

[46] C. Guo and X. Liu, “Automatic feature generation on heterogeneous graph for music recommendation,” in Proc. ACM SIGIR Conf. Research and Development in Information Retrieval, 2015, pp. 807–810.

[47] C. Guo and X. Liu, “Dynamic feature generation and selection onheterogeneous graph for music recommendation,” in Proc. IEEE Int. Conf. Big Data, 2016, pp. 656–665.

[48] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He, “Music recommendation by unified hypergraph: Combining social media information and music content,” in Proc. ACM Int. Conf. Multimedia, 2010, pp. 391–400.

[49] S. Kinoshita, T. Ogawa, and M. Haseyama, “A note on quantificationof relationship between users and musical pieces using graph structureanalysis (2): Verification of using link prediction (in Japanese),” ITETechnical Report, vol. 41, no. 5, pp. 31–34, 2017.

[50] Yoshida, T., & Hayashi, T. (2016). OtoPittan: A Music Recommendation System for Making Impressive Videos.

[51] Abu-El-Haija, Sami, et al. “YouTube-8M: A Large-Scale Video Classification Benchmark.” ArXiv.org, 27 Sept. 2016, arxiv.org/abs/1609.08675.