LEVERAGING THE MULTIMODAL INFORMATION FROM VIDEO CONTENT FOR VIDEO RECOMMENDATION

by

Adolfo Ricardo Lopes De Almeida

Submitted in partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

March 2021

SUMMARY

LEVERAGING THE MULTIMODAL INFORMATION FROM VIDEO CONTENT FOR VIDEO RECOMMENDATION

by

Adolfo Ricardo Lopes De Almeida

Supervisor(s):	Prof. J.P. de Villiers
Co-supervisor:	Dr. A. De Freitas
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Computer Engineering)
Keywords:	Audio features, action features, Bhattacharyya distance, deep learning features, item cold-start, item warm-start, multimodal feature fusion, matrix scaling, object features, scene features, statistical feature aggregation, video recommendation

Since the popularisation of media streaming, a number of video streaming services are continually buying new video content to mine the potential profit. As such, newly added content has to be handled appropriately to be recommended to suitable users. In this dissertation, the new item cold-start problem is addressed by exploring the potential of various deep learning features to provide video recommendations. The deep learning features investigated include features that capture the visual-appearance, as well as audio and motion information from video content. Different fusion methods are also explored to evaluate how well these feature modalities can be combined to fully exploit the complementary information captured by them. Experiments on a real-world video dataset for movie recommendations show that deep learning features outperform hand-crafted features. In particular, it is found that recommendations generated with deep learning audio features and action-centric deep learning features are superior to Mel-frequency cepstral coefficients (MFCC) and state-of-the-art improved dense trajectory (iDT) features. It was also found that the combination of various deep learning features with textual metadata and hand-crafted features provide significant improvement in recommendations, as compared to combining only deep learning and hand-crafted features.

LIST OF ABBREVIATIONS

CB	Content based
CER	Collaborative embedding regression
CF	Collaborative filtering
CNN	Convolutional neural network
DNN	Deep neural network
HMDB	Human motion database
ICM	Item content matrix
iDT	Improved dense trajectories
I3D	Inflated three-dimensional CNN
ItemKNN-CBF	Item-based k-nearest neighbors content-based filtering
MFCC	Mel-frequency cepstral coefficient
MAP	Mean average precision
MoSIFT	Motion scale-invariant feature transform
NDCG	Normalised discounted cumulative gain
NN	Neural network
REC	Recall
ResNet	Residual neural network
SSR	Signed square root
UCM	User content matrix
UMAP	Uniform manifold approximation and projection
URM	User rating matrix
VGG	Visual geometry group
WMF	Weighted matrix factorisation

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1 PROBLEM STATEMENT

1.1.1 Context of the problem

Following the recent increase in the popularity of video streaming services, a large amount of video data are continually uploaded to video sharing sites [1]. These sites depend heavily on video recommendation systems to assist the users in discovering videos they would enjoy. A video recommendation system is a user-level video filtering service that helps users explore the world of videos [2]. It offers a more personalised experience to users by recommending the most relevant and appropriate videos for them. This is performed by utilising algorithms to analyse the information about the videos and users, as well as information on past interactions between the user and videos [3, 4].

Existing recommendation systems mainly use one of three approaches, namely the collaborative filtering (CF) recommendation method, the content based (CB) recommendation method, and the hybrid recommendation method. The latter is a combination of the CF and CB recommendation approaches [2]. The CF recommendation method uses a user's explicit or implicit feedback, such as previous ratings and their watch history in order to predict the preference of the user. This is achieved by recommending a video to a user if like-minded users have watched it or given it a positive rating [2]. The CB recommendation method uses the target user's profile and video content to predict their preferences. Consequently, a video is recommended to a user if the content is similar to what the user liked or watched before [5]. On the other hand, the hybrid recommendation methods combine both the user's feedback and the consumed video content in order to improve recommendations.

Most video streaming services that use a video recommendation system to compute video relevance, based on user implicit feedback, use item-based CF methods because of their state-of-the-art accuracy [6–9]. The implicit user feedback is used to model the user-video preference and to provide personalised recommendations by computing video-to-video relevance scores. The main drawback of this strategy,

is the cold-start problem, which occurs when the recommender system does not generate accurate recommendations for users who have no historical interaction record (new user cold-start problem) or when the system is not capable of recommending items in the catalogue to a user because these items lack any interaction (new item cold-start problem) [7, 10]. Given the current much higher rate for newly uploaded videos than newly subscribed users in video streaming services [11], in this dissertation, the new item cold-start problem is studied.

As a result of the tremendous increase in new video uploads, video streaming services have to deal with unrated, unaudited, and completely new content, which they know nothing about [12]. This problem is more severe for video streaming services that usually purchase new movies and TV series from content providers to mine the potential profit [6, 13]. As such, the new item cold-start problem has to be handled effectively to ensure that the purchased content is discovered by most of their users. This dissertation addresses the new item cold-start problem by enhancing the recommendation task for video streaming services, by combining various video content features in order to effectively recommend newly added videos to users.

1.1.2 Research gap

Recent work on personalised video recommendation for streaming services has shown that recommendation based on deep learning object features and hand-crafted audio features, combined with collaborative filtering information have a higher recommendation quality compared to recommendations based only on deep learning object features or metadata, such as genre or cast [7, 14]. This is however still not the optimal solution, as deep learning action features that capture the motion information in videos, and their complementariness to deep learning visual-appearance and audio features, are not explored. This is important addition information that forms part of the rich and varied information present in videos. Videos are characterised by actions and scenes that help their narrative and pass on their message to the audience, which may have a significant influence on users' preferences [15–19]. For example, temporal sequencing of cars in a video where the cars in the scene might appear stationary, yet the background is continually moving, could be an indicator of a car chase; an irregular and complex kind of motion could be an indication of hand-held shot videos, which some people do not like [20]. In addition, in the field of video recommendation, the utilisation of several deep learning features that capture different aspects of the video content is still a rare, explored area compared to hand-crafted features [21]. It is evident that there is a need to solve the new item cold-start problem by implementing a hybrid video recommendation system, which uses the users' past interactions with the

videos and considers the complementary information from different deep learning features, extracted from the media contained in the videos. These features should capture the visual-appearance, as well as audio and motion information in order to best exploit their presence and provide more accurate personalised video recommendation to users in the new item cold-start scenario.

1.2 RESEARCH OBJECTIVE AND QUESTIONS

The objective of this research is to leverage deep learning action features extracted from videos and its complementariness among deep learning visual-appearance and audio features to provide more accurate personalised top-N video recommendations to users in the new item cold-start scenario given the users' implicit feedback. Top-N video recommendation is a method where N videos that a user potentially likes but never watched are recommended. The video recommendation system should use the users' past interactions with the videos and various deep learning features extracted from the media contained in the videos that capture their visual-appearance, audio and motion information. These features should capture enough information from the videos necessary for video recommendation, which should therefore play a noteworthy role in solving the new item cold-start problem. Accordingly, a comprehensive investigation of how to effectively combine these features to further improve the recommendation quality in terms of accuracy and beyond-accuracy metrics is conducted . Given the identified gap in existing literature, this research work is intended to address the following research questions:

1. Can the combination of the visual-appearance, audio, and action-related features, which capture the visual, aural and motion information contained in the videos, provide better video recommendation, with respect to accuracy and beyond accuracy metrics, than the visual-appearance and audio features, which only capture visual and aural information?
2. What motion information from videos is the most predictive of users' video preferences in new item cold-start scenarios?
3. To what extent can the combination of hand-crafted features, deep learning features, and textual features maximise the video recommendation performance?

1.3 HYPOTHESIS AND APPROACH

This research makes the following hypothesis: By finding, using, and augmenting existing video recommendation models, with a combination of different video content features, a video recommendation system, will improve its recommendation with respect to accuracy and beyond accuracy metrics. These features should capture visual-appearance, audio and motion information contained in the

videos. A combination of users' implicit preferences with these features further improves the quality of recommendation. In order to achieve this, the following approach was followed:

1. Investigate which audio, appearance and action-related features will best capture the multi-modal information from the video content.
2. Investigate and implement video recommendation models that are most likely to work well with these features.
3. Perform feature analyses to determine if these features are semantically meaningful before using them as complementary information for recommendation models.
4. Perform a series of planned tests to find which of these features are most suitable for the task of video recommendation.
5. Investigate and implement fusion methods in order to fully exploit the complementary information from these features and enrich the recommendations.

1.4 RESEARCH GOALS

This research aims to show that the motion information captured by deep learning action features and its complementariness among deep learning visual-appearance and audio features, in a video recommendation system, provide more accurate personalised video recommendation to users. The aim is to explore and implement different existing video recommendation models that use visual-appearance, audio, and action-related features on their content description. These features, extracted from the multi-modal and high dimensional information from videos, are combined into a single compact fixed-length vector that represents information from all aspects of the video. This compact video representation is used for the video recommendation in the item warm-start and cold-start scenarios. This research also aims to understand whether motion information captured by deep learning action features extracted with three-dimensional (3D) convolutional neural networks (CNNs) [15, 22], lead to improved recommendations compared to hand-craft action-related features.

1.5 RESEARCH CONTRIBUTION

The implemented system best exploits the availability of different features that capture the visual, aural, and motion information contained in videos in order to enhance the recommendation task and improve the user experience. A performance comparison between different features and algorithms is performed in this work in terms of accuracy and beyond-accuracy metrics in the item warm-start and cold-start scenarios. This adds to the knowledge of using recommendation methods for videos, as well as the selection and efficiency of techniques that fully exploit the complementary information

from the various feature modalities. This assists in understanding the role of action-related features among visual and audio feature modalities in the video recommendation process. This research could also enlighten various applications associated with the proposed system, such as recommendation systems for movie theatrical releases, music to video retrieval, click-through rate prediction for videos, personalised advertisement, and film distribution support.

1.6 RESEARCH OUTPUTS

1.6.1 Conference proceedings

The following conference paper was presented at the 23rd International Conference on Information Fusion (FUSION) in 2020 and published in the peer reviewed proceedings of the conference.

1. A. Almeida, J.P. de Villiers, A. De Freitas and M. Velayudan, "Visual comparison of statistical feature aggregation methods for video-based similarity applications," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020, pp. 1-8.

1.6.2 Journal publications

The following article was submitted to a peer-reviewed journal for publication:

1. A. Almeida, J.P. de Villiers, A. De Freitas and M. Velayudan, "Multimodal deep learning feature based information fusion for video recommendation," *Information Fusion*, submitted for publication.

1.7 OVERVIEW OF STUDY

This dissertation is separated into 5 chapters, each dealing with different aspects of the work conducted and is organised as follows:

- Chapter 2 presents a comprehensive literature survey of low-level and deep learning features in the video domain, feature aggregation, fusion methods, and existing video recommendation approaches that exploit video content. It underlines the theoretical background of different recommender systems approaches. In addition, this chapter includes the advantages and disadvantages of each method reviewed.
- Chapter 3 introduces the reader to the recommendation framework implemented in this work, and describes the experimental setup.
- Chapter 4 presents the results obtained by using the experimental setup and recommendation framework described in Chapter 3.

- Chapter 5 presents a detailed discussion of the results obtained.
- Chapter 6 draws conclusions about the research conducted and highlights recommendations for future work.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OVERVIEW

The goal of this chapter is to provide an overview of the state-of-the-art research in video recommendation approaches that exploit video content. This chapter begins by explaining and comparing the various types of features used in the video domain to accomplish this objective. It discusses the commonly used feature aggregation and fusion methods in video content analyses. In addition, the fundamental concepts of recommender systems and their evaluations are discussed. Finally, existing video recommendation approaches found in open literature, which exploit video content are reviewed in detail.

2.2 FEATURE EXTRACTION IN VIDEO DOMAIN

Video recommendation systems typically exploit high-level features extracted from a video's post-release textual metadata such as plot, reviews, cast, genre, tags and director to generate recommendations [5]. The metadata is generated by humans, which lead to features close to the humans' interpretation and perception of the video, which is prone to errors, rare or unavailable for new videos, biased, and might not fully represent the video [23–25]. For example, two videos can have similar metadata, but their visuals and style can be considerably different, which can affect the users' opinions and feelings differently [26]. Consequently, relying on human-generated textual metadata may lead to inaccurate recommendations and drastic degrading of system performance when this data is scarce [24, 27]. In this regard, audio and visual features, extracted from the media contained in videos, are likely to enhance the quality of recommendations as they generate valuable representations of the videos. This is shown in video content analyses tasks such as video classification, indexing, and retrieval [28–30].

In order to generate a video representation, features are extracted from the text, audio and visual

modalities of the video content. While applying concepts from film theory, features useful for video classification are usually extracted [20,31]. For example, the light level is a useful feature to distinguish between horror and comedy films as they typically have low-light and high-light levels, respectively. Whereas, the levels of motion are useful features to identify drama (low levels of motion), or sports and action movies (high levels of motion). These features can be categorised into three classes, namely text-based features, audio-based features, and visual-based features [31]. Regardless of which of these are used, new metadata is generated by analysing video scenes to classify segments of video, such as identifying horror scenes, violent scenes, car chase scenes, or specific types of sports in a video. In this regard, this section reviews the hand-crafted low-level features and deep learning features used in the video domain.

2.2.1 Hand-crafted low-level features in video domain

2.2.1.1 Text-based features

Text-based feature extraction is performed by extracting dialogue from speech, using speech recognition, or viewable text on screen [32–34]. It can also be provided in the form of closed captions, which display the dialogue as well as the information about sound effects occurring in the video. These are viewable text on the screen, which can be text on items that are recorded, for example, a street name on a name plate. These are extracted using text detection methods and optical character recognition (OCR) to convert the image of the text into machine-encoded text [35]. While working with text, a commonly used method is to generate feature vectors, which represent the text, using a bag-of-words model [36]. In a bag-of-words model, the number of occurrences of any word is used; however, the information about the order of these words is not kept.

The main advantage of text-based features in video classification, indexing, and retrieval tasks is the simple comprehension between a specific genre and the features. For example, a transcript generated from a news story weather report will have a lot of occurrence of the words *cloud*, *weather*, and *temperature*. The disadvantages are that in general, most of the text extracted from videos is a conversation between two or more people, which does not represent the plot well. Furthermore, not all the videos have closed captions or dialogues available. Lastly, generating feature vectors to represent text is computationally expensive since text can have a vast number of terms.

2.2.1.2 Audio-based features

Audio-based feature extraction is performed by first extracting the audio signals from the video. The features are extracted from the time or frequency domain representation of the audio signal. The audio

signal is sampled at a certain frequency, and these samples are then combined into frames [37].

Some commonly used time domain features are the root mean square (RMS), the zero crossing rate (ZCR), and the silence ratio [31, 38, 39]. RMS is a low level feature that represents the sound loudness sensed by humans [39, 40]. For example, sports videos have a nearly constant level of noise [41]. ZCR is a low level feature that represents the current frame by counting the number of times the signal amplitude sign changed in it. Higher frequencies result in higher ZCR, for example music has lower variability of ZCR compared to speech [42]. The silence ratio is a low level feature that represents the amplitude values below some threshold within a frame [43]. Typically, the silence ratio is lower in music than speech [44].

As mentioned, apart from the time domain features, frequency domain features are often extracted as well. The commonly used frequency domain features, include energy distribution, bandwidth and Mel-frequency cepstral coefficients (MFCC) [31, 44, 45]. The energy distribution is a low level feature that represents the signal distribution across frequency components [46]. This feature is used to provide the location in the frequency band where the frequency components are strong [39]. The frequency centroid is typically lower in speech than in music [47]. Bandwidth is another low level feature, which represents the frequency range of a signal [46, 48]. Normally bandwidth is higher in music than speech [48]. Lastly, MFCCs are low level features that are obtained using bins based on the Mel frequency scale, which contain the logarithm of the spectral components. The discrete cosine transform (DCT) is applied, and the coefficient values, in which most of the energy is concentrated are kept [49]. These coefficients allow the original values to be approximated. The main advantage of audio features is that they are not computationally expensive to extract, as they require less memory compared to textual or visual features [31].

2.2.1.3 Visual-based features

Visual-based features are the most frequently used features in the video classification literature [28], as humans perceive most of the information in the video based on their sense of vision [29]. These features are commonly extracted from the scenes of a video or the frames of a video [29]. The scenes of a video are the best way to segment a video. However, automatically identifying scenes and their boundaries is very difficult [50]. Currently, the methods used in video content analysis tasks rely on the extraction of frame-level and video-level features from consecutive frames where a video-level descriptor is created by aggregating the features over time [22, 51]. As shown in Figure 2.1 a video

is a collection of images that are called frames. A collection of frames is called a shot. A shot or a collection of shots is called a scene.

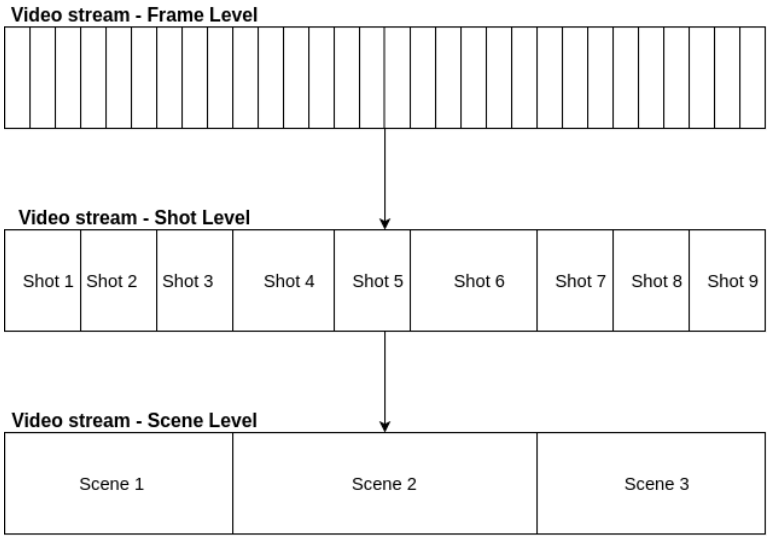


Figure 2.1. The hierarchical division of a video where the smallest unit is a frame. These consecutive frames then form a shot. Several shots combine to create a scene, and a video is then formed from one or several scenes. (Adapted from [52])

Most of the visual features are extracted from videos according to the cinematic principles [20, 26]. These features are mostly colour based to provide information about light levels [20, 26, 53], object-based to provide information about the specific types of objects [54, 55], motion based to provide information about the action, and pace based to provide information about the length of shots in the video [20, 56, 57].

A. **Colour-based features** are simple to obtain. They are extracted from a video frame by dividing the video frame into regions. For each region the number of pixels are counted to obtain a distribution. This distribution is represented using a colour histogram to capture spatial information, and is used to compare two frames [58]. In order to make this feature robust to different light conditions, before dividing the video frames into regions, the colour channel of each frame is first normalised then converted to hue, saturation, value (HSV) colour space. This feature is useful when using cinematic principles. For example, the mood of a user can be affected by the amount of light and distribution of colour [59].

B. **Object-based features** can be computationally expensive and limited by the number of objects in a video that are used to classify a video or retrieve videos that contain similar objects [60].

These are features such as colour histograms, texton histograms, and mass variance and skewness from objects [29, 61]. In order to extract these features, the objects in the videos have to be identified, which may be difficult and time-consuming [29]. The object features commonly used include scale-invariant feature transform (SIFT) [62], speeded-up robust feature (SURF) [63], locally normalised histograms of oriented gradient (HOG) [64], deformable part based features [65]. SIFT are features used for object detection and recognition. These features are partially invariant to occlusion and illumination and completely invariant to basic geometric transformation. It uses the general configuration of the image gradient. SURF are features computed using integral images and an approximation of the Hessian matrix to detect the interest points. The sum of the Haar wavelet transform around these points is calculated for object detection and recognition. Locally normalised HOG are features extracted from localised portions of an image by counting the number of occurrences of gradient orientation for object detection. These features are robust to changes in light and colour and small changes in directions and contour locations. Deformable part based features are features extracted by using a sliding-window to detect objects and small picture segments to represent visual properties of the object which are then arranged in a deformable configuration. This deformable configuration is used to calculate the deformation cost for each pair of connected parts. An energy function is computed by calculating the match cost for each part and deformation cost.

C. **Motion-based features** are usually extracted using moving picture experts group (MPEG) motion vectors or by first calculating the optical flow. Optical flow is a dense field of displacement vectors, which describes the translation of each pixel in a region. It is calculated using the velocities of pixel brightness patterns in a sequence of frames. Some commonly used approaches to extract optical flow are TV-L1 and MPEGFlow [66]. TV-L1 is the slowest to compute, but its performance is significantly higher than that of MPEGFlow, which is faster by a large margin [66]. Video motion can be categorised in two types, namely foreground and background motion. The foreground motion is caused by object motion, while the background motion is caused by camera motion. As a result, two types of motion features can be extracted and they generate different stimuli to the observer [20].

Camera-based motion features are content attributes that describes different camera movement, such as panning right or left, tilting down or up, and zooming in or out. Object-based motion features are content attributes that describe the motion of objects in the video. They have attracted much more interest in some recent video content analyses work [29]. Some of the most widely

used ones are SIFT-3D [67], HOG-3D [68], motion scale-invariant feature transform (MoSIFT) [69], spatio-temporal interest points (STIPs) [70], and improved dense trajectories (iDT) [71]. SIFT-3D and HOG-3D are extensions of SIFT [62] and HOG [64] descriptors into 3D. MoSIFT is another extension of the SIFT descriptor and it captures substantial motion information along with texture information by using optical flow to select SIFT features. STIPs is an extension of the Harris corner detector [72] to 3D, where HOG and histogram of optical flows (HOF) [73] are extracted and used as video features. The Harris corner detector is a combination of a corner and edge detector using a local auto-correlation function. Lastly, iDT features are content attributes that capture the motion of objects in the videos by using dense sampling and camera motion removing techniques. They are an improved version of dense trajectory features [74] with the difference being, the explicit computation of the estimation for the camera motion. iDT features, aggregated using Fisher vectors (FV), are currently the state-of-the-art hand-crafted motion features used on different video classification problems [75]. However, this performance comes at the cost of high computational complexity that becomes difficult to deal with on large-scale datasets.

D. **Scene-based features** are features that are extracted from a particular scene. Detecting scenes are important since it segments the video, and most of the visual-based features are dependent on it [29]. Scenes are detected and segmented before extracting the features. It is a challenging task since the transition from one scene to another is performed in more than 100 different ways [50]. The most used scene transitions are fades, hard cuts and dissolves. Fades are transitions in which a scene gradually fades out to a monochrome frame and another scene gradually fades in from a monochrome frame. Hard cuts are transitions in which a scene immediately stops and a different scene starts. Dissolve is a transition in which a scene fades out while another scene fades in.

   Current strategies for scene-based features detect fade, hard cut and dissolve scene transitions using a different method for each [76]. Fades are detected by calculating the first derivative of the luminance mean for monochrome frames. Hard cuts are detected by first converting the frames from the red, green, blue (RGB) colour space to HSV colour space in order to compute colour histograms to obtain their intersection [59]. Next, a sliding window and an adaptive threshold to frames with potential cuts are identified using a global threshold [76]. Dissolves are detected using the luminance variances of the scene before and after the dissolve scene transition in order to calculate a range from it. Thereafter, the first order difference of the luminance variance curve

is computed in order to identify if the difference is within the range calculated [76].

Other approaches to detect scene transitions includes using an audio and vision integration-based, as well as a fuzzy logic based approaches [77, 78]. The audio and vision integration-based approach uses the visual and audio content to detect the scene transition by selecting a scene boundary to be where the visual and audio content change simultaneously. The fuzzy logic based approach uses the features to construct fuzzy rules. For example, hard cuts are detected by computing colour histograms in the RGB colour space from frames and calculating the intersections between them. Fades are detected using the edge-pixel difference between consecutive frames, pixel differences, and intersection of colour histograms. Pixel differences are calculated in the RGB colour space using the Euclidean distance. Edge-pixel counts are calculated using a Sobel edge detector to detect edges [77].

2.2.1.4 Comparison of the low-level features in video domain

Table 2.1 summarises the advantages and disadvantages of the feature types described in this section.

Table 2.1. Comparison of features

Approach	**Feature type**	**Advantages**	**Disadvantages**
Text-based	OCR	Can extract video text not present in dialog	Computationally expensive
	Closed captions	Easy to extract, high accuracy when not produced using speech recognition	High dimensionality
Audio-based	Time domain, Frequency domain	Easy to extract, shorter in length and size	Difficult to differentiate similar sounds
Visual-based	Colour based	Easy to extract, useful when using cinematic principles	No spatial information, similar colour distribution regardless of the actual content
	Object based	Useful to describe the video content	Costly and constrained to a small set of objects

Table 2.1 continued from previous page

Visual-based	Motion based	Useful in a broad sense when identifying action movies	Difficult to distinguish between object or camera motion
	Video Scene	Not costly when processing some frames	Difficult to identify scenes, may not be accurate

2.2.2 Deep learning features in video domain

As a result of the massive number of videos being generated in modern times, it is unfeasible to rely on manual processing of multimedia data to solve a wide variety of multimedia problems [79]. Therefore, the task of automatically describing the content of a video has recently gained a lot of attention in video research [79]. Recent studies on video content analyses use deep-learning features due to their outstanding performance in different domains compared to hand-crafted features [22, 79]. These are deep convolutional neural network (CNN) activations that represent the appearance information, motion information, and sound information [22, 80–82]. These activations, also known as embeddings, represent discrete categorical variables in a low-dimensional learned vector of continuous numbers. They are extracted from two-dimensional (2D) or 3D deep CNNs that converts raw images into compressed representations while removing any redundancy. These networks also require fewer pre-processing steps compared to traditional methods [79], which makes it a practical solution for a vast number of tasks, especially when dealing with large-scale video datasets.

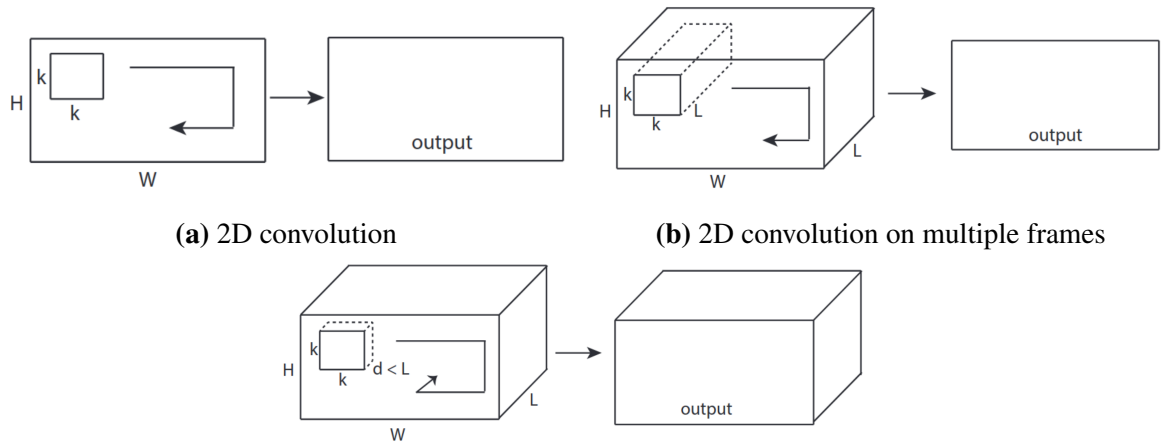
In the video analysis literature, 2D CNNs are usually pre-trained on large-scale datasets before being used on the target dataset. The datasets commonly used to pre-train these models are the ImageNet and Places datasets [83, 84]. The ImageNet and Places datasets are large-scale labelled datasets used for object and scene recognition. In particular, the ImageNet dataset contains images of generic objects, while the Places dataset contains images of scenes and places encountered in the world. These datasets are usually chosen for the models to learn generic object-centric or scene-centric features. Thereafter, the pre-trained models are utilised in transfer learning strategies.

In video analysis tasks, three commonly used transfer learning strategies are fine-tuning the 2D CNN models on the target dataset [85], knowledge transfer from the 2D CNN models to 3D CNN models [86], and using pre-trained 2D CNN models as a feature extractor [30]. The fine-tuning approach trains all layers or only the last layer of the target dataset. This is performed by using weights of the trained

2D CNN as an initialisation step and then executing the training method. The knowledge transfer approach uses the class probabilities of one model (2D CNN model) as a soft target for the other model (3D CNN model). The feature extractor approach generates embeddings using the pre-trained 2D CNN model given an input video file. These embeddings are used as a generic video descriptor. The feature extractor approach is closely related to the fine-tuning method, which only trains the fully connected layer (last layer) of the network, since the first layers remain intact. In contrast, the final layer specialises to the classes of the input video. It is also the commonly used approach in video retrieval and recommendation tasks that extracts and uses the non-textual video content features from the videos.

Generally in video analysis tasks, 2D CNNs models receive the spatial stream (static images) or temporal stream (multiple stacked optical flows) of the videos as input [80]. The spatial stream normally contains the object appearance and the scene appearance in the video frames. This information is represented by features extracted from the video frames, which is generated by the spatial convolutional layers. The temporal stream explicitly describes local temporal movement between video frames. The network, therefore, uses this stream to estimate the motion information present in the video explicitly.

On the other hand, 3D CNNs are trained on multiple video frames or short video clips using spatio-temporal convolutions [22]. These convolutional layers capture the spatial and temporal features of a video without the need for multiple stacked optical flows or recurrent neural network layers to extract temporal features. It is evident that the 2D and 3D CNN operations are very distinct, this is shown in Figure 2.2.



(a) 2D convolution **(b)** 2D convolution on multiple frames

(c) 3D convolution

Figure 2.2. Convolution operations with stride k performed by 2D and 3D CNNs models on a video. a) The application of 2D convolution on a single frame with shape $H \times W$, results in a frame. b) The application of 2D convolution on multiple frames, L, with shape $H \times W$, also results in a single frame. c) The application of 3D convolution on a short snippet of a video, results in another short snippet, thus preserving the temporal information contained in the original short snippet. (From [22], © 2015 IEEE)

A single frame or multiple frames from a video as input to a 2D CNN results in an image. On the other hand, a video data volume (short snippets) as input to a 3D CNN, generates another video data volume with the temporal information preserved.

One of the very first approaches exploiting CNNs to perform large-scale video classification tasks used only 2D CNN architectures composed of spatial convolution layers [87]. The proposed CNN architectures use as input multiple frames from the spatial stream of the videos. However, its performance is relatively lower compared to state-of-the-art hand-crafted representations on a popular real-life action recognition dataset, namely UCF-101 dataset [88]. The UCF-101 dataset comprises of 101 action classes captured by different users in the wild. It is focused on human actions that can be divided into five types, namely sports, body-motion only, person-person interactions, person-object interactions, and playing musical instruments. The performance obtained using the proposed models was a clear indication that given the nature of the 2D convolution operation, the models did not properly extract the temporal information present in the videos. Therefore, 2D CNN models, which do not use the temporal stream from videos, present significant limitations in video processing tasks on a motion-oriented dataset [89]. They neglect the essential temporal information in the videos and only effectively capture the spatial information of each frame.

A video is beyond a simple set of static visual appearances. It is not only characterised by diverse and complex visual information in each frame, but also carries embedded temporal information across frames, such as long-term events and short-term actions [90–92]. This temporal information is the very reason for cinema existence [26], and is a critical aspect for video analysis [93] and necessary to accurately characterise videos aesthetically [20]. This notion is supported and proven in automatic analysis of video content studies conducted over the past few years [15–19]. They show that the temporal information is essential for accurate video understanding tasks since its unavailability leads to a drop in performance. For example, the proposed two-stream CNN architecture in [89] uses single video frames and multiple stacked optical flow frames as input. This architecture is composed of two identical 2D CNN models, namely a spatial stream CNN model and a temporal stream CNN model. The spatial stream CNN model extracts the spatial information present in the input video file by learning it from individual video frames at a time. The temporal stream CNN model extracts the temporal information present in the input video file by learning it from multi-frame optical flow. As mentioned in Section 2.2.1.3, optical flow represents the motion information contained in the video frames explicitly. As a result, the performance obtained by this approach shows that the temporal information is indispensable when performing action recognition since the temporal stream CNN model significantly outperforms the spatial stream CNN model. The fusion of these two models further improves the performance of the system leading to competitive state-of-the-art accuracy. Although this architecture showed promising results by using stacked optical flow to capture the motion information of the video, there are still limitations because the architecture only uses 2D CNN models. For this reason, later studies proposed to use spatial convolutions with recurrent neural network layers on top [80]. The recurrent neural network layers are used to extract temporal features from the spatial features that in turn model high-level variation of the video frames. However, fine low-level motion information is critical in many cases and these models may not be able to capture them. In addition, these models are also expensive to train, and the input should be pre-segmented.

Many recent studies adopted the use of 3D CNN architectures to effectively learn spatio-temporal properties of videos [15]. This is due to their capability to simultaneously model the temporal and spatial information contained in the video frames. Figure 2.3 shows the commonly used video architectures to model a video for video analysis tasks.

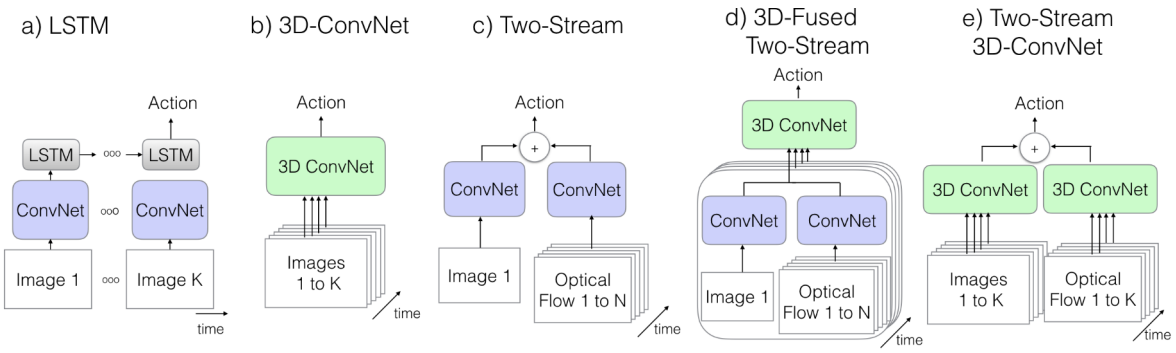


Figure 2.3. The commonly used video architectures to model a video where K is the total number of frames, and N is the total number of neighbouring frames. (From [15], © 2017 IEEE)

As can be seen in Figure 2.3, the architectures that use 3D CNNs are just a natural extension of the architectures that use 2D CNN. These architectures differ on the input stream that can be RGB or optical flow frames. The architecture that is more related to how humans perceives videos is the two-stream inflated 3D ConvNets (I3D) [15]. This architecture can handle multiple RGB frames from videos along with multiple optical flow frames.

The study that proposes the I3D architecture performs an experimental evaluation of the generalisability of the features generated by the model architectures shown in Figure 2.3 [15]. The models are first pre-trained on split 1 of the kinetics human action video dataset [94] then fine-tuned on the split 1 of UCF-101 dataset and human motion database (HMDB)-51 [95]. The kinetics dataset is a large-scaled labelled dataset that includes a total of 400 action classes. It is focused on person actions such as punching, person-person actions such as kissing, and person-object actions such as washing dishes [94]. The UCF-101 dataset is another large-scaled video action dataset that is smaller compared to the kinetics dataset. Lastly, the smallest and the most complex dataset being used in the experiment is the HMDB-51 dataset. This dataset is composed of 51 distinct action classes that occur in many clips in the exact same scene. The distinct action classes can be divided into five types, namely general body movements such as *jump*, general facial actions such as *laugh*, body movements with object interactions such as *shoot gun*, facial actions with object manipulation such as *smoke*, and body movements for human interaction such as *sword fight* [95]. The outcomes of the experimental evaluation of features show that the video features generated by the two-stream I3D architecture performs better than the features generated by the other four architectures shown in Figure 2.3. It is worth noting that the number of frames used to train each model differs from model to model. The I3D model has a high temporal resolution compared to other models in the study since it is trained

on 64-RGB-frame and 64-flow-frame video snippets at 25 frames per second (fps). The other models are trained with lower temporal footprint (input frames) because of their limitations to handle higher number of frames. Additionally, the I3D model is much deeper compared to the other 3D CNN models used in the comparison.

Aside from the experimental evaluation of features, the study in [15] also performed an experimental comparison of the architectures shown in Figure 2.3. It trained the 3D CNN models from scratch and the 2D CNN models were pre-trained on ImagedNet. The RGB I3D model performs better than the optical flow I3D model on split 1 of the miniKinetics dataset, but performs worse on split 1 of the other two datasets, namely, UCF-101 and HMDB-51. However, the RGB and optical flow I3D models obtain better performance compared to the other architectures across the three datasets. Furthermore, the best performance is obtained by late fusion of the RGB and flow I3D models prediction scores. It should be pointed out that the performance obtained by the flow I3D model comes at a very expensive cost of computing accurate optical flow. From the reported results, it is also noted that when videos have much more camera motion, a model using optical flow frames performs worse than or equal to its variant that uses RGB frames. This outcome is in line with results reported in previous studies that found the optical flow to be noisy in such cases [80]. In addition, when the accuracy of the I3D models are averaged over three splits of the UCF-101 and HMDB-51 datasets the performance of the RGB I3D model is similar to the performance of the optical flow I3D model. Nevertheless, the performance of the two I3D models are better than several state-of-the-art models, and once again the best performance is attained by the two-stream I3D model.

A video representation is not perceptually complete if it does not contain the aural information from the video. Recently, there has been an interest in large-scale audio classification using CNN model architectures given as input the raw audio from videos or log-Mel spectrogram features extracted from the audio stream [96, 97]. These studies have been motivated by the state-of-the-art performance of CNNs in image classification when trained on large amounts of image data. A log-Mel spectrogram is a visual frequency domain representation of the audio signal obtained using Fourier transforms and converting the frequencies to Mel scale [97, 98]. This representation indicates how the frequencies of the audio signal varies with time. It is a commonly used pre-processing step when using 2D CNNs since audio signals are one dimensional (1D) signals (amplitude with respect to time), whereas, log-Mel spectrograms are 2D signal (amplitude of a particular frequency at a particular time) [97].

CNN architectures are usually used to perform acoustic action recognition, genre classification, and object and scene classification [96–98]. These networks are also usually pre-trained on large datasets of unlabelled or labelled sound data from videos to learn high-quality audio embeddings (features) [96,97]. For instance, audio embeddings from a VGGish model trained on the YouTube-8m (8 million videos) dataset [97]. This model is a very deep 2D CNN that uses as input log-Mel spectrogram patches extracted from the audio signal. The sound representation learnt by this model generalises well, as evident by outstanding results reported in video retrieval and classification tasks [30]. Another example is audio embeddings from a proposed audio-visual CNN architecture named SoundNet architecture [96]. These embeddings have been used in video analysis tasks such as video ordering and action recognition [99]. The SoundNet architecture is composed of two 2D CNN models for visual recognition and one 1D CNN model for natural sound recognition [96]. This architecture is shown in Figure 2.4.

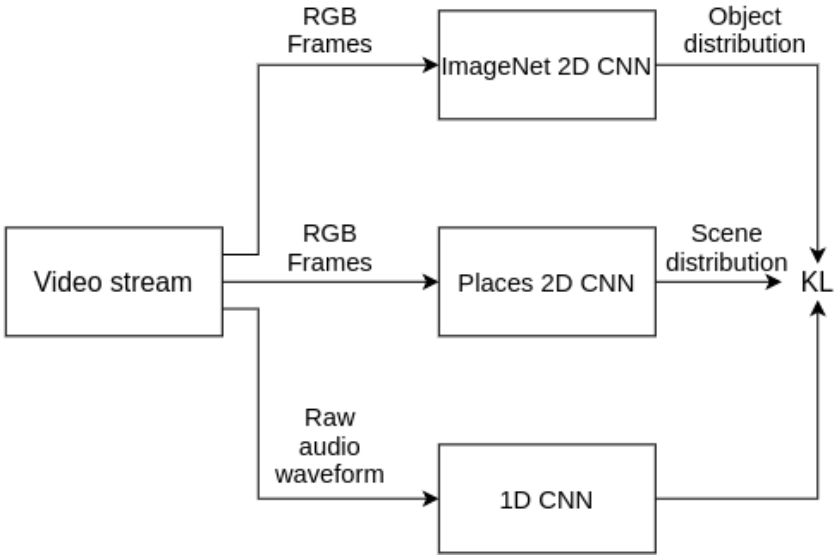


Figure 2.4. SoundNet architecture trained by transferring object and scene discriminative knowledge into the sound networks while optimising Kullback-Leibler (KL) divergence. The object and scene distribution, are generated by 2D CNN models pre-trained on imageNet and Places205 datasets. (Adapted from [96])

This architecture is a student-teacher network that transfers discriminative knowledge from the object and scene recognition networks into a deep 1D CNN architecture, which performs natural sound recognition. This is performed to exploit the natural audio-visual synchronisation. The SoundNet architecture is trained using two-million (2M) unlabelled videos. The main objective of the proposed study was to develop a neural network (NN) that learns a generic natural sound representation. The

generalisation of this descriptor is proven to be effective given the performance obtained in several applications such as action recognition and video ordering tasks [99].

The idea of learning a video representation, using the aural and visual signals separately, was extended to learning hierarchical audiovisual concepts [98]. For example, the audiovisual Slow-Fast (AVSlowFast) architecture proposed for video recognition tasks [98]. It is an architecture that models vision and sound in a unified representation using a hierarchical audiovisual synchronisation method. The architecture is shown in Figure 2.5.

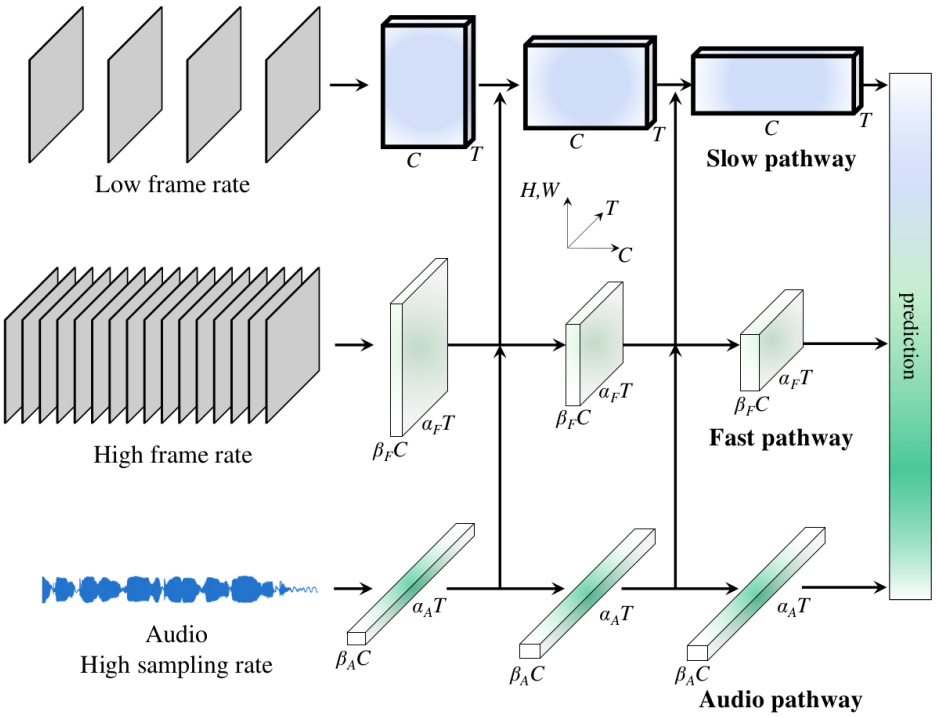


Figure 2.5. Audiovisual SlowFast model architecture that is composed of two pathways for the visual stream and one pathway for the audio stream from the video. The visual pathways are slow and fast pathways that extract spatial and temporal features from the videos. The dimensions of their kernels are denoted by the temporal T and channel C sizes, as well as the speed α_F and channel β_F ratios. The audio pathway is an even faster pathway with respect to the visual pathways. The dimensions of its kernel are denoted by the temporal T and channel C sizes as well as the speed α_A and channel β_A ratios. (From [98], © 2019 IEEE)

As can be seen in Figure 2.5, the visual and aural features are fused at multiple CNN layers to obtain a unified audiovisual representation. This architecture is composed of three pathways, namely a slow visual pathway, a fast visual pathway, and an audio pathway. The slow visual pathway extracts

features from the videos that capture the semantic contents that do not change at a fast rate. The fast visual pathway extracts features from the videos that capture the fast motion information contained in them. Lastly, the audio pathway extracts features that capture even finer temporal information compared to the slow and fast visual pathways. This pathway is faster than the fast visual pathway. Different from the SoundNet architecture but similar to the VGGish model, the 2D CNNs of the audio pathway receives as input log-Mel spectrogram patches extracted from the audio signal. Various experiments are carried out in the study that proposed the AVSlowFast model architecture. For action classification experiments, the AVSlowFast model architecture achieves state-of-the-art accuracy on the EPIC-Kitchens, Kinetics-Sounds, and Charades datasets. Additionally, the proposed architecture also attained state-of-the-art accuracy in action detection experiments using the AVA dataset. Furthermore, the generalisation of the features learnt by the AVSlowFast model is assessed on the HMDB-51 and UCF-101 datasets, similar to the study that proposed the I3D model architecture. However, the model is pre-trained on the Kinetics-400 dataset using a self-supervised learning (SSL) approach instead of a supervised learning approach. This is chosen to investigate the quality of self-supervised audiovisual features learnt with the AVSlowFast model. It was found that only fine-tuning the last fully connected layer on the HMDB-51 and UCF-101 datasets, the AVSlowFast model provides features that obtain significantly better performance compared to state-of-the-art SSL feature methods. Inversely, when fine-tuning all layers on the datasets, the features learnt by AVTS model slightly outperform the features learnt by the AVSlowFast model on the HMDB-51 dataset.

According to applied media aesthetic, the temporal and visual information from videos serve as crucial elements that have aesthetic, informative, and emotional effects on users, which can consequently influence user choices on videos [7, 12, 100]. It also produces a rhythm that is likely to stimulate user preferences for movie trailers [101]. As a result of all these findings, the exploited features used to represent the videos in this dissertation are neural network embeddings from pre-trained models. These features capture the visual, aural, and temporal information from videos. A hybrid video recommendation system uses the features comprehensively to exploit their availability and provide more accurate personalised video recommendations to users.

2.2.3 The semantic gap

The semantic gap is the dissimilarity between the low-level features and the semantic properties of the multimedia content that users interpret. This problem has recently received great attention in multimedia retrieval systems, given the exponential growth of video content available online [102].

Several studies conducted by the multimedia information retrieval community have shown that it remains challenging to bridge the gap between low-level features and high-level features [103].

In the context of video recommender systems, as mentioned in Section 2.2, these systems typically exploit high-level features extracted from post-release textual metadata of the videos for rating prediction. These features are primarily chosen because of the semantic gap. These video recommender systems assume that the stylistic properties of videos (such as the features mentioned in Section 2.2) do not significantly influence the user preferences compared to the high-level semantic features of the videos (such as genre and director metadata). Recent works carried out by the recommender system community, however, indicates the complete opposite [102]. The user preferences are mainly influenced by the visual properties of the items (low-level features) instead of their semantic properties (high-level features). However, low-level features may not be semantically understandable which can lead to recommendations that are difficult to explain [21].

Nevertheless, the semantic gap can be narrowed using low-level features, which achieve results that are semantically understandable. It can also be narrowed by leveraging the features oriented in a manner where the correlations between the human labels are learned as much as possible [102]. In this dissertation, the low-level features are chosen according to these principles, however to quantify the semantic gap it would be necessary to perform a user-centric online experiment [102], which is out-of-the-scope of this research work.

2.3 FEATURE AGGREGATION METHODS

As mentioned in Section 2.2, current methods for video content analysis rely on the extraction of frame-level and video-level features from consecutive frames. A video-level descriptor is created by aggregating the frame-level and video-level set of features over time using several aggregation methods [22, 51, 104–106]. Common temporal feature aggregation approaches are statistical summarisation, bag-of-visual-words, vectors of locally aggregated descriptors (VLAD), FV and recurrent neural network (RNN) [82, 104, 107–111]. Statistical summarisation is the simplest method to obtain a video-level descriptor from the set of extracted features that represent the video content. It discards the temporal ordering of the features in the video and only captures their distribution. The video-level descriptor is obtained by calculating a statistic summary of the video features using statistical functions such as maximum, mean, median, median absolute deviation, and variance [107].

Bag-of-visual-words, FV, and VLAD are feature encoding methods that obtain a video-level descriptor by quantising the video features. They are traditional orderless aggregation methods, which depend on a codebook that can be learned in an unsupervised, discriminative, or end-to-end manner [112, 113]. Typically, the Bag-of-visual-words method quantises the features by generating visual words using the k-means clustering algorithm [108]. The FV method quantises the features using a Gaussian mixture model (GMM) [110], and the VLAD method quantises the features by generating centroids using the k-means clustering algorithm [109]. Training these encoding modules in an end-to-end manner by integrating them in a neural network has been gathering interest and gaining importance as significant improvements have been noticed for a video-level descriptor and classification [51, 114].

Recent studies have been exploiting RNN models, such as long short-term memory (LSTM) and gated recurrent unit (GRU) for sequential aggregation of the video features to obtain a video-level descriptor [82, 105, 106, 115]. These models capture the temporal ordering of the video features explicitly, given its ability to learn long-term correlations in the time domain [80]. However, the training of RNN models require a large amount of data where the input must be pre-segmented. In addition, the video-level descriptors obtained from these models show similar results to temporal mean or max pooling methods [82, 105].

2.4 FEATURE FUSION METHODS

To further improve the video recommendation performance, visual, aural, and motion features that are extracted from the video content, as well as the textual features extracted from the metadata information, should be fused. This operation enriches the training and recommendations. In the literature of video content analysis, two general fusion strategies are used, namely early fusion and late fusion [116].

2.4.1 Early fusion

Early fusion is a scheme that combines unimodal features into a single multimodal representation before training a model. This fusion strategy obtains a truly multimedia feature representation, since it aims to map different video content features to a unified one [116]. Some of the commonly used early fusion methods are concatenation (concat) of unimodal feature vectors, summation (sum) method and maximum (max) method [14, 85]. These methods combine unimodal feature vectors associated with the same video to obtain a fused multimedia representation. The concatenation of unimodal feature vectors combines different feature vectors to form one large video level descriptor, which is used as input for model training. This enables the machine learning algorithm to model the concatenated features.

The sum method averages the feature set in a homogeneous feature space in order to form the final video-level descriptor [85]. The max method is similar to the sum technique, with the only difference being the way the features are merged [85]. It selects the highest value from the corresponding features to generate the final video-level descriptor.

The main advantage of early fusion techniques is that they take into account the correlation and interactions among the features of each modality. Furthermore, they only require one training phase and one single model for inference. The disadvantage is that it is a challenge to combine features into a common representation, especially when working with features that represent information that is diverse and heterogeneous. This is the case for multimodal information from video content. One approach to solve this problem is to use correlation analysis methods such as canonical correlation analysis (CCA) or its extensions [7,117,118]. CCA assumes that two sets of data have some underlying correlation in order to jointly learn the shared latent factors and reduce the dimension across two or more heterogeneous feature spaces [7]. This can close the **heterogeneity gap** encountered in the multimedia information [118]. Consequently, correlation analysis methods have recently drawn significant attention from the multimedia research community [118]. The general approach for early fusion is illustrated in Figure 2.6.

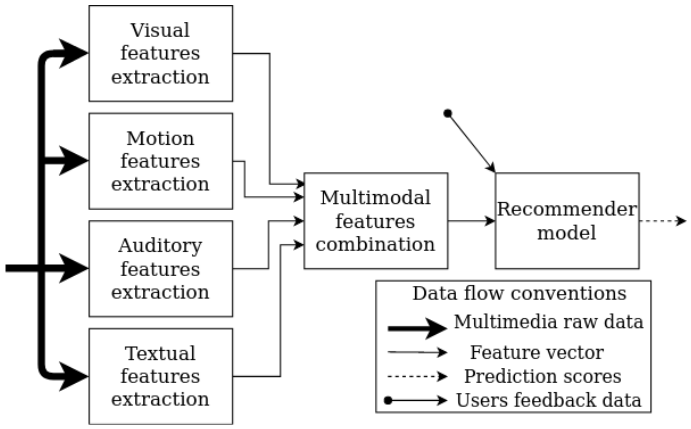


Figure 2.6. The general approach for early fusion. Visual, motion, auditory, and textual features are fused before training the model. (Adapted from [116])

2.4.2 Late fusion

Late fusion is a scheme that combines the prediction scores of separate models into a more accurate final set of results. Each model is trained with a single modality . This method focuses on the strength of the different modalities individually. Instead of obtaining a single multimodal representation at feature level, this method obtains a multimodal semantic representation. Some of the commonly

used late fusion methods are a simple or weighted score average, bilinear product, and learning-to-rank techniques [119]. The major advantage of late fusion techniques is that the performance divergence, usually encountered when working with heterogeneous content features, can easily be addressed because it does not depend on the representations of different modalities [14]. The significant disadvantage is that late fusion methods have the potential loss of correlation in mixed feature space because they do not analyse the correlation across features. Moreover, the combination of the prediction scores requires that each modality has a separate trained system, and that an additional learning stage is necessary for the combination. Figure 2.7 shows the general approach for late fusion.

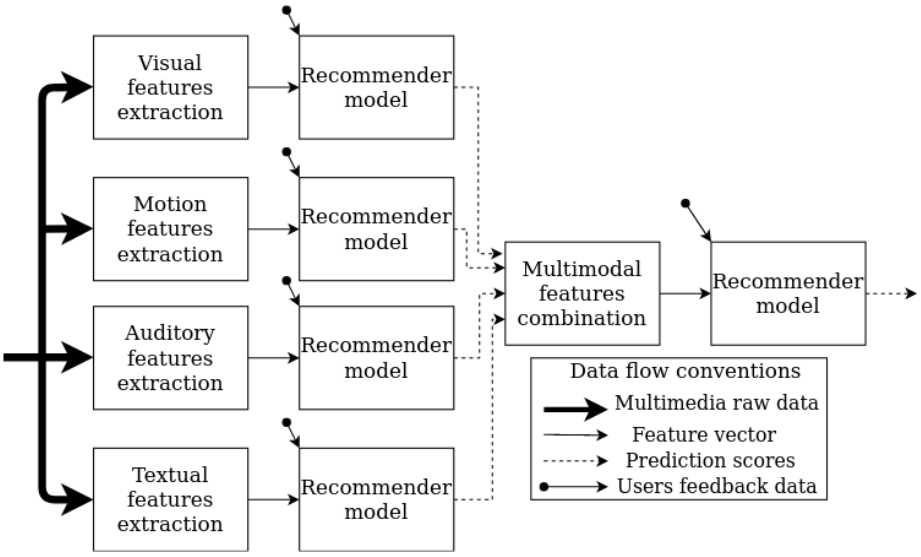


Figure 2.7. The general approach for late fusion. Different features are used by separate recommender models where the rating prediction scores of each model is combined to yield a final prediction score. (Adapted from [116])

2.5 RECOMMENDER SYSTEMS

Recommender systems are a subset of information filtering systems that aim to recommend the most relevant and appropriate items to users [120]. This is achieved by predicting the preference or rating a user would give to an unseen item. They can be used in different application domains such as e-commerce, music and video recommendation. In this research study, the domain of interest is video recommendation. Therefore, the term *item* will be used interchangeably with *video*, which refers to the element recommended to users. Table 2.2 shows the commonly used symbols for recommender models.

Table 2.2. Common symbols for recommender models

Symbol	Definition
U	Set of users
I	Set of items
F	Set of features
r_{ui}	Feedback of user u to item i
N	Number of recommended items

In order to make predictions, recommender systems can use different types of prior information. The most common types of prior information used in literature are user interactions with items, the information about the content of the items, and when available, the demographic information about the users [5]. This information is also known as profile. It is widely used in recommender system data structures, such as matrices, where U, I and F are the sets of all users, items, and features, respectively [121]:

A. **User rating matrix** (URM) is a $|U| \times |I|$ matrix used to represent the explicit or implicit user feedback to items. Depending on the problem at hand, the values of each cell usually are real numbers or binary (0 or 1). Real numbers are used when ratings that a user gave to items (explicit feedback) are represented. Binary values are used when only the implicitly collected observation of user interactions with the items (implicit feedback) are represented.

B. **Item content matrix** (ICM) is a $|I| \times |F|$ matrix used to represent the profiles of the items. These profiles are composed of features, which represents the information about the content of the items. These features can be of different types, such as real-valued, binary, or strings. Usually for strings, such as video metadata, the values are encoded in a binary profile. This means that for an item at row i and a feature at column j of the ICM, the value 1 or 0 represents the presence or absence of that feature, respectively.

C. **User content matrix** (UCM) is similar to the ICM with the only difference being that instead of the matrix being composed of the profiles of the items, it is composed of the profiles of the users. Its shape is $|U| \times |F|$, where in this case F is the set of all features attributed to users. Usually, this matrix represents the age, gender or other demographic data of the users which are encoded in a binary profile in the same fashion as the ICM.

As described above, recommender systems can use different types of prior information to solve the recommendation problem. For this reason, the availability of these types of prior information defines which techniques to use in order to generate recommendations. When only past users interactions with items are available, the CF approach is used. The CB approach is used when the available information is limited to the content of the items and about the target user. Alternatively, hybrid techniques are used when two or more types of prior information are available. For example, past user interactions with items, and information about the content of the items or demographic information about the users, or both. The hybrid approach is used to generate robust and higher quality recommendations. Figure 2.8 shows the structure of the different recommendation techniques.

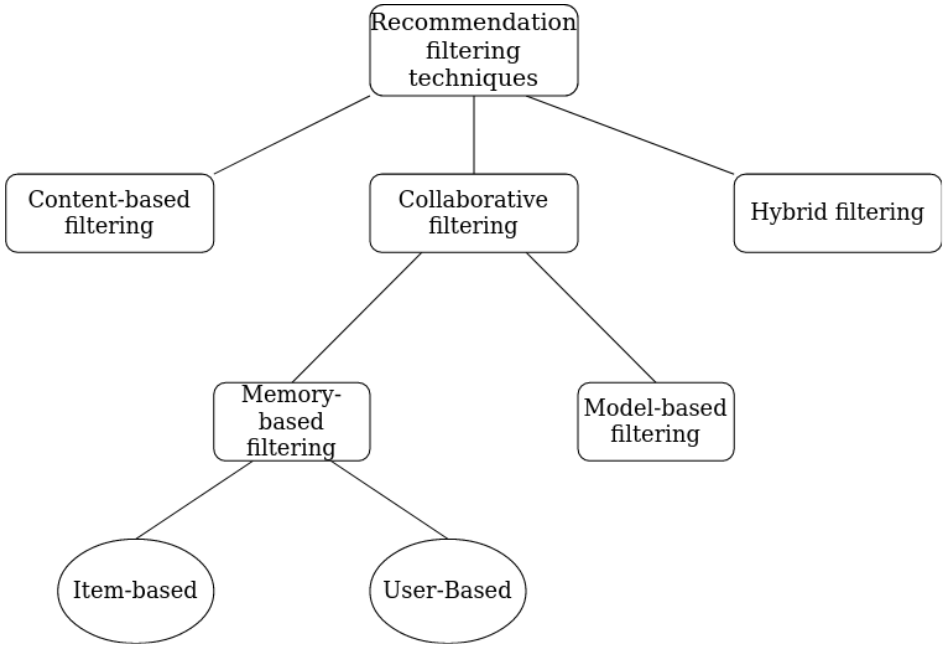


Figure 2.8. The structure of the three recommendation filtering techniques, namely content-based, collaborative, and hybrid filtering. (Adapted from [121])

2.5.1 Collaborative filtering (CF)

CF is a domain-independent prediction technique that evaluates and filters items according to the opinions of different users [122]. It assumes that items are rated similarly by users with similar preferences. Therefore, it does not require information about the content of the items since the URM is enough to predict the ratings and generate recommendations. This technique is the most frequently used in the recommendation literature [6–8, 122]. It usually achieves state-of-the-art accuracy due to the high correlation of the observed ratings across multiple users and items [6–8].

Ratings in a CF system can be gathered explicitly, implicitly, or both. They are user feedback obtained when the user rates an item according to a defined rating score through the interface of the system (explicit ratings); or inferred from user's actions during their interactions with the system (implicit ratings). For example, the playback times of videos watched by the user. Often explicit feedback is more reliable compared to implicit feedback. However, in many real world systems, the ratings can only be inferred from the user's implicit feedback. Furthermore, the type of rating gathered from this feedback influences the design of the recommendation model. These ratings can be defined as follows [122]:

A. **Scalar ratings:** consists of numerical ratings or categorical ratings. For example, 0.5 to 5 star rating and text opinions such as neural, agree, or disagree.

B. **Binary ratings:** represent disliked and liked values defined as 0 **and** 1, respectively.

C. **Unary ratings:** indicates if a user observed, or consumed, or liked the item. When this rating is absent in a cell at position (u,i) of the URM, where u and i are the user and item, respectively, it means that the user has not interacted with the item yet.

There are two main approaches to collaborative filtering, namely memory-based and model-based collaborative filtering.

2.5.1.1 Memory-based collaborative filtering

Memory-based collaborative filtering is a technique that performs predictions using all the stored ratings and similarity coefficients directly [123]. This method is also known as the neighbourhood-based method, because the relevance scores for any *user-item* pair are based on the user neighbourhoods. A very popular and widely used algorithm to perform this task is the k-nearest neighbours (kNN) algorithm. This algorithm can be implemented in two different ways [123]:

A. **User-based:** user-based recommendation methods predict the ratings and generate recommendations to a target user, u, by finding a set of similar users based on their item preferences. The idea is that if two users, u and v, are similar, then user v known preferences for item i can be used to predict the unobserved preference of item i for user u. Hence, it is necessary to define a similarity function, $sim(u,v)$, that computes a similarity score, which represents how much a user u and a user v have similar tastes. This score is usually stored in a *user-user* matrix. The prediction of the rating of user u for item i is obtained by using the calculated scores to obtain

the contributions of the ratings of the k most similar users on this item. This can be defined as

$$\hat{r}_{u,i} = \sum_{v \in U} \frac{r_{v,i} \cdot sim(u,v)}{|sim(u,v)|}, \tag{2.1}$$

where $\hat{r}_{u,i}$ is the predicted rating for user u and item i .

B. **Item-based:** item-based recommendation methods are similar to the user-based approach. The only difference is that instead of finding a set of similar users, a set of similar items is determined based on the preferences of other users to the items. This method assumes that if various users have rated two items i and j together, then these items might be similar. Therefore, a similarity function, $sim(i,j)$, that computes a similarity score between the target item i and another item j should be defined. This score is usually stored in an *item-item* matrix. After that, the predicted rating $\hat{r}_{u,i}$ for user u and item i is calculated by taking the weighted average of the k most similar items based on the computed similarity scores as

$$\hat{r}_{u,i} = \sum_{j \in I} \frac{r_{u,j} \cdot sim(i,j)}{|sim(i,j)|}. \tag{2.2}$$

2.5.1.2 Model-based collaborative filtering

In some application domains, it is necessary to handle a large-scale dataset, which makes memory-based methods an impracticable approach [123]. To solve such problems, model-based methods that do not need the whole dataset to generate recommendations are chosen. This method builds a predictive model of user rating to provide item recommendation. Such models use machine learning and data mining techniques to extract general behavioural patterns from the dataset. Predictions for missing values in the URM are produced and recommendations to users are generated. Some of the commonly used models, include Bayesian methods, decision trees, latent factor models, and rule-based models [121]. When necessary, the parameters of these models are learned and tuned within the context of an optimisation framework.

2.5.2 Content-based filtering

In contrast to collaborative filtering, content-based approaches analyse information about the users, such as age and location, or information about the content of the items such as video metadata and video-level descriptors, or both to generate the most relevant recommendations [5]. This is performed by using only the target user's interactions on other items. It is assumed that the user choice is influenced by a combination of individual user attributes and certain features extracted from the content of the items previously consulted [5]. For example, if it is known that a user has rated the movie *Batman* highly, it is likely that user may prefer the movies *Superman* and *Spider-Man* because the item descriptors of these movies contain similar genre metadata keywords.

The CB approach typically is structured in three main steps namely [124]

1. **Pre-processing and feature extraction**: pre-processing data and extracting discriminative features that captures the item information is the most important step in order to generate high-quality recommendations to users based on their past behaviour. Discriminative features are the highly predictive item features of user interests. These features are used to build the ICM presented in Section 2.5.
2. **Learning user profiles**: the preferences of the users are extracted from the user's preference data and a model is used to learn these preferences along with the item's features. One of the most commonly used models is the Item-based k-nearest neighbors content-based filtering (ItemKNN-CBF) algorithm. Similar to the memory-based collaborative filtering, this algorithm only requires defining a similarity function $sim(i, j)$, with the only difference being that in the CB approach the function describes how close two items i and j are in terms of their descriptors. Alongside the ItemKNN-CBF algorithm, the most widely used similarity function is the cosine similarity, due to its high performance when working with any type of features (binary or dense features).
3. **Filtering and recommendation**: the learned model from the previous step is used to predict the preference or rating users would give to items and recommend these items. For user u and item i this prediction is computed as

$$\hat{r}_{u,i} = \sum_{j \in I} \frac{r_{u,j} \cdot sim(i, j)}{|sim(i, j)|}, \tag{2.3}$$

where $\hat{r}_{u,i}$ is the predicted rating.

The main advantage of the CB approach is the ability to overcome the biggest challenge of CF, namely the item cold start problem. As seen in Section 1.1.1, the item cold start problem occurs when the preference data of an item is entirely unavailable, making the CF approach inapplicable. Hence, when using the CB approach as long as all the discriminative features that represent the items have been extracted, any item can be recommended, even if any user did not previously watch it. However, this recommendation approach suffers from various problems, such as high sensitivity to user input, limited content analyses, and over-specialisation. This results in recommendations that are too similar to what the user liked in the past which consequently are not the most useful because they are not novel [2].

2.5.3 Hybrid filtering

In order to overcome some of the limitations of the CF and CB approaches, and improve recommendations not only in item warm-start scenarios but also in item cold-start scenarios, the CF and CB approaches are combined within a single model [7]. This combination is performed using hybrid approaches, as used in many successful recommender systems. The hybridisation takes place in two phases [125]:

1. Performing item filtering using CF and CB models to generate candidate recommendations
2. Use of hybridisation methods to combine these sets of recommendations in order to generate the final recommendations for users

Some of the commonly used hybridisation methods are implemented in the following ways [125]:

A. **Weighted:** This method is similar to ensemble analysis in standard classification tasks where the predictions of different recommendation techniques are fused by calculating the weighted aggregates of the predictions to generate a single recommendation.
B. **Switching:** This method chooses the recommendation technique to be used at any given point in time based on several criteria in order to adapt and change in case of failure. Hence, the system must define a switching criteria that reflects the recommender's ability to generate good results. For example, switching between CF and CB approaches in order to avoid the item cold-start problem.
C. **Cascade:** This technique is a step-by-step process where a recommendation technique refines the recommendation list generated by another recommendation technique.
D. **Mixed:** Recommendation lists generated by different recommendation techniques are presented to users at the same time. Each item has various recommendations associated with it.
E. **Feature combination:** This technique treats the data from different recommendation techniques as an additional feature within a unified recommendation model. For example, using CB filtering features namely the information about the content of the items as one of the features in a CF model.
F. **Feature augmentation:** The predictions from one recommender model are used as an additional input by another recommender model.

G. **Meta-level:** This method uses the internal model built by one recommender algorithm as input for another recommender model. For instance, a CF model, which uses a model learnt using content-based features, in order to compute predictions while solving the sparsity problem.

2.5.4 Comparison of the recommendation filtering techniques

Table 2.3 summarise the advantages and disadvantages of the recommendation filtering techniques discussed above.

Table 2.3. Comparison of recommendation approaches

Approach	Advantages	Disadvantages
Collaborative approach	Not dependent on the content information Can work with any type of item whose information is unavailable or difficult to analyse No overspecialisation problem	Requires enough users to provide satisfactory results Suffers from the cold start problem
Content-based approach	Not dependent in the number of users to make recommendations No item cold start problem Easy to explain why the items were recommended Adjust the recommendations quickly according to changes in user preference Ensures privacy	Overspecialisation problem Knowledge of the field is often necessary Suffers from the limited content analysis problem
Hybrid approach	Can solve the cold start and overspecialisation problems altogether	Complexity is increased since it requires additional settings Often cannot explain why an item was recommended to a user

2.5.5 Similarity function

As presented in Section 2.5.1 and 2.5.2, often collaborative and content-based filtering recommender systems require the computation of the similarity between two profiles. Some of the most commonly used similarity functions are the cosine similarity, the asymmetric cosine similarity, the Dice-Sørensen similarity coefficient, the Jaccard coefficient, the Tversk similarity, and lastly the Euclidean similarity function [126]. The choice of the similarity function to be used depends on the item or user profiles at hand. As for example, set based similarities such as Dice-Sørensen similarity coefficient, Jaccard coefficient and Tversk similarity are only applicable to profiles composed of binary attributes. On the other hand, cosine based similarities are applicable to any type of data, and Euclidean similarity performs better when working with dense and continuous data.

2.5.6 Evaluation of recommender systems

In order to implement a reliable recommender system that works well in both laboratory and production environments, it is fundamental to design a suitable evaluation workflow and use an adequate quality measure to assess the performance of recommender systems. There are three main approaches to evaluate recommendation systems, namely online evaluations, user studies, and offline evaluations [127]. The most reliable evaluations of the quality of the recommendations generated by a recommender system are user studies and online evaluations. These two types of evaluations involve users, and the main difference between them is in how the users are recruited for the studies. Although these evaluations are the most reliable, it is hard and costly to enrol a substantial number of users for evaluation purposes. Secondly, deployed systems are usually not publicly available and are limited to some scenarios that it can handle. Lastly, the generalisability of the system is limited because the actions of the test users in the evaluation process cannot be fully controlled [127]. For these reasons, offline evaluations are the most popular evaluation techniques used to assess the quality of recommendation systems. A range of standardised frameworks and well-established evaluation measures have been proposed for this case. However, one major drawback of offline evaluations are that they do not measure future interactions of the users that should reflect the constantly changing user preferences. In fact, various user studies and industry reports indicate that maximised offline performance does not necessarily lead to better value for users or providers [3, 128]. Nevertheless, despite of these disadvantages, offline evaluations are still broadly used and accepted because they can be used to reduce the list of candidate algorithms that will be assessed in actual online experiments [3, 127]. This allows for faster innovations. In addition, the metrics are easy to understand and statistically robust. In this research study, offline evaluation metrics are presented and described in detail as well as used in later sections.

2.5.6.1 Accuracy metrics

In the literature, different methods have been used to assess the performance of recommender systems in terms of accuracy measures [129]. The challenge faced in this field by researchers is to select the best metric to measure the quality of their recommender systems in a specific context. However, in most cases, the main goal is to generate a ranked list of top-n relevant items. Accuracy metrics such as predictive, classification, and rank-aware accuracy metrics are the most used metrics in literature [129]. They measure the fraction of correctly predicted values.

1. **Predictive accuracy metrics:** measure how close the ratings, predicted by the recommender system, are to the real ratings given by users. Some of the commonly used error-based metrics are the mean absolute error (MAE), the mean square error (MSE), and the root-mean-square error (RMSE) [130].
2. **Classification accuracy metrics:** measure how often the recommendation system generates correct or incorrect recommendations through ground-truth values as in binary classification problems. These metrics are useful when evaluating a recommender system that uses implicit feedback to infer user preferences. In such systems, the positive class 1 represents the relevant items to the user and the negative class 0 represents the items that are not relevant. Some of the most commonly used classification metrics are the precision at top-N recommendations ($P@N$) and the recall at top-n recommendations ($R@N$) [129, 130]. These metrics provide the probability that a recommended item at top-N is relevant and the probability that a relevant item at top-N is recommended.
3. **Rank-aware accuracy metrics:** quantify the ability of the recommendation system to generate an ordered list of items to recommend, which corresponds to how the users would have arranged the same items according to their preferences. These metrics are more suitable to evaluate recommender systems in a domain where the relevance of the recommendations are non-binary. These metrics may be excessively sensitive in domains where the user will not be interested in the ranking of the items, apart from their relevance. Some of the most common and widely used rank-aware metrics are the mean average precision (MAP), mean reciprocal ranking (MRR), and normalised discounted cumulative gain (NDCG) [7, 130]. In addition, when the number of items is large, sampled rank-aware accuracy metrics that provides a good estimate of the exact metrics can also be used at the cost of increased variance [131].

2.5.6.2 Beyond-accuracy metrics

It is important to emphasise that even though accuracy metrics are the most straightforward and quite applicable measurement of quality, there are many important aspects beyond recommendation accuracy which relate to both business goals and user experiments [132]. Accuracy metrics, when used alone, may lead to a design and implementation of recommender systems that do not provide to the users an effective and satisfying experience [7]. For instance, a recommender system that only recommends popular items. This system often achieves a very high accuracy. However, it is not a very useful recommender since it does not simplify the exploration of the catalogue, by assisting the users to find new items. For this reason, complementary metrics such as beyond-accuracy metrics have been proposed to help understand the actual quality of the recommender systems. They are measurements that evaluate if the recommender system is only recommending relevant items that are highly popular, or if it can leverage the whole catalogue while diversifying its recommendations for different users. Some of the most common and widely used beyond-accuracy metrics are diversity, novelty, coverage, and business indicators, such as the total generated revenue [132].

2.5.7 Existing video recommendation approaches that exploit video content

The inspiration behind the video recommendation system stems from the recent gain in popularity of video streaming services [3]. Large amounts of video data are uploaded to video sharing sites that rely heavily on the video recommender system to help users discover videos that they would enjoy [2]. For most video streaming services, the computation of video relevance is based on user implicit feedback [3]. For example, search and watch user behaviours. This feedback is used with collaborative filtering to model the user-video preference in order to compute the video relevance. While collaborative filtering is very efficient in the execution of the recommendation task, it suffers from the new item cold-start problem [7, 10]. This problem can be divided into incomplete new item cold-start problem and complete new item cold-start problem [10]. The incomplete new item cold-start problem occurs when old videos have a limited number of records of user-video interaction. Generally, in this scenario, the sparsity of the URM is higher than 85% [10]. The complete new item cold-start problem occurs when a new video is added to the catalogue, and consequently no user-video interaction records are available for it. The sparsity of the URM in this scenario is 100%; otherwise, it is a warm-start scenario. As such, the system cannot make accurate recommendations for users since it has not yet gathered sufficient user-video interactions. In most video streaming services, new videos are continuously added at a very fast rate [3, 23]. Owing to this reason, it is crucial to discover and recommend these new videos to users to increase user satisfaction and discourage them from moving

to the competitor's platform. However, CF models are unable to achieve this [7].

The most straightforward approach is to use CB recommendation methods to compute video relevance directly from video content to address this problem [5, 12, 24, 27, 133–138]. It is possible to obtain almost all the information about a video from its content, from which a video relevance table can be generated, even without user feedback [6, 139]. Classic CB video recommendation systems typically exploit high-level features extracted from the post-release textual metadata of the videos [5,24,134,135]. However, when this data is scarce, there is a significant performance drop in these systems. Owing to this problem, video recommendation systems that do not resort to the metadata provided in textual form, but instead exploit non-textual content features have been proposed to counteract the new item cold-start problem [12, 27, 136–138]. They use manually engineered visual low-level features [136], deep learning visual features [12, 137, 138], and deep learning visual and audio features [27] extracted directly from the media contained in the videos. In this regard, CB recommendation systems and hybrid recommendation systems, which use these features, are presented in detail below.

A video recommender system named *Video Reach* [140–142] is one of the earliest approaches that leverage the rich multimodal information from the video content, namely audio, visual, and textual modalities for video recommendation. It is based on multimodal content relevance and user feedback. It calculates the similarity between two videos given their textual, visual, and aural features along with weights to balance the contribution of each modality to the relevance. The textual features used by the system are query, keywords and metadata as well as text obtained using automated speech recognition (ASR) and OCR. The visual features used are normalised colour histogram (64 features), motion intensity (1 feature), shot frequency (1 feature), and automatically recognised video concepts (36 features). Lastly, the audio features used are the average and standard deviation of aural tempos extracted for the entire video. It is found that video and audio content analyses improve video recommendations. This system, however, has some drawbacks. Firstly, videos with low textual similarity are filtered out to ensure that only valid videos exist before visual similarities are determined. Secondly, there is a limit to the number and form of visual and audio features. Only colour histogram is used and this feature has a range of possible drawbacks, as stated in Section 2.2. Motion and audio information are represented by only one and two features, respectively. Thirdly, weights are chosen to balance the contribution of each modality to the relevance calculation; however, textual features are given a much higher weight without investigating other arrangements. Because of these limitations, the generalisability of the findings using this system is not clear.

Other proposals use a content-based video recommendation system that utilises an ItemKNN-CBF algorithm with low-level visual features. These visual features are extracted from colour, motion, and video's shot to predict and generate top-*N* recommendations [26]. They have been proven to influence the audience preference presented by like or dislike of videos [101]. The proposed system first extracts the videos' keyframes. Secondly, the average length of the shots, the mean lightening key, colour variance across all keyframes, as well as the motion standard deviation and its average estimated across all frames are determined and used as the video content features. Thirdly, the ItemKNN-CBF algorithm is used where the cosine similarity between the videos is calculated. This algorithm recommends a video to a user if it is similar to what the user liked before [26]. Finally, the performance of the system is assessed using a small-scale dataset of complete movies and movie trailers obtained from Youtube, by calculating the precision and recall metrics through 5-fold cross-validation. A significantly higher accuracy is obtained compared to an ItemKNN-CBF recommendation system, which uses high-level semantic features based on metadata, such as genre. Therefore, the main finding is that low-level visual features can represent the stylistic characteristics of a movie, which are applied by the director to invoke specific emotions in the user. These effects are likely to affect the experience, opinions, and feelings of the users about the movie. Another very important finding is that in the absence of full-length movies to extract low-level features for recommendation task, features extracted from movie trailers may be utilised as a substitute since they are highly correlated with their corresponding full-length movies.

As pointed out in Section 2.2.2, deep learning approaches are being used to automate the learning of features from the original data, aiming to capture a more accurate representation. Inspired by this finding, a content-based movie recommendation system called *DeepRecVi* is proposed [139]. This system uses visual features extracted by a pre-trained CNN to provide relevant recommendations. These features are extracted from keyframes of movie trailers and represent objects and environments since the model used is trained using the concatenation of ImageNet and Places-365 datasets. The system requires a single feature vector that represents the whole movie trailer. This is obtained by performing aggregation of the keyframe features using a scene categorisation approach. This scene categorisation approach combines the vectors computed by assigning them to scene categories in an unsupervised manner using k-means clustering. The final video-level representation is used as input to the profile learner component of the system, which builds the user profile. Next, the user profile is sent to the filtering component where it is exploited to generate top-*N* recommendation lists. The system is evaluated using 9408 movie trailers from the Labelled Movie Trailer Dataset. The MAE, precision,

and recall metrics show that the deep learning features extracted using a CNN model outperform the low-level visual features [26]. This implies that the semantic representation by the former is more robust compared to the latter. However, the performance of this system could be further improved by extracting audio features to obtain a video representation that is perceptually complete. This, in turn, could lead to the best overall performance.

Subsequently, given the outstanding performance obtained by the deep learning features in video recommendation tasks, more video recommendation systems that use deep learning features have been proposed [12, 143]. These systems provide recommendations based solely on the implicit visual content in the videos. A content-based video recommendation system that uses pre-extracted visual features of videos and three fused long short-term memory (fusedLSTM) networks is developed to tackle the new item cold-start problem [143]. The three networks are combined to form a triplet network [144]. The pre-extracted visual features of the videos are frame-level and video-level features generated by a pre-trained inception-V3 network and a 3D CNN, respectively. These features are combined and passed as input to a fully connected layer that gives as output fused embeddings. These fused embeddings are passed to a similarity Kernel and triplet loss function. Three similarity kernels are used in the experiments, namely the radial basis function, the shifted cosine function, and the softmax function. The system is evaluated using the dataset provided by the content based video relevance prediction (CBVRP) challenge, which contains pre-extracted features from 7536 TV shows as well as 10826 movies trailers. Based on the results obtained, the fusedLSTM method with the softmax similarity kernel outperformed the other kernels. It is clear that this model learns the video content well by capturing the temporal relationships in the frames, which contribute to the video relevance prediction. Three models, namely a random forest regression on video pairs model, a deep learning based regression model, and a neural network with deep linear discriminant analysis (LDA) model, are proposed to solve this same problem [12]. Using simple distance metrics, the random forest regression on video pairs model calculates the similarity between two given video-level feature vectors. As soon as the best features are chosen, the probability of dissimilarity is found. The deep learning based regression model consists of two networks that work in parallel where the first network has time distributed dense layers and LSTM layers while the second network has only dense layers. Frame-level features and video-level features are used as input to the first and second networks, respectively. These two networks are followed by a final layer that concatenates their outputs and generates the probability of the similarity between two videos using a fully connected layer. Lastly, the neural network with the DeepLDA model consists of a deep neural network that has three fully connected layers and uses a

modified version of LDA as a loss function. LDA is an algorithm that reduces the dimension of the features by finding linear combination of these features that best explain the classes that they represent. This model uses the video-level features as input. Similar to the fusedLSTM model, the three models were tested on the dataset provided by the Hulu content based video relevance prediction (CBVRP) challenge. It was found that the DeepLDA model outperformed the other two models as well as the fusedLSTM model.

Although the performance of the video recommendation systems [12, 139, 143] using deep learning visual features is promising, their outcome could be further improved by using content-based audio features. Unlike the systems proposed in literature [12, 26, 139–143], which rely heavily on textual features or only on visual features, a CB video recommendation model is proposed in [27], which combines content-based visual and audio features from raw video and raw audio of the video, respectively. These features are obtained using the Inception-v3 network trained on ImageNet dataset and Visual Geometry Group (VGG)-inspired acoustic model with a modified version of residual neural networks (ResNet)-50, respectively. Since these two networks were not trained for recommendation task, the system uses semantics extracted from watch patterns and a feedforward network to fine-tune for recommendations of the extracted visual and audio features. As mentioned, to further improve the system performance, the information from two sets of features that are obtained from different methods should be combined using a fusion strategy of multiple features [140–142]. This work proposes two different network architectures where one uses early fusion, and the other uses late fusion. The early fusion strategy used is the combination of the two input features, namely visual and audio features just before fine-tuning them for recommendations. The late fusion strategy used is the combination of the two input features by element-wise multiplication after fine-tuning them separately for recommendation. It is found that the visual and audio features are better at representing videos compared to only using visual features. This outcome is expected since fusing these two modalities made the video representation perceptually complete. The reported results also show that the late fusion outperforms early fusion. Lastly, the visual and audio features need to be used with metadata, or as side information to a collaborative filtering system, in order to outperform a normal CF method with sufficient *user-item* interactions (ratings).

It is evident from the results of the previous study [12, 26, 139–143], and other literature studies, that the use of audio and visual features, extracted from the video content, provides the best overall performance [27, 136]. However, due to the nature of CB video recommendation models, the quality

of recommendation generated by them is limited. A CB video recommendation model ignores useful CF information necessary to exploit quality judgements of other users and to help users discover new interests [5, 135]. These shortcomings lead to over-specialisation and lower accuracy than CF methods in both incomplete item cold-start and item warm-start scenarios [7, 145].

A few recent works [7, 14, 146, 147] propose a hybrid video recommendation system to address the limitations mentioned above that are present in CB video recommendation systems, to provide more effective recommendations of videos that may be relevant to users. Hybrid systems incorporate, as stated in Section 2.5.3, both CF information and features extracted from the post-release textual metadata of the videos or features extracted from the media embedded in the videos, or both. One of the earliest works that proposed a hybrid recommendation system, which uses non-textual video content features uses a factorisation machine (FM) algorithm [146]. FM is a combination of support vector machine (SVM) with factorisation models [146]. As shown in Figure 2.9, the system includes low-level visual features as complementary video content information to the model to improve its recommendation accuracy.

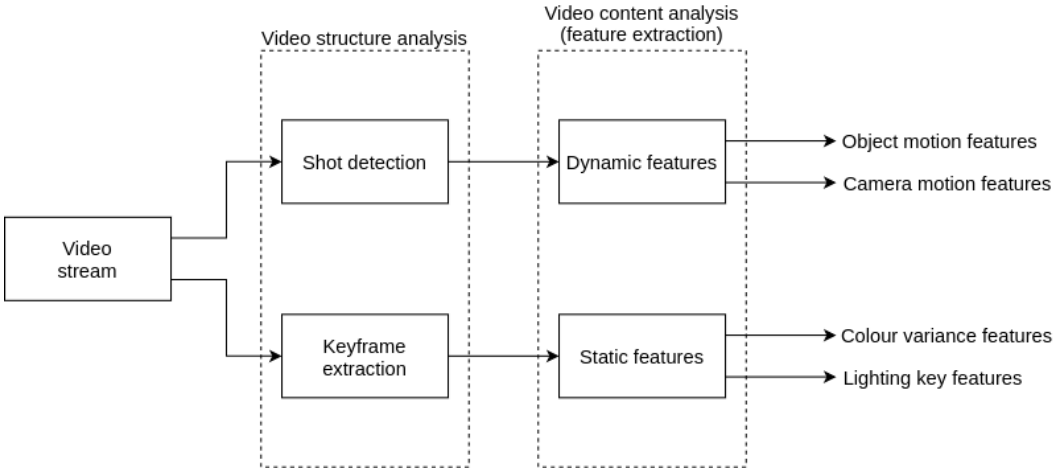


Figure 2.9. Generic framework of video analysis system adapted from [146] where object and camera motion features are extracted from a shot detected in the video. In addition, colour, as well as lighting features, are extracted from keyframes extracted from the video.

The features shown in Figure 2.9 are used as input to the FM model to generate recommendations. This model is assessed by using a dataset of 13 million ratings given by 182 000 users to 13 373 movie trailers that where downloaded from Youtube. The proposed model achieves an outstanding recommendation accuracy compared to the same algorithm using genre as a feature descriptor for the videos. Another hybrid approach proposed in literature extracts low-level visual features from

the video keyframes using a pre-trained deep neural network (DNN) and MPEG-7 descriptor [147]. The keyframes are obtained by first segmenting the video into shots using histogram similarity with a threshold set to 0.75, and then selecting a representative frame from each shot, which is typically the middle frame. The MPEG-7 descriptor captures the stylistic properties of a video while the pre-trained deep leaning network captures the objects in the video. MPEG-7 features are extracted from each keyframe. These features include scalable colour descriptor, colour structure descriptor, colour layout descriptor, edge histogram descriptor, homogeneous texture descriptor, and deep-learning features. The latter includes activation values of the inner neurons of the pre-trained DNN. A single feature vector that represents the entire video is obtained by using aggregation functions such as the minimum, mean, median, and maximum of the MPEG-7 features as well as the deep-learning features. A low-level correlation between the two sets of MPEG-7 features, and deep-learning features is exploited using canonical correlation analysis (CCA). As explained in Section 2.4.1, CCA is a fusion method that analyses and combines information of two sets of different features, which are extracted using different methods in order to create a fixed-length descriptor that contains the maximised pairwise correlation between them [146]. Top-n recommendation lists are generated by using the fixed-length descriptor from the CCA as a piece of side information in a collective sparse linear method (cSLIM). This hybrid method is a feature-enhanced CF, which assumes that a correlation exists between the user preferences and the similarity between two videos, represented by the video-level descriptors. The performance of the implemented system, as discussed in [146], was evaluated using the MovieLens 20 Million ratings (20M) dataset. It computed the top-N recommendation lists precision, recall, F1, and MAP metrics. The lengths of the recommendation lists investigated are $n = 1, 10$ and 20. The proposed system is compared to another hybrid system that uses genre and tags as side information. From the reported results, it can be seen that MPEG-7 features provides better video recommendations compared to the deep learning features, it also outperforms the genre and tag features. The best overall result is provided by the combination of the deep learning features and MPEG-7. From the study in [147], it can be concluded that the stylistics properties of a video, represented by the MPEG-7 features, are more powerful than the object semantic properties, represented by the deep learning features, and that fusion of these two features leads the best hybrid recommendations.

Even though these systems [146, 147] outperform a CB video recommendation model in incomplete item cold-start and warm-start scenarios, they are not applicable in a complete new item cold-start scenario [7]. In order to address this limitation, a hybrid recommendation system named collaborative-filtering-enriched content-based filtering (CFeCBF) has recently been proposed [7]. This system is not

limited to only visual features, but exploits audio features as well as the collaborative information. The system uses aesthetic-visual features (AVFs), block-level audio features, deep learning visual features, and i-vectors audio features as side information. AVFs are features associated with the aesthetics and style of a video. In the proposed work, the colour-related, texture-related, and object-related aesthetic-visual feature types are used. Block-level audio features capture the spectral, harmonic, rhythmic, and tonal aspects of audio. They are extracted from larger audio segments. Deep learning visual features represent the object's semantic properties contained in a video. Unlike the approach in [147] which uses features extracted from a pre-trained DNN, CFeCBF system extract these features from a pre-trained CNN. I-vector features represent the amount that a short audio segment is shifted from the average video clip in the acoustic feature space. Similar to the proposed cSLIM system [147] that combines various visual features, the CCA is used for the fusion of different features to improve the recommendations. The performance of the CFeCBF system is evaluated by executing unimodal and multimodal experiments using the MovieLens-20M dataset where, for the multimodal experiments, two features are fused using the CCA. Two categories metrics, namely accuracy metrics and beyond-accuracy metrics are calculated. The beyond-accuracy metrics used assess the diversification and catalogue coverage capabilities of the recommender system. As pointed out in Section 2.5.6.2, the beyond accuracy metrics are equally important compared to accuracy metrics, due to the fact that a recommender system may provide recommendation with high relevance, but may not facilitate exploration of the catalogue. The results of the experiments show that in the visual category, deep learning visual features outperform AVFs, while in the audio category block-level audio features outperform i-vector audio features. However, when visual and audio best features are combined, they are outperformed by the AVFs plus deep learning visual features combination with respect to accuracy metrics. In contrast, the beyond-accuracy metrics results show that the deep learning visual features plus block-level audio features exhibit the highest diversity. By carefully checking the reported unimodal and multimodal pure CB results, it is worth noting that in terms of accuracy metrics, surprisingly the implemented CCA fusion method does not lead to better results compared to the best unimodal content feature result. This outcome can be attributed to the diverse performance results among the different content features. Nevertheless, the CFeCBF model solves the new item cold-start problem better than a pure CB model by exploiting the collaborative information to optimise the content-based side of the algorithm. However, it has a few limitations. Firstly, it is dependent on the quality and noisiness of the item's content descriptors because it uses them to learn feature weights. Secondly, it is not applicable to any scenario since it approximates a collaborative model and collaborative models do not perform well when there are too few user-item interactions (incomplete

item cold-start scenario). Thirdly, it is limited to the fusion of only two features and treats all features the same (same weight). Lastly, it requires a gradual switch between the CFeCBF and CF model when recommending new items since CF algorithms are able to outperform the proposed model as soon as a few interactions for these items are available.

Finally, it can be seen from the study that proposed the CFeCBF model that it is challenging to achieve superior performance in both item warm-start and cold-start scenarios [7, 27]. A linear model named collaborative embedding regression (CER) is able to resolve this problem [14]. This model is a weighted matrix factorisation (WMF)-based hybrid recommendation approach. It uses the implicit feedback of users, along with high-level or low-level video content features extracted from videos, in order to learn the user preferences and generate top-N recommendation lists. The high-level features are captured from the plot and metadata of the videos. The low-level content features, namely MFCC, SIFT, iDT and deep learning visual features are extracted from the audio and visual streams present in the video. Similar to all recommender systems mentioned in this literature review, this system requires a video-level descriptor that represents the whole video. The model uses this descriptor as side information. The FV encoding method is used to perform the feature aggregation task. The performance of the proposed model is compared with existing state-of-the-art recommender models, namely WMF, collaborative topic regression (CTR), deepMusic (DPM), collaborative deep learning (CDL), Bayesian personalised ranking (BPR), and visual Bayesian personalised ranking (VBPR) using a processed MovieLens 10 million ratings dataset [14]. The CTR, DPM and CDL are weighted matrix factorisation based recommender models while VBPR is a Bayesian personalised ranking based recommender models [14]. The proposed fusion method is compared with different early and late fusion methods, namely early fusion by content vector concatenation, early fusion by latent content vector stacking, accuracy fusion, average fusion, ranking SVM, and ranking BPR. The proposed model and fusion method is better than many hybrid recommendation systems and fusion techniques used for comparison, according to the reported findings, because it achieves high recommendation accuracy when the videos have or have not been rated by the users (item warm-start and cold-start scenarios). It is also noted that there is not a major difference between the different non-textual video features in the item warm-start scenario. On the other hand, the deep learning visual features outperform the other low-level video content features in the item cold-start scenario. The priority-aware late fusion of all video content features delivers the best overall result.

While recent CB recommender models have begun using deep learning features to capture the visual

and aural video information perceived by the users, it is clear that hybrid recommender models still rely heavily on hand-crafted features. The motivation behind the choice of hand-crafted features in literature [21] is that, recommendations generated by hand-crafted features are easier to explain, given their semantically meaningfulness. However, as stated in Section 2.2.2, when dealing with huge datasets, hand-crafted features become unfeasible and this is the case for many real-world datasets, such as the one used in the Hulu CBVRP challenge. There are also signs that deep learning features, extracted from video frames, represents semantically interpretable information. This outcome is supported by the advancement of visual explanations from deep learning networks. This finding is shown in Figures 2.10 and 2.11.

Figure 2.10. Discriminative regions specific to each object class and scene class. These discriminative regions are generated given an input image A to an object-centric CNN model and a scene-centric CNN model. These models focus on different aspects present in the image, namely people and shelters, as shown in image B and image C, respectively. (From [85], © 2019 IEEE)

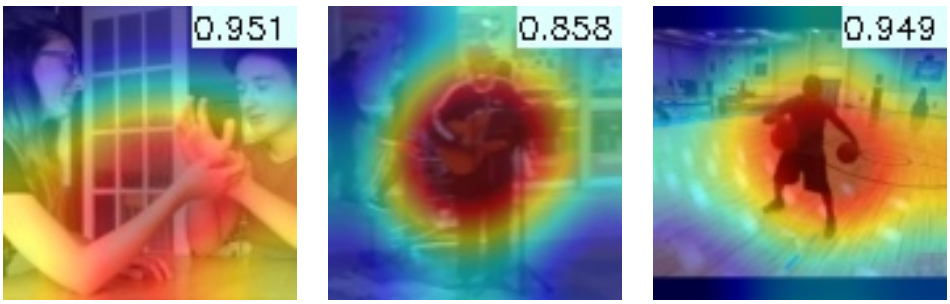


Figure 2.11. Class discriminative regions specific to each action that is taking place in the images. These discriminative regions are generated by an action-centric CNN model. The numbers at the top of each image indicate the scores obtained by the model, with respect to the ground truth action class. (From [66], © 2019 IEEE)

As can be seen from the figures above, the embeddings extracted from the last convolution layer of each CNN model represent the class discriminative regions very well. These regions vary from model

to model since each CNN model focuses on different elements presented in the image in order to perform classification. In Figure 2.10, the discriminative regions shown in image B are generated by an object-centric CNN model. It can be observed that this model focuses on the people present in the image. On the other hand, the discriminative regions shown in image C of Figure 2.10, are generated by a scene-centric CNN model. It is clear that, in contrast to the object-centric CNN model, this model focuses on the image background instead. Additionally, Figure 2.11 shows the discriminative regions generated by an action-centric CNN model. It can be seen that this model focuses on the action in the image.

In summary, hybrid recommender systems that leverage CF information along with visual and audio features have a higher recommendation quality in comparison to recommendations based on only audio and visual features in a complete item cold-start scenario [7, 14]. The quality of these recommendations is also better than the quality of recommendations generated by CF methods based only on past user interactions with the items in incomplete item cold-start and warm-start scenarios [14]. However, these approaches limit themselves while solving the new item cold-start problem, since deep learning action features and the correlation among object, scene, and audio deep learning features are not explored. Thus there is a need to investigate and improve these hybrid video recommendation systems [7, 14] such that they utilise all these features, in a comprehensive manner in order to fully exploit the complementary information from them and enrich the recommendations.

2.6 CHAPTER SUMMARY

A detailed literature review of the video recommendation approaches that exploit video content to achieve better overall performance was discussed in this chapter. This review begins with feature extraction techniques that are commonly used for video content analyses in the video domain. A video is typically represented by hand-crafted low-level features or deep-learning features that need to be aggregated overtime to form a video-level descriptor. It can be seen from various studies that deep-learning features outperform hand-crafted features in a number of video content analysis tasks, such as action recognition. To further enhance the system's performance, different studies in the video domain have also highlighted the importance of feature fusion. Two fusion techniques are widely used, namely early fusion and late fusion. Early fusion techniques normally deliver a better performance in comparison to late fusion techniques because of their capability to obtain a truly multimedia feature representation. These techniques are also more feasible in contrast to late fusion techniques since they only require one training phase. Additionally, it is explained how the collaborative filtering technique,

the content-based filtering technique, and the hybrid filtering technique work. The collaborative filtering techniques suffer severely from the item cold-start problem, which is the problem that is tackled in this dissertation. The content-based filtering techniques do not suffer from the item-cold start problem given the way they work, but their performance is extremely low when the items switch from cold (no ratings) to warm. The hybrid filtering techniques overcome the limitations of collaborative filtering and content-based approaches, such as the cold-start and overspecialisation problems. Finally, the video recommendation approaches that exploit the video content are discussed. Several recent studies in the video recommendation domain show that audio and visual based features are preferred over the text-based features in content-based video recommendation models. Visual based features are more informative than a set of genre data. Therefore, models trained using visual features perform better than models trained using only genre data. Similar to video classification tasks, the fusion of visual and audio features delivers better performance than only using visual features to train a model. However, using only these two features is not sufficient to outperform collaborative filtering models when the videos do not have a small number of ratings. In order to overcome this shortcoming, hybrid approaches which use visual and audio features in conjunction with collaborative information are proposed in literature. These approaches show that when items have a good number of ratings (item warm-start), the collaborative information helps the model to obtain comparable results to collaborative filtering approaches. In addition, this information also allows hybrid models to outperform a content-based model when the items do not have any ratings (item cold-start).

CHAPTER 3 METHODS

3.1 CHAPTER OVERVIEW

In this chapter, the methods used to obtain the results to answer the research questions of this dissertation, are described and discussed in detail. This required an investigation of the video recommendation models, as well as the feature extraction, and the feature aggregation methods used in the recommendation framework. It also involves the experimental setup used to investigate, evaluate, and compare different features and video recommendation models in the item warm-start and cold-start scenarios. In particular, the preparation of the data, the evaluation method, the evaluation metrics, and the experimental framework are discussed. The fundamental assumption in this work is that visual and aural features extracted from movie trailers are highly correlated with visual and aural features extracted from full-length movies, which can therefore be used as an alternative [7,26]. This enables the analyses of visual and audio content in videos to be computationally efficient. In general, movie trailers are short previews of movies, created in a way to garner interest from the audience. Furthermore, feature fusion methods to combine different features are explored and addressed. This includes additional investigation of different early fusion methods in order to fully exploit the complementary information from the various video features to enrich the recommendations.

3.2 PROBLEM DEFINITION

As described in section 1.2, the main objective of this research work is to generate top-N video recommendations to users in the new item cold-start scenario given the users' implicit feedback and the video features. This task can be mathematically described as

$$f_u : U \times I \longrightarrow relevanceScore, \tag{3.1}$$

where U is a set of all users, I is a set of all items and f_u is an utility function that measures the relevance score of a set of items to a set of users. As has been already noted in section 2.5, there are different techniques that can be used to estimate this utility function when using the user feedback data and the features of the items. In this work, the new item cold-start problem is solved with the hybrid approach.

3.3 RECOMMENDATION FRAMEWORK

3.3.1 Feature extraction

A video recommendation system generates recommendations given two scenarios, namely the item warm-start and item cold-start scenarios [14]. Given an item warm-start scenario, the video recommendation system can recommend the top-N videos to a target user using solely the users' implicit feedback (ratings). However, to recommend the top-N videos to a target user given an item cold-start scenario, video content features should be extracted and used to improve recommendations. Users are likely to prefer videos with similar visual and aural modalities to those that they have already liked. For this reason, the video features must be perceptually complete and semantically meaningful to produce a video representation that is well aligned with the way the user perceives it [148].

Perceptually complete means that at least one feature vector should be defined for each video component from which a user experiences the video, namely the visual and sound components [148]. On the other hand, being semantically meaningful means that the feature representations should provide meaningful variance for the different videos in which it is possible to infer or assign meaning (semantics) that are relevant to the target domain. With this in mind, object, scene, action and audio information contained in the videos are represented by various deep learning features. Even though an in-depth explication of CNNs are out of the scope of this work, a brief explanation of how CNNs represent the video frames is provided in this section. This is important to better understand the video features generated by the pre-trained CNN models chosen in this research work.

3.3.1.1 Deep learning features

The deep learning features used in this work are CNN embeddings generated by intermediate layers of CNN models. These embeddings are chosen because they are more generalised and robust to noise as opposed to features extracted from the final output layer [85, 117]. The visual-appearance information contained in videos is represented by object-centric and scene-centric CNN embeddings. The motion information is represented by action-centric CNN embeddings. Lastly, the audio information is

represented by audio CNN embeddings. The feature extraction processes for each visual-appearance, audio, and action deep learning features are discussed below.

A. **Object features**

The object information from videos is captured using the *Obj(IN)* model that is pre-trained on the ImageNet dataset [83] for the task of object classification. This model is a ResNet-152 network [149] that receives as input images of size 224×224 pixels. The videos are decoded at 1 fps, and each video frame is resized to 224×224 pixels. The object-centric embeddings are extracted from the last convolutional layer. This layer has 2048 dimensions. Thus, each video frame is represented by a tensor with 2048 dimensions. The object-centric embeddings of the last convolutional layer is a tensor. Global spatial average pooling is applied to transform it into a fixed dimensional vector. Each vector contains the video content object features, which are present in a frame.

B. **Scene features**

The scene where an action is taking place may provide relevant information that supports actions with object interactions. In this work, the scene information from video frames is captured using a DenseNet-161 model [150] pre-trained on Places365 dataset [84]. This model is a 2D CNN network with 161 layers. It consists of an input layer of size 224×224. Thus, each frame is first resized to this scale before it is passed to the model. Scene-centric embeddings of 2208 dimensions are extracted from the last global average pooling layer. Similar to the object-centric embeddings, the scene-centric embeddings are extracted from videos decoded at 1fps, and global spatial average pooling is used to transform the tensor into a vector. This vector contains video content scene features, which represents related contextual information about a scene in a frame.

C. **Action features**

The action features are extracted from videos with pre-trained 3D CNN models. These features capture the motion information in a video [15]. In particular, each video is decoded at 24 fps and the visual stream is used as input to the *Action(IG), Action(KN), Action(UCF),* and *Action(HMDB)* models. The *Action(IG)* model is a R(2+1)D-34 32-frames model [93] that is pre-trained on the IG-65m dataset [151], which includes 359 human action classes that are identical to the action labels of the Kinetics dataset [15]. This model consists of 34 layers, where 33 layers are convolutional layers and the final layer is a fully-connected layer with softmax (i.e. the classification layer). The input layer receives clips consisting of 32 consecutive RGB video frames with size 112×112 pixels. Thus, if the clips obtained from the video are composed of

video frames with different resolution, these frames need to be resized. The output size of the last convolutional layer is 512. The embeddings generated by this layer is passed to a global spatio-temporal average pooling layer and fed into a fully-connected layer with classification layer that predicts the 359 action classes. The action-centric embeddings, generated by the global spatio-temporal average pooling layer, are extracted and used as the video content action features. They form a 512-dimensional descriptor for each clip of 32 consecutive 112×112 pixel frames. The size of the features extracted for videos recorded at 24 fps, is $Tv \times 512$, where $Tv = 0.75 \frac{features}{second} \times duration(s)$. For example, a 120-second video has 90 R(2+1)D-34 global average pooled features with 512 dimensions.

The *Action(KN)* model is a network similar to the *Action(IG)* model, with the only difference being the final layer size of 400 rather than 359. This dimension corresponds to 400 human action classes of the kinetics dataset, since the network is pre-trained on the IG-65m dataset and fine-tuned on the kinetics dataset. Similar to the *Action(IG)* model, the action-centric embeddings generated by the Action(KN) model are extracted from the global spatio-temporal average pooling layer.

Lastly, the *Action(UCF)* and *Action(HMDB)* models are both a ResNeXT-101 64-frames network [152] pre-trained on the Kinetics dataset and fine-tuned on the UCF-101 and the HMDB-51 datasets, respectively. The network consists of 101 layers where the last convolutional layer is followed by a global average pooling layer and a fully-connected layer with a classification layer. The classification layer of the *Action(UCF)* model has 101 dimensions that correspond to the total number of action classes in the UCF-101 dataset. On the other hand, the *Action(HMDB)* model has a classification layer with 51 dimensions. The size of the input layer is 3 *channels* $\times$ 64 *frames* $\times$ 112 *pixels* $\times$ 112 *pixels*. Therefore, a 64-frame clip needs to be resized if it has a different resolution. For these two models, video frames are decoded at 24fps and processed in clips of 64 consecutive frames. Hence, every single clip spans approximately 2.67 seconds of the video. The frames are first resized to 112×112 pixels, before passing to the models. The action-centric embeddings generated by the global average pooling layer with 2048 dimensions, before the classification layer, are extracted and taken as the action features. In this way, the features are extracted at 0.375 features per second. For example, a 120-second video has 45 ResNeXt-101 global average pooled features.

D. **Audio features**

Audio features are extracted from audio frames with a *soundNet* model and a *VGGish* model [97]. The *soundNet* model is a 1D CNN network that consists of 8 convolutional layers and 3 max-pooling layers pre-trained with the supervision of object-centric and scene-centric VGG networks [96]. The object-centric and scene-centric VGG networks are 2D CNN models pre-trained on imageNet and Places datasets, respectively. These models are used to teach the *soundNet* model to recognise concepts given sound. The architecture of the *soundNet* model is presented in Figure 2.4. In contrast to the visual models, the output layer of this network is also a convolutional layer. The input layer extracts sound features from raw audio waveform in the range [-256, 256] and a sampling rate of 22 kHz. Therefore, requiring the audio stream to be re-scaled and re-sampled if necessary. In this research, the audio embeddings generated by the fifth pooling layer are chosen to represent the audio information from the videos. The dimension of the embeddings is 256. These embeddings are chosen because of their state-of-the-art results in action recognition and acoustic classification tasks.

The VGGish model is a 2D CNN network pre-trained on the YouTube-8m dataset for audio classification [97]. This model is a modified VGG architecture. In order to extract sound features using this model, the audio stream of each video has to be pre-processed. The raw audio waveform is first downsampled to a 16 kHz mono signal with 16-bit resolution and re-scaled to the range [-1.0, 1.0]. Next, the audio signal is divided into a sequence of successive non-overlapping 0.96s audio segments of the original video, and subsequently converted from time domain to frequency domain. The conversion is performed with a short-time Fourier transform (STFT). This operation is computed using a periodic Hann window that receives as input frames with size of 25 ms and stride of 10 ms. The resulting spectrogram is mapped to 64 log Mel-spectrogram bins which in turn gives patches of 96 *audio-frames* $\times$ 64 bins. These log Mel-spectrogram patches form the input to the VGGish model that maps them to 128-dimensional audio embeddings. As a result, each VGGish feature vector represents approximately 0.96×24 frames of a 24 fps video. For this reason, the size of the audio-level features extracted from the audio of a video is $Ta \times 128$ VGGish features, where $Ta = \frac{duration(s)}{0.96}$.

3.3.1.2 Textual features

Aside from the video features extracted from the video content, textual metadata features provide a good representation of the videos. Textual feature modality is the most used video representation in

traditional CB or hybrid approaches for video recommendation. Although the main objective of this research work is to investigate the effect of various visual and audio stimuli on user preferences, it is worth exploiting textual features as complementary information of video description.

A set of genres of each motion picture are used as the only type of textual feature in this work. The motivation behind this choice is that genre metadata is highly available in the domain and represent relevant elements in motion picture [7]. In addition, taking into account that genres are high-level semantics attributes of movies, when fused with non-textual content features they will probably remove ambiguity which in turn should lead to an improvement in performance.

Given the genres provided in the meta-information of videos, the genre feature vector is encoded to an D-dimensional binary vector, where D is the total number of unique genres. A bit value of 1 in the i^{th} column of the vector indicates that the corresponding genre describes the video, whereas a bit value of 0 indicates that the corresponding genre does not apply to the video.

The genre feature vector used in this work represent 19 genre labels from the metadata of the motion picture, namely *adventure, animation, children, comedy, fantasy, romance, drama, action, crime, thriller, horror, sci-fi, mystery, IMAX, documentary, war, film-Noir, musical*, and *western*. Thus, the dimensionality of the genre feature vector for each video is 19 where each feature represents one of the 19 annotated genres.

3.3.2 Feature aggregation

Six statistical feature aggregation methods are investigated, namely maximum, mean, median, variance, median absolute deviation and interquartile range. These types of feature aggregation methods are chosen due to their simplicity and low memory consumption. They have been used in various content-based video analysis tasks that utilise deep learning features and obtained significantly better results [7, 30, 104] compared to the state-of-the-art aggregation methods, namely FV and VLAD. One reason behind the poor performance obtained using the state-of-the-art FV and VLAD approaches is that they are limited by the curse of dimensionality [153]. FV and VLAD approaches collect an excessive number of features extracted from video frames that turn out to be unuseful and consequently serve as a degrading factor. As a result, the assumption that the larger the video-level descriptor the better, does not necessarily hold. Recent multimedia recommendation studies are in line with this outcome as well [154].

Another reason worth mentioning, is that deep learning features have different distribution properties and higher discriminative ability in contrast to hand-crafted features [155]. Consequently, FV and VLAD approaches perform significantly worse when aggregating deep learning features [155,156]. On the other hand, the FV and VLAD approaches perform well when aggregating hand-crafted features like SIFT [109,155,157]. The cause of this outcome is that the embedding step of these approaches improve the discriminative ability of the individual features. However, reduction techniques such as principal component analysis (PCA) should be used to reduce the dimensionality of the descriptors.

Nevertheless, statistical feature aggregation methods are the methods chosen in recent datasets which support the exploration of multimedia tasks and provide pre-computed state-of-the-art features that represents video content [107]. In this research work, they are used to create the video-level descriptor vectors by aggregating the deep learning feature vectors presented in Section 3.3.1, namely the object-centric embeddings, scene-centric embeddings, action-centric embeddings, and the audio embeddings. The six statistical feature aggregation methods are discussed in a bit more detail below,

1. **Maximum:** This method finds the maximum value of the individual feature values along each frame-level and video-level feature. It is defined as

$$max(\mathbf{x}) = max_{i=1}^{F}(x_i), \tag{3.2}$$

where $\mathbf{x}$ are the feature vectors and F is the total number of features.

2. **Mean:** This method computes the arithmetic mean of the individual feature values along the frame-level and video-level features. It is defined as

$$mean(\mathbf{x}) = \frac{1}{F}\sum_{i=1}^{F} x_i. \tag{3.3}$$

3. **Median:** The median aggregation method finds the middle feature of the individual feature values along the frame-level and video-level features. It sorts the features in ascending order and chooses the middle value if the total number of features is odd, otherwise it calculates the average of the terms in the middle. This is defined as

$$median = \begin{cases} (\frac{F+1}{2})^{th}term, & \text{if } F \text{ is odd,} \\ \frac{(\frac{F}{2})^{th}term+(\frac{F}{2}+1)^{th}term}{2}, & \text{if } F \text{ is even.} \end{cases} \tag{3.4}$$

4. **Variance:** Variance is a measure of the spread of a distribution. This aggregation method computes the spread around the mean of the individual feature values along the frame-level and

video-level features. It is defined as

$$var(\mathbf{x}) = mean(\mid \mathbf{x} - mean(\mathbf{x}) \mid^2). \tag{3.5}$$

5. **Median Absolute Deviation:** Median absolute deviation (MAD) is a measure of dispersion, and is more robust to outliers compared to the variance measure. This method aggregates the individual feature values along the frame-level and video-level features by computing the median over the absolute deviations from the median. It is defined as

$$MAD = median(\mid \mathbf{x} - median(\mathbf{x}) \mid). \tag{3.6}$$

6. **Interquartile Range:** Interquartile range (IQR) is a robust measure of statistical dispersion, which computes the difference between the third quartile and first quartile. This aggregation method finds the first and third quartile of the individual feature values along the frame-level and video-level features and computes their difference. It is defined as

$$IQR = Q_3 - Q_1, \tag{3.7}$$

where, given an even $2n$ or odd $2n+1$ number of features, the first quartile Q_1 and the third quartile Q_3 is calculated as

$$\begin{aligned} Q_1 &= \text{median of the } n \text{ smallest values,} \\ Q_3 &= \text{median of the } n \text{ largest values.} \end{aligned} \tag{3.8}$$

Additionally, to further enhance the discrimination of the video-level feature vectors, it is used the signed square root (SSR) normalisation followed by PCA on the raw features. SSR is executed in order to weaken the dominant dimensions of each video-level feature vector, so that they do not overshadow the other dimensions during the similarity computations [14]. This normalisation function is defined as

$$SSR(\mathbf{x}) = sign(\mathbf{x}) \times \sqrt{|\mathbf{x}|}, \tag{3.9}$$

where $\mathbf{x}$ is the video-level feature vectors and $sign()$ is the function that captures the sign of each feature. Moreover, PCA is applied to obtain features that are more discriminative and less redundant. Hence, the number of principal components is equal to the original list of features to ensure that no information is lost while covering maximum variance among them. Furthermore, each video-level feature vector is scaled into a unit vector by applying L_2-normalisation (L_2-norm) given by Equation (3.10),

$$L_2\text{-norm}(\mathbf{x}) = \frac{\mathbf{x}}{||\mathbf{x}||_2}, \tag{3.10}$$

where $||\mathbf{x}||_2$ is the Euclidean norm of the video-level descriptors defined as $||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{n}(x_i)^2}$. This is performed to ensure that each feature contributes approximately equally to the final similarity measure [158].

3.3.3 Hybrid recommendation model

The objective of this work is to explore different features that capture the rich and diverse multimodal information present in videos, which may influence the users' preferences to a considerable extent [21], thereby alleviating the new item cold-start problem. In addition, the recommendations in the item warm-start scenario should not be unacceptably low. For this reason, the state-of-the-art CER model [14] is chosen as it is a hybrid recommender model that could lead to the optimal recommendation performance in the item warm-start and cold-start scenarios while using a wide variety of features.

The CER model is a model based on the weighted matrix factorisation method for implicit feedback datasets where a large matrix is decomposed into smaller matrices to reduce the dimensions and learn latent vectors that describe users and items. Latent vectors are composed of latent factors which represent categories that are present in the data in a much lower dimensional space [159]. These vectors are used to predict ratings that are missing in the original URM since every user have videos that they have not watched before. These videos are recommended according to the predicted URM.

The matrix factorisation operation is shown graphically in Figure 3.1 below,

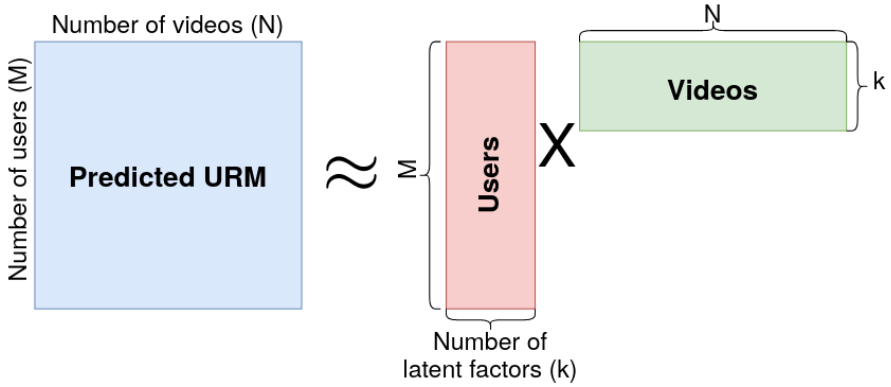


Figure 3.1. Matrix factorisation operation that decomposes the original URM in order to predict if a user is going to like a video never watched before. The predicted URM is the dot product of user and video matrix. Similar to the original URM, each row of the predicted URM represents each user, while each column represents different videos.

As can be seen in Figure 3.1, each user and video is represented by a latent vector that describes their relation towards all the latent factors. These factors describes user tastes with respect to the characteristics of the videos. As a result, the scalar product of the user and video latent vectors represents a sense of how much the user likes the video in terms of the latent factors. These factors are hidden factors that represent the user's preference towards a specific video like genre and actors [159]. The CER model with a single type of video feature follows a rating generation process described [14] below:

1. For each user u, generate a user latent vector $\omega_u \in \Re^{k\times 1}$ and an embedding matrix $E \in \Re^{d\times k}$, where
$$\omega_u \sim \mathcal{N}(0, \lambda_\upsilon^{-1} I_k), \tag{3.11}$$
$$E \sim \mathcal{N}(0, \lambda_e^{-1} I_k). \tag{3.12}$$
2. For each video i, generate a content latent vector $h_i' \in \Re^{k\times 1}$ and a latent video offset vector $\varepsilon_i \in \Re^{k\times 1}$, where
$$h_i' = E^T f_i, \tag{3.13}$$
$$\varepsilon_i \sim \mathcal{N}(0, \lambda_v^{-1} I_k), \tag{3.14}$$
and then set the video latent vector as
$$h_i = h_i' + \varepsilon_i. \tag{3.15}$$
3. For each user-video pair (u,i), generate the rating
$$r_{ui} \sim \mathcal{N}(\omega_u^T h_i, c_{ui}^{-1}), \tag{3.16}$$

where k is the dimension of the latent vector, d is the dimension of the content feature, λ_υ is the hyper-parameter for regularisation of the user latent vector, λ_e is the hyper-parameter for regularisation of the embedding matrix, λ_v is the hyper-parameter for regularisation of the latent video offset vector, I_k is the identity matrix, f_i is the feature vector, and c_{ui} is the confidence parameter for the user-video pair (u,i) defined as

$$c_{ui} = \begin{cases} 1, & \text{if } r_{ui} = 1, \\ 0.01, & \text{if } r_{ui} = 0, \end{cases} \tag{3.17}$$

where the values 1 and 0.01 of the confidence parameter were chosen following the good prediction performance achieved by CTR and CDL models [14]. The regularisation hyper-parameters are the parameters used to avoid over-fitting. Given these hyper-parameters, the CER model is trained by

minimising the negative log-likelihood as follows:

$$\sum_{u=1}^{m}\sum_{i=1}^{n}\frac{c_{ui}}{2}(\omega_u^T h_i - r_{ui})^2 + \frac{\lambda_\upsilon}{2}\sum_{u=1}^{m}\omega_u^T\omega_u + \frac{\lambda_v}{2}\sum_{i=1}^{n}(h_i - E^T f_i)^T(h_i - E^T f_i) + \frac{\lambda_e}{2}||E||_F^2, \tag{3.18}$$

where $||.||_F$ is the Frobenius norm, and then the optimal latent vectors for each user and video as well as the embedding matrix are obtained as follows:

$$\omega_u \leftarrow (HC_uH^T + \lambda_\upsilon I_k)^{-1}HC_uR_u, \tag{3.19}$$

$$h_i \leftarrow (WC_iW^T + \lambda_v I_k)^{-1}(WC_iR_i + \lambda_v E^T f_i), \tag{3.20}$$

$$E \leftarrow (\lambda_v FF^T + \lambda_e I_d)^{-1}(\lambda_v FH^T), \tag{3.21}$$

where H is the video latent matrix, W is the user latent matrix and F is the feature matrix. For each user u, C_u is a diagonal matrix with c_{ui}, $i = 1\ldots,n$ as the diagonal elements and R_u is a vector with r_{ui}, $i = 1\ldots,n$ as its elements. For each video i, C_i is a diagonal matrix with c_{ui}, $u = 1\ldots,m$ as the diagonal elements and R_i is a vector with r_{ui}, $u = 1\ldots,m$ as its elements. After obtaining the optimal latent vectors and the embedding matrix, the rating score for each user-video pair in the item warm-start scenario and the new item cold-start scenario are predicted as:

$$\hat{r}_{ui} = \begin{cases} \omega_u^T(E^T f_i + \varepsilon_i) = \omega_u^T h_i, & \text{item warm-start scenario,} \\ \omega_u^T E^T f_i, & \text{new item cold-start scenario,} \end{cases} \tag{3.22}$$

where $\hat{r}_{ui}$ is the estimated rating score given a user-video (u,i) pair. As mentioned in section 1.1.1, in the new item cold-start scenario items lack any interaction. Therefore, no latent video offset (ε_i) is observed for new items [14].

In this work, the CER model is trained using the optimal hyper-parameter set reported in the original paper [14]. However, the original paper does not mention the number of epochs and the stopping criteria used in the training step of the CER model. Therefore, in this work, the number of epochs is selected using the early stopping technique [126]. This method decreases the risk of over-fitting and also decreases training time.

The main limitation of the CER model is that it does not learn from multiple types of video content features at once. Therefore, there is a need to investigate different fusion methods to leverage the complementary information from the diverse range of features, explored in this work, to further enrich the recommendations. According to existing work found in literature [27, 160], the combination of video-level feature vectors should lead to higher recommendation quality in the new item cold-start scenario in contrast to the use of a single feature modality.

3.3.3.1 Improving CER model using matrix scaling

Recently, successful recommender models named *EIGENREC* [161] and *hybridSVD* [162] have shown significant recommendation quality improvement using a simple scaling trick. These models are matrix factorisation based top-n recommendation algorithms that apply singular value decomposition (SVD). The scaling trick is a matrix scaling technique which regulates how the popularity of items affects the predicted ratings. It is defined in [161] as

$$\widetilde{\mathbf{R}} \triangleq \mathbf{R}\mathbf{D}^{d-1}, \tag{3.23}$$

where $\mathbf{R}$ is the URM, $\mathbf{D} = diag\{||\mathbf{r}_1||, ||\mathbf{r}_2||, \ldots, ||\mathbf{r}_m||\}$ is a diagonal matrix that contains Euclidean norm scaling for a given scaling factor d of the columns $\mathbf{r}_i$ of $\mathbf{R}$ and lastly $\widetilde{\mathbf{R}}$ is the modified URM.

From Equation (3.23), it is clear that when d is 1, the standard model is obtained (URM is not modified). However, when the scaling factor is varied, the sensitivity of the SVD based models to the popularity of the items, is modified. Higher values of the parameter d increase the sensitivity to popular items, while smaller values increase the sensitivity to rare items. This adjustment leads to a new model with a latent space with different internal structure. It has been found that values slightly below 1 yield the best top-N recommendation performance for *EIGENREC* and *hybridSVD* models [161, 162].

Therefore, enlightened by these new findings, an improved CER model is proposed where the matrix scaling technique is used to enhance the performance of the CER model. The matrix scaling technique is used to produce a scaled-CER model in addition to the original non-scaled CER model. This choice is also supported given the fact that a value of 1 for the parameter d leads to the original non-scaled CER model. As a result, this indicates that the original non-scaled CER model implicitly chooses this value that leads to a model, which is extremely sensitive to the prior popularity of the items. Hence, this implicit default choice inevitably hinders the potential of the CER model in both item warm-start and cold-start scenarios. In this research work, using the matrix scaling technique, the confidence parameter c_{ui} for the user-video pair (u,i) of the scaled-CER model is defined as

$$c_{ui} = \begin{cases} 1 \times ||\mathbf{r}_i||^{d-1}, & \text{if } r_{ui} = 1, \\ 0.01, & \text{if } r_{ui} = 0. \end{cases} \tag{3.24}$$

The optimal scaling factor hyper-parameter d is searched by optimising the quality of the scaled-CER in terms of $MAP@5$ on the validation set. This measure is defined in Section 3.4.4.1. The hyper-parameter optimisation is conducted on all cross-validation folds individually with Bayesian

optimisation [7]. Bayesian optimisation selects the next set of hyper-parameters based on the results of the hyper-parameter sets previously evaluated. Once the optimal scaling factor is found on each CV fold, a single optimal scaling factor is selected corresponding to the best average $MAP@5$ result across all folds. Recent studies [7, 163] have shown that Bayesian optimisation is an efficient method for hyper-parameter tuning. The benefits of this method are a reduction in search time and better parameter values compared to a random search or grid search parameter optimisation method.

3.3.4 Enhancing the video recommendation task by combining different modalities from video content

Multimodal fusion can be a very important component in video recommendation systems where improving the overall recommendation quality of the system is considered as one of its most essential aspects. The feature fusion methods commonly used are late fusion and early fusion. Late fusion combines prediction scores of each model in order to obtain a more accurate final set of results. As a result, the main disadvantage of this method is the loss of complementary information represented by different features. This information is important for the final estimation. In addition, late fusion is computationally more expensive given the fact that it requires separate systems and a learning stage for the combination [7].

On the other hand, early fusion obtains a truly multimedia feature representation. It exploits the complementary information about various characteristics of a video at feature level. This in turn improves the discriminability of the video representations. In contrast to the late fusion approach, the early fusion approach only needs a single model and one learning stage. The video information is represented by features from different modalities, namely visual, aural, and textual that are combined into a single feature vector, before being fed to a machine learning algorithm. A recent study in video retrieval tasks shows that early fusion of object, action, face, audio, scene, optical character recognition, and text features allows the system to obtain a better similarity measure and therefore makes it capable of more robust video retrieval [30]. An increase on the overall performance of the system is observed when different features are cumulatively fused [30].

Inspired by the aforementioned findings, various early fusion methods are investigated to enrich the recommendations. This is executed to fully exploit the complementary information from the various feature representations extracted from the video content. Furthermore, this is also investigated to determine whether early fusion would achieve a similar outcome, observed in recent video retrieval

tasks [30, 164]. It is hypothesised that a video recommendation system, which uses videos represented in a shared unified space by a diverse range of deep learning features (visual-appearance, audio and motion), should further improve the quality of recommendations in the new item cold-start scenario.

The early fusion approaches, investigated to combine information from multiple modalities are the concatenation (concat) method, the summation (sum) method, and lastly the maximum (max) method [85]. The concat method is a technique that merges different feature vectors to obtain one large feature vector that represents the final video representation. As this feature vector contains many features, it increases the training time. Formally, for each video feature vector f, if there are L feature vectors of different modalities that are represented with $f_i \in \mathbb{R}^{d_i}$, the concatenation operation is defined as

$$f_f = \{f_1, f_2, \ldots, f_L\}, \tag{3.25}$$

where f_f is the final multimodal video-level representation by fusing the different features that capture visual-appearance, audio and motion information from videos as well as textual information from their metadata. The final size of this representation is the sum of the dimensions of all feature vectors denoted as $d = \sum_{i=1}^{L} d_i$.

The second early fusion method exploited in this work is the sum method. This method adds different feature vectors in order to obtain the final video representation. Given a set of L feature vectors with the same size that represents each video modality separately, their summation is denoted as

$$f_f = \sum_{i=1}^{L} f_i, \tag{3.26}$$

where f_f is the final multimodal representation with size d. As can be seen in Equation (3.26), the sum fusion technique is only defined if all feature vectors have the same size. For cases where a feature vector i of size d_i is greater than $min(d_1, d_2, \ldots, d_i, d_L)$, PCA is applied for feature reduction. The number of features is reduced to the size of the smallest feature vector before performing the fusion operation.

The last fusion technique investigated is the max fusion operation. This fusion method is similar to the sum fusion method in terms of the final multimodal representation size, but differs in the way the feature vectors are combined. The max fusion method selects the highest value of each feature from a

set of L feature vectors with the same size as

$$f_f = max_{i=1}^{d}(f_1^i, f_2^i, \ldots, f_L^i), \tag{3.27}$$

where f_f is the final video representation and d is the feature vector size. Similar to the sum fusion method, PCA is applied as a dimension reduction step for all feature vectors greater than the smallest feature dimension in the set of L feature vectors.

3.4 EXPERIMENTAL SETUP

In order to compare a set of candidate models to choose the best performing one, a methodology of evaluation is necessary. This section firstly describes the dataset utilised in this research work. Secondly, the feature analyses experimental approach performed to check if the features used in this research are indeed semantically meaningful, are presented. Lastly, a description of the evaluation methodology, used in the training and testing of the recommender algorithms, is provided.

3.4.1 Dataset description

A processed MovieLens-10M dataset [14] is used to test the hypotheses and answer the research questions set out in this work. This dataset is a processed version of the publicly available MovieLens-10M dataset. It contains 9,988,676 million binarized ratings given to 10,380 movies by 69,878 users, and 10,380 movie trailers for each movie. The processed dataset also provides five cross-validation folds where each fold is divided into three sets, namely a training set, an item warm-start test set, and a new item cold-start test set. The item warm-start and cold-start test sets correspond to the item warm-start and cold-start scenarios. The item warm-start test set contains items that have some of their ratings in the training set, whereas the new item cold-start test set contains items that do not have any ratings in the training set. Figure 3.2 shows how the warm and cold-start items present on these sets were selected.

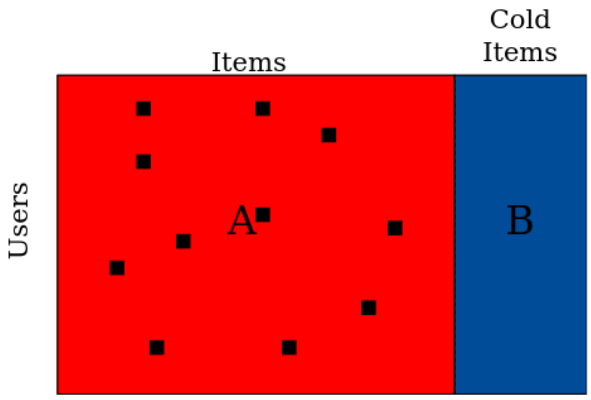


Figure 3.2. URM split for training, and item warm-start and cold-start test sets. The red colour represents the warm items and the blue colour represent the cold items. Subset A contains warm items where some interactions indicated by black squares (cells) represent the interactions used to form the item warm-start test set. The remaining interactions in this subset form the training set. Subset B contains cold items with respect to the training set for which their interactions are used to form the cold-start test set.

The red colour represents the warm items that are in a subset A and the blue colour represents the cold items that are in a subset B. The black cells in subset A represent the interactions that are randomly and uniformly chosen to form the warm-start test set. The remaining interactions in this subset are used for training. Subset B represents the new item cold-start test set that contains items that do not have interactions in the training set and warm-start test set. This set was built by randomly and uniformly item-wise splitting the URM. In addition, the dataset also provides trailers of movies and pre-computed 4,000-dimensional MFCC [14], MoSIFT [69] and iDT [71] feature vectors for each movie trailer. These hand-crafted features are used as the video content feature baselines.

3.4.2 Feature analyses

In this research, video content features are used by the video recommender models to solve the item cold-start problem. For this reason, it is necessary to have an experimental approach that analyses these features. This is performed to check if the features form semantically meaningful representations before being used to generate video recommendations.

As mentioned in Section 3.3.1, the video content features extracted in this work should be perceptually complete and semantically meaningful. This is extremely important because users are stimulated by video content in several ways. Hence, to generate high quality recommendation lists, well-defined discriminative features are required.

The video content feature representations chosen for this work are perceptually complete because they capture the aural and visual aspects of the videos. However, given the semantic gap described in Section 2.2.3, it is challenging to determine if the non-textual video content feature representations are semantically meaningful until using them to perform recommendations. As a result, an experiment is performed to visually and quantitatively analyse the video content representations before they are utilised in the video rating estimation. This is executed to evaluate if the feature representations are semantically meaningful in terms of their genres before generating recommendations.

The visualisation of the video content representations in the feature space is carried out using a uniform manifold approximation and projection (UMAP) algorithm [165]. This algorithm is a non-linear dimensionality reduction technique that operates with similar properties to t-distributed stochastic neighbour embedding (t-SNE) [166], but significantly faster and better at capturing the global structure of the data as well as preserving local neighbour relations. Therefore, like other dimensionality reduction techniques, its objective is to transform the data from higher-dimensional feature space to lower-dimensional feature space while preserving the relative distances between the data points. Another big advantage of this technique is that it can be used for pre-processing like PCA [167] while t-SNE does not have major use outside visualisations.

The quantitative analysis of the video content representations is performed using 16 movie sequels and the Bhattacharyya distance between them [168]. The Bhattacharyya distance is a similarity measure between two distributions [168]. This distance measurement is used to give an insight into the performance of the feature aggregation methods prior to recommendations. The greater the distance between the in-sequel-mean-distance and out-of-sequel-mean-distance, the better. The in-sequel-mean-distance is the average distance between a movie and its sequel movies. The out-of-sequel-mean-distance is the average distance from all movies in a sequel to all movies that are not part of that sequel. This in turn provides a more quantitative feature analysis that supports the visual feature analysis.

The measurements in-sequel-mean-distance and out-of-sequel-mean-distance are defined as follows

$$\mu_{in} = \frac{1}{T} \sum D_{in}, \tag{3.28}$$

where D_{in} is the in-sequel distances and T is the total number of in-sequel movies, and

$$\mu_{out} = \frac{1}{M \times T} \sum_{j}^{M} \sum_{i}^{T} D_{out}, \tag{3.29}$$

where D_{out} is the out-of-sequel distances, and M is the total number of $sequels - 1$. They are computed in the 2-dimensional UMAP space. Since a distance measure is used as a metric, the video-level descriptors have been normalised before applying UMAP and calculating their distances. This is necessary to eliminate the scale difference between the features to have them on comparable scales and thus improve the video similarity task.

The distance measure used is the Euclidean distance. It is chosen because it is simple and consistently found to be efficient in numerous video-based similarity applications [79, 107, 169, 170]. The smaller the distance between videos the higher their similarity. It is calculated as follows

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, \tag{3.30}$$

where $\mathbf{x}$ and $\mathbf{y}$ are video-level descriptor vectors and n is the total number of video-level features. Given the mean distance and standard deviation of the movie trailers in-sequel and out-of-sequel for each feature aggregation method, the Bhattacharyya distance is calculated. This measurement is mathematically described as

$$D_{bh}(D_{in}, D_{out}) = \frac{1}{8}(\mu_{in} - \mu_{out})^2 \big[\frac{\sigma_{in}^2 + \sigma_{out}^2}{2}\big]^{-1} + \frac{1}{2} ln\big(\frac{\big|\frac{(\sigma_{in}^2 + \sigma_{out}^2)}{2}\big|}{\sigma_{in}\sigma_{out}}\big), \tag{3.31}$$

where D_{in} is the in-sequel distances, D_{out} is the out-of-sequel distances, μ and σ denotes their mean and standard deviation, respectively.

Furthermore, the pre-extracted hand-crafted features described in Section 3.3.1 are also visualised in the 2-dimensional UMAP space. These features were combined using the state-of-the-art FV aggregation method due to their distribution properties [14].

3.4.3 Evaluation method

The evaluation process is performed using the dataset described in Section 3.4.1. It is conducted, a five-fold cross-validation (CV) experiment where the training set, and the item warm-start and cold-start test sets occupy 60%, 20%, and 20% respectively. The training set data is only used for training a model. In order to find the optimal hyper-parameter for the model, 10% of the data in the training set is used for validation and the remaining 90% of the data is used solely for training. Validation is performed during the training phase where results from different parameter settings are compared to select the best performing one. When the optimal hyper-parameter has been found, the tuned model is evaluated

using the warm-start and cold-start test sets. The optimal hyper-parameters used to evaluate the CER model are $\lambda_v = 0.1$ for the regularisation of the user latent vector, $\lambda_v = 10$ for the regularisation of the latent video offset vector, $\lambda_e = 1000$ for the regularisation of the embedding matrix and $k = 50$ for the dimension of the latent vectors. The optimal hyper-parameters used to evaluate the scaled-CER are equal to the CER model with the addition of the scaling factor hyper-parameter $d = 0.217$.

3.4.4 Evaluation metrics

The performance of the video recommendation systems is evaluated in the experiments using two metrics, namely accuracy and beyond-accuracy metrics [171]. These metrics are important when evaluating video recommendation systems, since they complement each other. Accuracy metrics evaluate the relevance of the recommendations. However, better user satisfaction beyond relevance is not necessarily achieved with higher accuracy [163]. Beyond-accuracy metrics evaluate the value that recommendations can generate to the user where the desire for variety is not ignored [171].

3.4.4.1 Accuracy metrics

In order to evaluate if the user enjoyed the videos recommended by the video recommendation system, using the various video content features, the MAP and NDCG rank-aware top-N metrics are utilised. These are discussed in more detail below:

1. MAP is the mean of the *average precision* at top-N recommendations $(AP@N)$ over the whole set of users in the test set. It calculates the overall precision of the recommender system by measuring how many of the recommended items are in the set of true relevant items [7]. $AP@N$ is computed by obtaining the arithmetic mean of precision values of the relevant items at their corresponding positions. This metric is chosen because it measures the rate of relevant items in the recommendation list that users may like and therefore consumed, while considering relevant items not in the recommendation list. It is an important metric if it is assumed that many users will not scan the entire recommendation list, but instead they would only look at the top of the recommendation list. This metric is defined as [7]

$$AP_u@N = \frac{1}{min(N,K)} \sum_{i=1}^{N} P@i \cdot rel(i), \tag{3.32}$$

$$MAP = \frac{1}{|U|} \sum_{u\varepsilon|U|} AP_u, \tag{3.33}$$

where N is the length of the recommendation list, K is the total number of relevant items, $P@i$ is the precision at top i recommendations, $rel(i)$ is a binary indicator which signals if the i^{th} recommended item is relevant or not, and $|U|$ is the total number of users in the test set.

2. NDCG is a utility-based ranking measure which considers the order of recommended items in the list [27]. It discounts the positions of the items recommended to a user [7]. This metric is chosen because in a video streaming service, users may be willing to scan all the relevant videos in the recommendation list from the beginning to the end. When the relevant videos appear at a lower ranked position the utility of recommendations is slowly penalised, since videos that are more useful for the user are highly relevant [171]. This metric also shows high robustness to the changes of the MovieLens-10M dataset after pre-processing [172]. Assuming the predicted rating values for the recommendations are sorted in descending order in the recommendation list for user u, DCG_u is defined as [7, 173]

$$DCG_u@N = \sum_{i=1}^{N} \frac{2^{r_{u,i}} - 1}{log_2(i+1)}, \tag{3.34}$$

where $r_{u,i}$ is the true rating of user u for the item ranked at position i. NDCG is the normalised DCG_u which is the ratio of DCG_u to the ideal discounted cumulative gain ($IDCG_u$), which is the value that represents the ideal ranking for user u calculated using the ground-truth ranking instead of the predicted one. This is computed as [7]

$$NDCG_u = \frac{DCG_u}{IDCG_u}, \tag{3.35}$$

where the overall NDCG is obtained by calculating the mean over the whole set of users in the test set [7].

3.4.4.2 Beyond-accuracy metrics

Evaluating the recommendations generated using the various video content features, solely according to accuracy, is not sufficient since the objective of a recommender system is not only restricted to generate relevant recommendation lists to the users. Instead, the features should also cover the whole set of preferences of the users, given the huge body of video data [174]. Beyond-accuracy metrics are used to help assess the quality of the various video content features explored in this work by capturing the coverage and diversity of recommendations. These metrics assess if the systems using these features are able to leverage the whole catalogue instead of only a few highly popular items [7]. It also assesses if the recommendation lists generated by the system for different users are being diversified.

In this work, the video recommendation system using the various video content features is evaluated using the following measures:

1. Intra-list diversity, which is a metric that measures the efficiency of the recommender to generate recommendation lists that cover the entire set of preferences of the users [174]. It is chosen because recommendation lists with similar items may not be of interest to the user [174]. It is calculated by using the cosine similarity between the items recommended based on genre features as [7, 174]

$$IntraL(L) = \frac{\sum_{i \varepsilon L} \sum_{j \varepsilon L \setminus i} (1 - cossim(i, j))}{|L| \cdot (|L| - 1)}, \tag{3.36}$$

$$cossim(i, j) = \frac{\vec{f}_i \cdot \vec{f}_j}{||\vec{f}_i|| \; ||\vec{f}_j||}, \tag{3.37}$$

where $|L|$ is the length of the recommendation list L, $cossim(i, j)$ is the cosine similarity between items i and j, and $\vec{f}_i, \vec{f}_j \in \mathbb{R}^{|F|}$ are the feature vectors of items i and j with $|F|$ the number of features, respectively. Recommendation lists that contain items which are very similar to one another in terms of their genres obtain low values for this metric.

2. Inter-list diversity is a metric that measures the uniqueness of the recommendation lists for the different users [175]. It is chosen because recommendation lists should be personalised according to individual user preferences. This implies that the proposed video recommendation system should not only provide recommendations that have high intra-list diversity but should also provide unique recommendations for all users. If the system generates the same recommendation list to all the users it will exhibit a very low inter-list diversity, implying that it is not able to create personalised recommendation lists to each user [175]. Given the recommendation lists L_u and L_v for two users u and v, this metric is calculated as follows [7]:

$$InterL(L_u, L_v) = 1 - \frac{q(L_u, L_v)}{|L|}, \tag{3.38}$$

where $|L|$ is the length of the recommendation lists and $q(L_u, L_v)$ is the number of items that the two recommendation lists have in common. In order to obtain the overall inter-list diversity, it is necessary to average $InterL(L_u, L_v)$ across all users in the test set, where $u \neq v$.

3. Item coverage of a recommender system is the percentage of items from the item catalogue that get recommended [174]. This metric is chosen because it measures the proportion of items in

the catalogue that have been recommended at least once over the number of potential items. If a recommender system has low coverage it will limit the recommendations for the user thus having a direct impact on business revenue of the system and the users' satisfaction. This metric is defined as [7, 174]

$$coverage = \frac{|\hat{I}|}{|I|}, \tag{3.39}$$

where $|I|$ is the total number of items in the test set catalogue and $|\hat{I}|$ is the number of items in I recommended at least once by the recommender system.

4. Shannon entropy is a measure that provides an overview of the recommender system as a whole by measuring the distributional inequality of recommendations across all users [7]. This metric is chosen to better understand the capability of each deep learning feature to generate different video recommendations within a certain item coverage value over the whole set of users. Shannon Entropy is defined as [7]

$$SE = -\sum_{i \epsilon I} \frac{rec(i)}{rec_t} \cdot ln \frac{rec(i)}{rec_t}, \tag{3.40}$$

where I is the set of items in the scenario being evaluated, $rec(i)$ is the number of times item i has been recommended across all users, and rec_t is the total number of recommendations. As can be seen in this equation, the Shannon entropy has a value range between 0 and $ln(n)$, which is when one item is recommended many times and when n items are recommended with equal frequency [171].

3.4.4.3 Summary of metrics

Table 3.1 summarises the aforementioned metrics that are used to evaluate the proposed improvements for the video recommendation systems.

Table 3.1. Summary of the evaluation metrics

Metrics	Description
	Accuracy metrics
MAP@N	Overall precision of the recommender system by measuring how many of the recommended items are in the set of true relevant items.

Table 3.1 continued from previous page

NDCG@N	Utility-based ranking measure that considers the order of recommended items in the list.
Beyond-accuracy metrics	
Intra-list Diveristy	Measures the ability of the system to include items that cover the user's entire set of preferences.
Inter-list Diveristy	Measures the different users recommendation lists' uniqueness.
Coverage	The number of items the recommendation model is capable of recommending.
Shannon Entropy (SE)	Represents when one item is being frequently recommended and when a set of items are recommended for the same number of times.

When evaluating the system in terms of accuracy metrics, large values are desired, as they represent video relevance with respect to the user preferences. On the other hand, moderate values are preferred when evaluating the system in terms of beyond-accuracy metrics, aside from the coverage metric [176]. This is because in general, only a random recommender obtains the highest diversity and this comes at the price of accuracy [7].

3.4.5 Baseline recommendation algorithms

In this research study, aside from the genre and hand-crafted video content feature baselines, a popularity-based ranking model, called *TopPopular* (TopPop) [177], is also included to validate the effectiveness of the scaled-CER model in the item warm-start scenario. In addition, a *random* recommendation model is also included to verify the results. The models are described as follows:

1. **Random recommender** is a non-personalised model that generates a randomly ordered list of videos in the item warm-start or cold-start scenarios. It is chosen to be used as a check for the recommender system evaluation.
2. **TopPop** is a non-personalised model that recommends the top-N most popular items to all the users and is a hard baseline to beat [177]. This model calculates the popularity of items by counting the number of times the item is rated by the users. Thus, it only works in the item

warm-start scenario. During inference, this model returns an ordered list of videos in descending order of the global popularity rating.

3.5 CHAPTER SUMMARY

This chapter described the process of developing a recommendation framework. The different video content feature modalities extracted as well as the feature aggregation methods, recommendation models, and multimodal fusion techniques that are investigated in this research work were explained in depth. The matrix scaling technique used to improve the recommendation quality of the hybrid recommendation model was also discussed. In addition, the experimental setup used to assess the performance of the various methods presented and the evaluation metrics utilised to compare and analyse them were described in detail. In Chapter 4, the results of the experiments described in Section 3.4 are reported.

CHAPTER 4 RESULTS

4.1 CHAPTER OVERVIEW

This chapter presents the results of the experiments described in Chapter 3. Feature analyses are conducted to determine whether different feature aggregation methods create semantically meaningful video representations. Recommendation models are evaluated to assess the overall user satisfaction in terms of relevance and beyond the relevance of the recommendations with respect to user preferences. The findings are presented for the item warm-start and cold-start scenarios. In addition, different fusion methods are evaluated and their performance is reported when combining the features that best represent each video content information, namely visual-appearance, audio, and motion information. Lastly, an ablation study is presented in order to understand the cumulative effect of all video content features on recommendation quality. Note that in this research work, 5-fold cross-validation is used for each evaluation metric. Thus, unless stated otherwise, the reported results for each recommendation model were averaged over five splits.

4.2 FEATURE ANALYSES

Visual and quantitative feature analyses are conducted to determine how well each feature aggregation method creates semantically meaningful video-level descriptors until being used by the recommendation models. As described in Section 3.4.2, the quality of the feature aggregation methods is evaluated using the movie trailers in-sequel-mean-distance and movie trailers out-of-sequel-mean-distance. These distance metrics are calculated in the 2-dimensional UMAP space where their distributions are visualised, and in the original F-dimensional feature space where F is the total number of features. Tables 4.1 - 4.16 and Figures 4.1 - 4.8 show the performance measurements and the distribution of features obtained by the best feature aggregation method in the 2-dimensional UMAP space, respectively. In addition, Figures 4.9 - 4.11 show the visual distribution of the hand-crafted features used in this research work. For all figures, movie trailers are coloured according their genres and their

shape is chosen according to their sequel. If the marker is not opaque, it means that the trailer is not assigned to any sequel. The movie genres of the chosen sequels are action, crime, comedy, horror and adventure, where the key difference between the action and adventure genres is the setting. Action genre usually focus on the execution of the plot, instead of the plot itself while in adventure genre typically, though not always, there is a search or quest for something set in a fantasy or exotic location [178, 179]. The colour **blue** represents action movie trailers, **sienna** represents crime movie trailers, **green** represents comedy movie trailers, **red** represents horror movie trailers and lastly **dark salmon** represents adventure movie trailers. The visual and quantitative results for each video content feature are presented in the sections below.

4.2.1 Object and scene features

In this section, the visualisation of the *Obj(IN)* and scene features, as well as quantitative measurements of the feature aggregation methods, are presented.

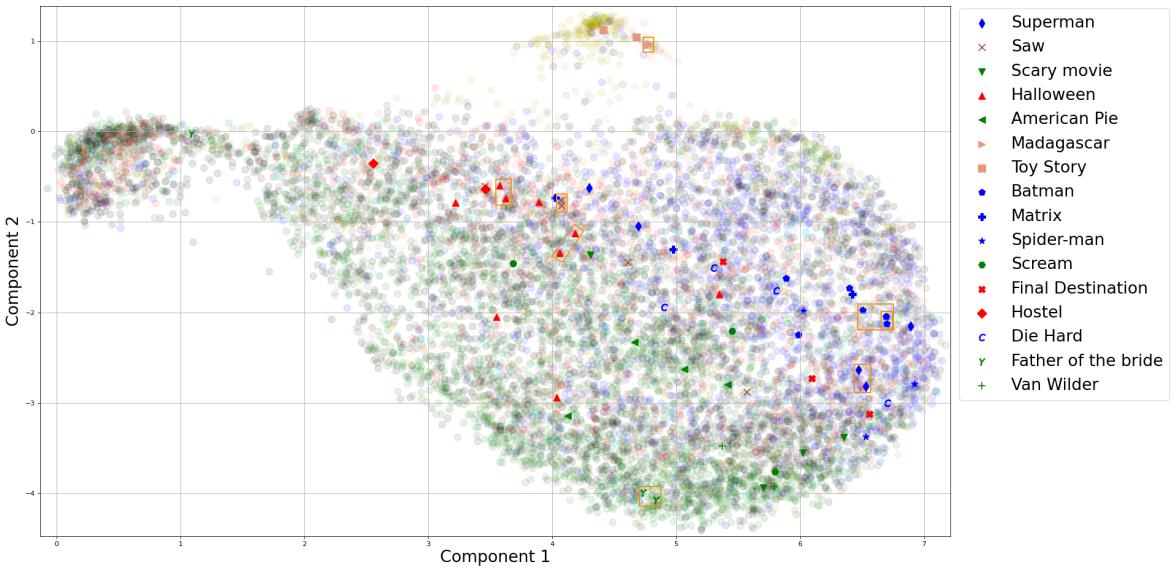


Figure 4.1. Visualisation of the *Obj(IN)* video-level descriptors in the UMAP space obtained using the maximum aggregation method. These descriptors represent movie trailers of different genres. They are obtained using the maximum aggregation method. The figure indicates that genre clustering exists and some movie trailers in the same sequel can be seen to be close to each other, for example *Batman* and *Halloween* enclosed by rectangles.

Table 4.1. Results for various feature aggregation methods in terms of Bhattacharyya distance. The Bhattacharyya distance is calculated using the movie trailers in-sequel-mean-distance and movie trailers out-of-sequel-mean-distance in the 2-dimensional UMAP space. The movie trailer's information is represented by *Obj(IN)* video-level descriptors. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	**1.813**	**1.128**	**2.798**	**1.184**	**0.0912**
Mean	2.223	1.638	2.996	1.315	0.0459
Median	2.337	1.687	3.257	1.435	0.0497
Variance	1.937	1.112	2.650	1.161	0.0497
MAD	1.796	1.366	2.710	1.193	0.0680
IQR	1.746	1.361	2.579	1.243	0.0531

Table 4.2. Quality of multiple feature aggregation methods in terms of Bhattacharyya distance in the original *Obj(IN)* 2048-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	**1.248**	**0.0989**	**1.374**	**0.0792**	**0.2609**
Mean	1.081	0.2460	1.353	0.2110	0.1827
Median	1.089	0.2639	1.359	0.2242	0.1588
Variance	1.181	0.1451	1.364	0.1189	0.2493
MAD	1.145	0.2145	1.380	0.1542	0.2239
IQR	1.165	0.1833	1.380	0.1334	0.2492

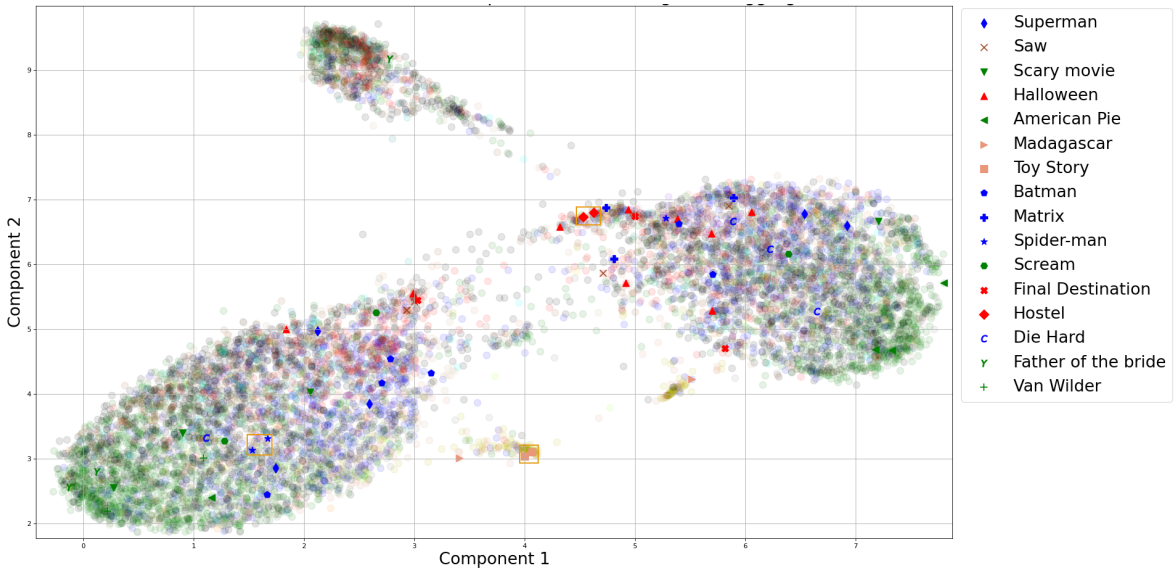


Figure 4.2. Visualisation of the scene-centric video-level descriptors in the UMAP space obtained using the mean aggregation method. The figure reveals that several sets of the scene features have similar semantics, since a number of movie trailers in the same sequel tend to be very close to each other. For example *Hostel, Toy Story* and *Spider-man* enclosed by rectangles.

Table 4.3. Quality of multiple feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the scene features from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	3.024	1.551	3.536	1.335	0.0212
Mean	**2.811**	**1.860**	**3.603**	**1.543**	**0.0355**
Median	3.019	1.849	3.755	1.564	0.0300
Variance	2.639	1.677	3.152	1.360	0.0250
MAD	2.707	1.883	3.317	1.461	0.0323
IQR	2.498	1.685	3.125	1.398	0.0292

Table 4.4. Quality of multiple feature aggregation methods in terms of Bhattacharyya distance in the original scene 2208-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.263	0.1211	1.386	0.1029	0.1563
Mean	1.155	0.2080	1.369	0.1692	0.1707
Median	1.170	0.2097	1.372	0.1619	0.1612
Variance	**1.229**	**0.1474**	**1.380**	**0.1140**	**0.1821**
MAD	1.233	0.1642	1.388	0.1226	0.1634
IQR	1.235	0.1601	1.387	0.1166	0.1730

4.2.2 Action features

Visualisation of the deep learning action-centric features as well as quantitative measurements of the feature aggregation methods.

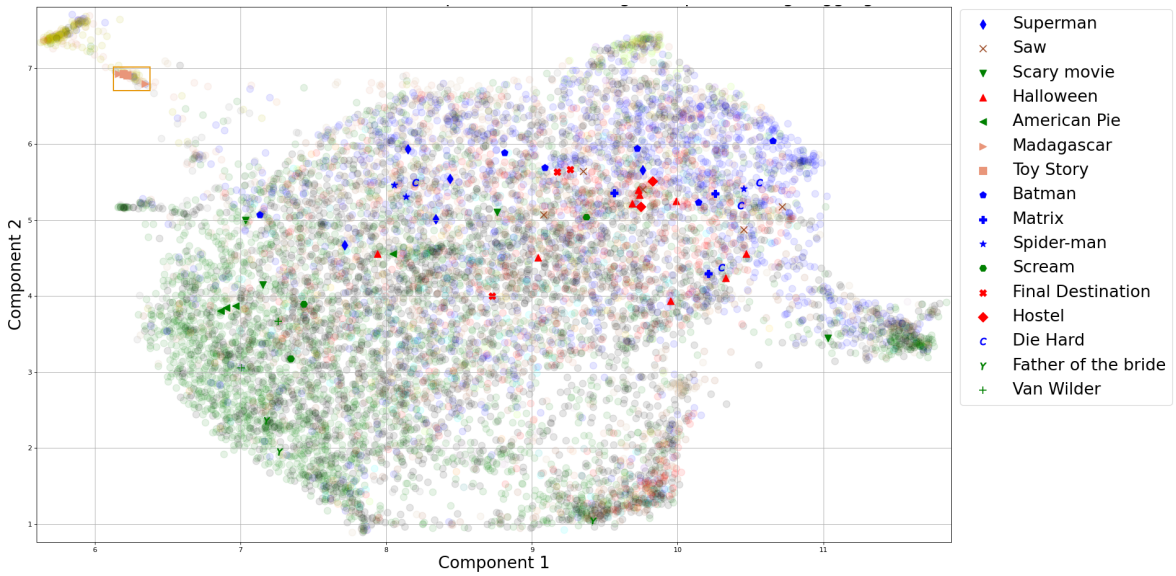


Figure 4.3. Visualisation of the *Action(IG)* video-level descriptors in the UMAP space obtained using interquartile range aggregation method. The figure suggests that action and horror movie trailers have similar semantics with respect to their motion information. A number of comedy movie trailers (green) are clearly separated from the action (blue) and horror movie trailers (red). *Toy Story* and *Madagascar* sequels enclosed by a rectangle are close to each other, as expected since they are both animated adventure films.

Table 4.5. Quality of different feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the Action(IG) features from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.364	0.761	2.428	1.151	0.1904
Mean	1.531	1.095	2.995	1.594	0.1777
Median	1.886	1.187	3.345	1.572	0.1565
Variance	1.405	0.888	2.579	1.382	0.1750
MAD	1.470	0.825	2.673	1.314	0.2027
IQR	**1.356**	**0.758**	**2.558**	**1.321**	**0.2291**

Table 4.6. Quality of different feature aggregation methods in terms of Bhattacharyya distance in the original *Action(IG)* 512-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	In-sequel		Out-of-sequel		Bhattacharyya Distance
	Mean	Std	Mean	Std	
Maximum	1.255	0.0920	1.387	0.0876	0.2676
Mean	1.015	0.2107	1.332	0.2147	0.2763
Median	1.031	0.2211	1.337	0.2097	0.2538
Variance	**1.187**	**0.1266**	**1.374**	**0.1116**	**0.3111**
MAD	1.166	0.1600	1.377	0.1281	0.2781
IQR	1.186	0.1465	1.381	0.1158	0.2867

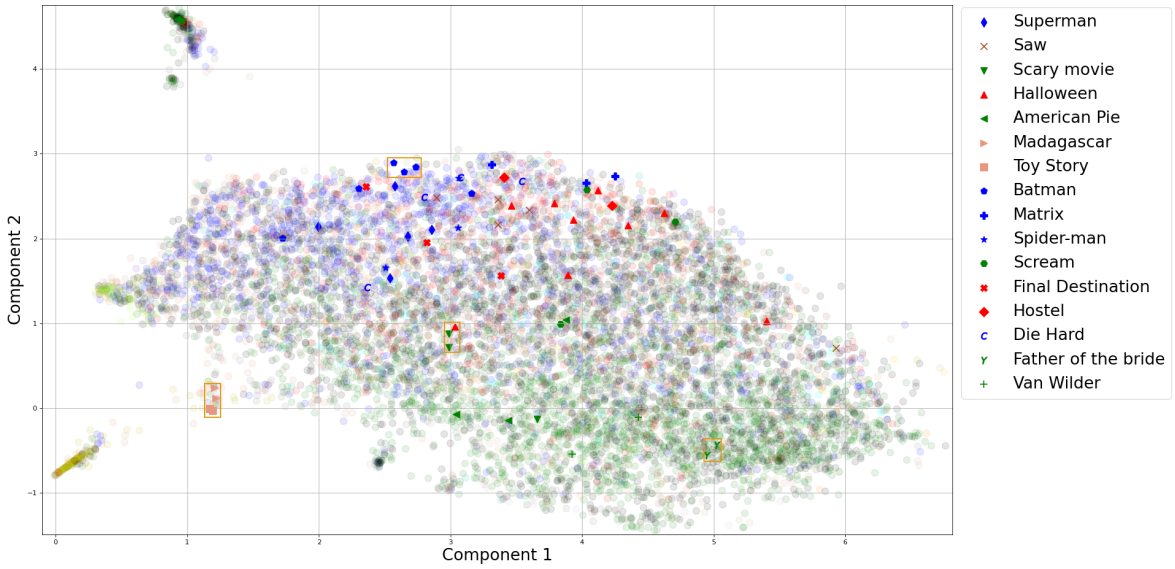


Figure 4.4. UMAP visualisation of the *Action(KN)* video-level descriptors produced by the variance aggregation method. The figure indicates that the distribution of *Action(KN)* tend to cluster action (blue) and horror (red) movies together. Some movie trailers in the same sequel are very close to each other, suggesting that they are similar in terms of the actions that occur in the trailers. For example, *Batman, Scary movie, Father of the bride, Madagascar,* and *Toy Story* enclosed by rectangles.

Table 4.7. Quality of different feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the *Action(KN)* features from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.994	0.906	2.075	0.892	0.0399
Mean	1.668	1.533	2.930	1.815	0.0776
Median	1.995	1.722	3.279	1.880	0.0654
Variance	**1.190**	**0.669**	**2.267**	**1.217**	**0.2348**
MAD	1.581	1.006	2.678	1.366	0.1276
IQR	1.401	0.835	2.517	1.334	0.1785

Table 4.8. Quality of different feature aggregation methods in terms of Bhattacharyya distance in the original *Action(KN)* 512-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.255	0.0827	1.376	0.0774	0.2829
Mean	1.046	0.1861	1.340	0.1741	0.3346
Median	1.070	0.1930	1.348	0.1683	0.2989
Variance	**1.186**	**0.1007**	**1.364**	**0.1018**	**0.3860**
MAD	1.160	0.1427	1.367	0.1230	0.3077
IQR	1.190	0.1148	1.371	0.1060	0.3359

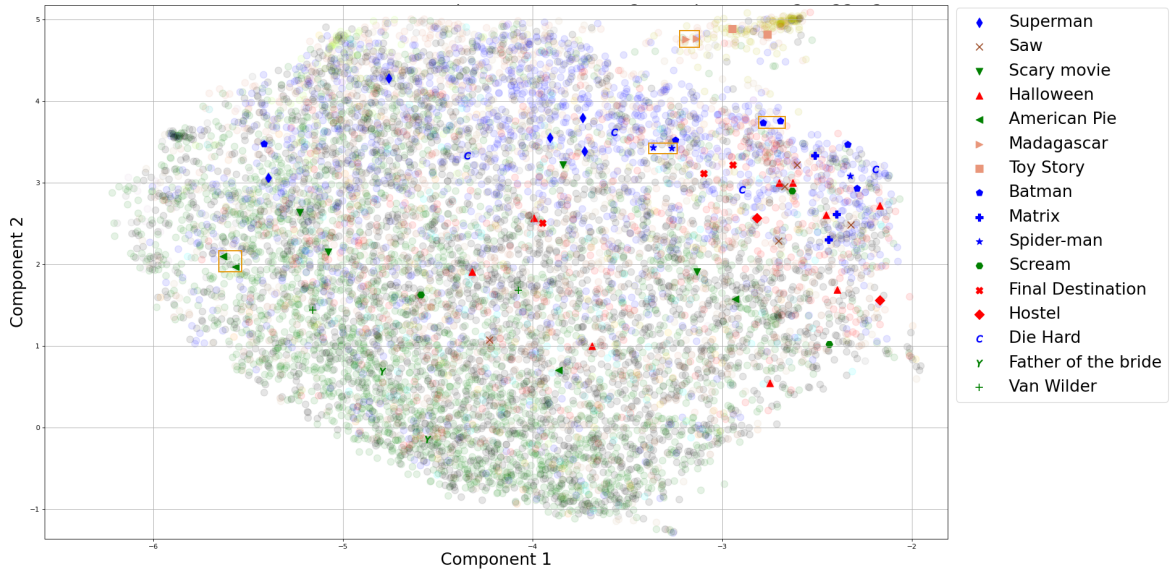


Figure 4.5. *Action(UCF)* feature visualisation in the UMAP space obtained using interquartile range aggregation method. Some movie trailers in the same sequels, such as *American Pie* enclosed by a rectangle, as well as a large part of action (blue) and horror (red) movie genres are close to one another, which suggests they have very similar actions in their narratives.

Table 4.9. Performance of multiple feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the Action(UCF) features extracted from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.349	0.833	2.210	1.016	0.1171
Mean	1.572	1.733	2.613	2.031	0.0444
Median	1.718	1.879	2.672	2.097	0.0316
Variance	1.350	0.798	2.238	1.218	0.1362
MAD	1.569	1.226	2.775	1.521	0.1067
IQR	**1.371**	**0.753**	**2.334**	**1.238**	**0.1698**

Table 4.10. Performance of multiple feature aggregation methods in terms of Bhattacharyya distance in the original *Action(UCF)* 2048-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.264	0.0707	1.377	0.0787	0.2872
Mean	1.103	0.1629	1.343	0.1677	0.2641
Median	1.125	0.1657	1.348	0.1610	0.2335
Variance	**1.231**	**0.0731**	**1.374**	**0.0910**	**0.3860**
MAD	1.207	0.1035	1.370	0.1130	0.2871
IQR	1.246	0.0721	1.379	0.0891	0.3506

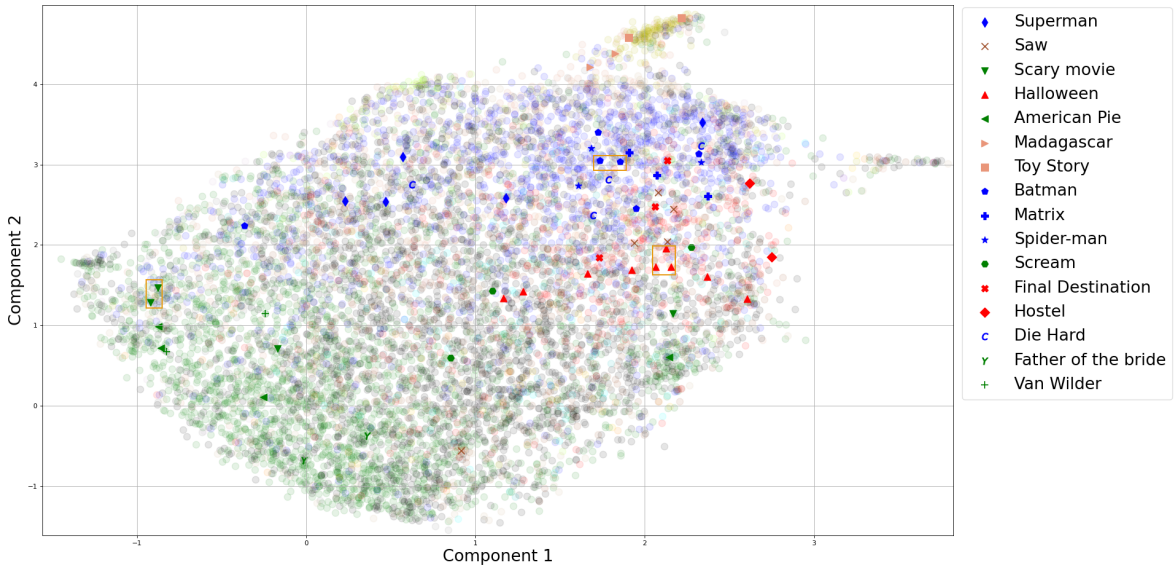


Figure 4.6. Projection of various *Action(HMDB)* video-level descriptors in the UMAP space produced by the variance feature aggregation method. The figure indicates that genre clustering exists. Movies in different sequels and with dissimilar actions (comedy (green) and action (blue)) tend to be far apart from movies in different sequels but with similar actions (horror (red) and action (blue)). This reveals that similar semantics are being properly captured by the *Action(HMDB)* features.

Table 4.11. Performance of multiple feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the *Action(HMDB)* features extracted from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	In-sequel		Out-of-sequel		Bhattacharyya Distance
	Mean	Std	Mean	Std	
Maximum	1.455	0.944	2.213	1.068	0.0744
Mean	2.684	1.901	2.684	1.901	0.0266
Median	2.073	2.069	2.766	1.838	0.0192
Variance	**1.274**	**0.957**	**2.431**	**1.382**	**0.1514**
MAD	1.803	1.881	2.589	1.697	0.0267
IQR	1.617	1.201	2.513	1.424	0.0650

Table 4.12. Performance of multiple feature aggregation methods in terms of Bhattacharyya distance in the original *Action(HMDB)* 2048-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	In-sequel		Out-of-sequel		Bhattacharyya Distance
	Mean	Std	Mean	Std	
Maximum	1.258	0.0846	1.373	0.0833	0.2364
Mean	1.092	0.1884	1.324	0.1753	0.2038
Median	1.117	0.1853	1.331	0.1651	0.1904
Variance	**1.217**	**0.1004**	**1.368**	**0.1001**	**0.2817**
MAD	1.193	0.1293	1.358	0.1195	0.2206
IQR	1.225	0.1023	1.368	0.1004	0.2472

4.2.3 Deep learning sound features

Feature visualisation of the deep learning sound features as well as quantitative measurements of the feature aggregation methods are presented here. In contrast to the visual deep learning features, the audio deep learning features have been combined using only the feature aggregation methods that

are a measure of location. These methods have been found to be robust to audio segments that are less representative of the audio signal [180]. As a result, they form audio descriptors that are more meaningful to describe the audio information which leads to state-of-the-art performance on audio classification tasks [180].

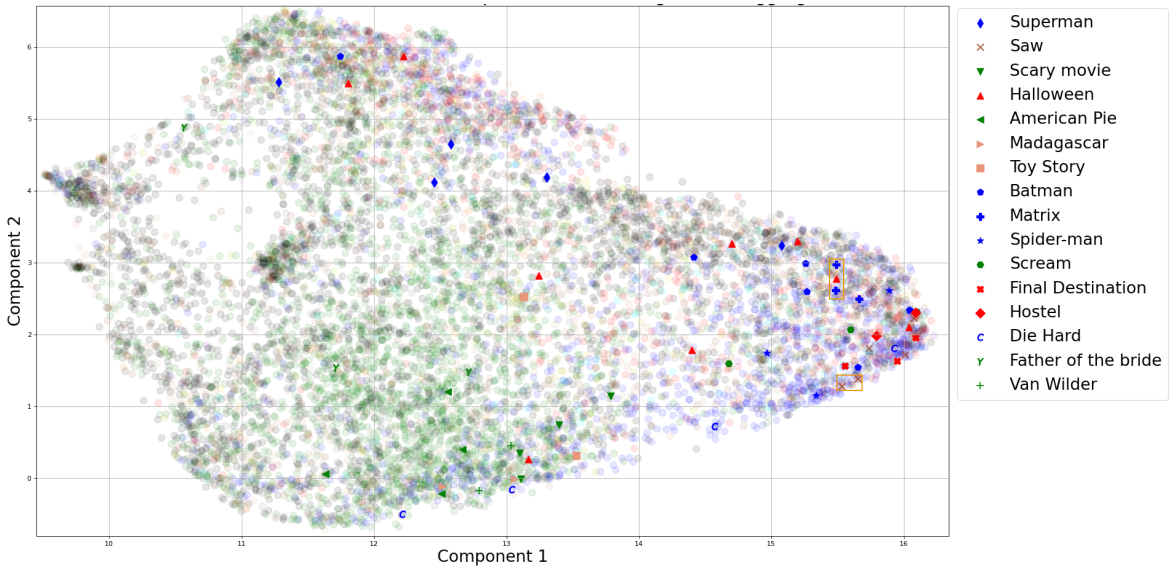


Figure 4.7. UMAP visualisation of the *VGGish* video-level descriptors produced by the median feature aggregation method. The figure presents very dense clustering for action (blue) and horror (red) movies, while the comedy (green) movies are more spread out. Some action and horror movie trailers tend to be close to one another.

Table 4.13. Performance of multiple feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the *VGGish* features extracted from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	2.415	1.358	2.901	1.019	0.0407
Mean	1.981	1.407	2.953	1.612	0.0562
Median	**1.845**	**1.260**	**2.784**	**1.439**	**0.0646**

Table 4.14. Performance of multiple feature aggregation methods in terms of Bhattacharyya distance in the original *VGGish* 128-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.205	0.125	1.339	0.1339	0.1343
Mean	**1.020**	**0.218**	**1.267**	**0.2514**	**0.1435**
Median	1.041	0.210	1.274	0.2510	0.1358

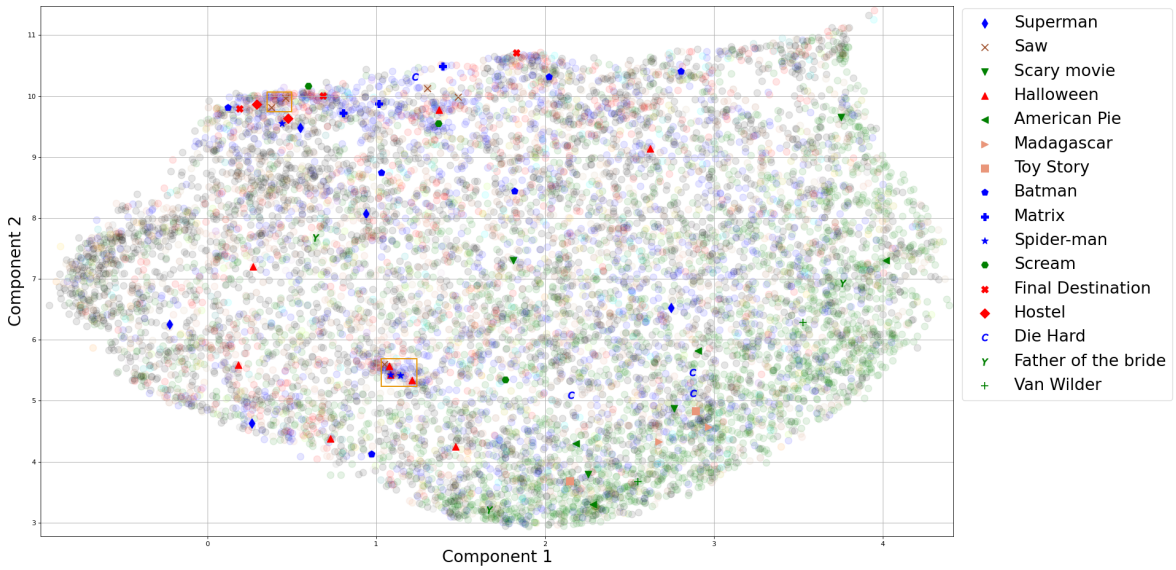


Figure 4.8. UMAP visualisation of the *soundNet* video-level descriptors produced by the median feature aggregation method. The figure presents a dense clustering pattern for comedy (green) movies, while action (blue) and horror (red) movies are more spread-out. Some movies in the same sequel (*spider-man, saw*) are close to each other while others are far apart (*Batman, scary movie*).

Table 4.15. Performance of multiple feature aggregation methods assessed in terms of Bhattacharyya distance in the 2-dimensional UMAP space using the *soundNet* features extracted from movie trailers. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	3.135	2.185	3.383	2.023	0.0032
Mean	2.191	1.202	2.577	1.044	0.0195
Median	**2.241**	**1.215**	**2.856**	**1.200**	**0.0324**

Table 4.16. Performance of multiple feature aggregation methods in terms of Bhattacharyya distance in the original *soundNet* 256-dimensional feature space. The feature aggregation method with the largest distance between in-sequel and out-of-sequel is marked in bold.

Feature Aggregation	**In-sequel**		**Out-of-sequel**		**Bhattacharyya Distance**
	Mean	**Std**	**Mean**	**Std**	
Maximum	1.321	0.119	1.366	0.1039	0.0250
Mean	1.150	0.246	1.320	0.2310	0.0647
Median	**1.158**	**0.240**	**1.343**	**0.2402**	**0.0738**

4.2.4 Hand-crafted features

Visualisation of the hand-crafted features, namely iDT, MoSIFT, and MFCC features in the UMAP space are presented here.

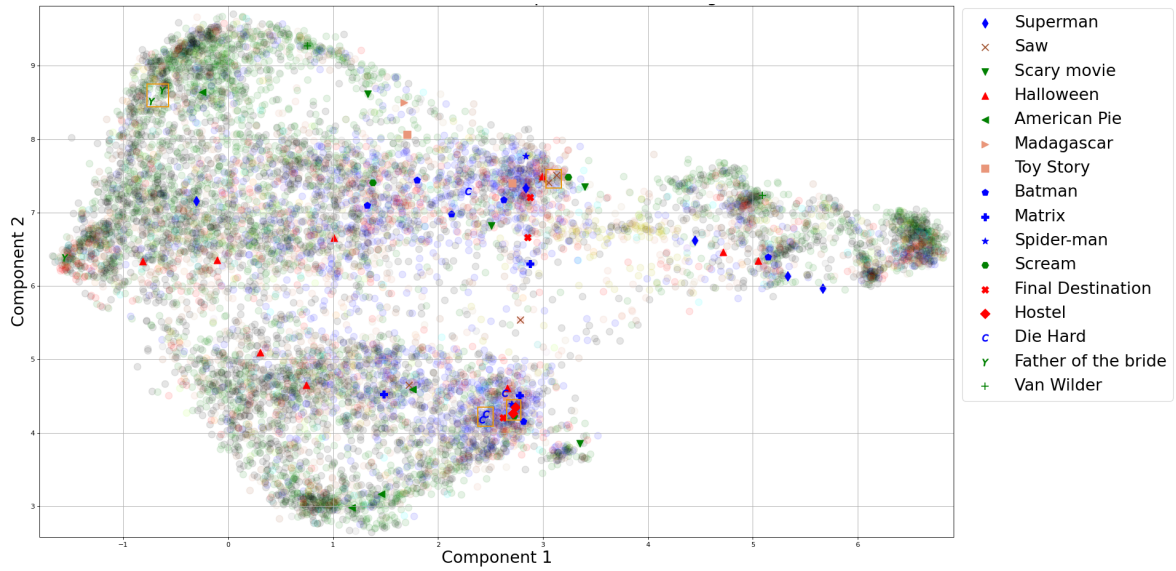


Figure 4.9. Visualisation of the movie trailers represented by iDT video-level descriptors in the UMAP space. The figure shows that some horror (red) and action (blue) movie trailers are close to each other in terms of their iDT features such as the movie sequel *Hostel* and *Die Hard* enclosed by rectangles. However, in some movie sequels the movie trailers are further apart suggesting that they are not similar in terms of iDT features, for example *Halloween* and *Matrix* movie sequels.

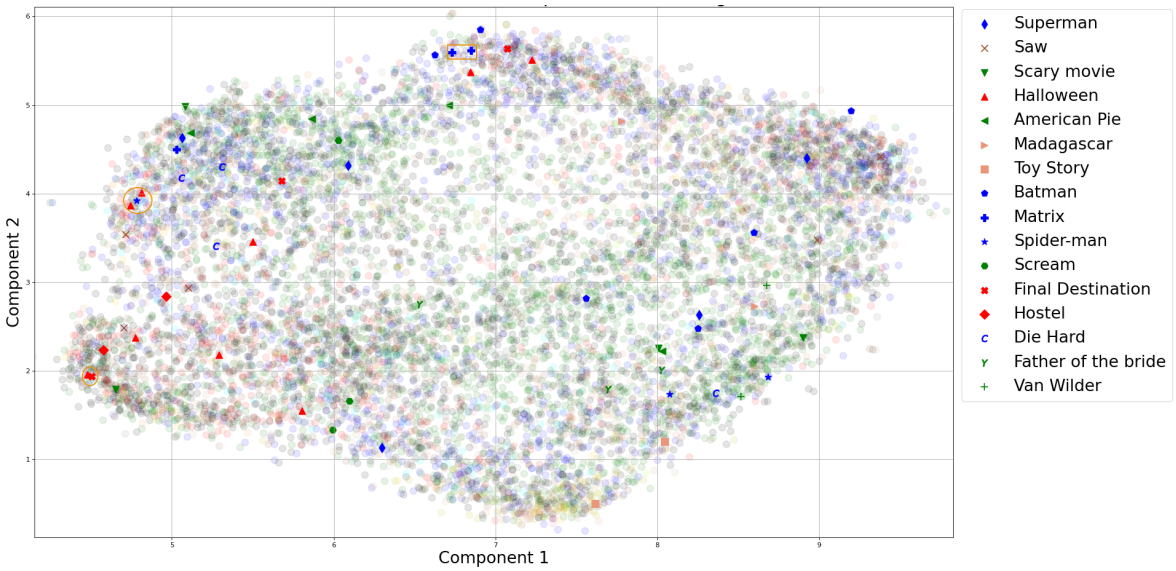


Figure 4.10. Visualisation of the movie trailers represented by MoSIFT video-level descriptors in the UMAP space. The figure indicates that a number of action (blue) and comedy (green) movie trailers are not similar in terms of MoSIFT features since they are in different locations. On the other hand, various horror (red) movie trailers are close to each other.

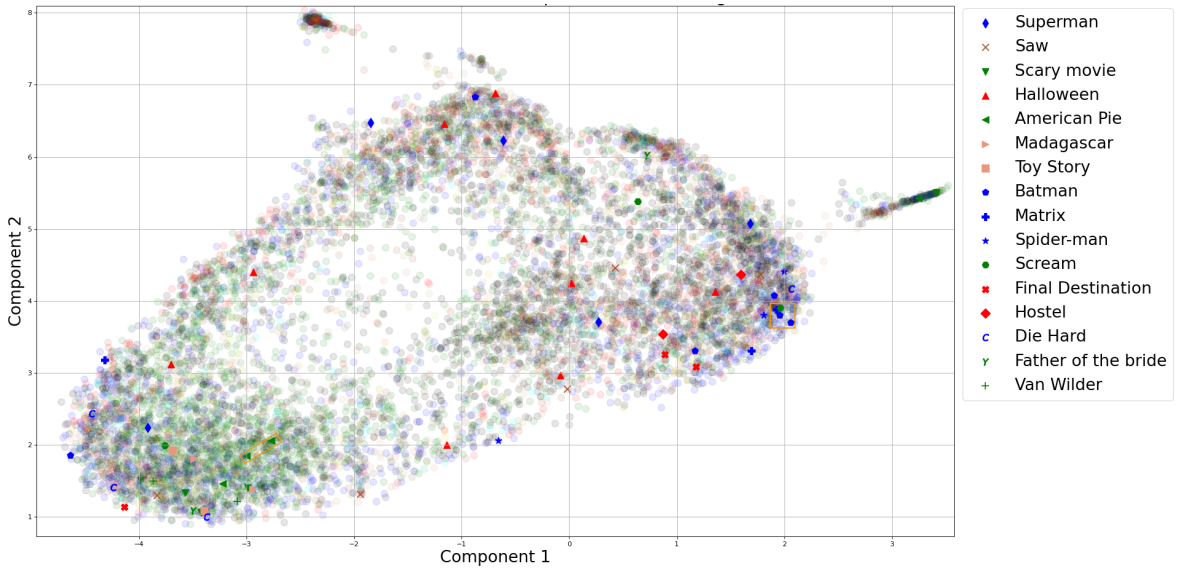


Figure 4.11. Visualisation of the movie trailers represented by MFCC video-level descriptors in the UMAP feature space. The figure indicates that the movies when represented in terms of MFCC features are quite diverse where genre clustering does not necessarily exists. However, few trailers in the same sequel can be seen to be close to each other, as for example *Batman* enclosed by a rectangle.

4.3 RECOMMENDATION IN WARM-START SCENARIO

In this section, the performance of the baselines described in Section 3.4.5 as well as the performance of the CER and scaled-CER models is reported, when using each feature presented in Section 3.3.1 on the content description of their systems. The quality of the recommendations produced by these models is assessed in terms of accuracy and beyond-accuracy metrics. As described in Section 3.4.3, the item warm-start scenario represents the case when some preference data for items in that scenario have been used to train the video recommendation model. In this work, the videos in the catalogue are sorted in descending order, based on the ratings estimated by the model being evaluated. Next, the Top-n videos are chosen to be the first n videos in the recommendation list. The length n of the recommendation list returned to each user is also known as the cut-off value.

Accuracy and beyond-accuracy metrics are calculated for 6 different cut-off values selected from {5, 10, 15, 20, 25, 30}. These values are chosen because the length of the recommendation lists equal to one of these values is manageable and realistic for a user to obtain in a real-word application [14]. For feature aggregation, only the results for the variant that led to the best performance are reported.

4.3.1 Accuracy metrics

As mentioned in Section 3.4.4.1, it is important to assess the ability of the recommender models to recommend relevant items. This is necessary to understand how effective they are in predicting the preference scores of each user to the items in the catalogue. Given the value of n as outlined above, the results of the experiments are presented below

Table 4.17. Results for the random, TopPop, CER, and scaled-CER recommender models with respect to MAP in the item warm-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The best performance along the respective metric is highlighted in bold.

Recommender models	Feature Agg.	MAP@5	MAP@10	MAP@15	MAP@20	MAP@25	MAP@30
non-personalised							
Random	-	0.0004	0.0004	0.0004	0.0004	0.0005	0.0005
TopPop	-	0.0684	0.0692	0.0716	0.0736	0.0755	0.0768
CER							
Genres	-	0.1101	0.1161	0.1224	0.1270	0.1305	0.1332
Obj(IN)	Max	0.1110	0.1171	0.1233	0.1279	0.1314	0.1341
Action(IG)	Var	0.1107	0.1169	0.1232	0.1278	0.1313	0.1340
Action(KN)	Mad	0.1108	0.1168	0.1230	0.1276	0.1311	0.1338
Action(HMDB)	Var	0.1113	0.1174	0.1236	0.1283	0.1318	0.1344
Action(UCF)	Var	0.1115	0.1175	0.1237	0.1283	0.1318	0.1345
iDT	Fv	0.1105	0.1165	0.1227	0.1273	0.1308	0.1335
MoSIFT	Fv	0.1111	0.1171	0.1233	0.1280	0.1314	0.1341
Scene	Var	0.1107	0.1167	0.1229	0.1275	0.1310	0.1336
MFCC	Fv	0.1112	0.1170	0.1232	0.1278	0.1313	0.1339
SoundNet	Mean	0.1100	0.1161	0.1224	0.1270	0.1305	0.1332
VGGish	Mean	0.1105	0.1166	0.1227	0.1274	0.1309	0.1335
scaled-CER							
Genres	-	0.1531	0.1517	0.1562	0.1603	0.1638	0.1665
Obj(IN)	Max	**0.1536**	**0.1524**	**0.1568**	**0.1609**	**0.1644**	**0.1671**
Action(IG)	Var	0.1530	0.1517	0.1562	0.1604	0.1638	0.1665
Action(KN)	Mad	0.1534	0.1521	0.1565	0.1606	0.1640	0.1668
Action(HMDB)	Var	0.1533	0.1520	0.1564	0.1606	0.1640	0.1668
Action(UCF)	Var	0.1535	0.1522	0.1566	0.1608	0.1642	0.1669
iDT	Fv	**0.1536**	0.1523	0.1567	**0.1609**	**0.1644**	**0.1671**
MoSIFT	Fv	0.1535	0.1522	0.1566	0.1608	0.1642	0.1670
Scene	Var	0.1531	0.1518	0.1562	0.1603	0.1638	0.1665
MFCC	Fv	0.1529	0.1517	0.1561	0.1603	0.1637	0.1664
SoundNet	Mean	0.1532	0.1521	0.1565	0.1606	0.1641	0.1668
VGGish	Mean	0.1531	0.1518	0.1563	0.1605	0.1639	0.1666

Table 4.17 shows the performance of different recommender models in terms of MAP metrics. As presented in this table, it includes experiments that evaluate non-personalised recommender models that do not exploit video content features; experiments evaluating the CER model using genre, hand-crafted and deep learning features; and finally, experiments evaluating the scaled-CER model, where the CER model is improved through the use of a matrix scaling technique. In this last experiment, the scaled-CER model uses the same set of video content features exploited in the previous experiments. As can be seen in Table 4.17, the scaled-CER model obtained the best overall performance using *Obj(IN)* features, and the random recommender model achieved the worst results.

Table 4.18. Results for the random, TopPop, CER, and scaled-CER recommender models with respect to NDCG in the item warm-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The best performance across the same metric is highlighted in bold.

Recommender models	Feature Agg.	NDCG@5	NDCG@10	NDCG@15	NDCG@20	NDCG@25	NDCG@30
non-personalised							
Random	-	0.0006	0.0009	0.0011	0.0013	0.0015	0.0017
TopPop	-	0.0886	0.1147	0.1302	0.1412	0.1507	0.1580
CER							
Genres	-	0.1447	0.1856	0.2102	0.2274	0.2408	0.2515
Obj(IN)	Max	0.1458	0.1867	0.2111	0.2285	0.2417	0.2523
Action(IG)	Var	0.1453	0.1863	0.2108	0.2282	0.2416	0.2522
Action(KN)	Mad	0.1454	0.1862	0.2108	0.2281	0.2414	0.2521
Action(HMDB)	Var	0.1459	0.1869	0.2114	0.2287	0.2420	0.2526
Action(UCF)	Var	0.1462	0.1871	0.2116	0.2290	0.2422	0.2529
iDT	Fv	0.1450	0.1859	0.2102	0.2275	0.2409	0.2515
MoSIFT	Fv	0.1457	0.1866	0.2110	0.2283	0.2415	0.2520
Scene	Var	0.1452	0.1860	0.2104	0.2278	0.2410	0.2516
MFCC	Fv	0.1459	0.1864	0.2108	0.2280	0.2413	0.2519
SoundNet	Mean	0.1445	0.1856	0.2101	0.2275	0.2407	0.2513
VGGish	Mean	0.1451	0.1860	0.2103	0.2276	0.2410	0.2517
scaled-CER							
genres	-	0.1841	0.2282	0.2538	0.2717	0.2853	0.2962
Obj(IN)	Max	**0.1846**	**0.2290**	**0.2546**	0.2725	0.2861	**0.2971**
Action(IG)	Var	0.1840	0.2282	0.2538	0.2718	0.2854	0.2962
Action(KN)	Mad	0.1843	0.2287	0.2542	0.2722	0.2858	0.2966
Action(HMDB)	Var	0.1843	0.2287	0.2543	0.2722	0.2859	0.2968
Action(UCF)	Var	0.1845	0.2289	0.2545	0.2724	0.2859	0.2968
iDT	Fv	**0.1846**	**0.2290**	**0.2546**	**0.2726**	**0.2863**	**0.2971**
MoSIFT	Fv	**0.1846**	0.2289	0.2545	0.2724	0.2861	0.2970
Scene	Var	0.1841	0.2284	0.2539	0.2716	0.2852	0.2961

Table 4.18 continued from previous page

MFCC	Fv	0.1840	0.2282	0.2538	0.2717	0.2854	0.2962
SoundNet	Mean	0.1844	0.2288	0.2543	0.2722	0.2859	0.2968
VGGish	Mean	0.1842	0.2284	0.2541	0.2721	0.2857	0.2966

Table 4.18 shows the performance of different recommender models in terms of NDCG metric. The scaled-CER model obtained the best overall performance using iDT features, and the random recommender model achieved the worst results.

4.3.2 Beyond-accuracy metrics

As stated in Section 3.4.4.2, when a minimum acceptable performance with regards to accuracy metrics is obtained, it is equally important to assess the ability of the recommender models beyond the relevance of the items they recommend. Thus, the beyond-accuracy metrics are computed for recommendation lists of length equal to the one used in the accuracy metric measurements. Note that given the unacceptably low performance of the random recommender model in terms of accuracy metrics, the results for this model are not taken into account when determining the best results, however they are still presented in grey for completeness. The results from the experiments are reported below.

Table 4.19. Results for the TopPop, CER, and scaled-CER recommender models with respect to intra-list diversity metric in the item warm-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result across the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Div. @5 IntraL**	**Div. @10 IntraL**	**Div. @15 IntraL**	**Div. @20 IntraL**	**Div. @25 IntraL**	**Div. @30 IntraL**
non-personalised							
Random	-	0.5947	0.6694	0.6942	0.7066	0.7140	0.7189
TopPop	-	**0.6026**	**0.6419**	**0.6869**	**0.6899**	**0.6971**	**0.7079**
CER							
Genres	-	0.4804	0.5659	0.5992	0.6178	0.6299	0.6383
Obj(IN)	Max	0.4899	0.5749	0.6076	0.6256	0.6372	0.6455
Action(IG)	Var	0.4883	0.5740	0.6067	0.6249	0.6366	0.6448
Action(KN)	Mad	0.4894	0.5739	0.6065	0.6245	0.6364	0.6448
Action(HMDB)	Var	0.4897	0.5746	0.6071	0.6250	0.6368	0.6451
Action(UCF)	Var	0.4886	0.5736	0.6065	0.6246	0.6364	0.6447
iDT	Fv	0.4901	0.5746	0.6073	0.6254	0.6372	0.6455
MoSIFT	Fv	0.4905	0.5744	0.6070	0.6251	0.6368	0.6451

Table 4.19 continued from previous page

Scene	Var	0.4895	0.5744	0.6069	0.6251	0.6369	0.6452
MFCC	Fv	0.4913	0.5753	0.6076	0.6256	0.6374	0.6458
SoundNet	Mean	0.4907	0.5745	0.6069	0.6248	0.6367	0.6451
VGGish	Mean	0.4897	0.5740	0.6063	0.6245	0.6364	0.6448
scaled-CER							
Genres	-	0.4707	0.5551	0.5883	0.6069	0.6191	0.6277
Obj(IN)	Max	0.4777	0.5625	0.5953	0.6135	0.6254	0.6338
Action(IG)	Var	0.4774	0.5624	0.5954	0.6138	0.6257	0.6342
Action(KN)	Mad	0.4770	0.5623	0.5955	0.6139	0.6258	0.6342
Action(HMDB)	Var	0.4776	0.5628	0.5957	0.6140	0.6260	0.6344
Action(UCF)	Var	0.4768	0.5622	0.5951	0.6134	0.6254	0.6339
iDT	Fv	0.4765	0.5620	0.5954	0.6140	0.6260	0.6346
MoSIFT	Fv	0.4783	0.5631	0.5962	0.6146	0.6266	0.6351
Scene	Var	0.4788	0.5626	0.5954	0.6136	0.6254	0.6337
MFCC	Fv	0.4787	0.5638	0.5969	0.6152	0.6271	0.6355
SoundNet	Mean	0.4782	0.5634	0.5964	0.6146	0.6264	0.6348
VGGish	Mean	0.4777	0.5627	0.5957	0.6140	0.6258	0.6341

Table 4.19 shows the performance of different recommender models in terms of intra-list diversity metric. The TopPop model obtained the highest intra-list diversity, and the scaled-CER model presents a lower intra-list diversity compared to the CER model. The lowest results are obtained by the scaled-CER model using genre features.

Table 4.20. Results for the TopPop, CER, and scaled-CER recommender models with respect to inter-list diversity metric in the item warm-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result across the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Div. InterL @5**	**Div. InterL @10**	**Div. InterL @15**	**Div. InterL @20**	**Div. InterL @25**	**Div. InterL @30**
non-personalised							
Random	-	0.9994	0.9988	0.9982	0.9976	0.9970	0.9964
TopPop	-	0.2539	0.2140	0.1873	0.1692	0.1561	0.1454
CER							
Genres	-	**0.9627**	**0.9443**	**0.9315**	**0.9213**	**0.9127**	**0.9052**
Obj(IN)	Max	0.9625	0.9441	0.9310	0.9207	0.9120	0.9045
Action(IG)	Var	0.9616	0.9433	0.9305	0.9204	0.9118	0.9044
Action(KN)	Mad	0.9626	0.9440	0.9311	0.9209	0.9124	0.9049
Action(HMDB)	Var	0.9623	0.9438	0.9309	0.9207	0.9121	0.9046
Action(UCF)	Var	0.9617	0.9435	0.9306	0.9204	0.9118	0.9043

Table 4.20 continued from previous page

iDT	Fv	0.9615	0.9432	0.9305	0.9204	0.9118	0.9044
MoSIFT	Fv	0.9623	0.9439	0.9309	0.9208	0.9123	0.9049
Scene	Var	0.9622	0.9440	0.9311	0.9209	0.9124	0.9049
MFCC	Fv	0.9619	0.9437	0.9309	0.9208	0.9123	0.9049
SoundNet	Mean	0.9610	0.9425	0.9296	0.9195	0.9109	0.9035
VGGish	Mean	0.9617	0.9433	0.9305	0.9202	0.9115	0.9039
scaled-CER							
Genres	-	0.9479	0.9291	0.9168	0.9072	0.8992	0.8923
Obj(IN)	Max	0.9478	0.9288	0.9163	0.9065	0.8984	0.8913
Action(IG)	Var	0.9469	0.9277	0.9152	0.9056	0.8975	0.8905
Action(KN)	Mad	0.9470	0.9279	0.9152	0.9054	0.8973	0.8902
Action(HMDB)	Var	0.9455	0.9258	0.9132	0.9034	0.8954	0.8883
Action(UCF)	Var	0.9471	0.9280	0.9156	0.9059	0.8978	0.8907
iDT	Fv	0.9468	0.9279	0.9155	0.9058	0.8977	0.8906
MoSIFT	Fv	0.9476	0.9286	0.9162	0.9064	0.8983	0.8912
Scene	Var	0.9481	0.9292	0.9168	0.9071	0.8990	0.8920
MFCC	Fv	0.9471	0.9279	0.9153	0.9055	0.8974	0.8903
SoundNet	Mean	0.9465	0.9273	0.9148	0.9051	0.8969	0.8898
VGGish	Mean	0.9469	0.9278	0.9152	0.9055	0.8973	0.8902

Table 4.20 shows the performance of different recommender models in terms of the inter-list diversity metric. The CER model using genre features obtained the highest inter-list diversity, and the TopPop model achieved the lowest inter-list diversity. All the variants of the CER model achieved results higher than the scaled-CER model.

Table 4.21. Results for the TopPop, CER, and scaled-CER recommender models with respect to item coverage in the item warm-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result along the same metric is highlighted in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Item Cov.@5**	**Item Cov.@10**	**Item Cov.@15**	**Item Cov.@20**	**Item Cov.@25**	**Item Cov.@30**
non-personalised							
Random	-	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
TopPop	-	0.0030	0.0046	0.0061	0.0074	0.0088	0.0104
CER							
Genres	-	**0.1568**	0.1877	0.2092	0.2270	0.2424	0.2561
Obj(IN)	Max	0.1540	0.1820	0.2007	0.2149	0.2271	0.2373
Action(IG)	Var	0.1558	0.1835	0.2016	0.2160	0.2282	0.2387
Action(KN)	Mad	0.1541	0.1824	0.2011	0.2150	0.2269	0.2373

Table 4.21 continued from previous page

Action(HMDB)	Var	0.1538	0.1818	0.1999	0.2139	0.2262	0.2364
Action(UCF)	Var	0.1559	0.1824	0.1998	0.2138	0.2265	0.2373
iDT	Fv	0.1542	0.1811	0.2001	0.2148	0.2261	0.2369
MoSIFT	Fv	0.1543	0.1813	0.1999	0.2145	0.2257	0.2359
Scene	Var	0.1548	0.1822	0.2007	0.2154	0.2274	0.2383
MFCC	Fv	0.1539	0.1806	0.1995	0.2138	0.2259	0.2362
SoundNet	Mean	0.1554	0.1833	0.2032	0.2175	0.2302	0.2407
VGGish	Mean	0.1538	0.1817	0.2014	0.2167	0.2288	0.2396
scaled-CER							
Genres	-	0.1480	**0.1935**	**0.2283**	**0.2598**	**0.2853**	**0.3096**
Obj(IN)	Max	0.1372	0.1771	0.2055	0.2287	0.2491	0.2669
Action(IG)	Var	0.1404	0.1822	0.2101	0.2338	0.2552	0.2738
Action(KN)	Mad	0.1380	0.1779	0.2068	0.2307	0.2515	0.2706
Action(HMDB)	Var	0.1385	0.1791	0.2103	0.2330	0.2560	0.2746
Action(UCF)	Var	0.1375	0.1774	0.2065	0.2301	0.2503	0.2686
iDT	Fv	0.1361	0.1750	0.2040	0.2252	0.2455	0.2632
MoSIFT	Fv	0.1365	0.1751	0.2025	0.2246	0.2437	0.2609
Scene	Var	0.1398	0.1804	0.2092	0.2343	0.2552	0.2742
MFCC	Fv	0.1375	0.1772	0.2051	0.2279	0.2487	0.2665
SoundNet	Mean	0.1384	0.1802	0.2091	0.2321	0.2538	0.2718
VGGish	Mean	0.1387	0.1806	0.2110	0.2348	0.2572	0.2759

Table 4.21 shows the performance of different recommender models in terms of item coverage. The CER model using genre features obtained the best item coverage result for the cut-off value 5. However, the scaled-CER model using genre features obtained the highest item coverage results for the cut-off values 10, 15, 20, 25 and 30. The TopPop model achieved the lowest item coverage results across all cut-off values.

Table 4.22. Results for the TopPop, CER, and scaled-CER recommender models in terms of Shannon entropy (SE) in the item warm-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result along the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Div. @5 SE**	**Div. @10 SE**	**Div. @15 SE**	**Div. @20 SE**	**Div. @25 SE**	**Div. @30 SE**
non-personalised							
Random	-	12.9983	13.0091	13.0125	13.0142	13.0153	13.0161
TopPop	-	2.9109	3.7900	4.3073	4.6797	4.9715	5.2109
CER							
Genres	-	**8.0829**	**8.4004**	**8.6157**	**8.7802**	**8.9137**	**9.0273**
Obj(IN)	Max	8.0806	8.3943	8.6059	8.7693	8.9026	9.0152
Action(IG)	Var	8.0791	8.3948	8.6102	8.7740	8.9078	9.0209
Action(KN)	Mad	8.0801	8.3943	8.6079	8.7727	8.9066	9.0192
Action(HMDB)	Var	8.0791	8.3944	8.6090	8.7733	8.9063	9.0186
Action(UCF)	Var	8.0755	8.3912	8.6058	8.7703	8.9035	9.0162
iDT	Fv	8.0725	8.3889	8.6051	8.7699	8.9032	9.0164
MoSIFT	Fv	8.0755	8.3912	8.6062	8.7713	8.9060	9.0195
Scenes	Max	8.0763	8.3930	8.6076	8.7718	8.9057	9.0184
MFCC	Fv	8.0725	8.3918	8.6080	8.7730	8.9071	9.0198
SoundNet	Mean	8.0627	8.3799	8.5958	8.7612	8.8954	9.0093
VGGish	Mean	8.0673	8.3864	8.6009	8.7649	8.8971	9.0088
scaled-CER							
Genres	-	7.5366	8.0164	8.3193	8.5408	8.7156	8.8602
Obj(IN)	Max	7.5225	7.9977	8.2966	8.5157	8.6888	8.8316
Action(IG)	Var	7.4964	7.9771	8.2807	8.5025	8.6773	8.8209
Action(KN)	Mad	7.4966	7.9751	8.2763	8.4977	8.6721	8.8164
Action(HMDB)	Var	7.4668	7.9496	8.2552	8.4786	8.6547	8.7996
Action(UCF)	Var	7.4988	7.9804	8.2841	8.5051	8.6792	8.8231
iDT	Fv	7.5018	7.9839	8.2871	8.5081	8.6819	8.8252
MoSIFT	Fv	7.5162	7.9940	8.2947	8.5137	8.6861	8.8288
Scenes	Max	7.5295	8.0061	8.3052	8.5254	8.6980	8.8406
MFCC	Fv	7.4987	7.9764	8.2777	8.4993	8.6735	8.8171
SoundNet	Mean	7.4857	7.9671	8.2711	8.4937	8.6688	8.8128
VGGish	Mean	7.4990	7.9776	8.2796	8.5004	8.6751	8.8190

Table 4.22 presents the results for various recommender models in terms of Shannon entropy. The CER model using genre features obtained the highest Shannon entropy results, whereas TopPop model achieved the lowest Shannon entropy results. The scaled-CER presents a lower Shannon entropy in comparison to all the variants of the CER model.

4.4 RECOMMENDATION IN COLD-START SCENARIO

In this section, the experiment results of the CER and scaled-CER models, as well as the baselines in the item cold-start scenario are presented. The item cold-start scenario represents the case when preference data for the items in the scenario is not known at all during the training. The results obtained in this scenario are the main focus of this dissertation as they represent the ability of the recommendation models to mitigate the new item cold-start problem. The experiments conducted are essentially the same as the experiments performed in the item warm-start scenario, with the only difference being the scenario that is investigated.

4.4.1 Accuracy metrics

The recommendation performance in terms of accuracy metrics for MAP and NDCG are presented here.

Table 4.23. Results for the random, CER, and scaled-CER recommender models measured with respect to MAP in the item cold-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The best performance across the same metric is highlighted in bold.

Recommender models	Feature Agg.	MAP@5	MAP@10	MAP@15	MAP@20	MAP@25	MAP@30
non-personalised							
Random	-	0.0016	0.0016	0.0017	0.0018	0.0019	0.0020
CER							
Genres	-	0.0099	0.0104	0.0112	0.0119	0.0125	0.0130
Obj(IN)	Max	0.0159	0.0161	0.0170	0.0177	0.0183	0.0189
Action(IG)	Var	0.0146	0.0144	0.0150	0.0156	0.0162	0.0166
Action(KN)	Mad	0.0142	0.0142	0.0149	0.0155	0.0161	0.0165
Action(HMDB)	Var	0.0136	0.0134	0.0140	0.0146	0.0151	0.0155
Action(UCF)	Var	0.0143	0.0141	0.0147	0.0153	0.0159	0.0163
iDT	Fv	0.0098	0.0097	0.0102	0.0106	0.0110	0.0114
MoSIFT	Fv	0.0095	0.0093	0.0096	0.0100	0.0103	0.0106
Scene	Var	0.0152	0.0150	0.0156	0.0162	0.0167	0.0172
MFCC	Fv	0.0111	0.0112	0.0117	0.0122	0.0126	0.0129
SoundNet	Mean	0.0093	0.0095	0.0101	0.0106	0.0111	0.0114
VGGish	Mean	0.0134	0.0133	0.0139	0.0145	0.0150	0.0154
scaled-CER							
Genres	-	0.0113	0.0117	0.0125	0.0132	0.0138	0.0143
Obj(IN)	Max	**0.0178**	**0.0180**	**0.0188**	**0.0196**	**0.0203**	**0.0208**
Action(IG)	Var	0.0159	0.0155	0.0160	0.0166	0.0171	0.0176
Action(KN)	Mad	0.0154	0.0153	0.0160	0.0166	0.0171	0.0176
Action(HMDB)	Var	0.0137	0.0135	0.0140	0.0146	0.0151	0.0155

Table 4.23 continued from previous page

Action(UCF)	Var	0.0154	0.0152	0.0158	0.0165	0.0170	0.0174
iDT	Fv	0.0114	0.0112	0.0117	0.0122	0.0126	0.0130
MoSIFT	Fv	0.0105	0.0102	0.0105	0.0109	0.0113	0.0116
Scene	Var	0.0162	0.0159	0.0165	0.0171	0.0176	0.0181
MFCC	Fv	0.0116	0.0116	0.0121	0.0126	0.0130	0.0134
SoundNet	Mean	0.0097	0.0099	0.0104	0.0110	0.0114	0.0118
VGGish	Mean	0.0138	0.0138	0.0144	0.0150	0.0155	0.0160

Table 4.24. Results for the random, CER, and scaled-CER recommender models measured with respect to NDCG in the item cold-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result along the respective metric is marked in bold

Recommender models	Feature Agg.	NDCG@5	NDCG@10	NDCG@15	NDCG@20	NDCG@25	NDCG@30
non-personalised							
Random	-	0.0022	0.0035	0.0045	0.0054	0.0062	0.0070
CER							
Genres	-	0.0141	0.0213	0.0268	0.0314	0.0352	0.0386
Obj(IN)	Max	0.0223	0.0307	0.0367	0.0417	0.0459	0.0497
Action(IG)	Var	0.0201	0.0277	0.0331	0.0375	0.0413	0.0447
Action(KN)	Mad	0.0196	0.0271	0.0327	0.0372	0.0410	0.0443
Action(HMDB)	Var	0.0185	0.0258	0.0309	0.0353	0.0391	0.0424
Action(UCF)	Var	0.0197	0.0272	0.0327	0.0371	0.0410	0.0443
iDT	Fv	0.0137	0.0193	0.0234	0.0269	0.0299	0.0326
MoSIFT	Fv	0.0131	0.0179	0.0215	0.0246	0.0272	0.0297
Scene	Var	0.0210	0.0283	0.0334	0.0376	0.0412	0.0445
MFCC	Fv	0.0155	0.0215	0.0258	0.0293	0.0323	0.0350
SoundNet	Mean	0.0133	0.0188	0.0229	0.0263	0.0293	0.0319
VGGish	Mean	0.0187	0.0257	0.0307	0.0348	0.0383	0.0414
scaled-CER							
Genres	-	0.0159	0.0236	0.0294	0.0341	0.0382	0.0416
Obj(IN)	Max	**0.0247**	**0.0339**	**0.0404**	**0.0455**	**0.0500**	**0.0538**
Action(IG)	Var	0.0215	0.0290	0.0345	0.0388	0.0426	0.0459
Action(KN)	Mad	0.0212	0.0292	0.0350	0.0395	0.0433	0.0467
Action(HMDB)	Var	0.0188	0.0260	0.0311	0.0353	0.0390	0.0422
Action(UCF)	Var	0.0212	0.0291	0.0347	0.0393	0.0432	0.0466
iDT	Fv	0.0157	0.0219	0.0264	0.0301	0.0333	0.0362
MoSIFT	Fv	0.0143	0.0196	0.0234	0.0267	0.0295	0.0321
Scene	Var	0.0221	0.0296	0.0350	0.0394	0.0430	0.0463
MFCC	Fv	0.0163	0.0223	0.0266	0.0302	0.0333	0.0360
SoundNet	Mean	0.0138	0.0195	0.0238	0.0274	0.0306	0.0335
VGGish	Mean	0.0193	0.0267	0.0319	0.0361	0.0398	0.0430

Tables 4.23 and 4.24 show the results for various recommender models in terms of accuracy metrics. The accuracy metrics under study are MAP and NDCG. As presented in these tables, the scaled-CER model obtained the best overall performance using *Obj(IN)* features, whereas the random recommender model achieved the worst results.

4.4.2 Beyond-accuracy metrics

The cold-start evaluation using beyond-accuracy metrics is similar to the warm-start scenario. Notice that considering the weak performance of the random recommender model with regards to accuracy metrics, the results obtained by this model are not considered when determining the best results. However, they are still presented in grey for completeness. The recommendation quality in the cold-start scenario in terms of beyond-accuracy metrics are presented below:

Table 4.25. Results for the CER, and scaled-CER recommender models in terms of intra-list diversity metric in the item cold-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result across the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Div. IntraL @5**	**Div. IntraL @10**	**Div. IntraL @15**	**Div. IntraL @20**	**Div. IntraL @25**	**Div. IntraL @30**
non-personalised							
Random	-	0.5950	0.6693	0.6940	0.7064	0.7138	0.7187
CER							
Genres	-	0.2376	0.2926	0.3217	0.3434	0.3607	0.3742
Obj(IN)	Max	0.4830	0.5572	0.5876	0.6058	0.6185	0.6283
Action(IG)	Var	0.4740	0.5467	0.5757	0.5932	0.6057	0.6152
Action(KN)	Mad	0.4705	0.5432	0.5728	0.5905	0.6032	0.6130
Action(HMDB)	Var	0.4825	0.5573	0.5876	0.6056	0.6179	0.6271
Action(UCF)	Var	0.4884	0.5661	0.5975	0.6156	0.6282	0.6376
iDT	Fv	0.4976	0.5720	0.6015	0.6187	0.6303	0.6389
MoSIFT	Fv	0.5352	0.6128	**0.6441**	**0.6622**	**0.6742**	**0.6830**
Scene	Var	0.4949	0.5685	0.5978	0.6148	0.6266	0.6355
MFCC	Fv	**0.5450**	**0.6163**	0.6418	0.6556	0.6645	0.6709
SoundNet	Mean	0.5283	0.6012	0.6279	0.6425	0.6519	0.6587
VGGish	Mean	0.4962	0.5691	0.5968	0.6123	0.6226	0.6300
scaled-CER							
Genres	-	0.2243	0.2787	0.3087	0.3309	0.3479	0.3611
Obj(IN)	Max	0.4791	0.5519	0.5819	0.5999	0.6129	0.6228
Action(IG)	Var	0.4711	0.5450	0.5751	0.5932	0.6058	0.6155
Action(KN)	Mad	0.4650	0.5382	0.5681	0.5860	0.5989	0.6089
Action(HMDB)	Var	0.4781	0.5540	0.5850	0.6033	0.6159	0.6254
Action(UCF)	Var	0.4827	0.5608	0.5926	0.6112	0.6241	0.6338
iDT	Fv	0.4874	0.5619	0.5919	0.6094	0.6214	0.6304
MoSIFT	Fv	0.5308	0.6071	0.6379	0.6562	0.6683	0.6772
Scene	Var	0.4868	0.5619	0.5919	0.6097	0.6219	0.6315
MFCC	Fv	0.5417	0.6134	0.6390	0.6526	0.6613	0.6676
SoundNet	Mean	0.5218	0.5946	0.6216	0.6364	0.6460	0.6530
VGGish	Mean	0.4888	0.5620	0.5897	0.6052	0.6156	0.6234

Table 4.26. Results for the CER, and scaled-CER recommender models in terms of inter-list diversity metric in the item cold-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result across the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Div. InterL @5**	**Div. InterL @10**	**Div. InterL @15**	**Div. InterL @20**	**Div. InterL @25**	**Div. InterL @30**
non-personalised							
Random	-	0.9976	0.9952	0.9928	0.9904	0.9880	0.9855
CER							
Genres	-	0.9612	0.9471	0.9365	0.9278	0.9205	0.9141
Obj(IN)	Max	0.9790	0.9693	0.9618	0.9554	0.9496	0.9443
Action(IG)	Var	0.9797	0.9703	0.9631	0.9569	0.9513	0.9461
Action(KN)	Mad	0.9808	0.9719	0.9649	0.9589	0.9535	0.9485
Action(HMDB)	Var	0.9826	0.9745	0.9681	0.9624	0.9574	0.9526
Action(UCF)	Var	0.9800	0.9711	0.9641	0.9580	0.9526	0.9476
iDT	Fv	**0.9836**	**0.9756**	**0.9693**	0.9637	0.9587	0.9540
MoSIFT	Fv	0.9675	0.9564	0.9485	0.9422	0.9366	0.9315
Scene	Var	0.9784	0.9695	0.9626	0.9567	0.9514	0.9465
MFCC	Fv	0.9752	0.9662	0.9594	0.9537	0.9485	0.9438
SoundNet	Mean	0.9777	0.9695	0.9631	0.9577	0.9528	0.9483
VGGish	Mean	0.9823	0.9742	0.9678	0.9622	0.9571	0.9523
scaled-CER							
Genres	-	0.9675	0.9553	0.9456	0.9375	0.9308	0.9248
Obj(IN)	Max	0.9766	0.9666	0.9589	0.9523	0.9464	0.9410
Action(IG)	Var	0.9833	0.9753	0.9688	0.9632	0.9581	0.9533
Action(KN)	Mad	0.9812	0.9727	0.9661	0.9603	0.9552	0.9504
Action(HMDB)	Var	0.9831	**0.9756**	0.9696	**0.9643**	**0.9596**	**0.9551**
Action(UCF)	Var	0.9799	0.9715	0.9649	0.9592	0.9540	0.9492
iDT	Fv	0.9803	0.9718	0.9651	0.9592	0.9539	0.9490
MoSIFT	Fv	0.9641	0.9527	0.9447	0.9381	0.9323	0.9271
Scene	Var	0.9797	0.9714	0.9649	0.9593	0.9542	0.9495
MFCC	Fv	0.9724	0.9632	0.9563	0.9503	0.9450	0.9402
SoundNet	Mean	0.9748	0.9659	0.9591	0.9532	0.9479	0.9431
VGGish	Mean	0.9806	0.9724	0.9659	0.9602	0.9551	0.9504

Table 4.27. Results for the CER, and scaled-CER recommender models in terms of item coverage metric in the item cold-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result across the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	Feature Agg.	Item Cov.@5	Item Cov.@10	Item Cov.@15	Item Cov.@20	Item Cov.@25	Item Cov.@30
non-personalised							
Random	-	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
CER							
Genres	-	0.4577	0.5800	0.6382	0.6760	0.6919	0.7058
Obj(IN)	Max	0.7851	0.8760	0.9215	0.9446	0.9597	0.9683
Action(IG)	Var	0.8029	0.8822	0.9179	0.9375	0.9523	0.9621
Action(KN)	Mad	0.7792	0.8636	0.8994	0.9207	0.9359	0.9472
Action(HMDB)	Var	0.8306	0.8983	0.9322	0.9505	0.9626	0.9693
Action(UCF)	Var	0.7757	0.8654	0.9098	0.9329	0.9483	0.9598
iDT	Fv	0.8090	0.8925	0.9303	0.9540	0.9653	0.9722
MoSIFT	Fv	0.5704	0.6986	0.7655	0.8129	0.8466	0.8714
Scene	Var	0.7875	0.8736	0.9115	0.9350	0.9501	0.9610
MFCC	Fv	0.7545	0.8509	0.9010	0.9307	0.9474	0.9589
SoundNet	Mean	**0.9139**	**0.9661**	**0.9832**	**0.9899**	**0.9946**	**0.9963**
VGGish	Mean	0.8777	0.9375	0.9615	0.9743	0.9818	0.9855
scaled-CER							
Genres	-	0.5136	0.6391	0.6803	0.7011	0.7198	0.7442
Obj(IN)	Max	0.7332	0.8405	0.8899	0.9205	0.9415	0.9541
Action(IG)	Var	0.8577	0.9269	0.9519	0.9661	0.9743	0.9801
Action(KN)	Mad	0.7884	0.8697	0.9062	0.9253	0.9420	0.9530
Action(HMDB)	Var	0.8440	0.9183	0.9470	0.9621	0.9730	0.9790
Action(UCF)	Var	0.7642	0.8611	0.9041	0.9317	0.9500	0.9602
iDT	Fv	0.7342	0.8369	0.8888	0.9191	0.9374	0.9494
MoSIFT	Fv	0.4944	0.6173	0.7003	0.7497	0.7882	0.8167
Scene	Var	0.7827	0.8705	0.9105	0.9336	0.9494	0.9611
MFCC	Fv	0.6790	0.7949	0.8496	0.8883	0.9136	0.9308
SoundNet	Mean	0.8863	0.9470	0.9709	0.9832	0.9889	0.9921
VGGish	Mean	0.8656	0.9283	0.9561	0.9694	0.9773	0.9834

Tables 4.25-4.27 show the results for various recommender model in terms of beyond-accuracy metrics for list diversity and item coverage. The list diversity metrics under study are intra-list and inter-list diversity. The CER model using MFCC features achieved the highest intra-list diversity for cut-off values 5 and 10. For cut-off values 15, 20, 25 and 30 the highest intra-list diversity are obtained by MoSIFT features. In terms of inter-list diversity, the CER and scaled-CER models obtained similar

results. For high cut-off values, the scaled-CER model achieved the best results using *Action(HMDB)*. On the other hand, the CER model using iDT features achieved the highest results for low cut-off values. The highest results for item coverage have been achieved by the CER model using *soundNet* features. The scaled-CER model using genre features obtained the lowest results for intra-list diversity, while the CER model using genre features obtained the lowest results for inter-list diversity and item coverage.

Table 4.28. Results for the CER, and scaled-CER recommender models in terms of Shannon entropy metric in the item cold-start scenario. The performance of different video content features is evaluated using the CER and scaled-CER recommender models. The highest result across the same metric is marked in bold. Please note that the random recommender model performance is only shown for completeness given its poor performance in terms of accuracy metrics.

Recommender models	**Feature Agg.**	**Div. SE @5**	**Div. SE @10**	**Div. SE @15**	**Div. SE @20**	**Div. SE @25**	**Div. SE @30**
non-personalised							
Random	-	11.0136	11.0163	11.0172	11.0177	11.0179	11.0181
CER							
Genres	-	7.8355	8.3121	8.5849	8.7786	8.9342	9.0612
Obj(IN)	Max	8.7835	9.1258	9.3267	9.4688	9.5778	9.6660
Action(IG)	Var	8.9225	9.2410	9.4278	9.5603	9.6614	9.7437
Action(KN)	Mad	8.9116	9.2494	9.4466	9.5831	9.6871	9.7716
Action(HMDB)	Var	9.0500	9.3815	9.5716	9.7040	9.8047	9.8850
Action(UCF)	Var	8.8602	9.2095	9.4132	9.5550	9.6640	9.7520
iDT	Fv	9.0538	9.3805	9.5686	9.6999	9.8004	9.8801
MoSIFT	Fv	8.1329	8.6010	8.8799	9.0784	9.2285	9.3488
Scene	Var	8.8372	9.1921	9.3987	9.5438	9.6542	9.7426
MFCC	Fv	8.6740	9.0791	9.3151	9.4771	9.5995	9.6971
SoundNet	Mean	9.0182	9.3816	9.5854	9.7258	9.8318	9.9162
VGGish	Mean	**9.1818**	**9.4903**	**9.6664**	**9.7864**	9.8774	9.9498
scaled-CER							
Genres	-	8.0557	8.5078	8.7664	8.9529	9.1036	9.2270
Obj(IN)	Max	8.6569	9.0267	9.2425	9.3942	9.5110	9.6058
Action(IG)	Var	9.1571	9.4576	9.6278	9.7483	9.8398	9.9143
Action(KN)	Mad	8.9553	9.3004	9.4975	9.6342	9.7392	9.8232
Action(HMDB)	Var	9.1299	9.4644	9.6537	9.7830	**9.8808**	**9.9591**
Action(UCF)	Var	8.8733	9.2397	9.4483	9.5943	9.7039	9.7923
iDT	Fv	8.8032	9.1743	9.3868	9.5336	9.6448	9.7340
MoSIFT	Fv	7.9336	8.4393	8.7381	8.9500	9.1104	9.2389
Scene	Var	8.8951	9.2568	9.4651	9.6096	9.7195	9.8077
MFCC	Fv	8.4761	8.9175	9.1700	9.3441	9.4762	9.5810
SoundNet	Mean	8.8617	9.2421	9.4580	9.6071	9.7203	9.8108
VGGish	Mean	9.1073	9.4363	9.6207	9.7472	9.8432	9.9195

Table 4.28 shows the results for various recommender models in terms of Shannon entropy. As can be seen in the table, the CER and scaled-CER models obtained comparable results. The highest results have been achieved by the CER model using the *VGGish* features and the scaled-CER model using the *Action(HMDB)* features. The CER model using the genre features obtained the lowest results.

4.5 EVALUATION OF DIFFERENT FUSION METHODS

The evaluation of different fusion methods is performed in the item cold-start scenario. The goal of this experiment is to address the following problem: how to improve the recommendations of newly added videos further, given videos represented by multiple features, such as visual-appearance, audio, and action features. The experiment is based on the combination of the most accurate deep learning feature modalities, namely visual-appearance, audio, and action features, reported in Section 4.4. The scaled-CER model is used to evaluate the fusion methods described in Section 3.3.4. This model is chosen due to the outstanding overall performance presented in Section 4.4. The best single video content feature is used as a unimodal baseline to determine whether a fusion method really improves the recommendation quality.

Table 4.29. Results of different fusion methods in terms of MAP metric using the best visual-appearance, audio, and action features. The highest result along the respective metric is marked in bold.

Features	Feature Fusion	MAP@5	MAP@10	MAP@15	MAP@20	MAP@25	MAP@30
Obj(IN)	-	0.0178	0.0180	0.0188	0.0196	0.0203	0.0208
Obj(IN) + VGGish	concat	0.0215	0.0214	0.0223	0.0231	0.0238	0.0244
Obj(IN) + VGGish	sum	0.0144	0.0143	0.0148	0.0154	0.0159	0.0164
Obj(IN) + VGGish	max	0.0122	0.0121	0.0126	0.0130	0.0134	0.0137
Obj(IN) + VGGish + Action(IG)	concat	**0.0234**	**0.0232**	**0.0241**	**0.0249**	**0.0257**	**0.0263**
Obj(IN) + VGGish + Action(IG)	sum	0.0111	0.0113	0.0119	0.0125	0.0130	0.0134
Obj(IN) + VGGish + Action(IG)	max	0.0086	0.0087	0.0091	0.0096	0.0099	0.0102

Table 4.30. Results of different fusion methods in terms of NDCG metric using the best visual-appearance, audio, and action features. The highest result along the respective metric is marked in bold.

Features	Feature Fusion	NDCG@5	NDCG@10	NDCG@15	NDCG@20	NDCG@25	NDCG@30
Obj(IN)	-	0.0247	0.0339	0.0404	0.0455	0.0500	0.0538
Obj(IN) + VGGish	concat	0.0295	0.0395	0.0464	0.0521	0.0568	0.0609
Obj(IN) + VGGish	sum	0.0201	0.0276	0.0328	0.0371	0.0407	0.0439
Obj(IN) + VGGish	max	0.0171	0.0228	0.0268	0.0300	0.0329	0.0354
Obj(IN) + VGGish + Action(IG)	concat	**0.0318**	**0.0423**	**0.0496**	**0.0554**	**0.0603**	**0.0646**
Obj(IN) + VGGish + Action(IG)	sum	0.0156	0.0226	0.0277	0.0318	0.0354	0.0386
Obj(IN) + VGGish + Action(IG)	max	0.0123	0.0173	0.0210	0.0241	0.0268	0.0292

Tables 4.29 and 4.30 show the results of different feature fusion methods with respect to accuracy metrics. The accuracy metrics under study are MAP and NDCG. The concatenation method obtained the best overall performance when combining *Obj(IN)*, *VGGish* and *Action(IG)* features, whilst the maximum method that combines these same features obtained the worst results.

Table 4.31. Results of different fusion methods in terms of intra-list diversity metric using the best visual-appearance, audio, and action features. The best result along the respective metric is marked in bold.

Features	Feature Fusion	Div. IntraL @5	Div. IntraL @10	Div. IntraL @15	Div. IntraL @20	Div. IntraL @25	Div. IntraL @30
Obj(IN)	-	0.4791	0.5519	0.5819	0.5999	0.6129	0.6228
Obj(IN) + VGGish	concat	0.4594	0.5345	0.5654	0.5837	0.5967	0.6068
Obj(IN) + VGGish	sum	0.5012	0.5727	0.6004	0.6163	0.6269	0.6349
Obj(IN) + VGGish	max	**0.5227**	**0.5940**	**0.6197**	0.6338	0.6430	0.6497
Obj(IN) + VGGish + Action(IG)	concat	0.4534	0.5272	0.5578	0.5766	0.5900	0.6006
Obj(IN) + VGGish + Action(IG)	sum	0.4869	0.5605	0.5900	0.6072	0.6190	0.6277
Obj(IN) + VGGish + Action(IG)	max	0.5137	0.5884	0.6177	**0.6350**	**0.6465**	**0.6547**

Table 4.32. Results of different fusion methods in terms of inter-list diversity metric using the best visual-appearance, audio, and action features. The best result along the respective metric is marked in bold.

Features	Feature Fusion	Div. InterL @5	Div. InterL @10	Div. InterL @15	Div. InterL @20	Div. InterL @25	Div. InterL @30
Obj(IN)	-	0.9766	0.9666	0.9589	0.9523	0.9464	0.9410
Obj(IN) + VGGish	concat	0.9767	0.9681	0.9616	0.9559	0.9507	0.9459
Obj(IN) + VGGish	sum	0.9764	0.9664	0.9588	0.9523	0.9464	0.9411
Obj(IN) + VGGish	max	0.9745	0.9657	0.9587	0.9526	0.9472	0.9422
Obj(IN) + VGGish + Action(IG)	concat	0.9777	0.9694	0.9630	0.9574	0.9523	0.9475
Obj(IN) + VGGish + Action(IG)	sum	**0.9791**	**0.9706**	**0.9638**	**0.9580**	**0.9526**	**0.9476**
Obj(IN) + VGGish + Action(IG)	max	0.9768	0.9675	0.9603	0.9540	0.9484	0.9433

Table 4.33. Results of different fusion methods in terms of item coverage metric using the best visual-appearance, audio, and action features. The best result along the respective metric is marked in bold.

Features	Feature Fusion	Item Cov. @5	Item Cov. @10	Item Cov. @15	Item Cov. @20	Item Cov. @25	Item Cov. @30
Obj(IN)	-	0.7332	0.8405	0.8899	0.9205	0.9415	0.9541
Obj(IN) + VGGish	concat	0.8540	0.9262	0.9552	0.9716	0.9800	0.9866
Obj(IN) + VGGish	sum	0.7661	0.8614	0.9056	0.9326	0.9470	0.9583
Obj(IN) + VGGish	max	0.7293	0.8343	0.8858	0.9176	0.9374	0.9513
Obj(IN) + VGGish + Action(IG)	concat	**0.8655**	**0.9336**	**0.9597**	**0.9740**	**0.9831**	**0.9880**
Obj(IN) + VGGish + Action(IG)	sum	0.8101	0.8955	0.9355	0.9561	0.9687	0.9773
Obj(IN) + VGGish + Action(IG)	max	0.6967	0.8096	0.8674	0.9032	0.9247	0.9414

Tables 4.31-4.33 show the results of different feature fusion methods with respect to beyond-accuracy metrics for list diversity and item coverage. The list diversity metrics under study are intra-list and inter-list diversity. For intra-list diversity, the max method combining *Obj(IN)* and *VGGish* features obtained the highest results utilising recommendation lists of length equal to 5, 10, and 15. However, for recommendation lists of length equal to 15, 25, and 30, the highest results are achieved by the max method when combining *Obj(IN)*, *VGGish* and *Action(IN)* features. The lowest results for intra-list diversity is obtained by the concat method when combining *Obj(IN)*, *VGGish*, and *Action(IG)* features. For inter-list diversity, the sum method combining *Obj(IN)*, *VGGish* and *Action(IG)* features achieved

the highest results, while the max method when fusing *Obj(IN)* and *VGGish* obtained the lowest results for cut-off values 5, 10 and 15. The sum method fusing the same set of features obtained the lowest results for cut-off value 25, followed by the baseline *Obj(IN)* features. For cut-off value 30, the baseline *Obj(IN)* obtained the lowest inter-list diversity result followed by the sum method fusing *Obj(IN)* and *VGGish* features. In terms of item coverage, the concat method when fusing *Obj(IN)*, *VGGish* and *Action(IG)* features, achieved the highest results, whereas the max method when fusing these same three types of features obtained the lowest results.

Table 4.34. Results of different fusion methods in terms of Shannon entropy (SE) using the best visual-appearance, audio, and action features. The best result along the respective metric is marked in bold.

Features	**Feature Fusion**	**Div. @5 SE**	**Div. @10 SE**	**Div. @15 SE**	**Div. @20 SE**	**Div. @25 SE**	**Div. @30 SE**
Obj(IN)	-	8.6569	9.0267	9.2425	9.3942	9.5110	9.6058
Obj(IN) + VGGish	concat	8.9131	9.2799	9.4868	9.6278	9.7337	9.8176
Obj(IN) + VGGish	sum	8.7458	9.1144	9.3273	9.4749	9.5860	9.6754
Obj(IN) + VGGish	max	8.6516	9.0670	9.3001	9.4591	9.5805	9.6776
Obj(IN) + VGGish + Action(IG)	concat	**8.9552**	**9.3153**	**9.5163**	**9.6545**	**9.7577**	**9.8403**
Obj(IN) + VGGish + Action(IG)	sum	8.9218	9.2806	9.4826	9.6219	9.7265	9.8093
Obj(IN) + VGGish + Action(IG)	max	8.6260	9.0289	9.2581	9.4175	9.5384	9.6366

Tables 4.34 show the results of the various feature fusion methods in terms of Shannon entropy. Overall, the concat method combining *Obj(IN)*, *VGGish* and *Action(IG)* features achieved the highest results for Shannon entropy, while the lowest results are attained by the unimodal baseline followed by the max method combining *Obj(IN)*, *VGGish* and *Action(IG)* features.

4.6 ABLATION STUDY

In this experiment, the cumulative effect of each video content feature to the overall recommendation quality is explored. The main goal is to empirically assess the importance of using a diverse range of video content features while taking full advantage of the available features in the item cold-start scenario. Thus, the experiment is performed by combining all the video content features explored in this research work. Two experiments were conducted, including one with only non-textual features and another with non-textual features and genre features.

The study is only based on the scaled-CER model using the concatenation fusion method, due to the outstanding overall performance shown in the previous experiments. The recommendation quality is measured in terms of MAP and item coverage, in order to gain an understanding of the extent to what the video recommendation system is able to explore the catalogue with high precision.

Table 4.35. Ablation study of the importance of all non-textual video content features explored in this work in the overall recommendation quality in terms of MAP and item coverage. The best result across the respective metric is highlighted in bold. Prev. denotes the features used in the previous row.

Features	**MAP@5**	**Item Coverage @5**	**MAP@15**	**Item Coverage @15**	**MAP@30**	**Item Coverage @30**
Obj(IN)	0.0178	0.7332	0.0188	0.8899	0.0208	0.9541
Prev. + *VGGish*	0.0215	0.8540	0.0223	0.9552	0.0244	0.9866
Prev. + *Action(IG)*	0.0234	0.8655	0.0241	0.9597	0.0263	0.9880
Prev. + Scene	0.0245	0.8456	0.0252	0.9503	0.0275	0.9827
Prev. + *Action(KN)*	0.0251	0.8487	0.0257	0.9471	0.0280	0.9817
Prev. + *Action(UCF)*	0.0248	0.8711	0.0251	**0.9622**	0.0273	0.9880
Prev. + *Action(HMDB)*	0.0252	0.8611	0.0256	0.9553	0.0280	0.9843
Prev. + *SoundNet*	0.0251	0.8569	0.0257	0.9566	0.0281	0.9866
Prev. + MFCC	0.0254	**0.8750**	0.0259	0.9617	0.0282	**0.9882**
Prev. + iDT	0.0262	0.8631	0.0267	0.9561	0.0291	0.9855
Prev. + MoSIFT	**0.0267**	0.8339	**0.0273**	0.9454	**0.0297**	0.9799

Table 4.36. Ablation study of the importance of all non-textual video content features and genre features in the overall recommendation quality in terms of MAP and item coverage. The best result across the respective metric is highlighted in bold. Prev. denotes the features used in the previous row. The box around the (Prev. + iDT) row highlights the highest MAP results for cut-off values 5 and 15 obtained in this work.

Features	MAP@5	Item Coverage @5	MAP@15	Item Coverage @15	MAP@30	Item Coverage @30
Obj(IN)	0.0178	0.7332	0.0188	0.8899	0.0208	0.9541
Prev. + *VGGish*	0.0215	0.8540	0.0223	0.9552	0.0244	0.9866
Prev. + *Action(IG)*	0.0234	**0.8655**	0.0241	**0.9597**	0.0263	**0.9880**
Prev. + Genres	0.0291	0.7180	0.0297	0.8806	0.0325	0.9493
Prev. + Scene	0.0298	0.7677	0.0303	0.9142	0.0331	0.9682
Prev. + *Action(KN)*	0.0309	0.7898	0.0312	0.9269	0.0339	0.9770
Prev. + *Action(UCF)*	0.0333	0.7631	0.0335	0.9082	0.0364	0.9656
Prev. + *Action(HMDB)*	0.0341	0.7539	0.0344	0.9032	0.0374	0.9643
Prev. + *SoundNet*	0.0343	0.7981	0.0345	0.9296	0.0374	0.9756
Prev. + MFCC	0.0362	0.7609	**0.0366**	0.9153	**0.0397**	0.9717
Prev. + iDT	**0.0365**	0.7975	**0.0366**	0.9301	0.0396	0.9774
Prev. + MoSIFT	0.0360	0.8141	0.0361	0.9404	0.0391	0.9807

4.7 CHAPTER SUMMARY

The results of the video recommendation system implemented in this research work were presented in this chapter. Various video-level descriptors have been visualised to check whether they capture meaningful semantics. The performance of the various feature aggregation methods is measured in both the 2-dimensional UMAP feature space as well as in the original feature dimensional space. All the video content features, feature aggregation methods, and fusion techniques were compared against one another. The best performing ones were selected for subsequent experiments. It is found that the scaled-CER model outperforms the CER model with respect to accuracy metrics in both the warm-start and cold-start scenarios. Furthermore, when combining the *Obj(IN)*, *VGGish* and *Action(IG)* features using the concatenation, sum and max fusion strategies, the concatenation method obtained the best MAP, NDCG and item coverage results. Finally, the results of the ablation study demonstrated that apart from one hand-crafted feature (MoSIFT features), all the other types of features are necessary to achieve the highest performance observed in this research work in terms of accuracy. In Chapter 5, the results presented in this chapter are carefully analysed and discussed.

CHAPTER 5 DISCUSSION

5.1 CHAPTER OVERVIEW

The results of the experiments presented in Chapter 4 are analysed and discussed in detail in this chapter. Each video recommendation model and video content feature used in the recommendation framework is addressed to fully understand its strengths and weaknesses in each scenario.

5.2 FEATURE ANALYSIS

Figures 4.1 - 4.11 visually highlight the distances between movie trailers given their genres. The correctness of the clustering patterns are clearly visible using this visual representation. It is obvious that 2-UMAP components correctly represent any movie trailer, since correct clustering of different genres occurs. Although the clustering of genres shown in these figures may not be effective for tasks like genre classification, it is worth remembering that a movie is usually represented by multiple genres and not by a single genre. From Figures 4.1 - 4.11, it can be seen that some movies in the same sequel tend to be close to each other in UMAP space. This suggests that even though the movie trailers are represented by non-textual features that are not semantic at first glance, the genre and sequel that describe these video-level representations are semantically associated. For example, in Figure 4.1, the video-level descriptors, which represent some movie trailers in the *Batman* sequel, are close to each other. These descriptors are semantically related because they reflect *Batman* movies. They also belong to the same genre, namely action genre. The clustering of these *Batman* movies indicate that some *Batman* movie trailers have very similar visual aesthetics. In addition, it can be seen, in Figure 4.9 to 4.11, that this remarkable clustering pattern is also present in the hand-crafted action and sound distributions for other movie sequels. This result is understandable because a movie sequel is a continuation narrative of the already existing movie. These findings clearly indicate that the non-textual features used in this research work capture different meaningful semantics before the recommendation stage.

Movie trailers are also represented by to the type of video feature. For this reason, movie trailers in the same sequel can be very close to each other in one UMAP feature space, showing movie trailers in terms of their visual modality, but can be further apart in another visualisation, showing movie trailers in terms of their aural modality. This observation is evident in the *Madagascar* film series. As can be seen in Figure 4.3 and 4.7, the *Madagascar* film series are very similar to each other in terms of their visual content, but differ in terms of their audio content. Furthermore, it is interesting to observe in Figures 4.3 to 4.5 that a distinctive global structure for each action-centric deep learning feature reduced to 2-UMAP components. This indicates that, even though some 3D-CNNs used to extract action-centric features have been pre-trained on the IG-65M or Kinetics datasets, each network generates action-centric features that focus on different actions learnt by fine-tuning on the Kinetics, UCF-101, or HMDB-51 datasets.

Beyond the visual observations, the quality of the feature aggregation methods used in this research work is assessed in terms of Bhattacharyya distance. This distance metric is measured in 2-dimensional UMAP space and the original F-dimensional feature space. These results are reported in Tables 4.1 to 4.16. It is noted that for certain video features, the best performing feature aggregation method is the same when the Bhattacharyya distance is measured in 2-dimensional UMAP space and when it is measured in the original feature space. As shown in Tables 4.1 and 4.2 for the *Obj(IN)* features, the most effective feature aggregation is the maximum method for both 2-dimensional UMAP space and original 2048-dimensional feature space. The highest performing feature aggregation method for the *Action(KN)* features and *Action(HMDB)* features is the variance method as reported in Table 4.7 and 4.8, and Table 4.11 and 4.12, respectively. For the *soundNet* features, as seen in Table 4.15 and 4.16, the median feature aggregation method outperforms the other aggregation methods. All these results suggest that 2-UMAP components can, to some extent, capture the variation in the data properly. This observation is consistent with the visualisation of the features referred to in Figure 4.1, 4.4, 4.6 and 4.8. However, it should be recalled that non-local distances are not well preserved when only 2-UMAP components are used. As a consequence, for certain video features when UMAP feature reduction is not applied, the best feature aggregation method is not the same as when UMAP feature reduction is applied. This is seen in the results obtained for the scene, *Action(IG)*, *Action(UCF)* and *VGGish* features. Nevertheless, for visual features the feature aggregation methods, which measure the spread of a distribution, achieve better performance compared to feature aggregation methods that are a measure of location. In addition, it should be pointed out that, for all video features, the in-sequel and out-of-sequel similarities and differences within each feature aggregation method is retained since

the in-sequel-mean-distance is lower than the out-of-sequel-mean-distance.

5.3 PERFORMANCE ANALYSIS: WARM-START SCENARIO

5.3.1 Accuracy metrics

As can be seen in Table 4.17 and 4.18, the CER model obtained the highest results in terms of MAP and NDCG at all cut-off values when using the *Action(UCF)* features, while the lowest results are obtained using genre, scene, and *soundNet* features. It can also be observed that the CER model exhibits similar performance across all types of video content features. This outcome is similar to [14] and suggests that in the item warm-start scenario, the interactions collected for the items are very important in order to obtain outstanding results, and that the additional video content features help predictive performance of items with very few interactions. The importance of items prior ratings is clearer when looking at the performance of the TopPop and scaled-CER models. The TopPop model obtained better performance than the random model, by recommending only the top popular items to users. The scaled-CER model obtained improved results over the CER model. The scaled-CER model obtained the best overall performance compared to any CER model variant, by using the matrix scaling technique presented in Section 3.3.3.1. Different from the CER model, the scaled-CER model achieved the best results in terms of MAP using the *Obj(IN)* features. In terms of NDCG, the scaled-CER model achieves the best performance using the iDT features.

The performance gains of the scaled-CER model using *Obj(IN)* features, over the CER model using *Action(UCF)* features in terms of MAP@5 and MAP@30 are 37.7% and 24.2%, respectively. In terms of NDCG@5 and NDCG@30, the scaled-CER model using iDT features outperforms the CER model using *Action(UCF)* by 26.2% and 17.4%, respectively. These outcomes clearly show that in the item warm-start scenario, the item content descriptor is not as important as the ratings of the items. Moreover, the results clearly illustrate the effectiveness of the matrix scaling technique, where the scaled-CER recommender model presents the best capability to generate recommendation lists that contain relevant items at the top positions. In addition, similar to the CER model, the scaled-CER model presents similar results along with the different types of video content features. However, the difference between the results is insignificant after proper scaling.

Nevertheless, by a closer inspection of the scaled-CER model variants, it can be seen that the state-of-the-art iDT feature vectors is the best baseline video content feature and outperforms almost all deep learning features, with the only exception being the *Obj(IN)*, *Action(HMDB)*, and *Action(UCF)*

features at certain cut-off values. The *VGGish* and *soundNet* features outperform the hand-crafted MFCC features. However, as pointed out above, the difference between the results of any type of video content feature is not significant. In addition, it is worth pointing out that for each model as the cut-off value increases, MAP and NDCG results also increase. This is as expected, because the likelihood of one of them to be a true label increases with the number of items being recommended. This means that more correctly predicted videos are obtained. Furthermore, it is also noted that most deep learning video content features, when used by the hybrid recommender models, achieved the best performance using a feature aggregation variant that is consistent with the quantitative feature analysis outcome presented in Section 5.2. This indicates that video recommendation systems, which use these feature aggregation methods, can simplify the task of selection of the best performing statistical feature aggregation method, by using movie sequels and the Bhattacharyya distance.

5.3.2 Beyond accuracy metrics

From the results summarised in Table 4.19 to 4.22, it can be seen that the random recommender model, used for sanity checks, achieved the highest results for all diversity metrics as well as for the item coverage metric. This outcome was expected since a random recommender model, as the name suggests, recommends items at random which therefore leads to high diversity and absolute coverage of the item catalogue. However, as seen in Tables 4.17 and 4.18, high diversity comes at an unacceptable cost of accuracy, since this model obtained results close to 0 with regards to accuracy metrics. Thus, while the generated recommendation lists are greatly diversified, these lists are not of interest to the user, since they do not contain relevant items. For this reason, this section does not analyse the beyond-accuracy results of random recommender models further, as it achieved an unacceptable low performance in terms of accuracy metrics.

When observing the intra-list diversity results presented in Table 4.19, it can be seen that scaled-CER presents a lower intra-list diversity compared to CER and TopPop. This outcome was expected because it is known that an inherent trade-off exists between accuracy and beyond-accuracy metrics [181]. The matrix scaling technique used by the scaled-CER model brings great improvements in terms of accuracy metrics at the cost of intra-list diversity. One possible solution to improve this outcome in terms of intra-list diversity would be to implement a re-ranking technique. This technique would select a video with relevance score above a certain threshold to be part of the recommendation list by considering both its score and its genre-dissimilarity. The video's score and genre would be compared to the videos already in the list if the list is not empty. Another possible solution would be to optimise

the model with a many-objective evolutionary algorithm [182, 183] which optimise the accuracy of the model as well as its intra-list diversity simultaneously.

Moreover, by a closer inspection of the video content features, the scaled-CER using genre features obtained the lowest results compared to other scaled-CER variants. This outcome was expected because genre features are used to calculate this metric. The results suggest that the model is generating recommendation lists with many videos of the same genre. In addition, similar to accuracy metric results, the intra-list diversity results improve with an increase in size of the recommendation list.

In terms of inter-list diversity, it can be seen in Table 4.20 that the TopPop model achieved the lowest results. This suggests that the recommendation lists being generated for the users have a high number of items which are similar to one another across lists. This outcome was expected because recommendations provided by TopPop model are not personalised for each user and, as the name suggests, the model only recommends popular items. The inter-list diversity results of the scaled-CER model is lower than the CER model results. The CER model using genre features achieved the best results. However, the scaled-CER model obtained values greater than 88% and 90% for high cut-off and small cut-off values, respectively. Moreover, as discussed in Section 5.3.1, the scaled-CER model achieved a noticeable improvement over the CER model with regards to accuracy metrics. This means that the scaled-CER model recommends a high number of items that are relevant to the users, without compromising a lot of its diversification capabilities. In addition, unlike the intra-list diversity results, as the recommendation list increases the inter-list diversity decreases for all other models. This was expected since recommendation lists with a cut-off value equal to the total number of items in the catalogue should contain all items across lists, leading an inter-list diversity value of 0.

The results for item coverage is presented in Table 4.21. As expected, TopPop model obtained the lowest results. This outcome is due to the model recommending only popular items to users. It is interesting to note that for all cut-off values, except 5, the highest results for item coverage were obtained by the scaled-CER model. This outcome is in line with the main purpose of the scaling factor, which is to increase the sensitivity of the model to rare items. However, the increase in item coverage is accompanied by less diverse recommendation lists. In addition, it is equally important to determine the magnitude to which the models are trying to recommend different items within a certain coverage value. This is performed by calculating the Shannon entropy. It can be observed in Table 4.22 that TopPop model provides recommendations that are easy to guess. For CER and scaled-CER models, it

is interesting to see that their recommendations are hard to guess, in contrast to TopPop model. These results confirm the outcome observed with regards to inter-list diversity, where TopPop model obtained the lowest performance. The CER model using genre features obtained the highest results. However, this variant of the CER model, obtained the worst performance in terms of accuracy metrics compared to other CER variants.

Furthermore, it is interesting to note that the scaled-CER model using *Obj(IN)* features did not obtain the lowest results in terms of beyond-accuracy metrics since this model obtained the best performance in terms of accuracy metrics. In addition, similar to accuracy metric results, the beyond-accuracy results for the CER and scaled-CER models are similar for the various video content features along with the respective cut-off values.

5.4 PERFORMANCE ANALYSIS: COLD-START SCENARIO

5.4.1 Accuracy metrics

In terms of accuracy metrics, it is observed in the Table 4.23 and 4.24 that in contrast to the item warm-start scenario, the CER and scaled-CER models in the cold-start scenario exhibit results that are more varied across different types of video content features. This suggests that these models are relying more on the features to generate recommendations, and that each type of video content feature discriminates the user preferences differently. Similar to the item warm-start scenario, the scaled-CER model achieved the highest results with regards to all the accuracy metrics and a noticeable improvement over the CER model is observed. This outcome also shows the effectiveness of the matrix scaling technique in the item cold-start scenario, which is able to improve the performance of different types of features. It suggests that the collaborative information learnt with the item popularity sensitivity adjustment along with video content features is very important to recommend cold items with high precision. It can be observed that the scaled-CER, using MFCC features, is the best baseline with regards to the cut-off value 5 across the MAP and NDCG metrics. However, with regards to the other cut-off values, the best baseline is the genre features. The best overall performance is obtained by *Obj(IN)* features, which outperform the MFCC features by 53.4% and 51.5% in terms of MAP@5 and NDCG@5, respectively. In terms of MAP@15, MAP@30, NDCG@15, and NDCG@30, the *Obj(IN)* features outperform genre features by 50.4%, 45.4%, 37.4%, and 29.3%, respectively. These results indicate that the scaled-CER model, using *Obj(IN)* features in its content descriptor, provides considerable better recommendations that are placed at the top of the recommendation list, compared to the genre and hand-crafted features. In addition, the model is able to recommend more relevant

items as the recommendation list gets longer.

It is interesting to note that action-centric deep learning features present noticeably better performance, compared to the hand-crafted MoSIFT features and state-of-the-art hand-crafted iDT features, across all accuracy metrics. The best action features, with regards to MAP across different cut-off values, is *Action(IG)* followed by *Action(KN)*, *Action(UCF)*, and *Action(HMDB)*. However, *Action(KN)* followed by *Action(UCF)* is slightly better than *Action(IG)* with regards to NDCG for cut-off values greater than 5. Nevertheless, *Action(IG)* provides highly relevant recommendations at the top of the recommendation list, given its performance for the top-5 cut-off experiments. This performance is highly desirable especially when many users do not scan the entire recommendation list, and when a service provider sends personalised newsletter video recommendations to their users to alleviate the new item cold-start problem. That being said, these results are very promising since it shows that the motion information captured by deep learning features lead to better recommendations compared to the hand-crafted iDT and MoSIFT features in terms of accuracy metrics.

On the other hand, the performance obtained by the hand-crafted MFCC features is better than one acoustic-centric deep learning feature, namely *soundNet* features. However, scaled-CER model using *VGGish* features outperforms MFCC features. This confirms the success of deep learning features in the video recommendation context in terms of accuracy metrics. In addition, it should also be noted that, except for *soundNet* features, all the deep learning features explored in this work outperforms genre features, which emphasises the importance of using non-textual features extracted from videos to improve cold item recommendations.

Lastly, as expected the non-personalised model, namely random recommender, is the worst model among all accuracy metrics. It obtained an extremely low performance, which suggests that the recommendations do not match the preferences of the users. This is understandable since the recommendations provided by the random model are not personalised, and the items are randomly recommended.

5.4.2 Beyond accuracy metrics

Similar to the item warm-start scenario, it can be seen from the results presented in Table 4.25 to 4.28, that the random model obtained the highest values for all diversity metrics and item coverage, as expected. However, this model provides poor recommendations to users according to its performance

in terms of accuracy metrics. In addition, recommending items that match the preferences of the users in the item warm-start scenario, while greatly improving the recommendation of new items, is perhaps the most important objective of a designer. Moreover, maximising the beyond-accuracy performance of a certain recommender system is only promising when a minimum acceptable performance, with regards to accuracy metrics, has been obtained. For this reason, the beyond-accuracy results of the random model are not analysed further in this section.

Overall, the CER model obtained the highest results for almost all four metrics with the exception being the InterL@20, InterL@25, InterL@30, SE@25, and SE@30. However, these results come at the cost of accuracy. The lowest results are obtained using the genre features. In contrast to the item warm-start scenario, in the item cold-start scenario, the scaling factor of the scaled-CER model does not lead to the highest results with regards to item coverage. The scaling factor only increases the item coverage of the genre, *Action(IG)*, and *Action(HMDB)* features. It is interesting to see that this increase comes with an increase in inter-list diversity and Shannon entropy, but with a slight decrease in intra-list diversity. This means that the number of items that are recommended equally often increased; however, the number of recommendations of the same genre also increased. In addition, it is important to note that the action-centric deep learning features present a better item coverage with recommendations that are harder to guess in comparison to the hand-crafted iDT and MoSIFT features, without compromising recommendation accuracy. A similar outcome is observed for the deep learning audio features compared to MFCC features. Furthermore, it is also worth mentioning that the best visual-appearance, action, and audio features in terms of MAP and NDCG, namely *Obj(IN)*, *Action(IG)*, and *VGGish* features, obtained item coverage results in the range of 0.7332 to 0.8656 for the smallest cut-off value, and 0.9541 to 0.9834 for the highest cut-off value. *These results are promising and suggest that the scaled-CER model is able to recommend more than 95% of cold items and these items are highly relevant to users.*

5.5 PERFORMANCE ANALYSIS: FUSION METHODS

From Table 4.29 and 4.30, it can be seen that the sum and max fusion methods are not able to outperform the *Obj(IN)* features baseline with regards to MAP and NDCG metrics across all cut-off values. An interesting observation is that the sum of *Obj(IN)* and *VGGish* features leads to better recommendation accuracy compared to *VGGish* features alone (Table 4.23 and 4.24). A noticeable drop in performance is observed when the *Action(IG)* features are combined with *Obj(IN)* and *VGGish* features using the two aforementioned fusion methods. This outcome suggests that the shared latent

space created by the sum and max fusion methods are not suitable for the scaled-CER model since the complementary video information encoded in the *Action(IG)*, *Obj(IN)*, and *VGGish* features are not learnt properly. On the other hand, the concat fusion method outperforms the baseline, and the sum and max fusion methods, with regards to all accuracy metrics across all cut-off values. The concat of *Obj(IN)* and *VGGish* features significantly improve upon the *Obj(IN)* performance by 20.7% and 19.4% for the lowest cut-off value (5), along the MAP and NDCG metrics, respectively. For the highest cut-off value (30) the increase over the baseline is 17.3% for MAP, and 13.1% for NDCG. In addition, different from the sum and max fusion methods, the concat of *Action(IG)*, *Obj(IN)* and *VGGish* features presents a significant positive effect in the overall recommendation performance. More precisely, the recommendation accuracy in terms of MAP@5 and MAP@30 increased by 8.8%, and 7.7% over the concat of *Obj(IN)*, and *VGGish* features, respectively. In terms of NDCG@5 and NDCG@30, the recommendation accuracy increased by 7.7% and 6.0%, respectively. These results suggest that the shared latent space created by the concat fusion method is more discriminative, thus leading to better recommendation accuracy. *VGGish* features complement the recommendation accuracy. In addition, *Action(IG)* features combined with *Obj(IN)* and *VGGish* features create video representations that are highly predictive of user preferences. This means that the content present in the videos are better described. As a result, enhanced recommendations are provided.

In terms of beyond-accuracy metrics, it can be observed in Table 4.31 that the outstanding performance achieved by the concat of *Obj(IN)*, *VGGish* and *Action(IG)* features with regards to accuracy metrics comes with a decrease in intra-list diversity. The concat fusion of these three features did not obtain intra-list diversity results higher than the baseline, but the difference is between 3.6% and 5.6%.

Surprisingly, noticeable performance improvements can be seen in Table 4.32 to 4.34 for the concat of *Obj(IN)*, *VGGish*, and *Action(IG)* features with regards to inter-list diversity, item coverage, and Shannon entropy. These improvements do not come at the expense of recommendation accuracy. The results are promising and indicate that the complementariness of *Obj(IN)*, *VGGish*, and *Action(IG)* features considers more items on the catalogue. These items are given a better chance of being recommended, leading to recommendations that are more equally spread out throughout all cold items.

5.6 ABLATION STUDY

From Table 4.35, it is interesting to observe that scene features bring an increase in recommendation accuracy, but with a drop in item coverage when combined with the three main features, namely *Obj(IN)*, *VGGish*, and *Action(IG)*. On the other hand, *Action(UCF)* features provide an increase in item coverage, but with a decrease in recommendation accuracy when combined with the three main features, scene and *Action(KN)* features. These outcomes demonstrate the trade-off between MAP and item coverage, as seen in the previous experiments. Similar to *VGGish* and *Action(IG)* features, MFCC features provide an improvement in MAP and item coverage. This demonstrates the complementary role of MFCC features when they are combined with deep learning features and especially deep learning audio features. Which means that, although *VGGish* and *soundNet* features represent the audio from the video, this audio has some information that is not captured by these features, including deep learning visual appearance and action features. The MFCC features capture this additional information, which results in better discriminative video descriptors, as seen by the improvements in MAP and item coverage. These descriptors are highly predictive of the user preferences leading to a wide range of relevant video recommendations.

Moreover, a significant improvement in recommendation accuracy is observed when all non-textual features are combined, compared to only using a single feature, or only using the three main features, as performed in the previous experiment. The high performance comes with the inherent trade-off between accuracy and beyond-accuracy metrics. There is a noticeable negative effect in item coverage, especially when MoSIFT features are combined. In addition, it is compelling to see that fusing hand-crafted features with deep learning features results in more precise recommendations by exploiting the complementary information available in the hand-crafted feature set. This means that the discriminative power of the deep learning features is strengthened with the valuable information captured by the hand-crafted features.

Furthermore, looking at Table 4.36, the overall outcome is quite different when genre features are combined with the various non-textual features. It can be observed that genre features provide a big improvement in recommendation accuracy in contrast to scene features, when considering Table 4.35. The genre features complement the recommendation accuracy of the non-textual features effectively. This improvement comes with a decrease in item coverage which is worse than the unimodal *Obj(IN))* features. Apart from MoSIFT features, a drop in recommendation accuracy is not noted when combining the other features. This outcome implies that genre features probably

remove ambiguity from the non-textual features, which in turn lead to an improvement in MAP, while sacrificing item coverage. This was expected because each non-textual content feature vector was fused with genres, which describe high-level concepts of a movie, thus creating a more discriminative semantically meaningful content descriptor. When the iDT features are combined with all the other video content features (Prev. + iDT), MAP results are obtained that are slightly above the combination of all the types of video features (Prev. + MoSIFT) for cut-off values 5 and 15. *These MAP results are the highest achieved for the respective cut-off values in this research work*. The addition of MoSIFT features provides a noticeable increase in item coverage for all cut-off values. This means that MoSIFT features provide a good balance between MAP and item coverage when combined with the other features explored in this work.

Nevertheless, the combination of all the various video content features provides recommendations that are very precise. However, about 18.59% of the total number of cold items are never recommended to a user for the lowest cut-off value. For cut-off values 15 and 30, more than 94% of cold items are recommended to users meaning that the descriptors obtained from the combination of all the features are highly predictive of the user preferences, leading to a wide range of relevant video recommendations. This shows the strong correlation between the deep learning features, hand-crafted features, and genre features in the overall recommendation quality.

CHAPTER 6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

This research has presented the development of a video recommendation framework that uses multiple video content features to solve the new item cold-start problem. Various deep learning features, extracted from the multi-modal, extremely high dimensional information from videos are used to enhance the quality of recommendations. The features capture visual-appearance, audio, and motion information from the media contained in the videos. Their distribution are visually and quantitatively analysed to determine if they are semantically meaningful before the recommendation task. It is found that using only two UMAP components is not enough to find the best performing feature aggregation method before the recommendation task, in terms of Bhattacharyya distance. However, two UMAP components are useful from a visualisation point of view, since they can be used to distinguish between movies in-sequel and out-of-sequel.

Moreover, a comparison between different video recommendation models is performed using various video content features. This research work proposes an improvement for the CER recommender model using a known matrix scaling technique. The proposed improved model is named the scaled-CER model. This model is sensitive to rare items by scaling the collaborative information before training. The scaled-CER model obtained the best recommendation accuracy in the item warm-start and cold-start scenarios. An improvement in item coverage in the item warm-start scenario upon the CER model is also observed. Furthermore, different fusion methods are investigated to effectively combine the features before training the model to improve the recommendation quality in terms of accuracy and beyond-accuracy metrics. The best fusion method was found to be the concat method.

From the results presented in Chapter 4, the research questions posed in this study are answered as follows:

- **Research Question 1:**
  "Can the combination of the visual-appearance, audio, and action-related features, which capture the visual, aural and motion information contained in the videos, provide better video recommendation, with respect to accuracy and beyond accuracy metrics, than the visual-appearance and audio features, which only capture visual and aural information?"

  As the results of the experiments suggested, the fusion of visual-appearance, audio, and action features provide more accurate personalised video recommendations to users when compared to the fusion of only visual and audio features. Apart from intra-list diversity measure, an improvement upon the fusion of visual and audio features is also observed for all the other beyond-accuracy measures. This means that the recommender is able to explore the catalogue of videos better while, generating relevant video recommendations. This also results in new videos being given a better chance of being recommended, which leads to recommendations that are more equally spread out throughout all new videos.

- **Research Question 2:**
  "What motion information from videos is the most predictive of users' video preferences in new item cold-start scenarios?"

  The motion information captured by action-centric deep learning features, extracted with 3D-CNNs, is better than hand-crafted action features. They lead to better recommendation accuracy and item coverage, with more balanced recommendations. For this reason, it can be concluded that the success of 3D-CNN features on tasks like action recognition and video classification also occur in the video recommendation context.

- **Research Question 3:**
  "To what extent can the combination of hand-crafted features, deep learning features, and textual features maximise the video recommendation performance?"

  The results of the ablation study demonstrated that, apart from one hand-crafted fea-

ture (MoSIFT features), all types of features, namely textual, hand-crafted, and deep learning features are necessary to achieve the highest performance observed in this study in terms of accuracy metrics. The results also showed that textual features, namely the genre features are the most important features in the overall result, since the largest improvement is observed when they are combined with the other features. However, the high precision comes with a decrease in item coverage, but this coverage is still well above the unimodal feature baseline (*Obj(IN)* features).

6.2 FUTURE WORK

As future work, it will be worth evaluating the features used in this research study in terms of user quality perception. This could assist in the better tuning of video recommendation system in industrial applications and in the creation of high quality recommendation explanations to increase trust in the system. In addition, it would be worth investigating the correlation between multimedia features, electronic programming guide (EPG) information and the user feedback gathered from the use of the remote control. The end goal would be to enhance existing recommender systems in this domain to provide more significant TV program recommendations to users that lead to a decrease in the use of the remote control, while improving their experience.

REFERENCES

[1] Y. Xu, T. Price, F. Monrose, and J. Frahm, "Caught Red-Handed: Toward Practical Video-Based Subsequences Matching in the Presence of Real-World Transformations," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1397–1406.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[3] C. A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, pp. 1–19, 2016.

[4] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[5] P. Lops, M. Degemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, 2011, ch. 3, pp. 73–105.

[6] M. Liu, X. Xie, and H. Zhou, "Content-based Video Relevance Prediction Challenge: Data, Protocol, and Baseline," *arXiv preprint arXiv:1806.00737*, 2018.

[7] Y. Deldjoo, M. F. Dacrema, M. G. Constantin, H. Eghbal-zadeh, S. Cereda, M. Schedl, B. Ionescu, and P. Cremonesi, "Movie genome: alleviating new item cold start in movie

recommendation," *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 291–343, 2019.

[8] J. Yuan, W. Shalaby, M. Korayem, D. Lin, K. AlJadda, and J. Luo, "Solving cold-start problem in large-scale recommendation engines: A deep learning approach," in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 1901–1910.

[9] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, pp. 1–25, 2019.

[10] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative filtering and deep learning based recommendation system for cold start items," *Expert Systems with Applications*, vol. 69, pp. 29–39, 2017.

[11] N. M. Ali and B. A. Novikov, "Big data: analytical solutions, research challenges and trends," *Proceedings of the RAS Institute for System Programming*, vol. 32, no. 1, 2020.

[12] Y. Kumar, A. Sharma, A. Khaund, A. Kumar, P. Kumaraguru, R. R. Shah, and R. Zimmermann, "Icebreaker: Solving cold start problem for video recommendation engines," in *2018 IEEE International Symposium on Multimedia (ISM)*.

[13] P. Wang, Y. Jiang, C. Xu, and X. Xie, "Overview of Content-Based Click-Through Rate Prediction Challenge for Video Recommendation," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2593–2596.

[14] X. Du, H. Yin, L. Chen, Y. Wang, Y. Yang, and X. Zhou, "Personalized Video Recommendation Using Rich Contents from Videos," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 492–505, 2020.

[15] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[16] D. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles, "What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7366–7375.

[17] J. C. Stroud, D. A. Ross, C. Sun, J. Deng, and R. Sukthankar, "D3D: Distilled 3D Networks for Video Action Recognition," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 614–623.

[18] V. Adeli, E. Fazl-Ersi, and A. Harati, "A component-based video content representation for action recognition," *Image and Vision Computing*, vol. 90, p. 103805, 2019.

[19] J. Wehrmann and R. C. Barros, "Movie genre classification: A multi-label approach based on convolutions through time," *Applied Soft Computing*, vol. 61, pp. 973–982, 2017.

[20] F. Álvarez, F. Sánchez, G. Hernández-Peñaloza, D. Jiménez, J. M. Menéndez, and G. Cisneros, "On the influence of low-level visual features in film classification," *PLOS ONE*, vol. 14, no. 2, pp. 1–29, 2019.

[21] Y. Deldjoo, M. Schedl, P. Cremonesi, and G. Pasi, "Recommender Systems Leveraging Multimedia Content," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–38, 2020.

[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[23] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The YouTube Video Recommendation System," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, 2010, pp. 293–296.

[24] M. Soares and P. Viana, "Tuning metadata for better movie content-based recommendation systems," *Multimedia Tools and Applications*, vol. 74, no. 17, pp. 7015–7036, 2015.

[25] M. H. Rimaz, M. Elahi, F. B. Moghaddam, C. Trattner, R. Hosseini, and M. Tkalcic, "Exploring the Power of Visual Features for the Recommendation of Movies," in *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, 2019, pp. 303–308.

[26] Y. Deldjoo, M. Elahi, P. Cremonesi, F. Garzotto, P. Piazzolla, and M. Quadrana, "Content-Based Video Recommendation System Based on Stylistic Visual Features," *Journal on Data Semantics*, vol. 5, no. 2, pp. 99–113, 2016.

[27] J. Lee and S. Abu-El-Haija, "Large-Scale Content-Only Video Recommendation," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 987–995.

[28] D. Brezeale and D. J. Cook, "Automatic Video Classification: A Survey of the Literature," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 416–430, 2008.

[29] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A Survey on Visual Content-Based Video Indexing and Retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, 2011.

[30] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, "Use What You Have: Video Retrieval Using Representations From Collaborative Experts," in *30th British Machine Vision Conference 2019*, 2019, pp. 279–293.

[31] Z. Zeng, W. Liang, H. Li, and S. Zhang, "A Novel Video Classification Method Based on Hybrid Generative/Discriminative Models," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, 2008, pp. 705–713.

[32] A. G. Hauptmann, R. Jin, and T. D. Ng, "Multi-Modal Information Retrieval from Broadcast Video Using OCR and Speech Recognition," in *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, 2002, pp. 160–161.

[33] N. Dimitrova, L. Agnihotri, and G. Wei, "Video classification based on HMM using text and faces," in *2000 10th European Signal Processing Conference*, 2000, pp. 1–4.

[34] N. Yamamoto, J. Ogata, and Y. Ariki, "Topic segmentation and retrieval system for lecture videos based on spontaneous speech recognition," in *8th European Conference on Speech Communication and Technology*, 2003, pp. 961–964.

[35] N. Radha, "Video retrieval using speech and text in video," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 2, 2016, pp. 1–6.

[36] W.-H. Lin and A. Hauptmann, "News Video Classification Using SVM-Based Multimodal Classifiers and Combination Strategies," in *Proceedings of the Tenth ACM International Conference on Multimedia*, 2002, pp. 323–326.

[37] L. R. Rabiner and R. W. Schafer, "Introduction to Digital Speech Processing," *Foundations and Trends® in Signal Processing*, vol. 1, no. 1-2, pp. 1–194, 2007.

[38] T. Giannakopoulos, D. Kosmopoulos, A. Aristidou, and S. Theodoridis, "Violence Content Classification Using Audio Features," in *Advances in Artificial Intelligence, 4th Helenic Conference on AI*, 2006, pp. 502–507.

[39] Z. Liu, J. Huang, and Y. Wang, "Classification TV programs based on audio information using hidden Markov model," in *1998 IEEE Second Workshop on Multimedia Signal Processing*, 1998, pp. 27–32.

[40] E. Wold, T. Blum, D. Keislar, and J. Wheaten, "Content-based classification, search, and retrieval of audio," *IEEE MultiMedia*, vol. 3, no. 3, pp. 27–36, 1996.

[41] Z. Liu, Y. Wang, and T. Chen, "Audio Feature Extraction and Analysis for Scene Segmentation and Classification," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 20, no. 1-2, pp. 61–79, 1998.

[42] J. Saunders, “Real-time discrimination of broadcast speech/music,” in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, 1996, pp. 993–996.

[43] Zhu Liu, Jincheng Huang, Yao Wang, and Tsuhan Chen, “Audio feature extraction and analysis for scene classification,” in *Proceedings of First Signal Processing Society Workshop on Multimedia Signal Processing*, 1997, pp. 343–348.

[44] T. Zhang and C.-C. J. Kuo, “Audio content analysis for online audiovisual data segmentation and classification,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, pp. 441–457, 2001.

[45] M. Rouvier, G. Linarès, and D. Matrouf, “Robust audio-based classification of video genre,” in *Tenth Annual Conference of the International Speech Communication Association*, 2009, pp. 1159–1162.

[46] U. Srinivasan, S. Pfeiffer, S. Nepal, M. Lee, L. Gu, and S. Barrass, “A Survey of MPEG-1 Audio, Video and Semantic Analysis Techniques,” *Multimedia Tools and Applications*, vol. 27, no. 1, pp. 105–141, 2005.

[47] G. Tzanetakis and F. Cook, “Sound analysis using MPEG compressed audio,” in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, 2000, pp. II761–II764.

[48] S. Venugopal, K. R. Ramakrishnan, S. H. Srinivas, and N. Balakrishnan, “Audio scene analysis and scene change detection in the MPEG compressed domain,” in *1999 IEEE Third Workshop on Multimedia Signal Processing*, 1999, pp. 191–196.

[49] K. Subashini, S. Palanivel, and V. Ramalingam, “Audio-Video based Classification using SVM and AANN,” *International Journal of Computer Applications*, vol. 44, pp. 33–39, 2012.

[50] R. W. Lienhart, “Comparison of automatic shot boundary detection algorithms,” in *Storage and Retrieval for Image and Video Databases VII*, vol. 3656, 1998, pp. 290–301.

[51] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *arXiv preprint arXiv:1706.06905*, 2017.

[52] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso, "Temporal Video Segmentation to Scenes Using High-Level Audiovisual Features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 8, pp. 1163–1177, 2011.

[53] L. Canini, S. Benini, and R. Leonardi, "Affective Recommendation of Movies Based on Selected Connotative Features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 4, pp. 636–647, 2013.

[54] K. van de Sande, T. Gevers, and C. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[55] J. Sivic and A. Zisserman, "Video Google: Efficient Visual Search of Videos," in *Toward Category-Level Object Recognition*, 2006, ch. 10, pp. 127–144.

[56] S. Biswas and R. V. Babu, "H.264 compressed video classification using Histogram of Oriented Motion Vectors (HOMV)," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 2040–2044.

[57] M. Svanera, M. Savardi, A. Signoroni, A. B. Kovács, and S. Benini, "Who is the Film's Director? Authorship Recognition Based on Shot Features," *IEEE MultiMedia*, vol. 26, no. 4, pp. 43–54, 2019.

[58] Yining Deng and B. S. Manjunath, "Content-based search of video using color, texture, and motion," in *Proceedings of International Conference on Image Processing*, 1997, pp. 534–537.

[59] Z. Rasheed, Y. Sheikh, and M. Shah, "On the use of computable features for film classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 1, pp. 52–64, 2005.

[60] R. Visser, N. Sebe, and E. Bakker, "Object Recognition for Video Retrieval," in *International conference on image and video retrieval*, 2002, pp. 262–270.

[61] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Studying Aesthetics in Photographic Images Using a Computational Approach," in *9th European Conference on Computer Vision*, 2006, pp. 288–301.

[62] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[63] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *9th European Conference on Computer Vision*, 2006, pp. 404–417.

[64] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, pp. 886–893.

[65] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[66] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "MARS: Motion-Augmented RGB Stream for Action Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7874–7883.

[67] P. Scovanner, S. Ali, and M. Shah, "A 3-Dimensional Sift Descriptor and Its Application to Action Recognition," in *Proceedings of the 15th ACM International Conference on Multimedia*, 2007, pp. 357–360.

[68] A. Klaeser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients," in *Proceedings of the British Machine Vision Conference*, 2008, pp. 99.1–99.10.

[69] M. Chen and A. Hauptmann, "MoSIFT: Recognizing Human Actions in Surveillance Videos," Carnegie Mellon University, Technical report, 2009.

[70] I. Laptev, "On Space-Time Interest Points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, 2005.

[71] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.

[72] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Proceedings of the Alvey Vision Conference*, 1988, pp. 23.1–23.6.

[73] N. Dalal, B. Triggs, and C. Schmid, "Human Detection Using Oriented Histograms of Flow and Appearance," in *9th European Conference on Computer Vision*, 2006, pp. 428–441.

[74] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3169–3176.

[75] M. Suresha, S. Kuppa, and D. S. Raghukumar, "A study on deep learning spatiotemporal models and feature extraction techniques for video understanding," *International Journal of Multimedia Information Retrieval*, vol. 9, no. 2, pp. 81–101, 2020.

[76] Ba Tu Truong and C. Dorai, "Automatic genre identification for content-based video categorization," in *Proceedings 15th International Conference on Pattern Recognition*, 2000, pp. 230–233.

[77] R. S. Jadon, S. Chaudhury, and K. K. Biswas, "Generic Video Classification: An Evolutionary Learning Based Fuzzy Theoretic Approach," in *Proceedings of the Third Indian Conference on Computer Vision, Graphics & Image Processing*, 2002, pp. 1–13.

[78] H. Sundaram and Shih-Fu Chang, "Video scene segmentation using video and audio features," in *2000 IEEE International Conference on Multimedia and Expo*, 2000, pp. 1145–1148.

[79] L. Shen, R. Hong, and Y. Hao, "Advance on large scale near-duplicate video retrieval," *Frontiers of Computer Science*, vol. 14, no. 5, pp. 14–38, 2020.

[80] Joe Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4694–4702.

[81] M. A. Jalal, W. Aftab, R. K. Moore, and L. Mihaylova, "Dual Stream Spatio-Temporal Motion Fusion With Self-Attention For Action Recognition," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–7.

[82] D. Wang, J. Yang, and Y. Zhou, "Human action recognition based on multi-mode spatial-temporal feature fusion," in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–7.

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

[84] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 Million Image Database for Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.

[85] G. Kalliatakis, S. Ehsan, A. Leonardis, M. Fasli, and K. Mcdonald-Maier, "Exploring Object-Centric and Scene-Centric CNN Features and Their Complementarity for Human Rights Violations Recognition in Images," *IEEE Access*, vol. 7, pp. 10 045–10 056, 2019.

[86] A. Diba, M. Fayyaz, V. Sharma, M. Mahdi Arzani, R. Yousefzadeh, J. Gall, and L. Van Gool, "Spatio-Temporal Channel Correlation Networks For Action Classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 284–299.

[87] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-Scale Video Classification with Convolutional Neural Networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[88] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[89] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, 2014, pp. 568–576.

[90] C. Li, J. Cao, Z. Huang, L. Zhu, and H. T. Shen, "Leveraging Weak Semantic Relevance for Complex Video Event Classification," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 3667–3676.

[91] C. Li, Z. Huang, Y. Yang, J. Cao, X. Sun, and H. T. Shen, "Hierarchical Latent Concept Discovery for Video Event Detection," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2149–2162, 2017.

[92] L. Yu, Z. Huang, J. Cao, and H. T. Shen, "Scalable Video Event Retrieval by Visual State Binary Embedding," *IEEE Transactions on Multimedia*, vol. 18, no. 8, pp. 1590–1603, 2016.

[93] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[94] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[95] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *2011 International Conference on Computer Vision*, 2011, pp. 2556–2563.

[96] Y. Aytar, C. Vondrick, and A. Torralba, "SoundNet: Learning Sound Representations from Unlabeled Video," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 892–900.

[97] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 131–135.

[98] F. Xiao, Y. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual SlowFast Networks for Video Recognition," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6201–6210.

[99] V. Sharma, M. Tapaswi, and R. Stiefelhagen, "Deep multimodal feature encoding for video ordering," *arXiv preprint arXiv:2004.02205*, 2020.

[100] H. Zettl, "Essentials of Applied Media Aesthetics," in *Media Computing: Computational Media Aesthetics*, 2002, ch. 2, pp. 11–38.

[101] Y. Hou, T. Xiao, S. Zhang, X. Jiang, X. Li, X. Hu, J. Han, L. Guo, L. S. Miller, R. Neupert, and T. Liu, "Predicting Movie Trailer Viewer's "Like/Dislike" via Learned Shot Editing Patterns," *IEEE Transactions on Affective Computing*, vol. 7, no. 1, pp. 29–44, 2016.

[102] M. Elahi, Y. Deldjoo, F. Bakhshandegan Moghaddam, L. Cella, S. Cereda, and P. Cremonesi, "Exploring the Semantic Gap for Movie Recommendations," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 326–330.

[103] L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, and A. G. Hauptmann, "Bridging the Ultimate Semantic Gap: A Semantic Search Engine for Internet Videos," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 27–34.

[104] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8m: A Large-Scale Video Classification Benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[105] C. Ma, M. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," *Signal Processing: Image Communication*,

vol. 71, pp. 76–87, 2019.

[106] N. Aafaq, A. Mian, W. Liu, S. Z. Gilani, and M. Shah, "Video Description: A Survey of Methods, Datasets, and Evaluation Metrics," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–37, 2019.

[107] Y. Deldjoo and M. Schedl, "Retrieving Relevant and Diverse Movie Clips Using the MFVCD-7K Multifaceted Video Clip Dataset," in *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, 2019, pp. 1–4.

[108] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.

[109] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311.

[110] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," in *11th European Conference on Computer Vision*, 2010, pp. 143–156.

[111] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential Deep Learning for Human Action Recognition," in *Human Behavior Unterstanding - Second International Workshop*, 2011, pp. 29–39.

[112] R. Aly, R. Arandjelović, K. Chatfield, M. Douze, B. Fernando, Z. Harchaoui, K. Mcguiness, N. O'Connor, D. Oneata, O. M. Parkhi, D. Potapov, J. Revaud, C. Schmid, J. Schwenninger, D. Scott, T. Tuytelaars, J. Verbeek, H. Wang, and A. Zisserman, "The AXES submissions at TRECVID 2013," in *2013 TREC Video Retrieval Evaluation*, 2013, pp. 1–13.

[113] V. Sydorov, M. Sakurada, and C. H. Lampert, "Deep Fisher Kernels - End to End Learning of the Fisher Kernel GMM Parameters," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1402–1409.

[114] R. Lin, J. Xiao, and J. Fan, "NeXtVLAD: An Efficient Neural Network to Aggregate Frame-level Features for Large-scale Video Classification," in *Computer Vision - European Conference on Computer Vision (ECCV) 2018 Workshops*, 2018, pp. 206–218.

[115] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.

[116] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus Late Fusion in Semantic Video Analysis," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, 2005, pp. 399–402.

[117] N. Holzenberger, S. Palaskar, P. Madhyastha, F. Metze, and R. Arora, "Learning from Multiview Correlations in Open-domain Videos," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 8628–8632.

[118] C. Guo and D. Wu, "Canonical Correlation Analysis (CCA) Based Multi-View Learning: An Overview," *arXiv preprint arXiv:1907.01693*, 2019.

[119] J. Perez-Rua, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie, "MFAS: Multimodal Fusion Architecture Search," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6959–6968.

[120] F. Isinkaye, Y. Folajimi, and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015.

[121] M. S. Ibrahim and C. I. Saidu, "Recommender Systems: Algorithms, Evaluation and Limitations," in *Journal of Advances in Mathematics and Computer Science*, 2020, pp. 121–137.

[122] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative Filtering Recommender Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, 2007, ch. 9, pp. 291–324.

[123] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-Based Collaborative Filtering Recommendation Algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295.

[124] D. Pazzani, Michael J.and Billsus, "Content-Based Recommendation Systems," in *The Adaptive Web: Methods and Strategies of Web Personalization*, 2007, ch. 10, pp. 325–341.

[125] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[126] M. F. Dacrema, S. Boglio, P. Cremonesi, and D. Jannach, "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research," *ACM Transactions on Information Systems*, vol. 39, no. 2, pp. 1–49, 2021.

[127] C. C. Aggarwal, "Evaluating Recommender Systems," in *Recommender Systems: The Textbook*, 2016, ch. 7, pp. 225–254.

[128] P. Cremonesi, F. Garzotto, and R. Turrin, "Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 2, no. 2, pp. 1–41, 2012.

[129] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[130] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 95–116, 2018.

[131] W. Krichene and S. Rendle, "On sampled metrics for item recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1748–1757.

[132] D. Jannach and M. Jugovac, "Measuring the Business Value of Recommender Systems," *ACM Transactions on Management Information Systems (TMIS)*, vol. 10, no. 4, pp. 1–23, 2019.

[133] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in *Recommender Systems Handbook*, 2011, ch. 1, pp. 1–35.

[134] C. Musto, F. Narducci, P. Lops, G. Semeraro, M. de Gemmis, M. Barbieri, J. Korst, V. Pronk, and R. Clout, "Enhanced Semantic TV-Show Representation for Personalized Electronic Program Guides," in *Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization*, 2012, pp. 188–199.

[135] C. C. Aggarwal, "Content-Based Recommender Systems," in *Recommender Systems: The Textbook*, 2016, ch. 4, pp. 139–166.

[136] Y. Deldjoo, M. G. Constantin, B. Ionescu, M. Schedl, and P. Cremonesi, "MMTF-14K: A Multifaceted Movie Trailer Feature Dataset for Recommendation and Retrieval," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 139–166.

[137] X. Chen, R. Zhao, S. Ma, D. Liu, and Z.-J. Zha, "Content-Based Video Relevance Prediction with Second-Order Relevance and Attention Modeling," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 2018–2022.

[138] J. Dong, X. Li, C. Xu, G. Yang, and X. Wang, "Feature Re-Learning with Data Augmentation for Content-Based Video Recommendation," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 2058–2062.

[139] R. J. R. Filho, J. Wehrmann, and R. C. Barros, "Leveraging deep visual features for content-based movie recommender systems," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 604–611.

[140] T. Mei, B. Yang, X.-S. Hua, L. Yang, S.-Q. Yang, and S. Li, "VideoReach: An Online Video Recommendation System," in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007, pp. 767–768.

[141] B. Yang, T. Mei, X.-S. Hua, L. Yang, S.-Q. Yang, and M. Li, "Online Video Recommendation Based on Multimodal Fusion and Relevance Feedback," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, 2007, pp. 73–80.

[142] T. Mei, B. Yang, X.-S. Hua, and S. Li, "Contextual Video Recommendation by Multimodal Relevance and User Feedback," *ACM Transactions on Information Systems (TOIS)*, vol. 29, no. 2, pp. 1–24, 2011.

[143] Y. Bhalgat, "FusedLSTM: Fusing frame-level and video-level features for Content-based Video Relevance Prediction," *arXiv preprint arXiv:1810.00136*, 2018.

[144] E. Hoffer and N. Ailon, "Deep Metric Learning Using Triplet Network," in *Similarity-Based Pattern Recognition - Third International Workshop*, 2015, pp. 84–92.

[145] I. Pilászy and D. Tikk, "Recommending New Movies: Even a Few Ratings Are More Valuable than Metadata," in *Proceedings of the Third ACM Conference on Recommender Systems*, 2009, pp. 93–100.

[146] Y. Deldjoo, M. Elahi, and P. Cremonesi, "Using Visual Features and Latent Factors for Movie Recommendation," in *Proceedings of the 3rd Workshop on New Trends in Content-Based Recommender Systems co-located with ACM Conference on Recommender Systems*, 2016, pp. 15–18.

[147] Y. Deldjoo, M. Elahi, M. Quadrana, and P. Cremonesi, "Using visual features based on MPEG-7 and deep learning for movie recommendation," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 4, pp. 207–219, 2018.

[148] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model," in *27th Annual Conference on Neural Information Processing Systems*, 2013, pp. 2121–2129.

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

[150] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[151] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-Scale Weakly-Supervised Pre-Training for Video Action Recognition," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 038–12 047.

[152] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[153] E. J. Keogh and A. Mueen, "Curse of Dimensionality," in *Encyclopedia of Machine Learning and Data Mining*, 2017, pp. 314–315.

[154] Y. Deldjoo, P. Cremonesi, M. Schedl, and M. Quadrana, "The Effect of Different Video Summarization Models on the Quality of Video Recommendation Based on Low-Level Visual Features," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, 2017, pp. 1–6.

[155] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," *arXiv preprint arXiv:1510.07493*, 2015.

[156] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, "Selective convolutional descriptor aggregation for fine-grained image retrieval," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2868–2881, 2017.

[157] H. Jegou and A. Zisserman, "Triangulation embedding and democratic aggregation for image search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 3310–3317.

[158] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *arXiv preprint arXiv:1703.09507*, 2017.

[159] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[160] J. Ma, G. Li, M. Zhong, X. Zhao, L. Zhu, and X. Li, "LGA: latent genre aware micro-video recommendation on social media," *Multimedia Tools and Applications*, vol. 77, no. 3, pp. 2991–3008, 2018.

[161] A. N. Nikolakopoulos, V. Kalantzis, E. Gallopoulos, and J. Garofalakis, "EigenRec: generalizing PureSVD for effective and efficient top-N recommendations," *Knowledge and Information Systems*, vol. 58, no. 1, pp. 59–81, 2019.

[162] E. Frolov and I. Oseledets, "HybridSVD: When Collaborative Information is Not Enough," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 331–339.

[163] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 101–109.

[164] A. Miech, D. Zhukov, J. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2630–2640.

[165] L. McInnes, J. Healy, N. Saul, and L. Großberger, "UMAP: Uniform Manifold Approximation and Projection," *The Journal of Open Source Software*, vol. 3, no. 29, pp. 861–914, 2018.

[166] L. V. D. Maaten and G. E. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[167] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[168] E. Choi and C. Lee, "Feature extraction based on the Bhattacharyya distance," *Pattern Recognition*, vol. 36, no. 8, pp. 1703–1709, 2003.

[169] A. Ansari and M. Mohammed, "Content based Video Retrieval Systems - Methods, Techniques, Trends and Challenges," *International Journal of Computer Applications*, vol. 112, no. 7, pp. 13–22, 2015.

[170] S. Bekhet, M. Hassaballah, A. Ahmed, and A. H. Ahmed, "Video Similarity Measurement and Search," in *Recent Advances in Computer Vision: Theories and Applications*, 2019, ch. 4, pp. 85–112.

[171] G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," in *Recommender Systems Handbook*, 2011, ch. 8, pp. 257–297.

[172] A.-M. Tousch, "How robust is MovieLens? A dataset analysis for recommender systems," *arXiv preprint arXiv:1909.12799*, 2019.

[173] I. Matveeva, C. Burges, T. Burkard, A. Laucius, and L. Wong, "High Accuracy Retrieval with Multiple Nested Ranker," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006, pp. 437–444.

[174] T. Silveira, M. Zhang, X. Lin, Y. Liu, and S. Ma, "How good your recommender system is? A survey on evaluations in recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 813–831, 2019.

[175] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.

[176] D. Kluver and J. A. Konstan, "Evaluating Recommender Behavior for New Users," in *Proceedings of the 8th ACM Conference on Recommender Systems*, 2014, pp. 121–128.

[177] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of Recommender Algorithms on Top-n Recommendation Tasks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, 2010, pp. 39–46.

[178] Y. Tasker, *Action and Adventure Films*, 2006.

[179] ——, *The action and adventure cinema*. Routledge, 2004.

[180] Y. Yin, R. R. Shah, and R. Zimmermann, "Learning and Fusing Multimodal Deep Features for Acoustic Scene Categorization," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 1892–1900.

[181] G. Adomavicius and Y. Kwon, "Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, 2012.

[182] X. Cai, Z. Hu, P. Zhao, W. Zhang, and J. Chen, "A hybrid recommendation system with many-objective evolutionary algorithm," *Expert Systems with Applications*, vol. 159, p. 113648, 2020.

[183] Z. Hu, Y. Lan, Z. Zhang, and X. Cai, "A many-objective particle swarm optimization algorithm based on multiple criteria for hybrid recommendation system." *KSII Transactions on Internet & Information Systems*, vol. 15, no. 2, 2021.