Conversational Fashion Outfit Generator

Narsaiah Putta
Dept. Computer Science Engineering
Vasavi College of Engineering
Hyderabad, Telangana, India
p.narsaiah@staff.vce.ac.in

Rahul Kasula
Dept. Computer Science Engineering
Vasavi College of Engineering
Hyderabad, Telangana, India
rahulkasula23@gmail.com

K. Vishal Reddy
Dept. Computer Science Engineering
Vasavi College of Engineering
Hyderabad, Telangana, India
kolipakavishal02@gmail.com

***Abstract* - In today's fast-paced digital world, to choose an apt outfit can be so overwhelming due to its huge options and its fashion trends. Customary e-commerce platforms rely on keyword-based searches, in addition to manual filtering, making the process time-consuming and inefficient, especially for users who are unsure about what they want. The Conversational Fashion Outfit Generator is one AI-powered fashion recommendation system designed for simple outfit discovery. It is unlike shopping platforms that are conventional. It enables the user to describe fashion needs in language that is natural, such as "I need a stylish pastel outfit for a summer wedding". Using Generative AI, CLIP embeddings, and multi-modal search, the system generally understands the query and retrieves the most relevant out-fits.**

Keywords - Semantic Search, Multimodal Embeddings, Fashion Recommendation System.

INTRODUCTION

The growth of e-commerce has profoundly redefined the way consumers find, consider, and buy fashion. Fashion is more accessible than ever due to online platforms that provide an immense range of apparel and accessories. But this has also resulted in decision fatigue and search inefficiency. Classic e-commerce search systems, which are mostly based on keyword matching and strict filters, usually fail when consumers try to express sophisticated preferences like "a stylish pastel outfit for a summer brunch" or "something dressy but not too tight for an evening party." These shortcomings are mostly caused by the absence of contextual knowledge and natural language understanding in traditional systems.

To address these issues, Conversational Fashion Outfit Generators are becoming smart tools that enable users to interact in a natural way with text-based inputs. They integrate innovations in natural language processing (NLP), computer vision, and vector search technologies for presenting customized and appropriate outfit suggestions. With the ability to comprehend both the semantic interpretation of inputs from users and the visual appeal of fashion items, they make it possible for interactions to be smooth and natural-like. These systems extend beyond mere product search; they know context, mood, occasion, and even abstract style signals, making for a richer and more personalized shopping experience. They also enable new voice-based shopping opportunities, chatbot support, and accessibility for users who do not know the precise product names but can tell what they want in natural language.

The development of e-commerce has fundamentally changed how people connect with fashion. Users sometimes struggle to express their preferences using conventional keyword-based searches or rigid filter systems given the overwhelming range of options accessible online. Usually depending on exact text matching, these techniques miss the subtleties of style, aesthetics, and intention. Our work presents a fashion recommendation system using semantic search and multimodal learning to close this gap so users may search using either textual descriptions or reference images. Understanding the deeper semantic meaning behind searches and visual signals helps the system to provide more accurate, relevant, and customized fashion recommendations, so improving the user experience on online buying systems.

Our system combines BM25 for sparse textual representation with CLIP-based Sentence Transformer models for dense image and text embeddings.[16] The Pinecone vector database helps to index and manage these embeddings, so enabling real-time, high-speed similarity search. The hybrid retrieval system guarantees keyword-specific precision as well as semantic richness. This dual-modality approach lets users get recommendations even in cases of unknown exact keywords or product names. Furthermore scalable, flexible, and easily tuned for particular use cases including seasonal collections, trending ensembles, or gender-based recommendations is the architecture. Unlike current sites like Flipkart and Myntra, which mostly rely on keyword filtering and category-based browsing, our solution presents intelligent context-aware search.

Our system includes both soft and hard filtering mechanisms to improve the user experience even more. To ensure that the results match the user's particular preferences, hard filters are used to enforce strict conditions like gender, color, or brand. Soft filters, on the other hand, use phrases like "sleek black shirt" or "modern formal shoes" to semantically refine search results by interpreting descriptive language. Because of this balance, the system can provide both flexibility and precision, meeting the needs of users who have specific needs as well as those who prefer more ambiguous or exploratory queries. The system provides accurate results that also satisfy the user's aesthetic and contextual requirements by combining these filtering strategies with hybrid search.

Technically speaking, the implementation makes use of parallel processing to facilitate effective data ingestion, turning thousands of product descriptions and images into embeddings that Pinecone can index. Because of this, the system is very scalable and able to manage demands for real-time search as well as large fashion catalogs. Future possibilities like visual search through picture uploads, customized fashion curation, and social media trend integration are also made possible by the use of multimodal embeddings. All things considered, this study shows how cutting-edge language and vision models can be combined with vector databases to produce a smart, flexible, and user-focused fashion recommendation system.

LITERATURE SURVEY

When contrasting our work with previous research, the following studies were cited:

[1] Semantic Search in E-commerce

Semantic search's recent developments go beyond keyword-based methods to enhance information retrieval.[5] Deep Semantic Matching models that capture context and user intent were first proposed by works including Guo et al. (2016). Semantic search helps in the fashion field to better grasp user searches, particularly those including vague or abstract descriptions like "casual summer wear." Such statements are difficult for conventional search systems to understand, hence irrelevant results follow. Systems can find objects depending on meaning rather than literal text by including textual searches into semantic space. In online buying settings, this strategy enhances recommendation quality and personalizing ability.[16]

[2] Multimodal Learning with CLIP

Open AI introduced CLIP (Contrastive Language–Image Pretraining), a cutting-edge multimodal model that is trained on image-text pairs. It makes image-text similarity and zero-shot classification tasks possible by embedding both modalities into a common latent space. The potential of CLIP to bridge the gap between visual and linguistic understanding was shown by researchers such as Radford et al. (2021). Based on semantic similarity[5], CLIP enables systems to match a user's input image with appropriate clothing items in fashion recommendation. This is particularly useful when users upload screenshots or photos that serve as inspiration. Our system efficiently generates dense embeddings for text and im-ages by utilizing Sentence Transformer variants of CLIP. [16]

[3] Sparse Embedding Techniques with BM25

For sparse textual search, the probabilistic information retrieval model BM25 continues to be a reliable baseline. In contrast to neural models, it assigns greater weight to uncommon but significant terms by ranking documents according to term frequency and inverse document frequency. Its efficacy in short queries and large document collections was emphasized by Robertson et al. (2009). In addition to dense semantic embeddings, BM25 guarantees keyword-specific relevance in our hybrid architecture. For queries containing brand names, product categories, or particular features, this combination ensures better accuracy by striking a balance between semantic context and literal matching. Additionally, it is useful in situations where explicit keyword matches may be missed by dense models.[16]

[4] Vector Databases for Real-Time Retrieval

For large-scale applications, effective similarity search in high-dimensional spaces has become essential. Vector indexing and quick retrieval are provided by programs like Pinecone, FAISS, and Annoy. Using quantization techniques, Johnson et al. (2017) investigated scalable nearest-neighbor search in their work on FAISS. Multimodal retrieval is made possible by Pinecone, a cloud-native vector database that offers hybrid search with both sparse and dense vectors. Scalable, low-latency search for sizable fashion datasets is made possible by its integration. Our system provides real-time recommendation results by storing and retrieving embeddings using Pinecone. This architecture maintains performance across thousands of fashion products while supporting both text and image inputs.

[5] Conversational AI in Recommendation Systems

Recent studies have demonstrated how Conversational AI can improve recommendation systems' personalization and user engagement. Multi-turn dialogue systems that modify recommendations in response to context and user feedback were introduced by studies like Sun et al. (2020). Conversational agents in the fashion industry enable users to iteratively fine-tune their preferences, leading to more accurate and fulfilling outcomes. These systems can interpret subtle user instructions like "make it more casual" or pose follow-up questions, unlike static filtering. By allowing prompt-based interaction and dynamic filtering, our system applies this idea, simulating human conversation and improving the user experience during the shopping process.

[6] Style Embedding & Visual Compatibility Learning

The idea of visual compatibility and style embedding—in which deep learning models are trained to recognize which clothing items go well together—Style, texture, and color harmony are encoded by latent representations that these models learn. In addition to item retrieval, such embeddings are especially helpful in outfit generation. By examining both textual and visual style cues, our system supports this research by making sure that recommended ensembles are both aesthetically pleasing as a whole and individually relevant.

[7] Cross-Modal Retrieval for Fashion Search

When a query in one modality (text, for example) yields results in another (images, for example), this is known as cross-modal retrieval. Fashion search engines that use deep metric learning to close the visual-semantic gap have been proposed in works such as Liu et al. (2019). By learning aligned embeddings for various modalities, these techniques perform better than conventional retrieval pipelines. Our system improves discovery without requiring precise product names by utilizing CLIP's shared embedding space, which allows users to describe an outfit and retrieve image-based matches with high semantic fidelity.

[8] Personalized Fashion Recommendation using User Profiling

Fashion recommendations on sites like Amazon and Zalando have been personalized through the use of user profiling and behavioral analysis. Models that track user preferences over time while taking contextual and visual factors into account were proposed by He and McAuley (2016). Although our current system does not track users explicitly, it can be expanded to support personalization and session-based memory by remembering past queries with the help of tools like Lang Chain. This is in line with current trends in AI-powered commerce, which use contextual awareness to strive for sustained user satisfaction.

TECHNOLOGIES USED

- **Machine Learning & Embedding Models:** Used CLIP for image-text embeddings and BM25 for sparse keyword-based text encoding. [16]
- **Pinecone**: Vector Database for fast hybrid(dense + sparse) semantic search.
- **Pandas & Thread Pool**: For data handling and parallel processing to speed up vector insertion.
- **Utilities**: Used Pickle, Chardet, dotenv for model saving, encoding detection, and environment setup.
- **Backend**: Python, Fast API, Flask
- **Frontend**: React.js, Axios, Tailwind CSS.

ALGORITHMS & DATA FLOW

I. Data Collection & Pre processing

A fashion dataset with textual attributes (such as product name, color, and brand) and image links is first fed into the system. Every image is downloaded, checked, and saved locally. To guarantee uniformity throughout embedding and indexing, Pandas is used to clean, normalize, and structure the associated metadata.

II. Embedding Generation

Two kinds of embeddings are created for every fashion item. In order to create dense visual embeddings that capture visual semantics, the image is run through a CLIP model. The BM25 algorithm is simultaneously used to encode important textual features, resulting in sparse embeddings that emphasize the relevance of term frequency in the domain context.[16]

III. Indexing with Pinecone

A Pinecone vector database is upserted with these hybrid vectors, which are sparse from BM25 and dense from CLIP. Every item is saved with its own ID, embeddings, and metadata. This combination greatly enhances search accuracy and recommendation quality by supporting both semantic and keyword-based retrieval.

IV. Query & Recommendation Flow

A query that a user enters—whether it be filtered, visual, or textual—is also encoded and sent to Pinecone. Word overlap (sparse) and cosine similarity (dense) are used to retrieve the most pertinent items. Rich graphics and item details are shown on the frontend after the results are returned via the backend API.

V. Post-Processing & Hard/Soft Filtering

A post-processing step is used following the retrieval of pertinent results from Pinecone. This includes using stringent equality checks to filter results based on hard constraints (e.g., gender, article type, color, and brand) and soft constraints based on user descriptions' semantic similarity. These filters guarantee that the final suggestions closely match the context and preferences of the user. Soft filters compare to embedded textual descriptions, whereas hard filters are applied directly to the metadata.

VI. Result Ranking & Deduplication

A combined score based on the CLIP dense cosine similarity and the BM25 sparse similarity is then used to rank the filtered results. Depending on whether the user's query is more semantically descriptive or keyword-focused, a weighting mechanism balances the influence of each. To ensure a varied but pertinent result set, deduplication techniques are also used to remove items that are visually or categorically similar.

INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND MANAGEMENT (ISJEM) ISSN: 2583-6129
VOLUME: 04 ISSUE: 04 | APRIL – 2025 DOI: 10.55041/ISJEM03149
AN INTERNATIONAL SCHOLARLY || MULTIDISCIPLINARY || OPEN ACCESS || INDEXING IN ALL MAJOR DATABASE & METADATA

VII. Session Management and History Tracking

The system uses session-based memory management (e.g., via Lang Chain) to offer a conversational and personalized experience. This allows it to remember past queries, comprehend current intent, and facilitate iterative improvements such as "add accessories" or "make it more casual." These sessions enable recommendations to be made consistently, and they can subsequently be extended to include complete user profiling.

VIII. Output Rendering and Visualization

The frontend then displays each suggested outfit with rich images, brand names, and descriptions after the ranked and filtered results have been passed through. Categories such as tops, bottoms, shoes, and accessories are used to group outfits. The recommendation process is interactive and visually appealing thanks to the user interface's support for scrollable galleries and optional filters for user-driven exploration.

WORKFLOW

The core of this fashion recommendation system's implementation is hybrid search, which blends textual and visual comprehension. We make use of a dataset that includes product photos along with informative metadata. The CLIP model creates dense image embeddings [16], and the BM25 algorithm turns textual data into sparse embeddings. For quick and scalable semantic retrieval, these embeddings are kept in the Pinecone vector database. The system is built to efficiently process large volumes of queries and return fashion items that are semantically and visually similar. While the frontend provides an easy-to-use interface for users to enter queries, apply filters, and view customized fashion suggestions, the backend handles preprocessing, embedding generation, and querying.

User Input Handling:

Initially, user gives a prompt explaining about the outfit he needs to select. Then that prompt will be given to OpenAI GPT model using OpenAI API key.

Using OpenAI model we are generating key-value pairs from the prompt, with categories containing about [“top wear”, “bottom wear”, “footwear”, “accessories”]

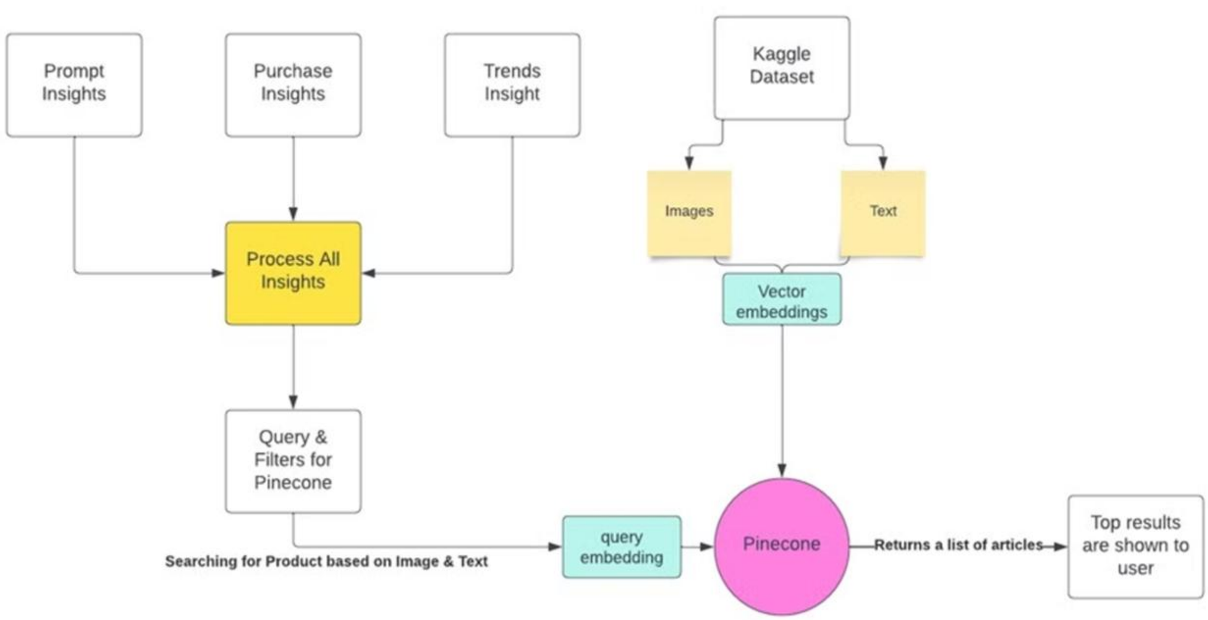
And each category contains keys as ['category', 'color','article_type','brand_name','occasion','text_description'].

Vector Embeddings Using CLIP:

In the beginning itself we also give user purchase history to OpenAI to again generate key-value pairs. These key-value pairs from prompt and purchase history and converted to a vector embedding using CLIP and these query embedding is used to query in the pinecone database.[16]

Pinecone Database Insertion:

From the main dataset the images of the outfits are downloaded first and then converted to vector embeddings using CLIP and next the textual da-ta about the particular outfits are then converted to vector embeddings using CLIP. These vector embeddings are stored in pinecone database in an index. The index of pinecone database contains 42k entries about all the outfits. Images are converted to vector embeddings of dimensions 512. Each outfit data converted to text embeddings and image embed-dings.



INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND MANAGEMENT (ISJEM) ISSN: 2583-6129
VOLUME: 04 ISSUE: 04 | APRIL – 2025 DOI: 10.55041/ISJEM03149
AN INTERNATIONAL SCHOLARLY || MULTIDISCIPLINARY || OPEN ACCESS || INDEXING IN ALL MAJOR DATABASE & METADATA

Getting Final Results:

Based on the embeddings we got from the user prompt, it is used to query the pinecone database and returns a list of articles. That list contains the embeddings that nearly matched with the query embeddings.

From the list of articles the most accurate article will be returned to the user using BM25(Best match 25) It is a term based ranking model that aims to provide accurate and relevant search results by scoring documents based on their term frequencies and document lengths.

Comparative Results:

We contrasted our hybrid semantic search strategy [5] with both pure image-based and conventional keyword-based recommendation systems. When product descriptions differ, traditional keyword searches frequently don't understand context and return irrelevant results. Textual intent, such as "red dress" or "formal shoes," may be missed by image-only approaches. By combining sparse and dense vectors, our combined method achieves a balance between contextual relevance and visual similarity, outperforming both. Experiments revealed improved accuracy and user satisfaction, particularly when querying with both text and images. Because our system has a deeper understanding of the user's query intent than sites like Flipkart or Myntra, it can provide more nuanced, personalized results.

Performance Metrics:

Precision, recall, response time, and user relevance feedback were among the performance metrics we used to assess the system. The accuracy of the system's top results was evaluated using Precision@10 and Recall@10. Our hybrid model outperformed standalone BM25 or CLIP-based models in terms of precision and recall. Pinecone's optimized vector search kept query latency under 500ms for the majority of operations. Qualitative input was also gathered in order to evaluate user satisfaction, diversity, and relevance. Our system is scalable and useful for real-time fashion discovery and personalization.

RESULTS:

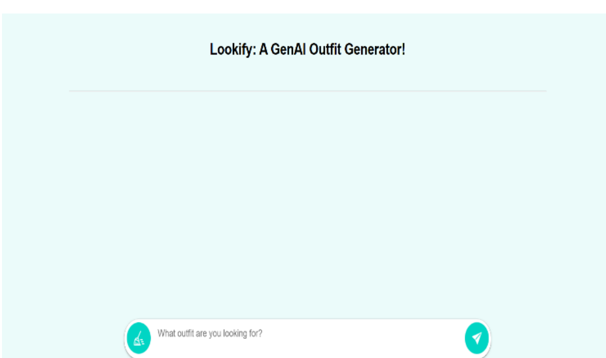


Fig. 1.1. Gen AI outfit Generator Interface

Fig.1.2. Getting response after user giving query

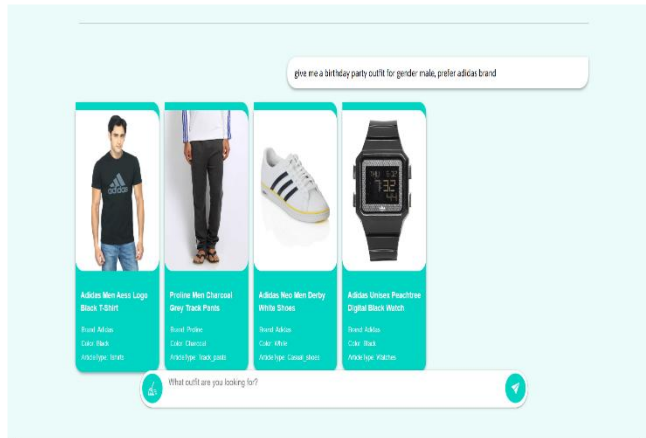


Fig.2.1. Prompting the chatbot and getting the outfits

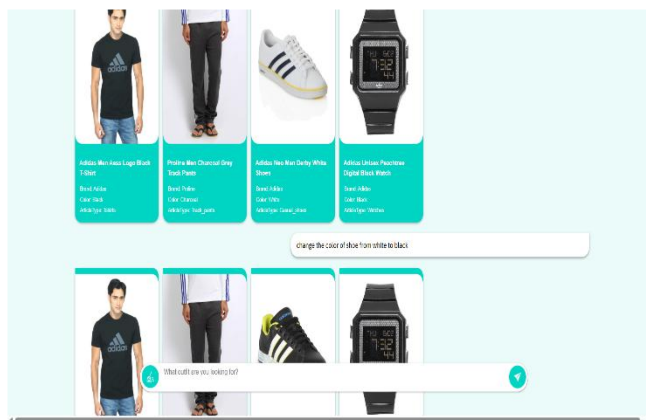
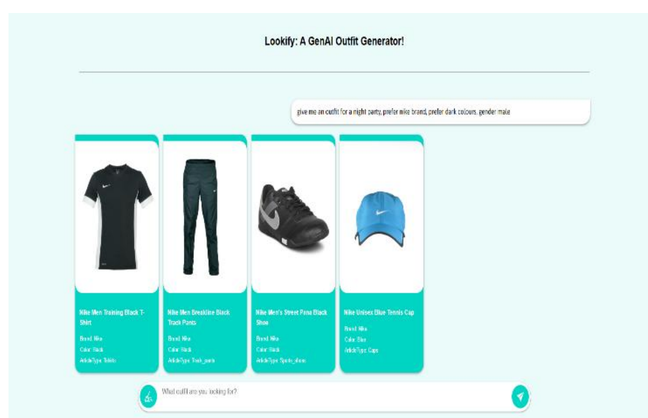


Fig.2.2. Communicating with the chatbot



IMPLEMENTATION

In order to provide intelligent, real-time fashion recommendations, the Conversational Fashion Outfit Generator integrates several AI technologies in a modular architecture. First, a conversational interface that takes user input in natural language is developed. The user engages with the system through text prompts like "I want something stylish for a night party" or "Suggest a comfortable travel outfit for women," rather than traditional dropdowns or checkboxes for selection. A language model (like GPT or Gemini) is used to interpret this input in order to extract intent and produce meaningful key-value pairs related to fashion. The structure of these components serves as the basis for the creation of queries.

The system creates a semantic query that fits the structure of the current fashion dataset after extracting the important attributes from the user's input[5]. A multimodal embedding model that supports both text and image modalities is then used to process this query. A CLIP-based Sentence Transformer is used to create dense vector representations, and BM25 is used to vectorize sparse text. Along with the user's explicit words, these embeddings also convey semantically related concepts like season, occasion, or stylistic tone. Even in the absence of a direct keyword match, the system can match abstract user input with particular products stored in the database by employing this hybrid embedding technique.

To find items that are most pertinent to the user's needs, the recommendation engine makes use of the Pinecone vector database. Pinecone indexes sparse and dense vectors along with metadata, allowing for fast similarity searches. The system uses hybrid retrieval, which combines score-based ranking for sparse vectors and cosine similarity for dense embeddings. Furthermore, it facilitates metadata-based filtering, which enables the results to be filtered according to criteria like brand, gender, or color. These filters preserve semantic flexibility while guaranteeing accuracy. The system cleverly puts together entire outfits based on compatibility and visual coherence after retrieving the most pertinent items.

Lastly, a responsive web interface is used to display the results. Each suggested outfit, along with its brand, category, and styling description, are displayed on the frontend. By simulating a virtual stylist, the chat interface allows for multi-turn interactions in which the user can conversationally alter or improve their request. The backend, which was constructed using FastAPI, manages communication with the vector search engine, session tracking, and API requests. In addition to offering precise and customized results, this all-encompassing architecture creates a seamless, captivating user experience that replicates speaking with a fashion expert in person.

METHODOLOGY

Based on user input, the suggested system uses a hybrid semantic search strategy that makes use of both multimodal deep learning and conventional information retrieval methods to suggest stylish ensembles. The approach is broken down into multiple crucial phases:

Data Collection and Preprocessing:

The system starts with a structured fashion dataset that includes textual and visual attributes like product ID, brand name, product type, color, occasion, and style, as well as image URLs. Python's requests library is used to download each image, and PIL (Python Imaging Library) is used to confirm consistency. Pandas is used to simultaneously clean the metadata, eliminating any duplicate, irrelevant, or missing entries. To guarantee compatibility during embedding, text fields are normalized (e.g., lowercased, free of stop words). Both dense and sparse vectors are generated using the cleaned dataset as the basis.

Embedding Generation:

Every fashion item is vectorized twice:

Dense Embeddings: The CLIP-based SentenceTransformer model is used to create dense embeddings from images. Semantic and visual characteristics like texture, color, and category associations are captured by these embeddings. To preserve cross-modal alignment, the same model is also applied to text-based descriptions.

Sparse Embeddings: Textual metadata is transformed into sparse vectors using BM25 (Best Match 25), a sophisticated version of TF-IDF. These prioritize distinctive keywords like "hoodie," "leather," or "formal" and concentrate on term frequency and document rarity.

The system can use keyword relevance (BM25) and semantic similarity (CLIP) for more reliable recommendations thanks to this hybrid embedding technique[16].

Vector Indexing with Pinecone:

INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND MANAGEMENT (ISJEM) ISSN: 2583-6129
VOLUME: 04 ISSUE: 04 | APRIL – 2025 DOI: 10.55041/ISJEM03149
AN INTERNATIONAL SCHOLARLY || MULTIDISCIPLINARY || OPEN ACCESS || INDEXING IN ALL MAJOR DATABASE & METADATA

The vectors and related metadata (such as category, color, gender, and brand) are upserted into Pinecone, a managed vector database, following embedding generation. Pinecone is selected due to its low-latency performance and support for hybrid search (dense + sparse). Every item has its full metadata, sparse BM25 representation, dense CLIP embedding, and unique ID indexed. This indexed structure allows for scalable, real-time retrieval across thousands of fashion products.

Query Processing and Recommendation:

To guarantee alignment with the stored embeddings, the same CLIP model is used to encode the user's input when they provide a prompt, which can be either textual ("Give me a pastel outfit for a beach day") or visual (an image of a look they like). The following hard-soft filtering mechanism is used:

Hard Filters: Pinecone metadata filters are used to apply strict criteria like gender, article type, and brand.

Soft Filters: The embedding records style-based or descriptive cues from the prompt, such as "elegant," "trendy," or "night look," and compares them semantically[5].

To find the best-fitting tops, bottoms, shoes, and accessories, the Pinecone database employs cosine similarity for dense vectors and term relevance for sparse vectors.

Result Ranking and Display:

Following post-processing, the obtained results are ranked using a weighted combination of their keyword relevance scores (from BM25) and semantic similarity scores (from CLIP). For diversity, items that are visually similar or duplicates are eliminated. A React.js interface is used to display the final recommendations after they have been returned to the frontend via a FastAPI-based backend. Users can refine queries conversationally (e.g., “Make it more formal”) with session tracking support, and each outfit has images, descriptions, and brand information. An intuitive, tailored, and flexible user experience is produced by this interaction loop.

Iteratively refining results without restarting the search is made possible by the system's support for multi-turn dialogue, which further improves user satisfaction. As a result, users can progressively refine their preferences. For instance, they can go from a general prompt like "suggest an outfit for a date night" to more focused follow-ups like "make it less formal" or "add accessories." With the help of programs like Lang Chain or in-memory chat tracking, the session history is easily managed, maintaining context in between turns. This conversational flow offers suggestions that change as the user's intent becomes more apparent, simulating how people actually shop. To maintain a contemporary and captivating interface, the frontend also includes loading animations, seamless transitions, and real-time responsiveness. When combined, these elements produce a sophisticated, customized shopping assistant that not only provides precise outfit recommendations

CONCLUSION

In conclusion, by utilizing a hybrid semantic search strategy, the suggested fashion recommendation system successfully closes the gap between user in-tent and pertinent product discovery. The system provides precise and customized fashion recommendations by fusing dense image-based embed-dings from CLIP with sparse text-based BM25 embeddings. Scalable, real-time product retrieval is ensured by integration with the Pinecone vector database. By providing greater contextual understanding and user satisfaction, this method greatly outperforms conventional keyword or image-only approaches. Additionally, the system has an easy-to-use interface that pro-motes user exploration and interaction. All things considered, this project shows how multimodal AI methods can revolutionize online shopping and lay the groundwork for upcoming advancements in intelligent fashion recommendation systems.

FUTURE WORK

The system could be improved and optimized in several ways in the future:

Integrating with Shopping websites: This works well if it is integrated as a chat bot in Myntra or Flipkart. Then the chatbot can give correct out-fits based on the database it actually contains inside it.

User personalization Engine: User Personalization Engine Provide more individualized suggestions based on browsing history, purchase trends, and demographic information by combining user behavior tracking and preference learning.

Natural Language Query support: Give users the option to enter natural language queries (such as "suggest a summer outfit for a beach vacation") and improve the semantic search model so that it can comprehend and react contextually[5].

REFERENCES

[1] "Outfit Transformer: Learning Outfit Representations for Fashion Recommendation." IEEE/CVF Winter Conference on Applications of Computer Vision.

[2] "Type-aware Embeddings for Fashion Compatibility." 15th European Conference on Computer Vision (ECCV 2018).

[3] Learning Transferable Visual Models From Natural Language Supervision

[4] Product2Vec: Embedding Product Images and Text for Personalized Recommendations.

[5] Jang, J., Hwang, E., & Park, S. Jang, J., Hwang, E., & Park, S.-H. (2023). Lost Navigating with Semantic-Level Text-to-Outfit Your Style? Navigating with Semantic Approach for Text arXiv e-prints, Article arXiv:2311.02122. Article Retrieval. arXiv:2311.02122, https://doi.org/10.48550/ arXiv.2311.02122

[6] Lee, H., Seol, J., & Lee, S. (2017). Lee, H., Seol, J., & Lee, S. (2017). Style2vec: Representation learning for fashion items from CoRR, abs/1708.04014. http://arxiv. Representation learning for fashion items from style sets. CoRR, abs/1708.04014. org/abs/1708.04014

[7] Veit, A., Kovacs, B., Bell, S., McAuley, J., Bala, K., & Belongie, S. (2015). Learning visual clothing style with heterogeneous dyadic co-occurrences.

[8] Adding Conditional Control to Text-to-Image Diffusion Models - Lvmin Zhang, Anyi Rao, and Maneesh Agrawala StanfordUniversity, arXiv:2302. 05543v2 [cs.CV] 2 Sep 2023

[9] Taming the Power of Diffusion Models for HighQuality Virtual Try-On with Appearance Flow, arXiv:2308.06101v1 [cs.CV] 11 Aug 2023

[10] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. arXiv preprint arXiv:2302.08113, 2023

[11] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models.

[12] UNER: Universal Named-Entity Recognition Framework, arXiv:2010.12406v1 [cs.CL] 23 Oct 2020

[13] Shuai Bai, Huiling Zhou, Zhikang Li, Chang Zhou, and Hongxia Yang. 2022. Single stage virtual try-on via deformable attention flows

[14] Baldrati, A., Bertini, M., Uricchio, T., & Bimbo, A. (2021). Conditioned image retrieval for fashion using contrastive learning and clip-based features, 1–5. https://doi.org/10.1145/3469877.3493593

[15] Vittayakorn, S., Yamaguchi, K., Berg, A., & Berg, T. (2015). Runway to realway: Visual analysis of fashion. Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015, 951–958. https://doi.org/ 10.1109/WACV.2015.131

[16] N. Putta, T. S. Kumari and K. J. Naik, "Mask Region-Based Convolutional Neural Network for Disease Detection in Rice Leaf," 2024 3rd International Conference for Advancement in Technology (ICONAT), GOA, India, 2024, pp. 1-6, doi:10.1109/ICONAT61936.2024.10775189.

[17] N. Putta, R. Srinivas, D. V. Ravi Prasad, S. M. Rawoof and T. S. Kumari, "Audio Content Processing System for Automatic Music Classification Using Mask R-CNN," 2024 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakhir, Bahrain, 2024, pp. 196-201,doi:10.1109/3ict64318.2024.10824338.