Deep Learning for Multimedia Feature Extraction for Personalized Recommendation

Aymen Ben Hassen[1], Sonia Ben Ticha[1,2] and Anja Habacha Chaibi[1]
[1]*RIADI Laboratory, University of Manouba, Tunisia*
[2]*Borj El Amri Aviation School, Tunisia*

Keywords: Deep Learning, Features Extraction, Multimedia Features, Image Feature Extraction, Video Feature Extraction, Recommender Systems, Image Recommendation, Video Recommendation.

Abstract: The analysis of multimedia content plays a crucial role in various computer vision applications, and digital multimedia constitute a major part of multimedia data. In recent years, multimedia content products have gained increasing attention in recommendation systems since the visual appearance of products has a significant impact on users' decision. The main goal of personalized recommender systems is to offer users recommendations that reflect with their personal preferences. In recent years, deep learning models have demonstrated strong performance and great potential in utilizing multimedia features, especially for videos and images. This paper presents a new approach that utilizes multimedia content to build a personalized user model. We employ deep learning techniques to extract latent features from multimedia content of item videos, which are then associated with user preferences to build the personalized model. This model is subsequently incorporated into a Collaborative Filtering (CF) to provide recommendations and enhance their accuracy. We experimentally evaluate our approach using the MovieLens dataset and compare our results with those of other methods which deals with different text and images attributes describing items.

1 INTRODUCTION

The swift evolution of Internet services and applications has given rise to an unprecedented influx of information. Within this overload of data, users grapple with the daunting task of sifting through multiple applications to uncover pertinent content. In response to this information overload, Recommender Systems (RS) have gained significance as they guide users by suggesting items tailored to their preferences from a vast array of choices. Personalized recommender systems emerge as a viable solution to the challenges posed by information overload. The main objective of recommendation systems is to provide recommendations that reflect the user's personal preferences. Although existing recommendation systems have demonstrated success in generating relevant recommendations, they face several challenges, such as the cold start problem, scalability issues, data sparsity, and the support for complex data types (e.g., images, audio, and videos) that describe the items to be recommended.

With the recent revolution in multimedia technology, each type of multimedia content, including text, graphics, video, and audio, holds significance in the realm of Big Data. Notably, video and image data have become more accessible and cost-effective to create, store, and transfer on a large scale. The big amount of generated data has prompted the research community to explore diverse study areas to support the vast proliferation of multimedia content. These areas encompass image representation, video classification, video features extraction, events and object detection, as well as video and image recommendation, among other video content analysis techniques. The efficacy of any analysis technique relies upon the extraction of visual features from multimedia content data. Given the success and effectiveness of deep learning techniques across various research fields, they have recently demonstrated exceptional performance, highlighting their significant potential in learning effective representations of complex data types (e.g., extracting relevant features from video content)(Zhang et al., 2019).

Collaborative Filtering (CF) and Content-Based Filtering (CB) are the two main approaches commonly employed in personalized recommendation systems (Aggarwal et al., 2016). CF (Burke, 2002)

Ben Hassen, A., Ben Ticha, S. and Chaibi, A. H.
Deep Learning for Multimedia Feature Extraction for Personalized Recommendation.
DOI: 10.5220/0013718800003985

In *Proceedings of the 21st International Conference on Web Information Systems and Technologies (WEBIST 2025)*, pages 267-276
ISBN: 978-989-758-772-6; ISSN: 2184-3252

relies primarily on user rating data to predict preferences, while CB (Pazzani, 2007) also considers user ratings but focuses on leveraging item features to generate personalized recommendations. These approaches are often considered complementary. Hybrid recommendation systems (Burke, 2007) combine two or more different techniques, though there is no consensus within the research community on the best methods for hybridization.

Previous studies (Be Hassen and Ben Ticha, 2020; Ben Hassen et al., 2022; Hassen et al., 2024) have proposed deep learning solutions that leverage images describing items to build personalized user models, which are then used to apply CF algorithms for recommendations. In this paper, we present a novel deep learning-based approach to extract features from videos. These features are associated with user preferences to build the personalized model, integrated into a CF algorithm for recommendations. Our results are also compared across movie domains. Specifically, Our system consists three components: (1) Extracting multimedia features through deep learning to extract and reduce the dimensionality of latent features representing video items; (2) Learning a personalized user model by inferring user preferences for the latent features of videos; (3) Using the personalized user model to compute the k nearest neighbors for each user and making recommendations based on a user-based Collaborative Filtering (CF) algorithm. To take into account scalability problems, the user model is computed offline, with only online prediction of recommendations. We evaluate the recommender system's performance empirically.

This paper is organized as follows. In Section 2, we give an overview of related work on the use of deep learning for feature extraction tasks. In Section 3, we describe our proposed approach. The experimental results are presented in Section 4, where we also compare them with those of other methods based on collaborative filtering algorithms that treat different types of item content. Finally, in Section 5, we conclude with a summary of our findings.

2 RELATED WORK

In recent years, deep learning has gained considerable attention for its role in multimedia feature extraction. Feature extraction plays a crucial role in computer vision tasks, and increasingly advanced technologies are contributing to the extraction of features that describe the content of items. This paper provides a brief literature review of deep learning techniques for feature extraction.

In the early stages of research on multimedia feature extraction techniques, several classical methods for dimensionality reduction in feature spaces were introduced, including Independent Component Analysis (ICA) (Sompairac et al., 2019), Principal Component Analysis (PCA) (Hasan and Abdulazeez, 2021), and Linear Discriminant Analysis (LDA) (Wen et al., 2018). However, these approaches, which rely on linear transformations, often fail to address the nonlinear challenges inherent in multimedia data. The advent of stream learning brought a solution to this issue by enabling the modeling of the intrinsic structure of nonlinearly distributed data and facilitating nonlinear dimensionality reduction. Additionally, some algorithms bypass dimensionality reduction by using kernel functions to map input data into higher-dimensional feature spaces (Jia et al., 2022), thereby simplifying the representation of complex nonlinear structures and reducing computational complexity.

The majority of approaches in the existing literature focus on visual aspects, driven by the human tendency to primarily perceive information visually (Ajmal et al., 2012; Otani et al., 2017). Visual features can be video-based features or frame-based features, such as average shot length and average number of faces, and frame-based, which are extracted from the sequence of frames within a video or keyframes of shots (Ibrahim et al., 2019). Examples of frame-based features include global features like color histograms and local features like SIFT, SURF, and HOG. Lyengar and Lippman (Iyengar and Lippman, 1997) propose two visual-based methods for video classification, where Hidden Markov Models (HMMs) are trained on motion information derived from optical flow and frame differences. Certain literature explores the integration of features from multiple modalities for enhanced classification. Xu and Li (Xu and Li, 2003) combine audio and visual features to classify videos into various genres. Audio features encompass the 14 Mel-frequency cepstral coefficients (MFCC), while visual features include the mean and standard deviation of MPEG motion vectors, along with MPEG 7 descriptors related to scalable color, color layout, and homogeneous texture. The emergence of deep learning techniques has facilitated the learning of more robust feature representations. Pretrained models (Puls et al., 2023) like AlexNet, VGGNet, GoogLeNet, and ResNet, trained on extensive datasets like ImageNet[1], are now commonly employed for video classification (Dewan et al., 2023; Ur Rehman et al., 2023). ImageNet is a dataset of over 15 million labeled high-resolution images belonging to roughly 22,000 categories. Moreover, it is

[1]http://www. image-net.org/

organized according to the WordNet hierarchy. Deep learning architectures allow direct input of all image pixels without the need for separate feature extraction steps (Mao et al., 2024). The primary approach involves selecting frames from a video, feeding them into a model, extracting features from a specific fully connected layer, and representing the entire video based on these features (Sharma et al., 2021; Ur Rehman et al., 2023; Duvvuri et al., 2023). The video can be classified by averaging the features of all frames or classifying each frame independently, followed by making a final decision based on the aggregated frame classifications, often utilizing traditional classifiers like SVM (Rehman and Belhaouari, 2023; Gayathri and Mahesh, 2020; Truong and Venkatesh, 2007; Ong and Kameyama, 2009). Zha et al.(Zha et al., 2015) conducted an in-depth study on event detection and action recognition using a CNN trained for image classification. They uniformly sample each video into 50 to 120 frames, extract CNN-based features from different layers (output layer, hidden layer number 6 and hidden layer number 7), and combine them using Fisher vector encoding. The fused features are then fed into an SVM for classification. Expanding on this pipeline, Li et al. (Li et al., 2017) introduced a temporal modeling approach that aggregates pre-extracted frame-level features into comprehensive video-level representations. Their method utilizes two separate sequence models: one dedicated to processing visual features and the other for audio features. The outputs of these models are then concatenated and passed through two fully connected layers, with a sigmoid activation function applied at the output layer to perform classification. As a widely used deep learning model, convolutional neural networks (CNNs) are capable of learning abstract multimedia features by leveraging shared local weights (Schmidhuber, 2015).

After reviewing the existing literature, it is evident that deep learning has been applied in various studies to address challenges faced by recommendation systems, such as data sparsity, cold start issues, and scalability(Bhatia, 2024; Tokala et al., 2024; Mouhiha et al., 2024). Recent research has also highlighted its effectiveness in processing and extracting features from multimedia data sources that describe items(Deng et al., 2019; Bhatt and Kankanhalli, 2011).

3 PROPOSED APPROACH

Our aim is to extract latent features from multimedia data that represent the content of items and use these features to infer user preferences based on their item preferences.

The approach involves leveraging the power of deep learning to extract latent features that describe videos. These features are then used to build a personalized user model for recommendation purposes. To achieve this, we apply a user-based collaborative filtering algorithm. In our approach, each item is represented by a single video. Once the latent features of each item are extracted, they are incorporated into the personalized user model, which is subsequently utilized in the collaborative filtering algorithm for making recommendations.

The overall structure of our approach is depicted in Figure 1 and is composed of three key components:

Component 1. Multimedia Features from Videos: This component focuses on extracting latent features and reducing their dimensionality using the CNN3D technique. The output is a matrix representing item profiles, where each profile encapsulates the essential features of the corresponding item.

Component 2. Personalized User Modeling: This component learns personalized user models by evaluating the utility of each extracted feature for individual users. It combines the item profiles with user preferences derived from the rating matrix to establish tailored user representations.

Component 3. Recommendations: The final component identifies and recommends the most relevant items to the active user. It calculates vote predictions for unrated items by leveraging the K-Nearest Neighbors approach in a user-based collaborative filtering algorithm. The personalized user model is utilized to measure similarities between users, enhancing the accuracy of recommendations through the rating matrix.

3.1 Multimeda Features from Videos

The purpose of this component, as illustrated in the figure, is to extract latent features from videos that describe the items and subsequently reduce the dimensionality of these features using an Autoencoder.

INPUT: Videos Describing Items.

A video is a chronological sequence of moving images, referred to as frames, possibly accompanied by an audio stream. Each frame represents an individual image captured at a specific moment in time (Orchard, 1991). The rapid succession of these frames creates the illusion of motion when the video is played. In addition to visual information, a video may incorporate audio data, allowing for a multimodal experience that integrates both sound and image (Amer and Dubois, 2005).

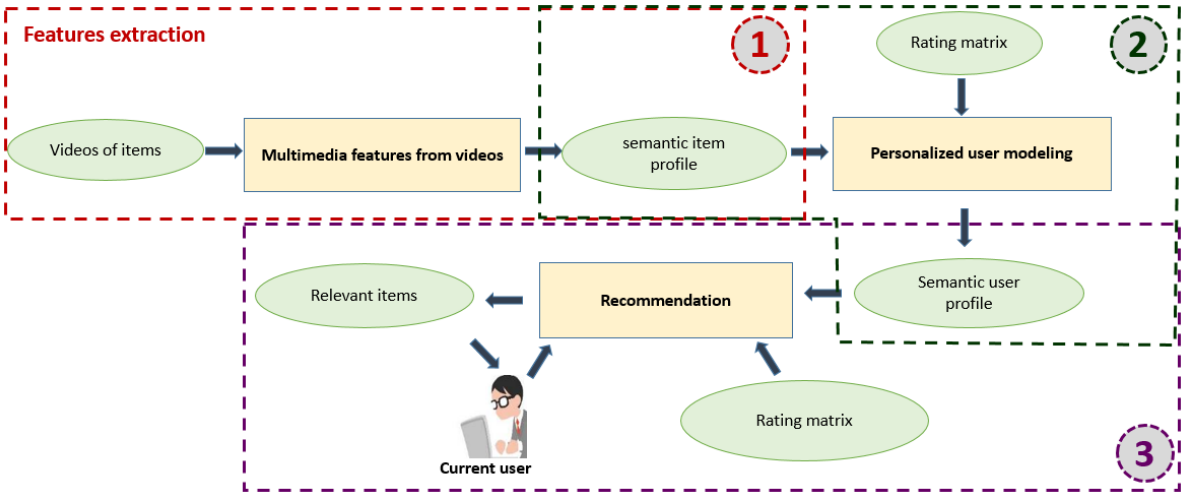

Figure 1: Proposed architecture.

Before introducing videos as inputs to our model, a crucial preprocessing phase is necessary to ensure optimal results. This step aims to enhance the quality of video data and make it compatible with the deep learning process. Preprocessing steps include video normalization to ensure a consistent scale of intensities, temporal cropping to extract relevant segments, and noise reduction to eliminate unwanted disturbances. Furthermore, special attention is given to the spatial resolution of video images, aiming to optimize the representation of important features. This systematic process guarantees that input videos are prepared adequately, providing the CNN3D with high-quality data for the extraction of significant features. Each video is decomposed into a sequence of frames, where the number of frames is determined by an adjustable hyperparameter. In other words, the video representation is dependent on the sampling frequency, where each second can be translated into one or more frames, depending on the specified value for this hyperparameter. This decision regarding the sampling frequency has direct implications on the approach's outcomes. In the end, this representation can significantly influence the model's accuracy, determining whether the final predictions will be of high quality or less precise. Thus, the thoughtful choice of the sampling frequency hyperparameter is crucial, as it impacts how videos are interpreted and processed by the CNN3D, playing an essential role in achieving accurate results.

The representation of a video V_i consists of a sequence of frames, providing a comprehensive temporal perspective, and a keyframe KF in the form of a vector. This duality in representation allows capturing both the dynamic and static aspects of the video, offering a balanced view of the important visual features encapsulated in the KF vector.

So, each video V_i is segmented into a set of frames, and a keyframe (denoted KF) representing a single frame. A video is represented as defined by (1):

$$V_i = KF_{i,j}, j = 1 \ldots NF_i \quad (1)$$

with NF representing the number of frames in the video V_i.

OUTPUT: Profile of Items.

After feature extraction, we obtain the latent features of videos, which will represent items profile. The profile of the items is then modeled by the matrix $MIP(N,F)$, N is the number of items and F is the number of latent features extracted, Where $f_{ij} = MIP(i, f_j)$ represents the value of feature f_j in item i, thus each item i is modeled by the vector $\overrightarrow{P_i}$ of dimension F defined by:

$$\overrightarrow{P_i} = (f_{i_j})_{(j=1,\ldots,F)} = \begin{pmatrix} f_{i_1} \\ \vdots \\ f_{i_k} \end{pmatrix}$$

Features Extraction. Feature extraction is an important and commonly used technique in video processing. This technique is employed to detect features. Our goal is to represent video by features extracted in the keyframes of its shots, taking advantage of the success that deep learning techniques have had in computer vision and especially for features extraction. Thus, we proposed a new method aiming to map the video into a multidimensional vector of values. Our method's basic idea is to represent a video using features that are extracted from each created shot's keyframe. To do this, we use on the CNN3D to extract latent features from the video items. This component extract features using CNN3D which is a deep learning technique that uses the convolutional layers with

the correction layer ReLu (Linear rectification), some of which are followed by Max-Pooling layers.

CNN3D aiming to simultaneously enhance spatial and temporal information. (Liu et al., 2017) Where the spatiotemporal flow can learn more details about local movements through the mutual enhancement of individual flow. The CNN3D has been proposed for human action recognition. CNN-3D exploit variations in texture in sequences of images by extending their convolutional kernel to the third dimension. The 3D convolutional layer takes a volume as input and produces another volume. Spatial and temporal information is extracted layer by layer. Tran et al (Tran et al., 2015) suggested a simple yet effective approach for learning spatiotemporal features using a three-dimensional convolutional neural network, demonstrating that CNN3Ds can achieve faster and more accurate performance. In particular, the features used in (Tran et al., 2015) have four properties for an effective video descriptor: generic, compact and efficient. CNN3Ds were initially proposed for optimizing convolutional neural networks (2D) and for solving tasks based on videos. Pre-trained CNN-3D (Karpathy et al., 2016) is a deep learning approach designed to optimize performance in machine learning by leveraging knowledge and tasks previously completed by other models (Wei et al., 2014). This method serves as a robust tool for training on large target networks while minimizing the risk of overfitting. Pre-trained CNN-3D models enable us to utilize existing architectures for new tasks efficiently. The primary advantages of using pre-trained models include: (1) facilitating transfer learning by reusing models to extract features from new datasets, (2) reducing the computational power required to train large models on extensive datasets, and (3) saving time by bypassing the need to learn the networks. In our approach, we employed pre-trained 3D CNN models, specifically VGG-11, VGG-16, and VGG-19 (Simonyan and Zisserman, 2014), to extract features from each frame of the videos in our complex dataset. Typically, the initial layers of these models capture generic features, while the deeper layers focus on more specific characteristics. The pre-trained 3D CNN models we used were originally trained on the ImageNet dataset. For features extraction, we utilized the convolutional layers of the models, excluding the fully connected layers typically used for classification. The VGG architecture in the pre-trained models consists of a composite of five blocks of convolutional layers, with some blocks followed by Max-Pooling layers to further refine features extraction.

The keyframe of each frame is passed through a stack of convolutional layers, where the filters were used with a very small receptive field: 3×3. In one of the configurations, it also utilizes 2×2 convolution filters, which can be seen as a linear transformation of the input channels. The convolution stride is fixed to 1 pixel, the spatial padding of convolutional layer input is such that the spatial resolution is preserved after convolution, i.e. the padding is 1-pixel for 3×3 convolutional layers. Spatial pooling is carried out by five max-pooling layers, which follow some of the convolutional layers. Max-pooling is performed over a 3×3 pixel window, with stride 2. In the VGG11: 8 convolutional layers.In the VGG16: 13 convolutional layers. In the VGG19 model: 16 convolutional layers. The width of convolutional layers (the number of channels) is rather small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

The next step involves calculating the average of features for each video, utilizing the matrix that aggregates the extracted features from the video frames. This averaging operation aims to condense the extracted features into a more concise representation, thereby facilitating the understanding of the video's overall features. The formula above defines this process, where each column of the matrix represents a specific features, and the average calculation is applied along each column to obtain an average representation. Below, you'll find the average equation for the entire frame set of the video. In this case, each video V_i of each item i is represented by the set of Deep Features Frame DFF of all its keyframes. We have calculated the $DFV_{i,j}$ (Deep Features Video) as the average of the DFFs of all its keyframes $KF_{i,j}$.

$$DFV_{i,j} = \frac{1}{NF_i} \cdot \sum_{l=1}^{NF_i} DFF_{i(l,j)} \tag{2}$$

Where $f_{ij} = MIP(i, f_j) = DFV_{i,j}$.

3.2 Personalized User Modeling

In this section, we will present the second component allowing personalized user modeling. The idea is to build a new user profile.

INPUT:

- Items profile modeled by MIP result of first component.
- Usage data is represented by rating matrix Mv having L rows and N columns. The lines represent the users and the columns represent the items. Ratings are defined on a scale of values. The rating matrix has missing value rate exceeding 95%, where missing values are indicated by a "?", $v_{u,i}$ the rating of user u for item i.

OUTPUT:
At the end of personalized user modeling, we obtain a personalized user model which is represented by a matrix which we will call *"Matrix User Profile "* ($MUP_{L,F}$) without missing values, having L rows representing the users and F columns representing the features. This profile defines user preferences for the extracted features describing the items based on their assessments for these same items. $MUP(u,f)$: represents the utility of feature f for user u .

Personalized User Modeling. The idea is to infer the utility of each feature of items (the result of component 1) for each user. To do this we were inspired by (Ben Ticha et al., 2013) which gives different formulas for calculating matrix of user profiles. We used the formula which gave better results (as given in formula 3).

$$MUP_{(u,j)} = \sum_{i \in I_{u_{relevant}}} v_{u,j} \times MIP_{(i,j)} \quad (3)$$

We denote by $I_{u_{relevant}}$ the set of relevant items of user u. To compute $I_{u_{relevant}}$, we used the formula given in (Ben Ticha, 2015).

3.3 Recommendation

The idea is to take advantage of the efficiency and simplicity of user-based collaborative filtering algorithm to make recommendations using the Personalized User Model to determine the nearest neighbors of the current user. The personalized user model is used to compute similarities between users. Similarities are used to select the K nearest neighbors of the current user in a user-based collaborative filtering algorithm .

The User Profile u (PU_u) is represented by index line u in User Profile matrix (MUP) modeling the personalized model of users. Computing the similarity between two users then amounts to calculating the correlation between their two profiles. In our case, the user profile u (PU_u) models the importance of the hidden features for the user u. The Cosine is utilized for calculating the correlation between two users u and v. It is defined by the formula (4).

$$sim(u,v) = cos(\vec{PU_u}, \vec{PU_v}) = \frac{\vec{PU_u} \cdot \vec{PU_v}}{||\vec{PU_u}|| \; ||\vec{PU_v}||} \quad (4)$$

To compute predictions of rate value of an item i not observed by the current user u_a, we applied the formula 5 keeping only the K nearest neighbors. The similarity between u and u_a being determined in our case from their user profiles applying the formula 4.

$$pred(u_a,i) = \bar{v_{u_a}} + \frac{\sum_{k \text{ nearest neighbors}} sim(u_a,u)(u_{ui} - \bar{v_u})}{\sum_{k \text{ nearest neighbors}} |sim(u_a,u)|} \quad (5)$$

The rating prediction in our approach is calculated by applying user-based collaborative filtering algorithm. In the standard algorithm, the similarity between users is calculated from rating matrix (Kluver et al., 2018). In our case, we use MUP matrix modeling the personalized users profile to calculate the similarity between users.

Our approach provides solutions to the scalability problem. The first two components, namely feature extraction and personalized user modeling, are executed in offline mode. To reduce the time complexity of computing the rating prediction, the determination of K nearest neighbors of each user is also computed in offline mode, keeping only the k nearest to them. The calculation of predictions for the current user is executed in real-time during his interaction with e-service.

4 PERFORMANCE STUDY

To evaluate our approach, we opted for offline evaluation mode. The offline evaluation allows the performance of several recommendation algorithms to be compared objectively. We have adopted an empirical approach. The performances were analysed through different experiments on datasets.

We evaluated the performance by measuring the accuracy of the recommendations, which measures the capacity of a recommendation system to predict recommendations that are relevant to its users. We measured the accuracy of the prediction by calculating the Root Mean Square Error (RMSE) (Desrosiers and Karypis, 2011), which is the most widely used metric in CF research literature.

$$RMSE = \sqrt{\frac{\sum_{(u,i) \in T} (pred(u,i) - v_{ui})^2}{|T|}} \quad (6)$$

Where T is the set of couples (u,i) of R_{test} for which the recommendation system predicted the value of the vote. It computes the average of the square root difference between the predictions and true ratings in the test data set, the lower the RMSE is, the better the accuracy of predictions.

4.1 Experimental Datasets

In the main part, we focus on the domains of movie recommendation and partly as well video recommendation. We evaluate our proposed approach on two public datasets: dataset for item content and dataset to train the recommendation models.

For the item content data, we used the MovieLens

20M YouTube Trailers Datase[2] to extract movie trailers. We obtained YouTube video IDs for the movie trailers by conducting queries on www.google.com. Out of the 27,278 unique movie IDs used in MovieLens-20M, our method successfully retrieved YouTube IDs for 25,623 trailers, achieving a success rate of 0.94. This dataset is publicly available on the MovieLens website.

We used the HetRec 2011 dataset of the MovieLens recommender system[3], which contain user ratings. The HetRec-2011 dataset provides the usage data set and contains 1,000,209 explicit ratings of approximately 3,900 movies made by 6,040 users with approximately 95% of missing values.

4.2 Performance Evaluation of Features Extraction Based on the Number of Layers

To evaluate our approach, firstly, we started by features extraction, and we took all the features extracted of Convolutional Neural Networks 3D. We used CNN3D with one frame per second with three featurs extraction models: CNN3D with VGG11, VGG16 and VGG19 models in the first component 3.1 (Multimeda Features from Videos) available included in the library keras[4] with Python programming language[5] with version 3.7 and run on TensorFlow [6]. The three models (VGG-11, VGG-16, and VGG-19) generate the same number of features F for all these models, which is 25,088 features. Each item i has the importance of feature f which is a value between [0.100]. The precisions of the three models CNN3D with (VGG11, VGG16 and VGG19) are shown in Figure 2. The RMSE is plotted against the number K of neighbors. In all cases, the RMSE converges between 50 and 60 neighbors.

The accuracy of predictions ratings of the VGG19 model is higher than those observed by CNN3D with VGG11 and VGG16, for all the neighbors. The best performance is obtained by CNN3D with VGG19 whose RMSE value is equal to 0.9407. On the other hand, the best performance for VGG11 is an RMSE value of 0.9525, and for VGG16 it is 0.9456, for 60 neighbors.

It is observed that, for the CNN3D with VGG19 model, the performance is better compared to the

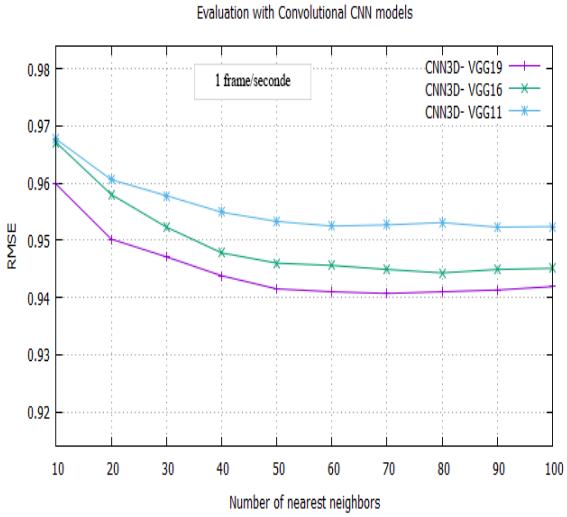

Figure 2: Evaluation with VGG models.

cases with CNN3D using VGG11 and VGG16. This suggests that increasing the number of layers has a positive impact on performance in our experiments. While we have not tested architectures with more layers than VGG19, our results indicate a correlation between the depth of the network and the achieved accuracy, which may guide future tuning of this hyperparameter.

4.3 Performance Evaluation Based on the N-Numbers of Frames Extractions per Second

To improve the performance of our approach and further improve upon the previous best result with VGG19, we evaluated the performance of CNN3D in three distinct scenarios based on the N-number of Frames per second. The first scenario involves extracting one frame per second, the second involves extracting two frames per second, and the third involves extracting three frames per second with VGG19 model.

As illustrated in Figure 3, we observe that extracting with 3 frames per second produces more accurate results than the other extraction frequencies. The accuracy of predictions of the CNN3D model with an extraction of three frames per second are higher than those observed with an extraction of one frame per second and two frames per second. The best performance is achieved with an extraction of three frames per second, presenting an RMSE value of 0.9275 for 70 neighbors.

[2]https://grouplens.org/datasets/movielens/20m-youtube/

[3]https://grouplens.org/datasets/hetrec-2011/

[4]https://keras.io/

[5]https://www.python.org/

[6]https://www.tensorflow.org/

Table 1: Comparison of Approaches: CF for Personalized Recommendations.

Approach	Item Content	RMSE
VGG19	Image	0.9263
VGG19 with TOP-K Reduction)	Image	0.9165
VGG19 with AE Reduction	Image	**0.9116**
CONV AE	Image	**0.9228**
CONV AE with Reduction	Image	**0.9098**
CNN3D-VGG19 1Frame/s	Video	**0.9407**
CNN3D-VGG19 2Frame/s	Video	**0.9301**
CNN3D-VGG19 3Frame/s	Video	**0.9275**
Genre	Text	**0.9044**
Origin	Text	**0.9118**

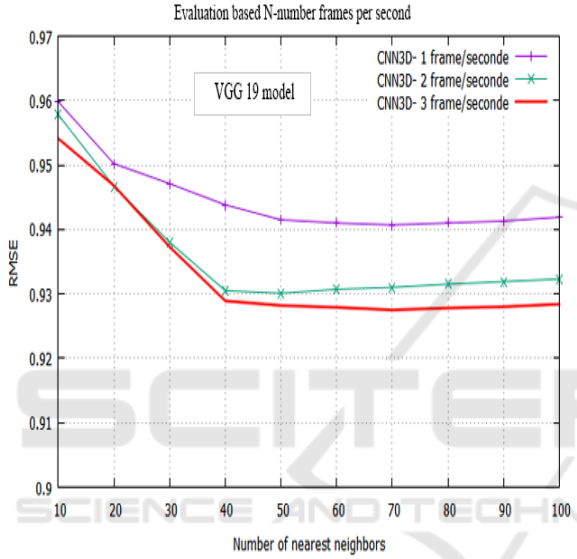

Figure 3: Performance evaluation based N-number frames per second.

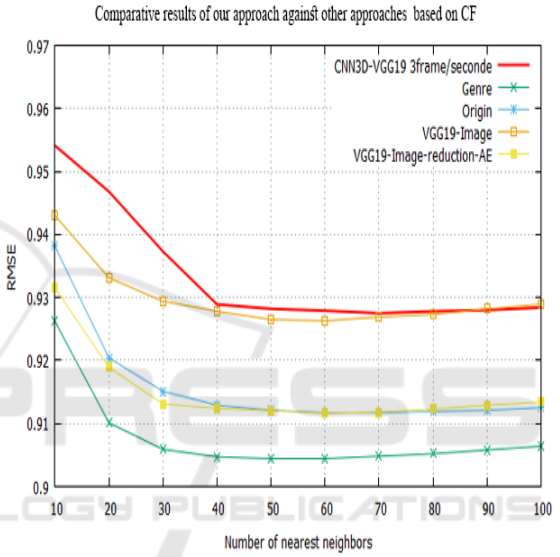

Figure 4: Comparative results of our approach against other approaches based CF.

4.4 Comparative Results of Our Approach Against Other Approaches Based on CF

In Figure 4, we compare the performance of our best method using VGG19 with 3 frames per second against our previous work titled "Transfer Learning to Extract Features for Personalized User Modeling" (Be Hassen and Ben Ticha, 2020), "Deep Learning for Visual-Features Extraction Based Personalized User Modeling" (Ben Hassen et al., 2022) which treated the images and to a "User Semantic Collaborative Filtering" approach (Ben Ticha, 2015) which treated with different text attributes describing movies (Genre, Origin).

We represented the performances of our approach with the Genre of movie attribute (e.g., comedy, drama) represented by the "Genre" plot, the origin of movie attribute (The country of origin of movie) represented by the "Origin" plot, the movie poster with VGG19 represented by the "VGG19-Image" plot and the movie poster with reduction of the dimension represented by the "VGG19-Image-reduction-AE" plot.

The details of the comparative results are presented in the table 1. The reported RMSE corresponds to the best performance achieved with respect to the number K of neighbors. In all cases, the best performance is observed when the RMSE converges for several neighbors between 50 and 70.

In conclusion, we can say that the best performance which deals with the textual data describing the item (Genre). The results of our approach are acceptable compared to the results of (Ben Ticha, 2015; Be Hassen and Ben Ticha, 2020; Ben Hassen et al., 2022), which explains this by the fact that the trailer of a movie has an importance in the preferences of the users and it may not be discriminating enough as the genre.

5 CONCLUSIONS

In this paper, we have proposed to apply Convolutional Neural NetworK 3D to extract latent features of videos describing items. We have used the resulting model for personalized user modeling by inferring user preferences for latent features of videos from the history of their preferences for items and thus building the user model. The personalized model obtained was then used in a collaborative filtering algorithms to make recommendations.
We evaluated the performance of our approach by applying deep Convolutional Neural NetworK 3D for extract the latent features. To improve the performance, we modified and increased the N-number of frame per second. Finally, we compared the accuracy of our proposed approach to other approaches based on hybrid filtering which deals with different text and images attributes describing items.

Despite the promising results, the current work has several limitations. It relies solely on the visual modality and does not incorporate audio, textual reviews, or user demographics. Additionally, the evaluation is limited to RMSE, which does not capture the ranking quality or diversity of the recommendations.

In future work, we plan to:

- Integrate multimodal content (e.g., audio, text,) to enrich item representations;
- Explore lightweight deep learning architectures to improve scalability in real-time environments;
- Apply the approach to other domains such as e-commerce, education, or streaming services;
- Incorporate user feedback mechanisms for online adaptation and continual learning.

We believe this work offers a solid foundation for incorporating video content into personalized recommendation systems and provides multiple directions for future enhancement.

REFERENCES

Aggarwal, C. C. et al. (2016). *Recommender systems*, volume 1. Springer.

Ajmal, M., Ashraf, M. H., Shakir, M., Abbas, Y., and Shah, F. A. (2012). Video summarization: techniques and classification. In *Computer Vision and Graphics: International Conference, ICCVG 2012, Warsaw, Poland, September 24-26, 2012. Proceedings*, pages 1–13. Springer.

Amer, A. and Dubois, E. (2005). Fast and reliable structure-oriented video noise estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):113–118.

Be Hassen, A. and Ben Ticha, S. (2020). Transfer learning to extract features for personalized user modeling. In *WEBIST*, pages 15–25.

Ben Hassen, A., Ben Ticha, S., and Chaibi, A. H. (2022). Deep learning for visual-features extraction based personalized user modeling. *SN Computer Science*, 3(4):261.

Ben Ticha, S. (2015). *Hybrid Personalized Recommendation*. PhD thesis, Faculty of Sciences of Tunis.

Ben Ticha, S., Roussanaly, A., Boyer, A., and Bsaïes, K. (2013). Feature frequency inverse user frequency for dependant attribute to enhance recommendations. In *The Third Int. Conf. on Social Eco-Informatics - SOTICS*, Lisbon, Portugal. IARIA.

Bhatia, V. (2024). Dlsf: Deep learning and semantic fusion based recommendation system. *Expert Systems with Applications*, 250:123900.

Bhatt, C. A. and Kankanhalli, M. S. (2011). Multimedia data mining: state of the art and challenges. *Multimedia Tools and Applications*, 51:35–76.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12:331–370.

Burke, R. (2007). Hybrid web recommender systems. *The adaptive web: methods and strategies of web personalization*, pages 377–408.

Deng, J., Li, C., and Zhou, B. (2019). Analysis of feature extraction methods of multimedia information resources based on unstructured database. In *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, pages 236–240. IEEE.

Desrosiers, C. and Karypis, G. (2011). A comprehensive survey of neighborhood-based recommendation methods. In *Recommender systems handbook*, pages 107–144. Springer.

Dewan, J. H., Das, R., Thepade, S. D., Jadhav, H., Narsale, N., Mhasawade, A., and Nambiar, S. (2023). Image classification by transfer learning using pre-trained cnn models. In *2023 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI)*, pages 1–6. IEEE.

Duvvuri, K., Kanisettypalli, H., Jaswanth, K., and Murali, K. (2023). Video classification using cnn and ensemble learning. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 66–70. IEEE.

Gayathri, N. and Mahesh, K. (2020). Improved fuzzy-based svm classification system using feature extraction for video indexing and retrieval. *International Journal of Fuzzy Systems*, 22(5):1716–1729.

Hasan, B. M. S. and Abdulazeez, A. M. (2021). A review of principal component analysis algorithm for dimensionality reduction. *Journal of Soft Computing and Data Mining*, 2(1):20–30.

Hassen, A. B., Ticha, S. B., and Chaibi, A. H. (2024). Extracting visual features for personalized recommendation using autoencoder. *Procedia Computer Science*, 246:3634–3643.

Ibrahim, Z. A. A., Saab, M., and Sbeity, I. (2019). Videotovecs: a new video representation based on deep learning techniques for video classification and clustering. *SN applied sciences*, 1(6):560.

Iyengar, G. and Lippman, A. B. (1997). Models for automatic classification of video sequences. In *Storage and Retrieval for Image and Video Databases VI*, volume 3312, pages 216–227. SPIE.

Jia, W., Sun, M., Lian, J., and Hou, S. (2022). Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8(3):2663–2693.

Karpathy, A. et al. (2016). Cs231n convolutional neural networks for visual recognition. *Neural networks*, 1.

Kluver, D., Ekstrand, M. D., and Konstan, J. A. (2018). Rating-based collaborative filtering: algorithms and evaluation. *Social information access: Systems and technologies*, pages 344–390.

Li, F., Gan, C., Liu, X., Bian, Y., Long, X., Li, Y., Li, Z., Zhou, J., and Wen, S. (2017). Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*.

Liu, H., Tu, J., and Liu, M. (2017). Two-stream 3d convolutional neural network for skeleton-based action recognition. *arXiv preprint arXiv:1705.08106*.

Mao, M., Lee, A., and Hong, M. (2024). Deep learning innovations in video classification: A survey on techniques and dataset evaluations. *Electronics*, 13(14):2732.

Mouhiha, M., Oualhaj, O. A., and Mabrouk, A. (2024). Enhancing movie recommendations: A deep neural network approach with movielens case study. In *2024 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1303–1308. IEEE.

Ong, K.-M. and Kameyama, W. (2009). Classification of video shots based on human affect. *The Journal of The Institute of Image Information and Television Engineers*, 63(6):847–856.

Orchard, M. T. (1991). Exploiting scene structure in video coding. In *Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems & Computers*, pages 456–457. IEEE Computer Society.

Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., and Yokoya, N. (2017). Video summarization using deep semantic features. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, pages 361–377. Springer.

Pazzani, M. (2007). Content-based recommendation systems.

Puls, E. d. S., Todescato, M. V., and Carbonera, J. L. (2023). An evaluation of pre-trained models for feature extraction in image classification. *arXiv preprint arXiv:2310.02037*.

Rehman, A. and Belhaouari, S. B. (2023). Deep learning for video classification: A review. *Authorea Preprints*.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Sharma, V., Gupta, M., Kumar, A., and Mishra, D. (2021). Video processing using deep learning techniques: A systematic literature review. *IEEE Access*, 9:139489–139507.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sompairac, N., Nazarov, P. V., Czerwinska, U., Cantini, L., Biton, A., Molkenov, A., Zhumadilov, Z., Barillot, E., Radvanyi, F., Gorban, A., et al. (2019). Independent component analysis for unraveling the complexity of cancer omics datasets. *International Journal of molecular sciences*, 20(18):4414.

Tokala, S., Nagaram, J., Enduri, M. K., and Lakshmi, T. J. (2024). Enhanced movie recommender system using deep learning techniques. In *2024 3rd International Conference on Computational Modelling, Simulation and Optimization (ICCMSO)*, pages 71–75. IEEE.

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*, pages 4489–4497.

Truong, B. T. and Venkatesh, S. (2007). Video abstraction: A systematic review and classification. *ACM transactions on multimedia computing, communications, and applications (TOMM)*, 3(1):3–es.

Ur Rehman, A., Belhaouari, S. B., Kabir, M. A., and Khan, A. (2023). On the use of deep learning for video classification. *Applied Sciences*, 13(3):2007.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., and Yan, S. (2014). Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*.

Wen, J., Fang, X., Cui, J., Fei, L., Yan, K., Chen, Y., and Xu, Y. (2018). Robust sparse linear discriminant analysis. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(2):390–403.

Xu, L.-Q. and Li, Y. (2003). Video classification using spatial-temporal features and pca. In *2003 International Conference on Multimedia and Expo. ICME'03. Proceedings (Cat. No. 03TH8698)*, volume 3, pages III–485. IEEE.

Zha, S., Luisier, F., Andrews, W., Srivastava, N., and Salakhutdinov, R. (2015). Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1503.04144*.

Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):5.