Graduation Thesis Final Report

Prompt-Based Fashion Outfits Retrieval System

4th Year Student Name

Nguyen Quoc Dung

Pham Hoang Nam

Dao Duy Hung

Instructor

Dr. Tran Van Ha

Bachelor of Information Technology

Hoa Lac campus - FPT University

12 December 2023

Acknowledgment

Our deepest thanks to Dr. Tran Van Ha, our thesis advisor, for invaluable guidance, feedback, and support throughout our graduation thesis journey. His knowledge, constructive criticism, and unwavering encouragement have been instrumental in shaping not only the outcome of this thesis but also our academic development. It was a privilege and honor to work under his devoted supervision.

We express our sincere appreciation to FPT University for providing us with the necessary prerequisites, infrastructure, and appropriate channels essential for the successful completion of our graduation thesis. The institution's contribution to education and research has undoubtedly ignited our aspirations. As recipients of this esteemed academic institution, we take pride in being an integral part of its scholarly legacy.

Furthermore, we formally recognize the invaluable role our cherished families and stalwart friends have played through their unrelenting care, support, and motivation. Their belief in our potential and their tireless encouragement fortified our resilience and determination to pursue our ambitions without hesitation. It is with enduring gratitude that we acknowledge how instrumental they were in emboldening us to realize our accomplishment, we wish to affirm that their contributions were vital to our success. Our achievements would not have been possible without them standing steadfastly behind us every step of the way.

Lastly, we want to express our gratitude to the participants of this study for their time and willingness to contribute valuable insights to this research.

This thesis is a testament to the collective of those mentioned above, and we are profoundly grateful for the impact each of you has had on this academic endeavor.

Abstract

The exponential growth of e-commerce in recent years has transformed the fashion industry, propelling it into a new era of digital retail. With the convenience of online shopping, consumers now have access to an extensive array of fashion products from the comfort of their homes and as a result in need of more efficient and personalized shopping experiences. This demand paved the way for the advancement of recommendation and retrieval systems in fashion e-commerce. In this thesis, we build a system plan to streamline and enhance the retrieval of fashion outfits from vast and diverse collections. Our system consists of two components, a multimodal model to retrieve image items matching a textual description, and a network incorporating hashing modules to capture high-order interactive compatibility between fashion items, facilitating efficient and personalized fashion outfit recommendations. Through extensive experimentation and evaluation, we demonstrate the effectiveness of our system in providing accurate and personalized fashion outfit recommendations with desired descriptions by the consumers, like a particular color, style, occasion, season, and many more.

***Keywords** — Fashion Retrieval, Outfit Recommendation, Representation Learning, Hashing*

Table of Contents

Chapter 1: Introduction ... 7

1.1. Overview ... 7

1.2. Related works ... 8

1.2.1. Content-based Fashion Retrieval ... 8

1.2.2. Outfit Recommendation ... 9

1.3. Motivation ... 11

1.4. Contribution ... 11

Chapter 2: Background ... 12

2.1. AlexNet Architecture ... 12

2.2. Transformer Architecture ... 12

2.3. Language Model ... 15

2.4. Vision Transformer ... 16

2.5. Multimodal Model ... 17

2.6. Fashion Hashing Network ... 18

Chapter 3: Dataset ... 21

3.1. Polyvore ... 21

3.2. Fashion32 ... 23

3.3. Preprocessing ... 24

Chapter 4: Methodology ... 25

4.1. Negative outfits generation ... 25

4.2. Items retrieval ... 25

4.3. Outfit composing ... 27

4.4. Overall framework ... 27

Chapter 5: Experiments ... 29

5.1. Evaluation metrics ... 29

5.1.1. AUC ... 29

5.1.2. NDCG ... 30

5.1.3. FITB ... 30

5.2. Benchmark ... 31

5.3. Demonstration ... 34

Chapter 6: Discussion ... 36

Chapter 7: Conclusion and Future Works ... 37

Appendix ... 41

Annotations

List of Figures

Figure 2.1. The AlexNet architecture 12
Figure 2.2. The Transformer - model architecture, from [21] 13
Figure 2.3. Multi-Head Attention, picture taken from ResearchGate 14
Figure 2.4. The GPT-2 small architecture 15
Figure 2.5. The ViT architecture 16
Figure 2.6. CLIP architecture, taken from [6] 17
Figure 3.1. Some examples of items in the Polyvore dataset 21
Figure 3.2. Metadata of the Polyvore dataset 21
Figure 3.3. Metadata of the Polyvore dataset we will use 22
Figure 3.4. Distribution of categories in the Polyvore dataset 22
Figure 3.5. Sample outfit items in CSV format table of the disjoint dataset 22
Figure 3.6. Distribution of categories in the Fashion32 dataset 23
Figure 3.7. An outfit sample in the Fashion32 dataset 23
Figure 3.8. Sample of outfit items in preprocessed CSV format table in the Fashion32 dataset. 24
Figure 4.1. FashionCLIP pipeline 25
Figure 4.2. Python-like pseudocode for the multi-modal retrieval of the CLIP model 26
Figure 4.3. Retrieve items satisfying a user prompt using the FashionCLIP model 26
Figure 4.4. Fashion hashing network (FHN) Architecture 27
Figure 4.5. A full pipeline of prompt-based outfits retrieval 28
Figure 5.1. The ROC-AUC curve, from GeeksforGeeks 30
Figure 5.2. Visualization of the FITB task, from Vo et al. 2023 31
Figure 5.3. Sample of positive and negative outfit pairs, accompanied by corresponding scores, aligned with the positive outfits' descriptions. Positive outfit scores are emphasized in green, whereas negative outfit scores are highlighted in red 33
Figure 5.4. Sample of candidate outfits from the dataset, evaluating for FITB task. Positive outfit scores consistently appear in the first row across the four candidates. The highest scores are depicted in green, whereas the others are marked in red. Missing items are indicated by a yellow-line square 34
Figure 5.5. Application showcasing retrieval demonstrations for some specific prompts 35
Figure A.1. Validation curve of some metrics while training our model 42

Annotations

List of Tables

Table 5.1. Comparison of different training methods on the Fashion32 dataset 32

Chapter 1: Introduction

1.1. Overview

The fashion industry, with its ever-evolving trends and creative expressions, is a dynamic landscape characterized by ever-changing trends, styles, and personal preferences. The field has traditionally been driven by the instincts and intuitions of designers, fashion houses, and trendsetters. However, the advent of machine learning has introduced a new dimension, one where data-driven insights and algorithms wield significant influence. This evolving relationship between technology and fashion has recently captivated the industry, representing a profound shift.

The fusion of fashion and technology holds a multifaceted allure, grounded in several compelling factors. Machine learning, a subfield of artificial intelligence (AI), possesses the extraordinary capacity to extract intricate patterns from vast datasets, making it an ideal tool for decoding the complexities of fashion. From predictive analytics that anticipate the next big trend to personalized shopping experiences that cater to individual tastes, the potential applications are manifold.

One of the most captivating developments in the fashion domain in recent times is the emergence of fashion item retrieval systems, especially in the context of composite outfits. As the number of items within each garment category increases, the potential combinations for outfits grow exponentially. Given the typically vast size of fashion inventories, the sheer magnitude of possible outfits that can be curated from these items becomes orders of magnitude greater. The task of mining fashion ensembles from an extensive inventory poses significant challenges, underscoring the necessity for intelligent fashion recommendation techniques [1]. Furthermore, the concept of employing prompts to suggest fashion apparel is relatively new in this field, particularly in the context of recommending multiple harmonious items simultaneously. Consequently, our objective was to address this challenge.

1.2. Related works

1.2.1. Content-based Fashion Retrieval

Content-based fashion image Retrieval (CBFIR) methods retrieved the desired fashion items or products from the queried reference in the form of image, text, or visual clue [2]. The predominant focus within this task revolves around the utilization of referenced images or multimodalities (*i.e.*, image and text) to retrieve desired fashion products for a user. Rubio et al. 2017 [3] leverage both the images and textual metadata and propose a joint multi-modal embedding that maps both the text and images into a common latent space, helping effectively perform retrieval in this space. They utilize a loss consisting of both the contrastive loss and the weighted sum of the cross-entropy classification losses to train both the text network and the image network. Shin et al. 2019 [4] propose a style feature extraction (SFE) layer that decomposes the clothes vector into style and category. They append the layer to the Siamese CNN and train with a loss function composed of softmax loss, contrastive loss, and center loss to predict stylish matching clothes effectively. Zhu, J. et al. 2023 [5] introduce new modules called Fine-Granular Aggregator and Attention-based Token Alignment to exploit both the overall and detailed characteristics of clothing images.

In recent times, contrastive learning has emerged as a prominent method for acquiring meaningful representations of concepts within the field of machine learning. This approach is grounded in the notion that concepts with semantic connections (for instance, two images of the same object captured from different angles) should exhibit similar representations, whereas unrelated concepts should be distinctly represented. Radford, A. et al. 2021 [6] introduced CLIP which represents a multimodal neural network for vision and language, trained using contrastive learning to establish associations between visual concepts and text. The model consists of separate encoders for vision and text, each followed by a linear layer that projects the image and text representations into the same latent space.

The goal of CLIP is to position images and corresponding descriptions (like an image of a red shirt and the description "a red shirt") close together in vector space. Specific to the fashion industry, Chia et al. 2019 [7] trained their CLIP model on a fashion dataset containing 800k products. The model, called FashionCLIP, is shown to learn general concepts to be transferable across tasks in the domain. We leverage this model to retrieve fashion items from a textual description.

1.2.2. Outfit Recommendation

In recent years, there has been rising enthusiasm for the creation of intelligent fashion recommendation systems. These systems aim to assist users in finding and buying clothing and accessories that align with their styles. Given the vast array of outfits created from a diverse selection of fashion items, there is a heightened focus on personalized outfit recommendations. This involves suggesting outfits that cater to users' styles and align with their specific preferences. The increasing interest in this personalized approach underscores its growing importance in the fashion industry. This section provides the work that has been done on this problem.

The earliest approach to date is the use of a functional tensor factorization method to model the interactions between user and fashion items by Hu et al. 2015 [8]. They use gradient boosting together with a learning-to-rank formulation to optimize the model. However, their model is still limited since they did not use a deep learning approach. Another approach is to use the pairwise model compatibility between fashion items [9] [10] using the Siamese network or triplet loss. However, these methods lack the incorporation of the outfit's textual semantics into the whole training pipeline. The closest to our method is the work of Han et al. 2017 [11]. They utilize the BiLSTM network to model the compatibility of the outfit as a whole and propose to learn jointly with visual-semantic embedding for multimodal input. However, the BiLSTM is hard to scale in terms of training efficiency. Lai et al. 2020 [12] ‘s work is the first to model outfit compatibility conditional on a theme. They introduce a category-specific mask into the triplet embedding training process and finally train the network with theme classification loss. For outfit generation, it’s questionable how they could generate outfits matching a theme from a database in an efficient manner. Lin et al. 1970 [13] propose a novel approach that learns a category-based subspace attention network. This network takes the source image, its category, and a target category as input to generate a subspace embedding and then is trained to widen the distance between the outfit and negative samples and the distance between the outfit and positive samples by a margin. Overall, none of these works incorporate the prompt from the user as a textual description of the outfit to retrieve it from a database.

Some works utilize hashing techniques that learn data-driven binary codes. These techniques have become popular for enabling efficient similarity search in large-scale multimedia retrieval tasks. The aim is to maintain the nearest neighbor relation of the original space in the hamming space. The basic idea is to preserve the similarity, *i.e.*, to minimize the gap between the similarity computed in hash-coded space and the similarity in the original space. Many methods have been introduced by learning real-valued embedding and then taking the sign of the values to obtain binary codes. Due to the huge amount of fashion items, efficiency becomes an extremely important problem within a practical recommendation system. Learning to hash has been extensively studied for efficient image retrieval [14]. This network models outfit compatibility through pairwise interactions and employs the weighted hashing technique [15] [16] [17] for matching users and items. Lu, Zhi, et al. 2019 [1] introduce the Fashion Hashing Network (FHN) which models the pairwise interaction relations with the hashing technique, a method comparable to [8]. During inference, the model is required to generate all outfit samples to compute the score for each, aiming to identify the outfit with the highest score. To retrieve garments for the hashing network based on a textual description, an additional model is required. However, there is a potential challenge where items generated by the hashing network may not accurately match the outfit description. Our research is focused on investigating and addressing this issue.

1.3. Motivation

Lately, there has been a rising interest in the Multimodal system, particularly in the field of Text-Visual generation. In this context, the AI model is tasked with producing visual instances that best align with a given textual description, with generative and retrieval methods being the most popular approaches [6] [18] [19]. Our approach aims to elevate the capabilities of current AI Chatbots utilized in the Fashion Industry. Through our efforts, the Chatbot system has evolved into an Outfit Stylist, proficient in sourcing not just standalone items from diverse fashion databases but also assembling harmonious garments into stylistic ensembles tailored to the preferences of end users. Our work has the potential to streamline the fashion shopping experience, allowing customers and enthusiasts to effortlessly procure their desired outfits from the convenience of their homes.

1.4. Contribution

In our thesis, we develop a full pipeline for a fashion outfit retrieval system based on a user prompt. The primary contribution of our work is outlined as follows:

- Create a model for recommending fashion outfits based on textual prompts.
- Conduct experiments and demonstrations to assess the effectiveness and efficacy of our proposed approach.

Our research marks the pioneering exploration of prompt-based systems for retrieving multiple interactive visual instances. This work establishes a novel research direction within the AI-Fashion domain.

Chapter 2: Background

2.1. AlexNet Architecture

AlexNet [20] was the first Convolutional Neural Network architecture to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. The AlexNet architecture is shown in **Figure 2.1**. It consists of 5 convolution layers followed by 3 max-pooling layers, 2 normalization layers, 2 fully connected layers, and finally a softmax layer. AlexNet is recognized as one of the most impactful papers in the field of computer vision. It has inspired numerous subsequent publications that utilize Convolutional Neural Networks (CNNs) and Graphics Processing Units (GPUs) to expedite the progress of deep learning.

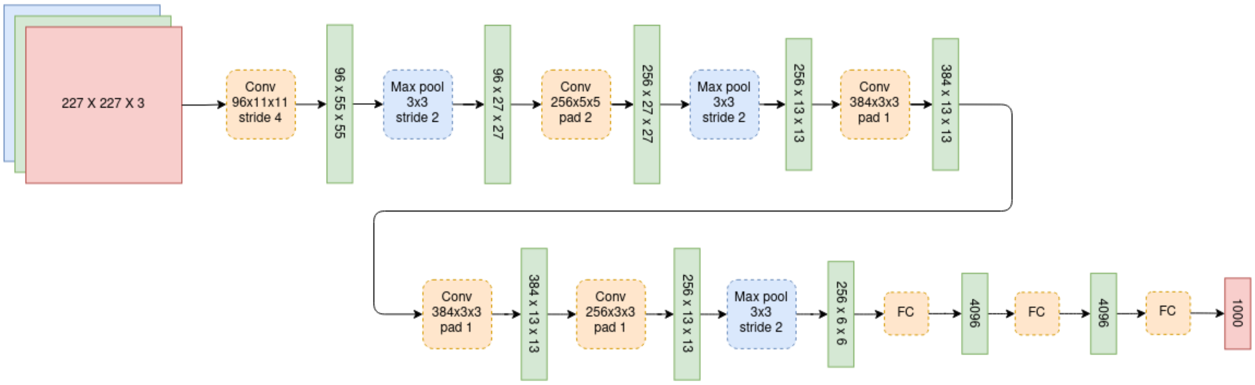


Figure 2.1. *The AlexNet architecture*

2.2. Transformer Architecture

The Transformer architecture is a type of neural network architecture introduced in the paper "Attention is All You Need" by Vaswani et al. (2017) [21]. It has since become a foundational model for a wide range of natural language processing (NLP) and other machine-learning tasks.

The key innovation of the Transformer architecture is the self-attention mechanism, which allows the model to weigh the importance of different words in a sequence when making predictions. This is in contrast to traditional recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) that process input sequences sequentially.

The Transformer model includes 2 phases: Encoder and Decoder.

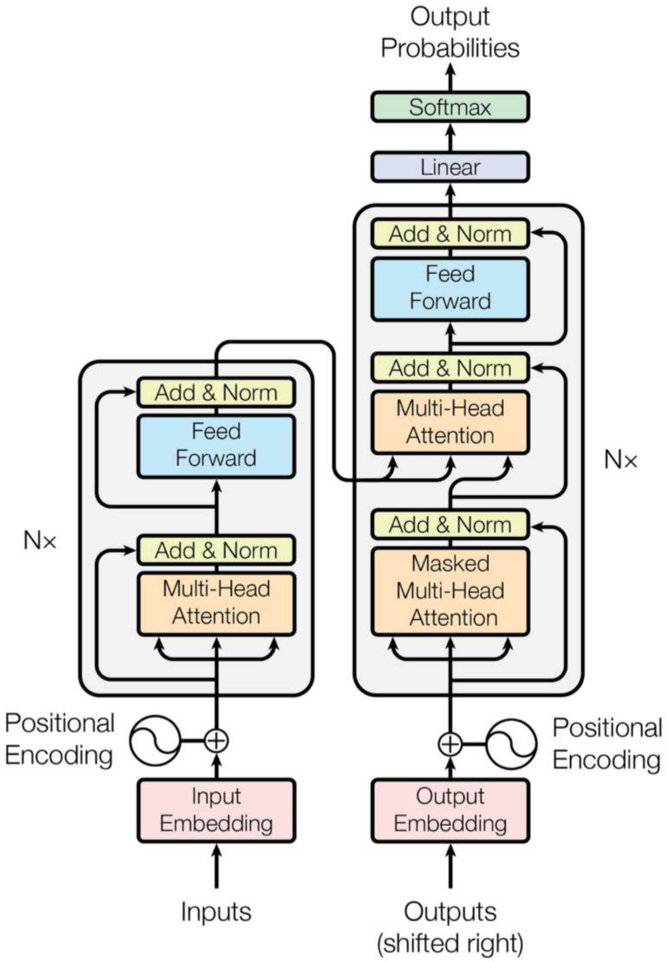


Figure 2.2. *The Transformer - model architecture, from [21]*

The first part of the model is the encoder. It takes an input sequence and transforms it into a sequence of encoded representations. It consists of multiple layers, each containing a multi-head self-attention mechanism and a position-wise feed-forward neural network. The self-attention mechanism allows the model to capture dependencies between different words in the input sequence.

The decoder takes the encoded sequence and generates an output sequence. Like the encoder, it consists of multiple layers with self-attention and feed-forward components. Additionally, it includes an additional attention mechanism called encoder-decoder attention, which helps the decoder focus on relevant parts of the encoded input.

The central component throughout the architecture is the Multi-Head Attention. The multi-head attention mechanism enhances the model's ability to capture different types of dependencies. It consists of multiple attention heads that operate in parallel, allowing the model to focus on different parts of the input sequence simultaneously. As the name suggests, each head utilizes the attention mechanism that allows the model to assign different weights to different parts of the input sequence when processing a particular element. It computes attention scores between each pair of positions in the input sequence and uses these scores to weigh the importance of different elements. This mechanism enables the model to capture long-range dependencies and improves its ability to understand context.

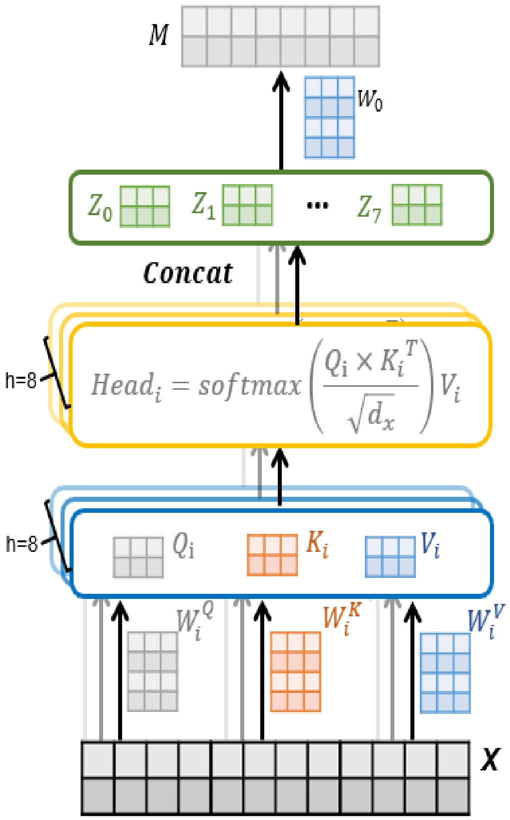


Figure 2.3. *Multi-Head Attention, picture taken from ResearchGate*[1]

[1] https://www.researchgate.net/publication/334427742_Stock_Volatility_Prediction_Based_on_Self-attention_Networks_with_Social_Information/figures?lo=1

2.3. Language Model

The advent of the transformer architecture has heralded a significant leap forward in the realm of natural language processing. The Generative Pre-trained Transformer (GPT) models, a noteworthy instantiation of this architecture [22] [23], have played a pivotal role in pushing the boundaries of state-of-the-art (SOTA) language models. The model consists of multiple layers of attention and feedforward mechanisms, allowing it to understand and generate complex sequences of data, such as language. The GPT architecture serves as the foundational framework for a contemporary class of deep learning models denominated Large Language Models (LLMs), which are widely popular these days.

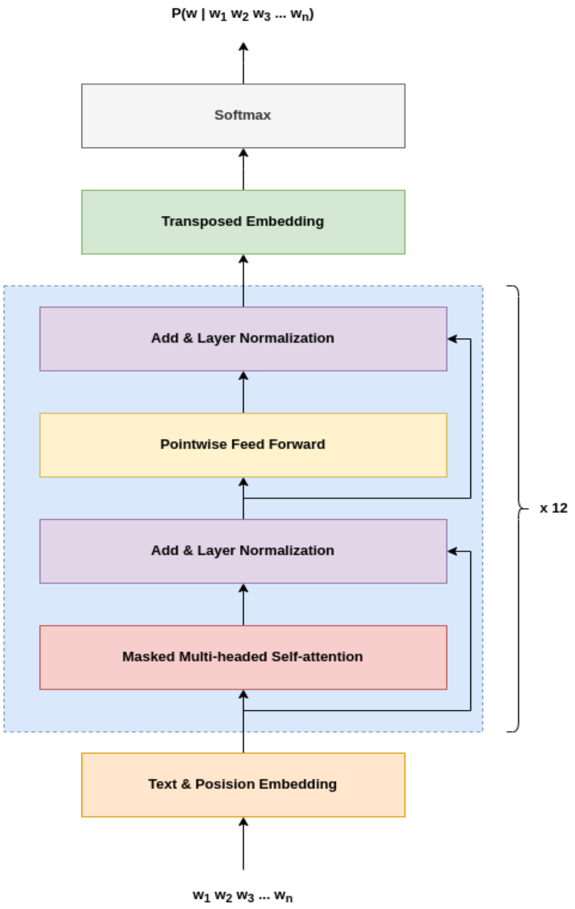

$P(w \mid w_1 w_2 w_3 \dots w_n)$
Softmax
Transposed Embedding
Add & Layer Normalization
Pointwise Feed Forward
Add & Layer Normalization
Masked Multi-headed Self-attention
x 12
Text & Posision Embedding
$w_1 w_2 w_3 \dots w_n$

Figure 2.4. *The GPT-2 small architecture*

2.4. Vision Transformer

The Transformer architecture has been very successful in natural language processing, but it has not seen the same dominance in computer vision. However, in 2021, researchers from Google Brain proposed a novel neural network that adapts the Transformer design for computer vision tasks. This architecture is known as the Vision Transformer (ViT) [24]. Unlike previous SOTA computer vision models that rely heavily on convolutional neural networks (CNNs), the ViT is composed purely of Transformer encoder blocks. When trained on large amounts of data, the ViT model rivals or exceeds the performance of CNN models trained on the same amount of data.

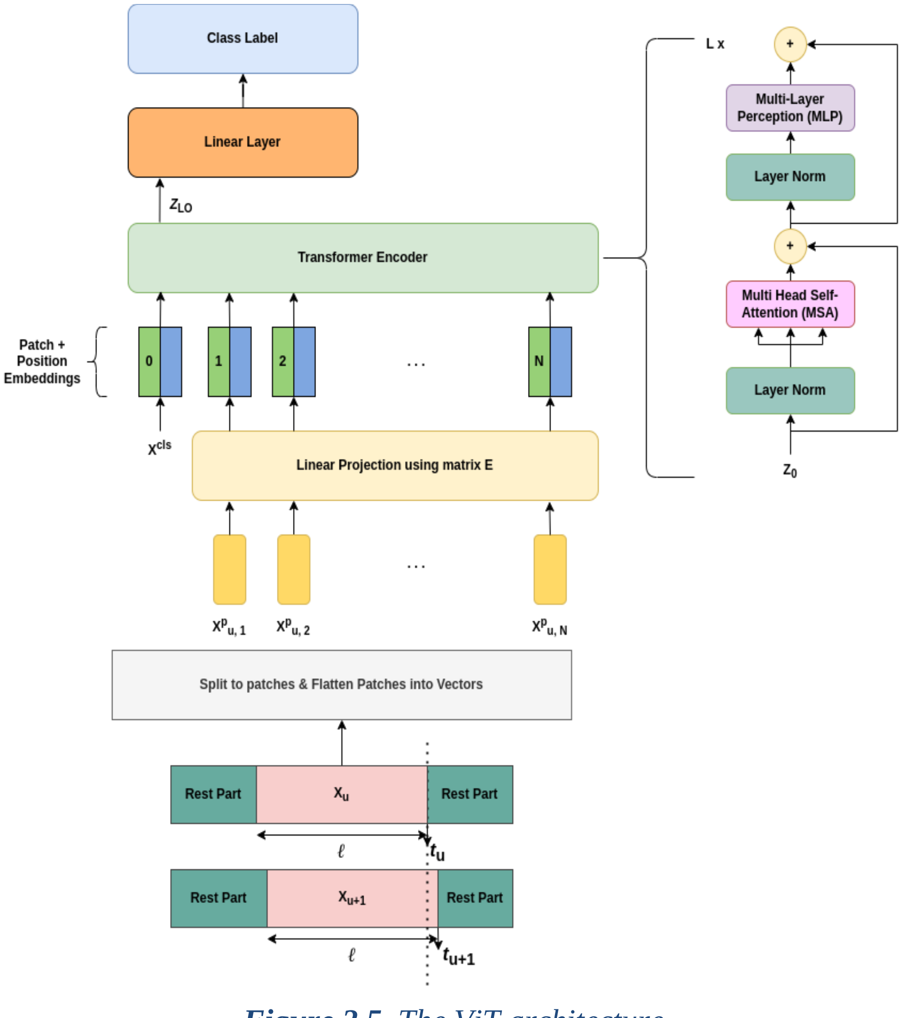


Figure 2.5. *The ViT architecture*

2.5. Multimodal Model

CLIP (Contrastive Language-Image Pre-training) [6] marked a significant milestone as the initial model capable of applying zero- and few-shot learning to various image classification tasks. It constitutes a multi-modal framework, designed through a training process that involves the maximization of cosine similarity scores for pairs comprising matching images and textual descriptions. Specifically, when provided with N pairs of images and texts, CLIP concurrently embeds all N images and N texts using their respective encoders. Subsequently, it computes the dot product of these embeddings to construct an $N \times N$ matrix, where each entry corresponds to a pair among the N images and N texts. Notably, the diagonal entries of this matrix, which represent the genuine N pairs, are maximized, while conversely, the non-diagonal entries, denoting spurious pairings among $N^2 - N$ possibilities, are minimized. This training methodology ensures the model's proficiency in associating images with their relevant textual descriptions.

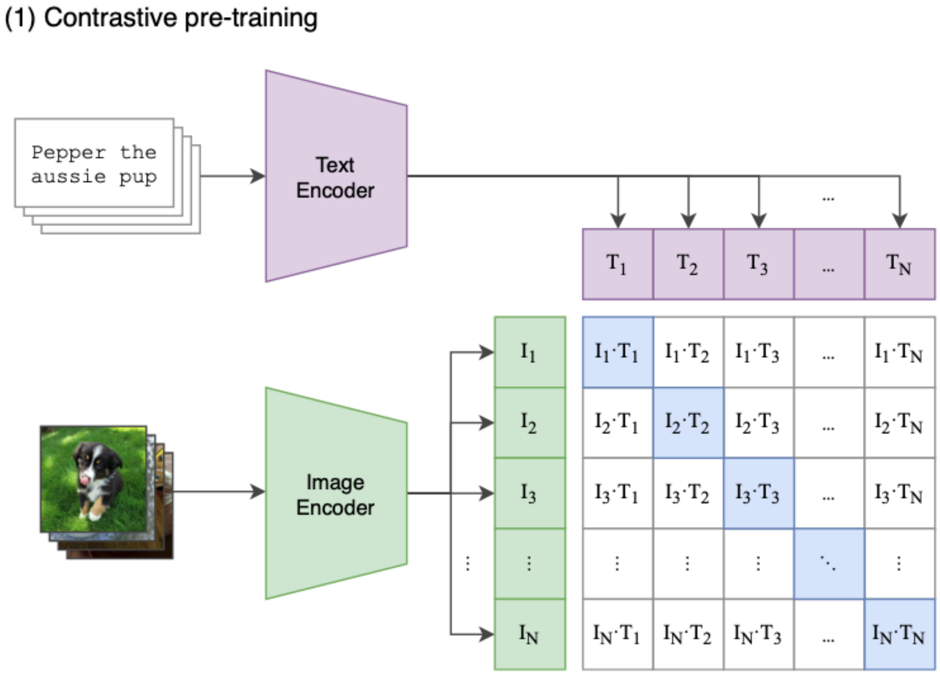

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T1 T2 T3 ... TN
I1 I2 I3 ⋮ IN
I1·T1 I1·T2 I1·T3 ... I1·TN
I2·T1 I2·T2 I2·T3 ... I2·TN
I3·T1 I3·T2 I3·T3 ... I3·TN
IN·T1 IN·T2 IN·T3 ... IN·TN

Figure 2.6. CLIP architecture, taken from [6]

2.6. Fashion Hashing Network

In this thesis we employ the Fashion Hashing Network (FHN) architecture from [1] to model the interactive compatibility between fashion garments. The author suggests that the most successful way to model high-order relationships is to decompose them into pairwise relationships. This section defines the theoretical formulation from the original paper with some modifications suited to our problem.

Suppose there are N fashion categories (*e.g.* top, bottom, shoes, jeweler, …). The number of items in the *n-th* category is donated by L_n. Let

$$X^{(n)} = \left\{x_1^{(n)}, x_2^{(n)}, \dots, x_{L_n}^{(n)}\right\} \tag{2.1}$$

denotes all items in the *n-th* category, where x_i^n is the *i-th* item in it. Then an outfit with N items, with each from one category, can be represented as

$$O_i = \left\{x_{i_1}^{(1)}, x_{i_2}^{(2)}, \dots, x_{i_N}^{(N)}\right\} \tag{2.2}$$

where $i = (i_i, \dots, i_N)$ is the index tuple.

Rather than incorporating user preferences in the original work, we use r_{t,O_i} to indicate the compatibility of a textual description to outfit O_i. The higher the score, the more matching the text to the outfit. Our task is to predict r_{t,O_i} for each description-outfit pair so that the most suitable outfits for a prompt are recommended to the user.

Due to the extremely large number of outfit pairs, the binary embedding technique is used. The binary codes are obtained by taking the signs of the continuous variables, *i.e.*

$$b_{i_n}^n = sign\left(h_{i_n}^n\right); b_t^t = sign(h_t^t) \tag{2.3}$$

where the function $sign(x)$ equals 1 when x is greater than or equal to 0 and equals -1 in other cases.

The model was built based on pairwise interaction relations for the efficiency of computing the preference scores and retrieving compatible outfits for recommendation.

Let $b_i, b_j \in \{-1, +1\}^D$ be the binary codes of two fashion objects, their compatibility is measured by:

$$m_{ij} = b_i^T \Lambda b_j \tag{2.4}$$

where Λ is a weighting matrix that is constrained to be diagonal.

The score for outfit O_i concerning prompt t is computed by:

$$r_{t,O_i} = \alpha \cdot r_{t,O_i}^{(t)} + r_{O_i}^{(i)} \tag{2.5}$$

where:

$$r_{t,O_i}^{(t)} = \frac{1}{z} \sum_n b_{i_n}^{(n)T} \Lambda^{(t)} b_t^{(t)} \tag{2.6}$$

$$r_{O_i}^{(i)} = \frac{1}{z} \sum_n \sum_m b_{i_n}^{(n)T} \Lambda^{(i)} b_{i_m}^{(m)} \tag{2.7}$$

with $\Lambda^{(t)}$ and $\Lambda^{(i)}$ are the weighting matrices for prompt-item and item-item pairs respectively; $b_t^{(t)}$ is the embedding of the textual description for the whole outfit. The scalar α is used to balance the contributions of the two terms.

Fashion items are usually depicted by some textual description, so the hashing model also tries to extract some features from them in our model.

Suppose binary codes from different modalities are donated by $b_{v,i_n}^{(n)}, b_{v,i_n}^{(n)}$, where v and f indicate visual and textual respectively.

The overall score with multi-modality information can be computed by:

$$r_{t,O_i}\left(\left\{b_{v,i_n}^{(n)}\right\}, b_t^{(t)}\right) + r_{t,O_i}\left(\left\{b_{f,i_n}^{(n)}\right\}, b_t^{(t)}\right) \tag{2.8}$$

The training set contains a set of outfit pairs:

$$\mathcal{P} \equiv \left\{(t, i, j) | r_{t,O_i} > r_{t,O_j}\right\} \tag{2.9}$$

where r_{t,O_i} and r_{t,O_j} are the score of a positive and negative outfit corresponding to a textual description respectively. A positive outfit means that the outfit matches the description while the reverse is true for a negative one.

Using the BPR [25] optimization criterion, the objective function is:

$$\mathcal{L}_{\mathcal{BPR}} = \sum_{(t,i,j)\epsilon\mathcal{P}} log\left(1 + exp\left(-\left(r_{t,O_i} - r_{t,O_j}\right)\right)\right) \tag{2.10}$$

Also, the objective function adds constraints to make the embeddings of visual and textual information more consistent with each other by adding the following loss:

$$\begin{aligned} \mathcal{L}_{\mathcal{VSE}} = &\sum_{v,k} max\{0, c - s(v, f) + s(v, f_k)\} \\ &+ \sum_{f,k} max\{0, c - s(v, f) + s(v_k, f)\} \end{aligned} \tag{2.11}$$

where (v, f) are binary codes for items from the two modalities. (v, f) are for the same item. (v, f_k) are for different items and so is (v_k, f). The similarity is denoted as $s(v, f) = v^T f$.

Overall, the objective function is:

$$\min_{\theta} \mathbb{E}(\mathcal{L}_{\mathcal{BPR}} + \lambda\mathcal{L}_{\mathcal{VSE}}) \tag{2.12}$$

Chapter 3: Dataset

3.1. Polyvore

The Polyvore dataset provides a large-scale corpus for research on fashion outfit composition. It contains over one million user-created outfits compiled from Polyvore, a popular fashion community website. Each outfit includes fashion items of different categories such as tops, bottoms, and shoes that Polyvore users put together. The dataset includes rich item metadata such as product images, descriptions, brands, categories, and user engagement statistics. Since its release, the Polyvore dataset has facilitated research on outfit compatibility learning and fashion recommendation systems. However, the dataset also presents challenges for evaluation due to its inherent biases, such as imbalance among categories and brands. Still, the size and diversity of the Polyvore dataset make it a valuable resource for data-driven fashion intelligence research.

We use a subset of the Polyvore dataset, which contains about 261k images of items with their metadata. We only use the images and the category of items in this dataset.

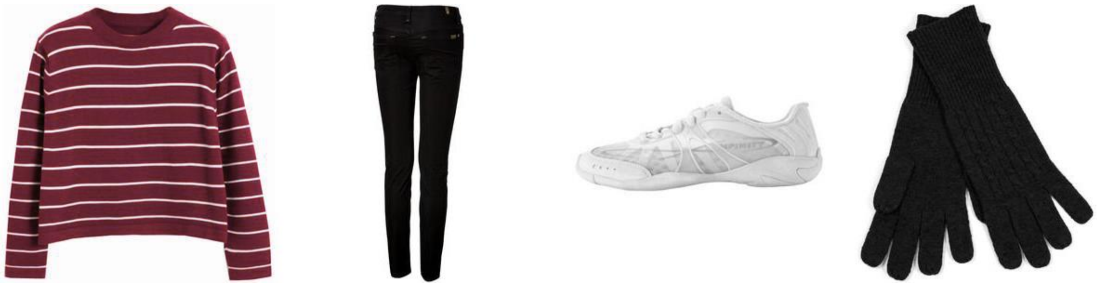
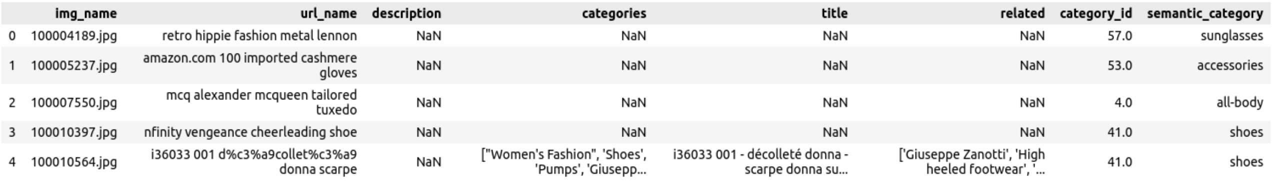
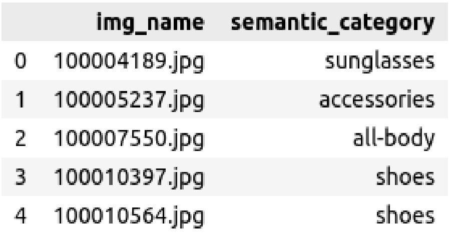


Figure 3.1. Some examples of items in the Polyvore dataset



	img_name	url_name	description	categories	title	related	category_id	semantic_category
0	100004189.jpg	retro hippie fashion metal lennon	NaN	NaN	NaN	NaN	57.0	sunglasses
1	100005237.jpg	amazon.com 100 imported cashmere gloves	NaN	NaN	NaN	NaN	53.0	accessories
2	100007550.jpg	mcq alexander mcqueen tailored tuxedo	NaN	NaN	NaN	NaN	4.0	all-body
3	100010397.jpg	nfinity vengeance cheerleading shoe	NaN	NaN	NaN	NaN	41.0	shoes
4	100010564.jpg	i36033 001 d%c3%a9collet%c3%a9 donna scarpe	NaN	["Women's Fashion", 'Shoes', 'Pumps', 'Giusepp...	i36033 001 - décolleté donna - scarpe donna su...	['Giuseppe Zanotti', 'High heeled footwear', '...	41.0	shoes

Figure 3.2. Metadata of the Polyvore dataset



	img_name	semantic_category
0	100004189.jpg	sunglasses
1	100005237.jpg	accessories
2	100007550.jpg	all-body
3	100010397.jpg	shoes
4	100010564.jpg	shoes

Figure 3.3. *Metadata of the Polyvore dataset we will use*

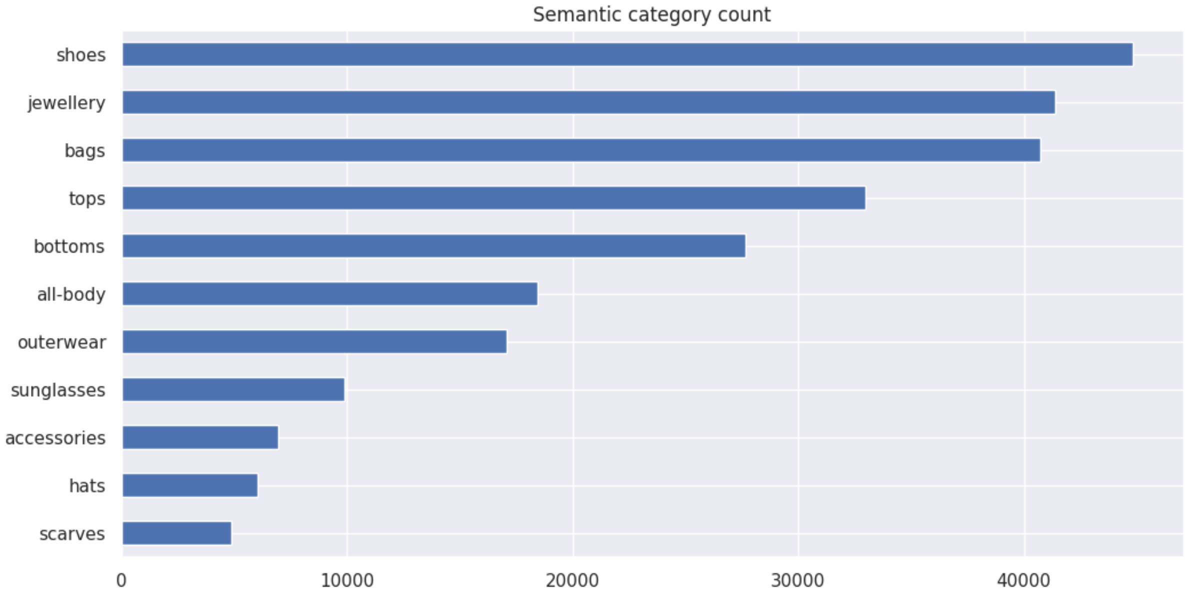
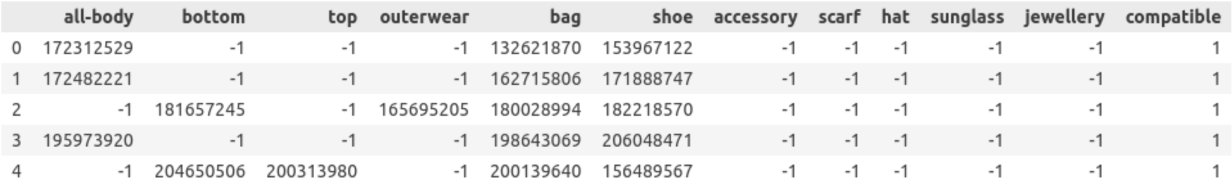


Figure 3.4. *Distribution of categories in the Polyvore dataset*

These images are combined into outfits which are classified into 2 outfit datasets: disjoint and nondisjoint. Disjoint dataset contains mutually exclusive categories, ensuring that each item belongs to only one class, simplifying the training and evaluation processes for machine learning models. On the other hand, a nondisjoint dataset permits instances to belong to multiple categories simultaneously, reflecting the real-world ambiguity in fashion categorization. In this thesis, we only focus on the disjoint dataset, containing about 62k sets of outfits.



	all-body	bottom	top	outerwear	bag	shoe	accessory	scarf	hat	sunglass	jewellery	compatible
0	172312529	-1	-1	-1	132621870	153967122	-1	-1	-1	-1	-1	1
1	172482221	-1	-1	-1	162715806	171888747	-1	-1	-1	-1	-1	1
2	-1	181657245	-1	165695205	180028994	182218570	-1	-1	-1	-1	-1	1
3	195973920	-1	-1	-1	198643069	206048471	-1	-1	-1	-1	-1	1
4	-1	204650506	200313980	-1	200139640	156489567	-1	-1	-1	-1	-1	1

Figure 3.5. *Sample outfit items in CSV format table of the disjoint dataset*

3.2. Fashion32

The primary training data for our model is sourced from the Fashion32 dataset [12]. To the best of our knowledge, this dataset is distinctive as it provides comprehensive descriptions along with diverse theme tags for each outfit and fashion item. The 32 themes are categorized into four groups: occasion, style, fit, and gender. Typically, each outfit consists of 2 to 3 items, accompanied by at least 4 images featuring a model showcasing these fashion items. The dataset has about 14k outfits with 41k fashion garments in total.

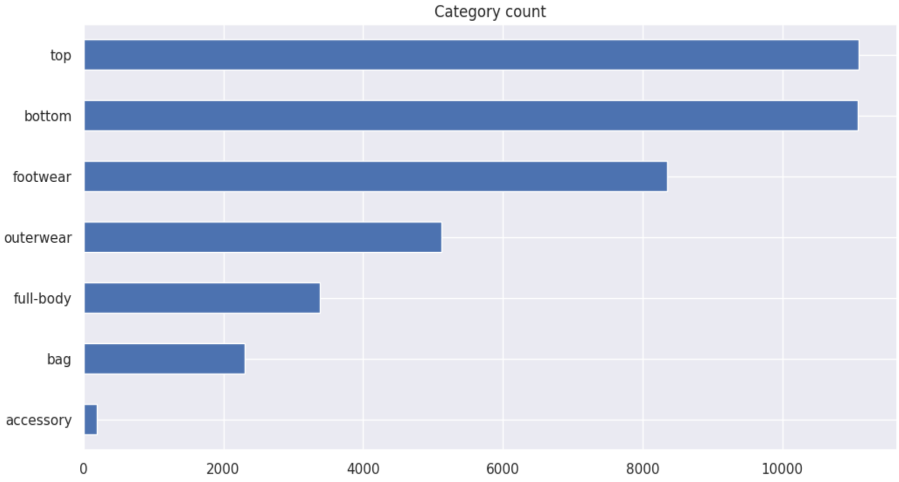


Figure 3.6. Distribution of categories in the Fashion32 dataset

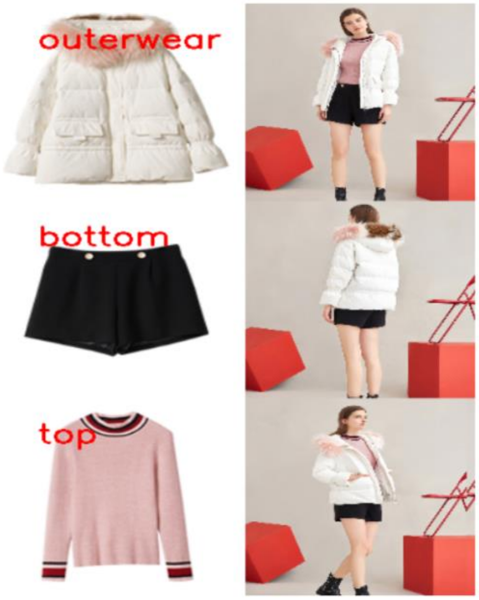
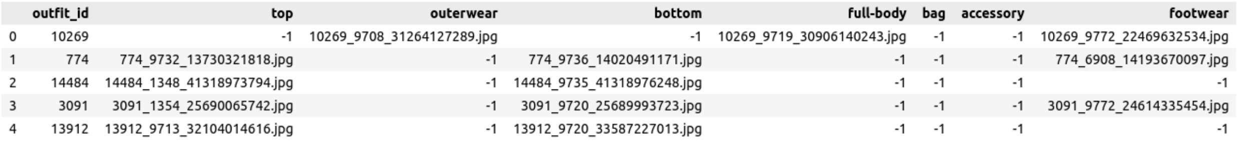


Figure 3.7. An outfit sample in the Fashion32 dataset



	outfit_id	top	outerwear	bottom	full-body	bag	accessory	footwear
0	10269	-1	10269_9708_31264127289.jpg	-1	10269_9719_30906140243.jpg	-1	-1	10269_9772_22469632534.jpg
1	774	774_9732_13730321818.jpg	-1	774_9736_14020491171.jpg	-1	-1	-1	774_6908_14193670097.jpg
2	14484	14484_1348_41318973794.jpg	-1	14484_9735_41318976248.jpg	-1	-1	-1	-1
3	3091	3091_1354_25690065742.jpg	-1	3091_9720_25689993723.jpg	-1	-1	-1	3091_9772_24614335454.jpg
4	13912	13912_9713_32104014616.jpg	-1	13912_9720_33587227013.jpg	-1	-1	-1	-1

Figure 3.8. *Sample of outfit items in preprocessed CSV format table in the Fashion32 dataset*

3.3. Preprocessing

In the Polyvore dataset, items are classified into eleven groups: all-body, bottom, top, outerwear, bag, shoe, accessory, scarf, hat, sunglass, and jewelry. They are combined into outfits based on the metadata and stored in Comma-separated value (CSV) format table files, where each row matches with an outfit along with its various items and the compatible attribute which determines if the outfit is well-matched.

In handling the Fashion32 dataset, since the outfit descriptions and diverse tags are in Chinese, we employ the Google Translate API to convert these texts into English. Subsequently, based on certain tags for each outfit item, we classify them into seven groups: top, outerwear, bottom, full-body, bag, accessory, and footwear. The resulting outfit items are then stored in CSV files, where each row corresponds to an outfit along with its various items. In cases where a category is missing, it is marked as -1. Steps are the same with the Polyvore dataset.

We limit our experimentation to five categories - top, outerwear, bottom, bag, and footwear - since these represent the most utilized categories in outfit composition.

Chapter 4: Methodology

4.1. Negative outfits generation

During the training phase, each outfit is divided into two instances: one paired with the detailed description and the other with the concatenation of the remaining 4 theme tags. The outfits from the dataset are labeled as positive. For each positive outfit, we randomly select items from the dataset to construct an incompatible outfit, labeled as a negative outfit, ensuring it does not match the unique description of the positive outfit. In the baseline method, the textual description is not incorporated into the training pipeline. We randomly choose fashion garments dissimilar to the positive outfit to compose a negative one.

4.2. Items retrieval

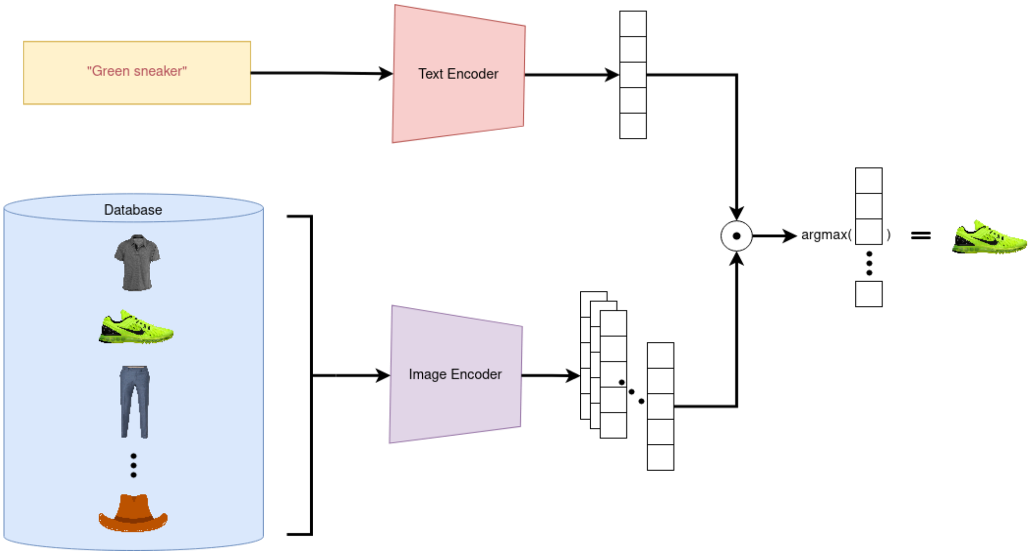
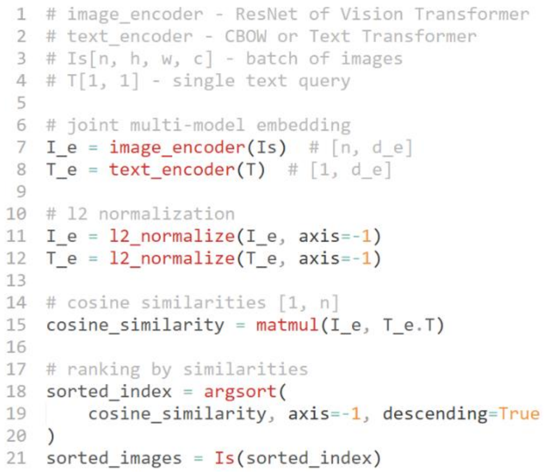


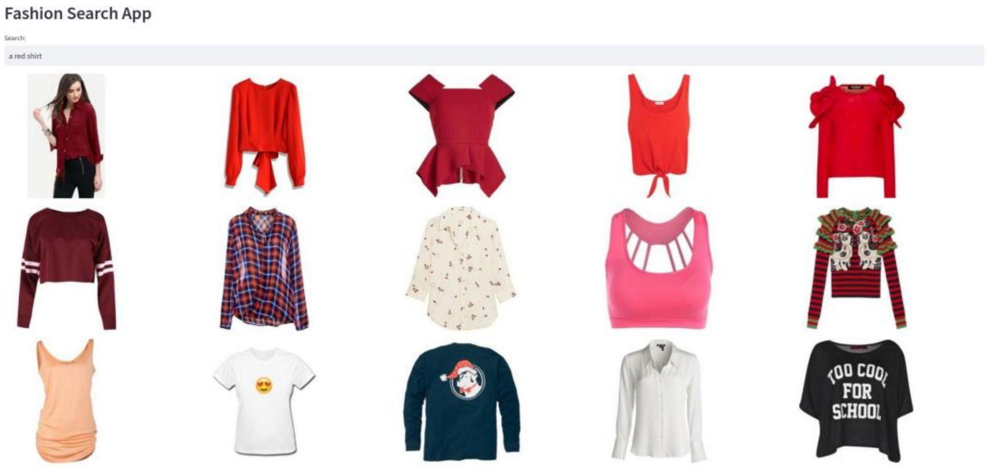
Figure 4.1. FashionCLIP pipeline

We utilize the FashionCLIP [7] model to retrieve items before composing an outfit. **Figure 4.1** illustrates the architectural framework of the model. Like the CLIP model in [6], the FashionCLIP model can be delineated into two distinct phases. In the initial phase, the image encoder undertakes the task of mapping all the garment images contained within the database into a vector space characterized by a dimensionality of 512. Subsequently, these resulting vectors are persistently stored within the database. In the second phase, when a user submits a query, the text encoder proceeds to project the query into a vector sharing the same dimensional characteristics as the image embedding vector. The prompt embedding vector is then subjected to a dot product operation with all the image embedding vectors, thereby facilitating the identification of the most compatible garment. The FashionCLIP model uses Transformers [21] with the architecture modifications described in [23] as the text encoder. The image encoder is a variant of the Vision Transformer (ViT) model [24]. The process can be summarized by pseudocode in **Figure 4.2**. Finally, the accompanying outfit description is projected into an embedding vector with dimension 512 through the text encoder of the model for the second phase. The sample demonstration of the item retrieval from a user prompt using this model can be visualized in **Figure 4.3**.



```
# image_encoder - ResNet of Vision Transformer
# text_encoder - CBOW or Text Transformer
# Is[n, h, w, c] - batch of images
# T[1, 1] - single text query

# joint multi-model embedding
I_e = image_encoder(Is)  # [n, d_e]
T_e = text_encoder(T)  # [1, d_e]

# l2 normalization
I_e = l2_normalize(I_e, axis=-1)
T_e = l2_normalize(T_e, axis=-1)

# cosine similarities [1, n]
cosine_similarity = matmul(I_e, T_e.T)

# ranking by similarities
sorted_index = argsort(
    cosine_similarity, axis=-1, descending=True
)
sorted_images = Is(sorted_index)
```

Figure 4.2. *Python-like pseudocode for the multi-modal retrieval of the CLIP model*



Fashion Search App

Search:

a red shirt

Figure 4.3. *Retrieve items satisfying a user prompt using the FashionCLIP model*

4.3. Outfit composing

The architecture of the hashing fashion model is illustrated in **Figure 4.4**, comprising three key components: a feature network for feature extraction, multiple type-dependent hashing modules that learn binary codes, and a matching block for predicting preference scores. For simplicity, we experimented with only one shared visual encoder for five categories. The textual embedding of outfits is directed through a Textual Encoder block, which is a stack of fully connected layers, then through a hashing layer similar to the visual one, contributing to the matching block and influencing the final score calculation.

To extract image features, we employ AlexNet as the feature network. Optionally, textual information can also be integrated. Distinct categories are treated as different types, and the hashing modules involve fully connected layers employing a sign function for binarization. The matching block computes the final score, encompassing two terms: one assessing interactive compatibility among items and the other considering textual semantic compatibility within the outfit.

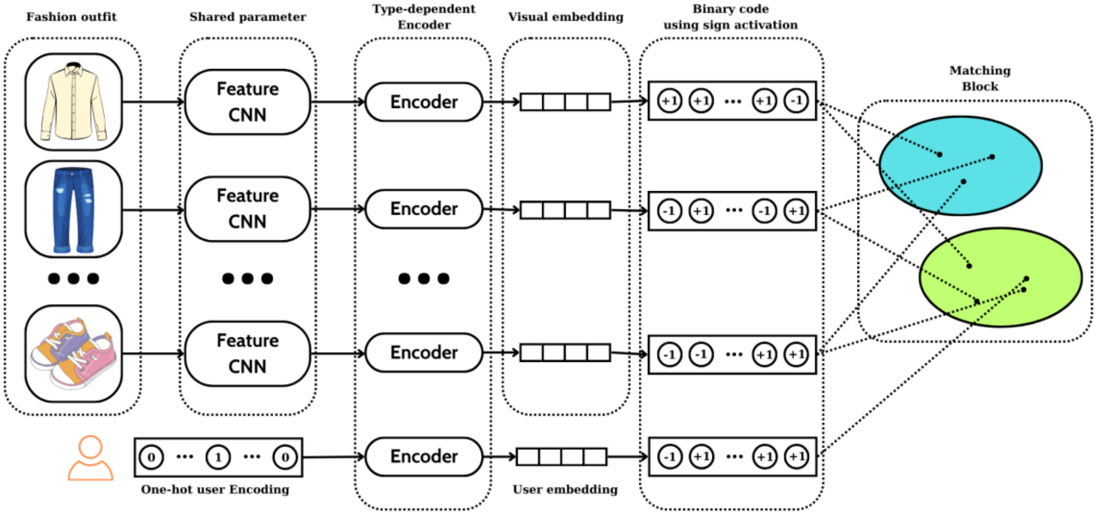


Figure 4.4. *Fashion hashing network (FHN) Architecture*

4.4. Overall framework

Our pipeline operates in the following manner: when presented with a textual prompt from a user, we utilize the FashionCLIP model to retrieve the top fashion products corresponding to the given prompt for each of the five categories. These retrieved images form a compact database, and hence the hashing network model employs a recursive technique for swift outfit composition using these items. Upon presenting these outfit queries, the matching block is responsible for computing the scores of each outfit, enabling the presentation of the top-scoring outfits to the user. The full pipeline is displayed in **Figure 4.5**.

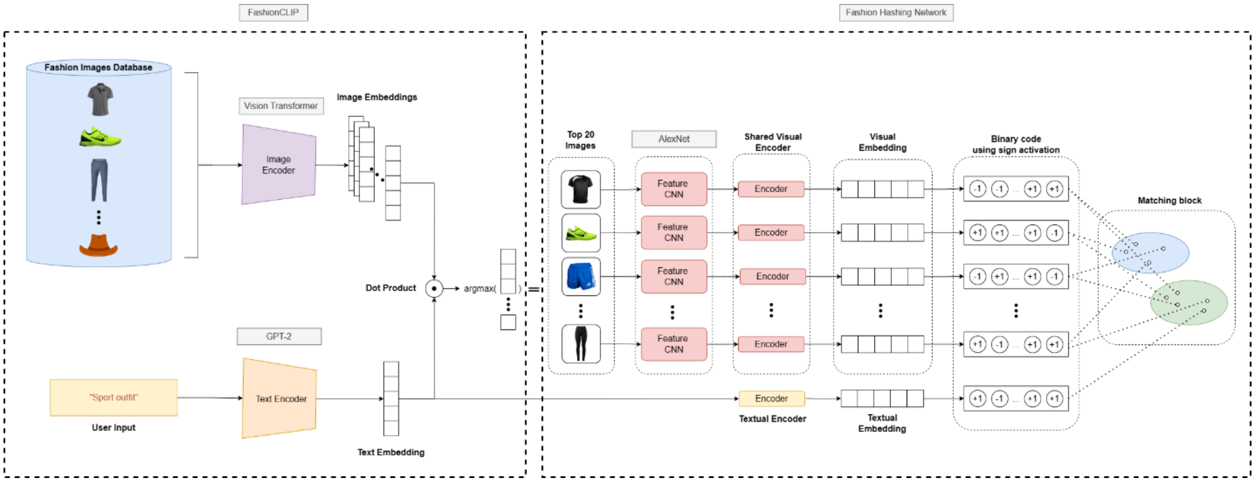


Figure 4.5. *A full pipeline of prompt-based outfits retrieval*

Chapter 5: Experiments

5.1. Evaluation metrics

We engage in experiments encompassing two recommendation tasks. The first involves outfit recommendation, wherein we rank the testing outfits in descending order of their compatibility scores. Evaluation of ranking performance is conducted using metrics such as Area Under the ROC curve (AUC) and Normalized Discounted Cumulative Gain (NDCG). The second task focuses on fill-in-the-blank (FITB) fashion recommendation experiments. The objective is to select an item from a set of candidate items (four in our experiments) that is not only highly compatible with the remaining items of the outfit but also aligns with the provided description.

5.1.1. AUC

To understand the AUC score, we need to define the Receive Operator Characteristic (ROC) curve. From a confusion matrix of a binary classifier, some important metrics are derived:

The True Positive Rate (TPR), also known as sensitivity or recall, measures the proportion of actual positive instances correctly identified by the model:

$$TPR = \frac{TP}{TP + FN} \tag{5.1}$$

The False Positive Rate (FPR) represents the proportion of actual negative instances incorrectly classified as positive:

$$FPR = \frac{FP}{FP + TN} \tag{5.2}$$

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the relationship between the True Positive Rate (TPR) and the False Positive Rate (FPR) across a range of threshold values. The Area Under the Curve (AUC), denoting the area beneath the ROC curve, functions as a quantitative metric to assess the model's overall discriminative capacity between classes. A model with a higher

AUC score is considered more effective at distinguishing between positive and negative instances. A perfect classifier would have an AUC score of 1.0, while a random classifier would yield an AUC score of 0.5.

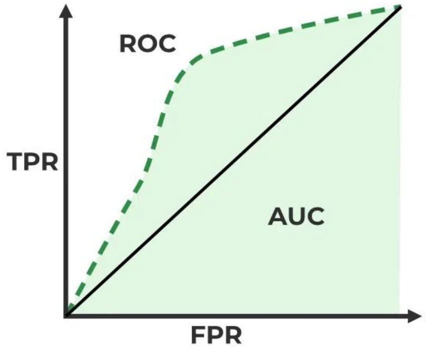


__Figure 5.1.__ The ROC-AUC curve, from GeeksforGeeks[2]

5.1.2. NDCG

NDCG, short for Normalized Discounted Cumulative Gain, is a popular measure for ranking quality and is used to evaluate the performance of search engines, recommendations, or other information retrieval systems. Following the definition in [8], letting π_i be the evaluated order of the rank, the formula for the NDCG at the *m-th* position is:

$$NDCG@m = (N_m)^{-1} \sum_{i=1}^{m} \frac{2^{y_{\pi'(i)}} - 1}{\log_2\big(\max(2, i)\big)} \tag{5.3}$$

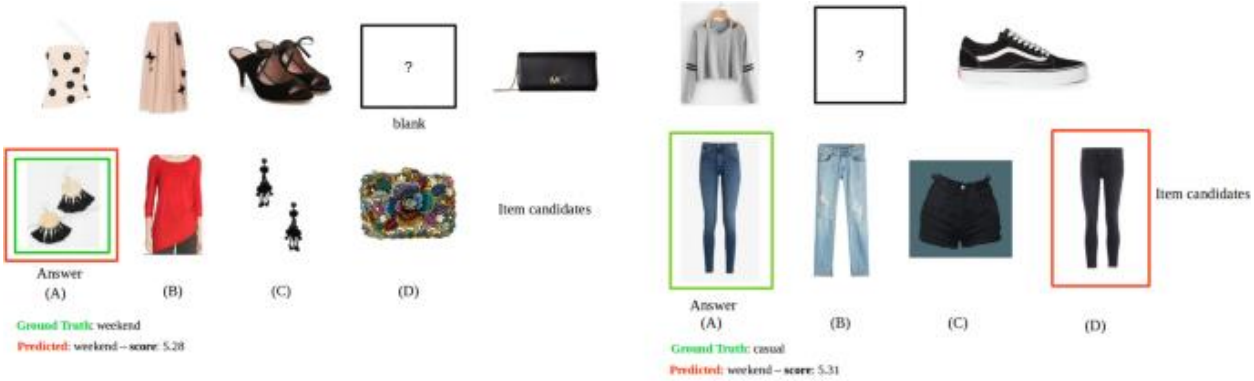
where N_m is the score of an ideal ordering, $y_{\pi'(i)}$ is 1 for positive outfits and 0 for neutral ones. Mean NDCG is the mean of $NDCG@m$ for $m = 1,2,\ldots,M$ with M being the length of the ordering. We will report the average mean NDCG for all outfits and refer to it as NDCG for short in the benchmark section.

5.1.3. FITB

The Fill-in-the-Blank (FITB) task is delineated as follows: when presented with a subset of items constituting an outfit and a set of candidate items (comprising four

[2] https://www.geeksforgeeks.org/auc-roc-curve/

items, with one positive item and three negative items), the objective is to identify the most congruent candidate. To streamline the evaluation process, a repetition of the task is undertaken four times for a given outfit, wherein one item among three candidates is randomly substituted with an item drawn randomly from the dataset. All four candidates are grouped into a batch, wherein the positive item consistently constitutes the initial outfit in each batch. The performance assessment is conducted based on overall accuracy.



(a) Some examples of the correct cases obtained by our method

(b) Some example of the failed cases obtained by our method

Figure 5.2. *Visualization of the FITB task, from Vo et al. 2023*[3]

5.2. Benchmark

To the best of our knowledge, this is the first work to study outfit retrieval matching a user prompt. The methods that can be compared are very limited.

For this problem, we only experimented with the fashion hashing network in two different training methods: one trained exclusively for outfit item compatibility, serving as a baseline for comparison, and the other trained for both outfit item and textual description compatibility. Additionally, we conducted a comparative analysis with a model trained solely from outfit item images within the Polyvore dataset. We call these models FHN-T3 to resemble the original paper [1]. Specifically, we compared 3 models, all of which are trained for 100 epochs:

[3] Vo, A.H., Le, T.B.T., Pham, H.V. et al. An efficient framework for outfit compatibility prediction towards occasion. Neural Comput & Applic 35, 14213–14226 (2023). https://doi.org/10.1007/s00521-023-08431-1

- FHN-T3 (Visual - Polyvore): the baseline method, the FHN model is trained on item images of the Polyvore dataset.
- FHN-T3 (Visual): the FHN model is trained on item images of the Fashion32 dataset.
- FHN-T3 (Visual + Outfit semantic): the FHN model is trained on item images of the Fashion32 dataset and outfit textual description embedding accompanying each outfit.

The corresponding results are shown in **Table 5.1** below.

Table 5.1. *Comparison of different training methods on the Fashion32 dataset*

Method	**Accuracy**	**AUC**	**NDCG**	**FITB**
FHN-T3 (Visual - Polyvore)	0.6232	0.6115	0.7153	0.3520
FHN-T3 (Visual)	0.8191	**0.8150**	**0.8518**	**0.5442**
FHN-T3 (Visual + Outfit semantic)	**0.8706**	0.7416	0.7982	0.5071

We carried out experiments involving the initial training of the model on the Polyvore dataset, followed by fine-tuning on the Fashion32 dataset using a pre-trained model. Unfortunately, this approach did not produce the intended results. We explored the inclusion of outfit description embedding in the training process, but the results did not surpass those of the top-performing model. This outcome may be attributed to the possibility that outfit semantics introduced noise rather than enhancing the model's performance. The most successful model, as depicted in **Table 5.1**, resulted from exclusive training on the Fashion32 dataset without incorporating outfit textual semantics.

Figure 5.3 illustrates the performance of our top model concerning various positive and negative outfit pairings. The negative outfit components are randomly selected from the training dataset, making it simpler for the model to assign elevated scores to positive items already present in the training data. In **Figure 5.4**, the challenge intensifies as the distinction between positive and other negative outfit candidates boils down to just a single item. The model encounters challenges in distinguishing positive outfits from negative ones, particularly when the outfit contains a substantial number of items.

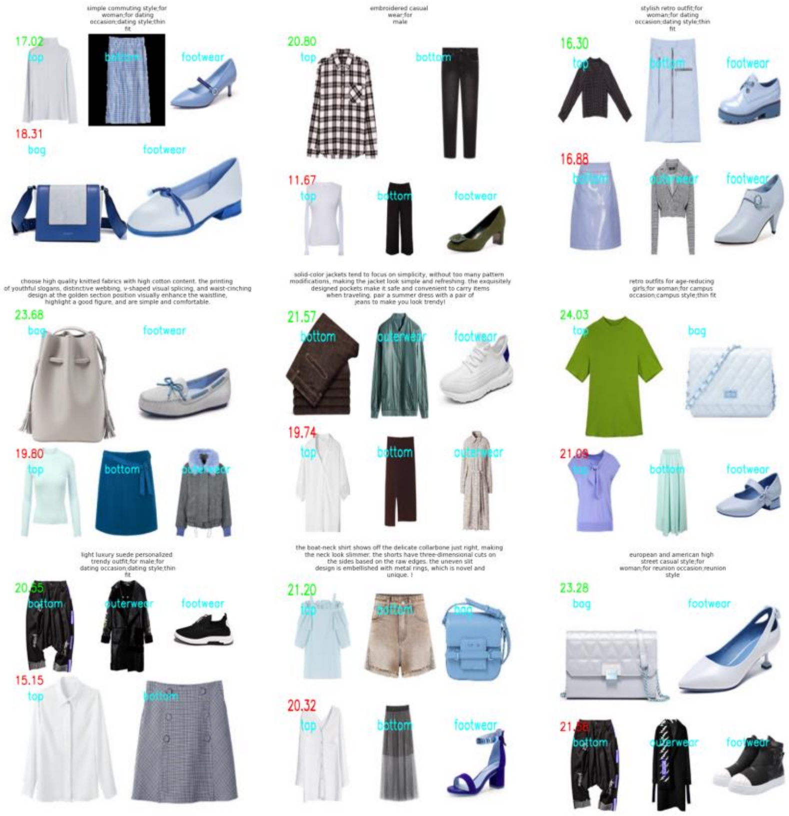

simple commuting style;for woman;for dating occasion;dating style;thin fit
17.02
top
bottom
footwear
18.31
bag
footwear
embroidered casual wear;for male
20.80
top
bottom
11.67
top
bottom
footwear
stylish retro outfit;for woman;for dating occasion;dating style;thin fit
16.30
top
bottom
footwear
16.88
bottom
outerwear
footwear
choose high quality knitted fabrics with high cotton content. the printing of youthful slogans, distinctive webbing, v-shaped visual splicing, and waist-cinching design at the golden section position visually enhance the waistline, highlight a good figure, and are simple and comfortable.
23.68
bag
footwear
19.80
top
bottom
outerwear
solid-color jackets tend to focus on simplicity, without too many pattern modifications, making the jacket look simple and refreshing. the exquisitely designed pockets make it safe and convenient to carry items when traveling. pair a summer dress with a pair of jeans to make you look trendy!
21.57
bottom
outerwear
footwear
19.74
top
bottom
outerwear
retro outfits for age-reducing girls;for woman;for campus occasion;campus style;thin fit
24.03
top
bag
21.09
top
bottom
footwear
light luxury suede personalized trendy outfit;for male;for dating occasion;dating style;thin fit
20.55
bottom
outerwear
footwear
15.15
top
bottom
the boat-neck shirt shows off the delicate collarbone just right, making the neck look slimmer. the shorts have three-dimensional cuts on the sides based on the raw edges. the uneven slit design is embellished with metal rings, which is novel and unique. !
21.20
top
bottom
bag
20.32
top
bottom
footwear
european and american high street casual style;for woman;for reunion occasion;reunion style
23.28
bag
footwear
21.56
bottom
outerwear
footwear

Figure 5.3. *Sample of positive and negative outfit pairs, accompanied by corresponding scores, aligned with the positive outfits' descriptions. Positive outfit scores are emphasized in green, whereas negative outfit scores are highlighted in red*

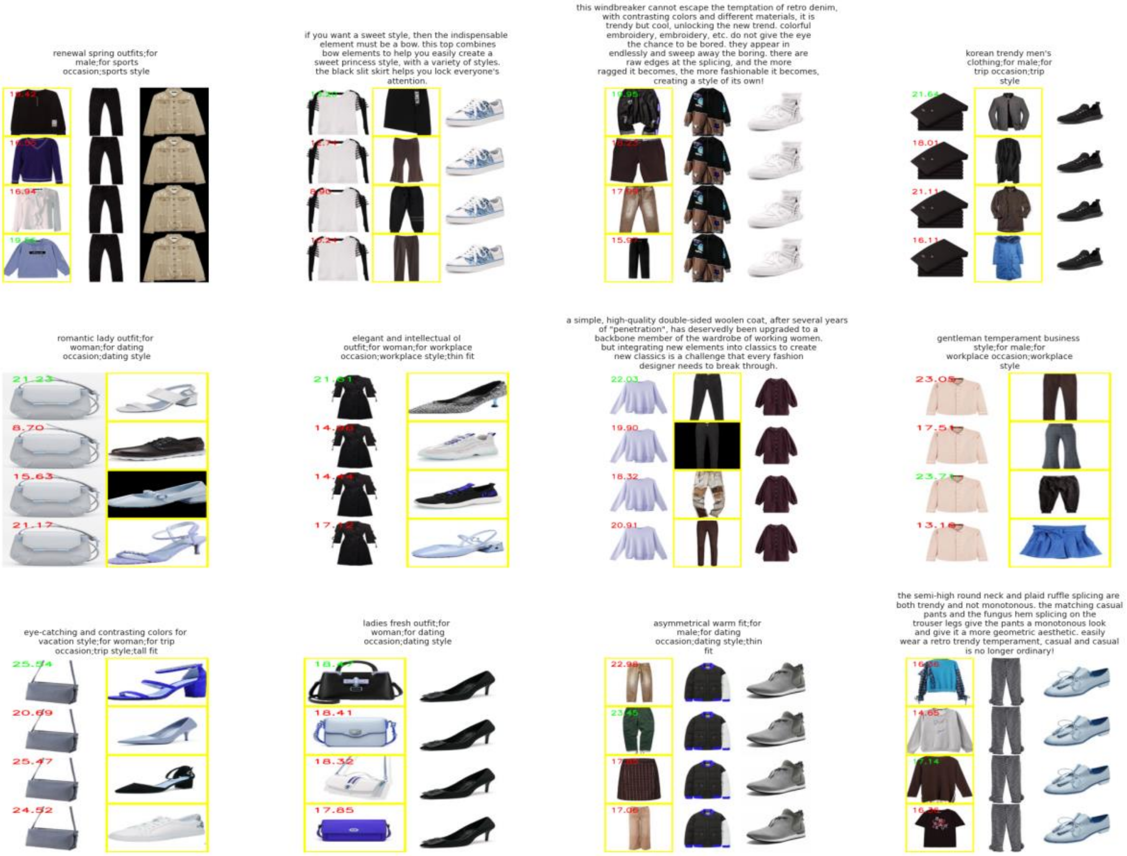
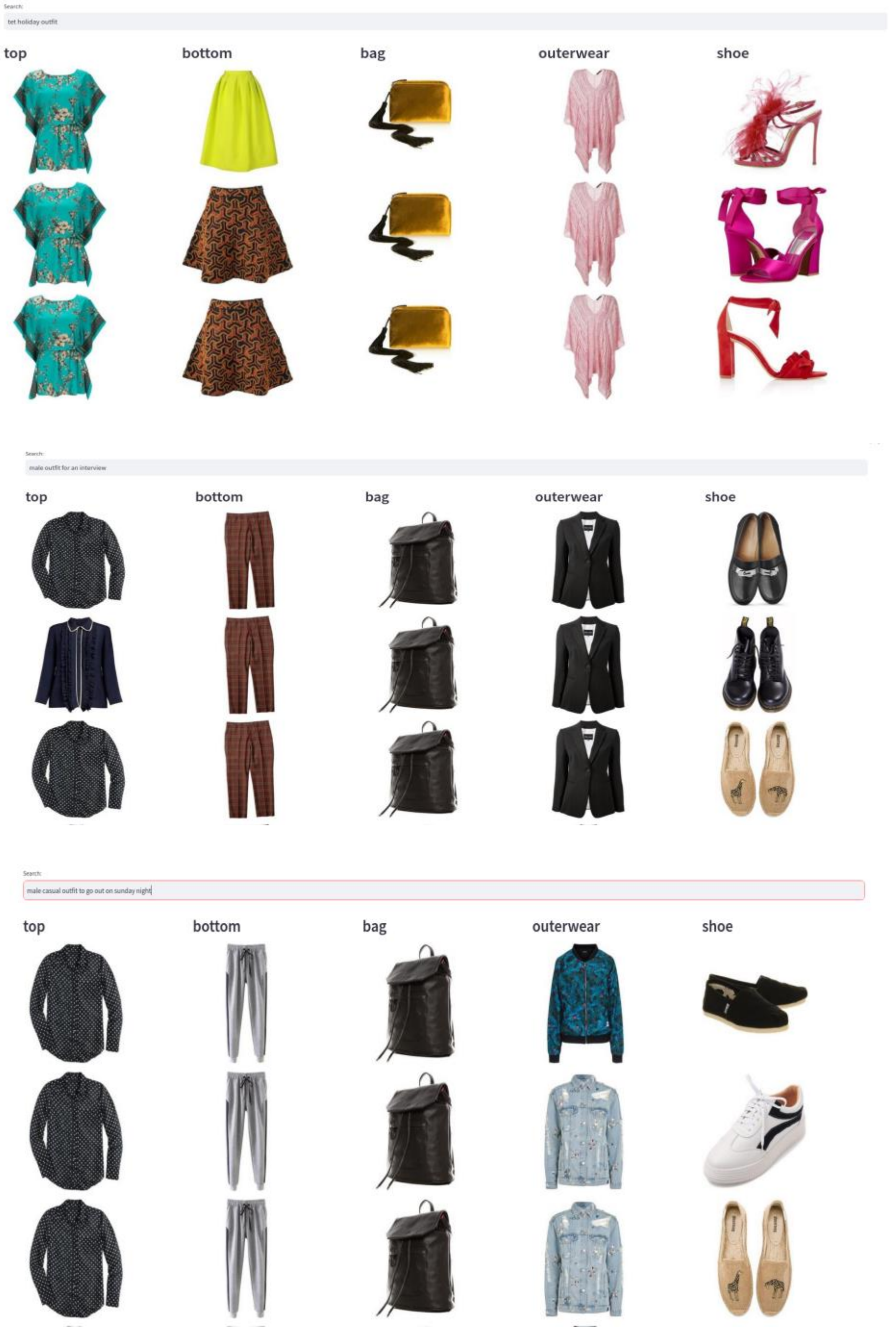

renewal spring outfits;for male;for sports occasion;sports style
16.94
if you want a sweet style, then the indispensable element must be a bow. this top combines bow elements to help you easily create a sweet princess style, with a variety of styles. the black slit skirt helps you lock everyone's attention.
this windbreaker cannot escape the temptation of retro denim, with contrasting colors and different materials, it is trendy but cool, unlocking the new trend. colorful embroidery, embroidery, etc. do not give the eye the chance to be bored. they appear in endlessly and sweep away the boring. there are raw edges at the splicing, and the more ragged it becomes, the more fashionable it becomes, creating a style of its own!
korean trendy men's clothing;for male;for trip occasion;trip style
21.64
18.01
21.11
16.11
romantic lady outfit;for woman;for dating occasion;dating style
21.23
8.70
15.63
21.17
elegant and intellectual ol outfit;for woman;for workplace occasion;workplace style;thin fit
a simple, high-quality double-sided woolen coat, after several years of "penetration", has deservedly been upgraded to a backbone member of the wardrobe of working women. but integrating new elements into classics to create new classics is a challenge that every fashion designer needs to break through.
22.03
19.90
18.32
20.91
gentleman temperament business style;for male;for workplace occasion;workplace style
23.05
eye-catching and contrasting colors for vacation style;for woman;for trip occasion;trip style;tall fit
20.69
25.47
24.52
ladies fresh outfit;for woman;for dating occasion;dating style
18.41
18.32
17.85
asymmetrical warm fit;for male;for dating occasion;dating style;thin fit
22.98
the semi-high round neck and plaid ruffle splicing are both trendy and not monotonous. the matching casual pants and the fungus hem splicing on the trouser legs give the pants a monotonous look and give it a more geometric aesthetic. easily wear a retro trendy temperament, casual and casual is no longer ordinary!
14.65

Figure 5.4. *Sample of candidate outfits from the dataset, evaluating for FITB task. Positive outfit scores consistently appear in the first row across the four candidates. The highest scores are depicted in green, whereas the others are marked in red. Missing items are indicated by a yellow-line square*

5.3. Demonstration

Our model pipeline is deployed using the FastAPI library, and the model implementation is showcased within a web application utilizing the Streamlit library. The model retrieves images from a PostgreSQL database comprising approximately 500 randomly selected images sourced from the Polyvore dataset. We assess the model's performance on a modest computing platform equipped with an NVIDIA GeForce GTX 1050Ti GPU and 8GB of RAM.



Search:

tet holiday outfit

Search:

male outfit for an interview

Search:

male casual outfit to go out on sunday night

Figure 5.5. *Application showcasing retrieval demonstrations for some specific prompts*

Figure 5.5 illustrates some prototypical examples of query-based retrieval scenarios, in which a user inputs a query into the search interface, thereby triggering the system to retrieve ensembles from the database that are compatible with the provided query. Each row of the output exhibits an ensemble comprising five items from five distinct garment categories listed above.

Chapter 6: Discussion

Despite the new approach in our model, certain limitations persist. One notable drawback lies in handling a variable number of outfit items, introducing a challenge for consistent modeling. Additionally, the scoring mechanism relies on the sum of pairwise relationships between items, lacking scalability across various item categories. It is because as an increasing number of item categories are trained and retrieved, the larger score pair tends to average out the smaller ones. A potential improvement could involve adopting a weighted score sum between pairs of items. However, the variable number of outfit items, resulting in a variable number of item pairs, complicates this approach, necessitating further investigation and resolution. Consequently, more work must be done to investigate and resolve this issue. Additionally, there has not been a proper metric for prompt-base outfit retrieval, so a new metric should be designed dedicated to this problem. Additionally, the absence of a proper metric for prompt-based outfit retrieval highlights the need for a dedicated metric tailored to this problem. Overcoming these challenges holds the potential to enhance the model's robustness, broaden its applicability, and invite further research studies into this space.

Chapter 7: Conclusion and Future Works

In summary, our thesis focuses on the application of the FashionCLIP model for image retrieval based on user prompts and the efficient implementation of a fashion outfit recommendation system through the utilization of a fashion hashing network. We experimented with the integration of textual descriptions into both the training and inference framework. Finally, we showed how to combine the two models, creating an efficient pipeline. While there are various approaches to represent outfit compatibility, our method proves to be the most practical for efficient outfit retrieval in both the training and inference phases. Through extensive experiments on the Polyvore and Fashion32 datasets, our approach demonstrates strong performance across a diverse array of prompts, considering factors such as gender, occasion, and style. While the system performs well in practical scenarios, there is room for future enhancements, particularly in terms of inference speed, aesthetic capabilities, and potential expansions into areas such as room design.

References

[1] Lu, Zhi, et al. “Learning binary code for personalized fashion recommendation.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, https://doi.org/10.1109/cvpr.2019.01081.

[2] Methods and Advancement of Content-Based Fashion Image Retrieval: A Review, arxiv.org/abs/2303.17371.

[3] Rubio, A., et al. “Multi-modal joint embedding for fashion product retrieval.” 2017 IEEE International Conference on Image Processing (ICIP), 2017, https://doi.org/10.1109/icip.2017.8296311.

[4] Y. -G. Shin, Y. -J. Yeo, M. -C. Sagong, S. -W. Ji and S. -J. Ko, "Deep Fashion Recommendation System with Style Feature Decomposition," 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 2019, pp. 301-305, doi: 10.1109/ICCE-Berlin47944.2019.8966228.

[5] Zhu, J. et al. (2023) Fashion image retrieval with multi-granular alignment, arXiv.org. Available at: https://arxiv.org/abs/2302.08902.

[6] Radford, A. et al. Learning transferable visual models from natural language supervision. In ICML (2021).

[7] Chia, P.J., Attanasio, G., Bianchi, F. et al. Contrastive language and vision learning of general fashion concepts. Sci Rep **12**, 18958 (2022). https://doi.org/10.1038/s41598-022-23052-9.

[8] Yang Hu, Xi Yi, and Larry S Davis. Collaborative Fashion Recommendation: A Functional Tensor Factorization Approach. In ACM MM, 2015.

[9] Vasileva, M.I. *et al.* (2018) *Learning type-aware embeddings for fashion compatibility*, *arXiv.org*. Available at: https://arxiv.org/abs/1803.09196.

[10] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning Visual Clothing Style with Heterogeneous Dyadic Co-Occurrences. In ICCV, 2015.

[11] Han, X. *et al.* (2017) *Learning fashion compatibility with bidirectional LSTMs*, *arXiv.org*. Available at: https://arxiv.org/abs/1707.05691.

[12] Lai, J.-H. et al. (2020) Theme-matters: Fashion compatibility learning via theme attention, arXiv.org. Available at: https://arxiv.org/abs/1912.06227.

[13] Lin, Y.-L., Tran, S. and Davis, L.S. (1970) Fashion outfit Complementary Item Retrieval, CVF Open Access. Available at: https://openaccess.thecvf.com/content_CVPR_2020/html/Lin_Fashion_Outfit_Complementary_Item_Retrieval_CVPR_2020_paper.html.

[14] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep Supervised Hashing for Fast Image Retrieval. In CVPR, 2016.

[15] Lei Zhang, Yongdong Zhang, Jinhu Tang, Ke Lu, and Qi Tian. Binary Code Ranking with Weighted Hamming Distance. In CVPR, 2013.

[16] Qifan Wang, Dan Zhang, and Luo Si. Weighted Hashing for Fast Large Scale Similarity Search. In CIKM, 2013.

[17] Jian Zhang and Yuxin Peng. Query-Adaptive Image Retrieval by Deep-Weighted Hashing. TMM, 2018.

[18] Rombach, R. et al. (2022) High-resolution image synthesis with Latent Diffusion Models, arXiv.org. Available at: https://arxiv.org/abs/2112.10752.

[19] Aditya, R. et al. (2021) Zero-Shot Text-to-Image Generation, arXiv.org. Available at: https://arxiv.org/abs/2102.12092.

[20] A Krizhevsky, I Sutskever, GE Hinton - Advances in neural information processing systems, 2012.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In Advances in Neural Radford, A. et al. Learning transferable visual models from natural language information processing systems, pp. 5998–6008, 2017.

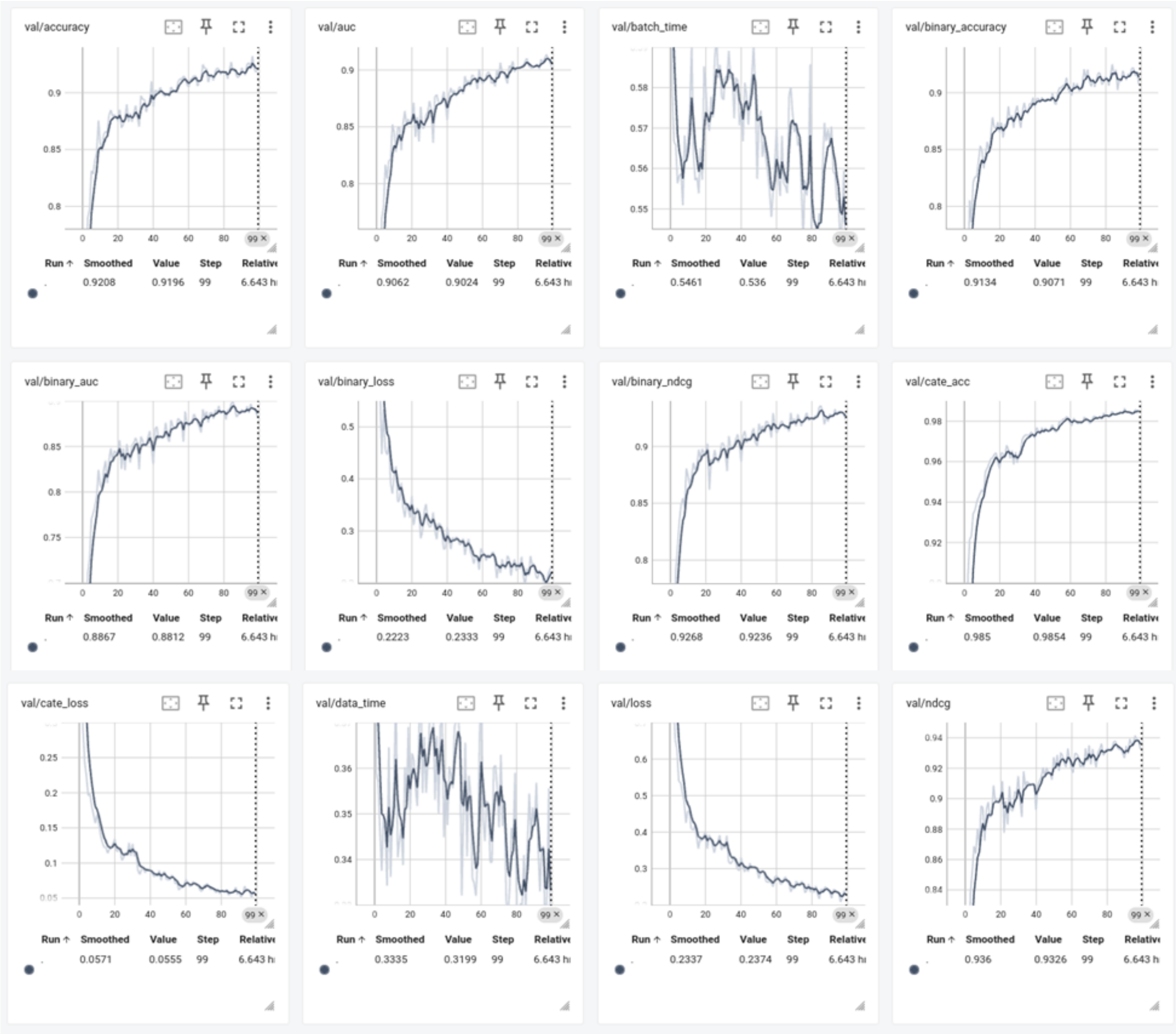
[22] Radford, Alec, and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training.” (2018).

[23] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

[24] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian Personalized Ranking from Implicit Feedback. In UAI, 2009.

Appendix



val/accuracy

Run ↑	Smoothed	Value	Step	Relative
●	0.9208	0.9196	99	6.643 h

val/auc

Run ↑	Smoothed	Value	Step	Relative
●	0.9062	0.9024	99	6.643 h

val/batch_time

Run ↑	Smoothed	Value	Step	Relative
●	0.5461	0.536	99	6.643 h

val/binary_accuracy

Run ↑	Smoothed	Value	Step	Relative
●	0.9134	0.9071	99	6.643 h

val/binary_auc

Run ↑	Smoothed	Value	Step	Relative
●	0.8867	0.8812	99	6.643 h

val/binary_loss

Run ↑	Smoothed	Value	Step	Relative
●	0.2223	0.2333	99	6.643 h

val/binary_ndcg

Run ↑	Smoothed	Value	Step	Relative
●	0.9268	0.9236	99	6.643 h

val/cate_acc

Run ↑	Smoothed	Value	Step	Relative
●	0.985	0.9854	99	6.643 h

val/cate_loss

Run ↑	Smoothed	Value	Step	Relative
●	0.0571	0.0555	99	6.643 h

val/data_time

Run ↑	Smoothed	Value	Step	Relative
●	0.3335	0.3199	99	6.643 h

val/loss

Run ↑	Smoothed	Value	Step	Relative
●	0.2337	0.2374	99	6.643 h

val/ndcg

Run ↑	Smoothed	Value	Step	Relative
●	0.936	0.9326	99	6.643 h

Figure A.1. *Validation curve of some metrics while training our model*