Deep visual genre-aware descriptors for movie recommendation

Athanasios Dritsas

Delft University of Technology

Deep visual genre-aware descriptors for movie recommendation

By

Athanasios Dritsas

in partial fulfilment of the requirements for the degree of

Master of Science
in Computer Science

at the Delft University of Technology,
to be defended publicly on Thursday January 17, 2019 at 1:00 PM.

Supervisor: M. Larson
Thesis committee: A. Bozzon, TU Delft
M. Gutierrez Granada, RTL Nederland

TU Delft Delft University of Technology

The work in this thesis was supported by RTL Nederland. Their cooperation is hereby gratefully acknowledged.

Acknowledgements

I would like to express my gratitude to my supervisor Dr. Martha Larson for her valuable guidance and extraordinary support throughout this project.
Furthermore, I would like to thank Mateo Gutierrez Granada and Dr. Yashar Deldjoo for the useful comments, remarks and continued support during this master thesis.
Finally, I would like to thank my family and friends for their full and unconditional support.

Contents

1

Introduction

Movies are perhaps the form of art that entertains and influences people more than any other art. People typically spend an important part of their free time watching movies in order to get pleasure, experiencing situations that challenge their emotions and their logic [1]. Nowadays, people have the possibility to watch movies either traditionally on linear TV or on demand. The on-demand approach offers a more personalized experience, allowing the viewers to watch their favorite movie at the time and the device of their preference.

In the last years, the popularity of video-on-demand services has been constantly increasing, especially for the young audiences who are more adept at using new technologies. Through those platforms, the viewers have access to a huge volume of movies at any moment that makes the viewing decision for most of them a very challenging task. Surveys show that users spend on average 1 hour per day to discover their desired content and in 60% of the cases the ease in finding interesting items to watch is very important when they decide on which service they will subscribe to [2].

Recommender systems are employed by video-on-demand providers to address the former challenge. Three recommendation models are commonly used in the community of recommender systems: the collaborative filtering (CF), the content-based filtering (CBF) and the hybrid approach that combines properties from the two former approaches [3].

CF models are trying to predict the preference of a user by exploiting the feedback of the other users in the system. In a CF system, an item is recommended to a user if similar users have rated it positively. Similar users are typically considered the ones that have given similar ratings for the same set of movies [4].

On the other hand, CBF systems are trying to predict the user preferences by analyzing the content of the items. Critical issue in this approach is the use of a content representation that could reflect what a user likes or not in a movie. The more accurate the movie representation is, the more capable is the CBF system in recommending to the users items which are similar in content with the items they have enjoyed in the past. Available resources for generating a movie representation are the textual metadata defined by the movie experts (genre, directors, actors, reviews, summaries etc) or users (tags) and the actual movie content with respect to its visual, audio and textual modalities [5].

One very popular attribute used for movie representation is genre due to its capability to reflect important key elements of a movie and its high availability since it is provided by film producers to facilitate the movie's marketing [6]. The genre of a movie is associated with movie aspects like the setting, the period, the characters, the plot, the techniques, the audio etc that help the user to make an informed guess about the content of the movie and decide if she would be interested or not in it. The information related to the genre of a movie is quite

rich and it is impossible to express with a textual label. Movies of the same genre can differ in the degree that some genre characteristics are involved. Some action movies are characterized by scenes of gun violence whereas others are dominated by car chases scenes. Moreover, it is quite common nowadays for movies to include elements from many genres. Sometimes their presence is so strong that the genre of the movie is provided in a multi-valued form. For instance, the genre of the movie ‘Ocean’s Twelve’ in the Movielens [7] dataset has 4 values, namely 'Action', 'Comedy', 'Crime' and 'Thriller'. However, in cases that elements of various genres are present in a movie but they are not strong enough to be included in the genre metadata of the movie, this valuable information is ignored by the system.

As of today, the genre-based movie representation used in recommender systems indicates if a genre is present or not without revealing any information about the presence of specific genre characteristics in the movie or the degree that the genre in general applies to it. This information, if available, could enrich significantly the genre representation, enabling the recommender systems to identify movie similarities more accurately.

The recent years, the rise of deep learning techniques has led to outstanding performance in computer vision tasks like object recognition [8]. As the result of the development of powerful deep convolutional architectures, which have significantly enhanced the field of video understanding, we are provided with a rich genre-aware visual representation that could reflect the visual elements of each genre in the movie trailer. The proposed representation could capture the necessary information encapsulated in the genre that discriminates the movies a user likes from the ones he dislikes outperforming the typical genre-based movie CBF recommender systems.

The need for an effective CBF movie recommendation system is strong since it addresses challenges in the recommendation field like the new item problem and the need for explainable recommendations [5]. The new item problem refers to the situation that a movie enters the system and no user feedback is available for it. A CBF system can recommend such a movie to a user if it finds that is similar in content to items that the user has liked. That is not possible for a CF system since it requires the user feedback of an item in order to include it in the recommendation algorithm. Regarding the explainability issue, the recommendations of a CBF are easily understood by the user since they share some attributes with the items with which she has interacted in the past. On the other hand, it is very hard for a CF system to explain its recommendations since the way the machine learning algorithm exploits the historical data to estimate the item and user similarities is too complicated, if available, to be presented to the user. In addition, it should be mentioned that the privacy concerns of users about the use of their data along with the new privacy protecting possibilities for the users, emerging by the new GDPR European regulation, could limit the collection and use of historical data, which are crucial for the success of a collaborative system. Content-based systems, not relying on the user community data, could offer a valuable alternative approach provided that are supplied with rich item representations that can capture what a user likes and what not in a movie.

Towards this end, we propose a novel movie recommender system that filters movies based on the genre-related visual elements of their trailers. The proposed system combines the visual content of the trailer and the genre information of the movie into a single representation which is then exploited by a pure CBF recommender system. We call the system ‘Genre-aware visual CBF movie recommender system”.

First, we decided to choose the trailer as the source of the visual information since it is created to illustrate the most dominant visual elements of a movie and enables a

computationally efficient content analysis compared with the use of the full movie. Fundamental assumption in our work is that the trailers are visually representative of the full movies as it is suggested in [10].

Our second decision concerns the method used to extract the visual representation of the trailer. Recent work in CBF movie recommendation systems uses 2D deep convolutional networks to extract the spatial features of the still frames of the trailers capturing the objects and the scenes of the movie. However, movies, similarly to every video, are characterized by actions that support the narrative of the movie and convey their messages to the audience. Furthermore, genres are commonly characterized by particular actions that a genre-aware representation should not ignore. Apart from appearance, motion information is quite important in capturing actions in a movie. An appearance-based representation could capture the presence of a car in a trailer, but it would be difficult to identify a car chase. As a result, we decide to overcome this issue by using pre-trained 3D deep convolutional networks to extract the spatio-temporal features of the segments of the trailers. Recent advancements in their architectures have shown that they can capture both appearance and motion of the videos [11].

Finally inspired by the Deep Bag of Frames (DBoF) approach described in [12], we use a feature pooling network that learns a single movie representation, using as input the visual features of the segments extracted by the 3D deep ConvNet and as labels the genre of the movie. We name our network Deep Bag of Segments (DBoS) and we investigate the use of its layers' activations as the movie representation in a CBF movie recommendation system. We investigate how the proposed representations affect the utility of the recommendations when they are compared with genre in its typical metadata form and a pure visual representation as it derives from statistical aggregation of its 3D deep visual segment features.

Our main research hypothesis is that a recommender system that uses as movie representation a continuous genre representation, which reflects genre specific visual elements of the movie, performs better than the same recommender system using a binary genre representation, which indicates if a genre applies to a movie or not. In contrast to the second system, the first system, being more accurate in determining which movies are similar to each other, could recognize which movies within a genre are interesting to a user. The second hypothesis we make is that the visual features extracted by a pre-trained deep network are not capable enough of representing movie specific concepts and the genre information could be used to specialize the visual representation to the movie domain. The proposed representation could outperform the visual representation by learning, through the DBoS network, features that are genre related and not focused on the general objects of the frames or the general actions of the clips. In this regard, we also make the hypothesis that the proposed genre-aware descriptor provides better recommendations than the visual features extracted by a 3D deep network. By saying better, we are not limited only to the accuracy of the recommendations, but we assess the capability of the representation to provide novel and diverse recommendations that do not ignore items from the long-tail.

We articulate our research hypothesis into the following research question:

RQ1: Can a 3D deep visual genre-aware descriptor built by a DBoS network provide better movie recommendations with respect to accuracy and beyond accuracy metric (diversity, novelty, and coverage [13]) than genre and visual features extracted by a 3D pre-trained deep convolutional neural network?

We address our question using the above-mentioned representations to a pure content recommender system and we evaluate their performance conducting offline and online experiments.

For the offline evaluation we are using the publicly available dataset MMTF-14K [14] and a sample dataset provided by the video-on-demand service Videoland [15]. Conducting an online experiment, which involves real users using our proposed system in the streaming platform of Videoland (figure 1.1), we pose our second research question:

RQ2: Can the introduction of a deep visual genre-aware descriptor in a movie recommender system provide recommendations of better user perceived utility compared with genre and a pure visual representation?

Our contributions are:

To the best of our knowledge, our work presents the first recommender system that utilizes a continuous genre representation built by a Deep Bag of Segments (DBoS) pooling network that exploits the visual content of the movies trailers.

It is also the first work that examines the use of the spatio-temporal features of the movie trailers extracted from pre-trained deep 3D ConvNets in a movie recommender system.

Last but not least we are conducting an online experiment in a real-world streaming platform to evaluate the user perceived utility of the recommendations produced by a pure content-based recommender system using our proposed genre-aware movie descriptor against the same system using genre and visual 3D deep features.

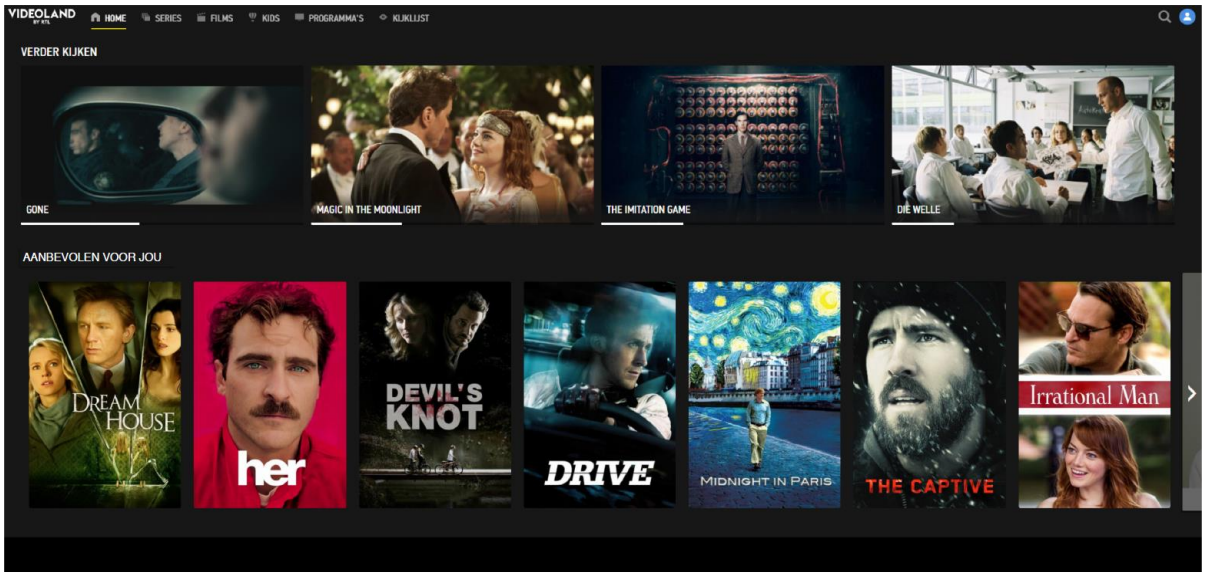

Figure 1.1: Videoland platform [15]

2

Background and Related Work

2.1. Recommender systems

Recommender systems are utilized in a broad spectrum of applications in order to reduce the information overload and help users discover items that suit their needs. Their role is getting more and more important since the volume of the available choices constantly increases, challenging very often the ability of users to find quickly the items that are relevant to them.

They can be seen as a recommender function which takes as input the user and item data in the system and either predicts the rating that the user would give to an unseen item or generates a recommendation list where the ranking of the items reflects the preferences of the user.

The most common and successful methods, aiming to accomplish the recommendation task, are based on the collaborative filtering (CF), the content-based filtering (CBF) and hybrid approaches. Their basic difference is related to the data they use as input. The CF approach is based on the collective intelligence coming from the interactions between the users and the items, whereas the CBF algorithms recommend items exploiting the content of the items. The hybrid approach combines properties from the two former approaches [3].

2.1.1. Collaborative Filtering

The CF technique tries to predict the preferences of the users using the historical data of the user community in the system. A big advantage of that approach is that the system doesn't need any domain knowledge to generate recommendations since it relies exclusively on the ratings of users. The main shortcomings of that approach are that it cannot recommend items for which no historical data are available (new items), and that it performs poorly with very large and sparse datasets.

The algorithms in this type of recommendation approach are grouped in two categories: the memory-based and the model-based [16].

Memory-based

The memory-based algorithms take into account all the available ratings of the users in order to estimate the similarities in users and items, which are necessary for the system to provide recommendations.

The main type of memory-based CF approach is neighborhood-based and it is split into two categories, the user-based and the item-based, which differ in the way that the historical data are used. Both categories focus on the prediction of the user ratings and not on the generation of a list with the most interesting to the user items.

The user-based CF technique relies on the assumption that users with similar ratings share similar preferences. This implies that the system recommends to users unseen items that other similar users have liked in the past. Recommendations in applications that use this kind of algorithms could explain the recommendation of the item B to a user saying, “Similar users also liked item B”.

On the other hand, the item-based CF technique is based on the assumption that the similarity between the items can be inferred by the similarity of their ratings by the system users [17]. The system considers the items that a user has liked and finds the users that also rated those items positively. Finally, the system recommends to the user unseen items that those users have enjoyed in the past. Recommendations in applications that use this kind of algorithms could explain the recommendation of the item B to a user having consumed item A saying, “People who liked item A also liked item B”.

The memory-based models are easy to implement and integrate new ratings, but their performance deteriorates with large and sparse datasets.

Model-based

In contrast to the memory-based algorithms, the model-based systems use the ratings matrix to learn a model that predicts the rating of a user to an item. The models are commonly based on machine learning or data mining techniques and have the capability to identify complex patterns in the ratings that are exploited to provide accurate predictions of user ratings to unseen items.

The first works in this kind of algorithms explored techniques like Bayesian networks [18], probabilistic models [19], latent semantic analysis [20] etc but most recent works are based on the matrix factorization method [21]. In [22], the Probabilistic Matrix Factorization (PMF) achieves state-of-the-art accuracy in rating prediction, scaling very well in large datasets and performing well in cases where very few ratings are available. The recent years, deep learning techniques have been employed to capture non-linear relationships inside the rating matrix and have managed to become the state of the art in the explicit ratings prediction task [23][24][25][26].

The model-based systems are effective, fast and scalable, but quite inflexible since it is not easy for them to incorporate new historical data, once they are trained.

2.1.2. Content-based Filtering

In contrast to the CF paradigm, CBF systems recommend to a user items that have similar attributes or properties with the items the user has liked in the past. In this type of the systems, the ratings of other users are not used to estimate item and user similarities and the accurate representation of the items is crucial for the system’s performance. Content-based systems analyze the content of the items and produce their representations, based on which the user profiles are generated. Each user profile is based on historical data, which determine what attributes in the item representations reflect the user’s preferences, and finally the system recommends to the user unseen items that match up his/her profile [5].

The big strengths of this type of systems are that they are able to integrate new items in their recommendation process and that their recommendations are easily explainable. They also do not require a large user community in order to provide effective recommendations.

The greatest shortcoming of this type of systems is known as the limited content analysis problem. For many domains, a content analysis, that discriminates the preferred items from the items that the user would not be interested, is extremely challenging and difficult. Representing the items with meaningful and useful features requires expert knowledge and tremendous human effort, making the automatic item representation necessary for the wide deployment of CBF systems. The limited content analysis problem requires sophisticated and innovating approaches for the item representation learning and our work heads to that direction.

Another important drawback of this approach is the overspecialization problem. A CBF system, recommending to a user only items similar to the ones he has already liked, is often not able to satisfy users with diverse tastes and it is very hard to adapt to the changes of the user's preferences over time.

2.1.3. Hybrid Systems

The limitations of the CBF and CF systems can be addressed by combining them into another type of systems called hybrid. The most common hybrid systems are model-based CF systems, which incorporate content as side information aiming to address the new item problem and improve their accuracy for users with few interactions with the system. Future work could explore the value of our proposed movie representation as side information in a hybrid recommender system.

2.1.4. Multimedia recommender systems

Multimedia refers to items that comprise one or more modalities, namely the textual, the aural and the visual [27]. Typical examples include movies, books, music songs, e-commerce products and even news articles when they are accompanied by images.

In the literature, collaborative filtering, content-based filtering and hybrid models have been proposed to address the recommendation problem of such items. Generating a representation that captures the content of a multimedia item seems quite challenging and it might be the reason why collaborative filtering models outperform the content-based ones in applications where users' historical data is available.

In multimedia CBF systems, two types of features are commonly used for item representation, namely the high-level and the low-level features [10]. The high-level features express the semantics of the items and either they derive from expert generated metadata or they are extracted from textual sources like the item description, reviews and so on. This type of features is the most common in the content-based recommendation due to their good availability in commercial products and the maturity of the techniques that analyze and process textual data. For instance, a movie is represented by expert annotated metadata like title, genre, cast, directors, synopsis etc but it is quite questionable if they possess the complete information required for an effective content-based filtering model. In this regard, low-level features, which are extracted directly by the media content, could generate valuable representations, capturing important information for the user preferences understanding.

2.2. Related work

2.2.1 Low-level features in movie domain

Our work proposes a movie recommender system that exploits the visual content of the movie trailers. In this regard, a presentation of work in the movie domain that exploits low-level features is important in order to identify gaps and opportunities in the movie recommendation task.

In the movie domain, early work suggested the use of low-level features in order to learn semantics of a movie like the genre. In [28], Rasheed et al process the frames of movie trailers in order to compute four features that according to the film literature are associated with the high-level semantics of a movie. A shot detection algorithm provides the key frames of the trailer and the system combines the extracted features (average shot length, color variance, motion content and lighting key) into a feature vector, which is used as input in a mean shift clustering model that assigns each trailer to one or more of the four genres (horror, action, comedy, drama) present in the movie dataset. The results of their experiment suggest that the selected visual low-level features are capable of predicting the genre of a movie. Extended work on that direction could be beneficial to the ambitious goal of automated film understanding.

In [29], Jain et al extend the approach by including audio features of the movie clips in the movie representation. They created a neural network based movie genre classifier which displayed improvements in the classification results.

A different approach is presented by Zhou et al in [30]. The authors make use of image descriptors to extract high-level features from the trailers' key frames and they build a bag of visual words (BOVW) representation for each trailer. Finally, a k-NN classifier is employed to assign the trailers to their associated genres.

In [31], the authors extract 277 visual and audio features from the trailers, and they use a feature selection mechanism in order to use the most relevant of them, as the input in a SVM genre classifier. It is worth mentioning that the visual features in this work are not selected according to the cinematic principles, but they are generated by the MPEG-7 descriptors.

The most recent work in the movie genre classification task proposes the use of deep learning techniques for the feature extraction of the trailers. The authors in [32] suggest the use of various ConvNet models to extract features from the trailers frames in order to capture different aspects of the media content. Pre-trained 2D ConvNets on ImageNet and Places datasets are fine-tuned with the movie trailer dataset and provide features that capture the appearance and the context in the trailers frames. Additionally, a 3D ConvNet, pre-trained on the Sports-1M dataset, is fine-tuned on the trailers dataset to focus on the motion in the trailers and an MLP model using MFCCs audio features explores the audio aspect of the trailer. Finally, a 2D ConvNet model trained on the trailer dataset from scratch is employed to capture movie-related elements of the trailers. The 2D ConvNets use as input the key frames from the trailers whereas the 3D ConvNet and the MLP models use segments from the trailers. The predictions of all the networks are combined to a trailer representation which is used in an SVM classifier to provide state-of-the-art genre classification accuracy.

The works mentioned above exploit the visual content of the multimedia items using either hand-crafted low-level features or features derived from image descriptors. The deep learning approach in [32] doesn't involve a feature extraction stage as the raw content is

used directly as input in the classification model. However, in content-based recommendation, the items need to be represented according to the vector space model in order to be analyzed and to be recommended by the system. Below, pure CBF recommender systems and CF systems using movie content as side information are presented in detail, as they are the most relevant to our work.

In [10], Deldjoo et al propose a pure content-based system that uses stylistic visual features extracted from movie trailers and full-movies and compare its performance against conventional systems that are based on metadata like the movie genre. The proposed approach could be of great value when the metadata is not available or possess low discriminatory power. The authors claim that the aesthetic visual features of a movie like the lighting, the color and the motion reflect the wish of the movie makers to convey specific emotions to the users and it is likely to determine whether a movie is appealing or not to a user. The key frames of the full movies and the trailers are extracted and a feature vector with 5 dimensions is computed. The dimensions refer to the average shot length of the video, the mean color variance over the key frames, the mean motion average and standard deviation across all frames and the mean lightening key over the key frames. For the aggregation of the features into a video representation, statistic functions like average are explored. The system computes the cosine similarity between the items and employs a "k-nearest neighbor" (k-NN) content-based algorithm to recommend to the users items that are similar to those they have liked in the past. The authors test their approach in a dataset of 167 full-length movies and trailers, and they evaluate its performance using the precision and recall metrics. The work concludes that the stylistic low-level visual features can predict the user preferences in a comparable way to the genre and suggests that the trailers could be used as alternative to the full movies when visual features need to be extracted for a content-based recommendation task.

In [33], the authors investigate how the stylistic visual features proposed in [10], could enhance the performance of a collaborative filtering movie recommender system addressing especially its weakness to deal with the new-item problem. The visual features are incorporated as side information in a Factorization Machine algorithm, and the model is tested on a dataset of 13K movie trailers. This model shows an impressive tenfold improvement in prediction accuracy, compared to the same algorithm exploiting the genre of the movie.

The work in [34] doesn't propose a new approach for exploiting visual content in the recommendation field but investigates the contribution of the stylistic visual features on the quality of the recommendations in terms of precision, recall, diversity and novelty. The authors conduct a user study to evaluate the user perceived quality and they identify a disagreement between the online evaluation and offline evaluation findings. The offline evaluation shows that stylistic visual features perform better than high-level features with respect to diversity and novelty, whereas the online evaluation suggests the opposite.

The work in [35] extracts features from the trailers frames using MPEG-7 visual descriptors and a pre-trained deep learning network. Aggregation functions like intersection, average and median are deployed to provide a single feature vector for the whole trailer, which can be integrated as side information in a collective SLIM collaborative filtering method. The system is evaluated on a trailer dataset of 13K movie trailers and it is compared with collaborative filtering systems that use genre and tags as side information. The precision, recall and mean average precision metrics of the Top-N recommendation lists show that the MPEG-7 descriptors outperform deep learning features implying that style features are more powerful than the content appeared in the frames. The MPEG-7 descriptors perform better

than the genre and tag features and, not surprisingly, the Canonical Correlation Analysis (CCA) based fusion of the MPEG-7 and deep features delivers the best overall results.

The most recent work in the movie content-based recommendation field suggests the use of multiple deep learning models in the task of feature extraction, aiming to capture various visual aspects of the trailer that could contribute in a more accurate movie representation. In contrast to [32], the authors in [36] do not include audio, motion and trailer-related features in the feature extraction process but focus on the content and the context, deploying very deep ConvNets pre-trained on ImageNet and Places-365 datasets. The models extract features of the key frames of the trailers and data augmentation techniques are applied to enhance the descriptive power of the computed feature vectors. The aggregation of the key frames features is critical as the recommender systems require a single representation for the whole movie. The authors address this issue by combining the extracted key frame features through a scene categorization approach. Using k-means clustering, the feature vectors are assigned to scene categories and each trailer is represented by a vector where each dimension reflects the relevance of the trailer to the corresponding scene category. The final representation is utilized by a pure content-based recommender system. The results of their experiments suggest that the features extracted by deep ConvNets are more appropriate than low-level visual features in [10] for predicting the users' preferences in the movie domain. The work exploits the power of deep leaning networks in learning visual features of images but limits itself by not considering the temporal dimension of the trailers. The extraction of action-related features could improve further the value of their proposed representation.

Finally in [14], the authors release the MMTF-14K dataset for the evaluation of video recommender systems. Besides the opportunities for advanced research that a dataset with metadata, audio and visual descriptors of 13,623 trailers brings, the authors provide baseline results for recommender systems based on various content modalities that allow the fast evaluation of new methods and approaches in the content-based recommendation task.

2.2.2. Video representation learning

Our work proposes a visual video representation applied to movie trailers in order to benefit the movie recommendation task in cases where the collaborative filtering approach is not sufficient. In this regard, a presentation of high profile work in video representation learning is considered necessary to understand the challenging nature of the field and how our work utilizes and evolves the current practices in order to achieve its goal.

For many decades, the computer vision community applied a tremendous effort to discover techniques and methods that could form a visual representation of an image content that could address the quite challenging task of the automatic image understanding [37][38][39]. Not surprisingly, early work in video representation learning tried to extend those techniques in 3D space aiming to produce video descriptors that could facilitate the task of action recognition [40] [41]. In this category of hand-crafted descriptors, the improve Dense Trajectories (iDT) [42] algorithm achieves the best performance in action recognition task by identifying dense feature points in video frames and using the optical flow to track them. However, iDT is quite computationally expensive to be adopted as a generic video descriptor that could be applied to large-scale datasets [11]. Another approach that could deal with a massive amount of videos in the new digital world was under demand and again the research effort would begin from the image domain.

The recent years, the progress in the GPU hardware and the public release of the very large ImageNet [43] dataset led the research community to re-examine the use of neural networks in computer vision tasks and propose techniques that enabled the development of deep convolutional networks, which vastly outperformed the hand-crafted approaches in visual recognition [8]. Since the requirement of very large datasets for the training of the deep learning networks could limit the wide deployment of this technique in many interesting visual tasks, transfer learning strategies were investigated to address that issue. Two approaches were proposed, the ConvNet as fixed feature extractor and the fine-tuning approach. The former approach suggests the feature extraction from images through the use of the convolutional part of the pre-trained on ImageNet deep convolutional networks, and then the use of those features in a classifier, which is adjusted to the specific task. According to the latter approach, all or some of the network's layers are fine-tuned on the target dataset by continuing the back-propagation method. Commonly the fine-tuning is applied to the last layers of the networks, since the first layers are designed to learn more generic and basic features of the images like edges, corners etc, whereas the last layers specialize more to the labels of the input data. Initializing the model with the weights of the pre-trained model is proved to generalize better than using random initialization in many kinds of applications [44]. In [45], the authors investigate the capability of a pre-trained on ImageNet deep convolutional network to be used as a generic image descriptor for visual tasks that are different from image classification task. Their experiments show that the proposed model shows competitive performance against sophisticated hand-crafted descriptors in similar tasks like scene classification, object detection, fine grained recognition and visual instance retrieval. This finding is of great value since it allows the exploitation of deep learning architectures in visual tasks where the data availability for their training is limited.

The tremendous success of convolutional networks in image domain led the researchers to explore their capabilities in the more challenging field of video understanding. The new architectures, evaluated mostly in the action recognition task, can be classified into 2 groups based on the dimensionality of their convolution layers: 2D ConvNet-based video representation; 3D ConvNet-based representation.

2D ConvNets in video representation learning

Karpathy et al in [46] present the first work that exploits the use of ConvNets in large-scale video classification. The authors propose the feature extraction of the video frames using 2D ConvNets and they explore various architectures in order to incorporate the temporal aspect of the videos and evaluate their contribution in the action recognition task. The proposed CNN architectures for the fusion of time information are presented in figure 2.1.

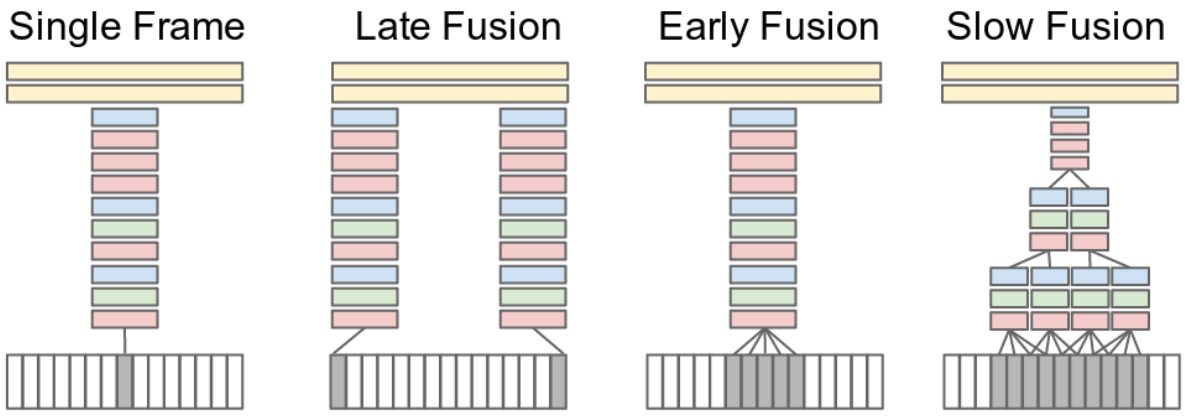

Figure 2.1.: Architectures for time information fusion [47]

The single frame architecture doesn't consider the time dimension of the video and learns to classify the videos into actions classes averaging the class predictions of its individual frames. It constitutes the baseline model used to identify how the temporal information benefits the performance of the classification network. The late fusion architecture applies single-frame networks to every 2 frames that are spaced 15 frames apart and combines their spatial information in the first fully connected layer of the network. In that way, the network aims to detect the presence of global motion in the system. On the other hand, the early fusion architecture combines the spatial information of a sequence of 10 frames in the first convolutional network aiming at detecting the local motion in the clip. The following layers of the network do not use the temporal dimension preventing the model from identifying global motion characteristics of the clip. Finally, the slow fusion architecture incorporates the temporal evolution of the 10-frame clip in the first 3 convolutional layers of the network, improving significantly the capability of the last layers to capture the global spatiotemporal characteristics of the clip. The proposed architectures are evaluated in the very large Sports-1M dataset showing that the slow fusion model outperforms consistently the other architectures. The single frame architecture performs better than the early and late fusion models showing that the integration of time in a small part of the network has limited impact on the system's performance. It can be inferred that extending the spatial convolutions in time across all the layers of the network can provide significant gains in the system's performance. Furthermore, the authors explore the transfer learning capability of the slow fusion model to the popular UCF-101 action recognition dataset. The results show that fine-tuning on the new dataset only the last 3 layers of the pre-trained model improves significantly the performance of the system against training the model from scratch or fine-tuning all the layers of the network. That finding confirms the generalization ability of a pre-trained ConvNet to small video datasets.

The poor performance of the proposed slow fusion model in [46] against the state-of-the-art hand-crafted representation [47] on the UCF-101 benchmark dataset (65.4% vs 87.9% classification accuracy) was an indication that a 2D ConvNet architecture with input multiple stacked frames was not able to capture effectively the motion aspect in the video. To address that issue, the authors in [48] proposed a two-stream ConvNet architecture composed by a spatial and temporal stream. The spatial stream takes as input single video frames and is responsible to capture the spatial information of the video based on the frames' appearance. The temporal stream processes multiple stacked optical flows and is responsible for capturing the motion in the video. The approach is presented in the figure 2.2. Both streams are implemented by the same 2D ConvNet and their prediction softmax scores are fused either by averaging or by utilizing a linear SVM classifier. In the training of the spatial stream, one single frame is randomly selected for each video in the dataset and undergoes scaling, cropping and rotating to provide the inputs to the network. The input for the temporal network is based on the computation of the optical flows for L consecutive frames around the selected frame. In the testing phase, the system selects for a given video 25 frames and after data augmentation produces 10 testing samples for each one of them. The prediction score of each video derives by averaging the scores of its sample frames. The approach is evaluated on the UCF-101 and HMDB-51 datasets. The results show that the temporal stream outperforms significantly the spatial stream, which indicates the importance of motion modelling in the action recognition task. The fusion of the two streams is also found to be important. It improves the performance of spatial and temporal streams by 14% and 6% respectively, providing state-of-the-art accuracy on the UCF-101 dataset (88%).

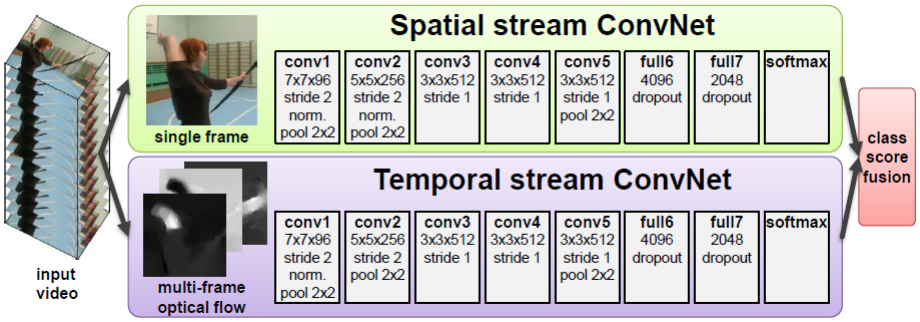

Figure 2.2: Two-stream architecture for video classification [49]

In contrast to the slow fusion model in [46], Ng et al in [49] do not adopt the use of multiple frames as input and they do not extend in time any convolution layer in their proposed CNN network. Similar to [46], they use 2D ConvNets to process the individual frames of the videos, but they deploy feature pooling and LSTM architectures to aggregate the frame representations into a single video descriptor. Various feature pooling architectures are explored to determine the position of the max-pooling layer which aggregates the video frames over time. The proposed architectures are presented in figure 2.3. In the conv pooling model, the max pooling layer aggregates over time the spatial features of the frames from the last convolution layer, whereas in the late pooling model, it aggregates the high-level features of the frames from the last fully connected layer. The slow pooling model aggregates the convolutional features of a small sequence of frames, combines the aggregated spatial information in the first fully connected layer and then aggregates the high level features of the sequences in a new max pooling layer before it delivers them in the last convolutional layer. The local pooling network aggregates the spatial information in the frame sequences but in contrast to the slow pooling model, it uses a larger final softmax layer that takes as input the high-level features of each sequence, without employing a second pooling layer among the 2 fully connected layers. The last proposed architecture, called time-domain convolution, performs a time convolution layer over the spatial features of the frames in order to capture local features inside small sequences of frames. Next, it performs max-pooling on the output of the time convolution layer and passes the aggregated features through the 2 fully connected layers. The feature pooling architectures aggregate the frame features over time but they do not take into account the temporal order of the frames. In addition to the feature pooling network, the authors propose a LSTM architecture to exploit the temporal order of the sequences. They evaluate the performance of the proposed architectures on the Sports-1M dataset. Regarding the feature pooling architectures, the conv pooling model displays the best performance showing that keeping spatial information of frames in the aggregation stage benefits the action classification task. The authors also examine the effect of the input size finding that using 120 frames (2 min at 1 fps) delivers better classification accuracy than using 30 frames. The need for a computationally efficient global video descriptor led the users to process frames at a rate of 1 fps losing, in that way, the motion information of the video. To address that issue, they precomputed optical flow frames and they used them as input in the same architectures. Fusing the prediction results from both streams, they found that the contribution of optical flow in the video classification, on noisy datasets like Sports-1M, is not significant. However, evaluating the contribution of the optical flow information on the less noisy UCF-101, the results are different. An improvement of 6% on the conv pooling performance is reported, which improves slightly the state-of-the-art performance on the dataset (88.2%).

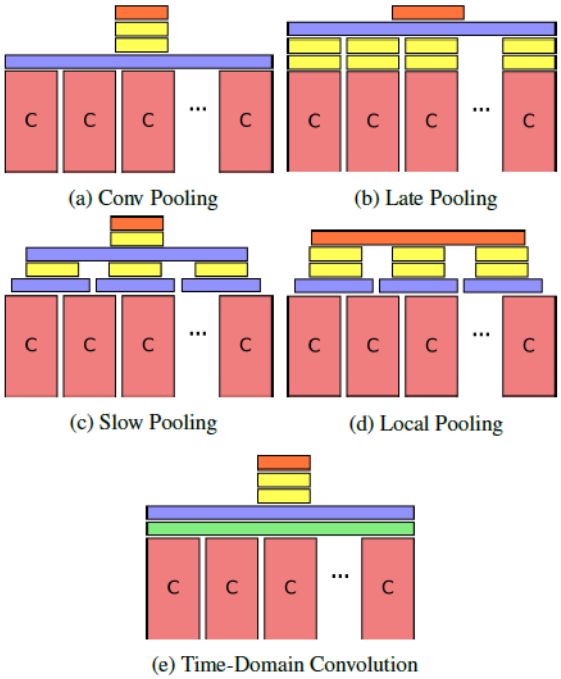

Figure 2.3: Feature-pooling architectures [50]

3D ConvNets in video representation learning

The research community and the industry acknowledge that an effective video descriptor could benefit significantly various video analysis tasks. The pre-trained on ImageNet 2D deep ConvNets have proved that are capable of capturing the appearance in a video with respect to the objects and the scenes in its frames, but their performance in modelling the video's motion is found to be very poor. Two stream architectures address that problem providing state-of-the-art results in the action recognition task but their structure does not facilitate the generation of a simple, efficient and scalable video descriptor. The 3D ConvNets utilize 3D convolution kernels providing spatio-temproral features that capture simultaneously the appearance and the motion in the video. In that way, they are able to integrate the temporal dimension of their multiple-frame input and provide a representation that could be used as a generic video descriptor. The figure 2.4 shows that only 3D convolutions on multiple frames retain the temporal information in the system. 2D convolutions, performed on a sequence of frames, produce as output an image where the time information is eliminated.

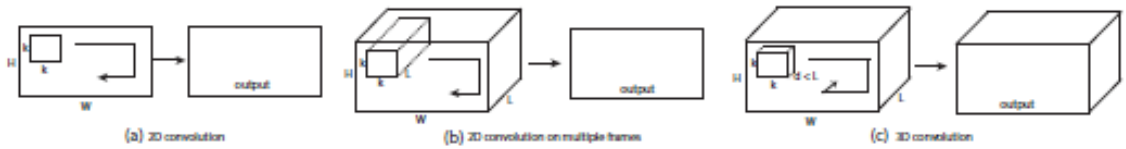

Figure 2.4: 2D and 3D convolution operations [11]

Tran et al in [11] propose the use of a 3D ConvNet (C3D) to extract spatiotemporal features from videos. The model trained on the Sports-1M dataset is capable of operating as fixed feature extractor for various datasets and tasks. The proposed ConvNet consists of 8 convolutional layers, 5 max pooling layers, 2 fully-connected layers and a softmax layer that provides the prediction scores of the input clip for each one of the available classes in the dataset (figure 2.5).

Figure 2.5: C3D architecture [11]

After experiments on the small UCF-101 action recognition dataset, the authors concluded that a 3x3x3 kernel for all the convolutional layers is the most effective ConvNet architecture for aggregating the temporal dimension on a deep network. The C3D model is trained with 16-frame clips and each frame is resized and cropped to give a fixed size input with 16x3x112x112 dimensions. After its training on the very large Sports-1M dataset, the last two layers of the model are removed to formulate the first deep learning based video descriptor.

The C3D descriptor of a video is obtained according to the following process:

Each video is split in 16-frame segments having 8 frames overlap between adjacent segments. The network processes each segment of the video and provides the activations of the fc6 fully connected layer as the clip representation. The clip representations are averaged and normalized to provide the descriptor of the whole video in a vector of 4096 dimensions.

The C3D descriptors combined with a linear classifier are evaluated in 6 benchmark datasets showing state-of-the-art performance in action similarity, scene classification and object recognition tasks. In action recognition task they provide the best results when only the RGB frames of the video are used as input of the model. The descriptor's performance is also comparable with architectures that additionally incorporate explicitly the motion information through the use of optical flows.

The performance of C3D descriptors in various tasks shows that the action recognition dataset possesses the appropriate information for a 3D deep-learning network to learn generic video representations.

Sports-1M dataset provides a large dataset that makes possible the training of deep 3D ConvNet architectures preventing their overfitting. However its weak annotation is very likely to limit the effectiveness of the provided features. A large expert annotated video dataset could enable the utilization of much deeper architectures, learning even more powerful video representations.

The Kinetics Human Action Video dataset [50] was developed to provide a large dataset of high annotation quality, that could boost the performance of deep ConvNets in human action recognition task. It contains over 300000 videos of human actions belonging in 400 classes that cover a wide spectrum of human activities. Each video has a duration of around 10s containing frames that are relevant to a certain action. Not specializing on sports actions and not including noisy frames are two elements that favor it against the biggest Sports-1M dataset for the training and evaluation of deep ConvNet architectures on the action recognition task.

The advent of the Kinetics dataset led the authors in [51] to investigate the ability of the new dataset to play a role similar to the ImageNet dataset for video analysis tasks. The size and the quality of ImageNet dataset were critical in the development of extremely powerful, advanced, deep 2D ConvNet architectures in object recognition task. In this regard, the utilization of the Kinetics dataset in the training of deep 3D ConvNets for action recognition, could provide models that learn more accurate video representations than the shallow C3D descriptor (10 layers). The authors select the successful in image classification architecture

ResNet [52] in various depths and extend their convolution and pooling operations in the 3D space. They also explore extended versions of the ResNet architecture like the pre-activation ResNet [53], wide ResNet (WRN) [54], ResNeXt [55], and DenseNet [56]. The figure 2.6 shows the block for each architecture. They evaluate the proposed models on 4 action recognition datasets, namely the Kinetics, the UCF-101, the HMDB-51 and the ActivityNet. They find out that the Kinetics dataset is large enough to train very deep 3D ResNet architectures without overfitting them. The kinetics dataset enables the utilization of 3D ResNet models with depth of 152 layers (ResNet-152), whereas the smaller datasets UCF-101 and HMDB-51 cannot prevent overfitting even for models with 18 layers (ResNet-18). Similar to the ImageNet case, it is found that increasing the depth of the 3D ResNet models from 18 to 152 layers leads to significant improvement in the classification accuracy. However, the improvement from ResNet-152 to ResNet-200 is very small indicating that the data of the Kinetics dataset is not sufficient to train a model with 200 layers.

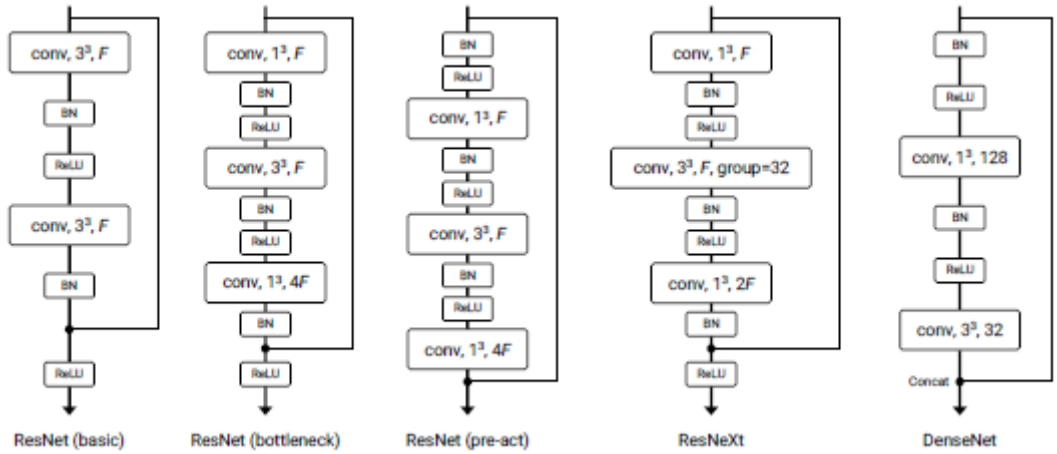

Figure 2.6: 3D ResNet-based architectures [52]

The authors compare the performance of the proposed models with the state-of-the-art 2D and 3D architectures on the 4 datasets. Among the proposed architectures the best performance on Kinetics dataset is achieved by the ResNeXt-101 model which is also comparable with the state-of-the-art model I3D [57]. Unlike the ResNeXt-101 model, the I3D model is a two-stream 3D architecture and its input is 64 times larger than the one used in the ResNeXt-101 model. The dimensions of the input for the ResNeXt-101 model is 3x16x112x112 whereas the dimensions of the input for the I3D model are 3x64x224x224. Keeping only the spatial stream of the I3D model and increasing the duration of the ResNext-101 model's input to 64 frames, the best accuracy in the Kinetics validation and test datasets is provided by the modified ResNeXt-101 model. Finally, the authors fine-tuned the pre-trained on Kinetics ResNet models on the UCF-101 and HMDB-51 datasets. The results are similar to the Kinetics dataset, showing that the ResNeXt-101-64f model outperforms all the 2D architectures and the 3D ones that are exploiting only the RGB frames of the video. The ResNeXt-101-64f model far outperformed the C3D model on the UCF-101 dataset (94.5% vs 82.3%) showing its power in generating accurate video representations.

Our work uses the pre-trained on Kinetics ResNeXt-101-64f model to extract spatiotemporal features from the movie trailers segments.

3

Approach

Our approach proposes the extraction of spatiotemporal features of the movie trailers using a pre-trained 3D ConvNet and their utilization into a feature pooling network towards the generation of a genre-aware movie descriptor, called GenreVis_3D. The contribution of the proposed movie descriptor in the recommendation task is evaluated by its use to a pure content-based filtering recommender system. Offline and online experiments are conducted to evaluate the performance of the proposed representation against pure textual and visual representations, namely the genre and the visual features provided by 3D deep-learning networks. The fundamental assumptions in our work are that the trailers are visually representative of their full-length movies as is suggested in [10] and that the 3D deep learning networks trained in a very large video dataset for the action recognition task can be used as fixed visual feature extractors for videos like movie trailers as suggested in [11].

3.1. GenreVis_3D movie descriptor

The process of generating the GenreVis_3D descriptor consists of two stages: the visual feature extraction stage and the Deep Bag of Segments (DBoS) pooling stage.

3.1.1. Visual feature extraction

One of our main hypotheses is that there is a value in representing a video using spatio-temporal features instead of considering it as a sequence of static frames where their order is not important. We claim that a descriptor that incorporates the temporal aspect of the video could provide a rich representation that could address the limited content analysis problem of the content-based recommender systems.

Inspired by the success of deep 2D Convolutional Neural Networks (ConvNets) in the generation of generic feature representations for images, we suggest the use of deep 3D ConvNets to acquire generic visual features for the segments of the trailers. The use of 3D kernels along with the use of consecutive frames as input enables the capture of the temporal aspect of the video. The big drawback of the 3D ConvNets is that they have a huge number of parameters that makes them hard to train. Training 3D ConvNets with small datasets results in their overfitting, limiting their ability to provide a generic visual representation. The recent launch of the very large video dataset Kinetics overcomes that issue enabling the training of deep 3D CNNs in the task of action recognition and the transfer of their learned parameters to other datasets and tasks. In this work, we employ a deep 3D ConvNet pre-trained on Kinetics dataset to extract features from the trailers

assuming that the network parameters are able to generalize in a movie dataset and provide features that could recognize actions in movie trailers.

We use the deep pre-trained on Kinetics 3D ResNeXt-101-64f network as it achieves the highest accuracy in the action recognition task on the Kinetics dataset compared to the state-of-the-art methods when the RGB frames of the videos are used as input. It also outperforms the other methods when it is used for fine-tuning on the popular but smaller video datasets UCF-101 and HMDB-51. This fact supports our hypothesis that this model is able to generalize well, learning from trailers features that reflect the motion and the appearance of the movie.

The 3D ResNeXt-101 network is a 101 layers deep network trained on Kinetics dataset to classify over 300,000 videos into 400 action classes. It consists of 5 convolution layers, 1 global average pooling layer and a fully connected layer. The network was trained using stochastic gradient descent (SGD) with momentum 0.9, weight decay 0.001 and learning rate that starts at 0.1 and is divided by 10 after the validation loss saturates. Spatial and temporal data augmentation was performed generating training samples of size 3 channels x 64 frames x 112 pixels x 112 pixels. Detailed description of the network is available in [52].

We provide to the pre-trained network, as input, every trailer of the dataset in order to extract its visual features. The network is designed to receive, as input, segments of 64 frames with size 3x112x112 pixels and consequently the trailers are preprocessed to meet that requirement.

Each trailer of the dataset is divided in segments of 64 frames which are scaled to 3x112x112 pixels as follows:

The small dimension of the frame becomes always equal to 112 pixels. A frame with dimension w x h (h > w) is converted to a frame with dimensions ((112 * height / width, 112). Then the frame is cropped in its center to provide a 112x112 frame.

No features are extracted from segments with less than 64 frames.

The network processes the trailer and we obtain, for every segment of the trailer, its visual representation through a vector, including the activations of the network's global average pooling layer. With that approach, each 64-frame segment of the videos is represented with a feature vector of 2048 dims.

3.1.2. Deep Bag of Segments (DBoS) pooling

Recommender systems use, in filtering, single item representations in order to discover items that fit the user preferences. The feature vectors of the trailer segments are preprocessed and then, they are used by a pooling network to provide the GenreVis_3D representation of the trailer. PCA is applied to all segments of the trailers and a compact representation is obtained that captures 95% of the variance, reducing the number of dimensions from 2048 to 128 for each segment.

Our approach is inspired by the notion of Deep Bag of Frames (DBoF) network [12] and extends it to the Deep Bag of Segments (DBoS) network, aiming to aggregate the feature vectors of the trailer segments to a single movie representation. In contrast with the DBoF approach we don't aspire to have a generalized video descriptor and we consider that a representation that specializes to the labels of the video, namely the genre, could provide a valuable movie representation for the task of movie recommendation. Our work investigates

which layer of the proposed DBoS network provides the genre-aware movie representation with the best performance in a CBF recommender system.

Our DBoS network (Figure 3.1) is composed by 2 fully-connected layers with ReLU activations, one max pooling layer between them and finally one fully connected layer that predicts the genre of the movies based on the 128-dimensional feature vectors of k segments of their trailer. The input segments can be selected either from the beginning of the trailer or randomly. The input is fed in the first fully connected layer where it is converted after ReLU into a sparse representation of 1024 x k dimensions. That representation is passed through the max pooling layer and provides the first available single representation for the whole movie. We name the representation GenreVis_3D_pool and each one of its 1024 dimensions represents a discriminatory element of the genre that is learnt by the network through back-propagation. Using this approach, we have the possibility to generate more movie representations adding fully connected layers between the pooling layer and the final fully connected layer that classifies the trailer into its genres. We believe that, such representations, which combine the visual information extracted from a 3D ConvNet network with the information contained in its genre label, could provide a meaningful and useful representation for the movie recommendation task. We add a fully connected layer before the output layer, and we name it GenreVis_3D_fc2. It provides a more compact representation with 512 dimensions that combines the learned features from the previous layer. We explore also how the ReLU activation function affects the performance of the representation in the recommendation task, creating the representation GenreVis_3D_fc2-R. The GenreVis_3D_fc2-R representation has also 512 dimensions but keeps only the positive values of the previous layer. The negative values of the previous layer become zero. Finally, we investigate the usefulness of the output of the network as movie representation. The output named GenreVis_3D_fc3 provides a very compact descriptor with dimensions equal to the number of available genres in the dataset and the value of each dimension reflects the extent that a genre applies to the movie.

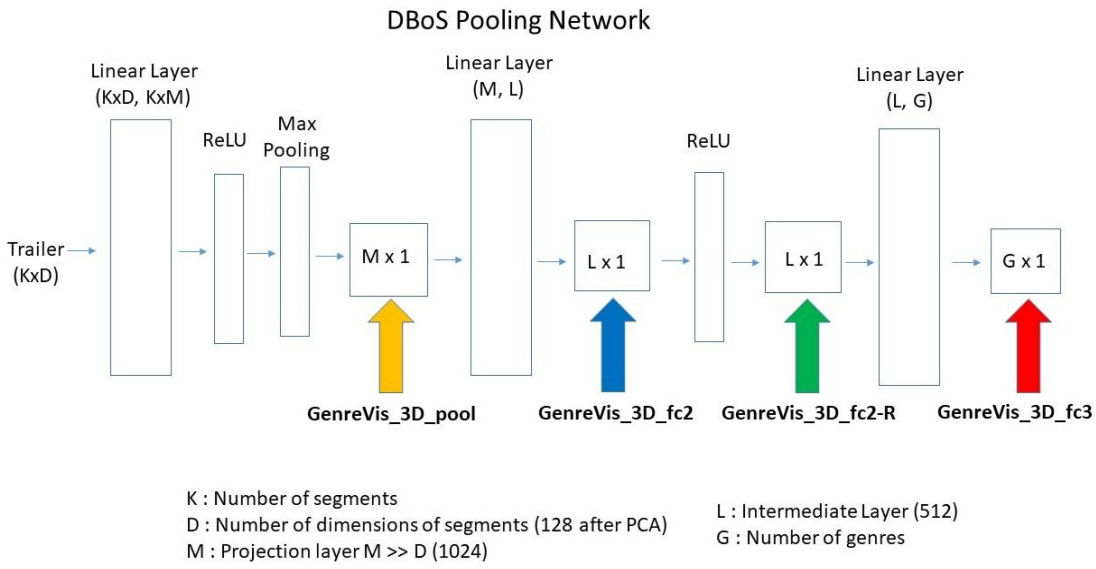

Figure 3.1: DBoS Pooling Network

The network is trained using Stochastic Gradient Decent (SGD) with learning rate 0.1 and weight decay 0.0001. The loss function of the network optimizes a multi-label one-versus-all loss based on max-entropy for each sample in the input batch. The network is trained until we achieve training error equal to 0 since we desire for the network to overfit and learn patterns that are specific to the movie dataset.

Experimentation is required to suggest which variant of the GenreVis_3D representation is the most beneficial to the recommendation task. Exploring the effect of the layer (pool, fc2, fc2_R, fc3), the number (20, 50, 80) and the order (first or random) of the input segments, we generate for each trailer 24 variants of the GenreVis_3D representation.

3.2. Baseline representations

We are using as baseline representations to compare and evaluate our approach the genre representation of the movie and the visual representation derived by the pre-trained 3D deep learning network, called Visual_3D.

3.2.1. Genre representation

The genre representation is a binary vector with dimensions equal to the number of the genres in the dataset. The value of each entry in the vector is 1 if the respective genre applies to the movie and 0 if it doesn't.

3.2.2. Visual_3D descriptor

The Visual_3D descriptor derives from the statistical aggregation of the deep visual features of the trailer segments. For each trailer we are generating 2 representations using the mean and the median functions. Each element of the Visual_3D_mean and Visual_3D_median descriptors is the average and the median of the corresponding elements of the feature vectors of the video segments, respectively. Standardization is applied to all the feature vectors of the trailers and PCA is employed to decorrelate the features and provide trailer vectors that are reduced to a dimensional space that represents at least 95% variability of the data. The reduced size of the final descriptors depends on the dataset that we use in our recommendation task.

3.3. Recommendation

We assess the value of our proposed representation using it as input to a pure content-based filtering recommender system. We employ a "k-nearest neighbor" (k-NN) approach according to which the system recommends or not to a user an item, considering the ratings of the user to its k most similar items. Each item i is represented by a feature vector f_i according to the selected representation approach and the cosine similarity equation (1) is applied to compute the similarity between all the items. For each item i, a set NN_i with its k most similar items is created. The predicted rating of a user u to an unseen item i is estimated using the ratings of the user to its neighbors' items in the set NN_i according to the equation (2). The goal of the system is to recommend items that are similar to the ones the user has rated positively in the past. In the end, the algorithm recommends to each user N unseen items with the highest predicted rating score. The cut-off value N defines the size of the recommendation list and typical values for movie recommender systems are 5, 10 and 25.

$$s_{ij} = \frac{f_i^T \cdot f_j}{\|f_i\|_2 \cdot \|f_j\|_2} \quad (1)$$

$$r_{ui} = \frac{\sum_{j \in NN_i} r_{uj} \cdot s_{ij}}{\sum_{j \in NN_i} s_{ij}} \quad (2)$$

4

Offline experiment

We conduct both, offline and online, experiments to evaluate how the use of our proposed representation affects the performance of a movie recommender system compared with the use of genre and Visual_3D representations. In addition, we use offline evaluation to understand how the various options in the generation of the GenreVis_3D descriptors affect the recommendation quality, then choose the ones that should be tested online in the real-world environment of Videoland. For our offline experiments we are using 2 datasets. The first one is a subset of the publicly available movie dataset MMTF-14K and the second one is a subset of the movie dataset of Videoland. The offline experiment in Videoland dataset is conducted to guide our model selection for the online experiment in a way that takes into account the properties of the system tested online. Furthermore, having two different datasets help us verify how our approach performs on datasets with different statistics.

4.1. Datasets

4.1.1 MMTF-14K Dataset

The MMTF-14K [14] dataset is a multi-faceted dataset that is designed to support the evaluation of video-based recommender systems and other multimedia tasks like auto tagging and genre classification. It includes metadata, audio and visual descriptors for 13,623 movie trailers along with their 12.5 million 1-to-5 star ratings provided by 138,492 users. For our experiment, we use a subset that includes ratings from randomly selected 3,000 users, with the condition that each user has rated more than 50 movies. The statistics of our experiment's dataset is shown in the following table.

Dataset	|U|	|I|	|R|	|R| / |U|	|R| / |I|	|R| / |U| * |I|
MMTF-14K	3,000	7,866	517,710	172.57	65.81	0.021

Table 4.1: MMTF-14K dataset

For each item in the dataset, we apply the methodology described in chapter 3 to generate its GenreVis_3D and Visual_3D representations. After applying PCA, the dimensions of the Visual_3D_mean and the Visual_3D_median representations of the items are 75 and 400, respectively. The size of the GenreVis_3D_fc3 and genre representations is equal to 18.

4.1.2. Videoland Dataset

The Videoland dataset has quite different properties from the MMTF-14K dataset. It has a much larger number of users and ratings, the number of its items is much smaller, its ratings derive from explicit and implicit user feedback and it includes not only movies but also series.

The Videoland dataset is composed of ratings from 100,000 randomly selected users to a subset of 1,600 items, with the condition that each user has rated more than 50 items. In contrast to the MMTF-14K, the ratings of the users are not only explicit. Videloand has in place a way to combine implicit and explicit ratings in a 0.1-1 scale. This is done through a series of functions, which map the viewing completion rate of the items to a number between 0.2 and 0.7. The completion rate for the movies refers to the minutes of the item watched by the user over the total duration of the item, whereas for the series the system considers the number of watched episodes over the total number of episodes. The ratings of the items reflect the probability that a user would like and watch an item. In the case that the user rates the item explicitly using the thumbs-up and thumbs-down buttons, the completion rate is not taken into account. Thumbs-down and thumbs-up actions are converted to 0.1 and 1 ratings, respectively. The way that Videoland uses the viewing time for the generation of implicit ratings is crucial for the performance of its recommender systems since it addresses one of the biggest issues in the recommendation task, which is the sparsity of the ratings matrix due to the typical unwillingness of users to provide explicit feedback.

$$r_{ui} = f_{ui}(x) = \begin{cases} 0.1, & x \text{ is a 'dislike'} \\ g_{ui}(x), & x \text{ is completion rate}, \quad i \text{ is a movie} \\ h_{ui}(x), & x \text{ is the number of episodes streamed}, \quad i \text{ is a series} \\ 1, & x \text{ is a 'like} \end{cases} \tag{3}$$

Another important characteristic of the Videoland dataset is that includes not only movies but also series. For every series, the system considers one item that refers to all the series' seasons and episodes. At this point, we should mention that for the items that the trailer was not available, representative video clips were collected manually from YouTube to provide the necessary visual content for our approach. This process was followed for almost all the series items.

For each item in the dataset, we apply the methodology in chapter 3 to generate its GenreVis_3D and its Visual_3D representations. After applying PCA, the dimensions of the Visual_3D_mean and the Visual_3D_median representations of the items are 70 and 250, respectively. The size of the GenreVis_3D_fc3 and genre representations is equal to 23.

4.2. Offline evaluation

We evaluate the accuracy of the proposed movie recommender system by using the mean average precision (mAP), precision and recall metrics as they focus on the capability of the system to include in the recommendation list items that the user would select to consume, being the appropriate metrics for our experiments. Finally, we perform five-fold cross validation to assess the capability of our approach to generalize in independent datasets.

We split the available dataset into a train and a test dataset using a ratio 80:20. As positive are considered the ratings that are equal or higher than 4 and 0.7 for the MMTF-14K and Videoland datasets, respectively.

The precision metric represents the proportion of the recommended items that are included in the test dataset. The meaning of this metric reflects the success of the system in providing items that meet the user preferences.

The mean average precision metric evaluates how the system performs in terms of precision for various cutoff values indicating the capability of the system to recommend relevant items at the first places in the list. The cutoff value represents the size of the recommendation list. This metric is quite important since we can assume that many of the users will look at the first recommendations and they will not persist to examine the full recommendation list.

The recall metric represents the proportion of the test items that are included in the list of the recommended items. This metric reflects the capability of the system to discover and deliver the items that the user likes.

Computationally, the difference is that the precision refers to the number of the positively rated recommended items over the total number of the recommended items, whereas the recall refers to the number of the positively rated recommended items over the total number of the positively rated items in the test dataset.

Furthermore, we estimate other metrics that could help us to evaluate the quality of our recommendation approach. The additional metrics considered are novelty, diversity and coverage.

The novelty metric reflects the capability to recommend relevant items that the user would hardly discover. It is expressed by the mean popularity rank of the items in the recommendation list. The most popular item has rank equal to the size of the movie dataset. The lower the mean popularity is, the more novel the recommendation list is.

The diversity metric assesses a different aspect, quantifying the capability of the system to recommend items which cover the whole spectrum of the users' preferences. This metric is computed by measuring the intra-list similarity of the recommended items with respect to their genre.

Lastly, the coverage metric expresses the capability to provide recommendations that span the item catalog. It refers to the number of the distinct items recommended over the total number of the items in the catalog.

4.3. Models

The total number of models evaluated for each dataset is 81, namely 72 using the GenreVis_3D representation, 6 using the Visual_3D representation and 3 using the genre representation. This is the result of our endeavor to explore how some key parameters in the models' design affect the recommendation performance. The GenreVis_3D representation used in the recommender models varies on the layer of the DBoS network selected to provide the single representation (pool, fc2, fc2-R, fc3), the number of trailer segments used as input in the DBoS network (20, 50 and 80) and the use of random trailer segments as input in the DBoS network (r) or the use of segments arranged in order starting from the trailer's beginning (f). For the Visual_3D representation, we examine the performance of the models when the mean and the median function are selected for the aggregation of the trailer's segments' features. For genre and all the above mentioned representations, we explore the impact of k in the k-NN recommendation algorithm, examining the use of 10, 64 and 100 neighbors. For example, a model named GenreVis_3D_pool_80f_64 corresponds to a model that uses 64 neighbors in the k-NN algorithm and as content representation the pool layer of the DBoS network in which the 80 first segments of the trailer have been used as input.

4.4. Implementation

The trailer files of both datasets were collected from YouTube. The feature extraction of the video files was implemented in an Amazon EC2 P2 machine with 1 NVIDIA K80 GPU of 12 GB memory using the PyTorch framework. The pre-trained model and the code used for the feature extraction was obtained from the github repository in [58]. The batch size used as input in the network was 32 trailers and in 1 hour the model extracts the features of around 1,5 GB trailer files.

The DBoS network providing the GenreVis_3D representation was also implemented in an Amazon EC2 machine using the PyTorch framework. The batch size for the MMTF-14K was 256 trailers in the training phase of the network and 30 trailers in the phase of the extraction of the layers activations as the movie representations. The respective batch sizes for the Videoland dataset were 90 and 30.

The generation of the baseline representations, Visual_3D and genre, was implemented in a personal laptop using Python. Python was also used in data processing tasks like PCA, standardization and the transformation of the deep segments' features to a form that can be used as input by the DBoS network.

The recommendation algorithm was implemented using the Turi Create platform. Turi Create is an open source framework owned by Apple which provides high performance toolkits that facilitate various machine learning tasks like image classification, object detection, recommender systems and so on. We use the item_content_recommender library to train our content-based models and evaluate it with respect to the accuracy metrics. The evaluation of our models regarding the non-accuracy metrics was implemented using common Python libraries (pandas, numpy and scikit-learn).

4.5. Results

4.5.1. MMTF-14K dataset

Since we evaluate the quality of the recommender systems with respect to many metrics, we select for each representation the models that have the best performance in terms of accuracy and then we additionally evaluate them on how they perform with respect to the diversity, novelty and coverage metrics. In the next stage of our tests, we analyze in more detail the results in order to understand the impact of each parameter on the utility of the recommender system. Having 3 accuracy metrics we select to use as single decision criterion the mAP metric since it considers, not only the accuracy, but also the order in which the relevant recommendations are provided. The best models, with respect to mAP metric, for the genre-aware visual, the pure visual and the genre representations are the GenreVis_3D_pool_80f_10, the Visual_3D_median_10 and the Genre_64, respectively. Figure 4.1 shows the mAP metric at cutoff values 5 and 10 for the most accurate models in each category of the representations.

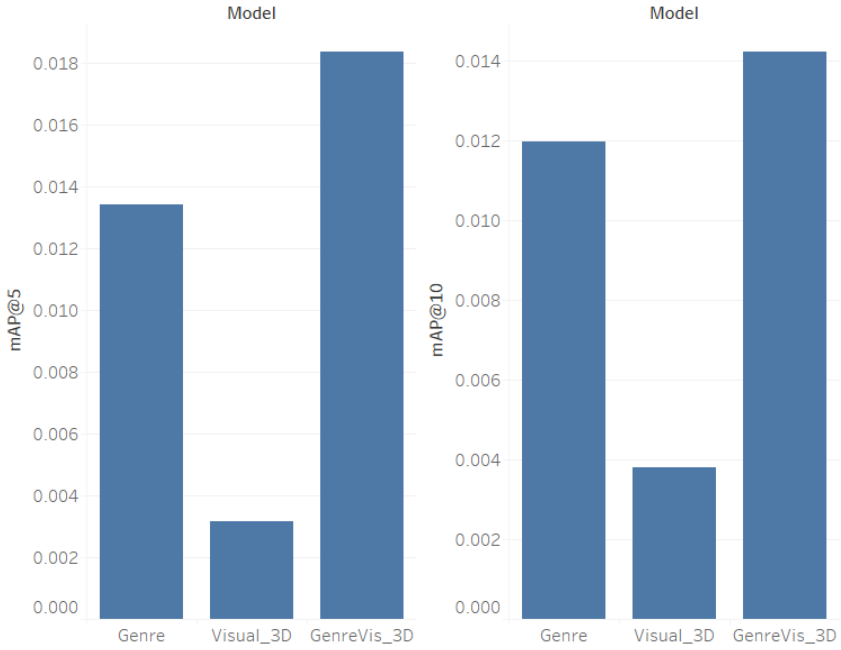

Figure 4.1: mAP@5 and mAP@10 (MMTF-14K)

The GenreVis_3D descriptor provides the best performance in this metric showing its capability to recommend relevant items placing them in the top positions of the recommendation list. However, it should be mentioned that the performance of genre representation is very close to the GenreVis_3D. The figure 4.2 shows the precision-recall metrics for the models, confirming the capability of the genre-aware representation to provide accurate recommendations. The genre representation provides very good results for high cutoff values, whereas the accuracy of the model based on the Visual_3D descriptor is very low compared to the other two approaches. At this point it is worth mentioning that GenreVis_3D descriptor is more accurate when the recommendation list is small, and that the genre performance increases significantly along with the size of the list.

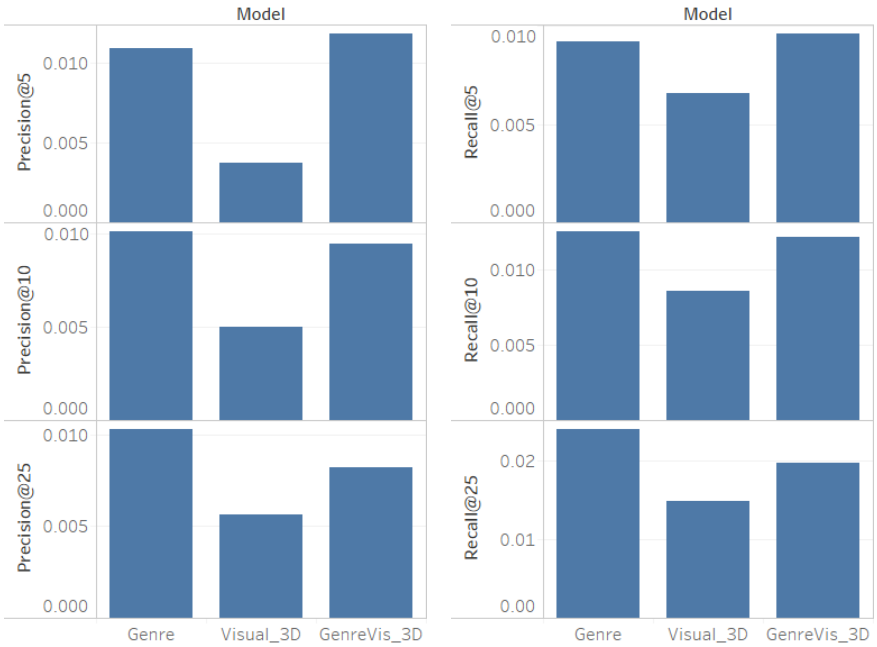

Figure 4.2: Precision – Recall (MMTF-14K)

The figure 4.3 presents clearly the superiority of the Visual_3D and the GenreVis_3D descriptors against genre in providing diverse recommendations which, as the research suggests, it is a property in recommender systems strongly connected to user satisfaction [59] [60]. The diversity of GenreVis_3D and Visual_3D descriptors is almost 4 and 5 times higher than the diversity provided by genre respectively for cutoff value equal to 25.

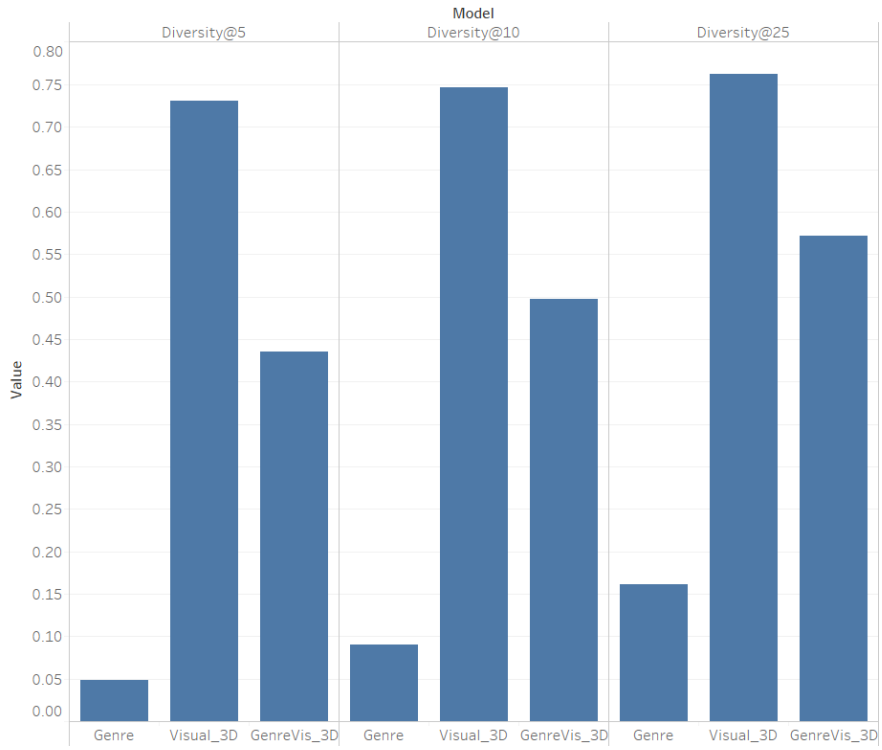

Figure 4.3: Diversity (MMTF-14K)

Regarding novelty, the results in Figure 4.4 illustrate that genre recommendations concern items with much higher popularity than the ones provided by GenreVis_3D and Visual_3D representations, making it less useful for the user and the movie provider. Our statement is based on the claim that the important benefit, in using recommender systems, is that users find relevant items that otherwise would have not found. In contrast, popular items are widely promoted, and it is rather easy for such an item to be found.

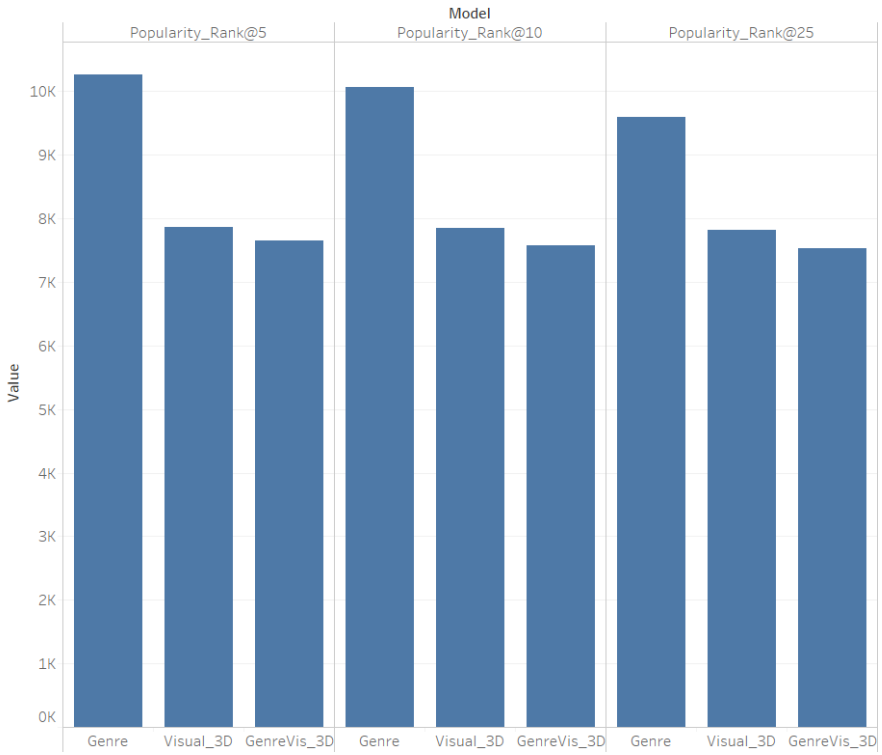

Figure 4.4: Popularity rank (MMTF-14K)

Finally, the results in coverage (Figure 4.5) show that more products of the catalog are considered when the GenreVis_3D and Visual_3D representations are favored against genre in the recommendation setting.

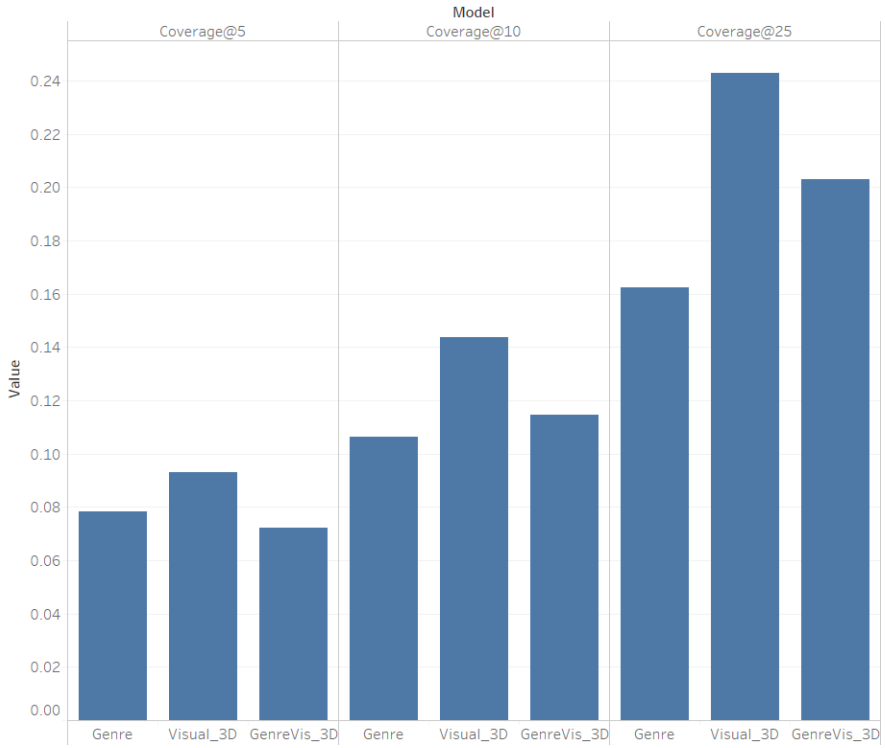

Figure 4.5: Coverage (MMTF-14K)

To conclude, the proposed representation GenreVis_3D provides very similar accuracy with genre providing however at the same time much better performance in other quality metrics like diversity, novelty and coverage. The Visual_3D representation has the best performance in all the non-accuracy metrics but its poor performance in accuracy metrics doesn't allow us to conclude that it is more beneficial than GenreVis_3D since recommending items that fit the user preferences is perhaps the most important requirement for a recommender system. Tables 4.2 and 4.3 present the accuracy and non-accuracy evaluation metrics on the MMTF-14K dataset.

	GenreVis_3D_pool_80f_10	Visual_3D_median_10	Genre_64
mAP@5	0.0183	0.0031	0.0134
mAP@10	0.0142	0.0038	0.0120
Precision@5	0.0118	0.0038	0.0109
Precision@10	0.0094	0.0050	0.0101
Precision@25	0.0082	0.0056	0.0103
Recall@5	0.0096	0.0066	0.0092
Recall@10	0.0121	0.0086	0.0125
Recall@25	0.0197	0.0149	0.0240

Table 4.2: Accuracy metrics (MMTF-14K)

	GenreVis_3D_pool_80f_10	Visual_3D_median_10	Genre_64
Diversity@5	0.44	0.73	0.05
Diversity@10	0.50	0.75	0.09
Diversity@25	0.57	0.76	0.16
Popularity_rank@5	7648.59	7857.48	10265.27
Popularity_rank@10	7570.81	7852.24	10070.51
Popularity_rank@25	7527.41	7823.85	9596.00
Coverage@5	0.07	0.09	0.08
Coverage@10	0.11	0.14	0.10
Coverage@25	0.20	0.24	0.16

Table 4.3: Non - accuracy metrics (MMTF-14K)

4.5.2. Videoland dataset

Finding the results of the MMTF-14K dataset quite interesting and promising, we follow the same procedure to evaluate our approach in a dataset with different statistics, like Videoland dataset and understand its generalization capabilities.

Similar to the MMTF-14K dataset, we focus on the mAP metric to select which models will represent each representation approach and will be evaluated on all the available metrics.

From the beginning of our analysis, we find out that the most accurate models do not share the same parameters with the models in the MMTF-14K dataset.

The best models with respect to mAP metric for the genre-aware visual, the pure visual and the genre representations are the GenreVis_3D_fc3_20r_64, the Visual_3D_mean_64 and the Genre_100, respectively. Surprisingly enough, we see that the best models in the 2 datasets are different across all the design parameters.

The figure 4.6 shows that the GenreVis_3D improves significantly the accuracy of the system compared to the Visual_3D and genre representations. The genre representation presents the worst performance in this metric.

The GenreVis_3D representation in both datasets seems to be capable of not only identifying which items are relevant, but also ranking them properly according to their degree of relevance.

Looking at the precision and recall metrics in figure 4.7 we confirm the superiority of the GenreVis_3D representation against the other two approaches in the task of predicting correctly the users' preferences.

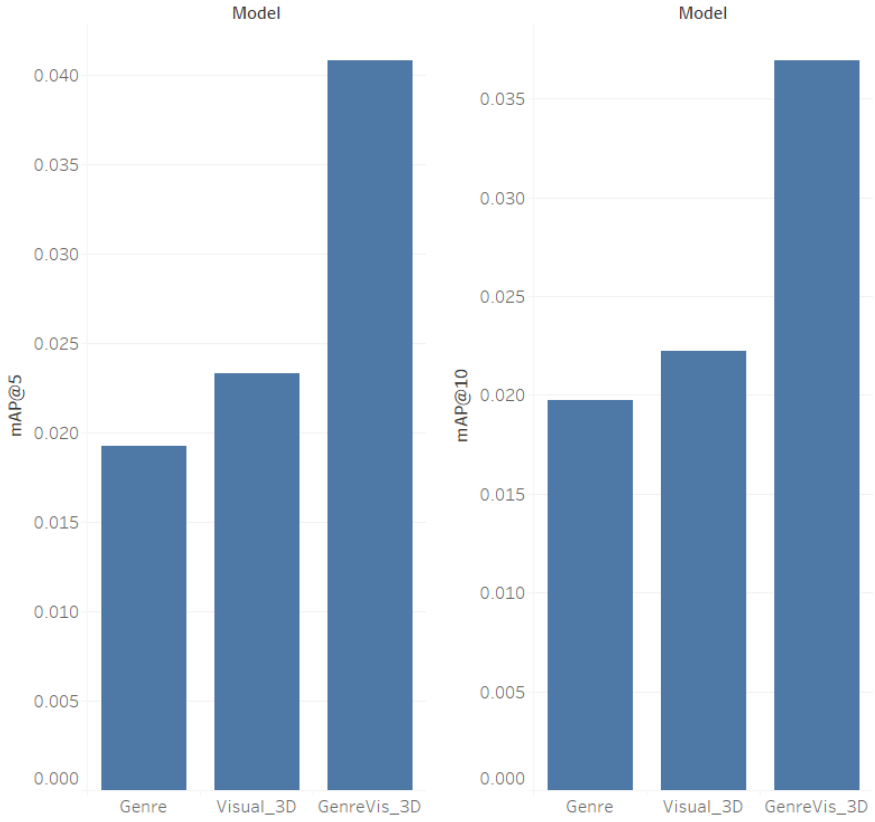

Figure 4.6: mAP@5 and mAP@10 (Videoland)

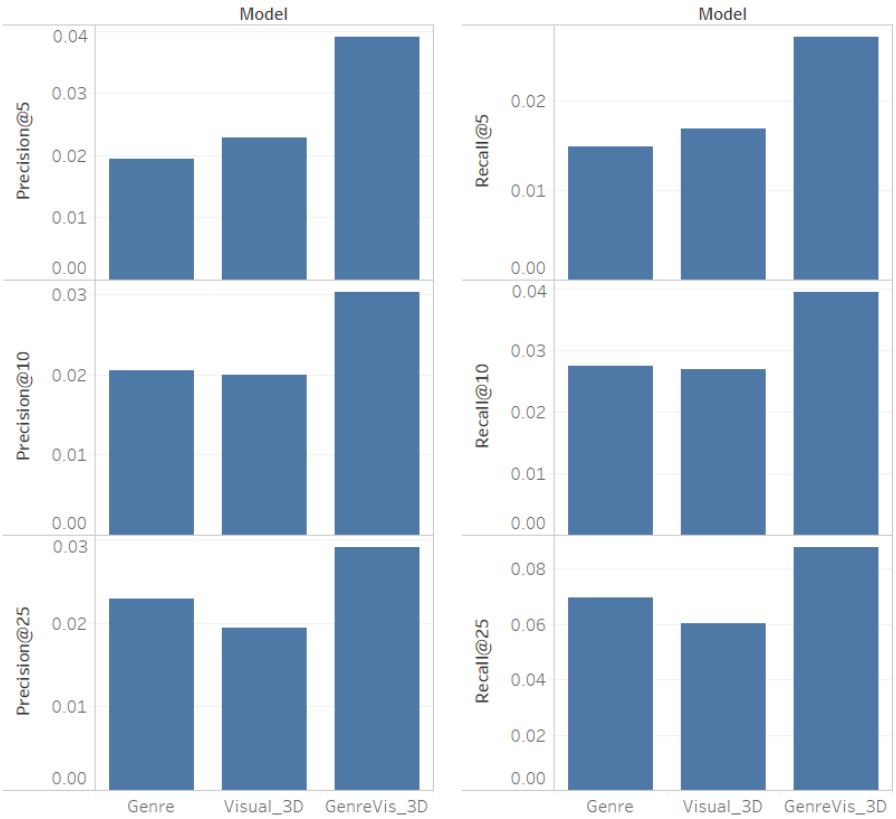

Figure 4.7: Precision – Recall (Videoland)

Regarding the diversity (figure 4.8), as expected, the best by far performance is shown by Visual_3D representation since it is absolutely independent from genre, attribute by which the diversity is defined. The GenreVis_3D representation, not surprisingly, improves slightly the extremely poor performance of the genre representation.

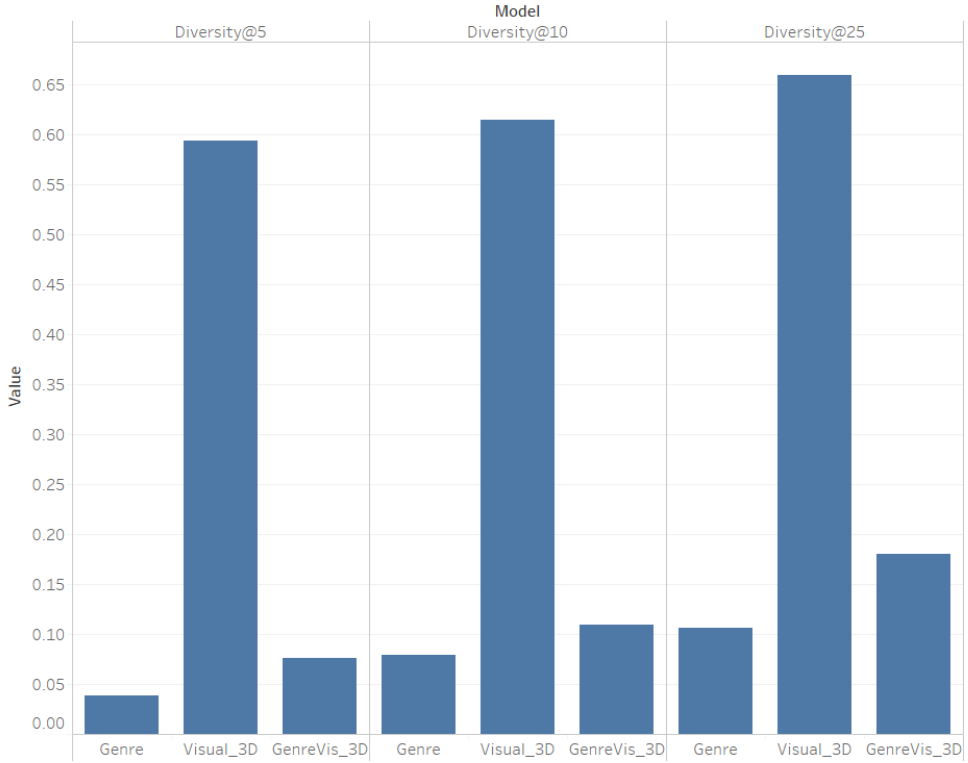

Figure 4.8: Diversity (Videoland)

Regarding the novelty, we see in figure 4.9 the same pattern with the diversity case. The Visual_3D descriptor recommends the less popular items whereas the GenreVis_3D representation performs slightly better than genre, not achieving however to recommend items of low popularity. Figure 4.10 shows that the performance in the coverage metric is quite similar for all the representations.

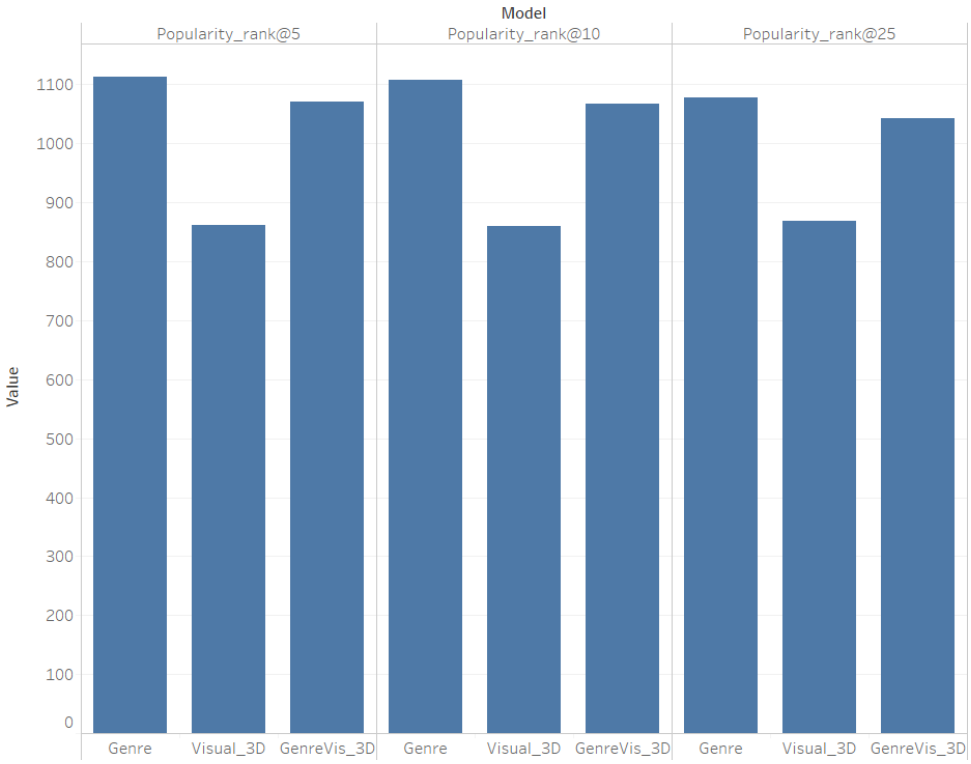

Figure 4.9: Popularity rank (Videoland)

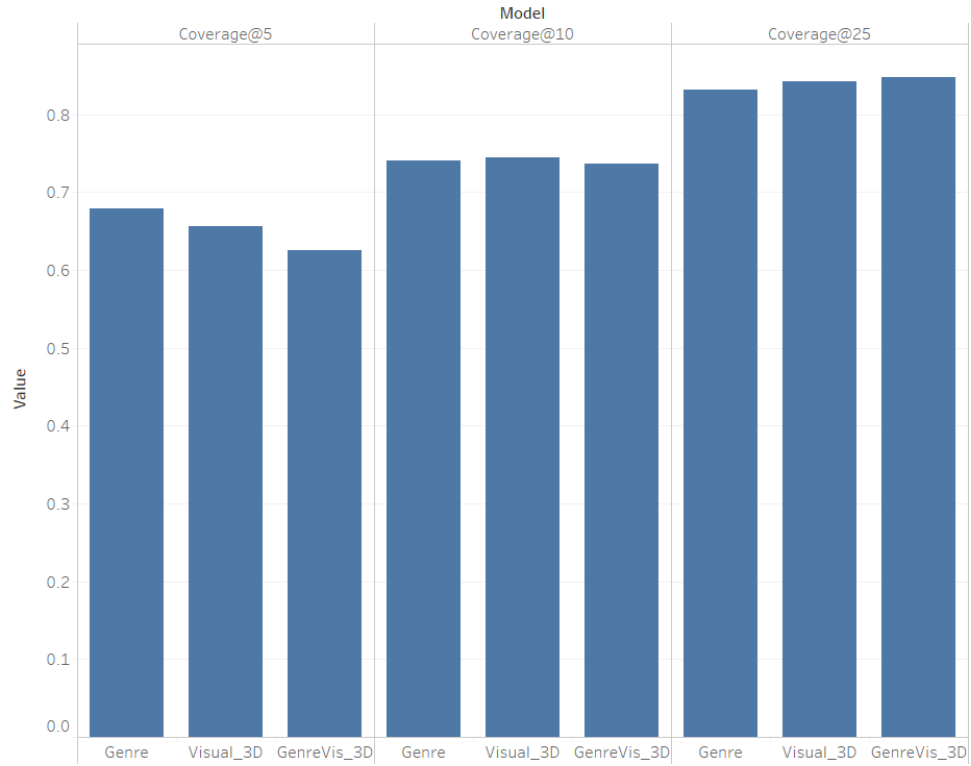

Figure 4.10: Coverage (Videoland)

To conclude, the results for Videoland dataset are quite different to the findings in the MMTF-14K dataset. The proposed representation GenreVis_3D delivers by far the best performance with respect to accuracy but regarding the non-accuracy metrics, it proves to

be less powerful than the Visual_3D representation. The Visual_3D representation outperforms the other approaches with respect to diversity, novelty and coverage achieving additionally competitive accuracy to genre. Tables 4.4 and 4.5 present the accuracy and non-accuracy evaluation metrics on the Videoland dataset.

	GenreVis_3D_20r_64	Visual_3D_mean_64	Genre_100
mAP@5	0.041	0.023	0.019
mAP@10	0.037	0.022	0.020
Precision@5	0.039	0.023	0.019
Precision@10	0.030	0.020	0.020
Precision@25	0.029	0.019	0.023
Recall@5	0.027	0.017	0.015
Recall@10	0.039	0.027	0.027
Recall@25	0.088	0.060	0.070

Table 4.4: Accuracy metrics (Videoland)

	GenreVis_3D_20r_64	Visual_3D_mean_64	Genre_100
Diversity@5	0.08	0.59	0.04
Diversity@10	0.11	0.61	0.08
Diversity@25	0.18	0.66	0.11
Popularity_rank@5	1070.62	862.48	1113.65
Popularity_rank@10	1066.92	860.48	1107.81
Popularity_rank@25	1042.95	869.32	1077.36
Coverage@5	0.62	0.66	0.68
Coverage@10	0.74	0.74	0.74
Coverage@25	0.85	0.84	0.83

Table 4.5: Non-accuracy metrics (Videoland)

At this point it is worth mentioning that the GenreVis_3D approach is quite flexible in providing models that could focus on the performance of the system in particular metrics. In Videoland dataset for instance, the model GenreVis_3D_pool_20f_64 model could be favored against GenreVis_3D_fc3_20r_64 to provide more diverse recommendations with competitive accuracy to the genre and Visual_3D representations, if the improvement in diversity is considered to be more important than the improvement in the accuracy. The performance in accuracy (cut-off value 10) and diversity of the 3 approaches with the new

GenreVis_3D model is presented in the figures 4.11 and 4.12. Obtaining a representation from a different layer of the DBoS network, we could have a genre-aware representation that improves significantly diversity at the expense of the delivered accuracy. Table 4.6 shows the results with the new model.

	GenreVis_3D_pool_20f_64	Visual_3D_mean_64	Genre_100
mAP@10	0.022	0.022	0.020
Precision@10	0.020	0.020	0.021
Recall@10	0.027	0.027	0.027
Diversity@5	0.32	0.59	0.04
Diversity@10	0.35	0.61	0.08
Diversity@25	0.42	0.66	0.11

Table 4.6: Evaluation metrics with GenreVis_3D_pool model (Videoland)

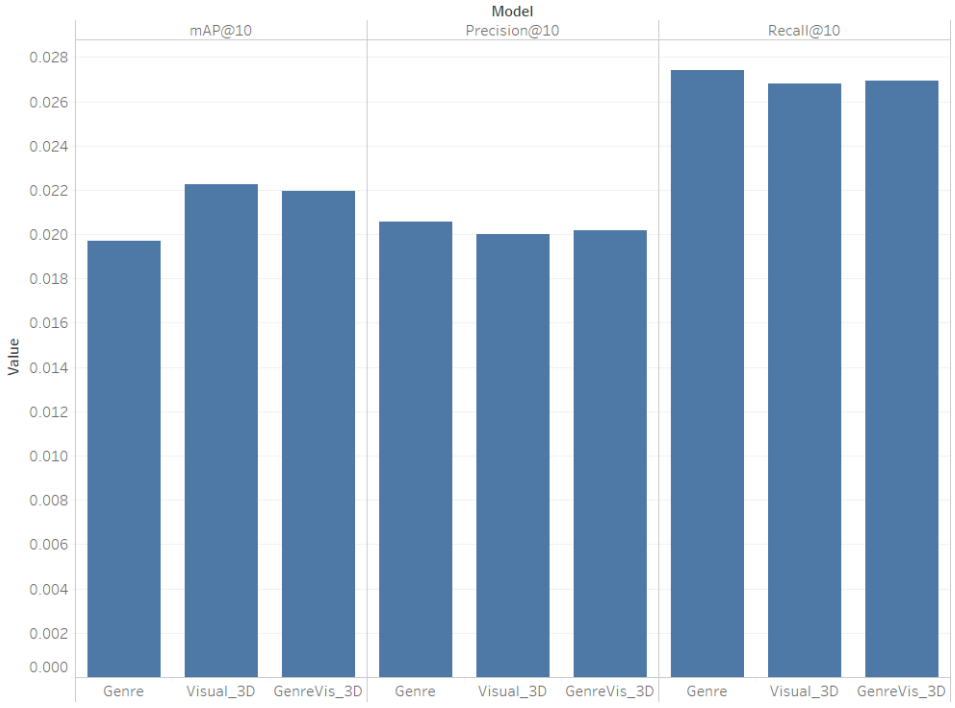

Figure 4.11: Accuracy with GenreVis_3D_pool model (Videoland)

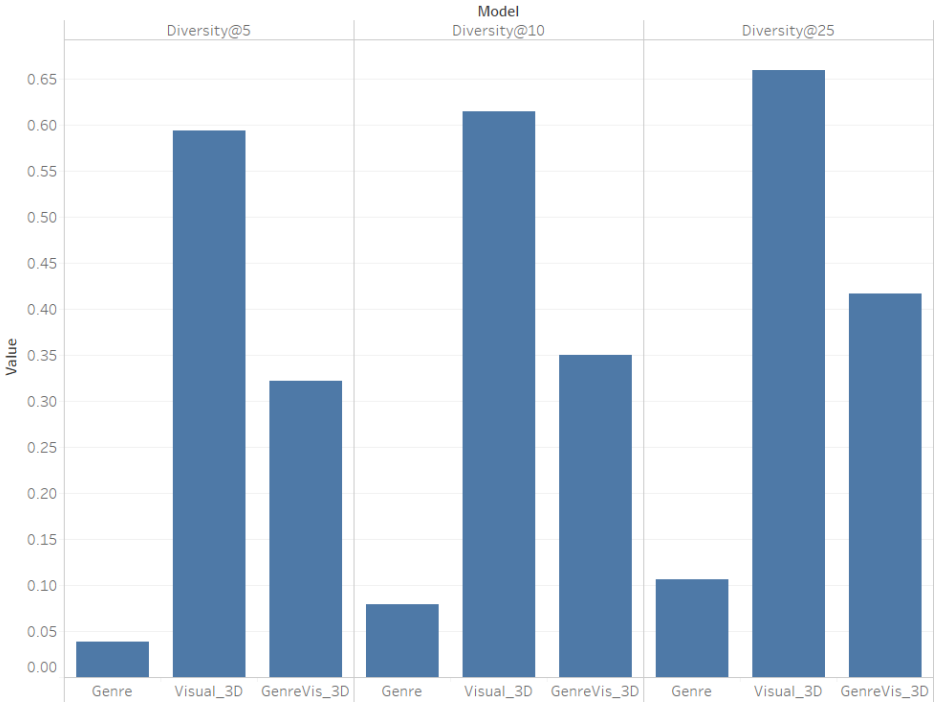

Figure 4.12: Diversity with GenreVis_3D_pool model (Videoland)

4.6. The impact of GenreVis_3D parameters

The generation of the GenreVis_3D representation requires the determination of 3 parameters, namely the layer of the DBoS network, the number of trailer's segments used as network's input and the way the segments are selected (randomly or in order). Examining also the values 10, 64 and 100 for the number of neighbors in the k-NN algorithm, we generate in total 72 GenreVis_3D models and we evaluate how each parameter affects their performance in the recommendation task. Our purpose is to identify how the parameters determining the representation should be chosen, in order to achieve the desirable recommendation utility.

4.6.1. MMTF-14K

Figures 4.15, 4.16, 4.17 present how the layers and the number of the input segments affect the performance of the recommender system with respect to mAP, precision, recall, diversity, novelty and coverage for the MMTF-14K dataset, when ordered or random input segments are used respectively. The figures refer to the cutoff value 10, since the other values do not offer additional insights to the analysis. We report the maximum value for each parameter and metric.

Regarding the layers (figure 4.15), the pool layer provides the best performance with respect to the accuracy, diversity and novelty. On the other hand, the fc3 layer achieves the best performance with respect to coverage.

The figure 4.16 shows that the use of a big number of segments is beneficial to the accuracy and coverage of the system, whereas the small number of segments seems to benefit diversity and novelty. That holds true for all the layers of the network, except for the fc3 layer.

Finally, the figure 4.17 suggests that the models using segments in order outperform the models using random segments in all the evaluation metrics except the case that the fc3 layer is used. In that case, the performance of the random segments is slightly better.

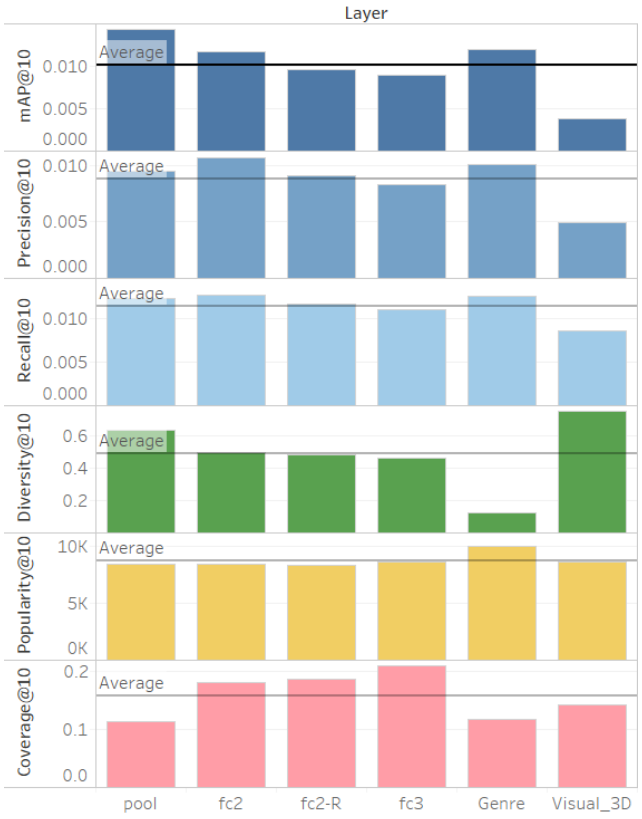

Figure 4.15: Impact of layer (MMTF-14K)

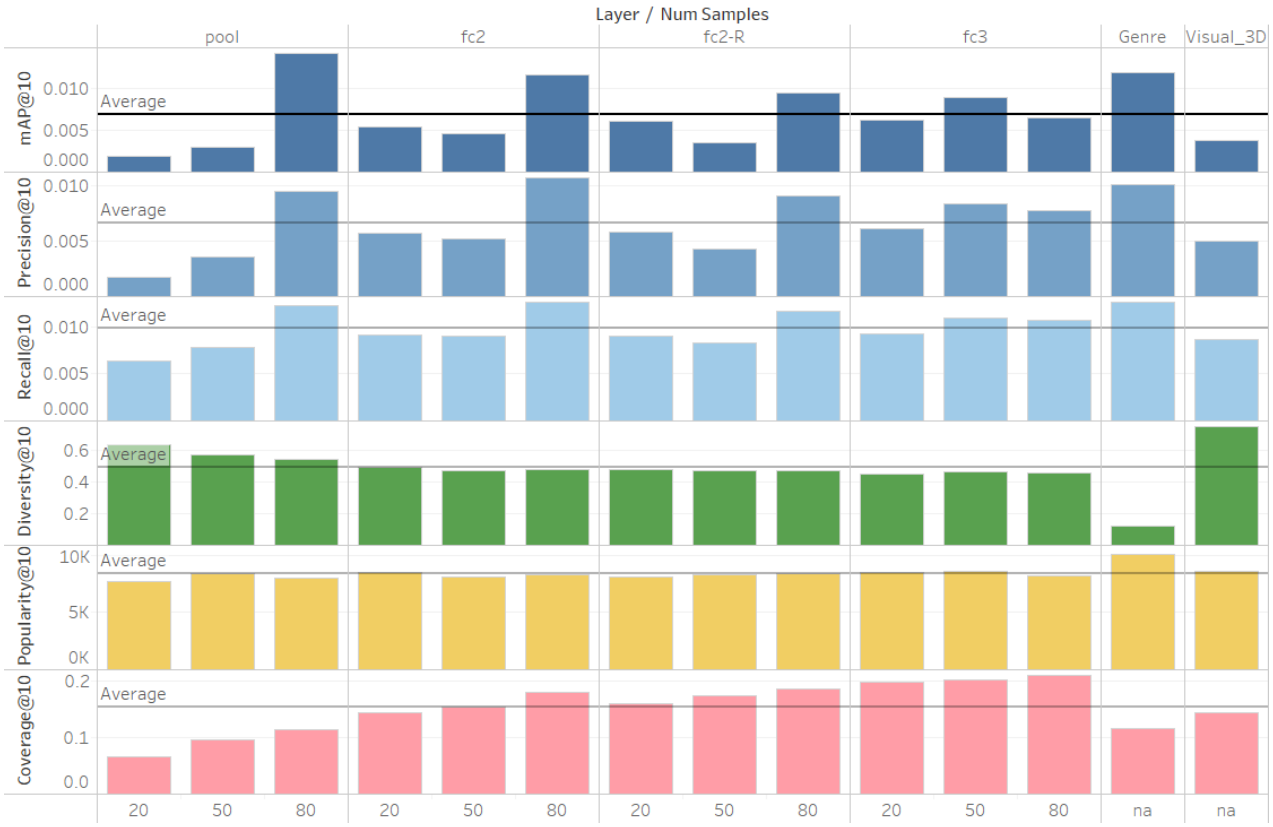

Figure 4.16: Impact of number of segments (MMTF-14K)

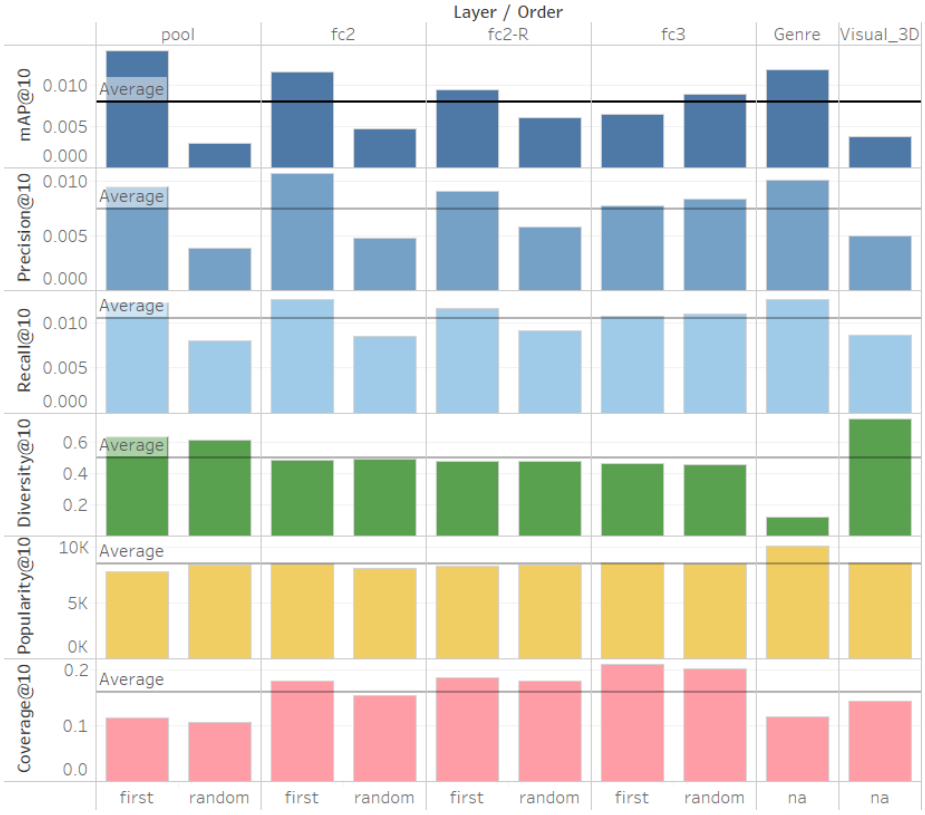

Figure 4.17: Impact of order of segments (MMTF-14K)

4.6.2. Videoland dataset

Figures 4.18, 4.19, 4.20 present the impact of the parameters determining the GenreVis_3D representations on the recommendation utility in the Videoland dataset. It is quite interesting that the results in Videoland dataset provide contradictory insights to the ones coming from MMTF-14K dataset at many points. Similar to the MMTF-14K, we report the maximum value for each parameter and metric.

Regarding the layer parameter, we observe the most significant contradiction between the 2 datasets. In MMTF-14K, the most accurate recommendations are provided by the pool layer, which is located closer to the visual input, whereas in Videoland dataset the best accuracy is achieved by the fc3 layer, which represents the layer that is more connected to the genre labels of the movies. In terms of non-accuracy metrics, the pool layer seems to be capable of providing more diverse and novel recommendations, similar to the MMTF-14K. Finally, both datasets suggest that the recommendations generated by the fc3 layer cover better the item catalog.

Except for the pool layer, selecting input segments in order doesn't seem to help the accuracy of the recommendations. In contrast to the MMTF-14K dataset, the performance in mAP, precision and recall metrics is enhanced when the trailers' segments are randomly selected. Even more interesting, the best performance is achieved when only 20 segments of the trailer are used. With respect to the non-accuracy metrics, similar to the MMTF-14K dataset, the pool layer seems to provide the most significant improvements, but requiring this time, 20 instead of 80 segments. In contrast to the MMTF-14K, an increase in the number of segments doesn't lead to an increased recommendation quality.

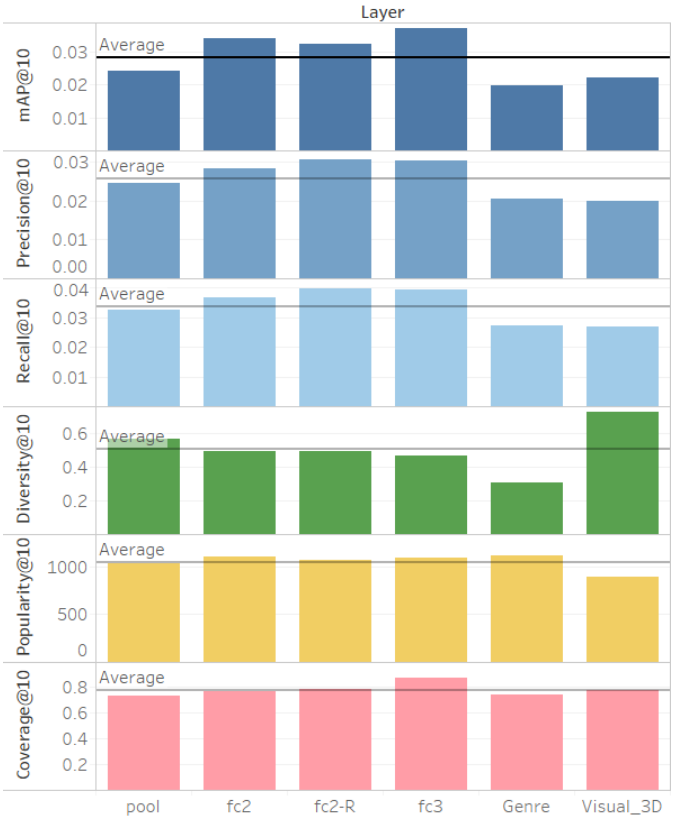

Figure 4.18: Impact of layer (Videoland)

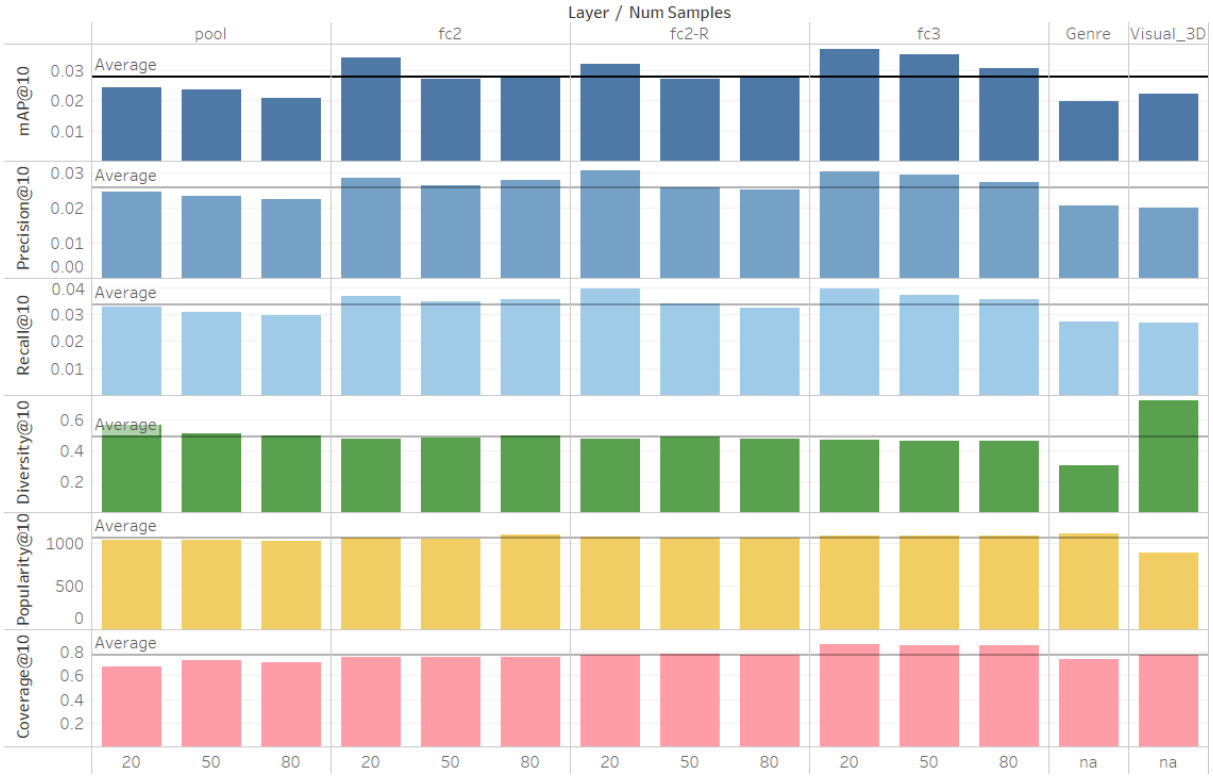

Figure 4.19: Impact of number of segments (Videoland)

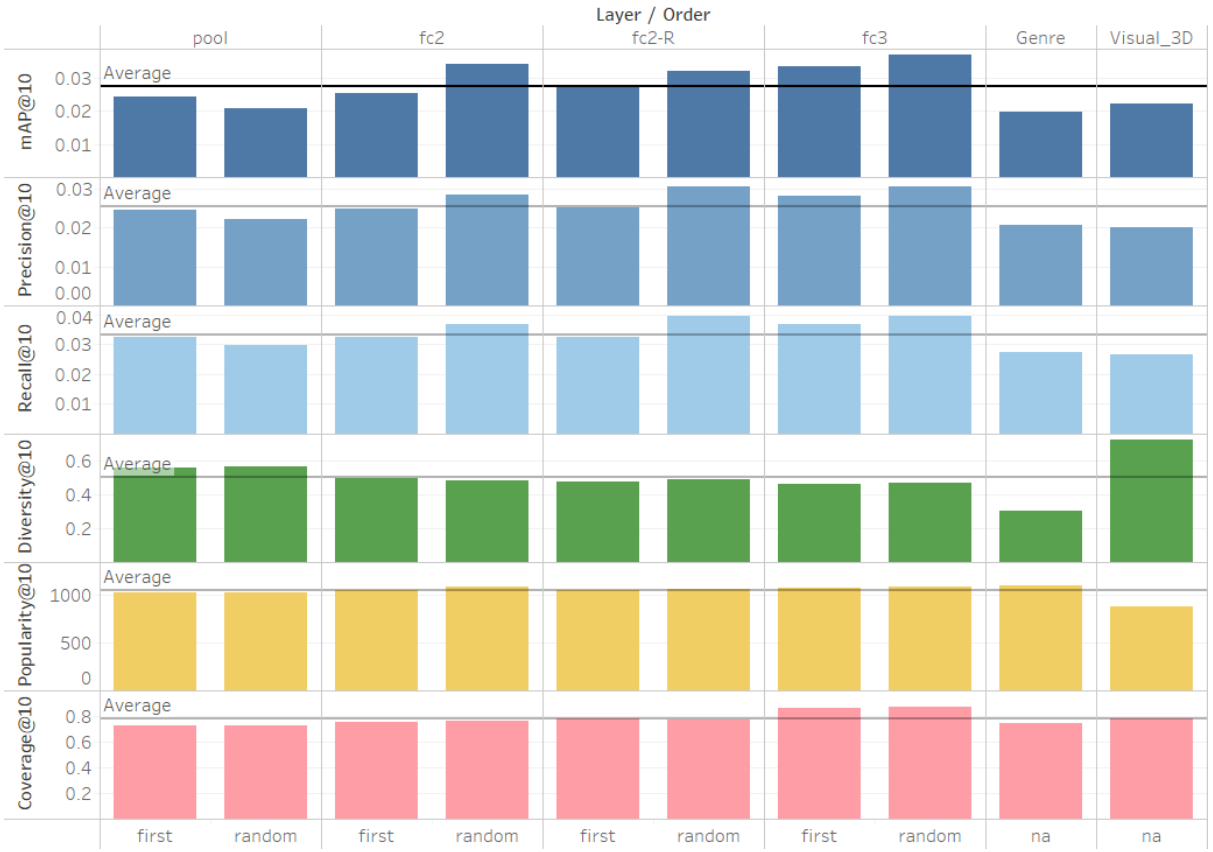

Figure 4.20: Impact of order of segments (Videoland)

4.7. The impact of the neighborhood size in k-NN

We evaluate the proposed representation in the recommendation task using a pure content-based filtering system based on the k-NN approach. Our experimentation towards the optimal number of neighbors revealed its significant effect on the utility of the system's recommendations.

Figure 4.21 shows with clarity that the number of neighbors affects significantly the capability of the representations to provide recommendations with desirable properties in the MMTF-14K dataset. We see that the best performance for the visual representations (GenreVis_3D and Visual_3D) is achieved with 10 neighbors whereas for genre a much larger number is required. It is also interesting that the larger number of neighbors is not translated automatically into better performance. Looking at the huge improvement in the accuracy of the genre representation when 64 neighbors are used, we realize how easily disregarding this parameter can misguide the evaluation procedure. It can be assumed that the need for a big number of neighbors reflects the weakness (limited informativeness) of the representation to identify the strong similarities between the items. As a result, a bigger number of similar items is required to provide the actual similar items that could contribute to the rating prediction task.

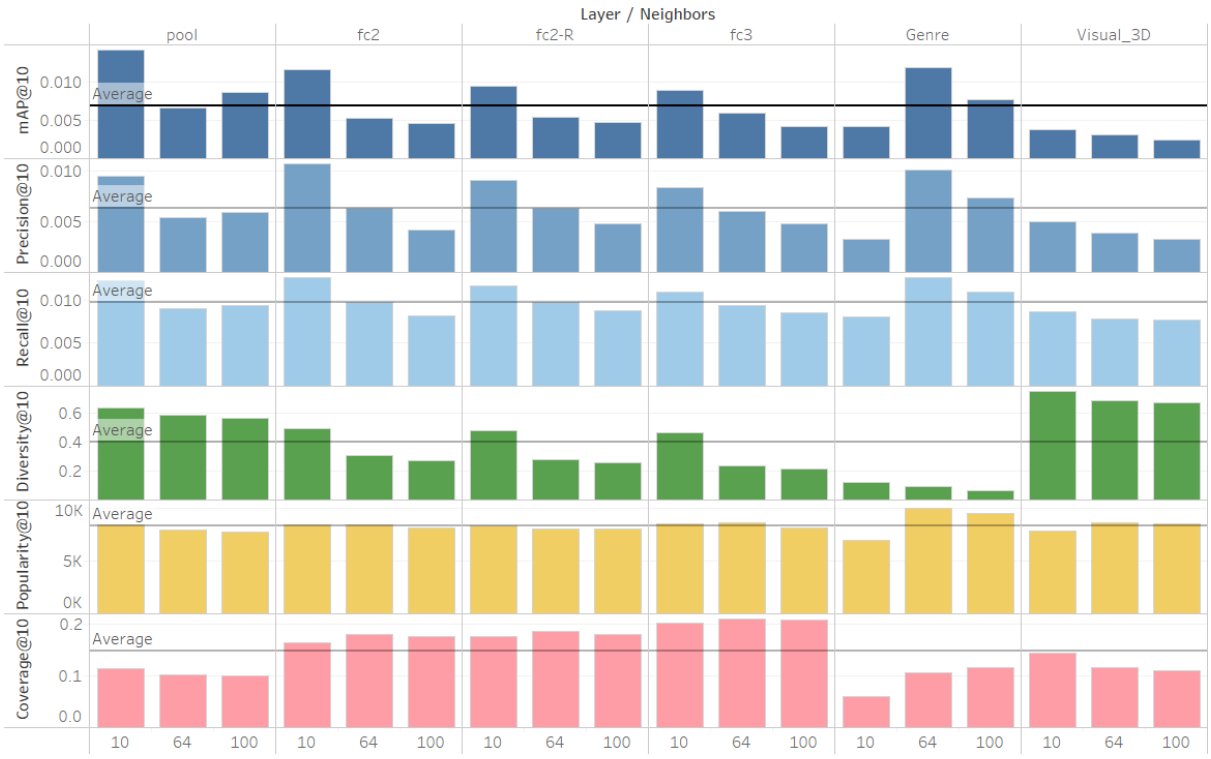

Figure 4.21: Impact of k (MMTF-14K)

The results of Videoland dataset in figure 4.22, show that all the visual representations (GenreVis_3D and Visual_3D) and the genre representation provide their best performance for 64 and 100 neighbors respectively. Additionally we see that for the GenreVis_3D representation the diversity with 10 neighbors seems to be independent of the other parameters. The last finding may indicate that for some users the system is not able to fill their recommendation list with items of non-zero prediction scores, and many of the items in

the list are selected randomly. For k=10, the rating score of a user to an item can be equal to 0 if the user has not rated its 10 most similar items. We assume that this is less likely in the cases that the system estimates accurately the similarities between the items, provided that the item similarity is associated with the user choices. The randomness injected in the recommendation list, increases the diversity in a way that doesn't improve the system's accuracy. Increasing the number of neighbors, reduces the random items, enabling us to evaluate better the diversity and accuracy capabilities of the representations.

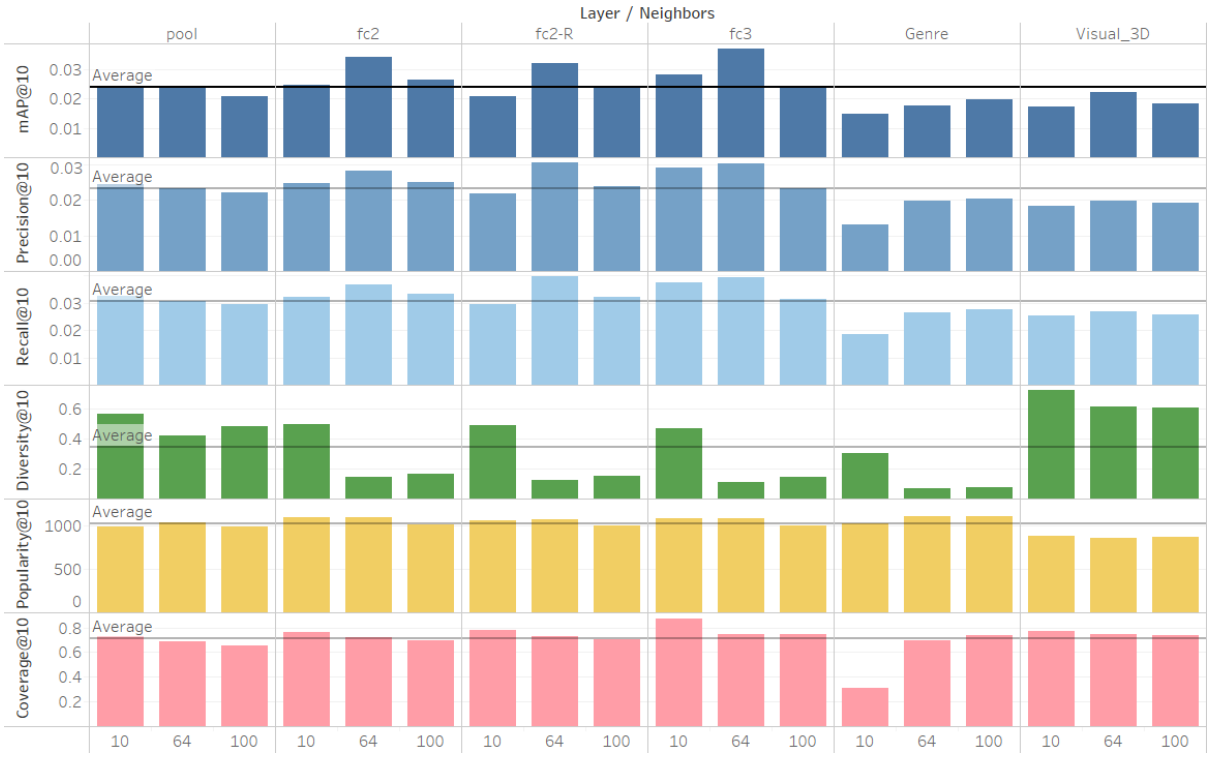

Figure 4.22: Impact of k (Videoland)

5

Online experiment

We conduct an online experiment on the Videoland platform, to evaluate the user perceived quality of recommendations, provided by a pure content-based recommender system, using the proposed genre-aware movie descriptor. To this end we compare our proposed representation against the pure visual representation derived by the pre-trained 3D ConvNets and the genre representation.

The models used in the online experiment are based on the algorithm described in the chapter 3. Regarding the GenreVis_3D representation, we are testing 4 models that differ in the network layer used to provide the representation. Guided by the results of the offline evaluation on Videoland's dataset, we decide to evaluate the model GenreVis_3D_fc3_20r_64, due to its superior performance with respect to the accuracy. Additionally, we include the models GenreVis_3D_pool_20r_64, GenreVis_3D_fc2_20r_64 and GenreVis_3D_fc2-R_20r_64 in the experiment, to examine how the layers of the DBoS network affect the recommendation quality. Regarding the Visual_3D and the genre representations, we select the models Visual_3D_mean_64 and the Genre_100 due to their performance with respect to the accuracy metric. We create 6 groups from a randomly generated subset of the Videoland's users, and we assign to each one of them, one of the six models to generate recommendations. The users are assigned to the groups in a random way using a hash function based on their profile id. Each group has around 17000 active users and the experiment ran for 15 days. The items included in the online experiment are the same 1,600 as the ones included in the offline Videoland dataset. The recommender models are trained daily using the users' historical data. This historical data is selected such that it goes back one year, starting from the day before the date, in which the models are trained. Upon visiting the platform's homepage, users are presented with 25 personalized recommendations, which they have not yet interacted with, from the same subset of active items used in the offline experiments.

5.1. Online evaluation

We evaluate the performance of our models in terms of 3 metrics: the average conversion rate per user, the average viewing time per user and the average number of streams per user. In our computations, only the active items of Videoland's offline dataset are considered. The conversion rate metric expresses the capability of the system to recommend items that the users would watch and eventually like. This metric is calculated by finding the number of items that were recommended and consumed, over the total number of items consumed per user per day.By the term conversion we mean the items that were recommended and watched by the user on a certain date. For instance, if one user watched three items on a certain date and two of them were included in her recommendations, the number of conversions equals to 2 and the conversion rate is equal to

0.66. If none of the watched items were included in the recommendations, the conversion rate equals to 0. We include in our computations, only the items which have a completion rate of at least 85% assuming that the low completion rate signals that the item was not interesting/relevant to the user. Regarding the other 2 metrics, the evaluation considers the total minutes of content and the number of streams that were watched per user in the period of the experiment. In our computations, we exclude items that the user has interacted with, before the experimentation period, in order to exclude the viewing time coming from series that the user started to watch before the recommendations of our models are available. Those metrics indicate how the recommender models affect the user satisfaction and consequently the usage of the platform.

We examine in our evaluation 3 scenarios. According to the first one, we evaluate our system taking into account all the items, including both series and movies. In the second and the third scenarios, we filter out the series and the movies respectively. We believe that the poor annotation of the series regarding their genre information, and the use of short clips, instead of trailers, for their visual content, is likely to affect in a negative way the capability of the DBoS network to learn a genre-aware representation, which in turn, degradates the quality of the recommendations. The movies using the official trailers and having more informative genre labels could provide better evaluation insights of our approach.

We are using one-way analysis of variance (ANOVA) and Tukey's multiple comparison tests at significant level a = 0.05 to evaluate if there is difference among the group means.

5.2. Results

The results for all scenarios are presented in the tables 5.1, 5.2 and 5.3. Figures 5.1, 5.2 and 5.3 show the percentage difference of the tested models from the genre model. For the viewing time and the number of streams metrics, the results are normalized based on the values of the genre representation. We see that our hypothesis that the genre-aware representation approach performs better for movies than for series seems to be valid.

The results in table 5.2 show that the recommender models based on the genre-aware representation have higher, but not statistically significant, scores than the Visual_3D and the genre representations with respect to the viewing time and the number of streams, when the users are interested in movies. The Visual_3D representation has the best performance with respect to the conversion rate, but this performance is not reflected in the other metrics.

The results for the series are similar to the results for the scenario including all the items. Tables 5.1 and 5.3 show that the genre has higher scores than the GenreVis_3D and the Visual_3D representations with respect to all the metrics. The very poor performance of the Visual_3D representation for series could be attributed to the fact that the trailers of the series were not available and the short clips that used to provide their visual content are very likely lacking valuable visual information.

The statistical tests show that there are no statistically significant differences between the results regarding the viewing time and the number of streams for all the scenarios. Regarding the conversion rate, in all the scenarios the difference between the Visual_3D representation and all the other representations is significant at the level 0.05. Additionally, in the movies scenario the GenreVis_3D_fc2-R representation outperforms the genre significantly whereas for series the genre outperforms significantly all the other representations except for the GenreVis_3D_fc2-R. The tables show in bold the results that are significantly different than genre. At this point, we should mention that the number of

users per group that interacted with movies is much smaller than the number of users in the series and all items scenario (7000 vs 14000 vs 17000 users).

	Conversion rate per day and user	Total viewing time per user (normalized)	Total number of streams per user (normalized)
Visual_3D	**0.070**	98.39	98.56
GenreVis_pool	0.076	99.47	100.78
GenreVis_fc2	0.076	99.38	98.43
GenreVis_fc2-R	0.081	98.37	98.82
GenreVis_fc3	0.073	99.46	99.34
Genre	0.080	100.00	100.00

Table 5.1: Movies & series

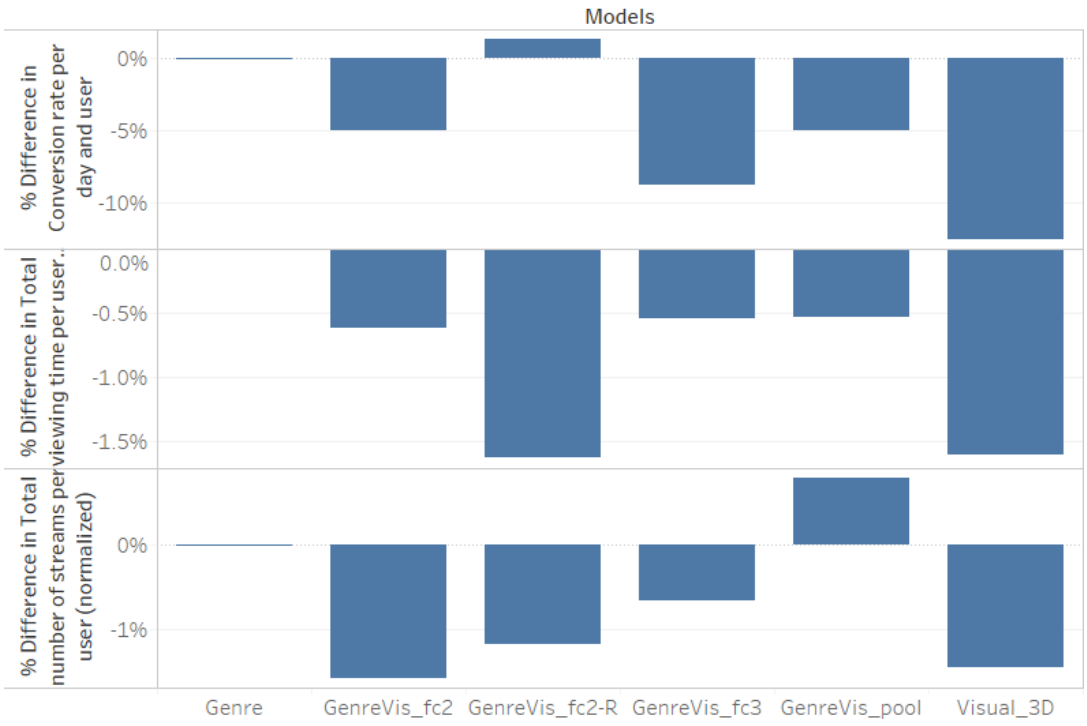

Figure 5.1: Movies & Series

	Conversion rate per day and user	Total viewing time per user (normalized)	Total number of streams per user (normalized)
Visual_3D	**0.098**	100.39	100.58
GenreVis_pool	0.076	99.79	100.00
GenreVis_fc2	0.076	101.29	101.76
GenreVis_fc2-R	**0.075**	102.23	102.35
GenreVis_fc3	0.069	102.67	102.94
Genre	0.066	100.00	100.00

Table 5.2: Movies

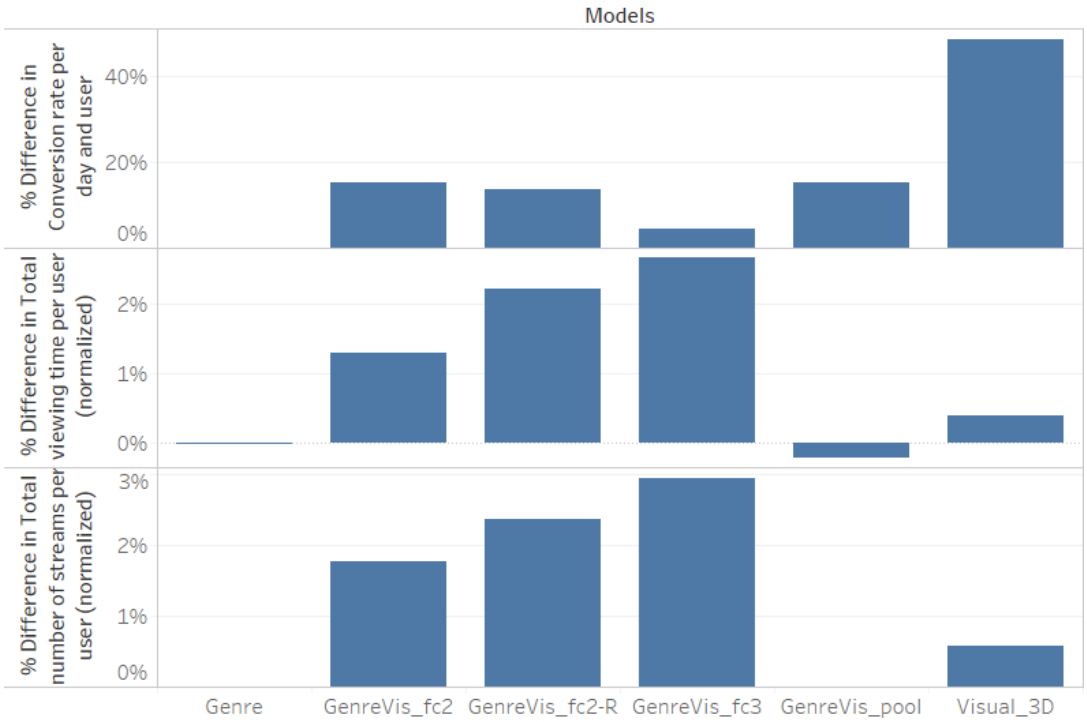

Figure 5.2: Movies

	Conversion rate per day and user	Total viewing time per user (normalized)	Total number of streams per user (normalized)
Visual_3D	**0.062**	98.08	98.94
GenreVis_pool	**0.074**	99.36	100.93
GenreVis_fc2	**0.074**	99.23	98.47
GenreVis_fc2-R	0.080	97.54	98.59
GenreVis_fc3	**0.073**	98.93	99.41
Genre	0.082	100.00	100.00

Table 5.3 : Series

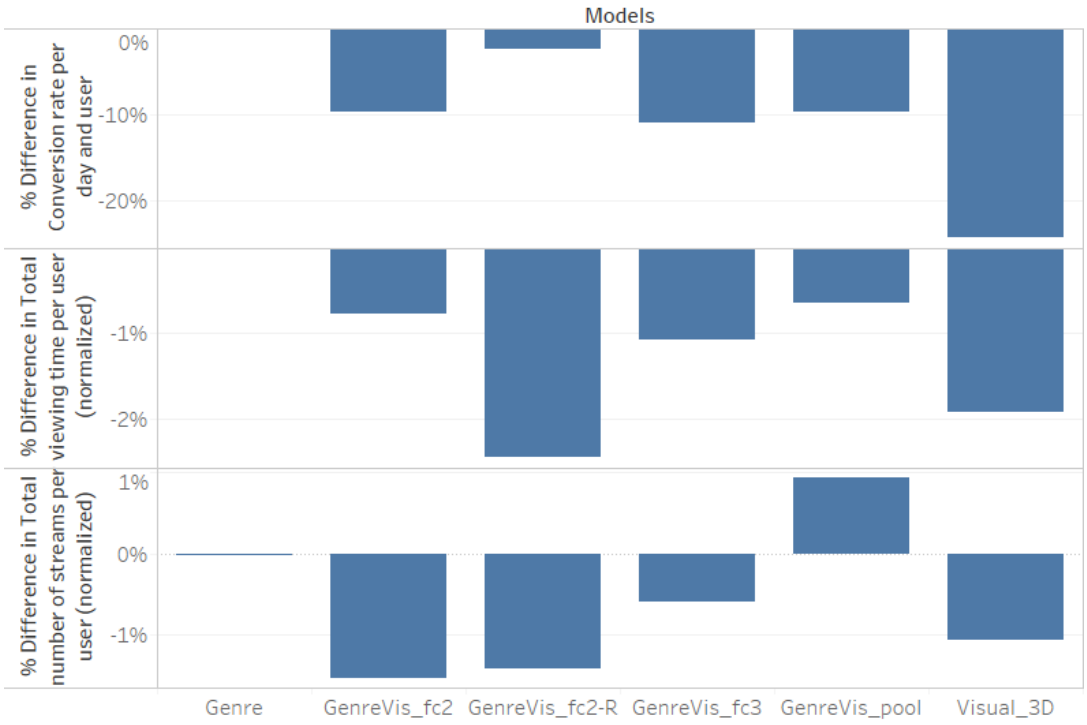

Figure 5.3 : Series

6

Discussion

In this chapter, we discuss the findings of our online and offline experiments, we draw our conclusions regarding our research questions and we suggest future work.

6.1. Offline discussion

We conducted offline experiments with 2 different datasets to evaluate how the use of our proposed representation affects the performance of a movie recommender system, compared with the use of genre and Visual_3D representations. We evaluate our approach on accuracy and non-accuracy metrics.

Accuracy metrics

The results in both datasets prove the capability of the genre-aware representation to provide accurate recommendations. The GenreVis_3D representation outperforms both genre and Visual_3D representations in both datasets. This confirms our hypothesis that a visual descriptor, as the GenreVis_3D, that represents genre-specific concepts of a movie trailer is able to provide users with relevant recommendations. The genre performance is comparable to GenreVis_3D in MMTF-14K dataset but it performs very poorly in the Videoland dataset, where it shows worse results than the second Visual_3D representation.

The results also show that the performance of the GenreVis_3D descriptor depends on the parameters of DBoS network that generates it.

We observe that the model with the best performance in the MMTF-14K dataset utilizes the pool layer of DBoS network. On the contrary, the best performance for the Videoland dataset comes by using the fc3 layer of the network. An explanation could be that the small size of Videoland's dataset does not allow the effective learning of the large pool representation.

The most accurate GenreVis_3D models in MMTF-14K and Videoland datasets differ also in the number and order of the input segments in the DBoS network. The Videoland dataset requires 20 random segments, whereas the MMTF-14K uses 80 first segments. We conclude that increasing the number of input segments does not always increase the accuracy of the system. When a small number of segments is used, random selection of input segments leads, in most cases, to increased accuracy. The genre performs significantly better only against the Visual_3D representation with respect to the onversion rate.

Finally, it is very important to mention that the size of the neighborhood in the k-NN algorithm has a big impact in the genre's performance in the MMTF-14K dataset. Selecting 10 neighbors for all the representations, the GenreVis_3D shows a huge improvement on

genre's performance. However, when using 64 neighbors for genre representation, GenreVis_3D is slightly better than genre. The necessity of using a large number of neighbors is indicative of the limited descriptive power of the representation.

Non-accuracy metrics

Our approach is also evaluated on non-accuracy metrics like diversity, novelty and coverage. The results show that the GenreVis_3D outperforms the genre in all the metrics, providing very good results, especially, when it is based on the pool layer of the DBoS network. However, the best performance in this category of metrics is shown by the Visual_3D representation. The Visual_3D representation, being independent of genre, provides, not suprisingly, the best performance with respect to diversity, in both datasets. Regarding novelty and coverage, the Visual_3D and the GenreVis_3D show comparable performance.

To conclude, we observe that the pool layer of the DBoS pooling network provides GenreVis_3D representations that improve the quality of the recommendations, with respect to both accuracy and non-accuracy metrics.

6.2. Online discussion

The results of the online evaluation contrast the results shown in the offline experiment. The superiority of the GenreVis_3D descriptor against the genre and Visual_3D representations, suggested in the previous section, is not evident in the results of the online experiment. We see that the genre outperforms the Visual_3D respresentation with repect to the conversion rate. However, when only the movies in the system are considered, the results are different. In that case, the GenreVis_3D_fc2_R and Visual_3D representations show better performance than genre with respect to the conversion rate. Regarding the viewing time and the number of streams, there are no statistically significant differences between the results for all the scenarios. The difference in the results between the 2 scenarios with respect to the conversion rate might be indicative of the importance of the proper item annotation and the proper selection of the visual content, in learning a genre-aware representation that could benefit the recommendation task. The genre annotation of the series items is poor, indicating simply if an item is series or not. Moreover, short clips of the series, instead of trailers, are used to provide the visual content. On the other hand, the quality of the data for the movies (informative labels, official trailers) is much better and this might be the reason why the results are different.

6.3. Conclusion

We have proposed a novel movie recommender system that filters movies based on the genre-related visual elements of their trailers. The proposed system extracts spatio-temporal deep features from the trailers and combines them, through a DBoS network, with the genre information of the movie to a single movie representation. The 3D deep visual genre-aware representation is exploited by a pure CBF system to provide personalized recommendations to users.

We posed two research questions and we conducted offline and online experiments to answer them.

Our first question was the following:

RQ1. Can a 3D deep visual genre-aware descriptor built by a DBoS network provide better movie recommendations with respect to accuracy and beyond accuracy metric (diversity, novelty, and coverage [13]) than genre and visual features extracted by a 3D pre-trained deep convolutional neural network?

The results of our offline experiments suggest that a CBF recommender system using a visual genre-aware movie representation shows better performance than genre, with respect to the accuracy and non-accuracy metrics. Compared with the visual features extracted by a 3D pre-trained deep convolutional neural network, the recommendations of the genre-aware descriptor are more accurate, but their performance is worse, with respect to diversity, novelty and coverage.

The second question was the following:

RQ2: Can the introduction of a deep visual genre-aware descriptor in a movie recommender system provide recommendations of better user perceived utility compared with genre and a pure visual representation?

The results of our online experiment show that the proposed representation does not provide better user perceived utility compared with genre. Conducting an online experiment on a real-world streaming platform, we observe that the genre outperforms the Visual_3D representation with respect to the conversion rate, when all the items (series and movies) are considered. Regarding the other metrics, there is no evidence to support that the approaches are different in their performance.

We conclude that a continuous genre representation, which reflects genre specific visual elements of the movie, provides interesting results in the content-based movie recommendation task. Exploring further its potential could bring important benefits to various tasks in the movie domain.

6.4. Future Work

Future work could explore the value of our proposed movie representation as side information in a hybrid recommender system. Additionally, the fusion of a genre-aware representation with other low-level and high-level features should also be investigated.

Regarding the generation of the genre-aware representation, it would be very interesting if a the DBoS network could be trained to predict accurately the genre of the movies. In such a way, our approach would become independent of the genre annotation which is often expensive to obtain.

Finally, our approach could be applied to other domains, suggesting the use of a neural network to learn continuous representations of categorical attributes of the items, from the media content.

7

References

[1] Barsam, R.M. and Monahan, D., 2013. *Looking at Movies: An Introduction to Film*. WW Norton.

[2] ConsumerLab, E., 2017. TV and Media 2017. *A consumer-driven future of media. Stockholm: Ericsson*.

[3] Ricci, F., Rokach, L. and Shapira, B., 2015. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook* (pp. 1-34). Springer, Boston, MA.

[4] Adomavicius, G. and Tuzhilin, A., 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6), pp.734-749.

[5] Lops, P., De Gemmis, M. and Semeraro, G., 2011. Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook* (pp. 73-105). Springer, Boston, MA.

[6] Branston, G. and Stafford, R., 2010. *The media student's book*. Routledge.

[7] Harper, F.M. and Konstan, J.A., 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *5*(4), p.19.

[8] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 1097-1105).

[9] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, *1*(4), pp.541-551.

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

[11] Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M., 2015. Learning Spatiotemporal Features With 3D Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4489-4497).

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

[13] Gunawardana, A. and Shani, G., 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook* (pp. 265-308). Springer, Boston, MA.

[14] Deldjoo, Y., Constantin, M.G., Ionescu, B., Schedl, M. and Cremonesi, P., 2018, June. MMTF-14K: a multifaceted movie trailer feature dataset for recommendation and retrieval. In *Proceedings of the 9th ACM Multimedia Systems Conference* (pp. 450-455). ACM.

[15] Videoland by RTL. http://www.videoland.com. Accessed: 2018-12-20

[16] Su, X. and Khoshgoftaar, T.M., 2009. A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, *2009*.

[17] Sarwar, B., Karypis, G., Konstan, J. and Riedl, J., 2001, April. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

[18] Chien, Y.H. and George, E.I., 1999, January. A bayesian model for collaborative filtering. In *AISTATS*.

[19] Getoor, L. and Sahami, M., 1999, August. Using Probabilistic Relational Models for Collaborative Filtering. In *Workshop on Web Usage Analysis and User Profiling (WEBKDD'99)*.

[20] Hofmann, T., 2003, July. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* (pp. 259-266). ACM.

[21] Koren, Y., Bell, R. and Volinsky, C., 2009. Matrix Factorization Techniques for Recommender Systems. *Computer*, (8), pp.30-37.

[22] Mnih, A. and Salakhutdinov, R.R., 2008. Probabilistic Matrix Factorization. In *Advances in Neural Information Processing Systems* (pp. 1257-1264).

[23] Salakhutdinov, R., Mnih, A. and Hinton, G., 2007, June. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning* (pp. 791-798). ACM.

[24] Sedhain, S., Menon, A.K., Sanner, S. and Xie, L., 2015, May. AutoRec: Autoencoders Meet Collaborative Filtering. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 111-112). ACM.

[25] Wu, Y., DuBois, C., Zheng, A.X. and Ester, M., 2016, February. Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (pp. 153-162). ACM.

[26] Guo, H., Tang, R., Ye, Y., Li, Z. and He, X., 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. *arXiv preprint arXiv:1703.04247*.

[27] Deldjoo, Y., Schedl, M., Cremonesi, P. and Pasi, G., 2018. Content-Based Multimedia Recommendation Systems: Definition and Application Domains.

[28] Rasheed, Z., Sheikh, Y. and Shah, M., 2005. On the Use of Computable Features for Film Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, *15*(1), pp.52-64.

[29] Jain, S.K. and Jadon, R.S., 2009, September. Movies genres classifier using neural network. In *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on*(pp. 575-580). IEEE.

[30] Zhou, H., Hermans, T., Karandikar, A.V. and Rehg, J.M., 2010, October. Movie genre classification via scene categorization. In *Proceedings of the 18th ACM international conference on Multimedia* (pp. 747-750). ACM.

[31] Huang, Y.F. and Wang, S.H., 2012, December. Movie Genre Classification Using SVM with Audio and Video Features. In *International Conference on Active Media Technology* (pp. 1-10). Springer, Berlin, Heidelberg.

[32] Wehrmann, J., Barros, R.C., Simões, G.S., Paula, T.S. and Ruiz, D.D., 2016, October. (Deep) Learning from Frames. In *Intelligent Systems (BRACIS), 2016 5th Brazilian Conference on* (pp. 1-6). IEEE.

[33] Deldjoo, Y., Elahi, M. and Cremonesi, P., 2016. Using visual features and latent factors for movie recommendation. CEUR-WS.

[34] Elahi, M., Deldjoo, Y., Bakhshandegan Moghaddam, F., Cella, L., Cereda, S. and Cremonesi, P., 2017, August. Exploring the Semantic Gap for Movie Recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems* (pp. 326-330). ACM.

[35] Deldjoo, Y., Elahi, M., Quadrana, M. and Cremonesi, P., 2018. Using visual features based on MPEG-7 and deep learning for movie recommendation. International Journal of Multimedia Information Retrieval, pp.1-13.

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

[37] Lowe, D.G., 1999. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on* Computer Vision(Vol. 2, pp. 1150-1157). IEEE.

[38] Bay, H., Tuytelaars, T. and Van Gool, L., 2006, May. SURF: Speeded Up Robust Features. In *European Conference on Computer Vision* (pp. 404-417). Springer, Berlin, Heidelberg.

[39] Mikolajczyk, K. and Schmid, C., 2005. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), pp.1615-1630.

[40] Scovanner, P., Ali, S. and Shah, M., 2007, September. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia* (pp. 357-360). ACM.

[41] Klaser, A., Marszałek, M. and Schmid, C., 2008, September. A Spatio-Temporal Descriptor Based on 3D-Gradients. In *BMVC 2008-19th British Machine Vision Conference* (pp. 275-1). British Machine Vision Association.

[42] Wang, H. and Schmid, C., 2013. Action Recognition with Improved Trajectories. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3551-3558).

[43] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (pp. 248-255). IEEE.

[44] Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In *Advances in Neural Information Processing Systems* (pp. 3320-3328).

[45] Sharif Razavian, A., Azizpour, H., Sullivan, J. and Carlsson, S., 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 806-813).

[46] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L., 2014. Large-scale Video Classification with Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1725-1732).

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

[48] Simonyan, K. and Zisserman, A., 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems* (pp. 568-576).

[49] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. and Toderici, G., 2015. Beyond Short Snippets: Deep Networks for Video Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4694-4702).

[50] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P. and Suleyman, M., 2017. The Kinetics Human Action Video Dataset. *arXiv preprint arXiv:1705.06950*.

[51] Hara, K., Kataoka, H. and Satoh, Y., 2018, June. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA* (pp. 18-22).

[52] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).

[53] He, K., Zhang, X., Ren, S. and Sun, J., 2016, October. Identity Mappings in Deep Residual Networks. In *European Conference on Computer Vision* (pp. 630-645). Springer, Cham.

[54] Zagoruyko, S. and Komodakis, N., 2016. Wide Residual Networks. *arXiv preprint arXiv:1605.07146*.

[55] Xie, S., Girshick, R., Dollár, P., Tu, Z. and He, K., 2017, July. Aggregated Residual Transformations for Deep Neural Networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on* (pp. 5987-5995). IEEE.

[56] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q., 2017, July. Densely Connected Convolutional Networks. In *CVPR* (Vol. 1, No. 2, p. 3).

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

[58] Hara, K., 2017, Video Classification Using 3D ResNet**,** GitHub repository, https://github.com/kenshohara/video-classification-3d-cnn-pytorch

[59] Celma, Ò. and Cano, P., 2008, August. From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition* (p. 5). ACM.

[60] Zhang, M. and Hurly, N., 2009, November. Evaluating the Diversity of Top-N Recommendations. In *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*(pp. 457-460). IEEE.