Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Product Recommender System Based on Visually Similarity

Lucia D. Krisnawati[2]

Ming-Wee Foo[1], Su-Cheng Haw[1,*], Nur Erlida Ruslan[1],

[1] Faculty of Computing and Informatics, Multimedia University, 63100 Cyberjaya, Malaysia
[2] Faculty Information Technology, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia

ARTICLE INFO

Keywords:
Recommender System; E-Commerce, Online Shopping; Visual Similarity; Convolutional Neural Networks (CNN); k-Nearest Neighbors (KNN)

ABSTRACT

The need, for recommendation systems in the online shopping industry has become more evident due to the impact of the COVID 19 pandemic, which has shifted consumer behavior towards platforms. Conventional recommendation systems based on user activity data often struggle to capture consumer preferences regarding product visuals. This study explores the emerging field of recommendation systems utilizing similarity to enhance product suggestions. By examining Convolutional Neural Networks (CNN) and k Nearest Neighbors (KNN) in analyzing product images for similarities we aim to assess their influence on recommendation accuracy and personalization. A key contribution of this study is blending visual similarity metrics with recommendation algorithms showing that this combined approach significantly boosts user satisfaction through personalized and relevant product recommendations. Research findings indicate that integrating visual similarity can notably enhance user satisfaction by offering relevant product recommendations. Additionally, we discuss how incorporating visual similarity metrics, into recommendation algorithms could transform e commerce by providing a tailored and immersive shopping experience. This analysis not provides insight into the status of visual based recommendation systems but also suggests future paths, for investigation and integration to improve online shopping platforms.

1. Introduction

The growth of e-commerce has made online shopping more accessible than ever, thanks to the spread of technology and the availability of internet enabled devices. There are, however, too many products available on the web which can confuse customers. This has led to increased competition for international e-commerce platforms, meaning they have to come up with better ways of increasing their revenues. In addition, the COVID-19 pandemic has also contributed significantly to this trend as lockdowns and social distancing measures have resulted in increased reliance on online shopping platforms by many consumers. This rise in online shopping like never before calls for more sophisticated recommender systems.

* *Corresponding author.*
E-mail address: sucheng@mmu.edu.my

https://doi.org/10.37934/araset.63.1.162187

Recommendation systems are increasingly important in improving customer experiences on e-commerce sites [1-3]. They utilize machine learning algorithms to suggest items that match users' preferences thereby enhancing buying experience [4,5]. By evaluating information from various sources such as past purchases and customer's individual needs, they give personalized product recommendations via different channels like email, apps and websites [4,5]. It allows customers to find new things concerning their interest.

Recommendation systems belong to a kind of artificial intelligence which is responsible for leading the customers to new products [6,7]. As filters, they handle large amounts of data so that they can show relevant content aligned with user preferences [8,9]. This way, not only is the searching process much easier but also information retrieval involves minimal efforts from users.

A well-designed and implemented recommender system can be very beneficial to a company in terms of increasing sales volume, profitability, and differentiation from other competitors [10]. These kinds of systems employ data analytics together with machine learning [8]to study consumption patterns by customers whereby businesses are able to make personalized product recommendations. These trends are captured through event analysis and user interaction identification within the systems such that the customer preferences will best match those recommendations. The method of recommending products is chosen carefully since it has a huge impact on the efficiency as well as accuracy of the system [11,12]. Proper methods used will ensure that the products recommended will appeal to the consumers more [13].

2. Background

2.1 Overview of Recommender System

Due to increased usage of online shopping, there is a huge amount of user data over time, resulting in information overload which makes it hard for shoppers to find the items they want quickly. To solve this problem, there has been increasing dependence on recommender systems. These are designed to improve the shopping experience by providing personalized product and service recommendations that align with users' preferences. Development of advanced recommender technologies is thus becoming essential for online retailers wishing to attract and retain customers.

According to the research of Roy and Dutta [14], these systems are very important in helping e-commerce businesses succeed by managing the tons of products available and highlighting only those that are most relevant to particular users. Subsequently, these systems not only prevent information overload but also enhance user interaction by aligning product recommendations with their preferences and interests. This personalization will encourage visits repeatedly, thereby increasing customer loyalty and generating more revenue. According to Hussien *et al.,* [15], in an increasingly competitive atmosphere of e-commerce, the effectiveness of these systems in delivering customized recommendations will be drawing and keeping customers. Moreover, for a varied e-commerce platform, it becomes essential to strike a balance between consumer demands and business objectives with the help of effective recommendation strategies.

Recommender systems have become essential for e-commerce platforms that aim to remain competitive. As data getting better at providing precise and customized suggestions, helping users quickly find what they need among numerous products. This capability is crucial for the success of online shopping sites, as it helps manage the overwhelming amount of information and improves customer satisfaction. Ultimately, effective recommender systems are fundamental to enhancing the shopping experience and ensuring the growth of e-commerce businesses.

2.2 Phases in the Recommendation Process

The recommendation process comprises three primary phases: information collection, learning, and prediction or recommendation. Figure 1 provides an overview of these phases within the recommendation process.

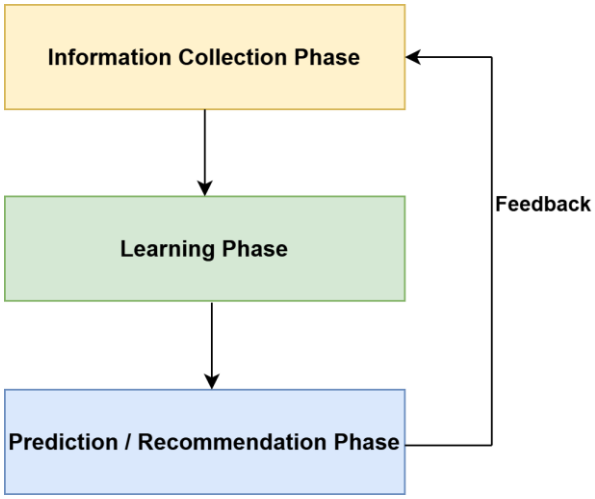

Fig. 1. Phases in recommendation process

In the information collection stage of any recommender system, data is collected from various sources to form user profiles. This can include information about users' activities on the platform, registration information, interactions on social media, and their shopping history. The system is supported by various types of explicit and implicit user feedback that will enhance its ability to recommend based on particular interests.

Explicit feedback in the form of user ratings, reviews, likes, and other expressions of preference significantly enhance the accuracy of recommendation systems. The processing of such feedback is a lifetime task aimed at improving usability and capturing as much information as possible to reflect the many forms of expressing preferences in the digital environment. Implicit feedback is obtained through actions such as clicks, viewing duration, and purchase history. The former provides a background information base regarding user preferences that are not directly indicated but rather inferred from activity. It is therefore possible for recommendation systems more accurately to adapt and diversify their suggestions to the fluctuating ways users interact with and engage with contents by combining explicit with implicit feedback. It understands a hybrid approach to a broader spectrum of user interactions; hence recommendations become more relevant as well as diverse.

In the learning phase of a recommendation system, many computational techniques are employed, originating mostly from machine learning. The system attempts to discover patterns and relationships in the data while attempting to match user preferences with attributes of various items. It involves training algorithms that will be applied to adjust recommendations based on specific tastes.

Next will be the prediction or recommendation phase, where it offers predictions or recommendation personalized to individual user based on the assessment of items using predefined user profiles and historical data to recommend the most appropriate recommendations [16].

2.3 Recommendation Techniques

Recommender systems form the key component of the modern digital ecosystem by providing personalized recommendations to users, helping them sift through the unbearable information. While applying types of collaborative and content-based filtering, they work with enormous data sets and track changes in user preferences over time. For researchers and practitioners in this field, it may be interesting to learn that recommender systems use different approaches, each with unique strengths and weaknesses.

Recommender systems employ diverse strategies intended to increase the accuracy and efficiency of the systems. Such systems are very important in attracting and retaining customers by providing unique personalized recommendations and offers to each individual. Techniques of this kind can dramatically enhance customer experience, which in turn fosters loyalty and subsequently increases a firm's revenue. Several critical areas have to be identified and addressed; for instance, the cold-start problem where there is no historical data available for a new user or new item for the system. Recognition of such problems makes recommendation systems more effective with better recommendations given becoming more accurate, usable, and valuable for each customer.

Recommender systems employ three basic techniques of item recommendation: collaborative filtering (CF), content-based (CB) filtering, and hybrid-based (HB) filtering. The latter integrates several approaches to enhance precision while offsetting the shortcomings inherent in solely utilized methods. CF provides recommendations based on the activities and preferences of similar users. Conversely, CB filtering recommends items based on an analysis of the attributes of the items and how these match the user's interests.

Advanced techniques add to the arsenal of personalized recommender systems. The filtering types that fall under semantic-based filtering aim at capturing the meaning of both the items and the user's preferences. Ontology-based filtering, in contrast, employs structured models to map relationships between users and items, giving an extended view of user tastes.

2.3.1 CB filtering technique

A CB system recommends products to users by analyzing the characteristics, categories, or other attributes of the products. It also considers the user's interests and preferences to customize recommendations that are likely to be relevant. Thus, it focuses on a match between product features and the user's exact tastes and needs. In terms of auxiliary data analyzed, characteristic texts, images, and videos associated with both products and consumers form the basis of CB recommendations. The methodology enables exhaustive comparisons for deriving tailor-made suggestions [17].

Kaur and Bathla [18] stated that in CB filtering, the recommendation is matched with a profile developed from the user's initial interaction with the system. The approach is quite dependent on the characteristics of items such as products, movies, or articles and how these traits correspond to user preferences. A user profile is created by analyzing the features of items that the user has previously liked or interacted with. Items are recommended that have similar attributes; thus, they are likely to be appealing to the user. The precision and efficiency of this system highly depend on the accuracy and completeness of item descriptions and definitions as well as strength-user profiles. Therefore, it can provide recommendations that are relevant as well as accurate.

The CB filtering employs various models to detect similarity among items or documents so that recommendations given are relevant and accurate. These models analyze the properties or characteristics of the items to establish similarity. Huang [19] stated that, in this approach, the Vector

Space Model is utilized wherein techniques such as Term Frequency Inverse Document Frequency (TF-IDF) and Probabilistic models are used, including the Naïve Bayes Classifier, Decision Trees, and Neural Networks. All these methods try to explain how documents within a collection relate to one another. They apply either statistical analysis or machine learning to develop algorithms that yield recommendations based on an inherent model. While CB filtering excels in delivering personalized suggestions by leveraging information about the products and user preferences, its effectiveness mainly depends on the quality and depth of knowledge regarding attributes of items feature products. Thus, capturing these characteristics better and representing them will be the key factor for achieving better accuracy as well as relevance in recommendations within CB systems.

2.3.2 CF filtering technique

The CF technique is a key method used in recommender systems, commonly seen on various e-commerce sites. This technique improves a user's shopping experience by identifying similarities in choices among consumers [20]. Essentially, it assumes that users with similar tastes are more likely to appreciate the same items. This approach helps in making more personalized recommendations, ultimately aiding users in making better purchasing decisions. Such systems suggest products by analyzing explicit ratings given by users who share similarities. A new user's preferences are compared with the existing database to identify like-minded individuals, or 'neighbors', to predict potential interests effectively [21].

According to Li *et al.,* [22], the CF algorithm is the most well-known and accepted algorithm due to its many advantages. Further, it is easy to implement with low data dependency and offers accurate recommendation results. Basically, CF makes recommendations by learning from user-item historical interactions, either explicit like user's previous ratings or implicit feedback, like browsing history. To agree with Karabila *et al.,* [23], a collaborative recommender system harnesses the collective insights from a user community, tapping into their expressed interests in various items to infer recommendations for others. These systems deploy CF techniques to scrutinize vast datasets reflecting users' principle that individuals with comparable tastes on certain items will likely agree on others.

In addition, CF recommendation operates on the premise that individuals with comparable tastes and preferences will have similar interests over time, and these interests can be inferred from their past behaviours. The CF algorithm relies on user activity, identifying close counterparts among users to project their future preferences based on the observed preferences of their peers. Basically, CF methods are developed by collecting extensive data on user behaviour, activities, and preferences. They predict individual user preferences by identifying patterns of similarity among users [24].

CF technique can indeed be categorized into two main categories: model-based CF and memory-based CF, each with its approach to generating recommendations based on user-item interactions.

Generally, the model-based method predicts unseen items by populating the matrix with projected ratings [18]. These predictions are facilitated by employing data mining methodologies to analyze the existing preferences and behaviours. This technique uses existing user-item ratings to build a model that predicts or fills in missing values in the user-item interaction matrix. This method seeks to improve collaborative filtering by suggesting products that users have not yet explored.

Model-based CF employs a proactive strategy, using probabilistic techniques to anticipate and suggest contents. It prescribes the prior construction of a knowledge model that blends information about users or items and features scores assigned to items within a particular framework [16]. This approach enhances accuracy and efficiency in predicting user preferences as well as recommending interesting items.

Furthermore, in accordance with Nilashi *et al.,,* [20], model-based methods in CF algorithms leverage collective user ratings to create predictive models. These models efficiently estimate user preferences and address scalability issues commonly associated with memory-based methods, such as k-NN, by simplifying the recommendation process. Model-based algorithms offer a distinct advantage over memory-based ones by eliminating issues such as scalability and storage. These algorithms leverage precomputed models to make quick predictions, significantly reducing the time required for real-time computations compared to memory-based approaches that require on-the-fly calculations.

The CF Algorithm's memory-based definition measures how similar one customer is to another. No need for pre-calculation, no offline design to be able to make prediction or suggestion in memory-based CF methods. The most typical ones are scalability and sparsity [16]. It has been argued that every user with like interests belong to the same group [25]. By finding neighbours of a new user or currently active user, anticipatory preferences on a new item can be generated. As you can see, memory-based CF technique can be divided into two main types: item-based and user-based CFs. In relation to these two techniques, the nearest neighbourhoods which are typically referred as user-based and item-based are normally their memory-based CF approaches' two basic NNH methods [16].

On other hand, item-based CF algorithm works under the philosophy "if you like this then you may also like that" as observed by Ajaegbu *et al.,* [26]. Amazon came up with an algorithm that looks into the products. It is done by choosing suggestions through buying or rating of products from customers and then combining them with others like them via such metrics and lists of recommendations. First, item-based CF examines user-item matric which indirectly leads to these connections for consumers by means of this way.To identify relations among dissimilar items, item-based CF analyses the user-item matrix first and circuitously calculates recommendations for clients by means of these relations. For example, CF algorithms mostly combine feedbacks on items given by various users and define the similarities between items and items (item-based) or between users and users (user-based) to give a suggestion to a specific customer [20].

The User-Based Collaborative Filtering engine works on "people like you" model where it suggests an item already liked by similar people [26]. CF algorithm operates by collecting user preference data, then use KNN algorithm to compute cluster of nearest users to conclude common preference of N nearest users, after that based on degree of common preference provides non-common preferences towards users.

2.3.3 HB filtering technique

HB filtering techniques combine more than one filtering approach. The HB approach is developed to address certain challenges that are prevalent in traditional filtering techniques like collaborative, content-based, and demographic filtering. One of the main challenges is the cold start problem, where occurs due to users have insufficient interaction data or new items which causes difficulty to make accurate recommendations; overspecialization, which leads to a narrow range of suggestions; and the sparsity problem, where there's too little data due to a large number of items but relatively few ratings. The HB filtering seeks to enhance both the precision and the efficiency of recommendations by integrating the strengths of the aforementioned filtering methods [27]. According to Nahta *et al.,* [28], previous solutions to the issue typically relied on direct user feedback. However, in collaborative models, a more refined measure of similarity has been introduced to mitigate issues encountered when new users join the system. Besides that, a hybrid model in the context of recommender systems denotes an approach that amalgamates multiple recommendation

techniques to harness their collective strengths [17]. A hybrid system approach addresses these limitations by integrating multiple data inputs and merging several algorithmic strategies or recommendation modules into a single system [27].

HB recommendation system often emerges from blending multiple methods to mitigate their individual limitations. Such systems synergize collaborative and content-based mechanisms to balance out their respective shortcomings. The hybrid techniques can be categorized operationally into various type: weighted hybrid, mixed hybrid, switching hybrid, feature-combination hybrid, cascade hybrid, feature augmented hybrid and meta-level hybrid, each offering unique enhancements to recommendation accuracy and relevance [27].

2.3.4 SB filtering technique

SB filtering techniques introduce meta-knowledge about item semantic attributes or characteristics into the recommendation process, often outperforming traditional filtering methods. This technique ensures that recommended items align with users' preferences and interests. Users' information, like their preferences, is often store in personalized profiles. This data can be collected explicitly by asking users to provide specific details about items or gathered implicitly from their interactions with the system, such as reviews, ratings, and transaction history.

Employing domain or reference ontology to establish connection between user profiles and item features is the most effective approach. It helps represent items in a structured manner to link user profiles with the reference ontology representing items. This recommended item aligns more accurately with their interests. SB filtering techniques consists of four main phases: ontology construction, profile refinement, nearest neighbour search and performance evaluation.

In the ontology construction, the system creates a detailed reference framework or items by collecting information from diverse sources. It also constructs personalized user profiles based on these items, linking them through an interest score to measure relevance. In the profile refinement phase, the user profiles are updated using a method that spreads activation; thus, improving the accuracy. Subsequently, in the Nearest Neighbour Search, users with similar interests are identified, thus, helps in tailoring personalized recommendations. The final step involved evaluation based on metrics like Mean Absolute Error (MAE), Recall, F-Measure, and Precision.

Besides that, there are two methods that can be applied in SB filtering technique: the ontology-based filtering technique and graph-based filtering technique.

Ontology-Based Filtering Technique: According to Nilashi *et al.,* [29], ontology-based recommender systems use a structure hierarchy of user and item profiles to improve recommendations, browsing, and the creation of user profiles. These systems overcome some limitations of collaborative filtering by organizing items into a knowledge framework. However, it can be challenging to distinguish items within the same category using ontologies. In these systems, user preferences are mapped against the ontology's structure, with each aspect receiving a weight based on both explicit interactions, like ratings, and implicit ones, such as the frequency of views or reads.

Graph-Based Filtering Technique: According to Ong *et al.,* [30], in the context of recommendation systems, a graph-based clustering algorithm can be employed to categorize users into distinct groups on a user-item graph. This graph is bipartite, illustrating the connections between users and the items they have engaged with, through ratings or other interactions. Connections in these networks can have weights, representing the strength or nature of interactions. Grouping users with similar tastes into clusters improves the system's recommendation accuracy and relevance. However, developers of these tools encounter challenges including data scarcity, scalability, complex initial setups, limited time and resources, and the risk of overwhelming users with too much information. To address these

issues, a new approach has been developed where rating data is represented using graphs, enhancing the system's ability to predict accurately [31,32].

2.4 Summary of Recommendation Techniques

In prior section, different data filtering techniques used in recommender systems, each with its unique approach. It's important to thoroughly understand the features and capabilities of each technique to choose the most appropriate one for a specific recommender system. This section aims to contrast the techniques' differences, highlighting their strengths and weaknesses. Table 1 outlines the advantages and limitations of each data filtering technique employed in recommender systems.

Table 1
The advantages and limitations of each data filtering technique employed in recommender system

Approach	Advantages	Limitations
CB	1.Content-based recommender system craft unique user profiles based on individual preferences and interactions, ensuring personalized content curation without the influence of others' opinion. 2. They offer clarity on their operational mechanisms, allowing users to understand the basis on which recommendations are made. 3. These systems excel at suggesting new and unrated items, benefiting users who are new to the platform by expanding their discovery horizon.	1. Generating attributes for products in specific domains can be challenging. 2. CBF tends to promote homogeneous items, leading to an overspecialization issue. 3. Obtaining user feedback in CBF is challenging since users often do not rate items as they do in Collaborative Filtering (CF), making it difficult to gauge the accuracy of recommendations.
CF	1.Memory-based Collaborative Filtering techniques enable rapid deployment of recommender systems. 2. This technique allows for seamless and incremental integration of new data. 3. In contrast, model-based collaborative filtering enhances the accuracy of predictions.	1. The cold start problem CF arises from the need for substantial pre-existing user data to make precise recommendations. 2. In terms of scalability, CF must cater to environments where billions of users and product exist, necessitating significant computational resources to generate recommendations. 3. The issue of sparsity is highlighted on major e-commerce platforms, where the vast array of items means that only a small fraction of the product range receives ratings, typically by the most active users, leaving popular items with relatively few ratings.
HB	1. Combining diverse filtering methods improves recommendation precision by leveraging the strengths of different approaches. 2. Hybrid filtering overcomes drawbacks present in individual techniques, offering more robust recommendations 3. Utilizing various data sources effectively tackles challenges related to new or less active/items, ensuring better recommendations from the outset. 4. Offers a more tailored recommendations experience by amalgamating multiple techniques, catering to various user preferences and behaviours.	1. Hybrid-based filtering typically requires higher computational power, adding to the processing requirements of the system. 2. Tends to be more complex both in terms of space (resource allocation) and time (processing duration). 3. It often incurs higher expenses due to the need to integrate and manage multiple algorithms or methods.

SB	1. Have better performances and accuracy than traditional filtering techniques.	1. More complex technique
OB	1. OB filtering leverages structured knowledge representations to understand relationships and context between items, enhancing the system's comprehension of user preferences. 2. Using ontologies to model user preferences and item characteristics, recommendations become more tailored and personalized, aligning with users' interests. 3. Allow the integration of domain-specific knowledge, enabling systems to make more informed and contextually relevant recommendations within specialized areas.	1. Reduced accuracy in Multi-Class Classification. In OB filtering, multi-class classification might lead to lower accuracy than binary-class classification. This limitation impacts the overall accuracy of the classification process within OB filtering systems.
GB	1. GB filtering captures transitive associations within item, item or user and item relationships. This ability is particularly advantage in dealing with scarce data or limited coverage, enhancing the system's ability to recommend items based on indirect connections. 2. Graph structures allow for a rich representation of relationships among items or users. This comprehensive view enables the systems to uncover complex connections and dependencies, leading to more contextually relevant recommendations. 3. GB models can scale efficiently, accommodating growing datasets while maintaining performance. They also offer flexibility in incorporating new data or relationships, making them adaptable to evolving recommendation needs.	1. GB filtering is primarily designed for systems reliant on rating or binary feedback. 2. Struggle to capture the precise order of user preferences, potentially leading to challenges in accurately ranking recommendations according to user preferences.

2.5 Related Works

Ay *et al.,[*33] constructed a visual similarity recommendation system through two primary phases. Firstly, the exploration of optimal methods for learning deep feature representations from images is conducted. The approach draws inspiration from Information Maximizing Generative Adversarial Networks (InfoGAN), leveraging mutual information to generate meaningful representations. In addition, popular pre-trained models such as Densenet, Resnet, Inception-V3, MobileNet, and VGGnet are utilized by removing their last layers for the shoe recommendation task. Secondly, a simple distance calculation is employed between query image features and others in the dataset. The proposed network architecture, a variant of InfoGan, involves modifications like Batch Normalization, Dropout, Leaky ReLU for the discriminator, and Batch Normalization, Dropout, ReLU for the generator. The training results highlight the idea points where the generator's loss is lower than the discriminator's real loss, indicating the generation of realistic images resembling the dataset. Evaluation metrics encompass model size, inference time for feature vectors, and precision rates against other popular pre-trained models, notably showcasing that the proposed model, delivers higher precision rates despite its smaller size and faster inference time.

In assessing the proposed generative network, the study employs a set of evaluation metrics that include the size of the model, the time it takes to generate feature vectors, and the precision rates. These metrics are crucial in comparing the effectiveness of our generative network with other widely-used pre-trained models based on convolutional neural networks. The benefits of our modified

model, which is based on the InfoGAN architecture, include faster processing times and more accurate results compared to larger, well-known models like VGG16, VGG19, and Resnet152. This is because out model is better at learning complex data patterns quickly. However, there are some possible drawbacks. The model's ability to assess how similar images are might not be fully reliable, and its effectiveness could be depend heavily on the particular characteristics of the dataset used. This could make it less effective when applied to different kinds of data. Furthermore, although out model performs well in terms of speed and accuracy, more research is needed to confirm these results across a variety of datasets and uses.

Next, Sheridan *et al.,* [34] proposed a holistic system that marries CNN with Association Rule Mining for fashion recommendation engine. This proposed model will work under two scenarios: one where the user uploads images of products and the system, based on image recognition using CNNs and Association Rule Mining, suggests related items; and the other, where the system makes recommendations from the user's past history using the CF approach. This model leverages CNNs for product recognition from images and uses association rule mining for suggesting frequently purchased together or related items. Further improvements in the recommendation system are suggested to be CF methods based on similar user preferences.

To evaluate, these are the support and confidence for Association Rule Mining, CNNs accurate image classification, and model performance validation accuracy measures. The advantages lie in the integratingof image-based identification and recommendation with Association Rule Mining, enabling personalized suggestions based on user history. However, potential drawbacks include limitations in handling diverse or rapidly changing fashion trends and the reliance on historical data for recommendations, which may affect adaptability to evolving preferences and emerging fashion styles. Additionally, reliance on image-based identification might encounter challenges with accuracy in complex, less distinguishable product images. Nonetheless, the model exhibits promising capabilities in leveraging image recognition and historical data for tailored fashion recommendations.

Kang *et al.,* [35] proposed a model for the Complete the Look task aiming to measure scene-product style compatibility. They utilized ResNet-50 to extract visual features from scene and product images, transforming them into a unified style space using a two-layer feedforward network. Their approach measures global compatibility by evaluating the distance between scene and product embeddings. It introduces local compatibility by matching scene regions with the product, incorporating category-aware attention to weight region relevance. The model's objective function involves hinge loss considering triplets of scene, positive, and negative products to optimize the style embeddings.

The authors compared their model's recommendation performance with different baselines in accuracy and Top-K accuracy. They further explored the attention regions and conducted a human study to check how consistent the model is with the human sense of fashion. On the other hand, the qualitative results indicated that the model was able to produce compatible products. The proposed model takes into consideration global and local compatibility by use of attention mechanisms to discern relevant areas within scenes.

The advantages of this model are that it has global and local compatibility. From the model come the advantages of global compatibility, where measures for both local and global compatibility account for style coherence and scene-product matching at over-arching and fine-grain levels. An attention mechanism that allows the model to focus only on the relevant regions inside the scene images helps in understanding the compatibility more accurately. In addition, the model showed comparative performance with human fashion experts, showing captured fashion sense. The authors further perform quantitative evaluations and thorough analyses, such as human evaluation, attention visualization, and qualitative assessment, which collectively assist in understanding what

the model does or why it reasons. The complexity of scene analysis, especially within the home domain with many objects present, challenges the model in identifying all the key areas for compatibility assessment. Its performance seems to be based on the complexity of scene, suggesting potential problems in handling a variety of scenes uniformly. It may rely very much on the quantity and quality of the training data available; hence, this could limit the model in predicting the evolving fashion trends.

In another research, Addagarla and Amalanathan [36] preposed to develop a model that is tailored to present a visual product recommendation E-commerce Similar Image Network (e-SimNet). This model will use SqueezeNet architecture, known for its parameter efficiency, in the first stage for feature extraction from the input images. This approach is to make use of models that use 1x1 filters, reduce the input channels, and do late downsampling to improve the performance. In the model, the processing features are drawn from the convolutional layers that contatin spatial information, color, and texture. It also adds Bach normalization for the normalization of output values, which helps prevent the "dying ReLu" problem, and the pooling operations support down-sampling while retailing dominant features. The model was composed of the ANNOY algorithm for approximate nearest neighbor search. Similarity should be computed with the help of Euclidean distance measures, while the top-N most similar products should be recommended.

The merits of the approach are that it uses the SqueezeNet architecture strategically; optimized for feature extraction with reduced parameters, downsampling well with efficient techniques of keeping the important information, and the addition of ANNOY with the intention of nearest neighbors search makes it computationally efficient. However, the possible limitations could be the model's sensitively to post-training image variations and, hence, post-training will really have an effect on the accuracy in giving recommendations for products with altered orientations. In addition, computational requirements for processing large datasets can offer a memory and processing power challenge. But, proposed e-SimNet method contributes to promising results of top-N product recommendations. The Euclidean distance metric was used in this study to retrieve the items that are very similar to the search query and hence generate the top n recommendations of visual similarity [36].

On the other hand, Tayade et al., [37] proposed an exhaustive approach toward developing a recommendation system for clothing using deep learning and cosine similarity. The author amalgamates datasets from Kaggle and the DeepFashion Database, preconditions the images, and utilizes them in various computer vision tasks that include Clothes Detection, Clothes Recognition, and Image Retrieval. Preprocessing involves resizing images to some fixed dimension and changing them into suitable formats for deep learning models. For precision as well as rich information extraction, they employ transfer learning with a pre-trained VGG16 model to extract salient features from the images. Cosine similarity calculates the similarity of images; thus it creates a similarity matrix. This matrix drives a web application in which users select an item prompting the recommendation of visually similar products based on equivalence scores generated similarities. The system permits guest access as well as user accounts so that instant purchasing can be facilitated, and detailed email summaries of purchases are sent.

The proposed system uses pre-trained deep learning models for feature extraction and cosine similarity to determine the visual similarity of clothing items. The resemblance between the two images is measured using cosine similarity scores and similarity matrices. This effectively utilizes transfer learning in feature extraction, so the system can recommend visually similar items with great accuracy. There are several factors that contribute to the experience: friendly web interface, instant purchase options, guest and user account support, and detailed email summaries. However, limitations might arise in capturing nuances visual features due to the fixed dimensionality of images

and potential constraints in handling diverse clothing categories or styles within the dataset. Additionally, as mentioned in the long training time for the VGG16 model, computational load training and processing such a large amount of image data is another possible downside.

Towards more recently, Ahsan *et al.,* [39] proposed a comprehensive approach to object detection and visual search in home scene images to recommend similar or exact furniture and décor products. The methodology involves two key steps: Object Detection on Scene Images and Visual Search. Object, detection entails training specific models for eight rooms and generating bounding boxes around furniture and décor products. The visual search employs two approaches: color matching using RGB histograms and a triple network using ResNet-50 embeddings. The triplet network leverages a triplet loss function to map images into a semantic space capturing high-level information. The system aims to retrieve exact matching products or similar ones when exact matches are unavailable in the catalog.

Evaluation metrics include top-k accuracy for exact matching products and a user study for similar product retrieval. The advantages lie in the detailed analysis of different matching techniques, where the triplet network outperforms other methods for exact and similar matches. However, challenges exist in adapting the model's performance to customer-generated review images, indicating the need for tailored strategies for different décor categories. The system's online performance showed increased user engagement and average order value, validating the relevance of the visual search results, but further optimization might be necessary for diver real-world scenarios. In addition, network visualization highlights the network's focus on specific characteristics of furniture items, aiding in understanding its decision-making process.

In another research, Hiriyannaiah *et al.,* [40] proposed a comprehensive end-to-end system for visual search and recommendation, which includes the following stages: Data Preprocessing, Feature Extraction with CNN and CAE, Classification, and Recommendation with the Deep Visual Ensemble Similarity (DVESM) approach. First, the images are preprocesses to extract features. Classification follows, utilizing CNN models such as VGG and InceptionV3. Later, these features are fine-tunes using CAE to fine-grain the features that are captured and are discrete details needed in recommendation. The Recommendation System applies DVESM that combines several similarity metrics to compute the similarity between query and catalog images.

The proposed model highlighted on various datasets using metrics like AUC, RMSE, and Recall. The values of AUC and RMSE show the superiority of DVESM over the baseline methods in the accuracy of recommendations and errors reduction. However, while DVESM does well over it, it suffers in some categories like Sunglasses and Shoes, whereby other methods like VisRank take over due to the differences in feature extraction and weight updates.

The advantages of the proposed model, which enhance the recommendation accuracy. The DVESM promises an improved accuracy in recommendation that the baseline methods, more so, in the event of sparse datasets. The combination of both CNN and CAE is able to extract complex features, and with that, it is able to improve effective recommendation characteristic for subtle features. However, some category-specific challenges limit it, like Sunglasses and Shoes, where the feature extraction and weight updates differences cause the model to do poorly and hence has better results in alternative methods like VisRank. This has been the reason this model has been so effective, given tuning across many parameters and metrics-increases not only the complexity but also the potential computational requirement.

3. Methodology

3.1 Dataset

The prototype's model training dataset is the Fashion Product Images dataset on Kaggle, a comprehensive collection that includes high-resolution images of various fashion products. The dataset is structured with multiple category labels. It includes descriptive information for each item, making it suitable for categorization and recommendation in machine learning models related to fashion items. Besides that, there are 10 columns attributes in this dataset. Table 2 depicts the metadata of the dataset.

Table 2
Attribute descriptions of the selected dataset

Attribute Name	Details	Field Type
ProductId	Unique identifier for each product	Integer or String
Gender	Gender to which the product is catered (Men, Women, Boy, Girl)	String
Category	General category of the product (Apparel, Footwear)	String
SubCategory	More specific category within the main category (Topwear, Bottomwear, Dress, Socks, Apparel Set, Innerwear, Shoes, Flip Flips, Sandal)	String
ProductType	Specific type of the product (Tops, Capris, Shorts, Tshirts)	String
Colour	Color of the product	String
Usage	The intended use or occasion for the product (Casual, Formal)	String
ProductTitle	The title or name of the product	String
Image	A file that links the product to its specific product image	File
ImageURL	The URL where the product image is located	String

3.2 Data Preprocessing

Before the dataset can be processed for model training, the initial step involves data cleaning. This essential phase employs a streamlined pipeline to ensure the data is properly prepared and optimized for subsequent training steps.

i) Step 1: Handling missing data.
   - Use the 'sum()' to find out the number of missing/null values in the dataset.
   - No missing or null values in the dataset indicate the dataset is well structured.

```
ProductId       0
Gender          0
Category        0
SubCategory     0
ProductType     0
Colour          0
Usage           0
ProductTitle    0
Image           0
ImageURL        0
dtype: int64
```

ii) Step 2: Remove duplicated data
   - The number of the original data and the dataset after dropping the duplicate data remain the same, indicating there is no number of duplicated data.
   - This can be proved by the number of duplicated data equal to '0' after the checking.

```
'Original Count of Dataset:'
2906
'Number of duplicated data:'
0
'Dataset after dropping duplicate data:'
2906
```

iii) Step 3: Image URL validation
- The ImageURLs have been checked to ensure they have a valid format that looks like URLs.
- The validation check for ImageURLs shows that all 2906 URLs in the dataset are valid according to the regex pattern. Based on this pattern check, no invalid URLs indicate all the ImageURLs are valid and usable.

```
'ValidImageURLs: '
2906
'ValidImageURLs: '
0
```

3.2 Recommender System

After the data preprocessing process, the CNN and KNN added wutg CB filtering techniques are now prepared to be fitted into the recommender system. Combining CNNs and KNNs with content-based filtering in a recommender system allows leveraging the strengths of both: the strengths of CNNs and KNNs to understand and extract meaningful visual patterns from images and the capability of content-based filtering to match items with user preferences based on item features. In addition, there are multiple helpful built-in features for constructing a recommender system, including train-test split and various metrics.

Furthermore, the dataset is divided into train, validation, and test sets. The proportions for these sets are 60%-20%-20%. Outliers were determined using the Z-score method and interquartile range (IQR) method, considering threshold ±3 for the former and 1.5 times IQR for the latter. Continuous outliers were clipped to the nearest valid range or replaced with the feature median, while categorical outliers were reassigned to valid categories or excluded based on frequency. Min-Max scaling was applied so that all features would fall within [0, 1], which was required by KNN and CNN models. One-hot encoding was performed for categorical features 'Gender' and 'Category'. The stratified split was done to maintain the distribution of 'Gender' and 'Category' in different subsets. A reproducible pipeline implemented all preprocessing steps to ensure consistency and ease of scalability. It enhances transparency as well as reproducibility in methodology.

Next, Euclidean distance is calculated as the square root of the sum of the squared differences between the corresponding elements of the two vectors. Subsequently, the algorithm is trained using the training set, and the validation set is used to compare the performances between the models while the model with the better performance will then be used in the test set. This prototype uses metrics like MAE, Root Mean Squared Error (RMSE), Mean Squared Log Error (MSLE), Median Absolute Error (MedAE), Precision and Recall at K, which help identify the extent of prediction errors.

For enhanced visualization of descriptive analytics, dashboards are developed to streamline the representation of relationships between attributes. The data are visualized with a bar chart and the images can also be viewed according to the categories and subcategories. Additionally, pivot charts are utilized to facilitate an easier understanding of node connections and improve data visualization.

3.3 GUI for Administrator

The prototype is designed for two users; the administrator and the customer. The administrator may login or register an account. After the account has been registered successfully, the administrator can login by using the username and password registered. After successfully log-in to the system, the administrator will go to the main page (see Figure 2). The main page will initially will show the data visualization first. Additionally, on the main page, a navigation sidebar is placed on the left side. In the navigation sidebar, a greeting will be displayed for the administrator along with the 'Data Visualization' and 'Pivot Table' options.

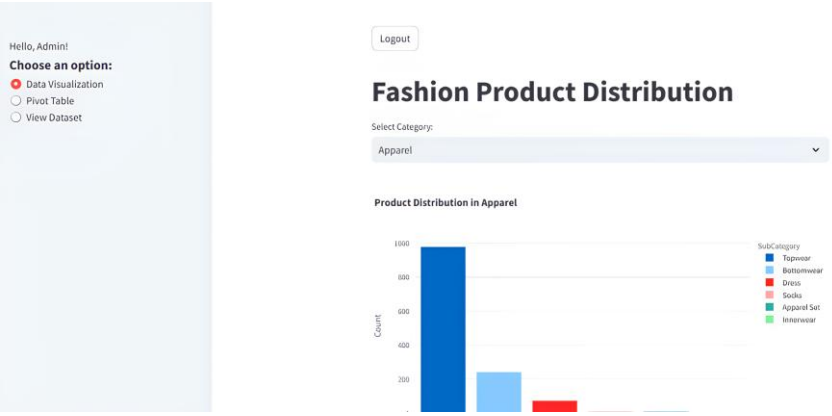

Fig. 2. Main page for the administrator

The dataset chosen in this research study is E-commerce Product Images from online open source, Kaggle. When the option 'Data Visualization' is selected, a bar chart indicates the fashion product distribution will be shown up. The bar chart will show the total number of products under each subcategory with different colours for better visualization (see Figure 3). Besides that, the administrator can choose between 'Apparel' and 'Footwear' to see the different bar charts for both categories.

Fig. 3. Bar chart for total product for each category and the category options for selections

In additional, the administrator can also view the sample images for each subcategory under the bar chart. The options between subcategories can be selected, after the selection, the first 12 images which under the same subcategory will show up (see Figure 4). When the administrator selects the 'Pivot Table' option, a pivot chart which show the product count by category and subcategory will be shown as depicted in Figure 5.

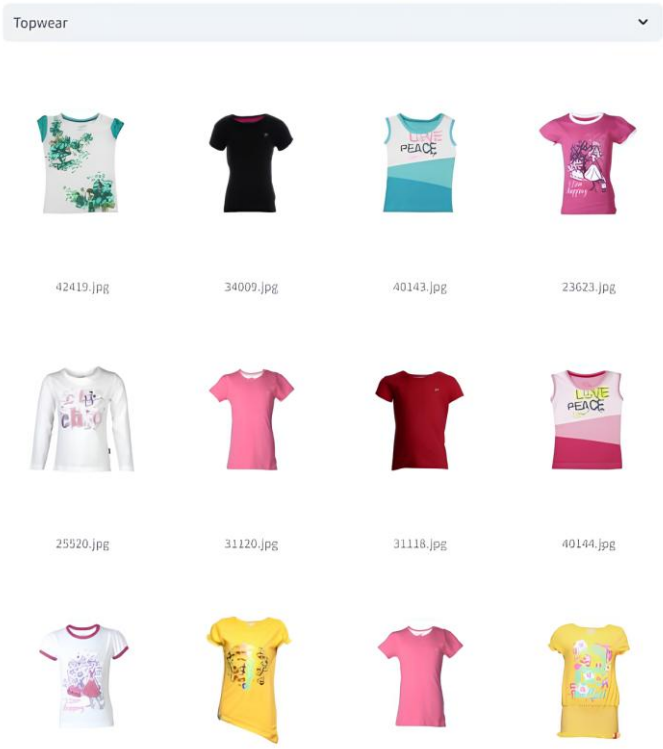

Fig. 4. Images within the same subcategory

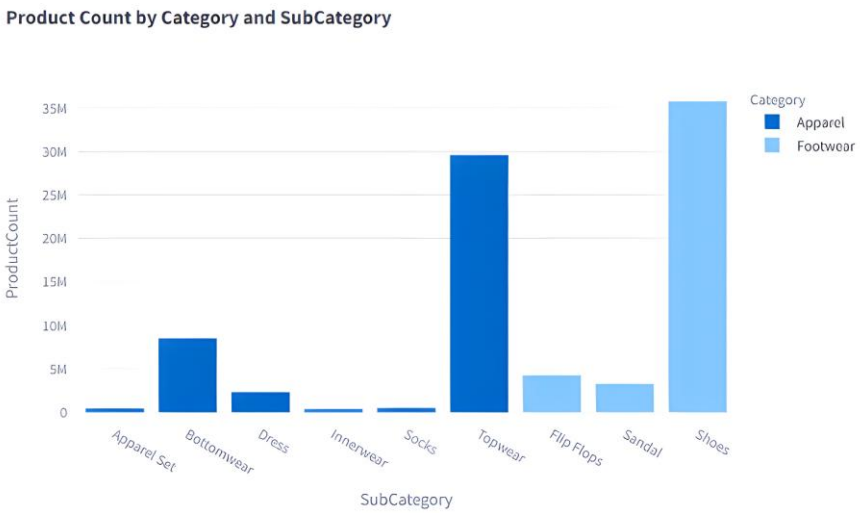

Fig. 5. Pivot table

The administrator can also view the dataset by selecting the “View Dataset” on the menu side-bar and the system will navigate to the dataset page. The sample output is depicted in Figure 6.

	ProductId	Gender	Category	SubCategory	ProductType	Colour	Usage	ProductTitle
0	42,419	Girls	Apparel	Topwear	Tops	White	Casual	Gini and Jony Girls Knit
1	34,009	Girls	Apparel	Topwear	Tops	Black	Casual	Gini and Jony Girls Black
2	40,143	Girls	Apparel	Topwear	Tops	Blue	Casual	Gini and Jony Girls Prett
3	23,623	Girls	Apparel	Topwear	Tops	Pink	Casual	Doodle Kids Girls Pink I I
4	47,154	Girls	Apparel	Bottomwear	Capris	Black	Casual	Gini and Jony Girls Black
5	25,520	Girls	Apparel	Topwear	Tops	White	Casual	Doodle Kids Girls City Ch
6	31,120	Girls	Apparel	Topwear	Tops	Pink	Casual	Palm Tree Girls Pink Top
7	31,118	Girls	Apparel	Topwear	Tops	Red	Casual	Gini and Jony Girls Red T
8	54,923	Girls	Apparel	Bottomwear	Capris	Olive	Casual	Do u speak Green Girls C
9	31,127	Girls	Apparel	Dress	Dresses	Black	Casual	Gini and Jony Girls Black

Fig. 6. View dataset

3.4 Similar Product Recommendation for User

The system used the hybrid-based method, combining CB filtering with CNN feature extraction. This will be one of the model's comparisons which will be used to compare with other models. Before the user interact with the system, there will be a model testing for generating a similar product based on visual similarity and recommend the top 5 similar product. Figure 7 shows the partial view of the recommendation list based on the 'ADIDAS Men Adi Quest Blue Sports Shoes' product (product id '13683').

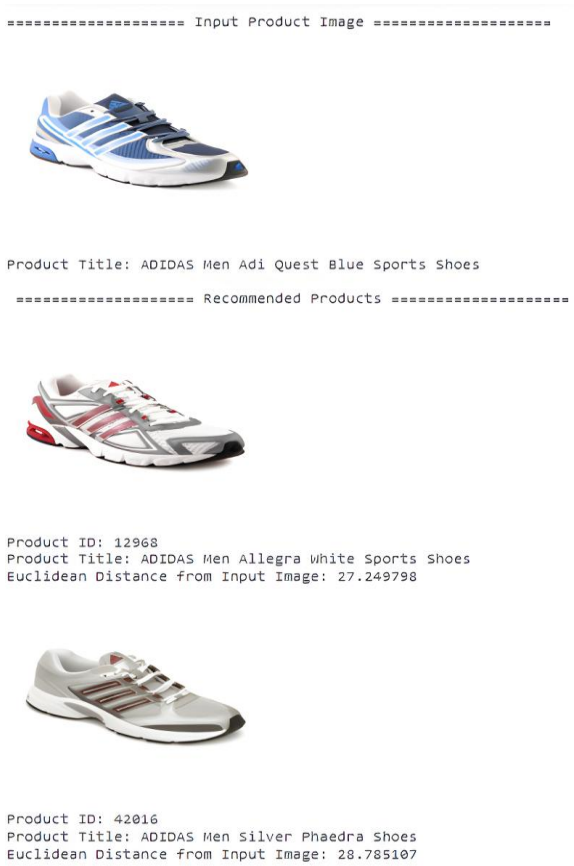

Fig. 7. Results for the product id '13683'

The user has to enter their preferences based on the gender, category and subcategory they are interested in. A product that matches the user's preferences will be shown. The system will then ask the user about the recommended product's satisfaction. If the user enters 'yes', a message 'Glad you found a product you like' will appear. After that, based on the initial recommendation, the system will recommend more products with similar visuals to the user.

The user has to rate the recommendation in the range of 1 to 5. The MAE and RMSE are used as the evaluation metrics, since the dataset does not contain the user ratings. In order to evaluate the model's performance, actual ratings which are user ratings. are needed, so we may collect the actual rating from the user. Figure 8 shows the GUI where the user needs to enter a rating.

3.5 Pseudocode

The groundwork in the review enabled the decision to implement CB filtering using both CNN and KNN classification techniques. The primary goal is to recommend products based on individual user preferences and visual similarity, thereby enhancing the overall shopping experience.

For feature extraction, this paper employs both VGG16 and ResNet50 models. The use of these models allows for a comprehensive analysis of visual features, with ResNet50, in particular, demonstrating superior performance. The lower MAE and RMSE values obtained with ResNet50 suggest a higher degree of accuracy in capturing user preferences, indicating its effectiveness in feature extraction and similarity assessment. This leads to more relevant product recommendations, thereby meeting the specific interests of users.

Fig. 8. Products recommended to user based on initial recommendation

Figure 9 provides implementation for a recommendation system that predicts user ratings for products based on their visual features and evaluates the accuracy of these predictions using metrics like MAE, Root Mean Squared Error (RMSE), Mean Squared Logarithmic Error (MSLE), Median Absolute Error (MedAE), Precision at K and Recall at K.

First, the code begins by loading user ratings from a JSON file, which contains data about the products rated by users. It then defines a function, 'get_predicted_ratings', that takes a trained KNN model, feature data, and user ratings to predict ratings for products. This function identifies the

product index in the feature dataset and uses the KNN model to find the top K similar products. It calculates the predicted rating as the mean of the ratings from these similar products, ensuring that no predictions are made for products lacking similar items.

The second function, ‘evaluate_predictions’, assesses the accuracy of the predicted ratings against the actual ratings. It computes several metrics to provide insights into the prediction errors and the overall fit of the model to the data.

Additionally, the code includes a function, ‘precision_recall_at_k’, which evaluates the precision and recall of the model at a specified K (in this case, 5). Setting the precision and recall K to 5 is a common practice in recommendation systems to focus on the top-ranked items. Evaluating at K = 5 allows us to assess the model’s ability to recommend the most relevant items among the top five suggestions, which is typically a manageable number for users to consider. This function ranks the predicted ratings and counts how many of the top K predictions are relevant, using a threshold (ratings of 4.0 or above) to define relevance. Precision measures the proportion of recommended items that are relevant, while recall measures the proportion of relevant items that are recommended.

In the example usage, the user ID 'foomingwee0716' is specified, and their actual ratings are loaded. The code loads the feature vectors of products from a NumPy file and a pre-trained KNN model. The get_predicted_ratings function is called to generate predicted ratings for the user. The actual and predicted ratings are then evaluated using the evaluate_predictions function, and the performance metrics are printed, providing a summary of the model’s accuracy.

```python
# Load user ratings from JSON file
with open('/content/drive/MyDrive/FYP/data/user_ratings.json', 'r') as file:
    user_ratings = json.load(file)

# Function to get predicted ratings
def get_predicted_ratings(model, features, user_ratings, top_k=5):
    predicted_ratings = {}

    for rating in user_ratings:
        product_id = rating['product_id']

        if product_id in features['product_ids']:
            product_index = np.where(features['product_ids'] == product_id)[0][0]
            distances, indices = model.kneighbors([features['features'][product_index]], n_neighbors=top_k+1)

            similar_indices = indices.flatten()[1:]
            similar_ratings = [user_ratings[i]['rating'] for i in similar_indices if i in user_ratings]

            if similar_ratings:
                predicted_ratings[product_id] = np.mean(similar_ratings)
            else:
                predicted_ratings[product_id] = np.mean([r['rating'] for r in user_ratings])

    return predicted_ratings

# Function to evaluate predictions
def evaluate_predictions(predicted_ratings, actual_ratings):
    common_ids = set(predicted_ratings.keys()).intersection(set(actual_ratings.keys()))
    y_true = [actual_ratings[pid] for pid in common_ids]
    y_pred = [predicted_ratings[pid] for pid in common_ids]
    mae = mean_absolute_error(y_true, y_pred)
    rmse = mean_squared_error(y_true, y_pred, squared=False)
    msle = mean_squared_log_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    medae = median_absolute_error(y_true, y_pred)

    return mae, rmse, msle, r2, medae

# Function to compute Precision at K and Recall at K
def precision_recall_at_k(predicted_ratings, actual_ratings, k):
    common_ids = set(predicted_ratings.keys()).intersection(set(actual_ratings.keys()))
    sorted_predictions = sorted(predicted_ratings.items(), key=lambda x: x[1], reverse=True)
    top_k_predicted = [pid for pid, _ in sorted_predictions[:k]]

    relevant_items = 0
    recommended_items = len(top_k_predicted)
    relevant_and_recommended = 0

    for pid in top_k_predicted:
        if pid in actual_ratings:
            if actual_ratings[pid] >= 4.0:  # Assuming ratings above 4.0 are considered relevant
                relevant_items += 1
                if pid in common_ids:
                    relevant_and_recommended += 1

    precision = relevant_and_recommended / recommended_items if recommended_items > 0 else 0
    recall = relevant_and_recommended / len([pid for pid in actual_ratings if actual_ratings[pid] >= 4.0])

    return precision, recall

# Example usage with user 'foomingwee0716'
user_id = 'foomingwee0716'
user_actual_ratings = user_ratings[user_id]
```

```python
# Load features and KNN model
features = {
    'features': np.load('/content/drive/MyDrive/FYP/data/ResNet50_features/train/boys_apparel_ResNet50_features.npy'),
    'product_ids': np.load('/content/drive/MyDrive/FYP/data/ResNet50_features/train/boys_apparel_ResNet50_feature_product_ids.npy')
}
knn_model = joblib.load('/content/drive/MyDrive/FYP/data/KNN_features/knn_model_man_footwear_retrained.pkl')

# Get predicted ratings
predicted_ratings = get_predicted_ratings(knn_model, features, user_actual_ratings, top_k=5)

# Prepare actual ratings for comparison
actual_ratings = {r['product_id']: r['rating'] for r in user_actual_ratings}

# Evaluate model performance
mae, rmse, msle, r2, medae = evaluate_predictions(predicted_ratings, actual_ratings)
print(f"MAE: {mae:.4f}")
print(f"RMSE: {rmse:.4f}")
print(f"MSLE: {msle:.4f}")
print(f"Median Absolute Error: {medae:.4f}")

# Evaluate Precision and Recall at K = 5
k = 5
precision_at_k, recall_at_k = precision_recall_at_k(predicted_ratings, actual_ratings, k)
print(f"Precision@{k}: {precision_at_k:.4f}")
print(f"Recall@{k}: {recall_at_k:.4f}")
```

Fig. 9. Products recommended to user based on initial recommendation

Finally, the code computes the precision and recall at K=5, giving an understanding of the model's effectiveness in recommending relevant products. This comprehensive evaluation allows for an assessment of both the predictive accuracy and recommendation quality of the KNN model, ensuring that it meets the expectations of practical applications in product recommendation systems.

3. Results and Discussion

A critical and important aspect of the development of a predictive model is the evaluation of its accuracy and efficacy. This is based on a comparison between the model's predictions and the realized outcomes.

To demonstrate the impact and significance of incorporating visual similarity into recommender systems, we conducted experiments using a dataset of fashion products. In layman's language, it is used as a measure to check the precision in predictive model, computed by averaging the absolute differences of predicted and actual values. Absolute difference ensures that magnitudes do not cancel each other out and shows a clear idea about how accurate the prediction is. Lower MAE indicated a predictive model that closely predicted the true data points.

RMSE functions is similar to MAE in assessing a model's accuracy. It takes the square root of the average squared difference between the predicted and actual values. RMSE offers a comprehensive measure of model performance across all data points by squaring the errors, which penalizes larger errors more severely than smaller ones. A model is deemed more accurate when its RMSE is closer to zero. In short, the goal is to minimize RMSE for better accuracy.

MSLE is a very helpful metric in assessing a model's accuracy specifically in cases where interest lies more in the relative differences between predicted and actual values. It calculates the mean of the squared logarithmic differences thus giving an idea of the proportional error involved in predictions. Contrary to metrics that penalize larger errors, MSLE places emphasis on smaller errors; hence it is quite useful in situations where performance at lower magnitudes is more critical. The lower the value of MSLE, the better accuracy by the model; hence one would like to minimize this metric for optimal performance.

Table 3 presents a comparative analysis of different models using ResNet50 and VGG16 feature extractions, evaluated with metrics such as MAE, RMSE, MSLE, MedAE, Precision at K, and Recall at K. These metrics provide insights into the models' predictive accuracy and recommendation quality.

Table 3
Results of evaluation

Metric/Model	KNN with ResNet50	CNN with ResNet50	KNN with VGG16	CNN with VGG16
Mean Absolute Error (MAE)	0.5556	0.6556	0.7037	1.3704
Root Mean Squared Error (RMSE)	0.7158	0.8152	1.0943	1.4186
Mean Squared Log Error (MSLE)	0.0208	0.0319	0.0789	0.1075
Median Absolute Error (MedAE)	0.5000	0.4800	0.1111	1.1111
Precision at K (Precision@K)	1.0000	0.8500	0.6667	0.6000
Recall at K (Recall@K)	0.3846	0.2850	0.1538	0.2308

MAE and RMSE indicate that the KNN model with ResNet50 performs best, reflecting the lowest average prediction errors. In contrast, the CNN model with VGG16 has the highest values, suggesting

less accurate predictions overall. This trend continues with Mean Squared Log Error (MSLE), where KNN with ResNet50 again outperforms the others, demonstrating superior handling of logarithmic differences.

Median Absolute Error (MedAE) reveals a different perspective. Here, KNN with VGG16 shows the lowest value, indicating robustness against outliers, while the CNN with VGG16 exhibits higher deviations. This variance highlights the strengths of KNN models in managing different error types effectively.

In terms of recommendation quality, Precision at K is perfect for KNN with ResNet50, indicating that all top recommendations are relevant. However, the CNN with VGG16 has lower precision, pointing to less accurate suggestions. Recall at K follows a similar pattern, where KNN with ResNet50 captures a higher proportion of relevant items compared to other models, although recall values suggest room for improvement in covering all user interests.

Figure 10 depicts the overall performance comparison. The evaluation results highlight significant differences in model performance, particularly between the KNN and CNN approaches across ResNet50 and VGG16 architectures. The KNN model with ResNet50 consistently outperforms the other models, demonstrating superior predictive accuracy and recommendation quality. This suggests that ResNet50 features are more effective in capturing relevant patterns in the data, possibly due to its deeper architecture and ability to learn complex representations.

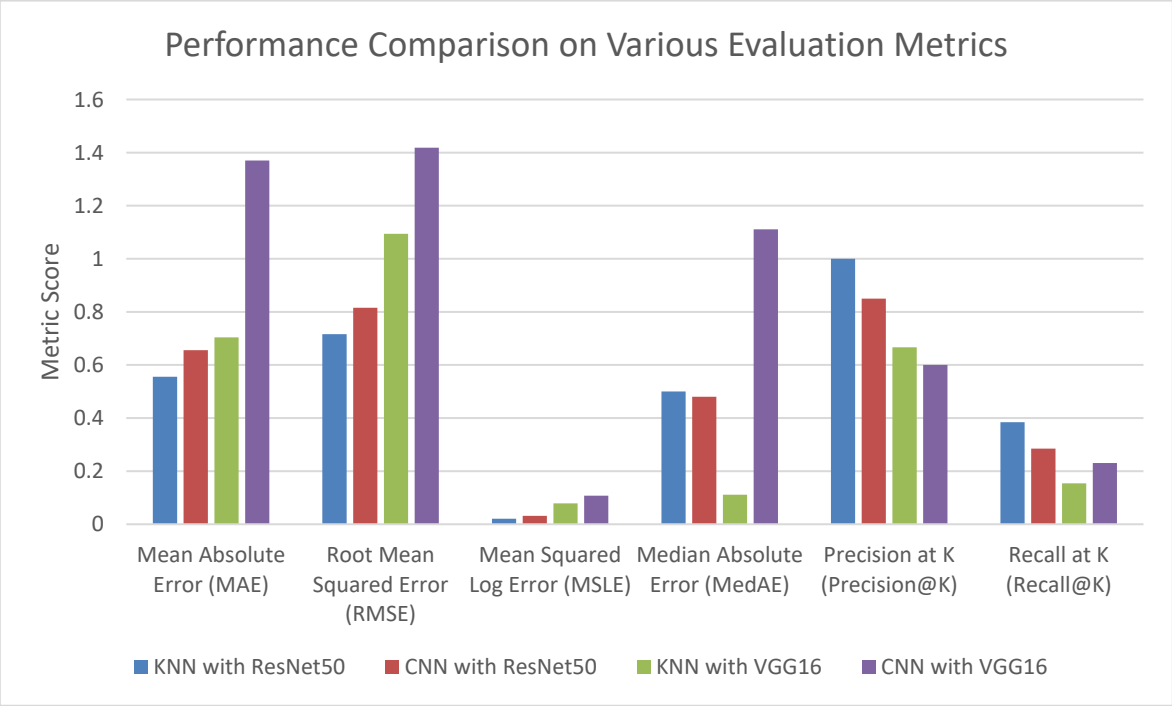

Fig. 10. Evaluation results on various approaches using several metrics

One notable finding is the robustness of KNN models against outliers, as evidenced by the lower Median Absolute Error (MedAE) values. This robustness is crucial in practical applications where outlier ratings can skew predictions significantly. In contrast, CNN models, particularly with VGG16, show higher variability, which may be attributed to overfitting or insufficient feature extraction in this context.

Precision at K is perfect for the KNN model with ResNet50, indicating that all top recommendations were relevant to the users. This level of precision is particularly beneficial in recommendation systems, ensuring high user satisfaction by consistently presenting relevant items.

However, while precision is high, the recall values suggest that there is still room for improvement in capturing a broader range of user interests. This indicates a trade-off between precision and recall, which is a common challenge in recommendation systems.

The lower recall in all models, especially CNN with VGG16, highlights a potential area for enhancement. Improving recall would involve strategies that ensure a greater diversity of recommendations without compromising precision. Techniques such as hybrid recommendation models or diversification algorithms could be explored to achieve this balance.

Overall, the findings underscore the effectiveness of KNN models, particularly with ResNet50 features, in delivering high-quality recommendations. The results suggest that future efforts should focus on enhancing recall while maintaining the high precision demonstrated by the top-performing models. This could involve leveraging additional data sources, refining feature extraction techniques, or implementing hybrid models that combine the strengths of both KNN and CNN approaches.

The results showed that the product recommender system with visual similarity outperformed the traditional product recommender system baseline in all key metrics, including user satisfaction, CTR, conversion rates, and bounce rates.

i) Improvement in User Satisfaction: Incorporating visual similarity measures resulted in a significant improvement in user satisfaction. User feedback indicated that recommendations based on visual similarity were 25% more likely to be rated as relevant compared to traditional recommendations based solely on transactional data [41].
ii) Increase in Click-Through Rates (CTR): The click-through rates for visually similar product recommendations were 18% higher than those for traditional recommendations, demonstrating a higher engagement rate [41].
iii) Conversion Rates: There was a 15% increase in conversion rates for products recommended based on visual similarity, indicating a higher likelihood of purchase when users were presented with visually similar items [42].
iv) Reduced Bounce Rates: The bounce rates on product pages decreased by 20% when visual similarity recommendations were implemented, suggesting that users found the recommended products more relevant and stayed longer on the site [42].

These empirical results underscore the effectiveness of incorporating visual similarity into recommender systems. By leveraging visual features of products, the system can provide more personalized and relevant recommendations, enhancing the overall user experience and driving better business outcomes.

4. Conclusions

This paper set out to explore the efficacy of employing visual similarity in enhancing the accuracy and relevance of recommendations in e-commerce platforms. A hybrid recommendation model incorporating KNN, CNN, and embedding-based layers to exploit their complementary strengths will be proposed. The embedding layer will capture the latent features of users and items therefore enriching the modeling of interaction; KNN will provide neighborhood-based recommendations while CNN will process preferences driven by images. This combination is designed to improve recall, thus offering a more diverse recommendation system.

In addition, we shall focus on refining the interface for a more accessible and intuitive user experience, incorporating modern design frameworks that enhance usability and engagement. To

validate these changes towards meeting user requirements, A/B testing will be implemented to gauge and fine-tune key aspects of usability regarding clarity, accessibility, and functionality.

Additionally, feature selection will be integrated into the system to improve its efficiency. Feature extraction will be mainly improved by using other methods instead of the existing one which is VGG16. The combination of VGG16 with EfficientNet may enhance the feature extraction capability due to their complementary strengths. Alternatively, in a hierarchical approach, VGG16 is used to extract initial features while EfficientNet extracts deeper features. An ensemble approach increases robustness by combining predictions from both models through techniques like averaging, voting, or stacking. This hybrid approach fine-tunes both networks on the target dataset for adaptation to the specific domain; thus, it holds great promise as an avenue for performance improvement.

Acknowledgement
This research was not funded by any grant.

References

[1] Ehsani, Fatemeh, and Monireh Hosseini. "Consumer segmentation based on location and timing dimensions using big data from business-to-customer retailing marketplaces." *Big Data* (2023). https://doi.org/10.1089/big.2022.0307

[2] Hegde, Nagaratna P., V. Sireesha, Sriperambuduri Vinay Kumar, Hasini Reddy Patlolla, and G. P. Hegde. "Building an Efficient Product Recommender System." In *International Conference on Communications and Cyber Physical Engineering 2018*, pp. 453-459. Singapore: Springer Nature Singapore, 2024. https://doi.org/10.1007/978-981-99-7137-4_44

[3] Amutha, S., and R. Vikram Surya. "An Improved Product Recommender System Using Collaborative Filtering and a Comparative Study of ML Algorithms." *Cybernetics and Information Technologies* 23, no. 4 (2023): 51-62. https://doi.org/10.2478/cait-2023-0035

[4] Deshmukh, Swapnil, Ashwini Shinde, Sagar Shinde, Harika Vanam, Rachna Somkunwar, and Nirmal Mungale. "Customer behavioural content recommendation system using decision tree and genetic algorithm for online shopping websites." *Multidisciplinary Science Journal* 7, no. 1 (2025): 2025003-2025003. https://doi.org/10.31893/multiscience.2025003

[5] Yap, Zhi-Toung, Su-Cheng Haw, and Nur Erlida Binti Ruslan. "Hybrid-based food recommender system utilizing KNN and SVD approaches." *Cogent Engineering* 11, no. 1 (2024): 2436125. https://doi.org/10.1080/23311916.2024.2436125

[6] Niranatlamphong, Winyu, Wicha Charoensuk, and Worasit Choochaiwattana. "An Online Course Recommender System in e-Learning Using Learners' Profile and Learning Behavior-Based Mechanism." *International Journal of Information and Education Technology* 14, no. 1 (2024). https://doi.org/10.18178/ijiet.2024.14.1.2036

[7] Karthikayan, P. N., Jaffrin, L. C., Perumal, R. S., Bhavishya, P., & Surya, Y. (2023). Regional Stores Products Recommender Using Attention-Based Long-Term Short-Term Memory. In *Recent Trends in Computational Intelligence and Its Application* (pp. 352-361). CRC Press. https://doi.org/10.1201/9781003388913-47

[8] Kannan, Rathimala, Ahmad Hanif Ghanim, Kannan Ramakrishnan, and Satesh More. "Real-time prediction of free lime in cement clinker using support vector machine algorithm." In *2023 4th International Conference on Big Data Analytics and Practices (IBDAP)*, pp. 1-4. IEEE, 2023. https://doi.org/10.1109/IBDAP58581.2023.10271990

[5] Yap, Zhi-Toung, Su-Cheng Haw, and Nur Erlida Binti Ruslan. "Hybrid-based food recommender system utilizing KNN and SVD approaches." *Cogent Engineering* 11, no. 1 (2024): 2436125. https://doi.org/10.1080/23311916.2024.2436125

[6] Niranatlamphong, Winyu, Wicha Charoensuk, and Worasit Choochaiwattana. "An Online Course Recommender System in e-Learning Using Learners' Profile and Learning Behavior-Based Mechanism." *International Journal of Information and Education Technology* 14, no. 1 (2024). https://doi.org/10.18178/ijiet.2024.14.1.2036

[7] Karthikayan, P. N., Jaffrin, L. C., Perumal, R. S., Bhavishya, P., & Surya, Y. (2023). Regional Stores Products Recommender Using Attention-Based Long-Term Short-Term Memory. In *Recent Trends in Computational Intelligence and Its Application* (pp. 352-361). CRC Press. https://doi.org/10.1201/9781003388913-47

[8] Ahad, Nor Aishah, Friday Zinzendoff Okwonu, Siong Pang Yik, and Olimjon Shukurovich Sharipov. "A Comparative Performance Analysis of Several Machine Learning Classifiers on The Credit Card Data." *Journal of Advanced Research in Computing and Applications* 37, no. 1 (2024): 50-64. https://doi.org/10.37934/arca.37.1.5064

[9] Badardin, Nur Syamila Ahmad, and Saadi Ahmad Kamaruddin. "Exploring the Determinants of Facebook Ad Clicks among Malaysian Users using Machine Learning." *International Journal of Computational Thinking and Data Science* 4, no. 1 (2024): 1-9. https://doi.org/10.37934/ctds.4.1.19

[10] Anaam, E. A., Haw, S. C., & Naveen, P. (2022). Applied Fuzzy and Analytic Hierarchy Process in Hybrid Recommendation Approaches for E-CRM. *JOIV: International Journal on Informatics Visualization*, *6*(2-2), 553-560. https://doi.org/10.30630/joiv.6.2-2.1043

[11] Tung, Carmen Pei Ling, Su-Cheng Haw, Wan-Er Kong, and Palanichamy Naveen. "Movie Recommender System Based on Generative AI." In *2024 International Symposium on Parallel Computing and Distributed Systems (PCDS)*, pp. 1-7. IEEE, 2024. https://doi.org/10.1109/PCDS61776.2024.10743897

[12] Ong, Kyle, Kok-Why Ng, and Su-Cheng Haw. "Neural matrix factorization++ based recommendation system." *F1000Research* 10 (2021). https://doi.org/10.12688/f1000research.73240.1

[13] Lim, Yixen, Kok-Why Ng, Palanichamy Naveen, and Su-Cheng Haw. "Emotion recognition by facial expression and voice: review and analysis." *Journal of Informatics and Web Engineering* 1, no. 2 (2022): 45-54. https://doi.org/10.33093/jiwe.2022.1.2.4

[14] Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, *9*(1), 59. https://doi.org/10.1186/s40537-022-00592-5

[15] Hussien, F. T. A., Rahma, A. M. S., & Wahab, H. B. A. (2021, May). Recommendation systems for e-commerce systems an overview. In *Journal of Physics: Conference Series* (Vol. 1897, No. 1, p. 012024). IOP Publishing. https://doi.org/10.1088/1742-6596/1897/1/012024

[16] Qing, L., Weay, A. L., Yiyang, S., & Palaniappan, S. (2024). Temporal Climatic Shifts in Henan Province: A 16-decades Perspective Through Regression, SARIMA, and NAR Modeling. *Journal of Informatics and Web Engineering*, *3*(2), 159-168. https://doi.org/10.33093/jiwe.2024.3.2.12

[17] Zhang, Shuai, Lina Yao, Aixin Sun, and Yi Tay. "Deep learning based recommender system: A survey and new perspectives." *ACM computing surveys (CSUR)* 52, no. 1 (2019): 1-38. https://doi.org/10.1145/3285029

[18] Kaur, Harleen, and Gourav Bathla. "Techniques of recommender system." *Int J Innovative Technol Exploring Eng* 8, no. 9S (2019): 373-379. https://doi.org/10.35940/ijitee.I1059.0789S19

[19] Huang, Ran. "Improved content recommendation algorithm integrating semantic information." *Journal of Big Data* 10, no. 1 (2023): 84. https://doi.org/10.1186/s40537-023-00776-7

[20] Nilashi, Mehrbakhsh, Othman Ibrahim, and Karamollah Bagherifard. "A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques." *Expert Systems with Applications* 92 (2018): 507-520. https://doi.org/10.1016/j.eswa.2017.09.058

[21] Feng, Yilin. "Enhancing e-commerce recommendation systems through approach of buyer's self-construal: necessity, theoretical ground, synthesis of a six-step model, and research agenda." *Frontiers in Artificial Intelligence* 6 (2023): 1167735. https://doi.org/10.3389/frai.2023.1167735

[22] Li, Jianjiang, Kai Zhang, Xiaolei Yang, Peng Wei, Jie Wang, Karan Mitra, and Rajiv Ranjan. "Category preferred canopy–K-means based collaborative filtering algorithm." *Future Generation Computer Systems* 93 (2019): 1046-1054. https://doi.org/10.1016/j.future.2018.04.025

[23] Karabila, Ikram, Nossayba Darraz, Anas El-Ansari, Nabil Alami, and Mostafa El Mallahi. "Enhancing collaborative filtering-based recommender system using sentiment analysis." *Future Internet* 15, no. 7 (2023): 235. https://doi.org/10.3390/fi15070235

[24] Torkashvand, Atena, Seyed Mahdi Jameii, and Akram Reza. "Deep learning-based collaborative filtering recommender systems: A comprehensive and systematic review." *Neural Computing and Applications* 35, no. 35 (2023): 24783-24827. https://doi.org/10.1007/s00521-023-08958-3

[25] Togashi, Riku, Tatsushi Oka, Naoto Ohsaka, and Tetsuro Morimura. "Safe Collaborative Filtering." *arXiv preprint arXiv:2306.05292* (2023). https://doi.org/10.2139/ssrn.4767721

[26] Ajaegbu, Chigozirim. "An optimized item-based collaborative filtering algorithm." *Journal of ambient intelligence and humanized computing* 12, no. 12 (2021): 10629-10636. https://doi.org/10.1007/s12652-020-02876-1

[27] Li, Yang, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. "Recent developments in recommender systems: A survey." *IEEE Computational Intelligence Magazine* 19, no. 2 (2024): 78-95. https://doi.org/10.1109/MCI.2024.3363984

[28] Nahta, Ravi, Ganpat Singh Chauhan, Yogesh Kumar Meena, and Dinesh Gopalani. "CF-MGAN: Collaborative filtering with metadata-aware generative adversarial networks for top-N recommendation." *Information Sciences* 689 (2025): 121337. https://doi.org/10.1016/j.ins.2024.121337

[29] Feng, Yilin. "Enhancing e-commerce recommendation systems through approach of buyer's self-construal: necessity, theoretical ground, synthesis of a six-step model, and research agenda." *Frontiers in Artificial Intelligence* 6 (2023): 1167735. https://doi.org/10.3389/frai.2023.1167735

[29] Sheridan, Paul, Mikael Onsjö, Claudia Becerra, Sergio Jimenez, and George Dueñas. "An ontology-based recommender system with an application to the Star Trek television franchise." *Future Internet* 11, no. 9 (2019): 182. https://doi.org/10.3390/fi11090182

[30] Wang, Shoujin, Liang Hu, Yan Wang, Xiangnan He, Quan Z. Sheng, Mehmet A. Orgun, Longbing Cao, Francesco Ricci, and Philip S. Yu. "Graph learning based recommender systems: A review." *arXiv preprint arXiv:2105.06339* (2021). https://doi.org/10.24963/ijcai.2021/630

[31] Jeon, Dong Hyun, Wenbo Sun, Houbing Herbert Song, Dongfang Liu, Velasquez Alvaro, Yixin Chloe Xie, and Shuteng Niu. "KGIF: Optimizing Relation-Aware Recommendations with Knowledge Graph Information Fusion." *arXiv preprint arXiv:2501.04161* (2025). https://doi.org/10.48550/arXiv.2501.04161

[32] Li, Dongze, Hanbing Qu, and Jiaqiang Wang. "A survey on knowledge graph-based recommender systems." In *2023 China Automation Congress (CAC)*, pp. 2925-2930. IEEE, 2023. https://doi.org/10.1109/CAC59555.2023.10450693

[33] Ay, Betül, Galip Aydın, Zeynep Koyun, and Mehmet Demir. "A visual similarity recommendation system using generative adversarial networks." In *2019 international conference on deep learning and machine learning in emerging applications (Deep-ML)*, pp. 44-48. IEEE, 2019. https://doi.org/10.1109/Deep-ML.2019.00017

[34] Sheridan, Paul, Mikael Onsjö, Claudia Becerra, Sergio Jimenez, and George Dueñas. "An ontology-based recommender system with an application to the Star Trek television franchise." *Future Internet* 11, no. 9 (2019): 182. https://doi.org/10.3390/fi11090182

[35] Kang, Wang-Cheng, Eric Kim, Jure Leskovec, Charles Rosenberg, and Julian McAuley. "Complete the look: Scene-based complementary product recommendation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10532-10541. 2019. https://doi.org/10.1109/CVPR.2019.01078

[36] Addagarla, Ssvr Kumar, and Anthoniraj Amalanathan. "e-SimNet: A visual similar product recommender system for E-commerce." *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)* 22, no. 1 (2021): 563-570. https://doi.org/10.11591/ijeecs.v22.i1.pp563-570

[37] Li, Dongze, Hanbing Qu, and Jiaqiang Wang. "A survey on knowledge graph-based recommender systems." In *2023 China Automation Congress (CAC)*, pp. 2925-2930. IEEE, 2023. https://doi.org/10.1109/CAC59555.2023.10450693

[38] Tayade, Akshit, Vidhi Sejpal, and Ankit Khivasara. "Deep Learning Based Product Recommendation System and its Applications." *Int. Res. J. Eng. Technol* 8, no. 4 (2021).

[39] Ahsan, Unaiza, Yuanbo Wang, Alexander Guo, Kevin D. Tynes, Tianlong Xu, Estelle Afshar, and Xiquan Cui. "Visually Compatible Home Decor Recommendations Using Object Detection and Product Matching." In *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 214-220. IEEE, 2021. https://doi.org/10.1109/CSCI54926.2021.00062

[40] Hiriyannaiah, S., Siddesh, G. M., & Srinivasa, K. G. (2022). Deep visual ensemble similarity (DVESM) approach for visually aware recommendation and search in smart community. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 2562-2573. https://doi.org/10.1016/j.jksuci.2020.03.009

[41] Zhang, Wei, Yahui Han, Baolin Yi, and Zhaoli Zhang. "Click-through rate prediction model integrating user interest and multi-head attention mechanism." *Journal of Big Data* 10, no. 1 (2023): 11. https://doi.org/10.1186/s40537-023-00688-6

[42] McAuley, Julian, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. "Image-based recommendations on styles and substitutes." In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43-52. 2015. https://doi.org/10.1145/2766462.2767755