Enhancing Interior Design: A Furniture Recommender System Integrating Deep Learning

Amrita Soney
Computer Science Department
Netaji Subhas Unitversity of Technology
Delhi, India
amrita.soney.ug21@nsut.ac.in

Laksshay Sehrawat
Computer Science Department
Netaji Subhas Unitversity of Technology
Delhi, India
laksshay.sehrawat.ug21@nsut.ac.in

Vanshika Gupta
Computer Science Department
Netaji Subhas Unitversity of Technology
Delhi, India
vanshika.gupta.ug21@nsut.ac.in

Gaurav Singal
Computer Science Department
Netaji Subhas Unitversity of Technology
Delhi, India
gaurav.singal.ug21@nsut.ac.in

Preeti Kaur
Computer Science Department
Netaji Subhas Unitversity of Technology
Delhi, India
preeti.kaur@nsut.ac.in

***Abstract*—Recommender systems are a subclass of information filtering systems. These systems are specialized software components, which usually make part of a larger software system, but can also be standalone tools. A recommender system's main goal is to provide the user software suggestions for items that can be useful. The suggestions are related to different decision-making mechanisms, different techniques, such as, what product to buy, what movie to watch, or what vacation to reserve. In the context of recommender systems, the general term "item" refers to what the system is actually recommending to its users. The paper presents the development of a recommendation system, capable of making item suggestions, based on image provided by the user and interaction data, using an image- based CNN model algorithm.**

I. Introduction

The main objective of our recommender systems in a furniture company is to narrow down a vast catalog to assist customers in finding exactly the products they want. Most recommendation algorithms (Collaborative filtering and Content-based filtering) that are used in e-commerce companies leverage customers' browsing history, which includes various types of interactions with products, such as clicks, add-to-cart actions, ratings, and orders. However, the uncertainty, diversity, and timelines of each customer's profile, as well as the absence of new customers' history, make it challenging for such algorithms to be robust to all customers. hence, we have implemented the newest method for aiding customers in their search for complementary items. Rather than depending on customer input, this model leverages an image-based pre-trained CNN model (VGG-16) to understand compatibility from product imagery, thereby mimicking the way professional interior designers match pieces of furniture together and eliminating the cold start problem in the process.

II. Related Work

The recommendation system operates on the concept of unified style to establish a 3D furniture recommendation system within a 3D environment[1]. This involves categorizing furniture into clusters, identifying user preferences through physiological signal collection and user information modeling, and suggesting other furniture items with a similar style. By conducting model retrieval within the same clusters, the overall search time is reduced, leading to more practical and efficient results, thereby enabling dynamic recommendations. Recognizing the enhanced information representation and content conveyance capabilities of 3D models, especially in line with Marr D's theory of three-dimensional visual environment information, furniture products displayed in 3D virtual scenes with user interaction become more user-friendly and memorable. This aligns with user observation habits, offering a tangible, interactive environment for users to experience products. This, in turn, enhances business intelligence, facilitating user perception and recognition of products and ultimately boosting sales effectively.

With an increase in the standard of living, peoples' attention gradually moved towards fashion which is concerned to be a popular aesthetic expression[2]. However, giving too many options of garments on the e-commerce websites has presented new challenges to the customers in identifying their correct outfits. Thus, in this paper, they proposed a personalized Fashion Recommender system that generates recommendations for the user based on an input given. Unlike the conventional systems that rely on the user's previous purchases and history, this project aims at using an image of a product given as input by the user to generate recommendations that are similar to that. We use neural networks to process the images from the Deep Fashion data set and a nearest neighbor-backed recommender to generate the final recommendations. In this paper, the author has presented a novel framework for fashion recommendation that is driven by data, visually related, and simple effective recommendation systems for generating fashion product images. The proposed approach uses a two-stage phase. Initially, their proposed approach extracts the features of the image using CNN classifier i.e. , for instance allowing the customers to upload any random fashion image from any E-commerce website and later generating similar images to the uploaded image based on the features and texture of the input image. It is imperative that such research goes forward to facilitate greater recommendation accuracy and improve the overall experience of fashion exploration for direct and indirect consumers alike.

This paper introduces an approach for assessing the compatibility of 3D furniture styles across different classes. The method is built upon the utilization of Triplet CNN to extract more representative features that effectively capture the style compatibility between pairs of 3D furniture models[3]. To validate it, a dataset containing 420 textured 3D furniture models was collected and assessed by a group of raters recruited from Amazon Mechanical Turk (AMT) to evaluate the comparative suitability of paired models within the dataset. The proposed method was trained and evaluated based on three datasets, including the collected textured dataset and two additional non-textured datasets. Experimental results indicate that the method outperforms the state-of-the-art approach, which learned a metric using pre-extracted geometric features. Furthermore, the method is utilized to create a furniture recommendation system that can assist potential users in interior design decisions. In the future, the aim is to enhance the reliability of the trained model through the expansion of the dataset and to employ the proposed method for the development of a virtual reality interior design tool, allowing users to directly immerse themselves in the designed house based on the recommendation results.

The assortment recommender system leverages product images to establish visually coherent trends within Overstock's product offerings. Two variants are introduced: a visual-only version and a multimodal version[4]. The visual-only variant generates a representation of product images by applying thresholding to activations from specific layers of a pre-trained deep residual neural network, Resnet-50. It subsequently employs topic modeling (LDA) on these product image representations to create visual trends among Overstock products. A greedy approach is then proposed, both with and without budget constraints, to formulate assortment recommendations based on seed items that maximize the visual compatibility of the set. The multimodal variant utilizes not only image representation but also text-based product attributes. This variant harnesses Polylingual LDA (PolyLDA) to establish trends based on two modalities: images and text. Multiple assortments generated by both models are featured, and their performance is evaluated through a series of offline validations and a large-scale online A/B test on Overstock. The experimental findings demonstrate that the incorporation of both image and text data results in a more cohesive visual style compared to using only images, thereby enhancing user engagement metrics within the recommendations module. Additionally, it is shown that PolyLDA offers a meaningful approach to simultaneously learn style from both text and image data.

The aim of this project is to build a furniture recommender that uses the previously mentioned generated images as input[5]. Then, using image processing techniques, AI as well as some user preferences, presents to the customer similar pieces with some additional information and where they can buy them. The idea was a system that aimed to identify different pieces of furniture in a room and tell the user where to buy or find them. They have built a bed recommender that has been built with its own database as well as a detector, a classifier, and an Image retrieval block. The system retrieves similar and relevant images to the user with all the information needed and also relies on a User interface that makes it easier for a user to interact with it. They have a database with 386 images separated into 3 categories by a classifier that has an accuracy of 95.41%. A Web Scraping algorithm has been developed for a specific site, in this case, Ikea. A user-friendly interface has an efficient and well-structured code. Their future work is to improve the performance of the whole system. A new classifier with more capacity and ability to choose between master and individual beds needs to be implemented. Improving the current database by adding new online stores to recommend a wider range of styles and by implementing another user filter, so they can choose the store. Right now the recommender only works for beds, but it would be useful to have bedding and to extend the possible recommendations to all bedroom furniture and even for other spaces.

III. State of Art

Nowadays artificial intelligence (AI), machine learning (ML) and deep learning (DL) are trending fields. These terms are frequently used as alternatives for each other, however, this is not always correct. It has to be mentioned that from all these terms AI is the most general concept, ML being just a subset of AI and DL making part of ML [7]. The main objective of any ML algorithm is to generalize beyond the training samples, to understand and interpret data with success. There are multiple techniques for creating a recommender system, all based on different aspects of the collected data and the environment it is part of.

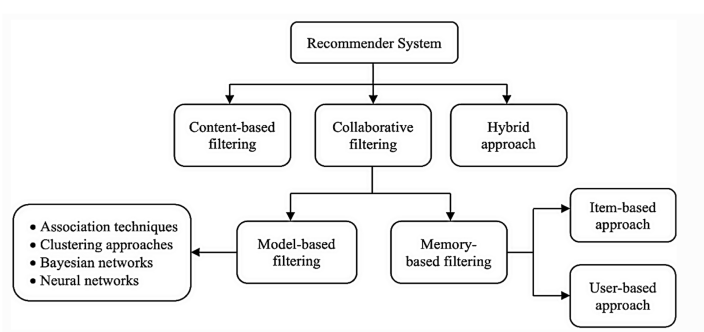

Fig. 1. Types of recommender systems

A. Content Based Filtering

Recommenders are also present in news suggestions, where DailyLearner has to be mentioned. A great web recommender is Letizia[6], which was developed as a cybersurfer extension that creates a model, acclimatized to the stoner. It builds from keywords from the stoner's interests by following the stoner's browsing. It's grounded on implicit compliance to learn the stoner's interests. Putting a runner in the bookmarks shows a strong sign that the stoner is interested in the content of that runner. The maturity of content-grounded recommenders is created as textbook classifiers using training sets of documents, that represent stoner interests or the lack of it. As a result, for achieving high delicacy the training sets must contain a large quantum of data. The main problem with this approach is the lack of" intelligence".

B. Collaborative Based Filtering

Collaborative filtering is realized by analyzing the behavior of a group of users to make provide suggestions to other users. The preferences of other users influence the recommendation. The main idea behind the collaborative filtering-based technique is that if a person has the same opinion as another person on a topic, then he is more likely to share that other person's opinion on another topic than that of a randomly chosen person. From another perspective, collaborative filtering can be viewed as a generalization of regression and classification. Using collaborative filter-based recommendation has several advantages compared to content-based systems, like no domain knowledge necessary, serendipity, affinity to nuances, benefits of large user bases. Collaborative filtering-based approaches also come with some drawbacks, like complexity and expense, cold start (the system needs enough information (user-item interactions) to work properly). A perfect example would be YouTube's recommendation system. As it is presented in [7], their system is composed of two neural networks working together to provide recommendations, one for candidate generation and one for ranking.

C. Hybrid System

The simplest and most direct way to build a hybrid recommender system is to take the independent result of content and a collaborative-based recommender system, then using a voting scheme combine their predictions. [8] presents a method where the combination is done by choosing items that correspond to the user's profile and at the same time having positive ratings from the user's neighbors. In [9], the technique used compares users according to their content-based profiles and uses a collaborative filtering system where the generated similarity measures are used. In [10], the predictions based on the content are used to enrich the rating matrix, and then collaborative filtering is run. In[11], item-based collaborative filtering is run, but before that uses the item's content descriptions and their associated rating vectors to calculate the similarity between them.

IV. Proposed Methodology

Most recommender systems in use today leverage classical machine learning models. They can be divided into collaborative filtering approaches, which perform matrix factorization on the user-item interaction matrix, and content-based approaches, which use regression or classification models on prior information about the users and/or the items to make recommendations. Both approaches analyze structured tabular data from the users or items. In this project, we were curious to see if deep learning approaches — specifically convolutional neural networks (CNN) — can learn useful latent features from unstructured image data and use them to make actionable recommendations. We have built a web application that takes in the user's room image, analyzes its design features using a convolutional neural network, and recommends products in other categories with similar style elements.

A. Data Collection

For this project, two datasets were acquired through various methods, including API, direct download, and web scraping.

The first dataset consists of Annotated Furniture Images, which were utilized in training the Object Detection Model. These images were sourced from the Open Images Dataset V6, comprising approximately 10,000 images distributed across six categories. The images in this dataset are annotated with image-level labels and object bounding boxes, providing valuable information for object detection tasks. Additionally, a separate set of Style-labelled Furniture Images was included in this dataset. The second dataset, the IKEA Product Catalogue Dataset, sourced from IvonaTau repository, serves as a recommendation source. Acquired through web scraping, this dataset contains information from the IKEA Product Catalogue. Around 1,400 products were collected from various categories, including Bed, Cabinetry, Chair, Couch, Lamp, and Table. This dataset is intended to contribute to the recommendation system aspect of the project, providing a diverse range of IKEA products for evaluation and analysis.

B. Data Preprocessing

In our data cleaning process, we implemented several common pre-processing steps and addressed special cases later on to enhance the quality of our results. Key techniques employed during data cleaning included:

- Balancing Categories in Dataset: Ensuring a balanced representation across different categories within the dataset to prevent biases and improve model performance.
- Image Resizing: Standardizing the size of images to a consistent format, facilitating uniformity in the dataset, and optimizing computational efficiency during model training.
- Dropping Duplicates: Removing duplicate entries within the dataset to eliminate redundancy and prevent skewed training.
- Image Augmentation: Applying real-time augmentation techniques to augment the dataset, such as rotation, flipping, or zooming, to increase variability in the training set

and enhance the model's ability to generalize to different scenarios.

C. Modelling and Architecture

Our modeling comprises a multi-stage approach for furniture recommendation, starting with an Object Detection model, MobileNet. This initial model is designed for detecting furniture within room images and extracting it. The different convolutional layers in the CNN(VGG16) act as feature extractors to generate feature maps for each furniture image. You can think of this step as the CNN creating custom "design filters" to encode the furniture designs.

The feature maps are then used to compute Gram matrices, which measure correlations between feature maps to highlight the most salient features that best represent the furniture. Mathematically, this is done by transforming the 3D matrix of feature maps into a 2D matrix and then computing its correlation matrix. The highly correlated feature maps will become strong latent representations of the furniture image.

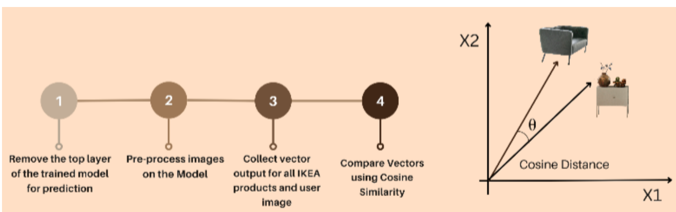

Fig. 2. modelling

Following the extraction of furniture images, our framework employs FAISS (Facebook AI Similarity Search) for image similarity detection, specifically tailored for furniture style identification. Gram matrices are high-dimensional representations of furniture design. To reduce them for the ease of similarity search, the matrices undergo dimensionality reduction using principal component analysis (PCA). The principal component vectors are saved as searchable indexes using Facebook AI Research's faiss library for search speed and scalability. FAISS facilitates the efficient indexing and retrieval of similar images, enabling a detailed exploration of furniture styles within the dataset. This step enriches our understanding of the diverse aesthetic characteristics inherent in the furniture images.

Subsequently, the recommendation system is introduced, where we integrate VGG16 in conjunction with transfer learning. VGG16 serves as a powerful feature extractor, capturing intricate visual patterns within the furniture images. Transfer learning enhances the model's proficiency by leveraging preexisting knowledge from a pre-trained network, optimizing performance even when faced with relatively smaller datasets. The reuse of a pre-trained model on a new problem is known as transfer learning in machine learning. The model uses the knowledge learned from a prior assignment to increase prediction about a new task in transfer learning.

The collaboration of these components—Object Detection (MobileNet), FAISS for style detection, and the VGG16-based recommender— helps in identifying the stylistic attributes and ultimately generating personalized recommendations. This sophisticated approach not only ensures accurate furniture detection and style characterization but also enhances the overall recommendation system's capacity to cater to diverse user preferences.

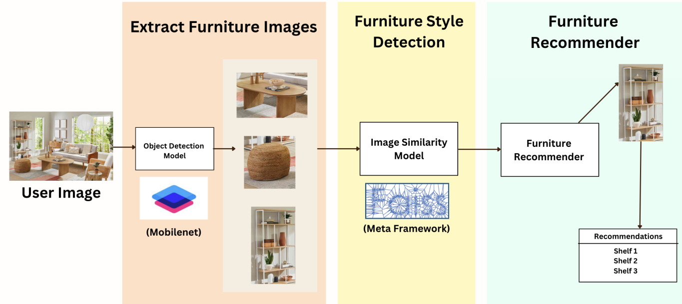

Fig. 3. Architecture

D. VGG16 vs VGG19

VGG16 and VGG19 are both convolutional neural network (CNN) architectures that belong to the VGGNet family. They were introduced by the Visual Graphics Group (VGG) at Oxford University and are known for their simplicity and effectiveness in image classification tasks. The main difference between the two lies in their depth, specifically in the number of convolutional layers. VGG16 comprises 16 layers (13 convolutional and 3 fully connected layers) and VGG19 extends to 19 layers (16 convolutional and 3 fully connected layers). VGG19's deeper architecture implies a higher number of parameters, potentially enabling it to capture more intricate features in data, but also demanding greater computational resources compared to VGG16. While VGG16 is computationally less intensive and requires fewer parameters, making it suitable for scenarios with limited computational resources and a smaller dataset, VGG19 may be preferred when tackling tasks that benefit from a more complex model, particularly when working with larger datasets. The choice between VGG16 and VGG19 ultimately hinges on the specific requirements of the task, available resources, and the nature of the dataset being used for training.

E. Transfer Learning with VGG16

In our research, transfer learning is harnessed as a powerful technique to leverage the knowledge embedded in a pre-trained VGG16 model, originally trained on the extensive ImageNet dataset. The implementation details of this process are encapsulated within the model.create_model() function, an essential component imported from the model module.

- Load the Pre-trained Model: The transfer learning journey commences with the loading of the VGG16 model, sans its top (fully connected) layers. This is achieved by setting include_top=False when invoking the VGG16() function.

By doing so, we tap into the wealth of knowledge encapsulated in the pre-trained convolutional layers, ready to be adapted for our specific task.

- Freeze the Layers of the Pre-trained Model: To preserve the wealth of features learned by the VGG16 model on ImageNet, the layers of the pre-trained model are frozen. This is accomplished by setting layer.trainable = False for each layer in the base model. This strategic decision ensures that the convolutional base retains its learned representations without being modified during subsequent training phases.
- Add Custom Layers: With the pre-trained base layers secured, new layers are introduced atop the base model to capture features specific to our unique dataset. This typically involves the addition of one or more Flatten or Dense layers, providing the flexibility to adapt the model to the nuances of our target task.
- Create a New Model: The culmination of the transfer learning process is the creation of a new model. This amalgamation combines the feature extraction capabilities of the pre-trained VGG16 base with the adaptability of the custom layers, forming a tailored architecture poised to excel in our specific domain.

Transfer Learning Process:

F. Model Space Optimization

In our research, we prioritize space optimization, focusing on making the model more compact for future deployment on mobile devices. We aim to streamline the model's architecture, reducing its computational and memory requirements. This aligns with the growing demand for efficient AI models tailored for mobile applications, emphasizing accessibility and adaptability to diverse deployment scenarios. Optimization techniques that are employed include:

- **Pruning:** The pruning technique implemented in the provided code is known as "magnitude-based pruning." This method is facilitated by the ***tfmot.sparsity.keras.prune_low_magnitude*** function from the TensorFlow Model Optimization toolkit.
- **Quantization:** Post-training quantization reduces the model size while also improving CPU and hardware accelerator latency, with little degradation in model accuracy. It does this by converting 32-bit floating point numbers to more efficient 8-bit integers or other lower-precision data types. The utilization of the TensorFlow Lite Converter streamlines the conversion process, ensuring compatibility with TensorFlow Lite for deployment on devices with limited computational resources.

The initial standard model, which weighed 760 MB, underwent a pruning process that strategically eliminated redundant and less critical neural network parameters. This resulted in a significantly optimized model, reducing its size to 395.3 MB while maintaining a commendable level of accuracy. Furthermore, to address computational constraints more comprehensively, we applied quantization to the pruned model. This involved the reduction of the number of bits used to represent the model's weights. By converting the floating-point parameters into a lower-bit precision format, we further diminished the model's size to a mere 98.9 MB. This dual optimization strategy—pruning for structural efficiency and quantization for precision reduction—resulted in a highly compact and resource-efficient transfer learning model (Fig. 4. shows comparison).

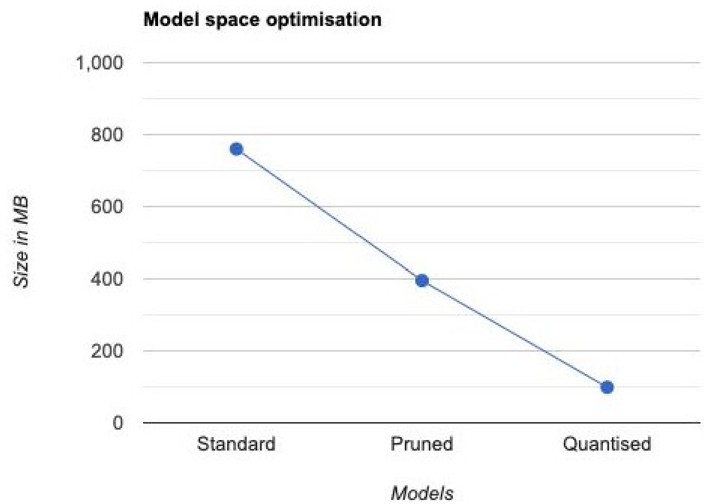

Fig. 4. Model space optimization comparison

G. Evaluation

Ideally, rigorous A/B testing was planned to assess the accuracy of its product recommendations. However, due to time and resource constraints, conducting such testing for Pair wasn't feasible. Consequently, an alternative approach was devised. The IKEA image dataset, which included item-to-room image and text information, presented an opportunity to evaluate our model's recommendations differently. The dataset featured curated furniture groupings in various room scenes, assumed to align with Swedish aesthetics.

These furniture arrangements, believed to be carefully selected by IKEA interior designers, served as a benchmark to measure the similarity between Pair's recommendations and IKEA interior designs. Specifically, the metric employed for this evaluation was hit ratio at n (HR@n). A "hit" was registered when one of Pair's top-n recommendations matched the IKEA interior designs. HR@n was calculated as the number of recommendations with a correct top-n match divided by the total number of recommendations issued. A high hit ratio, close to 1, indicated strong agreement between Pair's recommendations and IKEA interior designs, while a ratio of 0 suggested poor agreement. For example, if the HR@5 score was 0.5, it meant that in 50% of all furniture queries, at least one of our top-5 recommendations correctly matched one of the IKEA interior designs.

Conclusion

The image-based product recommender is a robust solution leveraging CNN-powered design feature extraction. Deployed as a Dockerized Streamlit container in the cloud, it excels in making cross-category recommendations based on design similarities. The live demo showcases its capability and skill in making accurate and effective recommendations based on design features or similarities. Evaluating it using IKEA room scenes, the hit ratio at 'n' emerged as a compelling metric, affirming its capability to align recommendations closely with IKEA interior designs. A transfer learning pipeline was established to further refine feature maps, enabling more precise detection of furniture pieces that are personalized and style-aligned product recommendations, continually evolving to enhance user experience.

Future Work

Future advancements involve introducing user feedback mechanisms that will enable personalized recommendations and learning from individual preferences. Exploring advanced deep learning techniques, like attention mechanisms or generative models, could refine its understanding of design elements. We can present a new approach for the "personalization" of text-to-image diffusion models. Given as input just a few images of a subject, we fine-tune a pre-trained text-to-image model such that it learns to bind a unique identifier with that specific subject. Once the subject is embedded in the output domain of the model, the unique identifier can be used to synthesize novel photorealistic images of the subject contextualized in different scenes[12]. This feature will offer users a clear and tangible representation of our recommendations within their own space, enhancing their understanding and aiding in decision-making. These enhancements aim to make it more adaptable, and personalized, and to understand user preferences and design elements, ensuring a context-aware recommendation experience.

Acknowledgment

We express our sincere gratitude to our esteemed instructors, Gaurav Singal and Preeti Kaur, whose unwavering support and guidance have been invaluable throughout the development of our model. Their expertise and mentorship have significantly enriched our understanding and skills in this endeavor. Our model is built upon the foundation laid by Austin McKay's StyleStack Github repository. Additionally, the IKEA catalog image dataset utilized for evaluation is sourced from Ivona Tautkute's Github repository.

References

[1] Yan Wang and Hengyu Wang, Xirui Li, An Intelligent Recommendation System Model based on Style for Virtual Home Furnishing in Three-dimensional Scene, International Symposium on Computational and Business Intelligence 2013

[2] M Sridevi et al, Personalized fashion recommender system with image-based neural networks, 2020 IOP Conf. Ser.: Mater. Sci. Eng. 981 022073

[3] Tse-Yu Pan, Yi-Zhu Dai, Min-Chun Hu, Wen-Huang Cheng, Furniture style compatibility recommendation with cross-class triplet loss, Multimed Tools Appl (2019) 78:2645–2665

[4] Murium Iqbal, Adair Kovac, Kamelia Aryafar, A Multimodal Recommender System for Large-scale Assortment Generation in E-commerce, ACM's Special Interest Group on Information Retrieval (SIGIR) workshops, 2018

[5] Mar´ıa Isabel Manresa Rom´an, A Machine Learning based Recommendation System for furniture selection, (UPC): Javier Ruiz Hidalgo Advisor, Pol Albacar Fern´andez Barcelona, 2022

[6] L.H. Li, R. Hsu, F. Lee, "Review of Recommender Systems and Their Applications". Computer Science, 2012.

[7] P. Covington, J. Adams, and E. Sargin, "Deep Neural Networks for YouTube Recommendations," Mountain View, California, 2016.

[8] A. Bellogín, P. Castells, I. Cantador, "Neighbor Selection and Weighting in User-Based Collaborative Filtering: A Performance Prediction Approach". ACM Transactions on the Web, No. 12, 2014.

[9] S.C. Stephen, H. Xie, and S. Rai, "Measures of Similarity in Memory-Based Collaborative Filtering Recommender System: A Comparison", 4th Multidisciplinary International Social Networks Conference, 2017.

[10] G.S. Milovanovic, "Hybrid content-based and collaborative filtering recommendations with ordinal logistic regression (1): Feature engineering", Data Science Central, 2017.

[11] M.G. Vozalis and K.G. Margaritis, "On the Enhancement of Collaborative Filtering by Demographic Data," in Web Intelligence and Agent Systems, Vol. 1, 20063

[12] M.G. Vozalis and K.G. Margaritis, "On the Enhancement of Collaborative Filtering by Demographic Data," in Web Intelligence and Agent Systems, Vol. 1, 20063