One size does not fit all: Improving Fashion Recommendation relevance via fit-aware Neural Re-ranking

Vishal G*
UMass Amherst
vishalg@umass.edu

Sai Sreenivas Chintha*
UMass Amherst
saisreenivas@umass.edu

Harshitha Kolukuluru*
UMass Amherst
hkolukuluru@umass.edu

Abstract

Fashion recommendation systems are playing an increasingly important role in enhancing user experience on e-commerce platforms by delivering personalized and relevant product suggestions to them. However, traditional recommendation models frequently fail to capture subtle but important details like personal fit and individual style preferences, resulting in suboptimal results. In this project, we aim to introduce a fit-aware neural re-ranking approach designed to improve the accuracy of fashion recommendations by leveraging custom-extracted features like bounding box dimensions, and color attributes. To achieve this, we follow a two-stage methodology: first, fine-tuning the CLIP model to generate semantically rich visual and textual embeddings from our data, and second, integrating these embeddings into a re-ranking neural network optimized for fit-aware recommendations. By combining these custom features with advanced re-ranking techniques, the approach aims to improve the recall of the retrieved items ensuring that the users receive more tailored and satisfying suggestions.

1. Introduction

Image retrieval is a fundamental task in the domain of information retrieval, where given a textual description of an image, the goal is to retrieve the top-k images that best match this description. With the rapid growth of e-commerce, especially in fast fashion, image retrieval systems have become integral to enhancing user experiences by delivering personalized and relevant fashion recommendations. By tailoring product suggestions to individual preferences, these systems not only improve user satisfaction but also boost sales through increased engagement by making product discovery both relevant and efficient.

Conventional image retrieval methods primarily rely on visual features and text embeddings, without paying sufficient attention to critical factors like individual fit and style preferences. This gap in recall becomes particularly relevant in the "top-k" recommendations, where users expect highly relevant and personalized options. In addition to this, traditional image retrieval models in e-commerce which might perform well in retrieving relevant images with high recall, often face limitations in accurately ranking these items by their relevance to specific user needs. For instance, a search for "short-sleeve summer dresses" might return results with varying sleeve lengths and styles, where only a subset truly matches the user's intent. Such mismatches highlight the need for precise re-ranking techniques that can distinguish between items based on fit, style, and other personalized criteria.

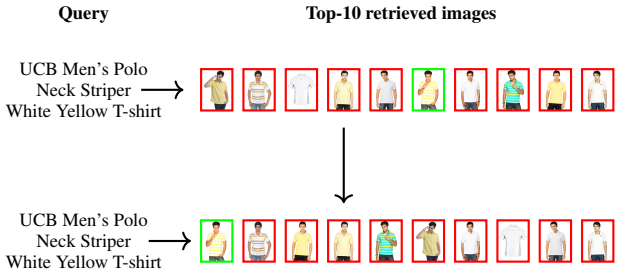

Figure 1. Improvement in rank of the ground truth image (green) which is retrieved at a higher rank, whereas our model (bottom) increases recall@k by ranking the ground truth image as the top result.

To address this challenge, we introduce a fit-aware neural re-ranking approach to improve the relevance of image retrieval in fashion recommendations. Our project consists of two stages: an initial retrieval phase using CLIP embeddings to generate a broad set of relevant results with high recall, followed by a re-ranking phase that leverages custom features extracted from the images. These features include bounding box dimensions and dominant colors. By focusing on aligning the top-k results with detailed user requirements, our method enhances the recall of the recommendations (Figure 1)

[1] all authors contributed equally

2. Related Work

Cross Domain Retrieval Methods: In multi-modal learning, two main approaches namely: single-stream and two-stream methods are commonly used to model the relationship between images and text. Single-stream transformer models, such as ViLBERT[14], VisualBERT[11], VL-BERT[20], Unicoder-VL[9], ImageBERT[15], and Unified VLP[25], combine image and text embeddings by concatenating them for input into a single BERT-style encoder, to achieve strong alignment by jointly processing image and text. These models are pre-trained on tasks like Masked Language Modeling (MLM), Masked Region Modeling (MRM), and Multi-Model Alignment Prediction (MMAP) [23], allowing the model to build a combined understanding of visual and textual information. Some models, like UNITER[4] and OSCAR[12], include additional pre-training steps to improve alignment between image regions and text by using object categories or leveraging image depth features rather than predefined regions of interest. While effective at building a comprehensive understanding, these one-stream methods are often computationally intense and less efficient.

On the other hand, two-stream methods, such as ViLBERT[14], LXMERT[5], and CLIP[16] address this problem by keeping the image and text embeddings separate, using distinct encoders for each modality. This approach allows for more efficient calculation of image-text similarities by relying on dot-product comparisons between image and text embeddings, making it particularly effective for cross-modal retrieval.

Among the two-stream models, CLIP (Contrastive Language-Image Pre-Training) stands out due to its effectiveness in aligning image and text embeddings. Trained on 400 million image-text pairs, [16] uses contrastive learning to learn joint representations of images and text. Positive pairs are taken from aligned text-image pairs while negative pairs are sampled randomly, with the InfoNCE loss [22] being optimized to align these representations via contrastive learning. A top-k item list is retrieved to maximize recall followed by a re-ranking step to optimize recall [19], where a neural network is trained to predict a relevance score $score_\theta(q, d)$ given a query q and document d, where parameters θ are learnable. This contrastive learning enables CLIP to perform zero-shot classification and retrieval tasks with competitive performance across domains, making it highly adaptable for specific applications, including fashion.

Feature Extraction:[3] show learning to rank can help improve the fashion recommendation systems by training models on various feature representations. The works of [24] further highlight how integrating custom extracted features like bounding boxes, colors, textures and other style elements can help improve the overall retrieval recall. There we propose extracting a set of custom features from the top-k retrieved images.

Fashion Based Tasks: Different models have been proposed for fashion-specific image-to-text retrieval tasks. For example, FashionBERT[7] uses a patch-based masking approach to learn representations of fashion items, focusing on capturing image and text features relevant to clothing attributes. MAAF[6] , aimed to improve fashion retrieval by employing a modality-agnostic attention fusion that better aligns images and descriptions through image-level attention. FashionVLP[8] introduced a pre-training framework designed for fashion analysis by jointly learning vision and language features.

KaleidoBERT[26] divided images into multiple patches, allowing the model to learn the relationships between image regions and descriptive text at a more granular level. Similarly, ViLT[10], streamlined the process by applying linear projection to image patches, making it more lightweight and efficient for multi-modal tasks, including fashion-specific retrieval. However, these approaches still face challenges in capturing fine-grained details, particularly when dealing with the intricate textures, patterns, and style attributes unique to fashion items.

While CLIP provides powerful cross-modal retrieval capabilities, it is trained on a broad dataset with generic image-text pairs, lacking the fashion-specific focus needed for high recall in this domain. Therefore, CLIP may miss fine-grained attributes like fabric texture, cut, or subtle color variations critical for fashion retrieval, highlighting the need for a tailored solution. CMA-CLIP [13] aims to address this by adding more cross-modal attention layers for finer, token-wise alignment between image and text embeddings. OpenFashionClip[2] tries to achieve the same by fine-tuning CLIP on multiple fashion-related datasets.
In the coming sections, we ameliorate these limitations by making it adapt for fine-grained fashion retrieval by introducing custom features specific to fashion, such as detailed color histograms, texture descriptors, and key clothing attributes (e.g., neckline and sleeve length). Through a re-ranking of these custom feature embeddings, we aim to improve CLIP's ability to capture and utilize these nuanced details, enhancing its performance on fashion-specific retrieval tasks.

Our work builds on CLIP's architecture by making it adapt for fine-grained fashion retrieval by introducing custom features specific to fashion. Through a re-ranking neural network trained on these custom feature embeddings, we aim to improve CLIP's ability to capture and utilize these nuanced details, enhancing its performance on fashion-specific retrieval tasks.

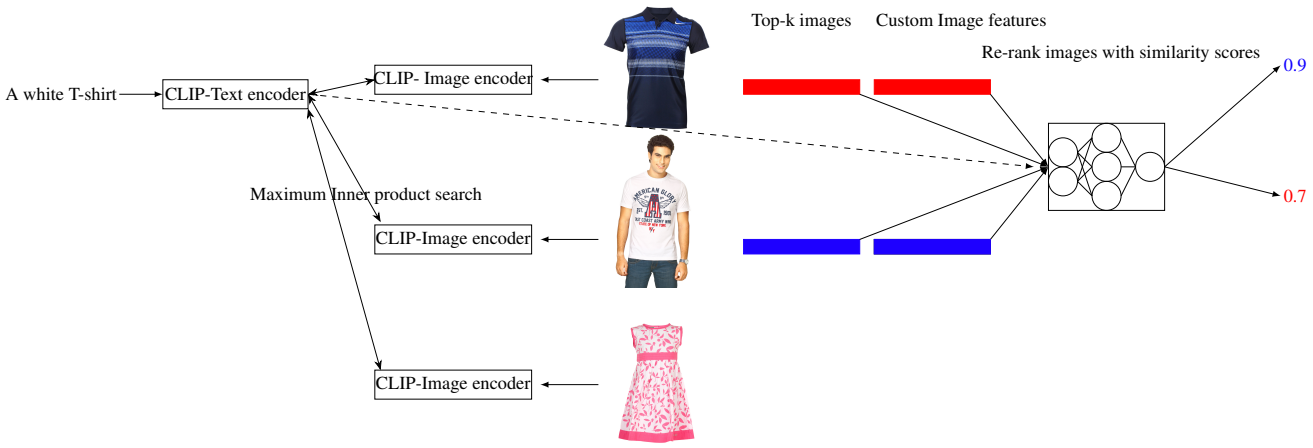

Figure 2. Diagram of our pipeline to first retrieve images and use re-ranking with custom image features such as bounding box, sleeve length etc. along with the embeddings of original image and query to achieve better recall

3. Technical Approach

Many of the previous works, use CLIP embeddings of texts and images and compute maximum inner product search to retrieve images given text and vice-versa. We wish to explore a pipeline where, we re-rank the retrieved images, via learning a scoring similarity function to improve the $R@k$ of the system. We propose to use features extracted from the given image such as dimensions of the bounding box, color of the item, and so on, and pass them to a neural network along with the input query to re-rank the top-k retrieved items. We also wish to do the following extensions to this

1. **CLIP (Baseline):**To build the foundational model for our project, we use the CLIP model to generate embeddings for both text and images. These embeddings are projected into a shared semantic latent space, allowing us to make meaningful comparisons based on their similarity. To validate the baseline performance, we conducted experiments on the Fashion30K[1] dataset, a benchmark dataset for image-text retrieval tasks. This dataset was systematically analyzed to identify textual attributes—such as garment type, color, and product descriptions—that significantly contribute to generating contextually meaningful embeddings.

   For this phase, we selected a subset of 5,000 images from the dataset. The embeddings for the textual queries as well as the images are then computed using the pretrained CLIP model. We used the cosine similarity between these embeddings to rank the retrieved images based on their relevance to each query. This ranking process leverages CLIP's ability to align text and image representations into a shared embedding space, which is crucial for accurately matching multimodal content.

   To evaluate the baseline model, we conducted a series of retrieval experiments and analyzed the results, which are presented in detail in subsequent sections. These evaluations provided insights into the model's strengths, including its capacity to retrieve semantically relevant items, and highlighted areas for potential enhancement, such as improved ranking recall for nuanced queries. Building on this foundation, we further incorporated fine-tuning, custom features, and re-ranking mechanisms to address the observed limitations.
2. **Fine-tuned CLIP:** While the baseline CLIP embeddings are effective for general image-text alignment, they often lack the ability to capture more nuanced, domain-specific characteristics that play an important role in understanding fashion items. To improve the retrieval process and enable our model to cater better to the specific requirements of fashion recommendations, we enhanced the baseline CLIP model by fine-tuning it on fashion30k dataset. Fine-tuning the model using a dataset curated specifically for fashion, we aimed to refine its understanding of critical attributes such as garment styles, proportions, colors, and textures. This approach allowed the model to generate embeddings that align more closely with the unique characteristics and demands of the fashion domain. We then evaluated our fine-tuned CLIP model similar to the baseline model and the results of the same are presented in the subsequent sections.
3. **Neural network for re-ranking (NN):** To enhance the relevance of fashion recommendations, we use images retrieved by the fine-tuned CLIP model to train a custom neural network. For each image, specific features are extracted to better understand the visual characteristics of fashion items. One important feature is bound-

Category	CLIP (baseline)					CLIP (finetune)				
	R@1	**R@3**	**R@5**	**R@10**	**R@20**	**R@1**	**R@3**	**R@5**	**R@10**	**R@20**
Innerwear (76)	0.28	0.62	0.72	0.87	0.95	0.54	0.87	0.92	1.00	1.00
Bottomwear (101)	0.31	0.52	0.63	0.75	0.92	0.46	0.68	0.80	0.95	0.99
Bags (123)	0.24	0.46	0.60	0.72	0.86	0.41	0.67	0.80	0.93	1.00
Topwear (468)	0.22	0.38	0.46	0.62	0.74	0.41	0.66	0.77	0.88	0.96
Watches (112)	0.08	0.21	0.29	0.42	0.60	0.21	0.40	0.54	0.71	0.83
Shoes (235)	0.12	0.34	0.45	0.64	0.80	0.41	0.61	0.75	0.91	0.97

Table 1. Category-wise performance comparison of CLIP baseline and finetuned models

ing box dimensions, which are determined using Faster R-CNN[18]. Faster R-CNN[18] combines region proposal generation and object detection into a single trainable framework, providing high accuracy and efficiency for object detection tasks. These bounding boxes help focus on the important parts of an image, like clothing items, while ignoring irrelevant areas. Once the bounding boxes are identified, we crop the image to isolate these sections for further analysis, ensuring we're working with the most important parts of the image.

Another key feature is the dominant color, identified with KMeans[21] clustering. After cropping the image to relevant areas, the KMeans clustering algorithm groups similar pixels of the image together to find the most common color in it. For example, when analyzing a shirt, KMeans can identify blue as the dominant color. This information helps improve recommendations by highlighting colors that align with what the user may prefer. By combining these features with CLIP embeddings, we aimed to create detailed and meaningful image representations.

We structure the problem as a binary classification and the training process involved creating enriched embeddings by combining the fine-tuned CLIP embeddings with these custom features. Each training sample consists of the following:

- **Input embedding:** This embedding is a created by concatenating the Query embedding (fine-tuned CLIP embedding of the text query) with Custom feature embedding $E_{custom-feature} = E_{cropped-image} + E_{dominant-color}, E \in \mathbb{R}^d$(where E is generated from CLIP).
- **Label:** A binary value 1, indicating the paired image is ground truth image and 0 otherwise.

For validation, the query and its list of retrieved images were fed into the network. The system reorders the images based on relevance scores predicted by the custom network. We then evaluate the model by calculating the $R@k$ for different Ks on the reordered list and also identifying cases where the GT image's position in the ranking has improved. This iterative process refines the retrieval and ranking mechanism.

By leveraging custom features like bounding box crops and dominant colors, combined with fine-tuned CLIP embeddings, the neural network learns richer representations, leading to more accurate and personalized recommendations tailored to user preferences.

Model	R@1	R@3	R@5	R@10	R@20
CLIP (zero-shot)	0.21	0.40	0.50	0.65	0.80
FashionVLP	0.26	-	-	-	-
CLIP (finetune)	0.42	0.66	0.78	0.90	0.95

Table 2. Model performance: CLIP (baseline) vs FashionVLP vs CLIP (finetune)

4. Results

We present the results of our proposed methodologies on the Fashion30K dataset below along with the qualitative analysis of the retrieved images. Subsequently we will provide a detailed discussion about the the key experiments conducted during our study in the next section.

1. **Quantitative Evaluation:**
   We evaluated the performance of our methodologies by measuring the Recall at various values of k (R@k) 1, 3, 5, 10, 20. The results for the CLIP baseline as well as the fine-tuned CLIP models on 1,500 queries of Fashion30K are summarized in Table 2. We observe that the fine-tuned CLIP model consistently & significantly outperformed the baseline CLIP model and FashionVLP across all settings. One crucial difference is that FashionVLP was evaluated on 200k test set originally, but we evaluated on 1.5k subset of the test set, due to computational constraints. This improvement can be attributed to the domain-specific fine-tuning of the CLIP model on the fashion dataset, which enabled the model to learn more nuanced features relevant to fashion queries. A qualitative analysis on the same is provided in the next section.

   In addition to evaluating the overall performance, we also assessed the category-wise recall at different values of k, to closely assess how the models performed across

different product types that are present in our dataset (Table 1). The fine-tuned CLIP model demonstrated significant improvement across all categories, with the most notable gains observed in Innerwear and Bottomwear product types. However, categories like Watches and Shoes showed relatively lower recall values, which could be attributed to challenges in capturing specific visual features, such as small dials or intricate designs, that require more nuanced training data.

Additionally, we evaluated our neural network trained using the custom features that were derived from the embeddings of both the CLIP baseline and the fine-tuned CLIP model (Table 3 & Table 4). Interestingly, while the neural network was effective in re-ordering certain examples, its overall performance did not surpass that of the CLIP models. This suggests that while the model was able to capture specific patterns, it may have lacked the robustness and generalization capabilities of the CLIP embeddings. Qualitative examples illustrating this behavior are provided in Figure 5.

Model	R@1	R@3	R@5	R@10	R@20
CLIP (zero-shot)	0.16	0.34	0.45	0.61	0.86
NN (baseline)	0.06	0.20	0.32	0.53	0.82

Table 3. Model performance: CLIP (baseline) vs NN

Model	R@1	R@3	R@5	R@10	R@20
CLIP (finetune)	0.30	0.49	0.58	0.75	0.93
NN (finetune)	0.13	0.31	0.44	0.64	0.86

Table 4. Model performance: CLIP (finetune) vs NN

2. **Qualitative Evaluation:**
CLIP vs. Fine-tuned CLIP The fine-tuned CLIP model demonstrated a notable improvement in understanding domain-specific queries compared to the baseline CLIP model, demonstrating its ability to align the visual and textual features more effectively.

For instance, Figure 3 provides an example of improved retrieval capabilities of the fine-tuned model. When presented with a query "*Catwalk Women Blue Ballet Shoes*" the fine-tuned model was able to successfully prioritize the relevant image of *ballet flat* over *casual or sporty shoes*. While the CLIP baseline model placed the ground truth (GT) image at rank 10, the fine-tuned version moved it to rank 2. This improvement can be attributed to the effectiveness of the fine-tuning process, which aligned the model to better understand the nuanced fashion attributes, such as ballet flats, specific styles, and gender-related visual features within the fashion domain.

However, the fine-tuned model also faced challenges in certain cases. For instance, in Figure 4, for the query "*Turtle Men Check Green Shirt*" while the fine-tuned model improved by recommending visually similar items with respect to the fashion pattern "*checked*", it failed to retrieve the ground truth (GT) image. This may be due to the limited number of brand-specific examples in the training dataset, leading the model to prioritize more general attributes (checked pattern) over the specific brand name("*Turtle*").

Neural Network with Custom Features The neural network trained with custom features, such as dominant color and cropped image regions, exhibited mixed results. While it could reorder some results correctly, its overall performance fell short when compared to that of the CLIP baseline and the fine-tuned models.

For example, in Figure 5, the query "*Maxima Men Green Dial Watch*" resulted in the neural network recommending a watch with a "*green strap*" instead of one with a "*green dial*". This mismatch highlights a limitation in the feature extraction process: while dominant color features capture the overall color of the image, they fail to focus on specific regions such as the dial or strap. Moreover, all features were given equal weight during training process, which likely contributed to the misalignment between the query and the recommendation.

To address this limitation, we hypothesize that introducing more refined feature extraction techniques, such as cropping specific image regions like straps, dials, or other product-specific components— to better align the visual features with the text query would improve our results. Additionally, incorporating a weighted feature integration technique during training could also help assign greater importance to relevant features based on the query context, improving the overall alignment and recommendation accuracy.

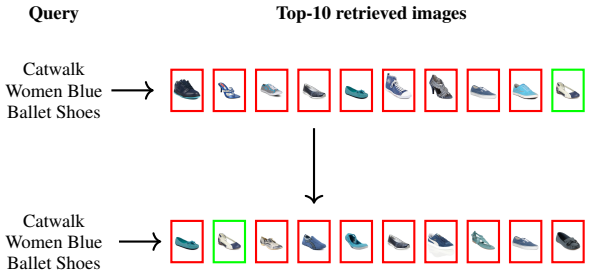

Figure 3. In the CLIP baseline model(top), the ground truth image (green) is retrieved at a lower rank (10), whereas the finetuned CLIP model (bottom) the ground truth image is retrieved at a higher rank (2).

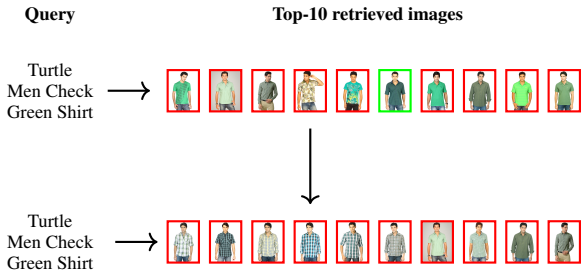

Figure 4. In the CLIP baseline model(top), the ground truth image (green) is retrieved at rank 6, whereas in the finetuned CLIP model (bottom) the ground truth image isn't in top-10.

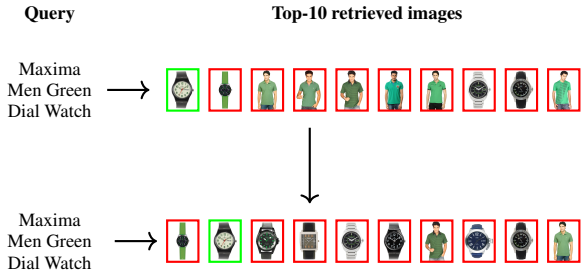

Figure 5. CLIP finetuned model (top) vs NN model (bottom)

5. Experiments

Dataset We utilized 5k samples from the Fashion30K dataset for training and evaluation, initially focusing on features such as *productDisplayName*, *season*, and *usage*. To reduce redundancy in the text query, we refined it by removing metadata and eliminating duplicate terms. However, we observed that *season* and *usage* still introduced redundancy, negatively impacting performance.

Additionally, analyzing the model performance across product categories of the dataset revealed that it underperformed in categories like watches, bags, and top wear, likely due to insufficient feature representation. To address this, we incorporated additional features, including *materialType*, *baseColour*, and *pattern*, which were more relevant for these categories of the dataset. This refinement improved model performance significantly.

Our final dataset consisted of 3.5k training samples, optimized to enhance the model's accuracy and generalization across diverse product categories.

Custom feature extraction We tested both YOLO[17] and Faster R-CNN for object detection and feature extraction. YOLO[17] was quick and efficient, making it suitable for scenarios requiring real-time processing. However, it struggled to capture fine-grained details in product images. Faster R-CNN, while slower and more resource-heavy, excelled at identifying intricate features, which was crutial for improving accuracy across complex categories. In the end, the precision and detail offered by Faster R-CNN outweighed its computational demands, making it the ideal choice for our application.

CLIP finetuning In our fine-tuning setup, we experimented with different hyperparameters such as batch size, temperature, and learning rate. The training process involved 3.5k image-query pairs with InfoNCE loss applied between the query and image pairs. We employed in-batch negative samples, using a batch size of 64 and training for 5 epochs. The Adam optimizer was used with a learning rate of 5×10^{-5} to minimize the InfoNCE loss. We experimented with various values of temperature while taking softmax, in our setting a temperature of 0.07 seemed to work the best, as it lead to faster convergence within 5 epochs.

In our current approach, we concatenate the product text features to form the query for training. Further, we believe that a more structured prompt engineering strategy, such as "*This is an image of a [productDisplayName] with pattern [pattern]...*", could better align the model's understanding of the query and image features, potentially leading to improved results. Due to time and compute limitations, we were not able to further explore the impact of alternative hard-negative sampling strategies and train for more epochs.

Neural network We conducted several experiments with various custom features, starting with cropped images and progressively incorporating additional features such as dominant colors. During training, we explored different strategies for combining embeddings and found that adding the embeddings for the custom features resulted in better performance as compared to concatenating them.

Moreover we experimented with various types of hard-negative sampling techniques. Initially, we randomly selected a non-ground truth image from the retrieved set and designated it as a negative sample for the query. However, this approach led to poor performance, likely due to the random nature of negative sampling within the retrieved set. To improve performance, we implemented a more robust negative sampling strategy. Specifically, we selected the least relevant images (i.e., the worst-ranked images from the CLIP retrieval list) as negative samples. This adjustment yielded better results, aligning more closely with our expectations.

Additionally, we performed hyperparameter tuning, including hidden layer sizes, batch sizes, dropout rates, and the use of batch normalization layers. These hyperparameter settings played a critical role in optimizing model performance, and the best results were achieved with the following settings: Weight initialization was done via kaiming uniform with a learning rate of 0.0001 with CLIP and 0.0008 for fine-tuned CLIP. The hidden layer sizes are 512, 256, and 64 with optimizer as Adam and batch size of 32, with a dropout of 0.5. Training was conducted for 50

epochs. We trained the model on a dataset of approximately 8k samples, as detailed in the approach section, to evaluate the impact of these features.

6. Conclusion

In this study, we explored methods to improve fashion recommendations using the Fashion30K dataset, focusing on both quantitative and qualitative performance evaluations. Fine-tuning the CLIP model on fashion-specific data significantly improved its ability to handle domain-specific queries, outperforming the baseline in Recall at k (R@k) metrics. While the model performed well across most categories, it struggled with items like Watches and Shoes, where intricate features required more nuanced training data.

To address the limitations of generic embeddings, we integrated custom features such as bounding box dimensions and dominant colors, with Faster R-CNN excelling at capturing detailed visual elements. Although a custom neural network incorporating these features showed promise in reordering recommendations, its performance highlighted areas for improvement, particularly in feature extraction and weighted feature integration. Efforts to refine the training process, including optimized sampling techniques and hyperparameter tuning, yielded incremental gains but also revealed the need for further enhancements.

This work demonstrates the importance of tailoring models and training techniques to the unique challenges of fashion recommendation systems, providing a strong starting point for future advancements in this field.

References

[1] Param Aggarwal. Fashion product images dataset, 2019.

[2] Giuseppe Cartella, Alberto Baldrati, Davide Morelli, Marcella Cornia, Marco Bertini, and Rita Cucchiara. Openfashionclip: Vision-and-language contrastive learning with open-source fashion data, 2023.

[3] Samit Chakraborty, Md. Saiful Hoque, Naimur Rahman Jeem, Manik Chandra Biswas, Deepayan Bardhan, and Edgar Lobaton. Fashion recommendation systems, models and methods: A review. *Informatics*, 8(3), 2021.

[4] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning, 2020.

[5] Jaemin Cho, Jiasen Lu, Dustin Schwenk, Hannaneh Hajishirzi, and Aniruddha Kembhavi. X-lxmert: Paint, caption and answer questions with multi-modal transformers, 2020.

[6] Eric Dodds, Jack Culpepper, Simao Herdade, Yang Zhang, and Kofi Boakye. Modality-agnostic attention fusion for visual search with text feedback, 2020.

[7] Dehong Gao, Linbo Jin, Ben Chen, Minghui Qiu, Peng Li, Yi Wei, Yi Hu, and Hao Wang. Fashionbert: Text and image matching with adaptive loss for cross-modal retrieval, 2020.

[8] Sonam Goenka, Zhaoheng Zheng, Ayush Jaiswal, Rakesh Chada, Yue Wu, Varsha Hedau, and Pradeep Natarajan. Fashionvlp: Vision language transformer for fashion retrieval with feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14105–14115, 2022.

[9] Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks, 2019.

[10] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision, 2021.

[11] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

[12] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks, 2020.

[13] Huidong Liu, Shaoyuan Xu, Jinmiao Fu, Yang Liu, Ning Xie, Chien-Chih Wang, Bryan Wang, and Yi Sun. Cma-clip: Cross-modality attention clip for image-text classification, 2021.

[14] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks, 2019.

[15] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data, 2020.

[16] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[17] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[19] Xi SHEN, Yang Xiao, Shell Xu Hu, Othman Sbai, and Mathieu Aubry. Re-ranking for image retrieval and transductive few-shot classification. In *Advances in Neural Information Processing Systems*, pages 25932–25943. Curran Associates, Inc., 2021.

[20] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations, 2020.

[21] Kashvi Taunk, Sanjukta De, Srishti Verma, and Aleena Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260, 2019.

[22] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[23] Yi Xin, Junlong Du, Qiang Wang, Ke Yan, and Shouhong Ding. Mmap : Multi-modal alignment prompt for cross-domain multi-task learning, 2023.

[24] Shuai Zheng, Fan Yang, M. Hadi Kiapour, and Robinson Piramuthu. Modanet: A large-scale street fashion dataset with polygon annotations, 2019.

[25] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa, 2019.

[26] Mingchen Zhuge, Dehong Gao, Deng-Ping Fan, Linbo Jin, Ben Chen, Haoming Zhou, Minghui Qiu, and Ling Shao. Kaleido-bert: Vision-language pre-training on fashion domain, 2021.