# Recommendation system of an image using vector search and embeddings

[a] Abinayasree R [b] Mahalakshmi E [c] Kavinesh R C [d] Naveenprasath S
[a,b,c,d] Deparment of computer Application, Bharathiar university - 2024

**Abstract:**

Image-based recommendation systems have become increasingly important in domains like e-commerce, fashion, and content discovery, where visual similarity plays a significant role in decision-making. In this paper, we present a methodology for building a recommendation system using Convolutional Neural Networks (CNNs) for image embeddings and vector search for similarity retrieval. We utilize pre-trained deep learning models to generate fixed-size embeddings from images, followed by indexing these embeddings using a scalable nearest neighbor search algorithm, such as FAISS. The resulting system allows users to retrieve visually similar images by inputting a query image, making it suitable for applications in image search, product recommendations, and visual content curation. Our approach leverages CNNs' ability to capture hierarchical image features and vector search's efficiency in handling large-scale datasets, providing a robust solution for real-time image-based recommendation systems

**Keywords:** Recommendation system, Image embeddings, Vector search, Deep learning.

## INTRODUCTION:

The rapid growth of digital content and e-commerce platforms, image-based recommendation systems have become essential in providing personalized, visually - driven suggestions to users. In traditional recommendation systems, textual andmetadata-driven features often dominate the recommendation process. However, in applications such as fashion, art, and product discovery, the visual similarity of items plays acritical role in user preferences.

Consequently, leveraging visualinformation from images is paramount for generating meaningful recommendations. Convolutional Neural Networks (CNNs), which have demonstrated outstanding performance in various image recognition tasks, are well-suited for extracting rich visual features from images.

By using a CNN, images can be transformed into dense, low-dimensional vectors (embeddings) that capture the most relevant visual features. These embeddings canthen be compared to find similar images, making CNNs a natural fit for building image- based recommendation systems. In this paper, we propose a method that combines CNN- based image embeddings with vector search algorithms, specifically utilizing FAISS (Facebook AI Similarity Search), to create an efficient image-based recommendation system. This approach allows for real-time image retrieval based on visual similarity, making it scalable for large datasets in practicalapplications.

## LITERATURE REVIEW:

Paper [1] focuses on learning visual similarity for product design using convolutional neural networks (CNNs). However, it considers cases where multiple products are contained in a single image, requiring the system to identify and handle multiple objects simultaneously. In our project, we assume that the user is searchingfor a specific product, and thus, each image contains only one product. Our approach

simplifies the scenario by concentrating on the similarity within single-product images, enhancing the recommendation accuracy by avoiding the complexities associated with multiple products in a single image. Paper [2] focuses on creating a personalized fashion recommender system using image-based neural networks. Unlike conventional systems that rely on user's previous purchase history, this approach utilizes an input image of a product provided by the user to generate recommendations. The system employs Convolutional Neural Networks (CNN) trained on the DeepFashion dataset and uses a nearest neighbor-backed method to identify similar products based on visual features. In contrast, our project aims to enhance recommendation accuracy by integrating additional data features and using a hybrid recommendation approach that combines both visual and textual information from multiple sources. Paper [4] proposes a content-based clothingrecommender system using deep neural networks (DNN). Unlike prior works that rely on manually extracting features or user face data for gender identification, this system automates feature extraction directly from product images and performs gender recognition based on clothing style rather than facial data. The system eliminates the need for manual labeling and resolves the cold start problem for new items, focusing on providing novel, relevant, and unexpected recommendations tailored to users. This approach is more efficient than methods solelyusing face-based gender identification. Paper
[6] proposes an enhanced content-based fashion recommendation system using a deep ensemble classifier combined with transfer learning techniques. Unlike previous models that rely on single pre-trained models, this approach integrates multiple models, including MobileNet, DenseNet, Xception, and two versions of VGG, to improve prediction accuracy. The deep ensemble classifier leverages the probabilities from these models to make more robust and reliable recommendations. In our project, we focus on simplifying the recommendation process by using a streamlined neural network architecture specifically tailored to classify fashion images and recommend similar products, prioritizing efficiency and speed without compromising accuracy. Paper [7] develops a fashion retrievaland recommendation model based on deep learning techniques. It introduces a Sketch- Product retrieval model that allows users to

search for fashion products by sketching them, which is then upsampled using GAN technology to produce image-based results. Thepaper also presents a vector-based fashion recommendation model that uses implicit profiling, unlike traditional profiling methods that rely on user surveys. This work differs fromother systems, such as Amazon's item-to-item collaborative filtering, by focusing on sketch- based retrieval and leveraging deep learning for feature extraction. Paper [9] focuses on developing personalized fashion recommendation systems using collaborative filtering techniques, which integrate both user preferences and fashion item attributes. However, it primarily addresses fashion coordination without considering the variability in user preferences across different contexts, such as weather or event type. In contrast, our project emphasizes providing context-aware recommendations by incorporating real-time factors like current weather conditions and user-specific events, ensuring that the suggested fashion items align more closely withthe user's situational needs. Paper [10] by Tuinhof et al. (2018) focuses on image-based fashion product recommendation using CNNs. The proposed system is a two-stage process thatfirst uses CNNs for image classification and then applies a k-NN algorithm for similarity- based ranking. However, it specifically addresses the challenge of recommending visually similar fashion products based on inputimages, assuming that users are looking foritems of similar style. In contrast, our project assumes that users may seek recommendationsfor a single product type at a time, and the focusis on enhancing product categorization and texture matching through tailored CNN architectures. Paper [11] explores the use of Variational Autoencoders (VAEs) for fashion product generation, similar image retrieval, and cross-category recommendations. This approach allows users to interactively generateproducts with desired attributes and retrievesimilar styles within the same category. However, it primarily focuses on clustering andretrieving images using latent codes without considering personalized user preferences or contextual factors such as occasion or location.In our project, we aim to address these gaps by incorporating user-specific data and context-aware filtering techniques, enhancing the personalization and relevance of the recommendations beyond basic feature similarity. Paper [12] by Lin et al. (2019) introduces a neural co-supervision learning framework, FARM, aimed at improving outfit recommendation by incorporating fashion generation into the process. The paper highlights two main challenges in fashion recommendation: visual understanding and visual matching. However, unlike our project, which focuses on improving product classification and matching based on individual fashion items, their approach integrates a generative process to enhance the quality of recommendations, particularly in complex matching scenarios between tops and bottoms

.Paper [15] by Duan and Saga (2019) focuses on developing an apparel goods recommender system using image shape features extracted by CNN. Their system incorporates image shape information into a probabilistic matrix factorization model to enhance recommendation accuracy. Unlike our project, which primarily uses textual and contextual information, their work highlights the importance of visual shape features for apparel, which is more suited to fashion-related recommendations. Paper [16] also explores the integration of visual signals into recommender systems, specifically using the Visual Bayesian Personalized Ranking (VBPR) framework. However, their focus is primarily on incorporating product images into collaborative filtering models, without handling completely unseen products during training. In contrast, our project extends the Skip-Gram architecture to include a decoder that reconstructs image features, which allows us to make recommendations even for cold-start products without historical data. Paper [17] also explores visual-based recommendations, similar to our project. However, it relies on pre-trainedconvolutional neural networks (CNNs) to extract image features, which are then integrated into a collaborative filtering framework (VBPR). In contrast, our projectintroduces a more dynamic approach by embedding both users and products in acommon latent space using the Visually Aware Skip-Gram (VASG) framework. This enables us to not only capture product image features but also recommend cold-start products through image feature mappings. Paper [19] also focuses on outfit recommendation and leverages a neural framework to improve recommendation quality. However, it integrates a co-supervision learning model, FARM, which combines both recommendation and generation tasks to enhance aesthetic feature extraction. Unlike our project, which centers on product matching using visual and textual descriptions alone, their approach incorporates a layer-to- layer matching mechanism to fuse aesthetic features more effectively.

## WORKFLOW:

**1. Data Collection & Preprocessing:**

- Collect Image Data: Gather a dataset of images. You can use publicly available datasets such as CIFAR-10, ImageNet, or your custom dataset.

- Preprocess Images: Resize, normalize, or augment the images if necessary to preparethem for model input.

  - Example: Resize all images to 224x224 pixels if using a pretrained model like ResNet.

**2. Image Embedding Generation:**

- Choose a Pretrained Model:

Use a pretrained deep learning model to generate embeddings from images. Common choices include:

  - ResNet, VGG, EfficientNet: These modelscan be used to extract high-dimensional embeddings from images.

  - Example: Load a pretrained ResNet50 model and remove the final classification layerto use the model as a feature extractor.

- Generate Embeddings: Feed the images intothe model to generate feature vectors (embeddings) for each image.

  - Example: For an image input, ResNet50 will generate a 2048-dimensional embedding that represents the features of the image.

**3. Indexing Embeddings for Vector Search:**

- Choose a Vector Search Library: Use a vector search library to index the generated embeddings, allowing for fast retrieval of similar images. Popular libraries include:

- FAISS (Facebook AI Similarity Search): Efficient for high-dimensional vector search.

- Annoy (Approximate Nearest Neighbors Oh Yeah):Fast and scalable for approximate nearest neighbor search.

- Milvus: A more robust open-source platform for vector similarity search.

- Index Embeddings: After generating embeddings, store them in a vector index for similarity search.

- Example: FAISS can be used to index the embeddings and perform similarity searchbased on distance metrics like cosine similarity or Euclidean distance.

**4. Defining Similarity Metric:**

- Cosine Similarity or Euclidean Distance: Choose a similarity metric that will be used to compare the embeddings. These metrics help find images whose embeddings are closest to the query image.

- Cosine Similarity: Measures the cosine of the angle between two vectors, useful for high-dimensional embeddings.

- Euclidean Distance: Measures the straight-line distance between two vectors.

- Example: Cosine similarity works well for images represented by embeddings, as it accounts for the angle between embeddings in high-dimensional space.

**5. Querying the Index for Recommendations**

- Input a Query Image: The user uploads or selects an image for which they want recommendations.

- Generate Embedding for Query Image: Pass the query image through the same feature extraction model to obtain its embedding.

- Example: If the query image is 224x224 pixels, it will be transformed into a 2048-dimensional vector.

- Retrieve Similar Images: Perform a vector search using the query image's embedding to find the nearest neighbors in the vector index.

- Example: Use FAISS or Annoy to retrievethe top-N most similar images based on cosine similarity.

**6. Return and Display Recommendations:**

- Fetch Top-N Results: Based on the similarity scores from the vector search, retrieve the top-N most similar images.

- Example : If the query image is of a dog, the system might return images of other dogs oranimals with similar features.

- Display Results: Show the recommended images (e.g., in a web app or mobile app),allowing users to explore visually similar images.

**7. Evaluate and Optimize:**

- Evaluate Model: Assess the quality of recommendations using metrics like:

- Precision at K: Measures how many of thetop-K recommended images are relevant.

- Mean Average Precision (MAP): Measuresthe average precision of ranked results.

- Optimize Embedding Quality: Fine-tune the feature extraction model or embeddings for better recommendations.

- Example: You could fine-tune the model onyour specific image dataset to improve the quality of embeddings.

Dimensionality Reduction (Optional): Use techniques like *PCA* or *t-SNE* to reduce the dimensionality of embeddings if needed forfaster search.

## Methodology:

The proposed image-basedrecommendation system consists of several keycomponents: image preprocessing, CNN-based feature extraction, embedding indexing, and similarity retrieval. The following steps outlinethe methodology in detail:

1. **DataCollection and Preprocessing:**

The system starts by gatheringa dataset of images that are relevant to the application domain (e.g., fashion, artwork, products). Each imageundergoes a preprocessing pipeline to standardize input dimensions and normalize pixel values. The images are resized to the input resolution required by the pre-trained CNN model (e.g., 224x224 for ResNet).

A normalization step is applied to align the pixel intensity distributions with the pre-trained model's expectations.

**Preprocessing Steps**:

1) Resize: Scale images to 256x256.
2) Center Crop: Crop the image to 224x224.
3) Normalization: Adjust pixel values to a normalized range based on the model requirements.

**2. CNN Model Selection and Embedding Extraction:**

We use a pre-trained CNN model, such as ResNet50, which is well-known for its strong feature extraction capabilities. The model is modified by removing the final classification layer, leaving the penultimate layer as the output, which provides a fixed-size vector representation (embedding) for each image. This vector encodes the key visual features of the image in a dense format.The embedding generation process is applied to each image in the dataset, converting it into a high-dimensional vector (e.g., 2048-dimensional for ResNet50). These embeddings serve as the basis for similarity comparisons.

**Key Steps**:

Load the pre-trained CNN model (ResNet50).Remove the last classification layer to obtain the embeddings.Extract a fixed-size embedding vector for each image.

**3. Indexing Embeddings for Vector Search:**

Once the image embeddings are generated, they are indexed using a vector search algorithm to allow efficient nearest neighbor search. We employ FAISS (Facebook AI Similarity Search), a popular tool for large-scale similarity searches. FAISS is designed to efficiently handle high- dimensional vectors and large datasets by providing both exact and approximate nearest neighbor search options.The indexed embeddings arestored in a searchable format, allowingfor rapid retrieval of similar images when a query image is provided.Steps for Indexing:Use FAISS to create an index for the image embeddings.Add all embeddings to the index for efficient querying.

**4. Query and Recommendation** :

For each query image, the system preprocesses the image and passes it through the CNN model to generate its embedding. This query embedding is then compared to the pre-indexed embeddings using FAISS to perform a nearest neighbor search. The system retrieves the top-k most similarimages based on vector distances, typically using Euclidean distance (L2 norm) or cosine similarity. The resulting images, which are visually similar to the query image, are displayed as recommendations.

**Steps for Query and Retrieval:**

Generate the embedding for the query image using the same CNN model. Perform a nearest neighbor search using FAISS to retrieve similar images. Display the recommended images to the user.

**5. System Optimization**:

To enhance the system'sperformance and scalability, several optimization techniques can be employed:

**Dimensionality Reduction:** Apply techniques such as Principal Component Analysis (PCA) to reduce the size of the embedding vectors, minimizing memory usage and search time without significantly impacting similarity accuracy.

**Fine-tuning the CNN**: In cases where domain-specific features are important, fine-tuning the pre-trained CNN modelon a domain-specific dataset can improve the quality of the embeddings. **Approximate Nearest Neighbor Search:** FAISS supports approximate nearest neighbor search, which reduces computation time for large-scale datasets with minimal trade- off in accuracy.

## CONCLUSION:

Building a recommendation system forimages using vector search andembeddings is an effective way to provide personalized and meaningful image suggestions. By leveraging powerful deep learning models like ResNet or VGG for feature extraction, and indexing the resulting embeddings using vector search libraries like FAISSor Annoy, you can efficiently retrieve similar images based on their semanticcontent.

This approach not only enhances search efficiency but also improves the quality of recommendations byfocusing on the visual and semantic similarities between images rather thanrelying on metadata or tags. The systemcan be fine-tuned and optimized based on specific use cases, making it adaptable to different domains such as e-commerce, social media, and visual search applications.

By following this workflow, you can build scalable, accurate, and high- performance image recommendation systems that deliver relevant content tousers, ultimately improving the user experience in image- based applications.

## REFERENCE:

1. Luyang Chen, Fan Yang, Heqing Yang, "Image-based ProductRecommendation System with Convolutional Neural Networks"

2. M Sridevi[1], N ManikyaArun[1], M Sheshikala[2] and E Sudarshan, "Personalized fashion recommender system withimage based neural networks"

3. Rohit Anand, Vaibhav Nirmal, Yaksh Chauhan, TusharSharma, "An Image-based Deep Learning Approach for Personalized Outfit Selection"

4. Narges Yarahmadi Gharaei, C. Dadkhah, Lorence Daryoush,"Content-based Clothing Recommender System using Deep Neural Network"

5. B. Suvarna, Sivadi Balakrishna, " An Efficient FashionRecommendation System using a Deep CNN Model"

6. Buradagunta Suvarna, Sivadi Balakrishna, "Enhanced content-based fashion recommendation system through deepensemble classifier with transfer learning"

7. Jaechoon Jo, Seolhwa Lee, Chanhee Lee, Dongyub Lee, Heuiseok Lim, "Development of Fashion Product Retrievaland Recommendations Model Based on Deep Learning"

8. Samit Chakraborty, Md. Saiful Hoque, Naimur Rahman Jeem, M. Biswas, Deepayan Bardhan, E. Lobaton, Fashion Recommendation Systems, Models and Methods: A Review"

9. Khushi Kandoi, R. N, Siddhant Gautam Singh, Ronak Bediya, Krishnanand Mishra, Sushil Kumar Singh , " Comparative Analysis of Deep Learning Models for FashionRecommendation in E-Commerce"

10. Hessel Tuinhof, C. Pirker, M. Haltmeier , " Image Based Fashion Product Recommendation with Deep Learning"

11. James-Andrew R. Sarmiento, "Exploiting Latent Codes: Interactive Fashion Product Generation, Similar Image Retrieval, and Cross-Category Recommendation using Variational Autoencoders"

12. Yujie Lin, Pengjie Ren, Zhumin Chen, Z. Ren, Jun Ma, M. de Rijke , "Improving Outfit Recommendation with Co-supervision of Fashion Generation"

13. Xu Chen, H. Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, H. Zha, "Personalized Fashion Recommendation with Visual Explanations based on Multimodal Attention Network: Towards Visually Explainable Recommendation"

14. Wenxin Yao, Kaoru Uchida, Jianpeng Zou, Xiaolin Zhong, "Image-Based Fashion Recommendation with Attention to Users' Interests"

15. R. Saga, Yufeng Duan, " Apparel Goods Recommender System Based on Image Shape Features Extracted by a CNN"

16. Qianqian Wu, Pengpeng Zhao, Zhiming Cui, "Visual and Textual Jointly Enhanced Interpretable Fashion Recommendation"

17. Parth Tiwari, Yash Jain, Shivansh Mundra, Jenny Harding, M. Tiwari, "Visually Aware Skip-Gram for Image Based Recommendations"

18. Yujie Lin,Pengjie Ren,Zhumin Chen,Maarten de Rijke,Jun Ma,Zhaochun Ren"Improving Outfit Recommendation with Co supervision of Fashion Generation".

19. Jingyuan Chen,Liqiang Nie,Hanwang Zhang,Xiangnan He,Tat-Seng Chua"A entive Collaborative Filtering: Multimedia Recommendationwith Item- and Component-Level Attention".

20. Shashank Rajput,Nikhil Mehta,Anima Singh,Raghunandan Keshavan,Trung Vu,Lukasz Heldt,Lichan Hong,Yi Tay,Vinh Q. Tran,Jonah Samost,Maciej Kula,Ed H. Chi,Maheswaran Sathiamoorthy"Recommender Systems with Generative Retrieval"