
MIU
Misr International University
جامعة مصر الدولية

An Efficient Content-Based Video Recommendation

by

Aly Mohamed, Amr Sherif, Foad Osama, Youssef Roshdy

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Bachelor of computer science

in

Department of Computer Science

in the

Faculty of Computer Science
of the
Misr International University University, EGYPT

Thesis advisor:
Dr.Walaa H. El-Ashmawi
Eng.Mennat Allah Hassan

(July 2020)

Abstract

In a world full of online videos, it is really hard to find relevant content as the data is simply too much. 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 which happens to freshly uploaded videos in which no data or reviews are available. Another problem facing users every day is finding the desired content is dependant on video being on labelled or has multiple views, as the search engine will find the videos based on keywords or tags, not on the content inside the video. In this paper, a system of recommendation system by content is created, by detecting the objects and sounds inside the video also granting the ability to search or block specific scenes. More experimental results have been done with various scenarios to demonstrate the effectiveness of the proposed system in terms of video recommendation by content.

Index Terms: Video streaming, Cold-start, Video Recommendation, Feature Extraction, Sound detection, Dynamic time warping algorithm.

Acknowledgments

We are thankful for the support and guidance provided to us by Misr International University. We heartily appreciate the efforts carried out by each of Dr.Walaa H.Ashmawi and Eng.Mennat Allah Hassan and we are grateful for the resources and knowledge provided to us by them, also we would like to thank Dr.Ashraf Abd ElRaouf for his guidance and effort. Finally we would like to extend our sincere thanks to our families and friends who have been patient with us throughout these 4 years.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

1.1.1 Background

Video recommendation is crucial in a world consuming media. Most of the existing recommendation systems consider two basic aspects, users and items. Based on the ranking of these items, the decision making can take place. A ranking is defined as the relation between the set of items. For any two given items, the first is either getting higher, lower or equal to the second one. However, the higher-ranked item gets more preferred compared to the lower-ranked items, as higher-ranked items are more relevant than lower ones.

Based on the aggregated user's behavior and his attributes on a video platform such as age, gender, country, and viewing history, a video is recommended. This type of recommendation is called Collaborative Filtering (CF) [1, 2]. It is about which users' preferences have been rated. These ratings are compared with other users according to similarity method to provide suitable recommendations to the user. A classical way of recommending videos as shown in Figure 1.1.

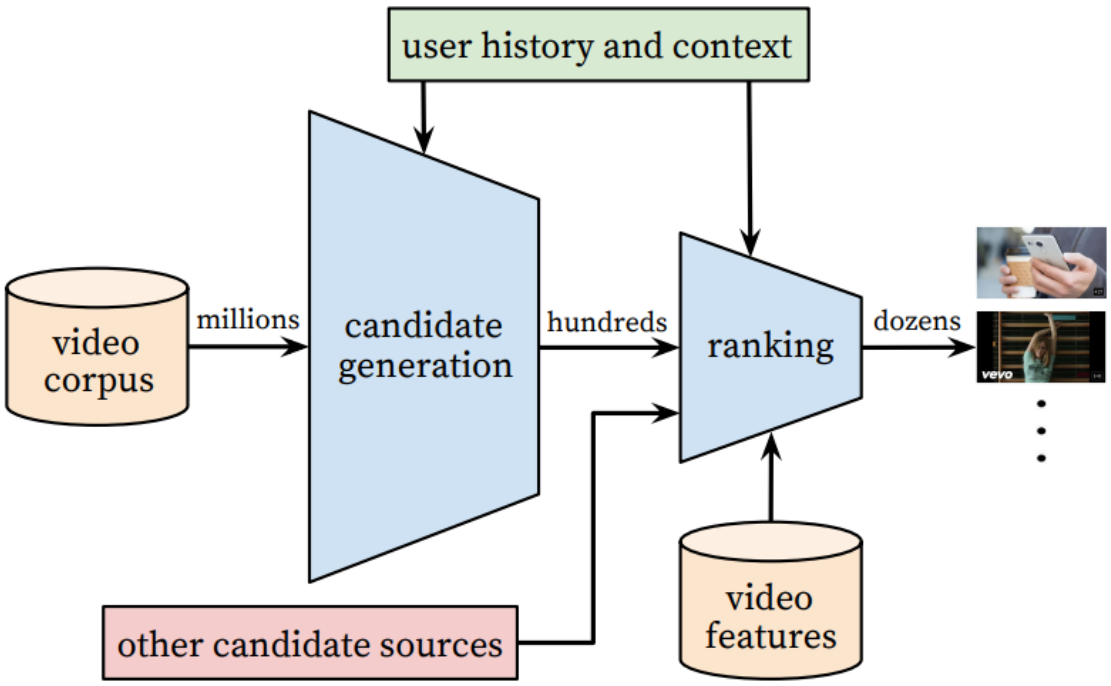

Figure 1.1: A typical video recommendation method

According to figure 1, the recommended method is heavily reliant on user data and video watching history.

One of the most popular similarity measurements is the cosine similarity. It beats most measurements where it measures the angle between the videos, rather than the distance in case of Euclidean distance. Therefore, making the similarity measurement much more accurate in terms of objects included. In addition, it uses the number of common attributes divided by the total number of possible attributes, rather than Jaccard's intersection divided by the union. Therefore, the best-used similarity technique for the proposed recommendation system is the Cosine similarity. CF has been widely used in many real-world systems, such as Netflix and Amazon [3] due to its simplicity and efficiency. Another significant type of recommendation is content-based filtering which based on analyzing the content [4] such as regularities of textual information. The major difference between both recommendations is that CF uses only rating data for a better recommendation, while content-based uses the features of information for a recommendation.

1.1.2 Motivation

Almost all of any video platform service uses algorithms related with numbers and calculations there is no really a way a user can find a related video with the actual content desired, market motivation is going for a video streaming platform or a video search engine.

1.1.3 Problem Definition

Enhancing the video recommendation system and improving the video classification precision with the ability to search specific scenes. By creating a search, filtering and recommendation system for videos which will analyse the content inside the video. The content will be analyses based on objects found in the video, these objects will be labelled. Labels are used as attributes for recommending and searching instead relying on calculations, user's watch history data and direct feedback. Calculations based on user's data are used by almost every video platform which makes our proposed solution unique and reliable enough to solve this problem.

1.2 Project Description

The system should able be detect the objects from the scenes of the video that a user inputs directly or its automatically chosen based on behavior of the user, then the objects in the scene classified and labeled, each object with its own label, from these data gathered, by using algorithms we classify the current scene based on the objects the system detected. The next phase uses feature extraction which each object will be ranked and based on these rankings an overall data for the analysed scene is gathered mainly to determine the genre of the video and matching it with more videos analysed in the database .These matched videos are recommended for the user or it can be used as a search tool for finding related videos in the same genre or having a scene with similar features.

1.2.1 Objectives

The system purpose to increase the accuracy of the videos recommendation to the users on many platforms, So the users will find the more interesting videos to them accurately. also it will save many calculations the platforms uses to recommend a videos to their users.

Also a video search feature will be a available for the users to insert a video and get similar videos as a result.

1.2.2 Scope

The system will work on YouTube Data-set which is the most popular platform also it's supported by an enormous number of videos with different genre. Feature extraction technique will take place to improve the recommendations of the videos to the users.

1.2.3 Project Overview

The proposed system implements a new function for searching by a scene just like a normal search engine. It aims to find similar content from video and output as a search result. Also,a great challenge is introducing a way to block certain scenes based on the custom-built filter, to achieve a clean watching experience. The proposed system overview is shown in figure 1.2. It consists of three main phases. In the first phase, the user can start watching videos normally. The input scene will be inserted using the videos online URL or the user can select a specific video to use as a search query. A video will be imported to be processed in the second phase. During the second phase, object detection and Sound detection 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 can take place. Also taking place in this phase is the object detection phase which will analyse all the objects in the video frame by frame , these objects are labeled. these labels are used to construct the video ID. After processing has occurred, a frequency table for objects has been generated. This table is used to compare the content of the video to the database videos which also has the table of data given. This table is used to create the video ID. By similarity measurements throughout the third phase, results should appear in the form of recommended videos based on highlights from the input video, or in a form of search result, from the user's input. It is also possible to have some scenes filtered and removed from the video based on a filtering created by the user to remove a certain content.

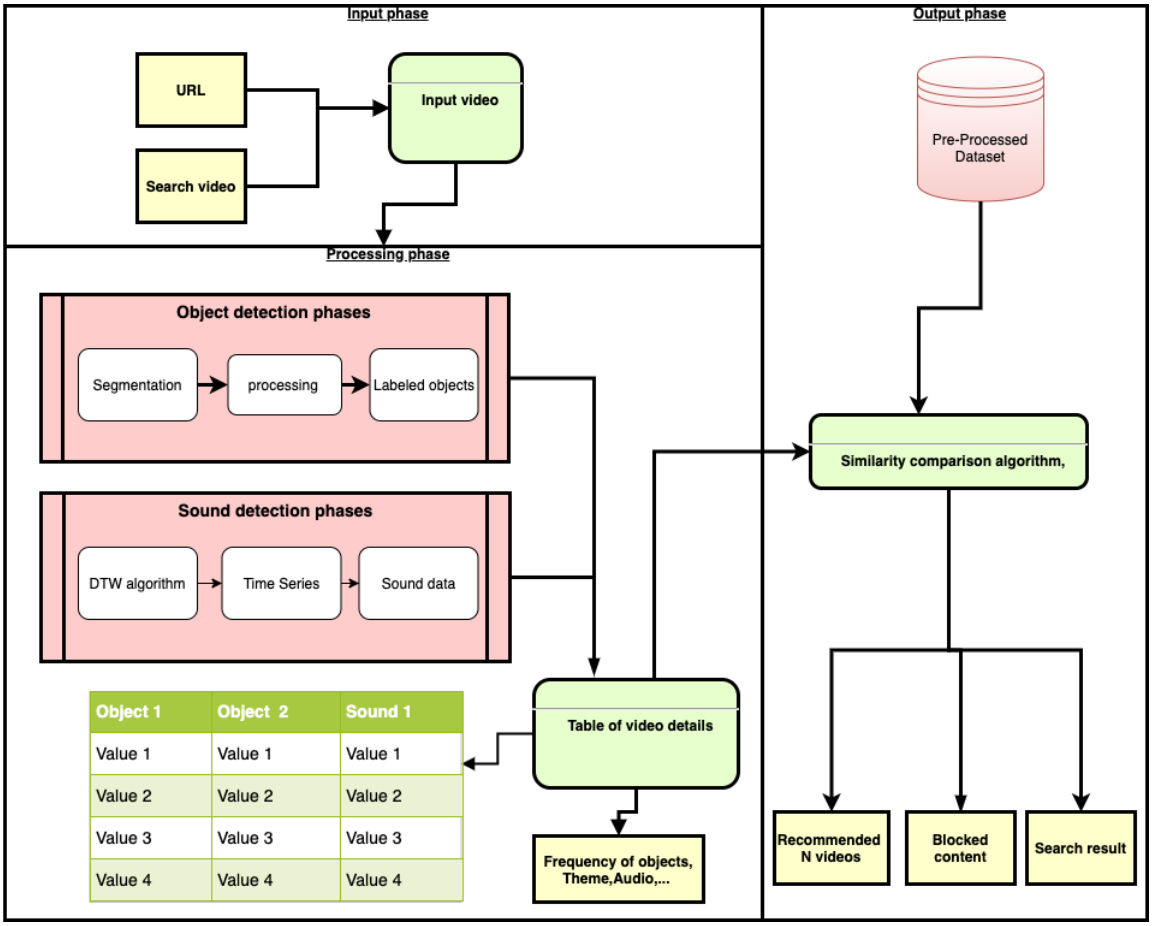

Figure 1.2: Proposed system overview

1.3 Project Management and Deliverable

1.3.1 Task and Time Plan

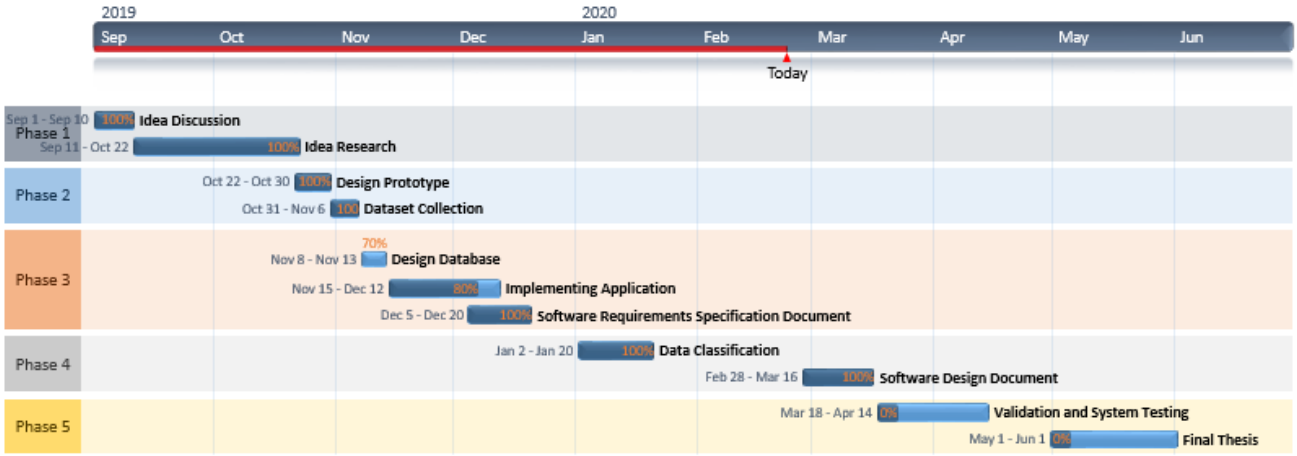

Figure 1.3: Tasks and plans time

1.3.2 Budget

[1] Amazon Sage Maker EU (Paris) Server

Price: Between $0.06 per hour to 0.325$

Reason: An online server to process the videos we need to use in our dataset.

[2] Google Cloud

Price: 74.99 EGP per month

Reason: A cloud storage to store our data-set for future usage in our project, to allow for an online status for the project.

Chapter 2

Literature Work

2.1 Similar System Information

Nowadays, recommendation systems play an important role in users' decisions. A lot of research work has been done on recommendation systems in general and a few of them on video recommendation. Therefore, the aim of this section is gaining information about the video recommendation methods and some feature extraction achievements in this area.

Li et al. [5] proposed a recommendation system based on content. It uses a technique to rank objects by calculating video properties which almost include all the information needed like pixels, audios, subtitles, and meta-data. Ideally, this information gained is enough to generate a relevance table which will be used to compare the user's interesting video with other videos on the platform. This will make sure content is relevant to the user's interest.

Yoshida et al. [6] proposal is to combine semantic and effective information of videos which is extracted from tags and audiovisual feature of videos to recommend videos to the user. The tag-based similarity can be measured by counting the number of common tags shared by two videos. The audio-visual one it uses colour information to capture valence of videos. That method is significantly outperformed.

A video recommendation system is proposed by Jain et al. [7] which provide personalized information using Web-Crawler (i.e., search engine) and Rating Factor Neural Network. It uses Content-based Filtering and Collaborative Filtering to find similar interest among users. Based on viewers' browsing and watching history the system is capable to recommend videos to the users.

Kumar et al. [8] introduced some techniques for the content-based video prediction by employing different architectures on the content-based video prediction data-set to make use of the provided frame and video level features to generate predictions to be similar to the other videos. The paper solves the cold-start problem by using methodologies as Deep Neural Network and Random Forest Regression on Video Pairs. The paper ended with improving the results of video recommendation significantly.

An approach for lecture video analysis is presented by Bhabad et al. [9] that based on the content of the video. A video segmentation applied to retrieve some frame when it has a given video at a specific time interval, then it uses the Optical Character Recognition (OCR) technology to retrieve keywords from the frames. At the same time, the Automatic Speech Recognition (ASR) technique extracts textual metadata from the audio track of the video. The system gives highly accurate results in less computation time.

Zongxian et al. in [10] solved the problem of cold-start by introducing a system based on a siamese network and compared with existing methods like collaborative filtering which is the most common method used by video services. Also, they mentioned that even the content-based system uses meta-data (e.g., actors and directors), it also has problems with new videos which will fall into a cold- start problem.

Seko et al. [11] proposed a new algorithm for recommending videos for groups, not individuals, as the calculations needed are taken by Viewing History and Viewer Preference as parameters. The similarity between new content and watched content is calculated and if the content's useful level exceeds a threshold, they consider the content to be useful to the group.

Bviskar et al. [12] used M-distance and Collaborative filtering algorithms for a fast recommendation [13]. So, the goal is providing personalized recommendations for users to facilitate discovering videos regularly associated with their interests. Though the problem prompted that recommendation to prevent the improvement of user satisfaction and generating recommendations precomputation. This is due to precomputation recommendations that don't fulfil the reflection of the user's recent activities. However, they solved the problem by computing new recommendations for users in real-time for each user's request and use a system that assists the user in choosing a suitable video. This way is effective to get highest user's satisfaction. Additionally, they considered attributes like mouse hover, video-watching time, and other attributes and recommends videos based on these attributes.

A Multimedia Ontology language (M-OWL) [14] is used to map user searches to domain-based concepts by Ghosh et al. [15]. From the web history obtained from the users' search using a search engine for videos, their preferences are learned. The videos are drawn using content-based choices that are based on MPEG-7 descriptors. Similar to YouTube's current search algorithm, by using this method, better search results can be given to the users by giving them similar videos to what they recently watched.

Feroze and Maud [16] 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, monophonic and polyphonic. The problem with monophonic sounds is it removes the chance of concurrent events from other sound sources. One of the works made on monophonic sounds event detection is the detection of sounds made by firearms[17]. In polyphonic sounds, sound events aren't restricted to just one sound, like in public places, more than one sound event can be heard from multiple sources, and distinguishing between them is still challenging for machines. The researchers of this paper used PLP (Perceptual Linear Predictive) [18] 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. [19] 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[20] signature of the guns.

Ozdes and Severoglu [21] studied the spectrum detection in further details, 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. Examples of used techniques for spectrum detection are energy and matched filter detection [22]. 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 [22]. 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 and was faster in comparison to the older methods being used.

2.1.1 Similar System Description

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 II summarizes the most existing algorithms used for recommending videos.

Table 2.1: Related work summary table

Reference	Year	Algorithms Used
10	2019	The Calculations needed are taken by Viewing History and Viewer Preference as Parameters
13	2019	Siamese Network technique
11	2018	Deep Neural Network and Random forest Regression on Video Pairs
12	2017	OCR technology and ASR technique
16	2016	M-Distance and Collaborative Filtering Algorithm
25	2016	Deep Convolutional Neural Network
28	2015	MAC-REALM scheme for extracting syntactic and semantic content
9	2013	Audio-Visual Algorithm Tag-based Similarity Algorithm
26	2012	RGB2Gray, RGB2HSV, and RGB2YCBCR as color space data
30	2012	Shoot boundary detection, Hierarchical video summarization
27	2008	City Block Distance, Euclidean Distance and Canberra Distance
6	2008	Content-Based Filtering and Collaborative Filtering algorithm
23	2007	Web history obtained from the users search using a search engine

2.1.2 Comparison with Proposed Project

In our project, a new method of video recommendation was proposed by comparing the content of the videos inserted with the database provided, as well as fixing the cold-start problem that other systems have. In the papers made by Bai et al. [23] and Chaudhary et al. [24] they provided solutions for this problem, in Bai's paper they used methods and algorithms such as collaborative filtering, content-based and hybrid methods to provide a recommendation similar to what was provided, and in Chaudhary's paper they focused on using the bi-clustering and fusion for their recommendation system, while in our paper

we focused on improving the video classification precision to use it along with the cosine similarity measurement to provide the closest video based purely on the content of the given video. As a part of finding a new method of video recommendation, audio recognition systems were also used alongside the content-based system to give the users the best suitable recommendation. The current audio system takes the sound file of the input video and converts it to a numpy array and then compares it with audio file that is also converted to a numpy array from different classes (e.g. Gunshots, Guitar) using the fast DTW which returns a number that represents how similar the audio file of the input video to the audio file of any class of the classes, while the paper made by Yue et al. [25] had a suggestion to use CNN for audio sources detection as well as using TODA to identify the angle of the source of said audio, which later can be used to detect more than one source of audio using this system.

Chapter 3

System Requirement Specification

3.1 Introduction

3.1.1 Purpose

This document is mainly for the full description of the requirements for the project InVideo Recommendation. This document will explain how the cycle of the system will go on, with the assist of the overview, constraints, functional and non-functional requirements which help this document to illustrate what should the user know and how the user will use the system.

3.1.2 Scope of this document

The project is a plugin tool, which help the users to search by a video they upload and the system will recommend videos which they desired.This system will help people to block or cut some scenes from the video they aren't interesting in it. The user will provide the system with a video as an input; the system will start processing on it, then recommend the most similar videos related to it as the system depend on feature extraction on the video so the resulted will be more accurate to the input one rather than the other platforms which used some calculations and algorithms. The system will give the user the chance to cut some scenes from the video he upload if there are some scenes not desired to him. This software needs internet access.

3.1.3 Overview

The proposed system implements a new function for searching by a scene just like a normal search engine. It aims to find similar content from video and output as a search result. Also,a great challenge is introducing a way to block certain scenes based on the custom-built filter, to achieve a clean watching experience. The proposed system overview is shown in figure 1.2. It consists of three main phases. In the first phase, the user can start watching videos normally. The input scene will be inserted using the videos online URL or the user can select a specific video to use as a search query. A video will be imported to be processed in the second phase. During the second phase, object detection and Sound detection 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 can take place. Also taking place in this phase is the object detection phase which will analyse all the objects in the video frame by frame , these objects are labeled. these labels are used to construct the video ID. After processing has occurred, a frequency table for objects has been generated. This table is used to compare the content of the video to the database videos which also has the table of data given. This table is used to create the video ID. By similarity measurements throughout the third phase, results should appear in the form of recommended videos based on highlights from the input video, or in a form of search result, from the user's input. It is also possible to have some scenes filtered and removed from the video based on a filtering created by the user to remove a certain content.

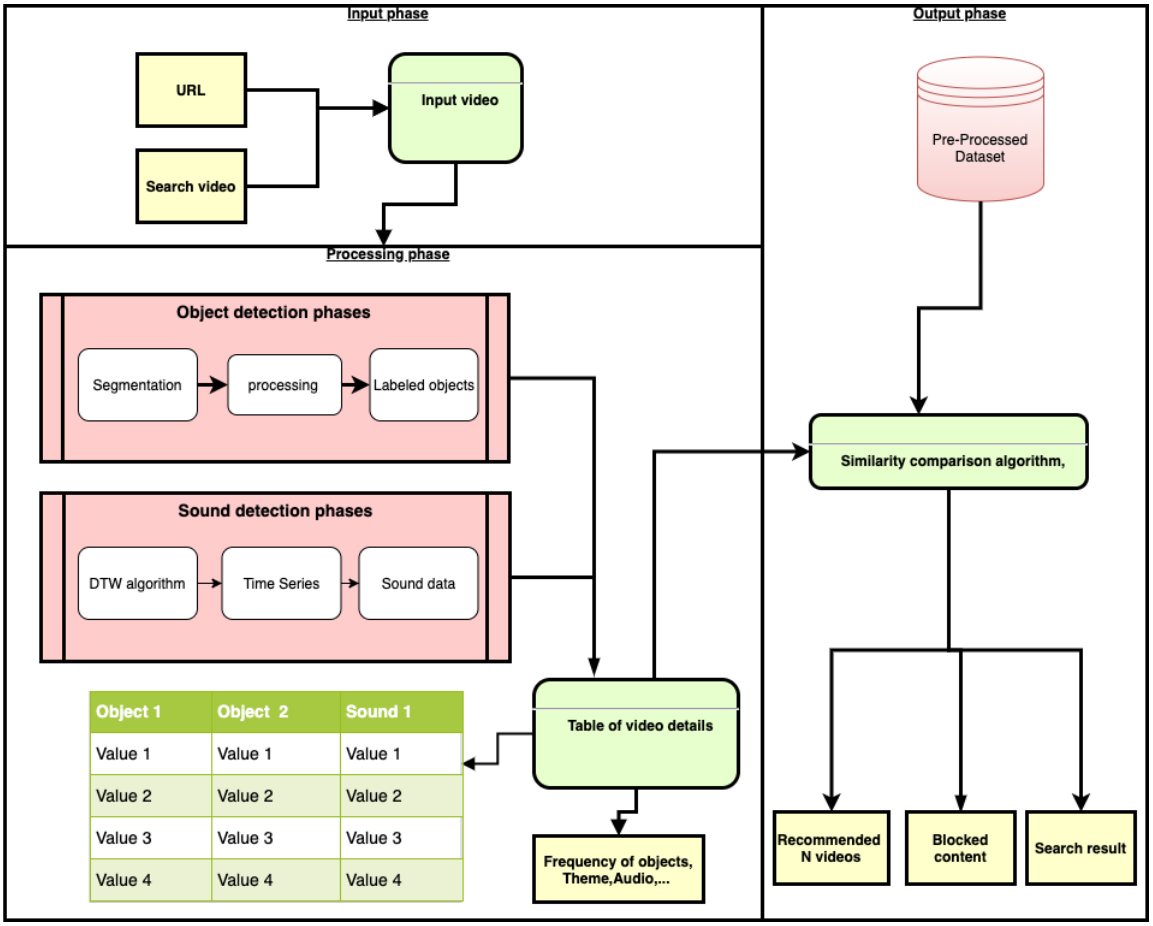

Figure 3.1: Proposed system overview

3.1.4 Business Context

Commercially this will enhance companies search engines (think google search with image), introducing a new way of searching through huge databases of videos. This will upgrade the company's offerings by using this search engine as a service. This will be more illustrated in Figure 3.1.

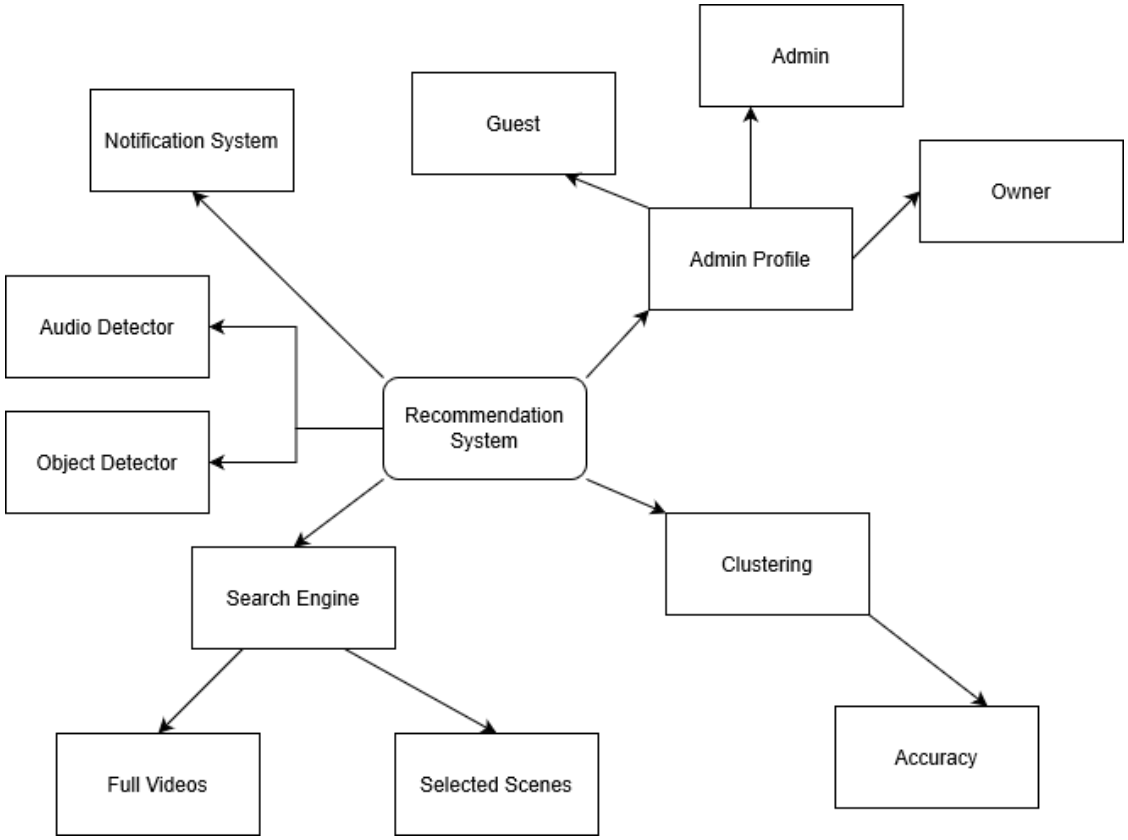

Figure 3.2: Context Diagram

3.2 General Description

3.2.1 Product Functions

- User can upload videos to let the system find similar videos to them
- System can process videos user uploaded
- User can browse history of videos he accessed
- User can search with a video to find related videos

3.2.2 User Characteristics

There is no Specific background for the user to deal with the system, as it shouldn't't be a specific user who can use the system, it's available for all who are interested in this new tool to use is as a new video-search engine. But its important for the user to know how to work on the internet or the web browser. In this system there is the normal user who

uploads a video and waiting for the results which are videos related to the content of the uploaded one. Also a normal user could use it to cut some scenes which are not interesting for him from the video he uploaded.

3.2.3 User Problem Statement

Video platforms uses a lot of calculations and algorithms such as: watch history and search results to recommend videos to their users which in many cases doesn't returns an accurate results especially for the new users or called the cold-start users who doesn't have for example a watch history or even an email on that platform. So, we aim to solve this problems according to detection and classifications done on each video uploaded by the user and this will be done by using feature extraction on a given video so it will give more accurate results rather than the algorithms using by other platforms. This will enhance the video recommendation system and increase the accuracy of the resulted videos.

3.2.4 User Objectives

Users main objective is to have an interested and accurate resulted videos by using this system, but there are some other objective that user will need such as block or cut some uninteresting scenes then start applying it on the video to cut and merge should be fast and accurate. A good interface makes the user satisfied with working on the system, so the good GUI will be important as well.

3.2.5 General Constraints

- 1- Since it is an online plugin, so it needs internet connection for uploading the videos.
- 2- The uploaded videos quality must be good.

3.3 Functional Requirements

Requirement ID	FR1
Name	Signup
Description	It lets the user sign up to start the system functions
Input	Username, first name, last name,email,password,gender, date of birth
Output	Redirect to login page and alert message previewed
Precondition	Form displayed to the users
Post-condition	Account created
Priority	10/10
Expected Risks	Existing username, database access failure
Dependencies	None

Table 3.1: Function Requirement 1

Requirement ID	FR2
Name	Login
Description	It lets the user login with his user-name and password to start the system functions
Input	Username, password
Output	Display account page
Precondition	User must have already signed up into the database
Post-condition	Open profile history
Priority	10/10
Expected Risks	Wrong username or password, database access failure
Dependencies	None

Table 3.2: Function Requirement 2

Requirement ID	FR3
Name	Create user account
Description	It lets the admin creates user
Input	Username, first name, last name, email, password, gender, date of birth
Output	Display account page
Precondition	None
Post-condition	User created
Priority	6/10
Expected Risks	Existing username, database access failure
Dependencies	None

Table 3.3: Function Requirement 3

Requirement ID	FR4
Name	Update user account
Description	It lets the admin edit user data
Input	Change data request
Output	Display updated account page
Precondition	None
Post-condition	User information updated
Priority	6/10
Expected Risks	Database access failure
Dependencies	3.3

Table 3.4: Function Requirement 4

Requirement ID	FR5
Name	Delete user account
Description	It lets admin delete user
Input	Delete request
Output	Display updated account page
Precondition	None
Post-condition	User deleted
Priority	4/10
Expected Risks	Database access failure
Dependencies	3.3

Table 3.5: Function Requirement 5

Requirement ID	FR6
Name	Initiate guest
Description	It gives an id to the user to access the system
Input	None
Output	Redirect to home page
Precondition	Button enter as a guest displayed
Post-condition	Guest created
Priority	10/10
Expected Risks	None
Dependencies	None

Table 3.6: Function Requirement 6

Requirement ID	FR7
Name	Count Frequency
Description	It counts the frequency of each object detected in the given video
Input	Video
Output	Objects' frequencies extracted
Precondition	Frequencies of objects not counted
Post-condition	Frequencies stored in the database
Priority	10/10
Expected Risks	Video corrupted
Dependencies	None

Table 3.7: Function Requirement 7

Requirement ID	FR8
Name	Show Recommendation
Description	It allows the user to see the recommended videos based on the relevance calculated from the similarity
Input	Video
Output	List of recommended videos
Precondition	None
Post-condition	Related videos
Priority	10/10
Expected Risks	No recommended videos displayed
Dependencies	3.7, 3.15

Table 3.8: Function Requirement 8

Requirement ID	FR9
Name	Select Scenes
Description	It allows the user to trim the video and select the specific frames he wants to search with
Input	Video
Output	List of related videos
Precondition	Full video
Post-condition	Video's selected scenes
Priority	7/10
Expected Risks	Video corrupted after trimming, user trimmed the whole video
Dependencies	None

Table 3.9: Function Requirement 9

Requirement ID	FR10
Name	Search With Video
Description	It allows the user to search with a video that he selected to get similar videos
Input	Video
Output	Similar videos
Precondition	None
Post-condition	Related videos to the video searched with
Priority	10/10
Expected Risks	Searched with corrupted video, Searched with image
Dependencies	Count Frequency(3.7)

Table 3.10: Function Requirement 10

Requirement ID	FR11
Name	Show History
Description	It allows the user to select videos from history and search with it to get similar videos
Input	Video
Output	Similar videos
Precondition	History videos
Post-condition	Selected videos to search with
Priority	6/10
Expected Risks	History data not found
Dependencies	3.1, 3.2

Table 3.11: Function Requirement 11

Requirement ID	FR12
Name	Upload video
Description	It lets the user to upload his own video
Input	Video file
Output	Notify the user for the uploading completeness
Precondition	None
Post-condition	Video uploaded successfully
Priority	10/10
Expected Risks	Corrupted file
Dependencies	None

Table 3.12: Function Requirement 12

Requirement ID	FR13
Name	Insert Video Link
Description	Lets the user insert the wanted video URL
Input	Video URL
Output	Notify the success of insertion request
Precondition	None
Post-condition	Video is read by the system
Priority	6/10
Expected Risks	Invalid URL
Dependencies	None

Table 3.13: Function Requirement 13

Requirement ID	FR14
Name	Clear History
Description	This function allows the user to clear history
Input	None
Output	Notify the user of the history clearance
Precondition	History displayed to the user
Post-condition	All of user history is deleted
Priority	6/10
Expected Risks	History data not found
Dependencies	3.1, 3.2

Table 3.14: Function Requirement 14

Requirement ID	FR15
Name	Cosine Similarity
Description	This function calculate the cosine similarity between two videos' objects
Input	Two videos objects and objects' frequencies
Output	Get the cosine similarity between the two input videos
Precondition	None
Post-condition	Cosine similarity is calculated
Priority	10/10
Expected Risks	Calculation error
Dependencies	None

Table 3.15: Function Requirement 15

Requirement ID	FR16
Name	Create Filter
Description	This function let the user creates a filter for mature content
Input	User puts filter boundaries
Output	Filter enabled automatically
Precondition	None
Post-condition	Filtered videos displayed
Priority	10/10
Expected Risks	Filter is not working efficiently
Dependencies	None

Table 3.16: Function Requirement 16

Requirement ID	FR17
Name	Edit Filter
Description	This function let the user edits a filter that was selected
Input	None
Output	Edited filter
Precondition	None
Post-condition	Edited filter applied
Priority	6/10
Expected Risks	Filter is not working efficiently
Dependencies	Create filter(3.16)

Table 3.17: Function Requirement 17

Requirement ID	FR18
Name	Delete filter
Description	This function let the user deletes a filter that was selected
Input	None
Output	Filter deleted
Precondition	None
Post-condition	Deleted filter removed from the filters' list
Priority	5/10
Expected Risks	Filter is not deleted correctly
Dependencies	Create filter(3.16)

Table 3.18: Function Requirement 18

Requirement ID	FR19
Name	Show Accuracy
Description	This function let the admin show the accuracy of the system such as Clustering, Training sets
Input	None
Output	Accuracy percentage
Precondition	None
Post-condition	Display accuracy in a list
Priority	4/10
Expected Risks	Accuracy isn't accurate
Dependencies	None

Table 3.19: Function Requirement 19

3.4 Interface Requirements

3.4.1 User Interface

Our system allows the user to upload videos or insert videos' URL. You can login as admin, user or enter as a guest.The System process uploaded video and then shows the most related videos.

3.4.1.1 GUI

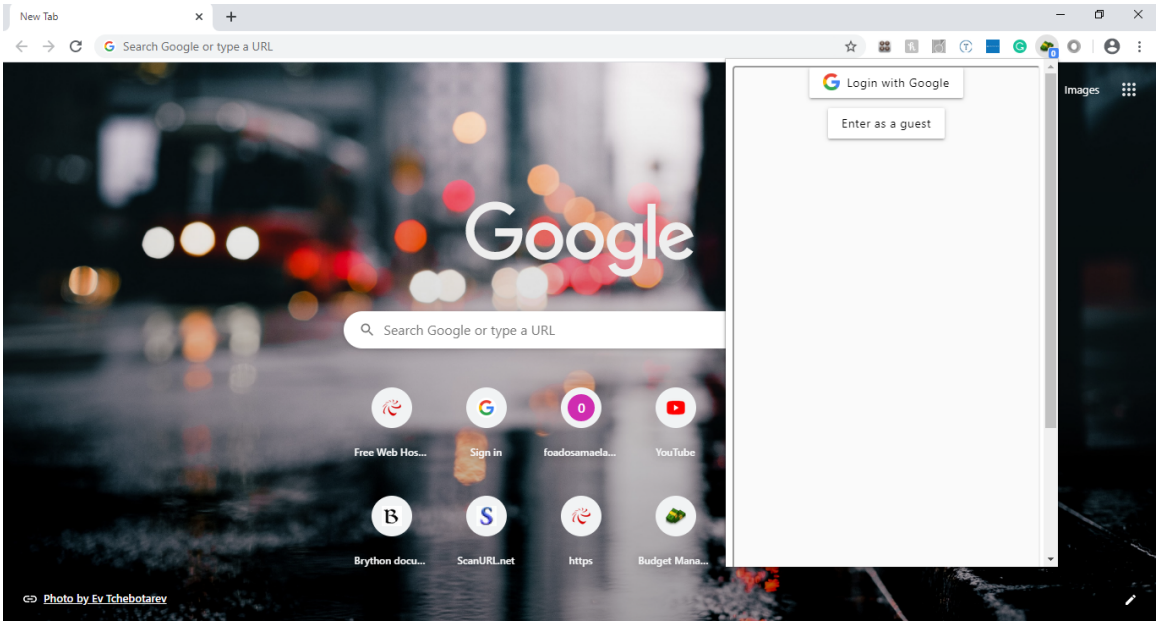

Figure 3.3: SignIn Screen

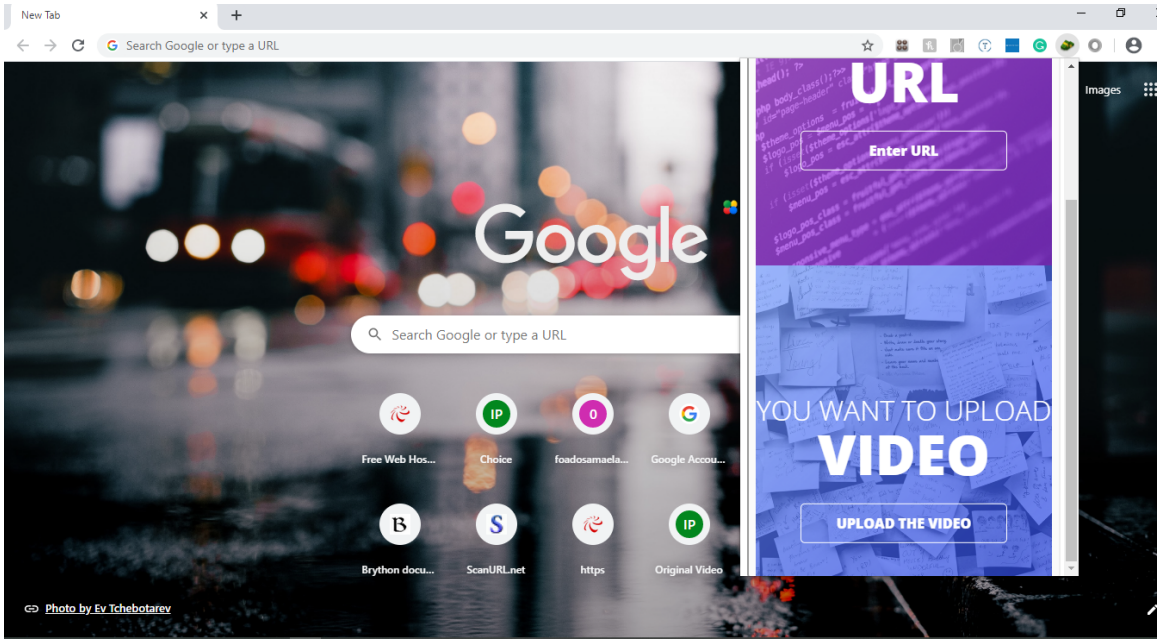

Figure 3.4: Main menu

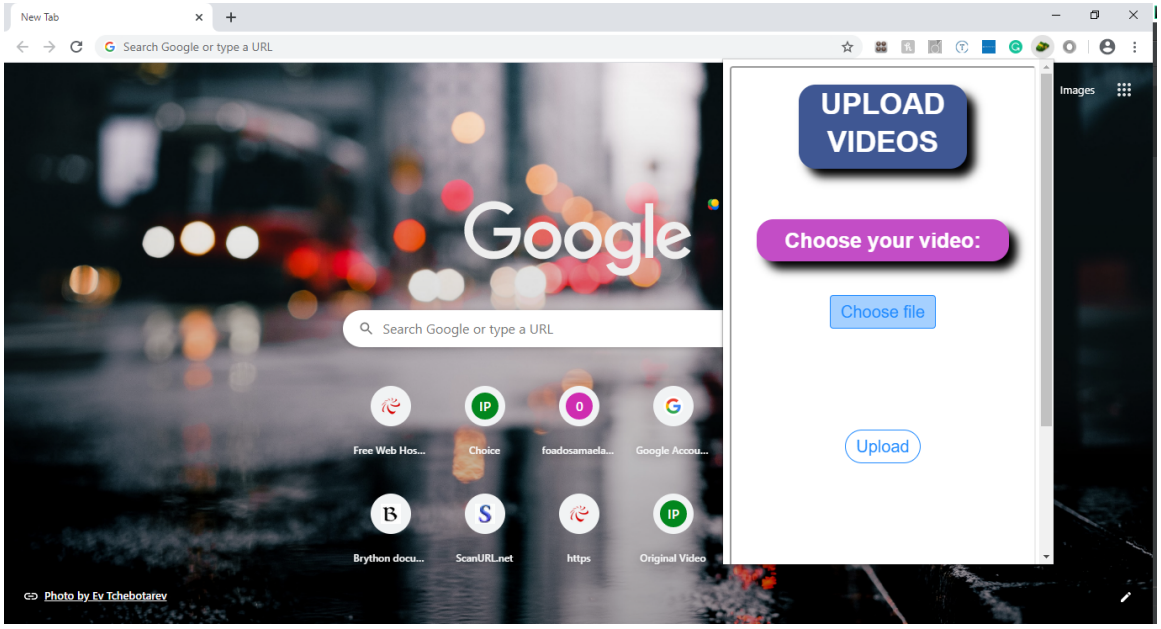

Figure 3.5: Upload/Chooses your Video

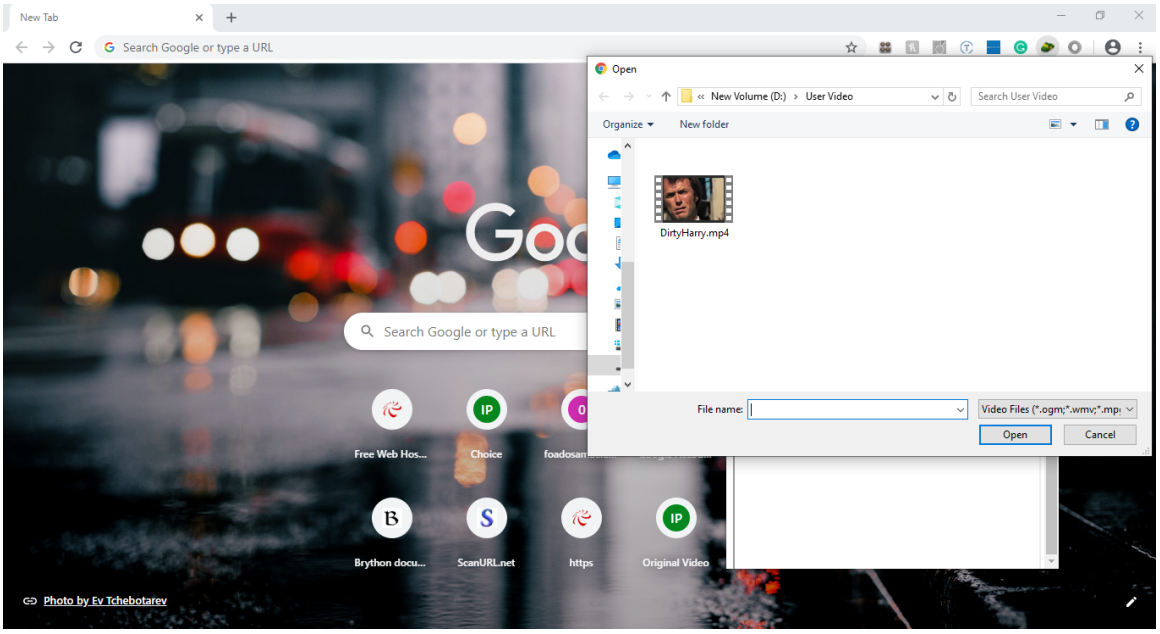

Figure 3.6: Chosen Video

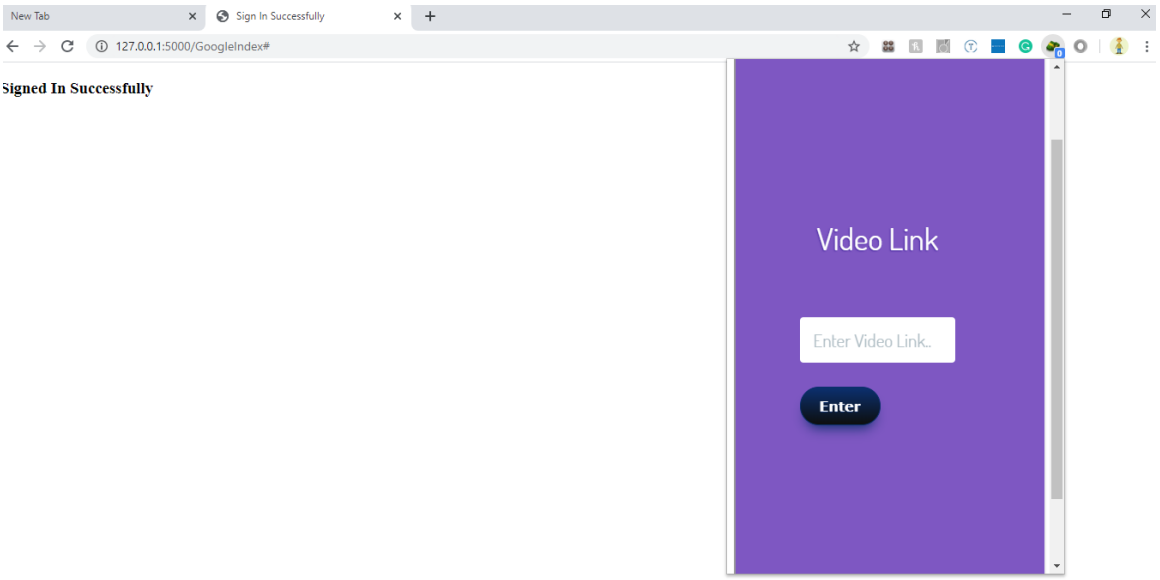

Figure 3.7: Insert Video Link

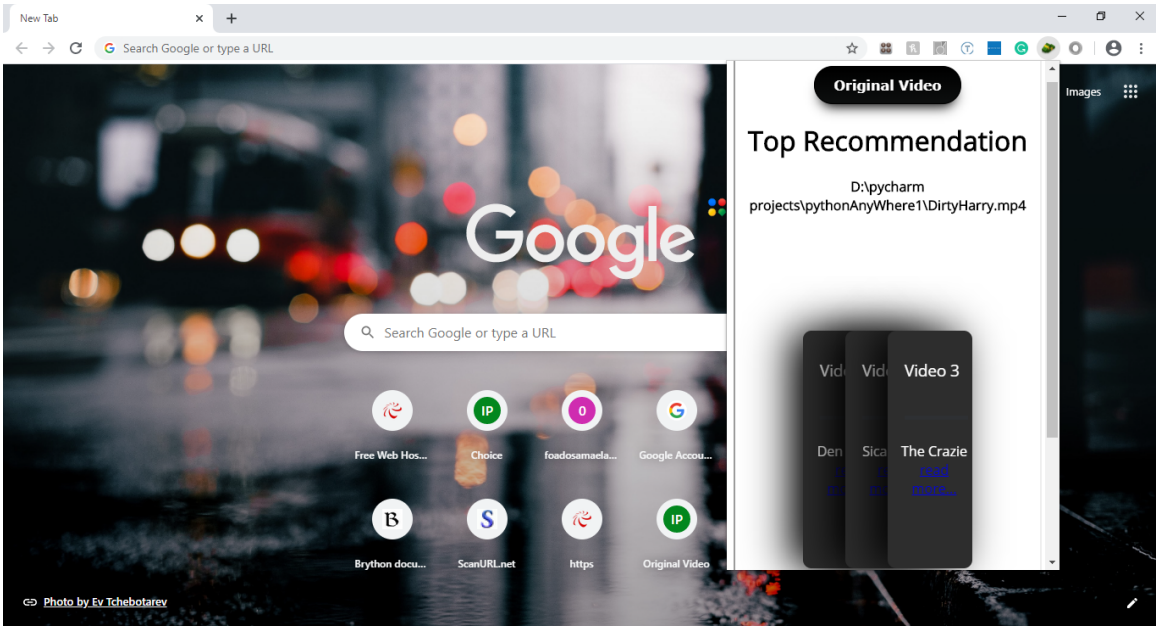

Figure 3.8: Top-N recommendation videos

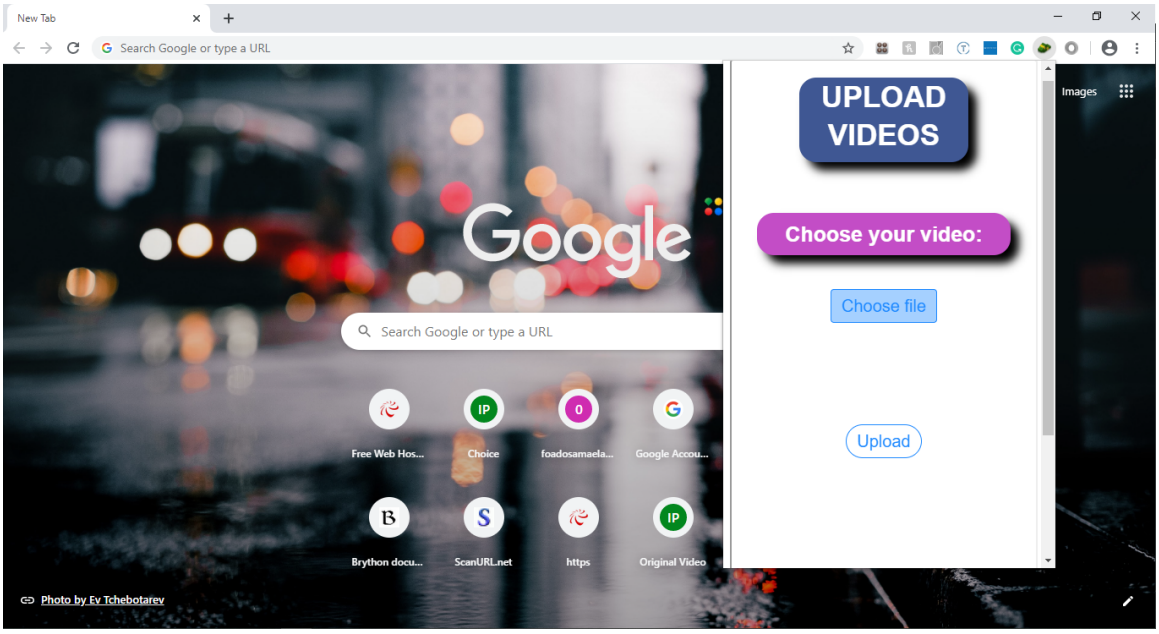

Figure 3.9: Upload/Chooses your Video for Filter Feature

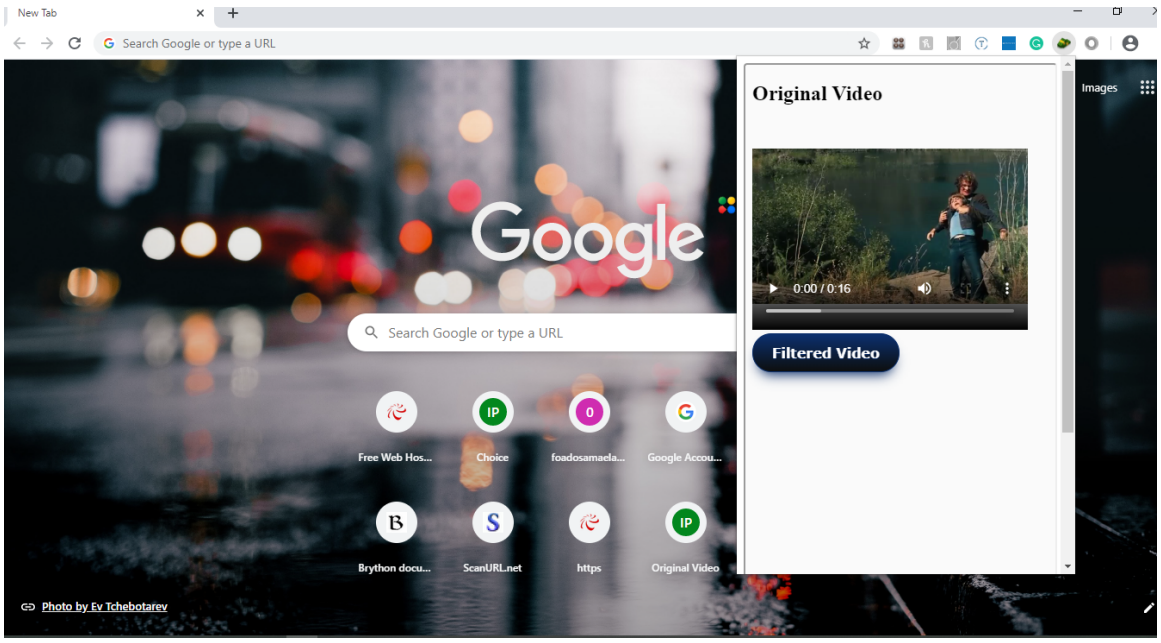

Figure 3.10: Filtration process

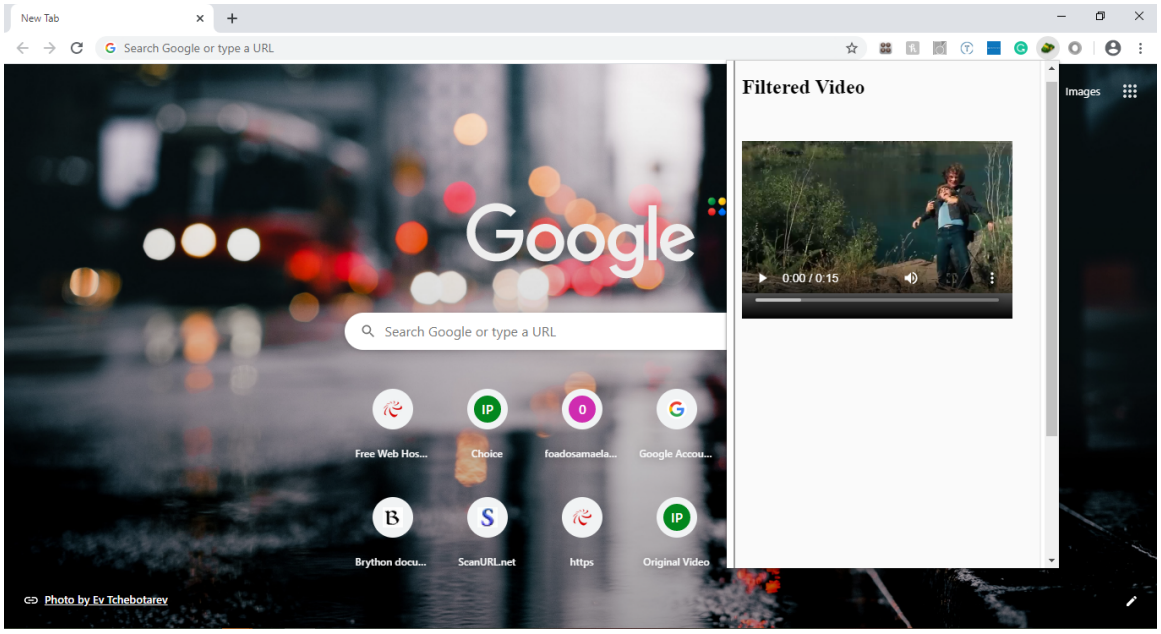

Figure 3.11: Filtered Video

3.4.1.2 API

Google Login API.

3.5 Performance Requirement

- Prepossessing Module compilation time:

File Size (KB)	Video Length (s)	Total time Consumed (s)
13721	97	31448
10649	11	3566
24268	223	72297

Figure 3.12: Prepossessing Module compilation time

3.5.1 Standards Compliance

- 64-bit operating system, x64 based processor.

3.6 Other non-functional attributes

3.6.1 Performance and Speed

The recommender system must have high processing speed and performance, to give the user his video recommendation as with minimum delay.

3.6.2 Reliability

The recommender system is reliable, where it provides the user with the most similar video to the one they provided on the first try, reducing the effort needed for the user to find similar videos.

3.6.3 Scalability

The recommender system is scalable. The more scenes the user searches, the more resources it'll have to recommend in the future.

3.6.4 Security and Safety

The admin panel must be accessed with a password to ensure protection of content.

3.7 Preliminary Object-Oriented Domain Analysis

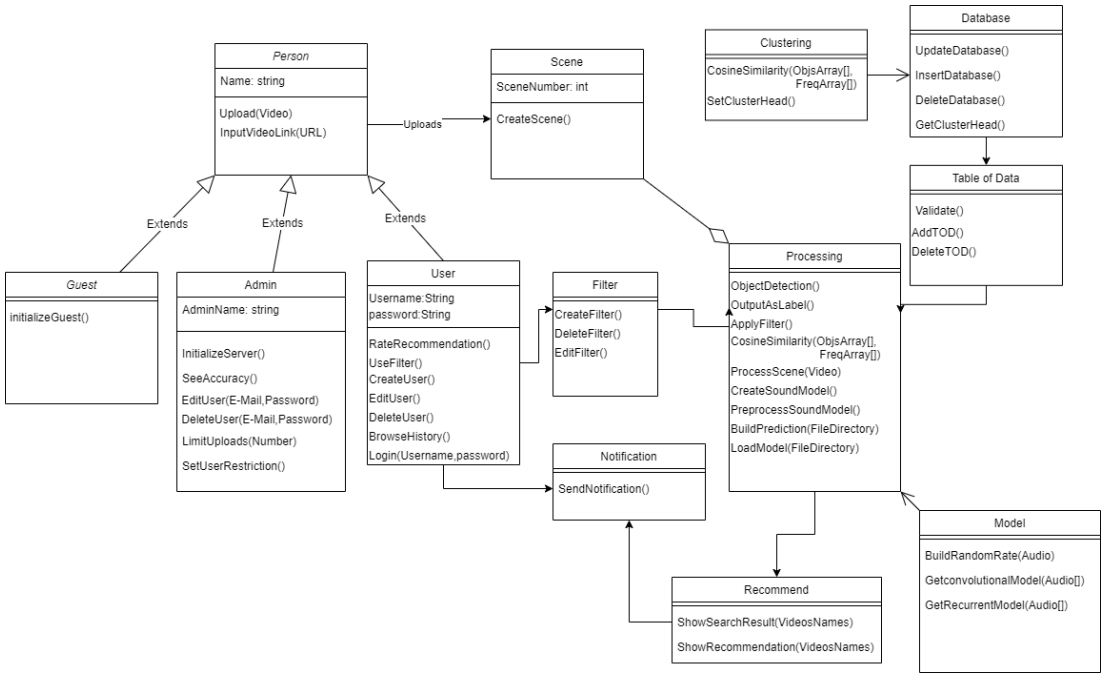

Figure 3.13: Class Diagram

3.7.1 Class descriptions

Class Name	Person
Type	Concrete
List of super classes	N/A
List of sub classes	Guest, Admin, User
Purpose	To generalize all users using this system
Collaboration	This class aggregates class usertype, aggregated by class GUISignUp,Inherited By class Admin and Hematologist
Attributes	Id, Fullname, username, password ,Gender, Mobile Number, Ad-dress, age, usertype object
Operations	SignIn(username, password)

Table 3.20: Person Class

Class Name	Guest
Type	Concrete
List of super classes	Person
List of sub classes	N/A
Purpose	To be able to interact with the system without logging in
Collaboration	This class Inherits class Person
Attributes	d, Fullname, username, password, Gender, Mobile Number, Ad-dress, age, usertype object
Operations	SignIn(username, password)

Table 3.21: Guest Class

Class Name	Admin
Type	Concrete
List of super classes	Person
List of sub classes	N/A
Purpose	To allow the developers to control the system and see how the system is performing
Collaboration	This class Inherits class Person
Attributes	N/A
Operations	InitializeServer() / SeeAccuracy()

Table 3.22: Admin Class

Class Name	User
Type	Concrete
List of super classes	Person
List of sub classes	N/A
Purpose	To be able to login, see his history, create or alter the filter and rate recommendation, and also has the privilege to edit his account
Collaboration	This class aggregates class Filter, Inherits class Person
Attributes	Username, Password
Operations	Login(Username, Password) / RateRecommendation() / UseFilter() / CreateUser() / EditUser() / DeleteUser() / BrowseHistory()

Table 3.23: User Class

Class Name	Scene
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This class is to represent the scene when it's uploaded as an object
Collaboration	This class aggregates class Processing, is aggregated by class Person
Attributes	SceneNumber
Operations	CreateScene()

Table 3.24: Scene Class

Class Name	Processing
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This class is the main class that carries and represents all the algorithms used
Collaboration	This class aggregates class Recommend, aggregated by class Table of Content and Scene and Filter
Attributes	N/A
Operations	ProcessScene() / ObjectDetection() / OutputAsLabel() / ApplyFilter() / CosineSimilarity()

Table 3.25: Processing Class

Class Name	Database
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This class represents the database containing all of the table of content's data
Collaboration	This class aggregates class Table of Content
Attributes	N/A
Operations	UpdateDatabase() / InsertDatabase() / DeleteDatabase()

Table 3.26: Database Class

Class Name	Table of Content
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This class represents the actual table of content as an object which contains the data expected from the video
Collaboration	This class aggregates class Processing, aggregated by class Database
Attributes	N/A
Operations	Validate() / AddTOC() / DeleteTOC()

Table 3.27: Table of Content Class

Class Name	Recommend
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This is the class which displays and outputs the results after being processed by the algorithms either it's recommended list of movies all filtered scenes
Collaboration	This class aggregates class Notification, aggregated by class Processing
Attributes	N/A
Operations	ShowRecommendation() / ShowSearchResult()

Table 3.28: Recommend Class

Class Name	Notification
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This is the class that sends the notification to the user
Collaboration	This class aggregates class Recommend and User
Attributes	N/A
Operations	SendNotification()

Table 3.29: Notification Class

Class Name	Filter
Type	Concrete
List of super classes	N/A
List of sub classes	N/A
Purpose	This is the class used to contain the filter for future application while processing scenes
Collaboration	This class aggregates class Processing, aggregated by class User
Attributes	N/A
Operations	CreateFilter() / DeleteFilter() / EditFilter()

Table 3.30: Filter Class

3.8 Preliminary Operational Scenarios

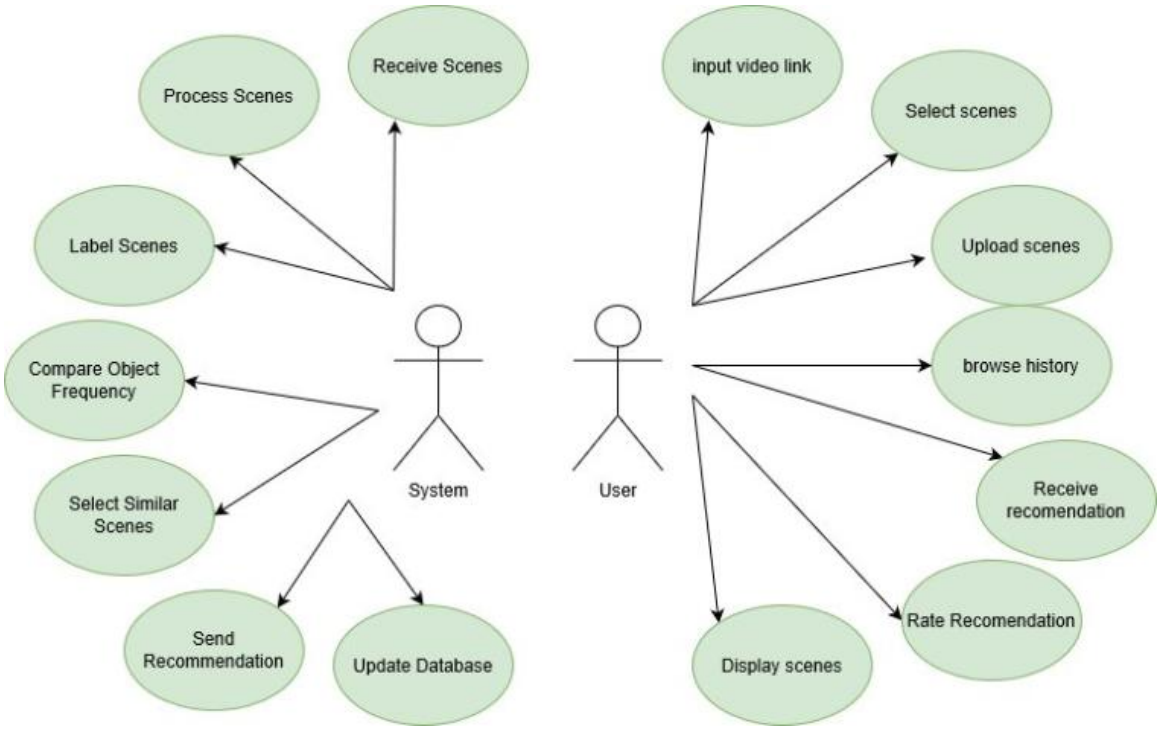

Figure 3.14: Usecase

3.8.1 System Scenario

The system receives scenes from the user, it gets processed into many frames before going through the frames and labeling the identified objects. Frequency for the objects

and taken and compared to the database with other processed scenes or movies. The ones with the highest similarity to the scene received get selected, and sent to the user as a recommendation, before sending the processed scene into the database for future usage.

3.8.2 User Scenario

In this system, the user can send his videos by either putting the link for the video he was watching, or by uploading a video he has downloaded on his device. The user can select the scenes he wants from the videos as well, which then the system replies with some recommendations, that the user can rate to enhance his experience in the future. The user also can check his history to see what he searched for and what was recommended to him, and can choose to display said recommended scenes as well.

Chapter 4

Software Design Document

4.1 Introduction

4.1.1 Purpose

This document is mainly for the full description of the requirements for the project InVideo Recommendation. This document will explain how the cycle of the system will go on, with the assist of the overview, constraints, functional and non-functional requirements which help this document to illustrate what should the user know and how the user will use the system.

4.1.2 Scope

The project is a plugin tool, which help the users to search by a video they upload and the system will recommend videos which they desired.This system will help people to block or cut some scenes from the video they aren't interesting in it. The user will provide the system with a video as an input; the system will start processing on it ,then recommend the most similar videos related to it as the system depend on feature extraction on the video so the resulted will be more accurate to the input one rather than the other platforms which used some calculations and algorithms. The system will give the user the chance to cut some scenes from the video he upload if there are some scenes not desired to him. This software needs internet access.

4.1.3 Definitions and Acronyms

Term	Definition
YoloV3 / Darknet	Used for Training and Detecting objects in videos design information
Cosine Similarity	Used to compare the content of the videos
Spectral Clustering	Used to split videos into separate groups with their similar videos
Convolutional Neural Network	Used for Training and Detecting type of Audio Files
Recurrent Neural Network	Used for Training and Detecting type of Audio Files

Table 4.1: Table of Definitions

4.2 System Overview

The proposed system implements a new function for searching by a scene just like a normal search engine. It aims to find similar content from video and output as a search result. Also,a great challenge is introducing a way to block certain scenes based on the custom-built filter, to achieve a clean watching experience. The proposed system overview is shown in figure 4.1. It consists of three main phases. In the first phase, the user can start watching videos normally. The input scene will be inserted using the videos online URL or the user can select a specific video to use as a search query. A video will be imported to be processed in the second phase. During the second phase, object detection and Sound detection 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 can take place. Also taking place in this phase is the object detection phase which will analyse all the objects in the video frame by frame , these objects are labeled. these labels are used to construct the video ID. After processing has occurred, a frequency table for objects has been generated. This table is used to compare the content of the video to the database videos which also has the table of data given. This table is used to create the video ID. By similarity measurements throughout the third phase, results should appear

in the form of recommended videos based on highlights from the input video, or in a form of search result, from the user's input. It is also possible to have some scenes filtered and removed from the video based on a filtering created by the user to remove a certain content.

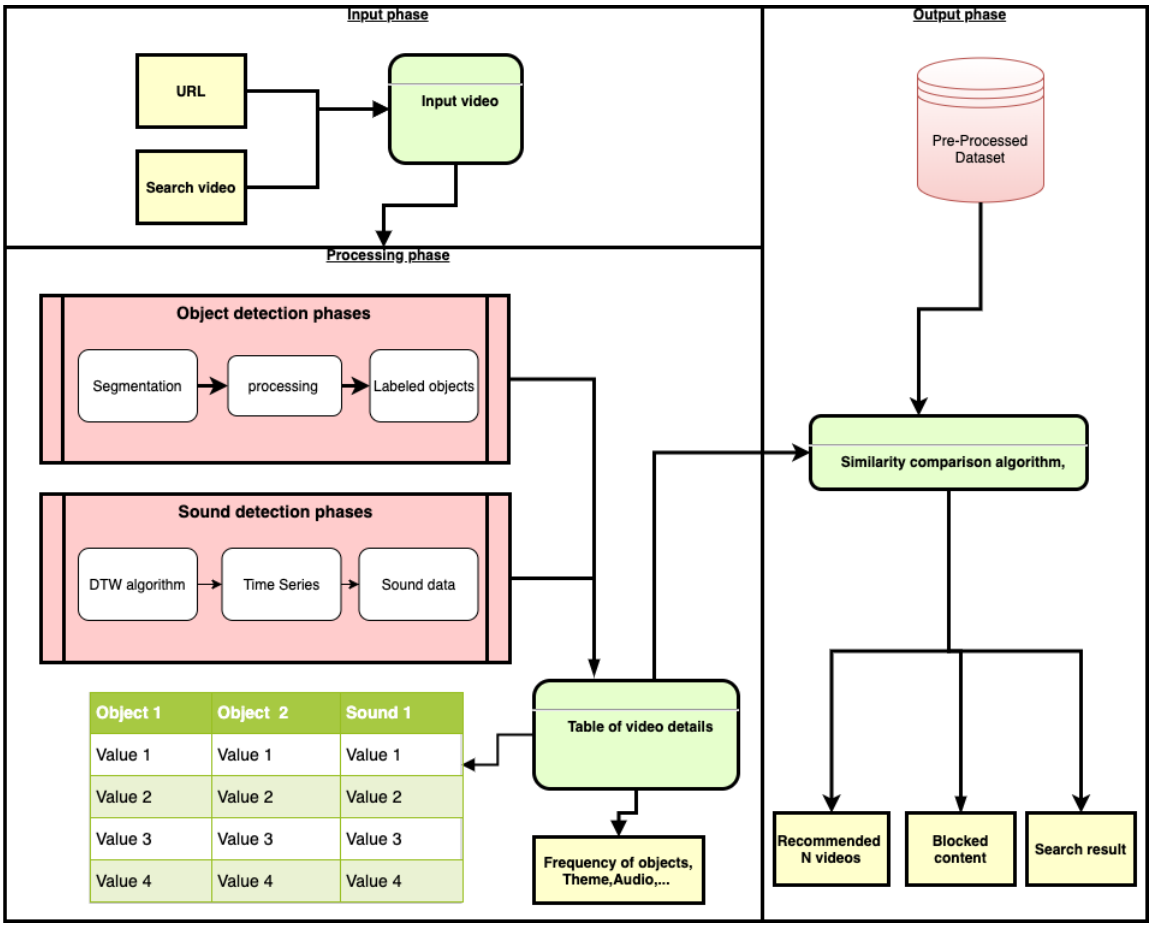

Figure 4.1: Proposed system overview

4.2.1 Dataset

This dataset [26] was created to make working with computer vision and using popular YouTube tube content easier , this Dataset 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

4.2.2 Processing Phase

After the video 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 at 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 4.2 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, 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.

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

Table 4.2: Video Sheet

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 objects preset. Then those images are extracted creating a file which has all the names of the objects intended to train the system on. Using Darknet framework weights to extract a file with the weights from our existing files to get 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 4.2.

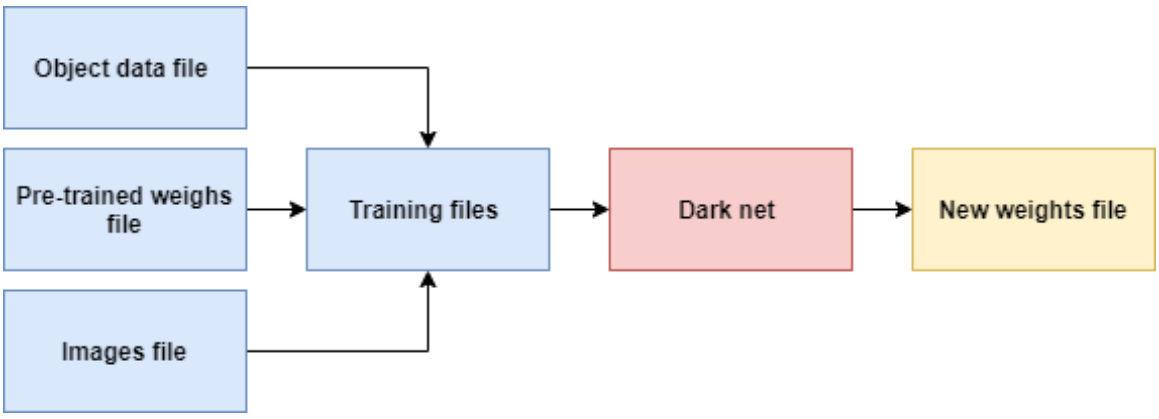

Figure 4.2: The training process

4.2.3 Classification

The classification stage is taking a place after the features extraction of data. For classification we use cosine similarity which are used to put the input video in the class. Cosine similarity is used due to item-frequency usage. Moreover, Spectral Clustering is used all the videos classes which the classification chooses between them. By using a cosine similarity measurement according to equation (4.1): 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.

$$Similarity = cos\theta = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} A_i.B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{4.1}$$

where (A_i) and (B_i) are the ith elements (i.e., frequency of object) of array A and B, respectively. n is the identical number of objects in the video.

4.3 System Architecture

4.3.1 Architectural Design

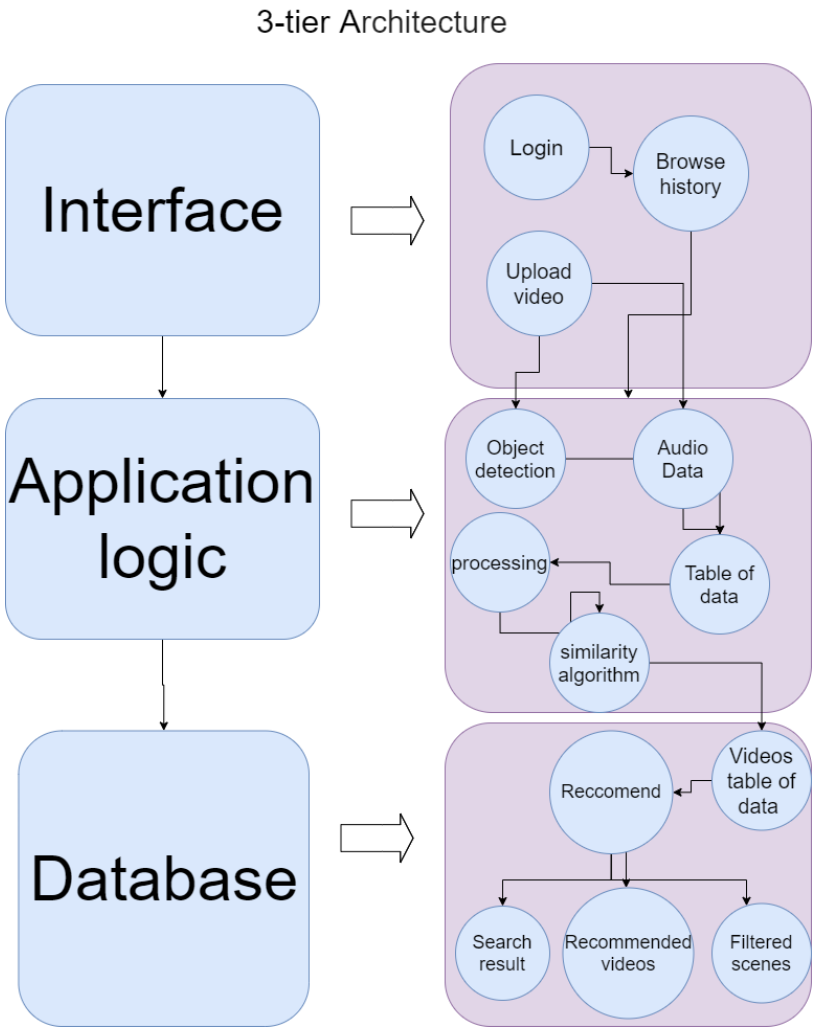

Figure 4.3: Architectural Design

4.3.2 Decomposition Description

Using 3-Tier architecture As shown in figure 4.3, Moving from top to bottom, Starting with interface, which is the plugin tool attached to any chromium based browser. The tool allows the users to upload videos or insert video's URL to get recommendation or search result, Also available in the plugin the ability to view and retrieve history. Phase two starts

after the video is successfully uploaded the processing takes place in then application logic phase which contains all the algorithms required. The objects and the sounds in the video are used to create the sheet of data which will be used to compare the video relevancy with other videos. Finally The last phase takes place which will compare the sheet of data generated from the video inserted with the sheets of data stored into the database using cosine similarity. The results can be displayed to the user in form of three different outputs, recommended videos, search result or video after being filtered from the custom built filter made by the user earlier from phase 1.

4.3.2.1 System Activity

Figure 4.4: System Activity Diagram

4.3.2.2 System Sequence

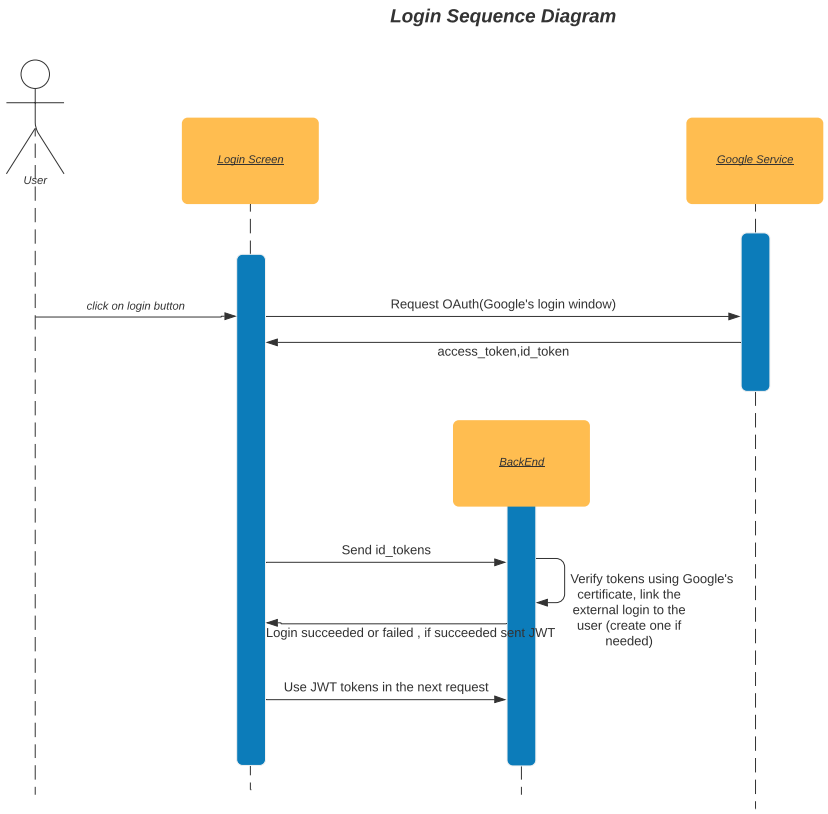

Figure 4.5: Login Sequence Diagram

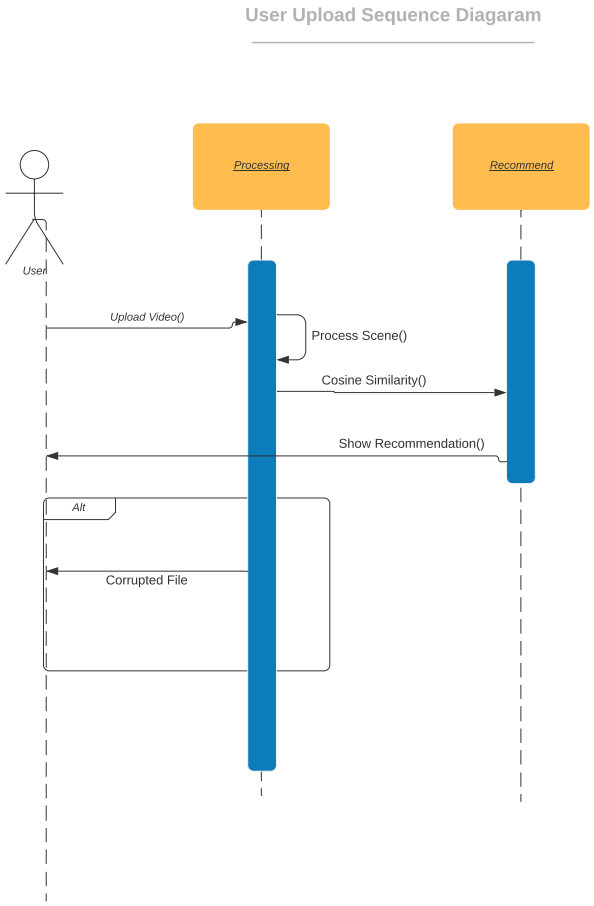

Figure 4.6: User Upload Sequence Diagram

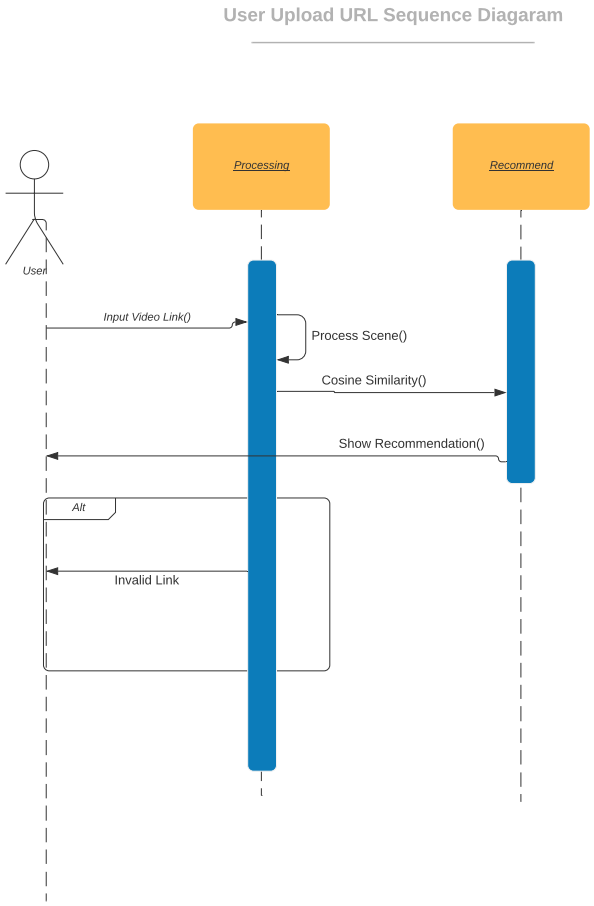

Figure 4.7: User URL Sequence Diagram

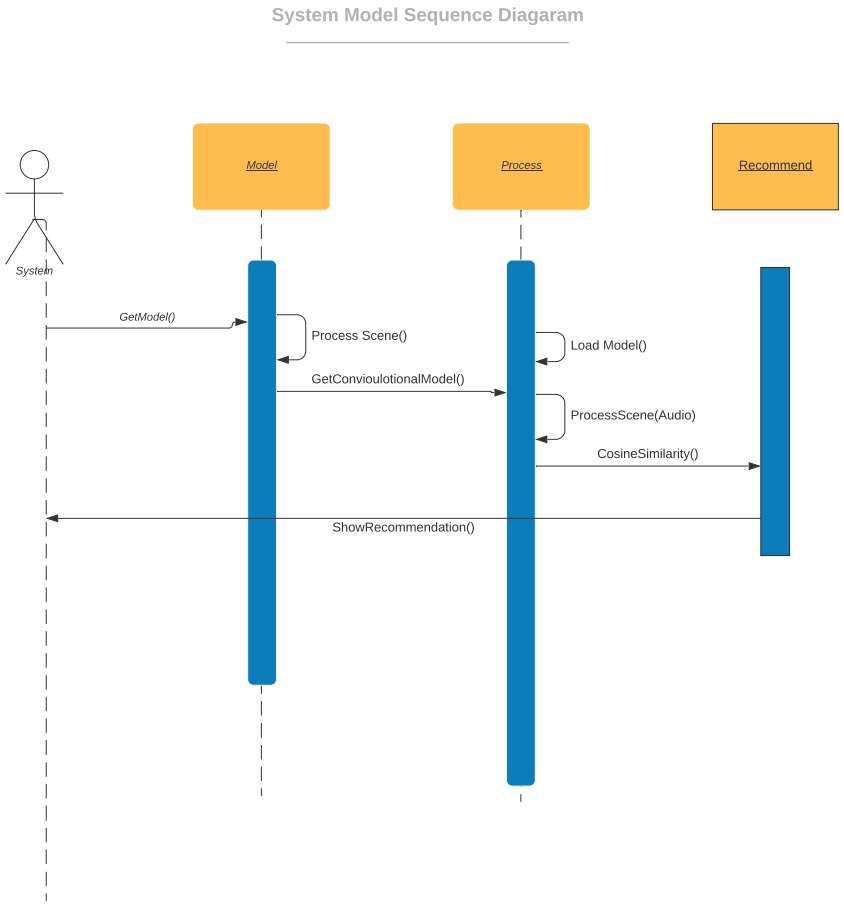

Figure 4.8: System Model Sequence Diagram

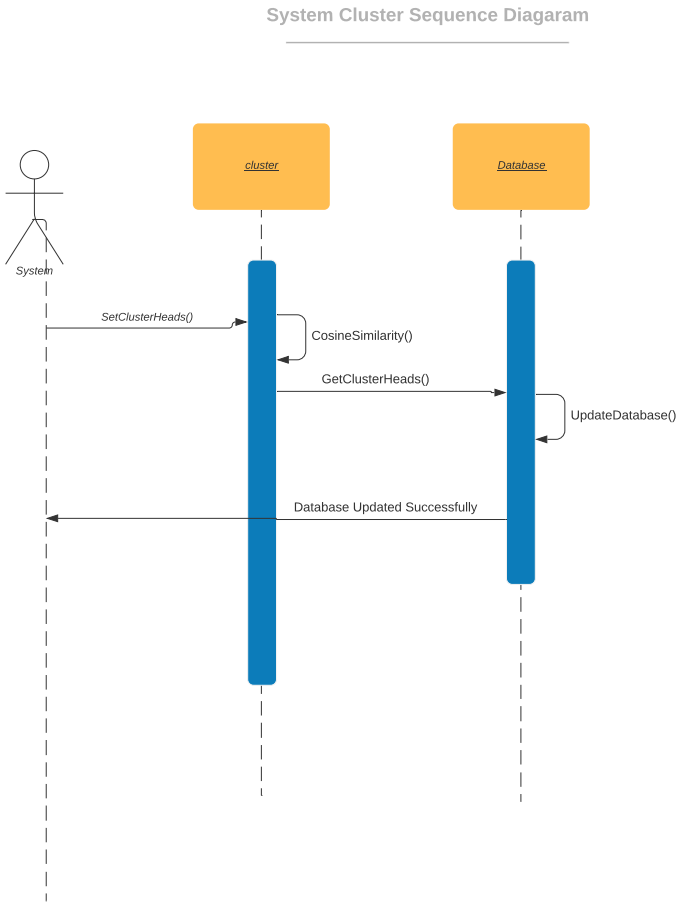

Figure 4.9: System Clustering Sequence Diagram

4.3.3 Design Rationale

This architecture allows our system to run seamlessly as the whole system processes are sequential.MVC architecture was not needed as its main purpose is to send data along Model,view and controller. Its better to use 3 tier architecture if the system is running in a sequential way as in this systems case. For the algorithms choice, A lot of algorithms are available to calculate similarity including Cosine similarity,Euclidean distance,Jaccard's intersection and Manhattan's distance. The Cosine similarity beats all of the above measurements where it measures the angle between the videos, rather than the distance in case of Euclidean distance. Therefore, making the similarity measurement much more accurate in terms of objects included. In addition, it uses the number of common attributes divided

by the total number of possible attributes, rather than Jaccard's intersection divided by the union. Therefore, the best-used similarity technique for the proposed recommendation system is the Cosine similarity.

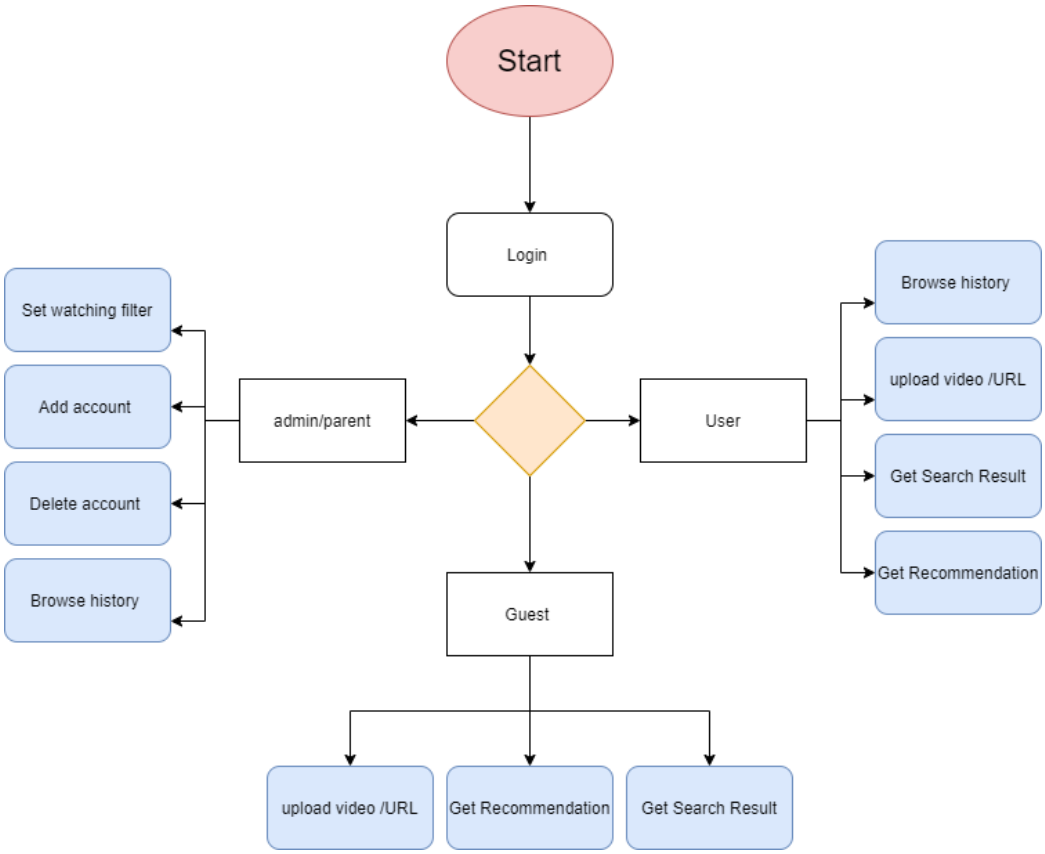

Figure 4.10: Process Diagram

4.4 Data Design

4.4.1 Data Description

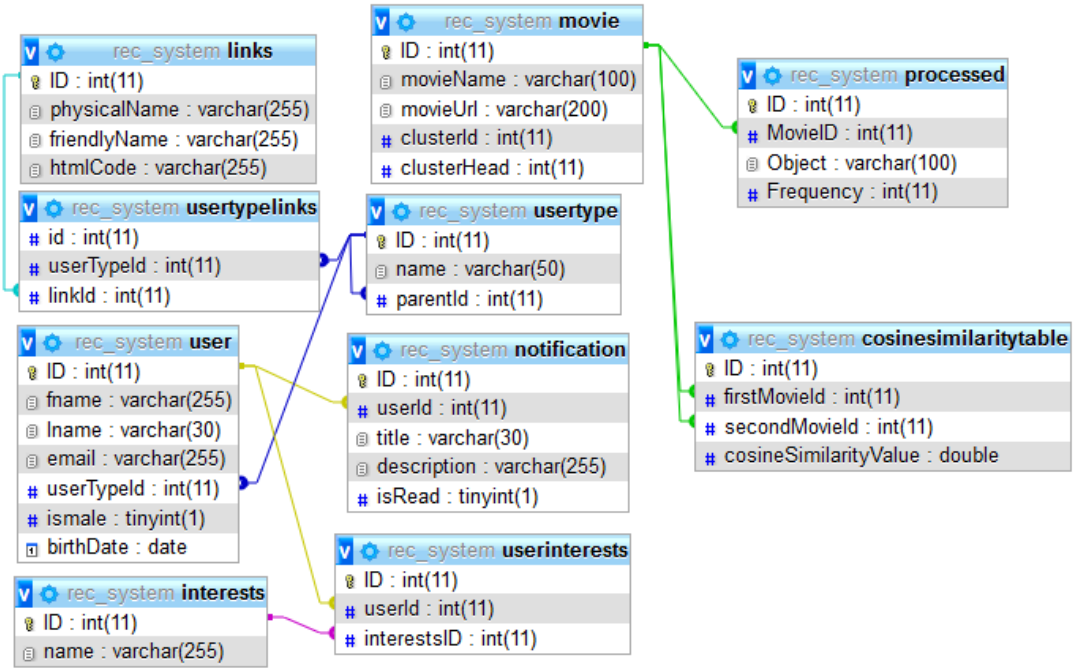

Figure 4.11: Database Schema

4.4.2 Data Dictionary

- user: This entity will hold the account information for every user. It will store information like: username, password and email.
- usertype: This entity will hold the type of the user.
- usertypelinks: This entity will store the allowed pages link of the plugin for each user type
- links: This entity will contain all the page links of the plugin
- userinterest: This entity will store every user interest of the videos in the plugin
- movie: This entity will hold the movie details
- interests: This entity will contain all interests users can be interested in
- cosine similarity: This entity will store the cosine similarity between videos in the database

- processed: This entity will store every object of all the videos have
- notification: This entity will hold the user notification and it's details.

4.5 Component Design

4.5.1 Machine Learning

4.5.1.1 Spectral Clustering

It clusters all videos into number of clusters to simplify the time of comparisons.

$$d_i = \sum_{j=1|(i,j)\epsilon E}^{n} w_{ij} \tag{4.2}$$

4.5.1.2 Fast Fourier Transform

It converts audio signals from its original domain to a representation in the frequency domain.

$$X(k) = \sum_{N=0}^{N-1} x(n).e^{-j(\frac{2\pi}{N})nk} (k = 0, 1, ..., N-1) \tag{4.3}$$

4.5.2 Neural Network

4.5.2.1 ReLU

ReLU stands for Rectified Linear Unit, it is used in Convolutional neural network. We use it to eliminate the negative values in the neural network.

$$f(x) = x^+ = max(0, x) \tag{4.4}$$

4.6 Human Interface Design

4.6.1 Overview of User Interface

The User interface is going to be flexible and easy. First the user will sign in with google, then the user will choose between inserting video URL or upload a video from his personal computer, after this the system will show the user the recommended videos.

4.6.2 Screen Images

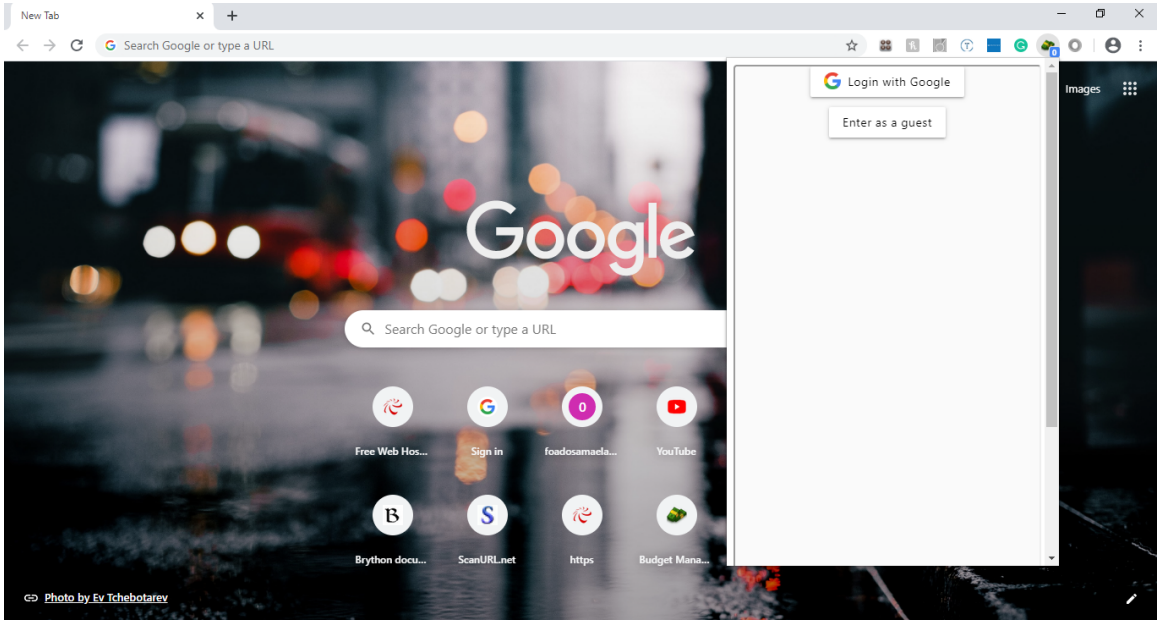

Figure 4.12: SignIn Screen

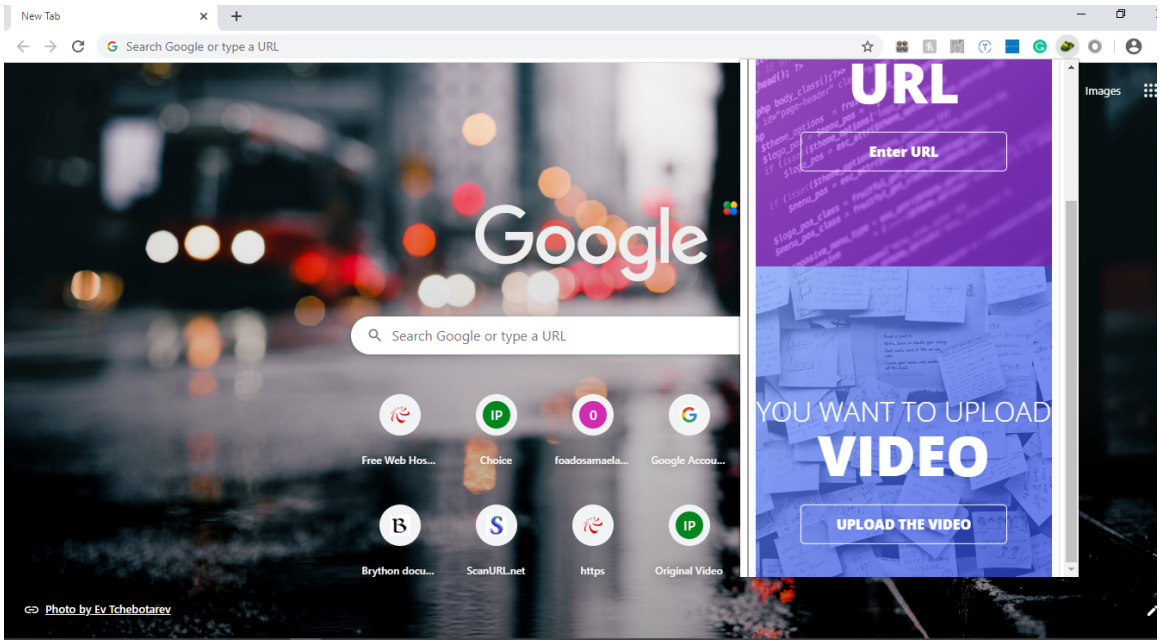

Figure 4.13: Main menu

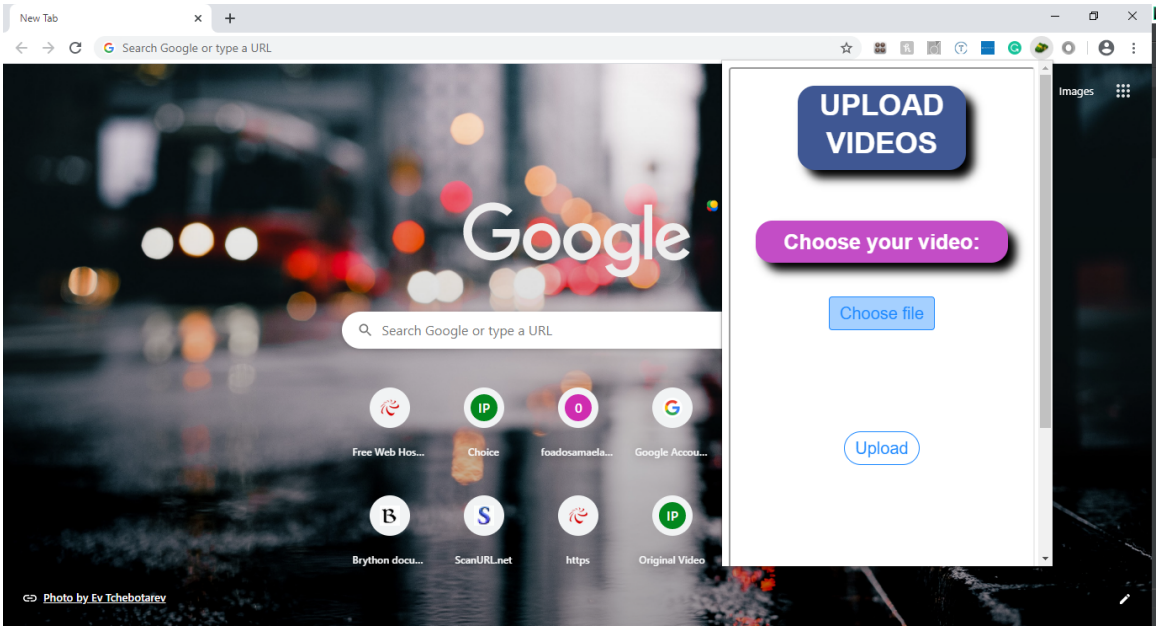

Figure 4.14: Upload/Chooses your Video

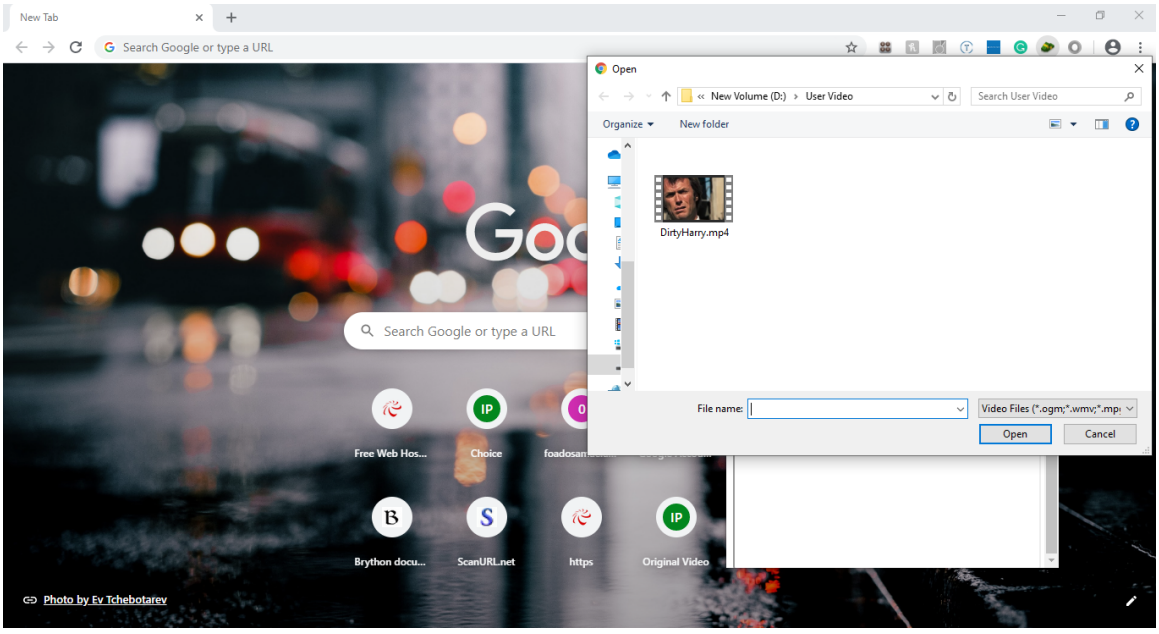

Figure 4.15: Chosen Video

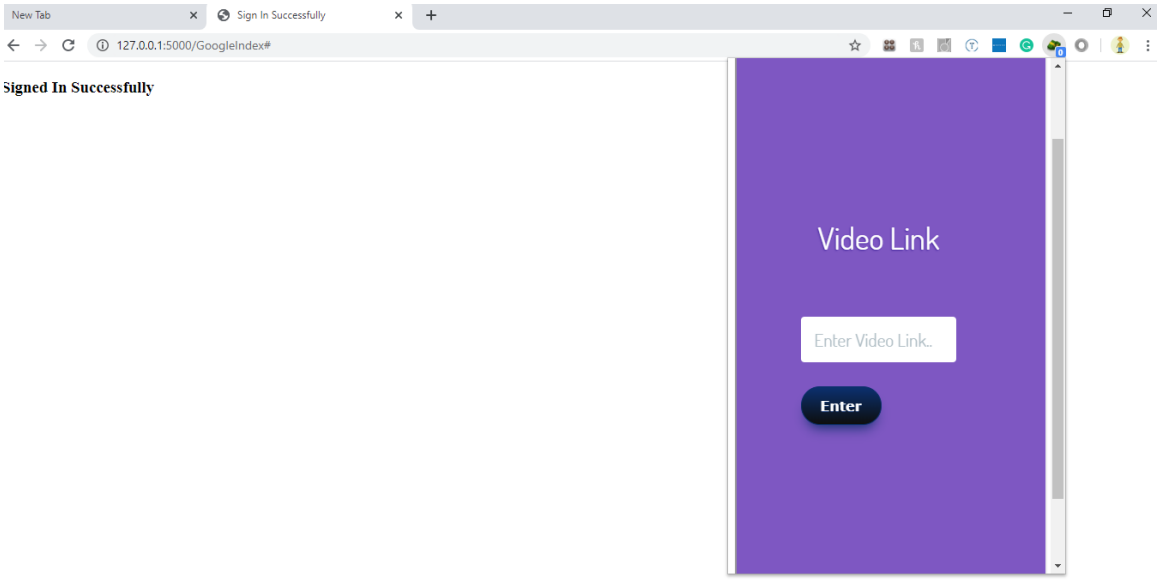

Figure 4.16: Insert Video Link

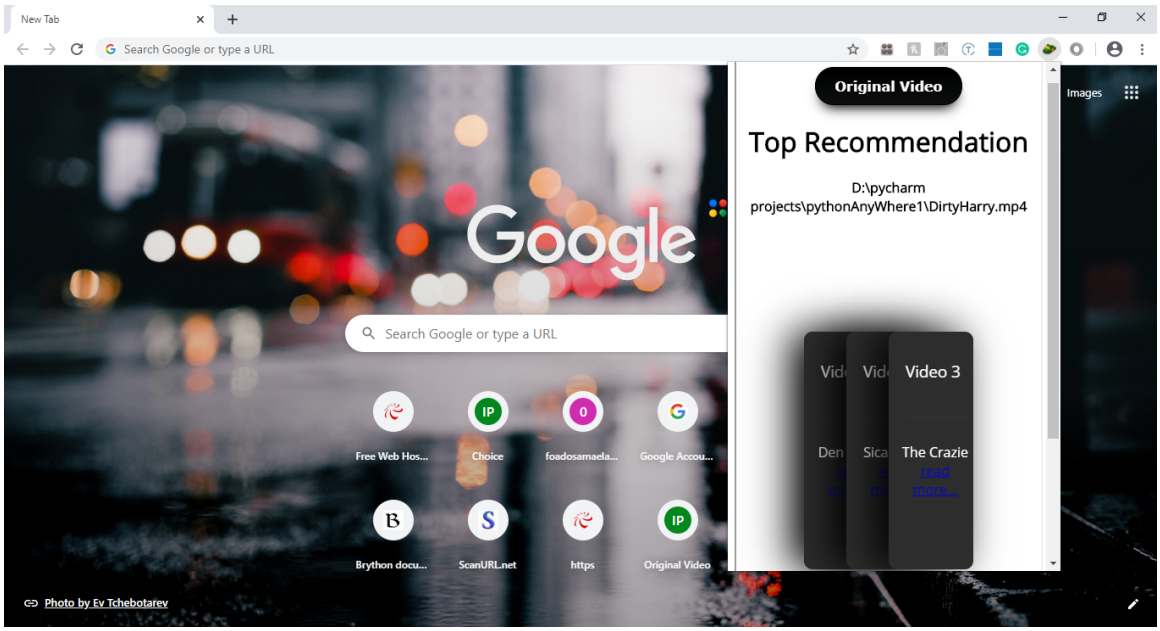

Figure 4.17: Top-N recommendation videos

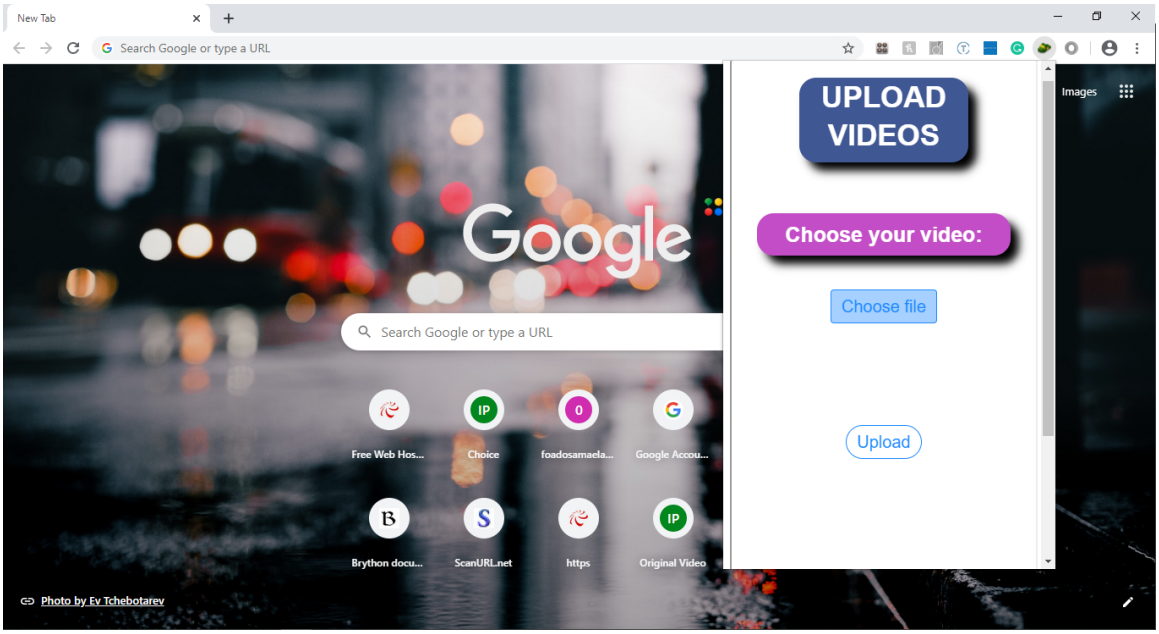

Figure 4.18: Upload/Chooses your Video for Filter Feature

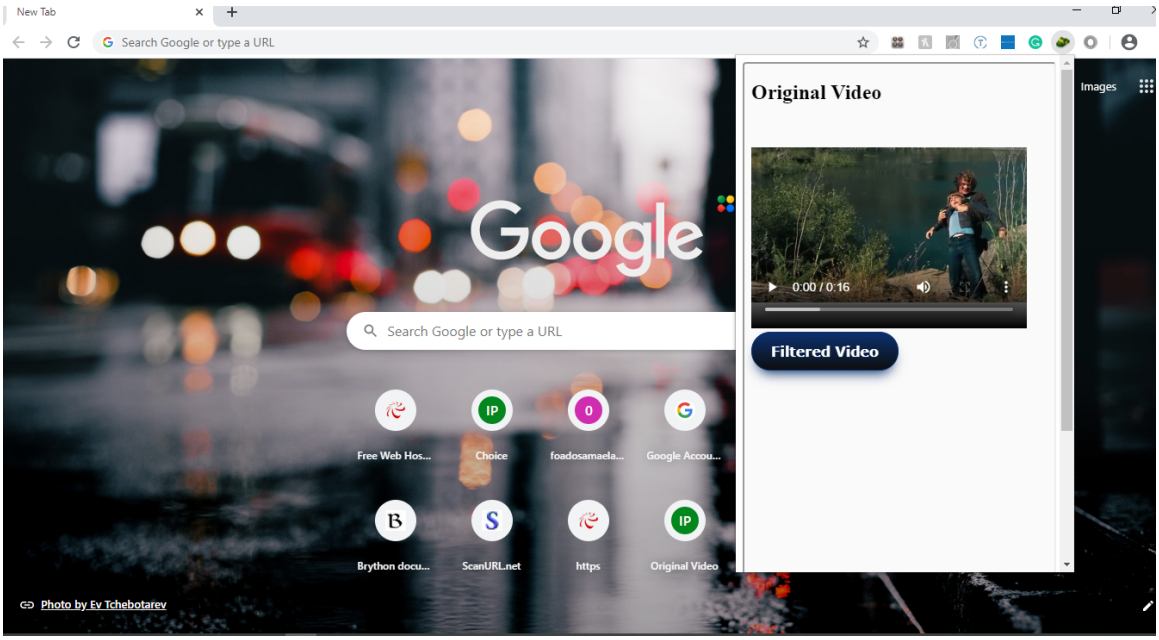

Figure 4.19: Filtration process

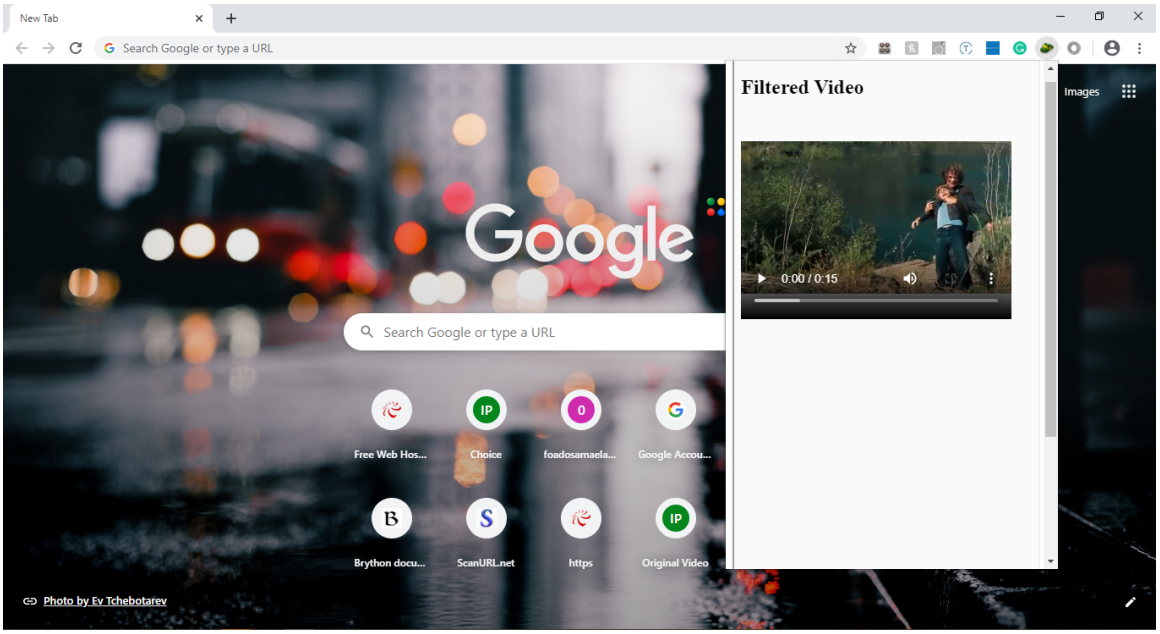

Figure 4.20: Filtered Video

4.6.3 Screen Objects and Actions

- Figure 4.12(Sign In screen): Selecting button sign in with google then entered with his/her email.
- Figure 4.13(Main menu): The user chooses whether to insert a video link or upload a video.
- Figure 4.14(Upload/Chooses your Video): The user chooses to upload a video.
- Figure 4.15(Chosen Video): The user select to upload a certain video.
- Figure 4.16(Insert Video Link): The user chooses to insert a video URL he/she wants.
- Figure 4.17(Top-N recommendation videos): The system show the user the recommended videos for the video the user uploaded.
- Figure 4.18: (Upload/Chooses your Video for Filter Feature): The user choose either to upload his video or to put the URL of the video He/She want.
- Figure 4.19: (Filtration Process): After the user put the video and click on the button (Filtered Video), the filtration process from the unwanted scenes started.
- Figure 4.20: (Filtered Video): The could now watch the video after being filtered.

4.7 Requirements Matrix

Code	Name	Type	Description	Test Strategy
F1	Show Recommendation	Required	It allows the user to see the recommended videos based on the relevance calculated from the similarity	Must give N number of recommendations to the user
F2	Select Scenes	Required	It allows the user to trim the video and select the specific frames he wants to search with	User can select scenes from the video he chose
F3	Search With Video	Required	It allows the user to search with a video that he selected to get similar videos	Results are returned to the user after he searched using his video
F4	Show History	Required	It allows the user to select videos from history and search with it to get similar videos	User was successfully shown the history of videos he received before
F5	Upload Video	Required	It lets the user to upload his own video	Upload was successful and an error wasn't returned
F6	Insert Video Link	Required	Lets the user insert the wanted video URL	Video is successfully uploaded and processed
F7	Create Filter	Required	This function let the user creates a filter for mature content	User was able to create a suitable filter for his needs
F8	Show Accuracy	Required	This function lets the admin show the accuracy of the system such as Clustering, Training sets	Accuracy is returned to the admin without any problems
F9	Sign up	Not Required	It lets the user sign up to start the system functions	Account was created successfully and saved
F10	Login	Not Required	It lets the user login with his username and password to start the system functions	User can successfully login to the system without any problems

Table 4.3: Requirement Matrix Table

Chapter 5

Evaluation

5.1 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 [26]. 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.

5.1.1 Experiment setup

All experiments are conducted by using a YOLO library to process the scenes in each video for object detection. Then, using python script to get the objects and accuracy out of the processed videos. For sound detection, 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.

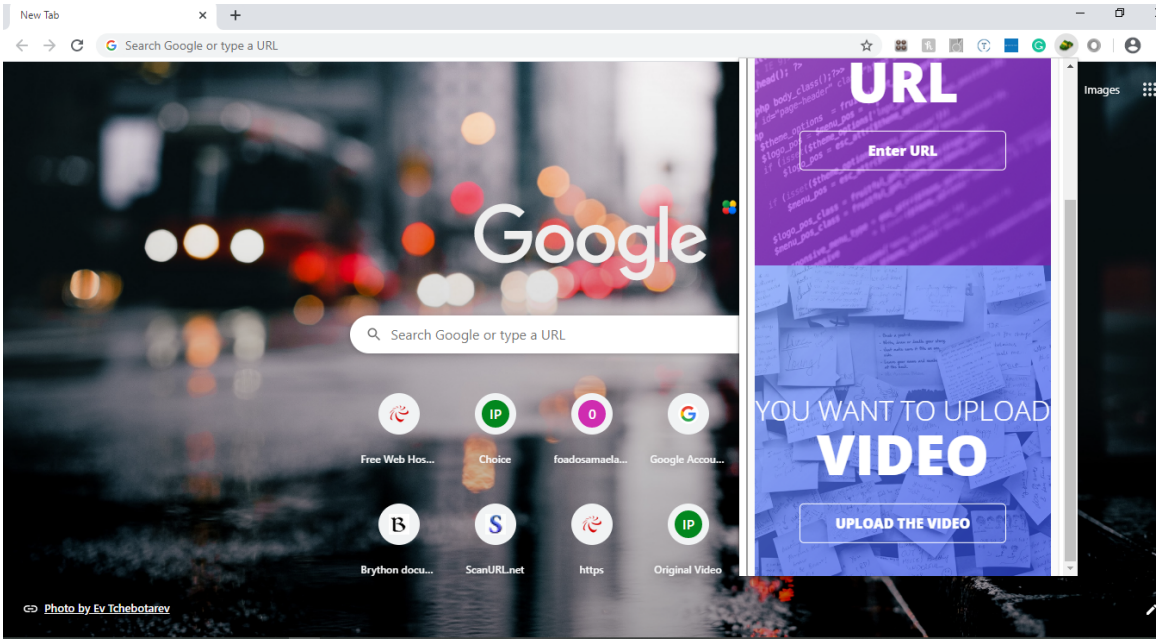

Figure 5.1: Main menu

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

5.1.2 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 5.2 (i.e., the recommended Top-N videos).

5.1.3 Different scenarios during various phases

Recalling to the main three phases of the proposed system, each one can be summarized below.

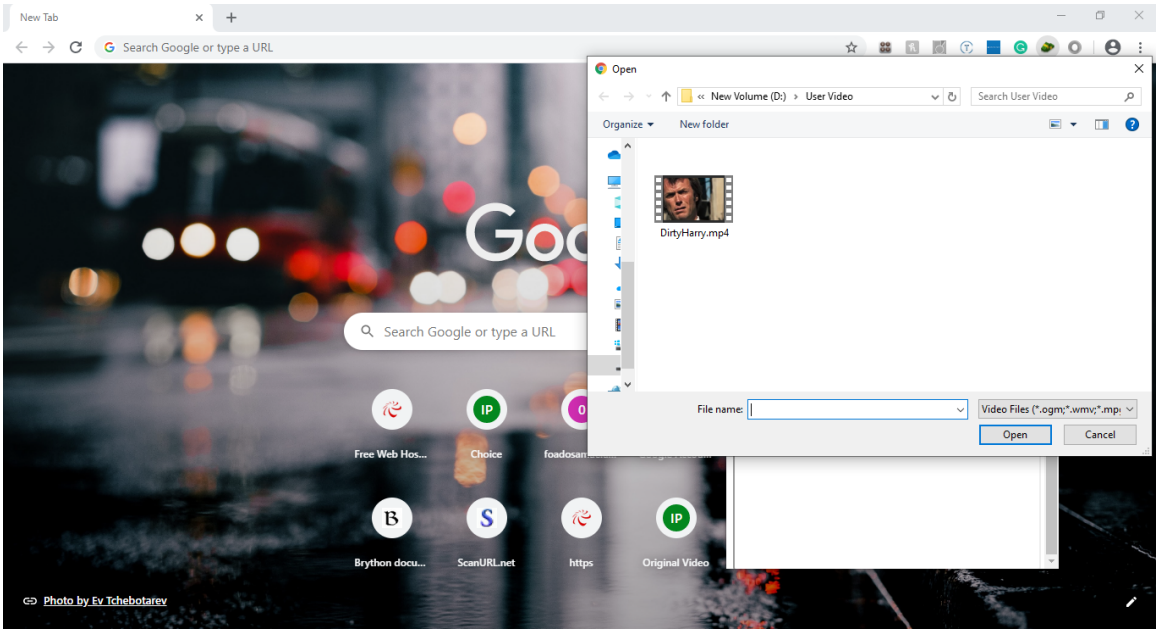

Figure 5.2: Top-N recommendation videos

The first phase is the process of obtaining the video. Videos can be inserted by the user directly in the form of a search query, otherwise, the recommend function will take place which will use the scenes from a video.

The second phase, which is the object detection phase, the video will pass by this phase from the search entered by the user. Object detection will take place, but first, some segmentation takes place using only 5 to 10 percent of the frames to save computations, the output of this phase will be objects detected from the video with a label over them. Alongside the object detection algorithm, the sound detection algorithm takes place extracting the audio and giving them values, The labels and sound values will be used later to construct the Video ID.

The video ID for the inserted video is generated, comparisons will take place as *the third phase* by using a cosine similarity measurement using objects and sound data as attributes. This will result in an output of similar recommended videos, search result if the user selected the search function. Also, if the user enabled filtering, the output phase can display the video after being filtered from certain objects.

Different scenarios as discussed below as illustrative examples to demonstrate the working process of the proposed system.

5.1.3.1 Scenario 1

According to the first phase, the user's search query was a video interval containing a person and a car, the video will be processed as explained in the diagram. After the object detection phase, the table of data will start recording properties of the car and of the person. After that, similarity will take place, in this case, results should contain relative content to the car and the person.

5.1.3.2 Scenario 2

or phase two, let's create a scenario when a video contains 3 cars and 2 trucks, the data output from this phase should be in the object-frequency table. For the object car, the frequency of an object car should be higher than object truck. 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

5.1.3.3 Scenario 3

At phase three, as an example, a video containing a person, gun and a knife, the objects detected should be the contained objects, respectively. In the table of data, the results will contain these objects, and output the videos with the highest similarity form the comparison of the data. On the other side, if a video containing a ball, a clock and a person. The object ball was from the list of objects to be blocked, the scenes containing a ball will be blocked while the scenes containing the clock and person will be viewed normally.

5.1.3.4 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.

Here is an example from an early demo for the proposed system is shown in Figure 9. In this case, we have a frame from a video containing some vehicles. From the Figure, the cars are successfully labelled.

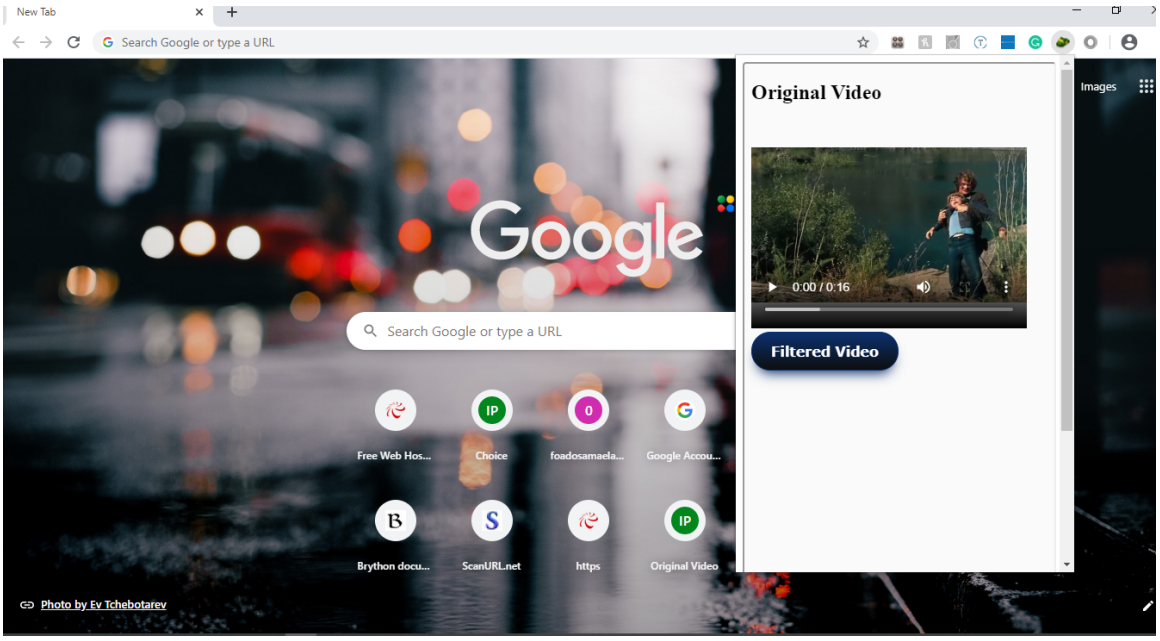

Figure 5.3: Filtration process

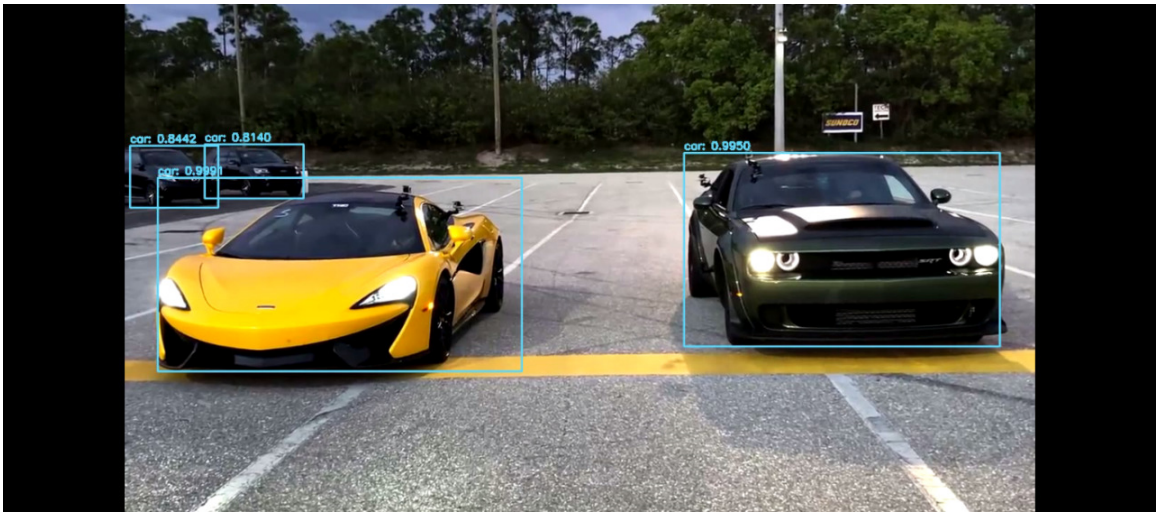

Figure 5.4: Object detection and labeling

As an example of the object-frequency table is shown below. It currently showing the frequency of objects from a randomly selected video

5.1.4 Performance measure and analysis

In order to validate the accuracy of the proposed recommendation system and against YouTube recommendation, equation (5.1) can be used to determine how the relevancy results are achieved.

$$Relevency(\%) = \frac{\sum_{i=1}^{n}((O(i) * f(i)) + SV(i))}{\sum_{j=1}^{x}\sum_{k=1}^{m}((O_j(k) * f_j(k)) + SV_j(k))} \tag{5.1}$$

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. $O_j(k)$ represents the object k of the recommended video that the algorithm matched with recommended video j along with its frequency $valuef_j(k)$, and $SV_j(k)$ is the matching sound value of object k for video j.

Figure 5.5 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.

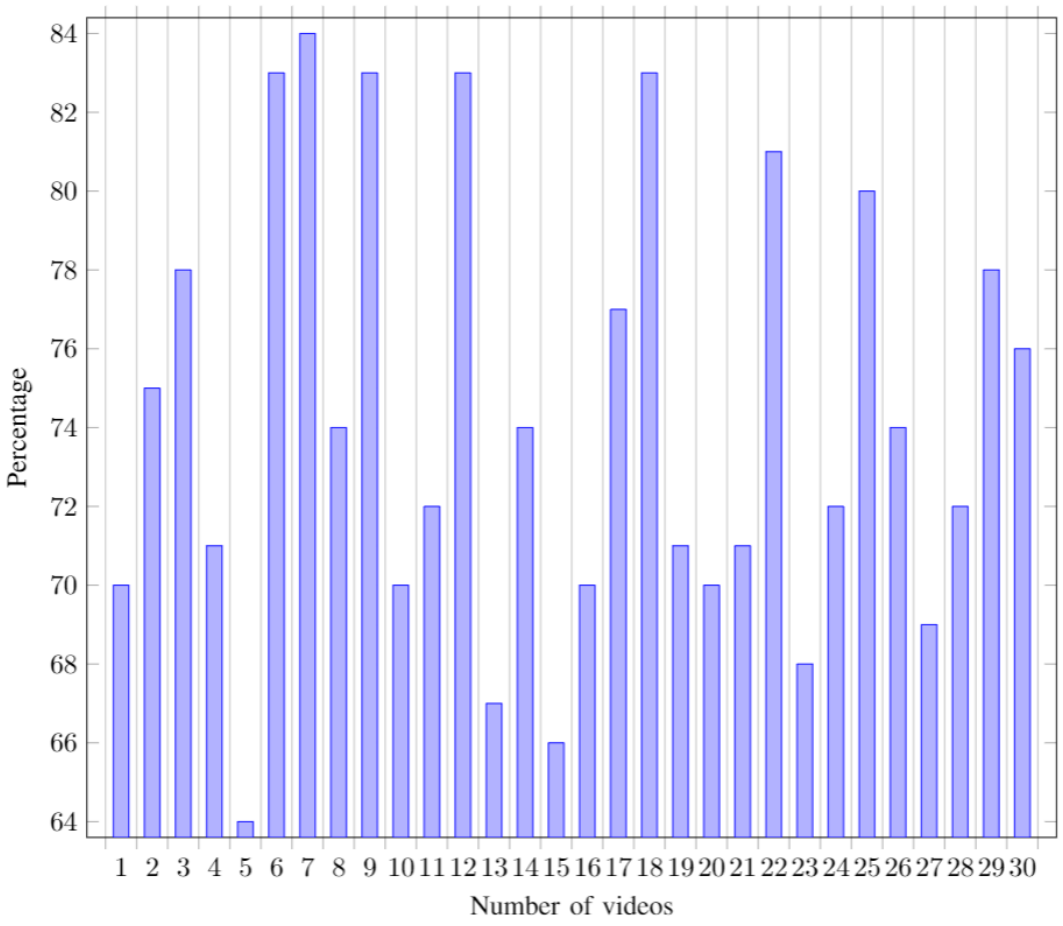

Figure 5.5: The relevancy of the proposed recommended system

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 to ensure fair recommendations on both platforms and to compare how relevant are the videos we recommend to users with the videos that the YouTube system recommends.

Each video selected 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, creating a sheet of data representing their objects, respectively. These data can be used in equation (5.1) 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 seen that, the proposed system has achieved an average accuracy 69.4 % while the YouTube overall recommendation system has achieved relevancy of content 62 %.

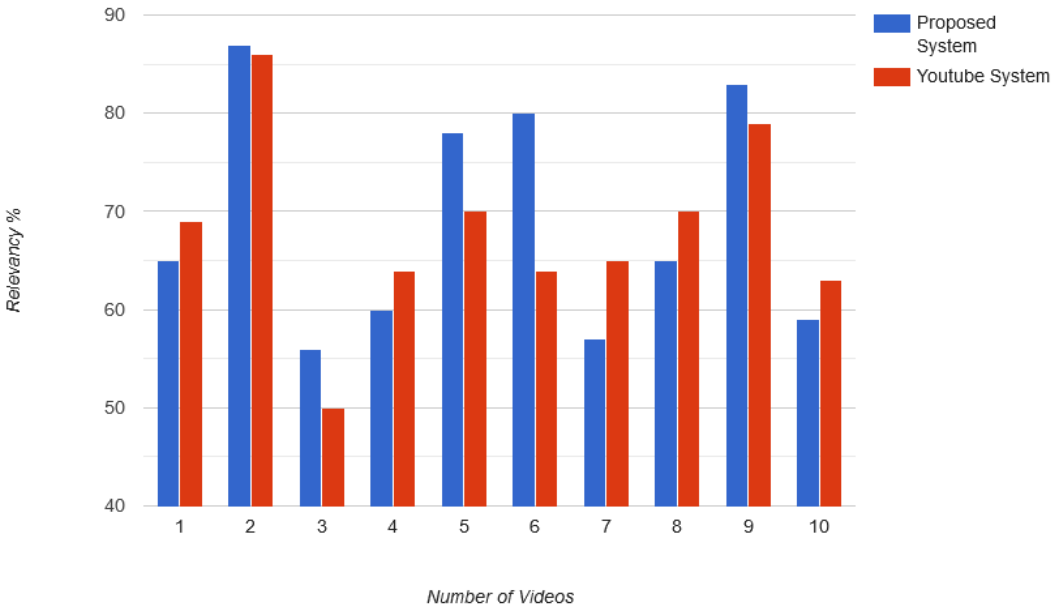

Figure 5.6: Proposed recommendation system vs YouTube recommendation system

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

Chapter 6

Conclusion and Future Direction

6.1 Conclusion and future directions

This paper proposes an efficient content-based video recommendation system. The proposed system is built upon extracting the visual features from video content rather than the semantic 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 consists of three main phases starting by capturing 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 solving the problem of cold-start and the second is the recommendation from a scene. Some future directions are proposed including detection 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 the accuracy of it.

Bibliography

[1] Xiaoyuan Su and Taghi M. Khoshgoftaar, "A Survey of Collaborative Filtering Techniques," Advances in Artificial Intelligence, vol. 2009, Article 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. doi:10.1145/2556270.

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

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

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

[6] 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).

[7] 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).

[8] 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

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

[10] Zongxian Li1,2, Sheng Li1, Lantian Xue1,2, Yonghong Tian1,2† 1 National Engineering Laboratory for Video Technology, School of EECS, Peking University, Beijing, China 2 Pengcheng Laboratory, Shenzhen, China

[11] 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).

[12] 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.

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

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

[15] 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

[16] 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).

[17] 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.

[18] 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. $0_37 - 0_38$.

[19] Samireddy, S. R., Carletta, J., Lee, K.-S. (2017). An embeddable algorithm for gunshot detection. 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS).

[20] 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

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

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

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

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

[25] 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)

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