Blocking Note

This Thesis should be blocked from public use until xxxxxxxxxxxxx. Publication, photocopying and viewing may be possible only with consent from the author.

Dornbirn, July 2025

Viktoriia Simakova

FHV
Vorarlberg University
of Applied Sciences

[Titel der Arbeit]

[Untertitel der Arbeit]

Master Thesis
Submitted in Fulfillment of the Degree

Master of Science in Engineering (MSc)

University of Applied Sciences Vorarlberg

Submitted to
DI Dr. techn. Sebastian Hegenbart

Handed in by
Viktoriia Simakova
Dornbirn, July 2025

0.1 Kurzreferat

0.1.1 [Deutscher Titel Ihrer Arbeit]

[Text des Kurzreferats]

Keywords in German: Machine Learning, Computer Vision, ...

Abstract

[English Title of your thesis]

Functionally, the proposed system will process input images and generate an output score that quantifies the aesthetic quality of the outfit. Scientifically, the work contributes to advancing AI's ability to interpret subjective domains such as fashion, where cultural, contextual, and individual factors significantly influence perceptions of style.

Keywords in English: Machine Learning, Computer Vision, ...

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

AUC Area under the curve

AI Artificial Intelligence

Bi-LSTM Bidirectional Long Short-Term Memory

CV Computer Vision

CNN Convolutional Neural Network

DL Deep Learning

DNN Deconvolutional Neural Network

FITB Fill-in-the-Blank

GAN Generative Adversarial Network

GAP Global Average Pooling

GloVe Global Vectors for Word Representation

GCN Graph Convolutional Network

HR Hit Rate

HCD Human-Centered Design

HCC Human-Centered Computing

LR Learning Rate

ML Machine Learning

MRR Mean Reciprocal Ranking

MLP Multi-Layer Perceptron

SCL Self Supervised Contrastive Learning

NC-SSL Non-Contrastive Self Supervised Learning

RNN Recurrent Neural Network

RN Relational Network

ReLU Rectified Linear Unit

ResNet Residual Network

SSL Self-Supervised Learning

SMPL Skinned Multi-Person Linear model

VSE Visual Semantic Embedding

1 Introduction

The integration of Artificial Intelligence (AI) into the fashion industry has opened new doors for innovation and has transformed key areas such as design, production, retail and marketing. Within this rapidly evolving landscape, one particularly interesting application is personalized styling. Specifically, the use of AI to evaluate (and recommend) fashion outfits tailored to individual preferences. This thesis investigates the application of Deep Learning (DL) techniques to assess the visual quality of fashion ensembles, with the central research question being:

> **How can existing DL models be used to give a score representing evaluation of visual compatibility of a fashion outfit based on images of individuals wearing clothes?**

This introductory chapter describes the initial situation and defines objectives and expected outcomes. The result of this master's thesis is a computational solution capable of analyzing images of individuals wearing outfits and assigning a score that reflects the overall aesthetic appeal and visual coherence of the look.

1.1 Motivation

The motivation for this thesis lies in the potential of AI to address gaps in personalized fashion. Current AI-driven fashion applications often focus on generic outfit recommendations, neglecting the nuanced interplay between visual aesthetics, personal preferences and contextual factors. There exists an opportunity to develop systems that, on the one hand, understand general styling guidelines and recognize patterns in fashion styles and, on the other hand, adapt to individual tastes and situational demands. By developing a robust AI-powered outfit evaluation system, the groundwork is laid for creating a personalized AI-powered stylist application. Such an application could not only assess outfits but also provide tailored recommendations that align with individual preferences and contextual requirements. This thesis represents a foundational step toward understanding how AI can be leveraged to enhance user experiences in the fashion domain, satisfying the growing demand for personalized fashion solutions in our digital age.

1.2 Objectives and Problem Statement

This thesis addresses four significant gaps in the field:

1. There is limited exploration of AI models that are specifically designed for outfit evaluation and consider all person's individual features. [1, vgl.], [2, vgl.]

2. Existing methods evaluate the compatibility between only two items (e.g. top and bottom) for a specific user. However, real-world outfits typically consist of multiple items such as shoes, accessories and more. Current models assume a fixed input size, limiting their ability to handle outfits with varying numbers of items. [1, vgl.]

3. There is a lack of methods that explore compatibility without relying on predefined category labels. In realistic fashion scenarios, items may belong to overlapping or ambiguous categories. [1, vgl.]

4. Existing approaches often fail to account for the interplay between visual aesthetics, contextual factors and personal preferences. [1, vgl.], [2, vgl.]

This thesis addresses these challenges by proposing a solution that assesses the visual personal quality of outfits from images of individuals wearing them. The approach integrates existing DL techniques and is capable of evaluating compatibility across outfits with an arbitrary number of items, moving beyond the limitations of fixed-input models.

To achieve this, focus lies on several key tasks. First, existing research and the DL models used in it are analyzed for their applicability to outfit evaluation tasks. Based on this, a concept of a solution is created while considering tackling the gaps identified earlier. Then, suitable models are selected and integrated into a pipeline that processes input images and generates a numerical score that reflects the visual quality of the outfit. Third, experiments are conducted to evaluate the effectiveness of implementation alone as well as with incorporated supplementary data on contextual factors into the evaluation process.

1.3 Scope and Expected Outcomes

The scope of this thesis encompasses several key areas within AI and fashion technology. Technically, the study focuses on the utilization and adaptation of existing state-of-the-art DL models for image-based outfit evaluation. This includes experimenting with different types of ML models within Computer Vision (CV), transfer learning, feature extraction methods, embedding techniques and evaluation metrics to assess visual elements. Functionally, the project involves developing a pipeline that accepts images as input and outputs a numerical score reflecting the visual quality of the outfit. Scientifically, the research analyzes the effectiveness of combining different model architectures in capturing subjective aesthetic judgments.

The expected outcomes of this research include the development of a proof-of-concept prototype that demonstrates the feasibility of AI-driven outfit evaluation. This prototype will serve as the foundation for a personalized AI-Stylist application, enabling users to receive real-time feedback on their outfits and access tailored recommendations.

2 Requirements Analysis

This chapter outlines the requirements, challenges and constraints associated with the task of developing an AI-based outfit evaluation system.

2.1 Definition of Requirements

The requirements for the outfit evaluation system can be categorized into functional and non-functional requirements. Each addresses specific aspects of the design and operation of the system.

Functional Requirements:

- The system must be capable of accepting high-resolution images as input which serve as the primary data source for evaluation. These images capture individuals wearing outfits.
- The images must be preprocessed in order to satisfy the need for a format that is more suitable for analysis.
- The system must be analysed based on visual aesthetics including features of the outfit (e.g. colors, patterns, prints, shapes, cuts, texture) as well as the person's individual features (e.g. body shapes, hair colors, skin colors, age).
- The system must provide users with a numerical score as feedback.
- Optionally: The system must be capable of identifying common patterns in visual outfit aesthetics, while supporting the optional integration of contextual data and personal preferences to enhance the accuracy of outfit evaluations. This can data can include occasion details (e.g. occasion type, location, cultural and social background), environmental factors (e.g. season, temperature, weather conditions) and impressions/mood (e.g. formal, casual).

Non-Functional Requirements:

- The evaluation process must achieve a high degree of accuracy in assessing the quality of the outfit.

- The system must be accessible through an intuitive interface, enabling users to upload images and receive evaluations.

2.2 Challenges and Constraints

Several technical, functional and ethical challenges must be managed to ensure the feasibility and effectiveness of the solution.

Technical Challenges:

- The accuracy of DL models is highly dependent on the availability of a high-quality and diverse dataset. However, obtaining datasets that accurately represent a wide range of fashion styles and contexts poses a significant challenge.

- The concept of fashion is subtle and subjective. A critical challenge is the definition of "good" and "bad" and the quantification of subjective qualities such as "visual appeal" or "style harmony". The evaluation of the outfit varies widely between individuals and is influenced by subjective factors, including cultural norms, personal preferences and context. [1, vgl.]

- Since each outfit consists of multiple complementary pieces (such as tops, bottoms, shoes, accessories) item compatibility spans across categories and involves complex interrelationships. [1, vgl.]

Functional Constraints:

- To provide a seamless user experience, the system must evaluate outfits and generate scores in real-time. Achieving this within acceptable latency limits imposes constraints on model complexity and computational resources.

- Users expect clear and understandable explanations for the scores assigned to their outfits. Designing a system that not only evaluates but also interprets and communicates results effectively is a non-trivial task.

Ethical Constraints:

- While potential biases in outfit evaluation (such as those related to gender, ethnicity, body type, socioeconomic status) must be addressed to ensure fairness and inclusivity, this thesis does not explicitly tackle bias mitigation. As discussed in prior work (e.g. [2, vgl.]), such biases can lead to stereotypical recommendations, for instance by reinforcing traditional gender norms in fashion. However, defining and measuring fairness in fashion recommendation remains a complex challenge which is heavily influenced by cultural and contextual factors. This thesis focuses on developing a flexible framework that can accommodate diverse datasets in future applications, allowing for the integration of fairness considerations as needed.

3 Background and Literature Review

In the fashion industry, AI is applied to a range of tasks and objectives, including analysis, recommendation and synthesis, among others as described in [1, vgl.], [2, vgl.], [3, vgl.] and [4, vgl.] and as illustrated in Figure 3.1.

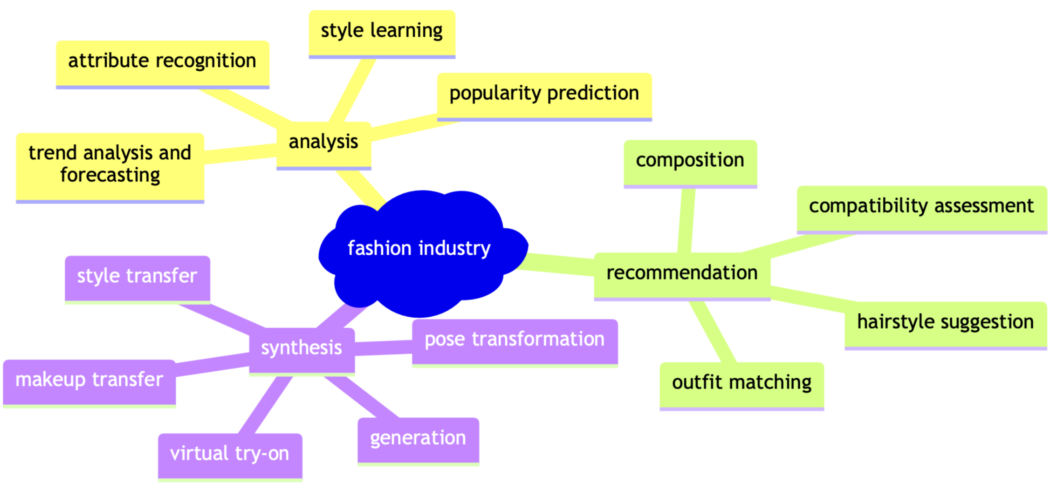

Figure 3.1: AI in Fashion Areas (Mindmap)

The goal of fashion recommendation is to automatically provide users with advice on what looks best on them and how to improve their style. This compatibility assessment is commonly associated with the task of outfit matching, where the overall collaboration between fashion items such as tops, bottoms, shoes, accessories is measured. [1, vgl.]

3.1 Relevant Techniques in Outfit Matching and Compatibility Evaluation

Outfit matching and compatibility evaluation combine principles from fashion theory, DL and CV to assess how well clothing items harmonize. This chapter explores key techniques, starting with foundational rules and progressing to computational models.

Techniques within Fashion Theory:

In the literature, fashion recommendation systems are categorized into two main types: [5, vgl.]

1. Similar item recommendation, which includes image retrieval techniques that suggest visually similar or identical items.
2. Complementary recommendation, which includes three approaches:
   - Item recommendation: Predicting one single item to complete an incomplete outfit based on a specific category.
   - Outfit recommendation / outfit completion: Building a full outfit from scratch or adding matching items to an existing partial outfit.
   - Capsule wardrobes: Recommending a minimal set of versatile items that can be mixed and matched to create multiple outfits.

Complementary recommendations can be approached in three ways:

- Product-based: Assessing compatibility between two items using product images.
- Scene-based: Incorporating contextual details like season, location or user-specific factors.
- Occasion-based: Tailoring recommendations to specific events or cultural/social contexts.

Compatibility evaluation among fashion items is typically modeled as:

- Pair-wise: Evaluating compatibility between two items.
- List-wise: Assessing sequences of items.
- Set-wise: Analyzing compatibility across an entire outfit as a holistic set.

Techniques within DL and CV:

In addressing the task of visually aware evaluation of the outfit composition, CV is employed. CV aims to create methods for computers to replicate the complexity of the human visual system by understanding digital images (e.g. photos, videos, other visual media) and extracting valuable information from them. [6, vgl.]

In this context, a DL model, an algorithm that is modeled after the structure of the human brain, acts as a foundational element. These models consist of layers of interconnected neurons that process and transform input data. The weights of the connections between neurons in the network are adjusted over time to recognize patterns in data that are relevant to a specific task. Thereby, complex representations of data are automatically learned. Some popular DL architectures include CNN which identify images at the pixel level RNN which handle sequential data. [7, vgl.]

One of the key DL models in fashion is ResNet. Its key feature is the use of skip/residual connections across one or more layers, which help mitigate the vanishing gradient problem in very deep networks. It uses building blocks and lets each block learn a modification (residual) to its input, rather than the desired underlying function. This allows gradients to flow backward. Therefore, this model can handle very deep networks (with over 100 layers: e.g. ResNet-50, ResNet-101, ResNet-152), making it suitable for complex fashion datasets where high accuracy is required. **<empty citation>**

MLP VSE RN GAN DNN Bi-LSTM

Siamese networks, often called twin networks, consist of a pair of neural networks that share their weights and aims at computing similarity functions. Essentially, their main objective is to identify whether a pair of data is dissimilar or not. Fig. 7.2 illustrates an example of a Siamese network architecture. Siamese networks reference

3.2 Previous Work and Related Research

Current chapter provides an overview of 10 papers that were selected in terms of their relevance to the use case. For all studies, their main goal and some technical elements, including model architecture, inputs, evaluation metrics, hyperparameters and other key components were summarized.

Pairwise Approach. Wang et al. present a system that can predict whether a set of clothing items representing an outfit looks good and explain why it does (not). The developed network uses a CNN (ResNet 50) and GAP to extract features from images of clothing items at different levels of abstraction. These features range from basic details such as color and texture (early layers) to more complex ones such as style and category (later layers). The extracted features are then used to compare each outfit item with every other outfit item in a pairwise comparison matrix. MLP then produces a score that reflects the final overall compatibility score. Thus, both visual and textual information is integrated using VSE which allows the model to learn a common representation between them. Gradient values generated by backpropagation are used to identify problematic pairs and to provide an explanation of why an outfit fails, pinpointing specific issues. As loss functions, the model employs binary cross-entropy and contrastive loss. The training is supervised, uses labeled data and learns with negative sampling. The prediction of compatibility is evaluated using metrics such as AUC and FITB accuracy. The key hyperparameters mentioned include: initial LR: 0.01, decay factor: 0.2 every 10 epochs, CNN depth: 4 layers, MLP depth: 2 layers. [8, vgl.]

Relational Approach. Moosaei et al. tackle the challenge of creating a system that could 1) work with any number of clothing items 2) without needing a specific order and 3) without relying on traditional labels. First, RN is used to treat each outfit as a "scene" and the items within it as "objects", thus learning how items relate to each other visually. After establishing the relationships, it combines them using MLP to create a single score that indicates how well the outfit fits together. The authors also develop a more sophisticated version of the network that additionally incorporates textual information. DenseNet is used for visual feature extraction, one-hot encoding for textual features. The model uses cross-entropy loss. The training is supervised. The evaluation uses AUC and FITB accuracy. The key hyperparameters mentioned include: LR: 0.001, batch size: 64, dropout rate: 0.35, epochs: 19, optimizer: Adam, MLP depth: 4 layers (number of filters: 512, 512, 256, 256) + 3 layers (number of filters: 128, 128, 32). [9, vgl.]

Generative Approach and Template Generation. Liu et al. aim not only to measure compatibility but also to generate a "compatible template" that could help in understanding why certain pairings succeed or fail. The authors trained a GAN on a massive dataset of clothing images paired with detailed textual descriptions to create a richer understanding of clothing compatibility. The architecture integrates down-sampling, multi-modal fusion and up-sampling. Convolutional layers are used for visual encoding, TextCNN for textual. The network learns to generate preliminary visual representations (templates) of what a compatible clothing item should look like based on a given one. AUC and MRR are used as metrics for evaluating the model. The

LR was set to 0.0002. [10, vgl.]

Generative Approach. In another research, Moosaei et al. show a model used to generate compatible fashion items for an (incomplete) outfit. The developed model consists of two parts. The GAN takes as input a partial outfit (images) along with a specified missing clothing item category (textual) and creates several potential outfit combinations. A compatibility network (CNN + MLPs + RN) checks if the generated item fits well with the rest of the outfit by identifying patterns in the relationships between items. It learns what makes different clothing items match each other based on their contextual relevance (relationships) and their visual aesthetics by incorporating the initial outfit input into its training. As a loss function, the model employs cross-entropy loss among others. Training is supervised for compatibility network and min-max game for GAN. The prediction is evaluated using inception score, multi-scale structural similarity and compatibility score. [11, vgl.]

Graph-based Approach and Try-on Approach. Zheng et al. address item-by-item matching (collocation) and overall outfit appearance (try-on). Both of these perspectives are combined in a network to give a better evaluation of outfit compatibility. The developed model consists of two parts. [12, vgl.]

1. The first part looks at each clothing item individually and checks how well they match with each other. This approach uses a disentangled GCN and includes nodes (each representing clothing items), edges (showing the connections between items), condition masks (acting like filters that separate out different features of clothing items) and an attention mechanism (deciding which features are more crucial for determining if items match). Convolutional layers are used for visual features, ResNet-like architecture for try-on images.

2. The second part imagines how the whole outfit would look when worn together and outputs the final try-on compatibility score. Thereby, the authors apply knowledge distillation and train a "teacher" network using available try-on images before transferring knowledge to a "student" network. This second network predicts how the outfit would look without the need for actual try-on images. Furthermore, item category information (such as top, bottom, etc.) is incorporated to understand the context of the outfit.

In this paper, cross-entropy is chosen as a loss function, while the Kullback-Leibler divergence and L1 regularization are used as regularization terms. Instance normalization is applied. The training is semi-supervised with mutual learning strategy. The prediction of compatibility is evaluated using metrics such as AUC, MRR, HR @1, @10, @100, @200. The key hyperparameters mentioned include: LR: 0.0002, batch size: 32, optimizer: Adam, activation function: ReLU (and Tanh), GCN depth: one 1-strided convolutional layer and four 2-strided convolutional layers (number of filters: 32, 64, 128, 256, and 512, respectively), teacher network depth: same as GCN for encoder + transform block composed of 6 residual blocks + decoder with four 2-strided deconvolutional layers and one 1-strided convolutional layer (number of filters: 32, 64, 128, 256, 512, 512, 512, 512, 512, 512, 512, 256, 128, 64, 32 and 3, respectively). [12, vgl.]

Graph-based Approach. Work done by Guan et al. presents a system designed to automate the assessment of outfit compatibility while dealing with irregular attribute labels, information loss during disentanglement and combining different levels of information. The system tackles this through a three-stage methodology: [13, vgl.]

1. It leverages a pre-trained model (ResNets 18) to extract visual features from each clothing item. MLPs are applied to break these features down into attributes (partially supervised disentangled learning). Despite the fact that the generated attribute labels are irregular, this partially supervised approach is used to guide the attribute-level learning.

2. To prevent losing information during breakdown, the authors introduce two strategies: orthogonal residual embedding layers (layers that reintroduce missing information) and visual representation reconstruction (a DNN that reconstructs the original image from fragmented attributes).

3. The system builds a graph where nodes represent fashion items and edges represent compatibility relationships (e.g. "matches", "does not match", "requires modification"). Hierarchical GCN is implemented to model the relationships between clothing items, considering both attribute-level and item-level compatibility. The final compatibility score is derived from the combination of both results.

In this paper, cross-entropy is chosen as a loss function, while orthogonal regularization is used as a regularization term. The evaluation uses AUC and FITB accuracy. The key hyperparameters mentioned include: LR: 0.0001, batch size: 32, optimizer: Adam, GCN depth: 1 layer, DNN depth: 5 transposed layers (output dimension: 256), MLPs depth: 2 layers (for each label with output dimension: 64), activation functions: LeakyReLU, ReLU, Tanh. [13, vgl.]

Colors and Textures. Kim et al. implement a model that can learn from unlabeled data using SCL and suggest items that complement each other based on shared color palettes and textures. On the one hand, the model learns to predict the distribution of colors present in images and to recognize color patterns. On the other hand, it learns to identify and recognize different textures (such as stripes, polka dots, etc.). Additionally, in order to filter out irrelevant information (e.g. shape), the model focuses on smaller, independent image patches and learns to identify the types of colors and textures present within these patches. The architecture consists of CNN (ResNet 50) and separate projection heads for sub-tasks. Contrastive loss (for shapeless local patch discrimination, texture discrimination) is chosen as a loss function, while the Kullback-Leibler divergence (for RGB histograms) and L1 regularization are used as regularization terms. The prediction is evaluated using AUC, FITB accuracy, recall@K. The key hyperparameters mentioned include: LR: 0.00005, optimizer: Adam, activation function: ReLU, epochs: 150, heads depth: two fully connected layers. [14, vgl.]

Styles and Textures. Dong et al. present a system that can automatically generate matching clothing items while considering style and texture using SSL. This is done without requiring pairs of outfits that already match, instead mapping an input image of a clothing item to a complementary image. The network utilizes three main parts: [15, vgl.]

1. First component (discriminator with ResNet backbon and MLPs) helps the system understand the style and texture representations of the input clothing. Later on it ensures that the synthesized clothing is compatible with the input clothing in terms of style and texture.

2. The second component (dual discriminator) ensures that the generated images are realistic and visually coherent. One discriminator is designed to favor real images (positive samples) and assigns high scores to latent codes produced by the encoding network, while the other discriminator favors generated images (negative samples) and assigns high scores to latent codes produced by the mapping network. Conversely, the first discriminator assigns low scores to latent codes from the mapping network, and the second assigns low scores to latent codes from the encoding network.

3. Build upon GAN inversion, the last component (generator) uses a pretrained model (StyleGAN) to understand the basic structure of clothing. It then applies style and texture information to generate a matching image, guided by the DST and dual discriminator losses.

Body Shape. Pang et al. designed a model that generates outfit recommendations that prioritize both visual compatibility and body shape suitability. This is achieved through a layered architecture that incorporates: [16, vgl.]

1. Seven body shape representations with 3D body models, measurements and front-view images captured from multiple angles for each body shape. These are used to train the model to understand the overall silhouette using SMPL and ResNet.

2. This part extracts visual features from images that show how an outfit looks when worn (available or generated with M-VTON) using ResNet. It also generates textual descriptions of clothing attributes using GloVe. Both are then represented as vector representations.

3. The final part of the model combines body shape representation and outfit representation into a single, unified representation. Cross-modal attention is used to identify correlations between body shape and outfit attributes, focus on the most relevant features when making recommendations and provide explanations for why an outfit is recommended.

Thereby, binary cross-entropy loss is chosen as a loss function. The evaluating metrics include AUC, mean average precision, average per-class precision, recall, F1 score, average overall precision. The key hyperparameters mentioned are: LR: 0.1, batch size: 10, optimizer: SDG, activation function: ReLU, weight decay: 0.0005, momentum: 0.9. [16, vgl.]

Occasion. In their work, Vo et al. create a system that can tell if different clothing items are compatible for specific occasions beyond simple style matching. The authors designed a framework with three main parts: [17, vgl.]

1. Bi-LSTM analyzes the entire outfit as a sequence (like words in a sentence) and learns how different clothing items relate to each other.

2. ResNet 50 extracts visual features from clothing images and VSE connects them to textual descriptions (one-hot encoded) to understand visual style and matching.

3. The last part focuses specifically on recognizing if an outfit is suitable for a particular occasion. It uses an auxiliary classifier with global average pooling, fully connected layers and softmax to classify outfits based on occasions.

The system is evaluated using metrics such as AUC and FITB accuracy. As a loss function, it employs triplet loss and cross-entropy loss among others. The key hyperparameters mentioned include: batch size: 10, initial LR: 0.2, then changed by a factor of 2 for every two epochs. [17, vgl.]

4 Proposed Solutions

4.1 Overview of Potential Approaches

FashionCLIP is a CLIP-like model fine-tuned for the fashion domain. Fine-Tuning Techniques

Human Pose Estimation reference

category-guided attention mechanisms

Future Research Directions (For Thesis): Investigate the use of machine learning to *dynamically* adjust compatibility rules based on the current outfit's characteristics. Develop a system that allows users to provide more detailed style preferences and integrate these preferences into the generation process. Extend the system to generate outfits that synthesize elements from multiple existing style guidelines, creating unique and novel combinations. - While the output is visually appealing, integration notes suggest a further refinement of the compatibility criteria, potentially incorporating factors like color palettes, silhouette considerations, and formality level. The system performs well with common garment types - shirts, pants, dresses - but struggles with more complex combinations.

Extract features -> compute matching -> compute overall compatibility score Your revised idea is much clearer, and it effectively outlines a strategy to simulate the functionality of a GAN-inspired system without actually implementing a full GAN. This approach leverages existing models and techniques in a creative way, making it both feasible and resource-efficient. Below, I'll break down your concept into its key components, provide feedback, and suggest refinements.

—

1. Core Idea: Simulating GAN-Like Functionality You're inspired by GANs but don't want to implement one due to computational constraints. Instead, you'll use a combination of existing models and techniques to achieve similar functionality.

Strengths: - GAN inspiration is a great starting point for thinking about generative and evaluative components. - By avoiding the complexity of training a GAN, you can focus on leveraging pre-trained models and simpler architectures.

Challenges: - Without a true generator-discriminator loop, you need to carefully design how the "generator-like" and "discriminator-like" components interact. - Ensuring that the system learns meaningful relationships between clothing items and aesthetic ratings will require careful feature engineering.

Suggestion: - Treat the "generator-like" component as a template-guided outfit creator or a random combiner of clothing items. - Use the "discriminator-like" component (e.g., a scoring model) to evaluate the quality of these outfits. - Iteratively refine the outfit creation process based on feedback from the scoring model.

—

2. Discriminator-Like Architecture The discriminator evaluates whether an outfit is "good" or "bad" and assigns an aesthetic rating (1-10).

Strengths: - A scoring model is central to your use case and aligns well with your goal of rating outfits. - Using a 1-10 scale provides fine-grained feedback.

Challenges: - The model needs to learn what makes an outfit aesthetically pleasing, which depends on subjective human preferences. - Training such a model requires high-quality labeled data.

Suggestion: - Start with a pre-trained vision model (e.g., CLIP, ViT, or ResNet) fine-tuned on fashion datasets like DeepFashion or Fashion-MNIST. - Use Siamese networks or triplet loss to create an embedding space where similar outfits are closer together. - Incorporate additional features (e.g., color harmony, balance) into the scoring process.

Let's design a modular discriminator architecture for rating outfits. This architecture will focus on evaluating the aesthetic quality of an outfit based on visual features.

Discriminator Architecture

The discriminator can be built using a combination of convolutional neural networks (CNNs) and multi-layer perceptrons (MLPs). Here's a modular approach:

1. Feature Extraction Module - Architecture: Use a pre-trained CNN like ResNet-50 or VGG16 to extract features from outfit images. This module will capture visual attributes such as color, texture, and composition. - Implementation: Load a pre-trained model and freeze its weights initially. You can fine-tune the model later if needed.

2. Feature Processing Module - Architecture: Implement an MLP to process the extracted features. This module will refine the features to better represent the outfit's aesthetic qualities. - Implementation: Use a 2-3 layer MLP with ReLU activation in the hidden layers. The output layer should have a single neuron for regression tasks.

3. Aesthetic Scoring Module - Architecture: This module uses the processed features to predict an aesthetic score. It can be a simple linear layer or another

MLP layer. - Implementation: Use a linear layer with a sigmoid activation function to output a score between 0 and 1, which can be scaled to a percentage.

Modular Design Considerations

- Modularity: Each module can be developed and tested independently, allowing for easier maintenance and updates. - Flexibility: Modules can be swapped with different architectures if needed. For example, you could replace the ResNet-50 with a more lightweight model like MobileNet for efficiency.

Example Code (PyTorch)

Here's a simplified example of how you might implement this architecture:

Implementation Tips

1. Data Preparation: Ensure your dataset includes diverse outfits with corresponding aesthetic scores. 2. Normalization: Normalize input images to ensure consistency. 3. Hyperparameter Tuning: Experiment with different learning rates, batch sizes, and number of epochs. 4. Validation Strategy: Use a validation set to monitor performance and prevent overfitting.

This modular design allows for flexibility and scalability, making it easier to refine and extend the model as needed.

— Answer from Perplexity: pplx.ai/share

—

3. Pose Estimation, Segmentation, and Clothing Type Recognition These techniques help identify and extract clothing items from images.

Strengths: - These tools enable the system to understand the individual components of an outfit. - They allow for the creation of realistic positive and negative samples.

Challenges: - Pre-trained models may struggle with occlusions, unusual poses, or low-resolution images. - Generating synthetic negative samples (shuffled outfits) might not always reflect real-world scenarios.

Suggestion: - Use pre-trained models for pose estimation (e.g., OpenPose), segmentation (e.g., Mask R-CNN), and clothing type recognition (e.g., fine-tuned EfficientNet). - For negative samples, apply controlled transformations (e.g., swapping colors, textures, or proportions) to ensure realism.

—

4. IDM-VTON for Virtual Try-On Using IDM-VTON to place segmented clothing items on a person is a clever way to simulate outfit combinations.

Strengths: - Virtual try-on ensures that generated outfits look realistic and contextually appropriate. - It adds an interactive element to the app.

Challenges: - IDM-VTON may not handle all body shapes or poses perfectly. - Computational overhead could be significant if used extensively.

Suggestion: - Use IDM-VTON sparingly, perhaps only for visualizing top-rated outfits. - Consider alternative lightweight virtual try-on solutions if computational resources are limited.

—

5. Siamese Network for Embedding Space Using a Siamese network to learn similarity-based scoring is an excellent choice.

Strengths: - Embedding spaces allow for meaningful comparisons between outfits. - Triplet loss can effectively cluster similar outfits and separate dissimilar ones.

Challenges: - Generating sufficient positive and negative pairs for training can be time-consuming. - The quality of the embedding space depends on the diversity of the dataset.

Suggestion: - Use pre-trained models like Fashion-CLIP to generate initial embeddings. - Fine-tune the Siamese network on your dataset using triplet loss or contrastive loss.

—

6. Learning Additional Features Incorporating features like color harmony, balance, contrast, and texture matching is crucial for evaluating outfit quality.

Strengths: - These features align with human intuition about fashion aesthetics. - They add depth to the model's understanding of what makes an outfit "good."

Challenges: - Extracting these features programmatically can be non-trivial. - Some features (e.g., fit analysis) may require 3D modeling or depth information.

Suggestion: - Start with simpler features like color harmony and balance, which can be computed using computer vision techniques. - Use pre-trained models or libraries (e.g., ColorThief) to analyze color palettes. - Gradually incorporate more complex features as the system evolves.

—

7. Rule-Based AI and Template-Guided Outfit Generation Incorporating rule-based systems and templates adds structure to the model.

Strengths: - Rules enforce hard constraints (e.g., avoiding clashing colors) and provide interpretability. - Templates guide the generation of coherent outfits.

Challenges: - Rules might not capture all nuances of fashion. - Templates could limit creativity and flexibility.

Suggestion: - Use rules as a starting point but allow the model to deviate based on learned patterns. - Combine rule-based systems with machine learning by using rules as priors or constraints during training.

—

8. 8-Point Rule of Fashion This is a creative way to quantify outfit balance.

Strengths: - Simple and intuitive. - Encourages balanced styling without being overly restrictive.

Challenges: - Assigning point values to clothing items might require domain expertise. - Users might interpret "zing and pep" differently.

Suggestion: - Use this rule as a heuristic to guide outfit creation or as a post-processing step. - Allow users to customize the point system based on their preferences.

—

9. Ensemble Modeling Using multiple pre-trained networks to extract diverse features is a strong strategy.

Strengths: - Combines the strengths of different models. - Improves robustness and generalization.

Challenges: - Increases computational overhead. - Requires careful fusion of features from different models.

Suggestion: - Start with a small ensemble of two or three models (e.g., one for global features and one for local details). - Use late fusion (e.g., concatenating outputs) or attention mechanisms to combine features effectively.

—

10. Final Output: Basic AI That Knows Fashion Styling Guidelines Your goal is to create a basic AI that understands fashion styling guidelines and can rate outfits.

Strengths: - The system incorporates a wide range of techniques and ideas, making it versatile and engaging. - It balances creativity with structure through the use of rules and templates.

Challenges: - Integrating all components seamlessly requires careful planning and testing. - Ensuring the system generalizes well to unseen data is critical.

Suggestion: - Start with a minimal viable product (MVP) that incorporates a subset of the ideas. - Gradually add complexity as you validate each component. - Leverage user feedback to refine the system over time.

—

Key Takeaways 1. Focus on Feasibility: Avoid computationally expensive tasks (e.g., training a GAN) and leverage pre-trained models wherever possible. 2. Prioritize Key Features: Begin with essential features like color harmony, balance, and similarity-based scoring before adding complexity. 3. Iterative Development: Develop the system incrementally, starting with a simple prototype and refining it based on feedback. 4. User-Centric Design: Ensure the final product is engaging and useful for users, with opportunities for personalization and interaction.

Your idea has great potential, and with careful implementation, you can create a unique and fun fashion AI! Let me know if you'd like me to dive deeper into any specific aspect of your concept.

4.2 Comparative Analysis of Solution Approaches

4.3 Justification for Selected Approach

For the task of compatibility, accuracy and AUC have been mainly utilized as shown by Wang et al. [56] Also, Papadopoulos et al. [46] for incompatibility detection of outfit fashion items, used the Mean Absolute Error (MAE) metric, in addition to accuracy and AUC in order to evaluate the performance of their model.

proposed a translation-based neural fashion compatibility model which contained three parts: (1) first mapped each item into a latent space via two CNN for visual and textual modality, (2) encoded the category complementary relations into the latent space, and (3) minimized a margin-based ranking criterion to optimize both item embeddings and relation vectors jointly.

5 Implementation

5.1 System Architecture and Technical Stack

5.2 Implementation of Selected Approach

5.3 Data Collection and Preprocessing

Data Collection: The first step is to gather labeled data, which typically consists of input features and their corresponding target labels. This data should be representative of the problem you want to solve. Data curation: The process of cleaning and organizing the collected data to ensure its quality and reliability. This step involves removing any outliers or inconsistencies, handling missing values, and transforming the data into a suitable format for training the model. Data Splitting: The collected data is usually divided into two subsets: the training dataset and the test data. Train the model with the training dataset, while the test data is reserved for evaluating its performance.

5.4 AI Model Development and Training

Model Selection: Depending on the problem at hand, you choose an appropriate supervised learning algorithm. For example, if you're working on a classification task, you might opt for algorithms like logistic regression, support vector machines, or decision trees. Training the Model: This step involves feeding the training data into the chosen algorithm, allowing the model to learn the patterns and relationships in the data. The training iteratively adjusts its parameters to minimize prediction errors with its learning techniques. Model Evaluation: After training, you evaluate the model's performance using the test set. Standard evaluation metrics include accuracy, precision, recall, and F1-score. Fine-tuning: If the model's performance is unsatisfactory, you may need to fine-tune its hyperparameters or consider more advanced algorithms. This step is crucial for improving the model's accuracy. Deployment: Once you're satisfied with the model's performance, you can deploy it to make predictions on new, unseen data in real-world applications.

Supervised learning reference Building an AI-Powered Outfit Recommendation System With Dataiku Smart Fashion Recommendation using ResNet50

6 Evaluation and Testing

6.1 Experimental Setup and Test Scenarios

Fashion Compatibility. Fill in the Black (FITB). Fashion Retrieval.

6.2 Performance Metrics and Evaluation Criteria

6.3 Results and Observations

Bibliography

[1] H.-J. Chen, H.-H. Shuai, and W.-H. Cheng, "A Survey of Artificial Intelligence in Fashion," *IEEE Signal Processing Magazine*, vol. 40, no. 3, pp. 64–73, May 2023, ISSN: 1053-5888, 1558-0792. DOI: 10.1109/MSP.2022.3233449. Accessed: Apr. 17, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10113373/.

[2] Y. Deldjoo et al., *A Review of Modern Fashion Recommender Systems*, 2022. DOI: 10.48550/ARXIV.2202.02757. Accessed: Apr. 17, 2025. [Online]. Available: https://arxiv.org/abs/2202.02757.

[3] E. Kouslis et al., "AI in fashion: A literature review," en, *Electronic Commerce Research*, Jun. 2024, ISSN: 1389-5753, 1572-9362. DOI: 10.1007/s10660-024-09872-z. Accessed: Apr. 16, 2025. [Online]. Available: https://link.springer.com/10.1007/s10660-024-09872-z.

[4] W.-H. Cheng, S. Song, C.-Y. Chen, S. C. Hidayati, and J. Liu, *Fashion Meets Computer Vision: A Survey*, 2020. DOI: 10.48550/ARXIV.2003.13988. Accessed: Apr. 17, 2025. [Online]. Available: https://arxiv.org/abs/2003.13988.

[5] S. Shirkhani, H. Mokayed, R. Saini, and H. Y. Chai, "Study of AI-Driven Fashion Recommender Systems," en, *SN Computer Science*, vol. 4, no. 5, p. 514, Jul. 2023, ISSN: 2661-8907. DOI: 10.1007/s42979-023-01932-9. Accessed: Apr. 16, 2025. [Online]. Available: https://link.springer.com/10.1007/s42979-023-01932-9.

[6] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery, 2019. [Online]. Available: https://books.google.at/books?id=DOamDwAAQBAJ.

[7] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*, eng, First edition. Sebastopol, CA: O'Reilly Media, 2017, ISBN: 9781449369415.

[8] X. Wang, B. Wu, Y. Ye, and Y. Zhong, "Outfit Compatibility Prediction and Diagnosis with Multi-Layered Comparison Network," 2019. DOI: 10.48550/ARXIV.1907.11496. Accessed: Apr. 17, 2025. [Online]. Available: https://arxiv.org/abs/1907.11496.

[9] M. Moosaei, Y. Lin, and H. Yang, *Fashion Recommendation and Compatibility Prediction Using Relational Network*, 2020. DOI: 10.48550/ARXIV.2005.06584. Accessed: Apr. 17, 2025. [Online]. Available: https://arxiv.org/abs/2005.06584.

[10] J. Liu, X. Song, Z. Chen, and J. Ma, "MGCM: Multi-modal generative compatibility modeling for clothing matching," en, *Neurocomputing*, vol. 414, pp. 215–224, Nov. 2020, ISSN: 09252312. DOI: 10.1016/j.neucom.2020.06.033. Accessed: Apr. 16, 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231220310043.

[11] M. Moosaei, Y. Lin, A. Akhazhanov, H. Chen, F. Wang, and H. Yang, "OutfitGAN: Learning Compatible Items for Generative Fashion Outfits," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 2272–2276, ISBN: 9781665487399. DOI: 10.1109/CVPRW56347.2022.00251. Accessed: Apr. 16, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9857247/.

[12] N. Zheng, X. Song, Q. Niu, X. Dong, Y. Zhan, and L. Nie, "Collocation and Try-on Network: Whether an Outfit is Compatible," en, in *Proceedings of the 29th ACM International Conference on Multimedia*, Virtual Event China: ACM, Oct. 2021, pp. 309–317, ISBN: 9781450386517. DOI: 10.1145/3474085.3475691. Accessed: Apr. 17, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3474085.3475691.

[13] W. Guan et al., "Partially Supervised Compatibility Modeling," *IEEE Transactions on Image Processing*, vol. 31, pp. 4733–4745, 2022, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2022.3187290. Accessed: Apr. 17, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9817021/.

[14] D. Kim, K. Saito, S. Mishra, S. Sclaroff, K. Saenko, and B. A. Plummer, *Self-supervised Visual Attribute Learning for Fashion Compatibility*, 2020. DOI: 10.48550/ARXIV.2008.00348. Accessed: Apr. 17, 2025. [Online]. Available: https://arxiv.org/abs/2008.00348.

[15] M. Dong, D. Zhou, J. Ma, and H. Zhang, "Towards Intelligent Design: A Self-driven Framework for Collocated Clothing Synthesis Leveraging Fashion Styles and Textures," 2025. DOI: 10.48550/ARXIV.2501.13396. Accessed: Apr. 17, 2025. [Online]. Available: https://arxiv.org/abs/2501.13396.

[16] K. Pang, X. Zou, and W. Wong, "Learning Visual Body-shape-Aware Embeddings for Fashion Compatibility," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 8041–8050, ISBN: 9798350318920. DOI: 10.1109/WACV57701.2024.00787. Accessed: Apr. 17, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10484117/.

[17] A. H. Vo, T. B. T. Le, H. V. Pham, and B. T. Nguyen, "An efficient framework for outfit compatibility prediction towards occasion," en, *Neural Computing and Applications*, vol. 35, no. 19, pp. 14 213–14 226, Jul. 2023, ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-023-08431-1. Accessed: Apr. 17, 2025. [Online]. Available: https://link.springer.com/10.1007/s00521-023-08431-1.

Statement of Affirmation

I hereby declare that all parts of this thesis were exclusively prepared by me, without using resources other than those stated above. The thoughts taken directly or indirectly from external sources are appropriately annotated. This thesis or parts of it were not previously submitted to any other academic institution and have not yet been published.

Dornbirn, July 2025 Viktoriia Simakova