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

CLIP Contrastive Language-Image Pre-training

CNN Convolutional Neural Network

DL Deep Learning

DeCNN 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

IDM-VTON Improving Diffusion Models for Authentic Virtual Try-On in the Wild

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

RelNN Relational Neural Network

ReLU Rectified Linear Unit

ResNet Residual Network

SSL Self-Supervised Learning

SAM Segment Anything

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 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. It outlines the motivation behind the research question, the objectives and the problem statement. The chapter concludes with a summary of the scope and expected outcomes.

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. AI models that assess outfits while factoring in individual features are still relatively unexplored. [1, cf.], [2, cf.]

2. Real-world outfits typically consist of multiple items such as shoes, accessories and more. However, existing methods assume a fixed input size and evaluate the compatibility between only two items (e.g. top and bottom) for a specific user. As a result, they are less capable of evaluating outfits with multiple or inconsistent item counts. [1, cf.]

3. Most models rely on predefined category labels to learn and evaluate outfit compatibility. But in real life, fashion items often belong to overlapping or unclear categories (e.g. hybrid pieces such as shirt-jackets or casual-formal dresses). There is a need for models that can learn compatibility based on visual, contextual or semantic features without depending on fixed category labels. [1, cf.]

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

This work tackles these challenges by introducing an approach to evaluate how good an outfit looks on a person, using photos of them wearing it. The proposed approach leverages existing DL techniques to assess the compatibility of an outfit. It can handle any number of clothing items and does not rely on fixed labels, allowing for optional integration of supplementary data.

To achieve this, the thesis on hand is structured around a few core steps. The first involves analyzing existing research and the DL models it employs. This is done to assess their relevance for outfit evaluation as well as to identify models of use to the presented use case. Consequently, the second step involves developing a concept of a solution while tackling the limitations and research gaps identified earlier. In the third step suitable models are selected and integrated into a pipeline. This pipeline processes input images and generates a numerical score that reflects the visual quality of the outfit. Afterwards, experiments are conducted to evaluate the effectiveness of the implementation alone as well as with incorporated data on contextual factors and personal preferences.

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. It includes experimenting with different types of ML models within 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 master's thesis include the development of a proof-of-concept prototype that demonstrates the feasibility of AI-driven outfit evaluation. The computational solution is capable of analyzing images of individuals wearing outfits and assigning a score that reflects the overall aesthetic appeal and visual coherence of the look. 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 analyse 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 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, cf.]
- Since each outfit consists of multiple complementary pieces (such as tops, bottoms, shoes, accessories), item compatibility spans across categories and involves complex interrelationships. [1, cf.]

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, cf.]), 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, cf.], [2, cf.], [3, cf.] and [4, cf.] and as illustrated in Figure 3.1.

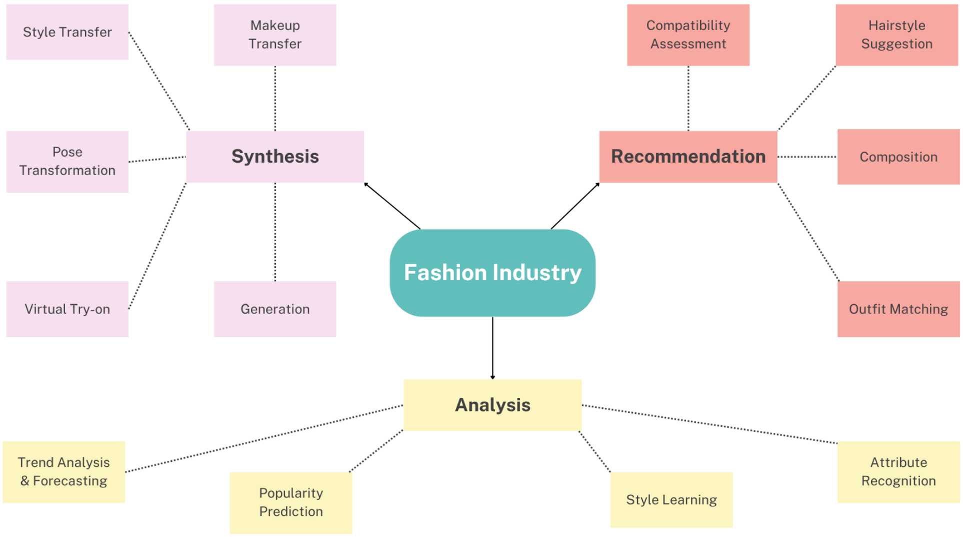

Figure 3.1: AI in Fashion (Mindmap)

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, cf.]

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 section explores key techniques, starting with foundational rules and progressing to computational models used in previous work.

3.1.1 Techniques within Fashion Theory

Compatibility evaluation among fashion items is typically modeled through fashion recommendation systems, which can broadly be divided into two categories: similar item recommendation and complementary recommendation. The former focuses on image retrieval techniques that suggest visually similar or identical fashion pieces. The latter aims to recommend items that complete or enhance a given outfit. [5, cf.]

Complementary Recommendation:

This type of system includes three strategies. First, item recommendation predicts a single missing item based on a specific category. Second, outfit recommendation or outfit completion involves assembling a full outfit from scratch or adding items to an existing one. Third, capsule wardrobe generation seeks to propose a minimal yet versatile collection of items that can be mixed and matched into various outfits.

Complementary recommendation can further be distinguished based on context (e.g. product-based, scene-based, occasion-based) and evaluation scope (e.g. pair-wise, list-wise, set-wise).

Context. Product-based methods assess the compatibility between items by analyzing their visual features, typically using product images. Scene-based approaches go a step further by incorporating contextual details such as the season, location or user-specific attributes. Finally, occasion-based recommendations tailor outfit suggestions to specific events, cultural settings or social contexts, ensuring relevance to the user's immediate needs.

Evaluation. In terms of structure: pair-wise evaluation looks at two items at a time, list-wise methods consider a sequence of items and set-wise approaches treat the outfit as a whole, analyzing how well all the pieces work together.

3.1.2 Techniques within ML

This section provides an overview of information that is foundational to this thesis. The methods are introduced both to establish a theoretical understanding and to contextualize their relevance within the review of related work and the implementation phase.

Computer Vision (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, cf.]

Deep Learning (DL):

In the context of this thesis, 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. [7, cf.]

Convolutional Neural Network (CNN). Some popular DL architectures include CNN, which process images by applying small, learnable filters over pixel regions. These filters extract local features such as edges or textures. As the image passes through multiple convolutional layers, the network builds a hierarchical representation. The deeper layers of the network are capturing increasingly complex and abstract features that are useful for tasks like classification or detection. [8, cf.]

To train a model, its parameters (weights and biases) are adjusted to reduce the loss, which measures the difference between predictions and true values. This is done using gradients, computed via backpropagation, which tells each layer how to update its parameters. In very deep networks, however, gradients can become extremely small as they are passed backward, causing earlier layers to learn very little. This is known as the vanishing gradient problem. Special architectures (e.g. ResNet) can address this by helping gradients flow more effectively during training. [9, cf.]

Residual Network (ResNet). One of the key CNN models in fashion is ResNet. Its key feature is the use of "skip/residual connections" across layers. These connections perform mappings that skip one or more layers and pass information directly to deeper layers. This network uses building blocks and lets each block learn a modification (residual) to its input, rather than the desired underlying function. Therefore, this model can handle very deep networks (with over 100 layers: e.g. ResNet-50, ResNet-101, ResNet-152), making it suitable for feature extraction from complex fashion datasets where high accuracy is required. [9, cf.]

Deconvolutional Neural Network (DeCNN). DeCNN inverts the operations of CNN to reconstruct high-dimensional data (e.g. image segmentation, object detection, image synthesis). It leverages deconvolutional layers and unpooling, recovers spatial details lost during downsampling and learns hierarchical representations critical for both low-level (e.g. deblurring) and high-level (e.g. segmentation) tasks. [10, cf.]

Recurrent Neural Network (RNN). A RNN is a class of neural networks designed to process sequential data, involving temporal or ordered data (e.g. language modeling or time-series prediction). It maintains a hidden state that captures information over time. However, the network often struggles with learning long-term dependencies due to issues such as vanishing gradients. [11, cf.]

Bidirectional Long Short-Term Memory (Bi-LSTM). A Bi-LSTM network is an advanced type of RNN that addresses the aforementioned limitations. It processes sequential data in both forward and backward directions and combines their outputs to capture full context at every step. This architecture is especially powerful for tasks where understanding both past and future elements in a sequence is important. [12, cf.]

Relational Neural Network (RelNN). A RelNN is designed to tackle relational reasoning tasks by explicitly modeling how objects in a set relate to one another, rather than focusing solely on their individual properties. Given a set of object representations (e.g. features extracted from image regions using a CNN, word embeddings from a sentence using a Bi-LSTM) the model evaluates all possible pairs using a small neural network to capture their interactions. These pairwise relations are then aggregated and passed through another network to generate the final output. The architecture's ability to model structured interactions makes it applicable to domains involving object relationships such as evaluating visual harmony in outfit composition. [13, cf.]

Siamese Network. A Siamese (Twin) Network is a neural network architecture designed for similarity learning. Its goal is to determine how alike or different two inputs of the same modality (e.g. images) are. This network consists of a pair of neural networks that share their weights and aim at computing similarity functions. It learns to compare pairs of inputs by mapping them into a feature space where similar samples are close and dissimilar samples are far apart. This approach is powerful in fashion scenarios where the task involves verifying whether two samples belong to the same class (e.g. similar style despite differences in type or color). [14, cf.]

Visual Semantic Embedding (VSE). These models enable cross-modal tasks such as image-text retrieval, caption generation or visual question answering. Their goal is to align representations across different modalities (e.g. images and textual descriptions) in a shared vector space. CLIP is a specific implementation of a VSE-like model including important advancements. FashionCLIP is domain-adapted version of CLIP which is fine-tuned and tailored specifically for fashion-related tasks. [15, cf.]

Multi-Layer Perceptron (MLP). MLP is a type of fully connected neural network composed of an input layer, one or more hidden layers with nonlinear activation functions and an output layer. While conceptually simple, MLPs are powerful function approximators and are commonly used in the final stages of many deep learning pipelines. In outfit recommendation or compatibility scoring tasks, MLPs are often employed to map high-level features that are extracted by earlier modules (e.g. CNN, RNN, RelNN) into a final score representing the compatibility between items. [16, cf.]

Generative AI. Generative AI refers to a class of DL models that can create new content (e.g. images, videos, music, text) by learning patterns from existing data. These models typically use self-supervised, unsupervised or weakly supervised techniques to capture the underlying structure of the data, rather than relying on fully labeled datasets. Depending on the approach, they may generate content unconditionally or be conditioned on input prompts or examples. A key feature of generative models is their ability to produce diverse and often realistic outputs. [17, cf.]

Generative Adversarial Network (GAN). A GAN is a framework where a generator and discriminator are trained in opposition. This results in a creation of highly realistic data. This process is widely used for image synthesis, data augmentation, text-to-image generation or style transfer. [18, cf.]

3.2 Previous Work and Related Research

Current section provides an overview of 10 papers that were choosen out of a total of 54. The selection process prioritized articles which are published within the last six years (from 2019 to 2025), written in English and are openly accessible. The search was conducted using specific terms such as "outfit compatibility", "fashion compatibility", "fashion recommendation", "deep learning", "aesthetic recommendation", "outfit generation" as well as a mix of these keywords. The most interesting articles based on their captions, abstracts and conclusions were subsequently downloaded and further investigated in terms of their relevance to the use case. For all selected studies, their main goal and important technical elements are summarized below.

remove hyperparameters later? check if all methods / techniques are explained earlier

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. The model 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. [19, cf.]

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, RelNN 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). [20, cf.]

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 downsampling, multi-modal fusion and upsampling. 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. [21, cf.]

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 a partial outfit (images) along with a specified missing clothing item category (textual) as input and creates several potential outfit combinations. A compatibility network (CNN + MLPs + RelNN) 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 minimax game for GAN. The prediction is evaluated using inception score, multi-scale structural similarity and compatibility score. [22, cf.]

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. [23, cf.]

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). [23, cf.]

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: [24, cf.]

1. It leverages a pre-trained model (ResNet-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 DeCNN 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, DeCNN depth: 5 transposed layers (output dimension: 256), MLPs depth: 2 layers (for each label with output dimension: 64), activation functions: LeakyReLU, ReLU, Tanh. [24, cf.]

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. [25, cf.]

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: [26, cf.]

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: [27, cf.]

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. [27, cf.]

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: [28, cf.]

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. [28, cf.]

4 Proposed Solutions

This chapter presents an overview of potential solution approaches relevant to the task of visual outfit evaluation within the field of CV. The main goal of the desired solution is to extract features from images, evaluate matching and compute overall compatibility score.

4.1 Overarching Framework

Based on an in-depth review of related literature, the specific use case and the available resources, a general direction for the solution has been defined: a GAN-inspired architecture is most suitable for addressing the challenges of this task.

The generator-like component acts as an outfit creator (template-guided or as a random combiner of clothing items). The discriminator-like component evaluates the quality of these outfits and assigns an aesthetic rating (1-10). The outfit creation process is then iteratively refined based on feedback (e.g. from the scoring model, from the user).

This approach simulates the functionality of a GAN without actually implementing a full network and avoiding the complexity of training it. It leverages existing pre-trained models and techniques in a creative way, making it both feasible and resource-efficient. However, without a true generator-discriminator loop, the carefull design of the interaction of generator-like and discriminator-like components is am important task.

4.2 Individual Components

While the overarching framework is clear, the implementation of individual components within this architecture can vary significantly. Therefore, this section explores a range of alternative methods for each key module, analyzing their respective strengths, limitations and applicability.

The different modules in the choosen strategy are illustrated in Figure 4.1 and described bellow.

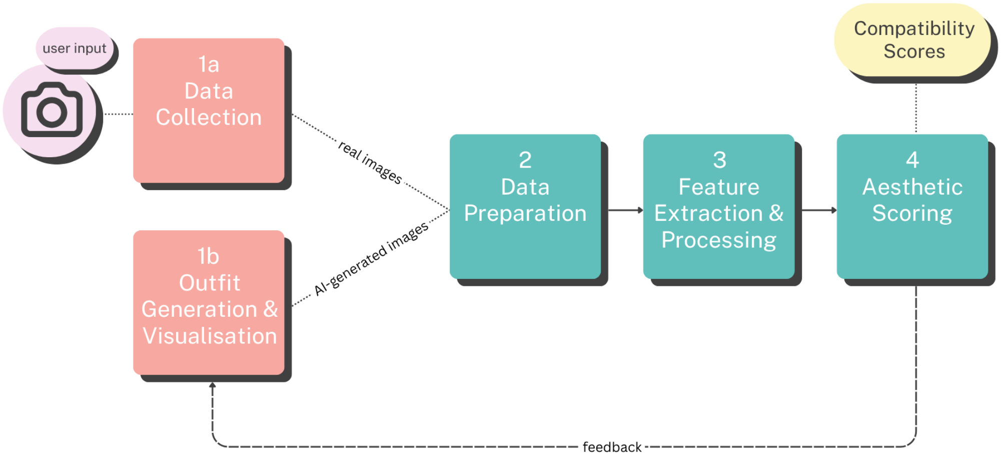

Figure 4.1: GAN-Inspired System and its Modules (Workflow)

4.2.1 Data Collection Module (1a)

The data collection strategy must align with the chosen method to learn data representations. In this work, two relevant approaches are considered: NC-SSL and SCL. These methods influence how data should be organized.

Positive Pairs (Non-Contrastive Self Supervised Learning (NC-SSL)). This approach that relies exclusively on positive sample pairs and data augmentations. These are well-balanced and stylish outfits, typically sourced from fashion experts or curated user sets.

Positive and Negative Pairs (Self Supervised Contrastive Learning (SCL)). This learning strategy uses both positive and negative pairs. While the positive examples remain the same as in the previous approach, negative samples consist of visually unappealing outfits (e.g. those with clashing colors, disproportionate silhouettes, excessive layering).

Comparative Analysis:

Both, positive and negative outfit data samples can be sourced from platforms like Pinterest and Google or from public datasets such as DeepFashion, Polyvore and FashionGen. While there are some positive outfit examples available, identifying truly representative positive and negative samples remains difficult. This is because the quality of an outfit is often subjective. This means, that what looks good to one person might not appeal to another. As a result, both positive and negative samples require human evaluation, ideally with personalization in mind. Randomizing outfit pieces can help generate negative examples, but the results are often unrealistic or too obviously mismatched. Moreover, many bad outfits are only subtly off, making them hard to label without contextual knowledge. Collecting a well-balanced dataset, therefore, demands time and careful curation.

Selected Approach incl. Justification:

Due to the challenges in obtaining reliable contrastive data the decision is made to begin with a non-contrastive training strategy, using only known good outfits. This allows the model to first learn what makes an outfit aesthetically and semantically coherent. Once the training is finished and the model can reliably score outfits using the discriminator-like component, the generator-like component creates outfit combinations and recieves scores. Out of these newly generated outfits, only high- or low-scoring examples are selected for further training. This self-bootstrapping strategy gradually improves the model without relying on manually labeled negative samples.

Current approach is also suitable for a possible future phase, where each user is able to provide a personalized outfit history containing both liked and disliked looks. This allows the model to fine-tune its understanding of style preferences on an individual level, using real user behavior as training input for personalized outfit scoring.

Furthermore, in later stages, additional data can be gathered through user-uploaded outfits, styled looks from partner brands, second-hand shops, or stylists, as well as previously AI-generated combinations as discribed in the module 1b.

4.2.2 Outfit Generation and Visualisation Module (1b)

A few approaches to generate and display fashion outfits are being explored:

Random Outfit Generation. As a baseline, outfits can be randomly sampled by selecting items across predefined categories (e.g. top, bottom, shoes, accessories). While these combinations are unstructured, they allow evaluation of the model's ability to detect outfit quality without prior structure.

Rule-Based and Template-Guided Generation. This approach imposes structural constraints on outfit creation, provides interpretability and guides the generation of coherent outfits. Fashion heuristics and predefined templates are used. For instance, the 8-Point Rule of Fashion offers a quantitative framework where each item contributes a defined number of "style points", promoting visual balance and coherence. These constraints can be applied during generation or as a post-processing filter to refine candidate outfits. Additionally, users can personalize the rule system, enabling flexible alignment with subjective fashion preferences.

Outfit Visualisation. One way to display an AI-generated outfit is by using IDM-VTON to place segmented clothing items on a person. This virtual try-on network ensures that generated outfits look realistic and contextually appropriate. However, it may not handle all body shapes or poses perfectly or lead to a computational overhead if used extensively.

Comparative Analysis:

text missing

Selected Approach incl. Justification:

Rules are used as a starting point but allow the model to deviate based on learned patterns. text missing

4.2.3 Data Preparation Module (2)

Both modules 1a and 1b feed raw images incl. optional metadata into an aesthetic scoring model. The data needs to be processed accordingly in the current module.

Image Data. To prepare images several preprocessing steps are involved. The process begins with object detection or clothing item recognition, which identifies garments using models such as YOLOv8, Detectron2, Grounding DINO or fine-tuned EfficientNet. Following detection, segmentation is applied to isolate clothing items from the background using techniques such as SAM, Mask R-CNN or SegFormer. Next valuable step is pose estimation and pose normalization, which helps to understand how items are worn and ensures consistent alignment (e.g. mapping a person to a standard pose). Applying this step would make the comparison between outfits more fair and easier (all images in the same pose = easier to judge fit and compatibility). In order to do pose estimation, tools like OpenPose, MediaPipe or HRNet can be used. Once the relevant clothing regions are identified, cropping and background removal is used to focus solely on the garments. This is done by leveraging segmentation results to crop the items and removing the background using models such as U2Net, RemBG or SAM. To ensure consistency in the dataset, images are then resized to a typical resolution (e.g. 224x224 or 512x512) and pixel values are normalized (e.g. to a [0, 1] range). Optionally, data augmentation is utilized to improve generalization.

Textual Data. When raw image data is accompanied by textual metadata such as descriptions, tyle, weather, event, gender, tags, captions or user comments, it is important to preprocess this information to ensure consistency and usefulness. The first step involves cleaning the text, which includes removing special characters, emojis, irrelevant hashtags and links, as well as normalizing whitespace and converting the text to lowercase. Next, the pipeline includes a text filtering stage to remove uninformative or noisy content. Once the text is cleaned and filtered, it undergoes tokenization, which depends on the model in use.

Comparative Analysis:

The mentioned data preprocessing steps are well-established and widely implemented. They might include minor variations depending on the specific requirements, but do not serve as a key differentiator among the considered approaches.

Selected Approach incl. Justification:

items of clothes (segmented) whole outfit person segmented
While the data preparation module remains largely conventional, its correct implementation is vital to ensure compatibility with the chosen pretrained models. The described preprocessing steps provide a solid foundation that supports effective model training. Any future experiments involving alternative preprocessing strategies can be explored without altering the fundamental structure of the data pipeline.

4.2.4 Feature Extraction and Processing Module (3)

Ensuring that the system learns meaningful relationships between clothing items and aesthetic ratings requires careful feature engineering. This module analyses cleaned images from the previous module and captures basic visual attributes such as color and texture. It also learns additional, more complex features, such as color harmony, texture matching, balance, contrast and composition. These features align with human intuition about fashion aesthetics. They add depth to the model's understanding of what makes an outfit "good" and are crucial for evaluating outfit quality.

This module adapts pre-trained models to the desired task. A pre-trained network (e.g. ResNet, a body shape-sensitive model) can be taken and continued to be trained on own data to extract complex features from outfit images. To do so, following techniques are considered: transfer learning and fine-tuning.

Transfer Learning. Transfer learning reuses knowledge from a source task to improve performance on a related target task. The pre-trained model serves as a feature extractor. Early layers which capture general patterns like edges or textures are typically frozen, while new task-specific layers are added and trained. This way, the model can adapt to new data patterns without requiring extensive retraining.

Fine Tuning. This approach takes a single pre-trained model and trains it further so that it is adapted to new data patterns, adjusting some or all of its weights. Thereby, the weights of the pre-trained models are updated during training on the target task. This may involve unfreezing and retraining deeper layers to capture task-specific nuances.

Ensemble Modeling. This is a strategy of using multiple pre-trained networks in order to extract diverse features. It combines outputs from multiple feature extractors and focuses on the strengths of different models, improving robustness and generalization. However, it may increase computational overhead and requires careful fusion of features from different models.

Additionally, this module refines the features to better represent the aesthetic qualities of an outfit. To process the extracted features, an MLP can be implemented.

Comparative Analysis:

Fine-tuning refines pre-trained models for niche tasks, while transfer learning repurposes their core knowledge with minimal adjustment. Fine tuning is suitable for a larger target dataset where the model can safely adapt without overfitting or when high task-specific accuracy is critical. It can modify the original model's knowledge more extensively than transfer learning. Transfer learning is ideal when the target dataset is small or the new task is similar to the source task. It requires fewer computational resources and less data because most model weights remain fixed.

text missing

Selected Approach incl. Justification:

The selected approach combines several ideas. First, multiple pre-trained models are taken and customized based on the collected data and the task of the outfit scoring. Later, an ensemble fusion model is fine-tuned to combine the features of the pre-trained models. This way, this module can be considered a fashion intelligence layer.

Using a pre-trained ResNet model (trained on xy) to xy by freezing its convolutional layers and training only a new classifier head. Taking a pre-trained xy model and updating its weights on outift data to improve performance on outfit analysis.

This choice is justified by the key benefits: performance, efficiency, scalability. Utilizing pretrained models significantly reduces training time and computational resources compared to training from scratch. Moreover, these models can be fine-tuned or extended to handle additional modalities or downstream tasks with minimal architectural changes.

4.2.5 Aesthetic Scoring Module (4)

This module uses the processed features to predict an aesthetic score. There are two options:

1. It can be a simple linear layer or another MLP layer to output a score between 0 and 1, which can be scaled to a percentage.
2. Siamese network can be used to learn similarity-based scoring. This approach allows for meaningful comparisons between outfits, clustering similar outfits and separating dissimilar ones. The quality of the embedding space depends on the diversity of the dataset.

Comparative Analysis:

text missing

Selected Approach incl. Justification:

text missing

, which evaluates the generated outfits using learned visual-semantic embeddings (e.g., from FashionCLIP) and learned similarity metrics.

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.

Score compatible with me Score compatible with every other piece I am wearing or could buy

compatibility clothing with me and clothing with eachother

relations (scores)

splitting it up? possibility explainable AI -> highlight the least compatible item

The final section provides a justification for the selected combination of techniques, aligning each choice with the project goals and practical constraints.

5 Implementation

The following sections detail the system architecture, technical stack and specific implementation choices.

5.1 System Architecture and Technical Stack

5.2 Implementation of Selected Approach

The implementation follows a step-by-step procedure. Initially, fundamental features such as color harmony, balance and similarity-based scoring are prioritized before introducing more complex elements. The system is developed incrementally, starting with a basic prototype and refining it progressively based on experimental results. This iterative process is essential for building a robust and effective solution. Ultimately, the final product aims to be engaging and user-friendly, offering opportunities for personalization and interaction.

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. - 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. - 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.

5.3 Data Collection and Preprocessing

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.

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.

segmentation masks and structured metadata

Textual Data.

After tokenization, the pipeline proceeds with semantic embedding extraction, transforming the processed text into vector representations. FashionCLIP is preferred for fashion-specific contexts, but alternatives such as CLIP, BERT, SBERT, Sentence-T5 or GloVe can be used. Optionally, metadata fusion can be applied by converting the attributes into descriptive prompts (e.g. "A red wool sweater with a relaxed fit") or encoding them as key-value pairs for structured input formats.

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.

Fine-Tuning Techniques

Human Pose Estimation reference

SAM category-guided attention mechanisms

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

7 Discussion and Impact Analysis

7.1 Interpretation of Results

7.2 Limitations and Challenges

7.3 Ethical Considerations and Sustainability Implications

8 Conclusion and Future Work

8.1 Summary of Findings

8.2 Future Research Directions

explore the possibility of implementing different types of feedback (Explicit + Implicit -> Rating Matrix). Mastering Recommendation System: A Complete Guide

Additionally, insights gained during the experimentation phase will be documented, particularly regarding the trade-offs between simplicity (single-input systems) and complexity (multi-input systems). These findings are anticipated to inform future research directions in AI and fashion, particularly in areas such as recommendation systems, virtual styling assistants, and augmented reality try-ons.

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.

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] K. O’Shea and R. Nash, *An Introduction to Convolutional Neural Networks*, arXiv:1511.08458, Dec. 2015. DOI: 10.48550/arXiv.1511.08458. Accessed: Jun. 19, 2025. [Online]. Available: http://arxiv.org/abs/1511.08458.

[9] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385, Dec. 2015. DOI: 10.48550/arXiv.1512.03385. Accessed: May 21, 2025. [Online]. Available: http://arxiv.org/abs/1512.03385.

[10] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2528–2535. DOI: 10.1109/CVPR.2010.5539957.

[11] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," 2018. DOI: 10.48550/ARXIV.1808.03314. Accessed: Jun. 19, 2025. [Online]. Available: https://arxiv.org/abs/1808.03314.

[12] Z. Huang, W. Xu, and K. Yu, *Bidirectional LSTM-CRF Models for Sequence Tagging*, arXiv:1508.01991, Aug. 2015. DOI: 10.48550/arXiv.1508.01991. Accessed: May 25, 2025. [Online]. Available: http://arxiv.org/abs/1508.01991.

[13] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, *Learning to Compare: Relation Network for Few-Shot Learning*, arXiv:1711.06025, Mar. 2018. DOI: 10.48550/arXiv.1711.06025. Accessed: May 21, 2025. [Online]. Available: http://arxiv.org/abs/1711.06025.

[14] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," 2015.

[15] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, *VSE++: Improving Visual-Semantic Embeddings with Hard Negatives*, arXiv:1707.05612, Jul. 2018. DOI: 10.48550/arXiv.1707.05612. Accessed: May 21, 2025. [Online]. Available: http://arxiv.org/abs/1707.05612.

[16] upGrad, *An overview on multilayer perceptron (MLP) in machine learning*, Jun. 2023. Accessed: May 21, 2025. [Online]. Available: https://www.upgrad.com/blog/multilayer-perceptron-mlp-in-machine-learning/.

[17] S. S. Sengar, A. B. Hasan, S. Kumar, and F. Carroll, "Generative artificial intelligence: A systematic review and applications," en, *Multimedia Tools and Applications*, Aug. 2024, ISSN: 1573-7721. DOI: 10.1007/s11042-024-20016-1. Accessed: Jun. 19, 2025. [Online]. Available: https://link.springer.com/10.1007/s11042-024-20016-1.

[18] I. J. Goodfellow et al., *Generative Adversarial Networks*, arXiv:1406.2661, Jun. 2014. DOI: 10.48550/arXiv.1406.2661. Accessed: May 25, 2025. [Online]. Available: http://arxiv.org/abs/1406.2661.

[19] 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.

[20] 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.

[21] 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.

[22] 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/.

[23] 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.

[24] 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/.

[25] 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.

[26] 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.

[27] 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/.

[28] 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