Learning Style Compatibility on Fashion Data

LUKAS FRÖSSLUND

Masters Programme in Computer Science
Date: June 20, 2021

Supervisors: Somayeh Aghanavesi, Therese Persson
Examiner: Marco Chiesa
School of Electrical Engineering and Computer Science
Host company: Sellhelp AB
Swedish title: Lärande av Stilkompatibilitet på Modedata

Abstract

Fashion Recommendation can be defined as a set of systems that tries to predict and retrieve a curated and often ranked selection of fashion items based on the preference of one or more target consumers. Traditional systems relied on providing substitute recommendations, meaning that they were centered around finding similarities between fashion items. However, recent approaches have aimed to provide complementary recommendations that are instead built on item compatibility. Outfit Matching has recently emerged as a popular task when modelling compatibility between fashion items. The objective of the task is to retrieve a set of fashion items, each of a different item category, such that the items collectively can be considered visually compatible. In this thesis, two state-of-the-art deep neural network models, earlier used for the task of matching outfits, were implemented to investigate their performance on a novel task of matching fashion styles. This more unconstrained task accepted duplicate item categories as well as mixed demographics, enabling the retrieval of a larger and more diverse selection of fashion items.

A fashion dataset was constructed for the thesis, where the two models were evaluated on the data using the Fill-in-the-blank (FITB) experiment commonly used in fashion compatibility modelling. Additionally, an item retrieval test was conducted, evaluated using recall @ top k to determine the ability of the models to learn style compatibility in a retrieval setting. Results showed that both models struggled when introducing fewer constraints, with an FITB accuracy of 48.97% when matching fashion styles, compared to 63.73% on the outfit matching task. However, an increase in the embedding dimension of the data yielded a significant increase in accuracy. When performing experiments using previously unseen classes of data, no significant decrease in performance was noted, suggesting an ability in both models to generalize well to new fashion styles. Retrieval tests could show a clear preference in both models to retrieve relevant items, with recall values reaching 54.30% for a k-value of 50.

Suggestions for future work include efforts to be put on improving shortcomings in the data by ensuring all samples to be distinct in style, and as well to move beyond solely visual data and include semantic textual data in the embedding representation. Finally, the construction of a benchmark dataset for style compatibility modelling would be beneficial in drawing attention to the task.

Sammanfattning

Moderekommendationssystem kan definieras som en uppsättning system som försöker förutsäga och hämta ett rankat urval av modeprodukter baserat på en eller flera målkonsumenters preferenser. Traditionella system förlitade sig på att ge ersättningsrekommendationer, vilket innebar ett fokus kring att hitta likheter mellan produkterna. De senaste systemen har dock ämnat till att ge kompletterande rekommendationer som istället bygger på produktkompatibilitet. Outfitmatchning har nyligen etablerat sig som en populär uppgift vid modellering av kompatibilitet mellan modeprodukter. Syftet med uppgiften är att hämta en uppsättning modeprodukter, var och en av skilda produktkategorier, sådant att produkterna tillsammans kan anses vara visuellt kompatibla. I denna uppsats implementerades två djupinlärningsmodeller, som tidigare användes för att matcha outfits, till att nu undersöka deras prestanda på en ny uppgift som ämnade till att matcha modestilar. Denna mindre begränsade uppgift accepterade dels duplicerade produktkategorier samt även blandad demografi bland produkterna, vilket möjliggjorde hämtning av ett större och mer varierat urval av modeprodukter.

Ett modedataset konstruerades för uppsatsen, där de två modellerna utvärderades på data med Fill-in-the-blank (FITB) experimentet som vanligtvis används vid modellering av kompatibilitet. Dessutom genomfördes ett objekthämtningstest, utvärderat med hjälp av recall @ top k för att bestämma modellernas förmåga att lära sig stilkompatibilitet i ett hämtningsscenario. Resultaten visade att båda modellerna hade problem att hantera färre begränsningar, med en FITB-noggrannhet på 48,97% vid matchning av modestilar, jämfört med 63,73% för outfitmatchning. En ökning av bilddatans inbäddningsdimension gav emellertid en signifikant ökning av noggrannheten. Vid experiment med tidigare osedda dataklasser noterades ingen signifikant minskning av prestanda, vilket tyder på en förmåga i båda modellerna att generalisera väl till nya modestilar. Hämtningstestet påvisade en tydlig preferens hos båda modellerna för att hämta relevanta produkter, med återkallningsvärden upptill 54,30% för ett k-värde på 50.

Förslag för framtida uppsatser inkluderar föbättringar i datan genom att se till att alla par av datapunkter är tydliga i stil, samt att gå bortom enbart visuell data och även inkludera semantisk textinformation i inbäddningsrepresentationen. Slutligen skulle konstruktionen av ett referensdataset för modellering av stilkompatibilitet vara till nytta för att vidare uppmärksamma uppgiften.

Acknowledgments

Todo

Stockholm, June 2021
LUKAS FRÖSSLUND

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Simply defined as the style of clothing and accessories worn by individuals and groups of people at any given time [8], fashion plays a large role in shaping the cultures and social structures that surround us [9]. The fashion industry, which constitutes the manufacturing and selling of clothes, is today the third-largest industrial sector globally, trailing only electronics and automotive manufacturing [10]. A large portion of fashion sales happens in online markets, a portion that is expected to grow rapidly from $545 billion globally in 2019 to an estimated $713 billion in 2022 [11]. The growth of the fashion industry in recent times, which has led to an approximate doubling in clothing production between the years 2000 and 2015 [12], can be attributed mainly to the rise of the fast fashion phenomenon, an accelerated fashion business model characterized by quick turnarounds, low prices and often subpar material quality [13, 14]. This economic model, which not only has led to a dramatic increase in production but also a steady decrease in the utilization of clothes [12], has put an enormous environmental and social cost on fashion.

Today, textile production is a larger contributor to climate change than shipping and international aviation combined [13]. Extraction of large amounts of non-renewable resources, massive consumption of freshwater, and the creation of chemical waste and pollution are just some of the devastating through-puts of the fashion industry. The under-utilization of clothing and the lack of recycling additionally puts large economic restraints on the industry, with an estimated $500 billion in lost value every year due to this [14]. Furthermore, the social impact of the fashion industry and fast fashion is dire, with horrible working conditions for garment workers [15].

The need for a new circular economic system within fashion and textiles is drastic. A system built on restoration rather than wastefulness. Commitments have recently been made to identify decarbonization pathways in the fashion industry to meet science-based targets under the Paris Agreement [16, 17], pathways that point to the need for fundamental and drastic changes within the industry. As one of the keys to achieve a circular fashion industry, the resale of used fashion goods has been identified as an efficient and promising pathway towards sustainable fashion consumption [18]. In 2019, the global resale market grew 25 times faster than the general retail sector [19], with an emphasis on the online second-hand and consignment market.

1.1 Background

Sellpy is an online consignment store focused on the resale of any used goods, but predominantly clothing and fashion accessories. Different from an ordinary online retail store, resale of used goods have to consider each and every single item as a unique product and hence cannot keep multiple stock of one individual item. This often results in these types of online stores carrying huge amounts of products in their selection, which is the case for Sellpy. While this massive selection promotes a certain degree of diversity in the available fashion products, it can be bothersome for anyone browsing the website to find stand-out items for their personal needs, despite effective filters and search functions. This unfortunate downside suggests a need for well-functioning recommendation systems and general guiding.

Sellpy has dealt with the above-mentioned problem in several ways, one of them being the establishment of collections of items that adheres to the same theme or style. The vast majority of these, referred to as *stylistic collections*, contain solely, or almost exclusively, clothing and fashion accessories. Clothing items residing in one or more of these collections have been sold in higher capacity, implying that they provide an effective method of recommending relevant items for a selection of customers. However, the utilization of them has been limited due to the inconvenience of the manual curation and maintenance necessary.

As one of several areas where fashion meets computer vision, fashion recommendation can be broadly defined as a set of systems and tasks that can mitigate the problem of choice overload [20] by retrieving a curated selection of fashion items estimated to be the most appealing to one or more

target consumers [21]. Based on one or more source items, recommendations can be categorized as either substitute or complementary [22]. Substitute recommendations are built on the notion of similarity, meaning that recommended items will be similar to the source items, while complementary recommendations offers compatible items to the source items (for example retrieving a bag that goes well with a pair of jeans), and hence are built on the notion of compatibility.

Traditional fashion recommendation systems relied on providing substitute recommendations of clothing garments and fashion accessories, but recently the construction of systems that can provide complementary recommendations, and hence provide compatible garments, have gained significant interest [7, 23, 24, 25]. This recent upsurge in fashion compatibility modelling can be mostly attributed to a core problem within fashion recommendation which is the outfit matching task [26]. The core objective of this task is the effective retrieval of a collection of fashion items and clothing garments of different types that collectively can be considered visually compatible, which implies that the items share a similar style [7]. Additionally, the collection should constitute a complete outfit without duplicate types. State-of-the-art methods within outfit matching for fashion compatibility modelling are based on deep neural networks and computer vision [7, 23, 24, 25].

This thesis aims to investigate if recent state-of-the-art methods within outfit matching in fashion compatibility modelling can be utilized for the retrieval of selections of clothing garments and fashion accessories each adhering to a similar style beyond the type constraint of outfit matching, meaning that this altered and more unconstrained task accepts duplicate types and hence aspires to discover compatibility in larger collections of items.

1.2 Problem

1.2.1 Original Problem and Definition

As a provider for styling suggestions, a solution for excessive amounts of choices and a potential source for business revenue [27], recommendation systems capable of retrieving appealing items for target consumers are highly desirable for fashion e-commerce sites, including online consignment store and collaborator of this project Sellpy. As a part of their process of providing recommendations to consumers and to mitigate the problem of choice overload

in their diverse selection of almost one million unique fashion items, smaller collections of items all adhering to a similar style have been created. These collections denoted as *stylistic collections*, have however been constrained by the current need of constructing them in a manual fashion. Current recommendation systems at the company fair well in providing substitute recommendations, meaning that they excel in retrieving similar items. They are not however built to provide complementary recommendations built on the notion of compatibility, which is necessary to retrieve the diverse selections needed for the establishment of highly functional stylistic collections.

1.2.2 Research Questions

This project aims to remedy the above-mentioned constraints by investigating current state-of-the-art methods within fashion compatibility modelling and outfit matching to determine their effectiveness in automating the construction of stylistic collections by performing automatic complementary item retrieval based on compatibility by style. With this investigation, the aspiration is to get a deep understanding of the relationship between the outfit matching task and this slightly altered and more unconstrained novel task of stylistic compatibility modelling with duplicate types. The research questions investigated in this thesis are therefore the following:

How well can state-of-the-art deep learning methods for the outfit matching task be adapted to a more unconstrained style compatibility task, where a style constitutes an undefined number of items with no constraint on duplicate item types?

How well do deep learning and computer vision models constructed for the outfit matching task on public benchmark datasets perform on a newly assembled industrial fashion dataset of clothing garments and fashion accessories?

1.3 Purpose

By altering and extending the outfit matching task, and incorporating a newly assembled industrial fashion dataset of visual as well as categorical data, a main aspiration of this thesis is to make valuable contributions to the research fields of fashion compatibility modelling and fashion recommendation within deep learning and computer vision. Additionally, the ambition is to be granted a deep understanding of the relationship between the outfit matching task and

the altered task presented in this thesis of matching fashion items without the item type constraint. With the assembling of the new Sellpy dataset, the aim is as well to gauge the robustness of state-of-the-art deep learning models for the outfit matching task, by training and testing them on novel data.

Alongside the above mentioned academic purposes, a more practical ambition of this thesis is to leverage an end-to-end model capable of providing complementary recommendations and successfully perform retrieval of compatible items of duplicate types to curate stylistic collections of second-hand clothing garments and fashion accessories. With this, in a more general sense, the hope is to make a contribution to fashion resale and possibly provide means to simplify the navigation of online resale and consignment stores, and mitigate the problem of choice overload prevalent on these sites. The motivation to strive for development in the online fashion resale sector is undeniable, as it provides one of the most promising pathways towards sustainable fashion consumption [19]. Moreover, fashion resale has a bright future with an expected significant increase in market share as well as an economic growth far exceeding that of traditional retail in coming years [19].

To reach the purposes of the thesis and achieve the desired change, several concrete goals are set. To discover and evaluate the correlation between the outfit matching task and the altered task presented in this thesis, state-of-the-art models will have to be implemented, investigated, and compared using appropriate and fair evaluation methods. A well-functioning dataset will have to be constructed to fit the models in question and to accomplish the purpose of measuring the robustness of the implemented models.

1.4 Research Methodology

The structure of this thesis project, and the steps with which to carry out the intended research methodology had inspiration drawn from Walliman [28] and Höst et al. [29]. At the initial stages of the project, the research problem was formulated and objectives of the thesis were clearly defined along with goals and purposes. This process was in large carried out with the host company, as it was their problem that laid the foundation for the original problem definition of the thesis, and from there shaped the subsequent steps of conducting research among related works and formulating appropriate research questions.

Once the problem formulation, research questions, and the surrounding body

of related research were clearly identified, a comprehensive literature study was conducted to gauge a deep understanding of the research area, the specific problems that were to be tackled, and as well to make appropriate decisions on method choices. Concurrently with the literature study, the practical implementation was initiated, which in the first stages consisted of exploratory data analysis and general data collection. This was followed by the construction of various neural network models together with well-functioning datasets. Experimental analysis was carried out on the constructed networks, where details and parameters of the models were tweaked for optimal performance and comparative fairness until main experiments yielded results which were then compared, analyzed, and thoroughly discussed. As the analysis and discussion concluded, final conclusions could be drawn.

1.5 Delimitations

This thesis aims to reimplement state-of-the-art deep learning models for purposes and goals described in section 1.3, rather than the conception of original ones. Novelty within the thesis will instead be reached by extending the outfit matching task, and by the establishment of an industrial fashion dataset of second-hand clothing garments and fashion accessories. The number of reimplemented models will as well be limited as the project is individual and there are not enough resources to implement more. In terms of data, the host company Sellpy will provide the project with all necessary raw visual and textual data. The data will have to be processed and assembled into an appropriate dataset, but there won't be any need for an additional gathering of raw data.

1.6 Structure of The Thesis

The project report of this thesis is arranged in seven chapters, including the introductory first chapter. Chapter 2 gives a theoretical background on the broad surrounding areas of deep learning and computer vision needed to grasp the problem presented in the thesis. Chapter 3 introduces works from related research areas, including descriptions of state-of-the-art networks selected for re-implementation. Subsequently, chapter 4 gives an overview of all method choices in the project, including data, models, and evaluation tests, as well as the experimental steps taken to perform the evaluation. Chapter 5 presents the results from said experiments, which are later discussed in chapter 6. The

final seventh chapter presents conclusions from the thesis, including thoughts on future research and general reflections. A complete list of references is provided towards the end of the report.

Chapter 2

Background

This chapter offers a theoretical background for the areas of deep learning and computer vision that are of interest in this thesis. A background on Convolutional Neural Networks, including their various layer types, will be presented. In addition, the concerned paradigms of transfer learning, as well as deep metric learning, will be introduced, with specific presentations of methods, network types, and cost functions utilized in this project.

2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN's) are a common specialized class of artificial neural networks that processes data which is represented by a grid-like, or tensor-shaped topology [1], most commonly image data which can be thought of as a 2-D grid of pixel values. Originally applied to speech recognition in time-delay networks [30] and document reading [31], CNN's are today widely applied both commercially and academically in most areas of computer vision including recommendation tasks. The name derives from the mathematical convolution operation, a kind of linear operation applied in CNN's. As such, a CNN can simply be described as a neural network that utilizes convolution operations in place of general matrix multiplication in one or more of its layers. While the exact configuration and layering of an individual convolutional network can vary, most high-performing architectures are composed of the following building blocks.

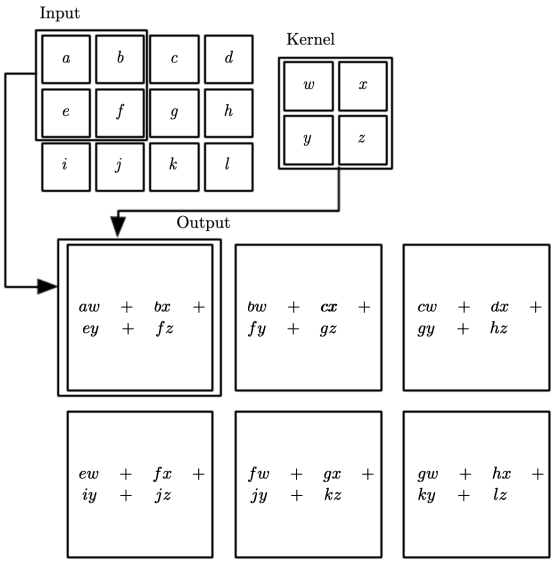


Figure 2.1: Example of convolution operation, from [1]

2.1.1 Convolutional Layers

The convolution operation takes place in layers titled convolutional layers. In a given layer of this type, the tensor input is abstracted to a feature map by passing a smaller filter, which in the case of image data could be thought of as a two-dimensional tensor of weights, across the complete input volume. As the filter is passed through the input, i.e. the pixels of the image, a sum of the element-wise product of each corresponding filter-sized patch of the input is computed. The size of the filter varies but is customarily much smaller than the dimension of the input. Another varying parameter of the network, known as the stride, determines the number of steps that the filter moves at each iteration of the procedure. A powerful result of using CNN's is that filters are learned during training, rather than being crafted manually as they were in traditional image processing methods.

2.1.2 Pooling Layers

A core issue with the feature maps that are extracted from a single or a series of convolutional layers is the high sensitivity of features in the input with regards to location. To combat this issue, downsampling of the feature maps has been noted to be effective and also reduces the computational complexity of the

architecture. This downsampling is typically performed in pooling layers, which essentially infers a summation of patches in the feature maps. The by far most common operations to be performed on these patches is to select either the maximum value of the patch or the average of all values in the patch, hence the specialized layer names max pooling layers and average pooling layers. The filter size as well as the stride is allowed to vary, but is in almost all cases for image data a 2x2 filter with a stride of 2, meaning that the feature map will be halved in size by passing it through a pooling layer.

2.1.3 Deep Residual Learning

While the original CNN's of the early and late 1990s displayed academic success and were utilized commercially, for example the LeNet-5 architecture used for character recognition [30], their applications were still limited. It wasn't until 2012, with the introduction of a deep CNN model known as AlexNet [32], that a newfound interest for convolutional networks was sparked. This paper would also be seen as the marked beginning of the deep learning dominance in computer vision. AlexNet initiated several of the modern standards of deep CNN's, including the use of the Rectified Linear Unit activation function (ReLU) after convolutional layers in the architecture as well as utilizing max pooling rather than average pooling.

After AlexNet, the development and research on deep CNN's suggested that the depth of the networks was of crucial importance in learning enriched levels of features and thus performing better [33, 34]. However, it was noted that new challenges emerged when simply stacking more layers and increasing depth. The issue of vanishing and exploding gradients in the backpropagation [35, 36] became more prevalent. While this did hamper convergence, it had already been largely dealt with by normalizing at initialization and in intermediate layers of the architecture, a technique known as batch normalization [37]. More critically it was recognized that by increasing the depth above a certain threshold, a notable degradation of the accuracy occurred which proceeded to raise the training error, suggesting that the degradation was not due to overfitting the network [38].

To solve the degradation problem, a deep residual learning framework was presented [2]. The idea was to use shortcuts known as skip connections which skip over layers in the architecture, creating building blocks knows as residual blocks where the output of a given layer is added to a layer deeper in the block

and thus making the skip connection. With this approach, the fundamental idea was to learn a different residual mapping F(x), hypothesized to be easier than the original underlying mapping H(x). As the residual mapping can be defined as the difference between the underlying mapping and the input (F(x) := H(x) - x), the principle of the framework relied on optimizing F(x) and later obtain the underlying mapping by adding the original input at the skip connection. By stacking residual blocks, a deep CNN known as ResNet was created, an architecture that managed to reach a much larger depth than previously without suffering from the aforementioned degradation problem.

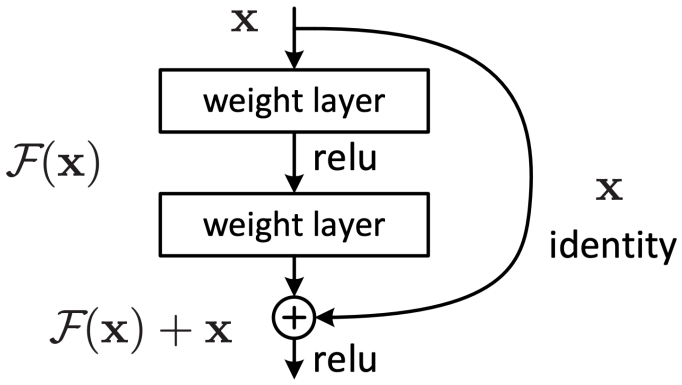


Figure 2.2: Residual blocks used in the ResNet architecture presented in [2].

2.2 Transfer Learning

Most of the high-performing neural network architectures in computer vision and deep learning today require vast computing resources due to their large size and immense number of learnable parameters. As such, a full training procedure is often infeasible. Transfer learning overcomes this by utilizing a model with weights pre-trained in a different domain, and then fine-tune the weights or a subset of the weights on the domain-specific data for the task at hand [39]. A common approach is to fine-tune only the final layers of the architecture, as there has been noted that lower level features are often shared among different sets of visual categories [1].

2.3 Deep Metric Learning

In most ranking scenarios, which include recommendation and retrieval systems, the training revolves around learning a distance or similarity function across data objects, rather than classifying the output. This approach, which often operates in a weakly supervised setting, is commonly denoted as metric learning. The motivation is to learn a mapping that places samples deemed as similar from the underlying distribution close in some kind of output embedding space, and vice versa. An assumption that has been held true is that deep hierarchical model architectures, which include deep convolutional networks, can extract useful representations of data [40]. In regular classification tasks, these representations are simply side effects, which is in contrast to metric learning where the representations and the extraction of them are explicitly desired. The coalescence of the metric learning approach and the use of deep model architectures and the extraction of useful embedded data representations is known as deep metric learning.

To leverage the deep metric learning approach effectively, multiple aspects need to be considered. The actual metric, with which similarity between representations will be measured, needs to be explicitly defined. A simple euclidean distance is the most common for most applications. Additionally, an appropriate network architecture is of importance, which will be further discussed in section 2.3.1. Finally, a loss function whose objective is to predict the relative distance between inputs based on the chosen metric, rather than predicting a label directly, is crucial. A set of these losses, which combined often goes under the term ranking losses, will be further explored in sections 2.3.2 and 2.3.3 respectively.

2.3.1 Siamese Networks

Initially applied for the task of signature verification [41], Siamese Neural Networks are today a typical choice of architecture for most metric learning problems. The fundamental idea of siamese networks is to share weights across multiple inputs and compare the generated output representations using an appropriate metric. As such, siamese networks can be thought of as containing two or more identical sub-networks, where the updates of learnable parameters are replicated for each sub-network [3].

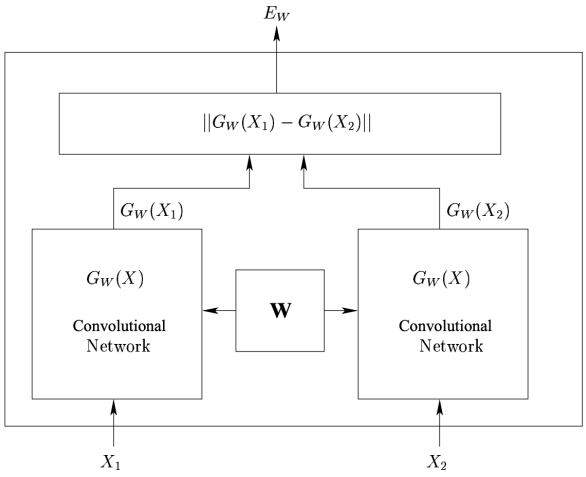


Figure 2.3: Example of a Siamese CNN using two identical sub-networks with shared weights. Image taken from [3].

Because learning in siamese networks involves comparison of multiple input samples, a ranking loss function is necessary [42]. Additionally, the aspiration is that the loss trains the network through modifying learnable variables in a way that draws the representations of similar inputs closer to each other, and vice versa. Various ranking losses for siamese networks subsists, with different numbers of output representations compared in several distinctive constellations.

2.3.2 Triplet Loss

In siamese networks where triplets of input samples are contrasted in their respective output embedding space, often referred to as triplet networks, the triplet loss is used [6, 24, 43, 25, 5]. In it, each one of the three inputs plays their distinct part in the dynamics of the loss. The anchor sample a represents the baseline of the function, where it shares the label or some kind of similarity with another sample, which is denoted as the positive sample p. The last component is the negative sample n, which does not share the manner of similarity of the other two samples and is therefore considered to be dissimilar to both of them. As an example, in the case of using siamese networks of triplets for face identification [4], the anchor sample a and positive sample p accommodates different images of the same person, while the negative sample n contains an image of a different person.

The triplet loss enforces itself such that it aims to minimize the distance $d(a, p)$ between the samples conceived to be similar, i.e. the anchor a and the positive p, while simultaneously maximizing the calculated distance $d(a, n)$ between the anchor and the negative sample n. Typically, this distance is the euclidean distance calculated in the output embedding space of the respective samples. The function is defined as follows

$$L(a, p, n) = max(d(a, p) - d(a, n) + \mu, 0)$$

where μ is a slack variable denoted as the margin, which reinforces and pushes the negative distance $d(a, n)$ to be greater than the sum of the positive distance $d(a, p)$ and the margin μ. Without the margin, there is a risk of converging to the trivial solution of allowing both distances to be zero.

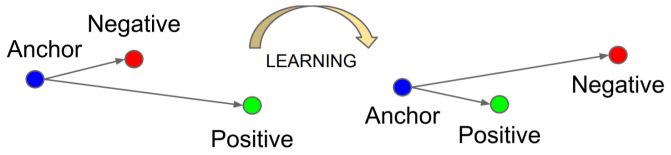


Figure 2.4: Triplet loss function. Image taken from [4].

2.3.3 Outfit Ranking Loss

For many tasks in a multitude of domains, the triplet loss has proven to be effective in siamese networks on metric learning problems. However for certain tasks, extensions of the triplet loss have been implemented to further drive the efficacy of the learning process. For the task of person re-identification, i.e. identifying people across different cameras in a wide area video surveillance setting, the quadruplet loss was proposed [44]. Contrasted with the traditional triplet loss, this configuration included a second negative instance sampled from a different label compared to the first negative, and as well an additional accompanying margin parameter.

In the fashion compatibility modelling domain, a scarce number of alternatives to the triplet loss have been proposed. One of them is the outfit ranking loss, proposed in [5]. The loss was suggested for the outfit matching task to operate

on entire outfits of garments rather than a subset of an outfit, as is the case for the triplet loss, and thus more efficiently leverage the relationship in an entire outfit and prove more appropriate for compatibility modelling. The idea is to sample multiple anchor instances, which together constitutes the outfit O, and in addition sample one positive instance p and a set of negative instances N. Similar to other ranking losses, the outfit ranking loss operates on the notion that the positive sample p expresses some form of closeness, for this task a measure of compatibility, to the anchor samples, which in this specific case includes each item in the anchor outfit O. This is contrasted with the set of negative samples N, which are not considered to be compatible with any of the other instances in the complete training samples $\Upsilon = \{O, p, N\}$.

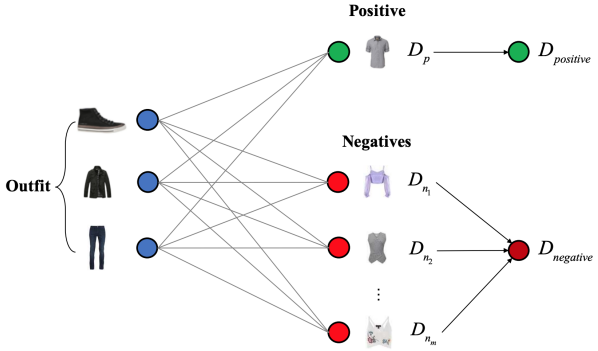


Figure 2.5: Outfit ranking loss function, presented in [5].

Fundamentally, the outfit ranking loss operates on the same principles as the triplet loss, by pushing a negative distance D_n further away than the sum of a positive distance D_p and a margin parameter μ. The main difference between the two lies in the process and calculation of these respective distances. Consider a complete training sample to consist of three components, which includes the outfit $O = \{I_1^o, \ldots, I_n^o\}$, a positive instance $p = \{I_p\}$ and a set of negative instances $N = \{I_1^n, \ldots, I_m^n\}$. The specific siamese network architecture for which the outfit ranking loss is defined, which will be further explored in sections 3.3 and 4.2.3, predicts subspace attention weights in a sub-network by concatenating one-hot vectors representing the semantic item category of the instances. This means that each of the negative instances as well as the positive sample will have multiple embedding representations by iterating over different anchor category vectors $c(I_i^o)$.

$$\mathbf{f}_i^o = \psi(\mathbf{I}_i^o, c(\mathbf{I}_i^o), c(\mathbf{I}^p)); i = 1 \longrightarrow n$$
$$\mathbf{f}_i^p = \psi(\mathbf{I}^p, c(\mathbf{I}_i^o), c(\mathbf{I}^p)); i = 1 \longrightarrow n$$
$$\mathbf{f}_i^{n_j} = \psi(\mathbf{I}_j^n, c(\mathbf{I}_i^o), c(\mathbf{I}_j^n)); j = 1 \rightarrow m, i = 1 \rightarrow n$$

$\psi(.)$ represents the specific siamese network architecture for which the embeddings $\mathbf{f}_i^o$, $\mathbf{f}_i^p$ and $\mathbf{f}_i^{n_j}$ are extracted. Further, we define the distance to the outfit O, i.e. the anchor samples, from each instance s remaining from the complete training sample (positive or negative) as

$$D_{outfit}(O, s) = \frac{1}{n} \sum_{i=1}^{n} d(\mathbf{f}_i^o, \mathbf{f}_i^s),$$

where $d(.)$ is the euclidean pairwise distance when comparing two specific instances. To achieve a single distance metric from the set of multiple negative instances, an aggregate function φ, which depending on the configuration draws either the minimum value or the average, is utilized.

$$D_N = \varphi(D_n 1, ..., D_n m)$$

Finally, the calculated distances are compared in the contrastive loss evaluation which similar to the triplet loss includes a distance margin μ.

$$l(O, p, N) = max(0, D_p - D_N + \mu)$$

Chapter 3

Related Work

This chapter describes related work within the concerned research area of fashion recommendation, with emphasis on fashion compatibility modelling. Recent deep learning and computer vision methods and models for the outfit matching task will be presented, with more detailed descriptions of models chosen to be reimplemented in this thesis.

3.1 Fashion Compatibility Modelling

Fashion Recommendation can be defined as a set systems that tries to predict and retrieve appealing selections of fashion items to one or more target consumers [21], and have gained significant interest within deep learning and computer vision in recent years [7, 23, 24, 25].

Traditional approaches to fashion recommendation relies on substitute recommendations, meaning the retrieval of visually similar items to one or more source items. Similarity-based fashion recommendation has several use cases, for instance matching street images of people to online fashion products [45, 46]. However, in many cases target consumers are more interested in fashion items of various types that can act as a complementary garment or accessory to the source items in question and adheres to a similar style, for instance a certain pair of jeans that goes well with a specific shirt. These cases are built on the notion of compatibility.

Modelling compatibility between fashion items have gained interest within deep learning and computer vision in recent years, particularly for the task of matching outfits[7, 23, 24, 25]. McAuley et al. [23] utilizes a pre-

trained CNN to estimate pairwise compatibility between two fashion items of different types in a shared feature space using distance metric learning, where the learned distance between two items corresponds to their compatibility. Veit et al. [24] improves on this using a Siamese Neural Network [47] to compute distances between pairs of fashion items, a network structure which involves two identical sub-networks conjoined at the output stage. The main drawback of this pairwise approach is that the method struggles to capture the relationship between larger collections of items, for instance complete outfits. While it would be feasible to employ a voting strategy and average out all pairwise combinations, this would incur a high computational cost and neglect certain coherent contextual information only found when combining more items.

To remedy this shortcoming and allow for estimating the compatibility of a complete outfit, Han et al. [7] considers clothing garments to be an ordered sequence of items. A bidirectional LSTM [48, 49] is used to handle sequences of feature vectors from fashion images and semantic textual information extracted by the InceptionV3 CNN [50]. This allows the model to handle inputs of various lengths but this sequential approach, which means a fixed ordering of garment from top to bottom followed by fashion accessories, leaves the method burdensome to use if one desired to tweak or extend the task, as in this thesis, or in cases where one or more categories of garments were unavailable. This work additionally proposed the fill-in-the-blank (FITB) evaluation metric, which has been a standard benchmark test for compatibility modelling since and will be utilized for quantitative model evaluation in this thesis.

Similar to [7], Vasileva et al. [25] also use semantic as well as visual embeddings in their method, utilizing the Conditional Similarity Network (CSN) presented in [43]. The core idea here is to countermeasure the shortcomings in earlier works that used a singular embedding space, by instead capturing compatibility utilizing a multitude of type-conditioned embedding subspaces. Adopting this network, they learned a total of 66 subspaces for each combination of pairs of item types. This procedure, which has alleviated some of the flaws in pairwise approaches by using different projections conditioned on type, was then further improved by Tan et al. [6].

3.2 Similarity Condition Embedding Network

A weakness in [25] was that the type-conditioned subspaces needed to be predefined, a weakness which is conquered by Tan et al. in [6] with the introduction of the Similarity Condition Embedding Network (SCE-Net), which allows the network to learn the subspace representations without any explicit supervision. Instead, the representations are learned as a latent variable, and a conditional weight branch is used to gauge the importance of each subspace. With this idea, it was found that better results could be achieved with much fewer subspaces.

The conditional weight branch, which enacts itself as a sub-network that predicts attention weights for weighing different subspaces at each iteration, is processed using combinations of inputs from the triplet input data which are concatenated and sent through a series of fully-connected layers. The predicted attention weights are then further utilized together with the image feature vectors and a learnable set of masks when calculating the final embedding as the weighted sum of all subspace embeddings. The network will be explained in more depth in section 4.2.2.

3.3 Category-based Subspace Attention Network

While the work by Tan et al. [6] achieves strong quantitative results on public benchmark datasets for fashion compatibility modelling, their method is impractical for individual retrieval tasks as it relies on input image pairs, where one image represents the target item. To overcome this, Lin et al. [5] introduces the Category-based Subspace Attention Network (CSA-Net), which functions similarly to the network proposed in [6], with the main variation the network now solely relies on the input of one image along with pairs of category vectors.

The input image is initially passed through a ResNet18 [2] CNN. Alongside it, the two category vectors are first processed using one-hot encoding and then concatenated and sent through a sub-network, similar to the one in the SCE-Net, where the outputted attention weights serves the same purpose of weighing the different subspace embeddings. The network will be explained in more depth in section 4.2.3.

Chapter 4

Methods

In this thesis, two state-of-the-art deep learning models and one baseline model for fashion compatibility modelling will be investigated and compared for the task of matching stylistic collections of fashion items. To settle on the specific models, a literature study has been conducted covering the research fields of fashion compatibility modelling and fashion recommendation within deep learning and computer vision. In addition to the reimplemented models, industrial fashion datasets of second-hand clothing garments and fashion accessories have been constructed. In this section, the datasets will be presented along with common public benchmark datasets within fashion compatibility modelling. The chosen models will as well be presented and described in detail. In addition to this, selected evaluation methods will be introduced and described.

4.1 Data

For this thesis, multiple configurations of an industrial fashion dataset of visual as well as semantic categorical data have been constructed from raw data of clothing garments and fashion accessories. The raw data has been provided by the host company and collaborator of this project Sellpy. Beyond this, feature extractors of the chosen models for reimplementation have been pre-trained on the ImageNet [51] dataset. In this section, the Sellpy dataset configurations, as well as the very common benchmark dataset for fashion compatibility Polyvore Outfits, will be presented.

4.1.1 Polyvore Outfits

The first widely used dataset for the outfit matching task and fashion compatibility modelling was presented by Han et al. [7]. The dataset, which was named Maryland Polyvore, consisted of outfits put together by users at the former community-powered social website Polyvore. Each outfit consisted of a varying number of clothing garments and fashion accessories, with no duplicate types of items, for a total of 21889 outfits and 164379 items. Vasileva et al. [25] later expanded on this dataset by introducing Polyvore Outfits, a bigger dataset of 68306 outfits and 251008 items crawled from the same website. A more restricted version known as Polyvore Outfits Disjoint (Polyvore Outfits-D) was also constructed to avoid one item from being present in multiple splits of the data.

Since its release, Polyvore Outfits and Polyvore Outfits-D have been by far the most commonly used datasets for fashion compatibility modelling and for the outfit matching task. Because of the dominance of Polyvore Outfits within the research field, it's unclear how state-of-the-art models would perform on industrial fashion datasets. Additionally, considering all outfits have been constructed by regular users of Polyvore and not certified fashion experts, the credibility of the dataset can be questioned. At the very least, an urgency to test state-of-the-art models on new datasets is warranted. In this thesis, models trained on Polyvore Outfits will be reimplemented and trained on novel industrial datasets constructed using visual as well as semantic categorical data from Sellpy. The models will additionally be retrained on Polyvore Outfits to ensure correct reimplementation.

Figure 4.1: Examples of outfits in Polyvore Outfits dataset.

4.1.2 Sellpy Data

Raw product data of both previously sold items, as well as items currently on sale in the online store was provided by Sellpy. Each item contained a set of images, specific textual and categorical metadata and additional product information, including the set of existing stylistic collections, which that unique item belonged to.

The core idea when transforming the data was to use these existing stylistic collections as a weak label under the assumption that items previously curated within the same collection shared a stylistic resemblance. This then allowed the reimplemented models to be trained in their intended weakly supervised setting. From the complete set of items, roughly 100 000 items were associated with one or more stylistic collection, a number which continuously increased during the project as more collections were curated. Only these items was considered to be used as training or test data.

Data Collection & Preprocessing

As an initial filtering step, a selected subset of the collections were chosen as not all adhered to a notable and distinct style. Initially, 52 collections were chosen from an original set of over 300. Among the selected collections, a number of further analysis steps were performed to filter out unwanted collections. Because the aim was to only consider fashion data in the project, all collections were filtered based on item category where only clothing, shoes and fashion accessories remained in the collections. At this point, collections that previously carried a large percentual amount of non-fashion items were excluded as it was concluded that the style of those collections were heavily reliant on non-fashion items.

Because items could belong to more than one stylistic collection, a correlation test was performed between all pairs of collections where the number of joint unique items between both collections in the pair was calculated. A selected number of collections was filtered out due to this, as their inclusion was deemed superfluous. Additionally, it was noted which pairs of collections that had a high correlation as this would later be taken into account when creating transformed data samples. Lastly before settling on the final selection of stylistic collections, a subset of collections were manually removed, mainly due to poor image quality of the included items. From the final selection, 25 collections remained with a collective total of 8702 items.

Because the models only considered image data, or image data together with processed categorical data of the item type, only this data was further processed. From the set of images for each item, only a single main image of that item was chosen, and further processed by first resizing it to fit the input dimensions to the ResNet18 CNN feature extractor of 224x224 pixels, and later normalized. For the categorical item type data, a one-hot encoding was performed which only considered item types for the selected 8702 items.

Before constructing transformed samples of different configurations, the data was split into an 80% training split, 10% validation split, and 10% test split. The split was stratified with respect to the stylistic collection, i.e. all splits contained the same percentual share from each collection. Additionally, two randomly selected collections were completely removed from the training and validation splits for the purpose of using previously non-seen labels in experimentation. From the training and validation splits, two transformed configurations were constructed based on the criteria of the loss functions of the respective state-of-the-art models. The held-out test split, later denoted as the evaluation data, was configured to specifically suit the nature of the custom evaluation tests.

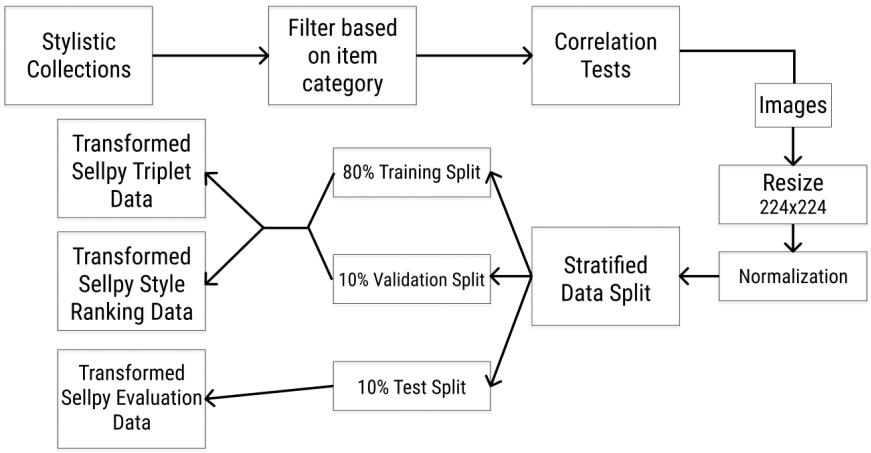


Figure 4.2: Overview of the major data preprocessing steps performed in the thesis.

Sellpy Triplet Data

The first constructed dataset, transformed from the preprocessed image data instances of the training and validation splits, was the triplet data, built to utilize in conjunction with the triplet loss function. Thus, the construction followed the principles of the triplet loss by structuring samples in triplets of instances, where each sample consisted of an anchor image a and a positive image p sampled from the same label, i.e. the same stylistic collection, and a negative image n sampled from a different collection. A few factors were of importance when sampling appropriate triplets. The two stylistic collections considered in each triplet formation had to not encompass a significant correlation, as the intention was to detect two relatively divergent styles. Further, emphasis was placed on not allowing any instance in a triplet to exist in the other collection contained in the triplet sample.

From the preprocessed data splits, 18170 triplets of transformed data were constructed. This entails that several unique image instances occurred in multiple triplets, although no two triplets, and as well no two anchor-positive pairs, were identical. For the experiments performed in this report, only visual image data was used for the triplet data.

Figure 4.3: Two examples of triplets used in the Sellpy Triplet data.

Sellpy Style Ranking Data

To adhere to the dynamics of the outfit ranking loss presented in [5], the construction of a second transformed dataset based on the preprocessed training and validation splits was necessary. Because the samples in this case were not drawn from constructed outfits but rather stylistic collections of fashion products, where duplicate item types were allowed, the choice was made to denote this as the style ranking data, rather than outfit ranking (not to be confused with the actual loss function, which will still be denoted as the outfit ranking loss). The outfit ranking loss function is defined such that it can be configured in various different settings, as the amount of anchor instances as well as negative instances remains undefined. After altering the settings in initial experiments, the choice was made to include three anchor instances and two negative instances in one complete training sample, together with the singular positive instance. This setting showed promising performance in initial testing, and as well alleviated the system from heavy computational requirements that larger training samples would incur.

Beyond the image data of the included item instances, one-hot encodings of the item categories were incorporated. This was based on all included categories in all three combined data splits. In total, 36 fine-grained item categories were encompassed in the data and as such, this was also the length of the resulting one-hot vector for each item.

When sampling instances for the construction of complete training samples, several factors were taken into account. Based on the dynamics of the outfit ranking loss function and the structure of the CSA-Net, all negative instances together with the positive instance had to be of the same item type, or category. All anchor instances together with the positive were sampled from the same stylistic collection, where as both negative instances were drawn from two distinctly different collections, both not displaying any significant correlation with the anchor-positive collection. Similar to the constraints of the triplet data, none of the negative instances were allowed to subside in a collection from which the anchors and the positive instance were sampled.

The total number of complete style ranking samples in the transformed dataset equaled that of the triplet data, namely 18170. Thus, similar to the triplet dataset, several unique item instances occurred in a multitude of style ranking samples. The same constraints on not allowing multiple identical samples

were still enforced.

Figure 4.4: Two examples of complete data samples from the Sellpy Style Ranking data.

Sellpy Evaluation Dataset

Because the custom evaluation tests presented in sections 4.3.1-4.3.3 required a certain structure of the data, an additional configuration needed to be constructed from the preprocessed test split to adhere to these requirements. This structure, which is denoted as the evaluation data, exists in two configurations for the networks trained on the triplet data and style ranking data respectively, with the only difference being that the style ranking configuration includes the one-hot encoded item type vectors.

The core approach was to take the test split and reconstruct the data to fit the prerequisite of the evaluations. More specifically, complete samples with sizes ranging from four to six item instances were constructed, where each instance belonged to the same stylistic collection.

Figure 4.5: Example of Sellpy Evaluation data sample.

4.2 Models

For the main experiments, two state-of-the-art deep learning models, namely the Similarity Condition Embedding Network (SCE-Net) and the Category-based Subspace Attention Network (CSA-Net), were reimplemented together with the construction of a simpler baseline model. For fairness in comparison, all models used the same backbone CNN feature extractor, namely the ResNet18 pre-trained on ImageNet data.

4.2.1 Baseline Model

To achieve a useful starting point for comparison, and to be able to gauge the state-of-the-art models against a benchmark, a simple baseline model was constructed. In most situations, the customary approach is to utilize a baseline at the company or institution where the project is being carried out. However, as the host company does not currently have a network structure for fashion compatibility modelling able to provide complementary recommendations built on the notion of compatibility, this option was not available.

The fundamental approach for the baseline was to utilize the same backbone CNN architecture, namely the ResNet18 model, as the two state-of-the-art models. Although instead of extending the model beyond that backbone architecture with a more complex structure, the baseline simply used the features extracted from the ResNet18 as the output embedding space with which the distance metric in the ranking loss was calculated on. The baseline was trained using triplet loss and as such, the triplet dataset was utilized.

Standard hyperparameters of the network included a batch size of 64, an initial learning rate of $5e^{-5}$, embedding representations of 64 dimensions, and a loss

margin μ of 0.2. If not explicitly stated otherwise, these hyperparameters were used in experimentation.

4.2.2 SCE-Net

As previously cited in section 3.2, the SCE-Net was presented by Tan et al. [6] as an improvement and extension on the work done by Vasileva et al. [25] on fashion compatibility modelling utilizing the idea of multiple subspace embeddings in network structures inspired by the Conditional Similarity Network (CSN) presented in [43]. The major improvement in the SCE-Net was that the subspaces could now be learned without explicit supervision, and instead a sub-network gauged the importance of each subspace by weighing scalar subspace attention weights.

Similar to other network structures based on the CSN, the SCE-Net function on pairwise inputs of visual image data. As the network is optimized using triplet loss, two pairwise comparisons is of interest for the complete loss calculation, namely the anchor-positive pair and the anchor-negative pair. The network is trained end-to-end as a singular model, where the input pair is first passed into a ResNet18 backbone CNN to extract features of a given embedding dimension D, denoted V_1 and V_2 for both respective instances in the pair. This is the general embedding space at which the baseline model described in section 4.2.1 is evaluated. At this point, to enable the idea of modelling different aspects of similarity across multiple subspaces, a set of M number of learnable masks denoted as $C_1, \ldots, C_M$ are introduced, where each mask represents a given subspace. Each mask C_i carries the same dimension as the general embeddings V_1 and V_2, and subspace embeddings are generated by applying the element-wise (hadamard) product for both general embeddings on each mask.

$$E_{ij} = V_i \odot C_j;\ i = 1 \rightarrow 2, j = 1 \rightarrow M.$$

We can think of the output of this subspace masking operation across all learnable masks for a given general image embedding V_i as a matrix O of dimension MxD.

$$O = [E_{i1}, \ldots, E_{iM}]$$

To measure the relevance of each given subspace for a certain pair of image input data, i.e. for that specific comparison, both general embedding outputs V_1 and V_2 are concatenated and sent to a sub-network denoted as the condition weight branch, which produces a subspace attention weight vector of dimension M, thus containing one scalar weight per subspace signifying the respective importance. The weight branch is a simple feed-forward neural network consisting of two fully connected layers of dimension D and M respectively, with a ReLU activation function succeeding the initial layer and a final softmax layer that serves the purpose of normalizing the weighted output. This output weight vector w is then multiplied with the already calculated masked subspace embedding matrix to achieve the desired final aggregated output embedding representation.

$$E_i = wO^T$$

As previously mentioned, the SCE-Net is trained using the triplet loss function, described in detail in section 2.3.2. The loss function operates by drawing the anchor-positive pairs closer while simultaneously pushing away the anchor from the negative instance. Beyond the triplet loss, we further regularize the network in two distinct ways. First, we impose an l1 norm on the learnable masks to promote sparsity in them and additionally we extract the l2 norm from the extracted features in the general embedding space.

$$L1(C) = ||C||_1$$
$$L2(Vi) = ||Vi||_2^2$$

The final loss imposed in the training process is the combination of the triplet loss and the two regularizations, which both are constrained with their respective hyperparameters λ_1 and λ_2.

$$L_{total} = L_{triplet} + \lambda_1 L1 + \lambda_2 L2$$

Standard hyperparameters of the network, as stated in [6], includes a batch size of 256, an initial learning rate of $5e^{-5}$, the number of subspaces set to 5, embedding representations of 64 dimensions, and a loss margin μ of 0.2. If not explicitly stated otherwise, these hyperparameters were used in experimentation.

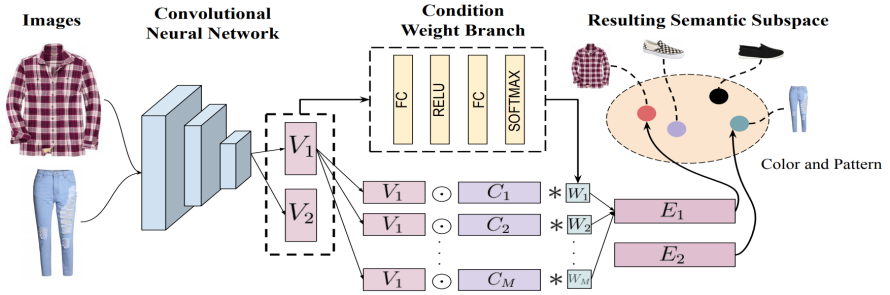


Figure 4.6: Model architecture of SCE-Net, as presented in [6].

4.2.3 CSA-Net

As cited previously, the SCE-Net reached strong results on the common fashion compatibility benchmark tests, and the idea of learning masked subspace attentions without explicit supervision showed itself to be a powerful concept. However, the main noted drawback of the architecture was that it lacked functionality as well as flexibility for certain tasks. This concern was addressed in the work by Lin et al. [5] and their presentation of the CSA-Net. They expressed the lack of practical functionality of the SCE-Net in any retrieval setting, as the network relied on pairwise inputs in the condition weight branch, which hindered it from retrieving complementary items based on a singular input in its default configuration. Instead of relying on product pairs, the CSA-Net improved the flexibility in a retrieval setting by allowing a singular product as input along with a pair of one-hot encoded category vectors, one representing the category of the input and the other a target category, i.e. the category of the item to be retrieved.

Apart from this main variation of the two networks, which fundamentally changed the dynamics with the CSA-Net not relying on input product pairs, they are architecturally very similar. The features of the visual input image is first extracted through a backbone CNN architecture, namely the ResNet18 CNN, pre-trained on ImageNet data. The network then relies on the same concept of multiple subspace embeddings and conditional, learnable masks which are weighted by attention weights outputted by a sub-network which is a simple feed-forward neural network of two fully-connected layers and a final softmax layer that normalizes the attention weight vector. The main difference, as previously mentioned, is the input of the sub-network (or condition weight branch) which now takes in the concatenation of the category vector pair of

the input as well as the target category.

While the architecture of the CSA-Net bears a strong resemblance to that of the SCE-Net, the training process is quite distant as they utilize different loss functions. The CSA-Net introduced a novel loss denoted as the outfit ranking loss, described in detail in section 2.3.3. As such, multiple embeddings were calculated for different instances in a complete training sample, iterating the various categories represented.

Standard hyperparameters of the network, as stated in [5], includes a batch size of 96, an initial learning rate of $5e^{-5}$, the number of subspaces set to 5, embedding representations of 64 dimensions, and a loss margin μ of 0.3. If not explicitly stated otherwise, these hyperparameters were used in experimentation.

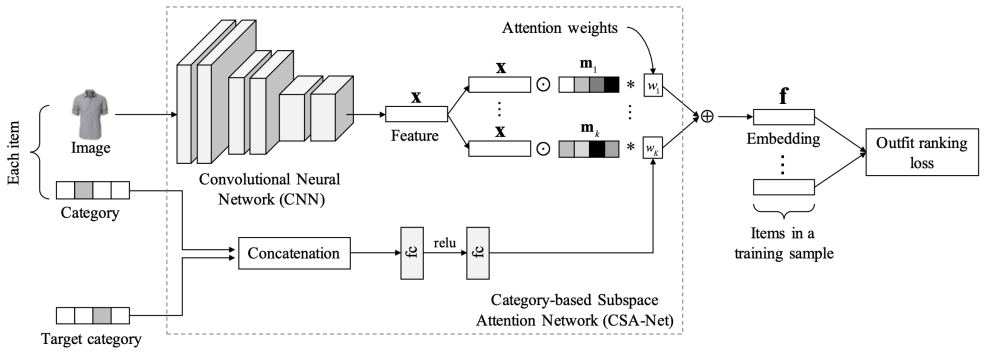


Figure 4.7: Model architecture of CSA-Net, as presented in [5]

4.3 Evaluation

The experimental study of this thesis will consist of two main quantitative evaluations, Fill-in-the-blank (FITB) and Style Retrieval. The first is a standard method of evaluating the closely related outfit matching task, and the second is an altered version of the Outfit Complementary Item Retrieval evaluation task presented in [5]. All models implemented for the thesis, including the baseline model, will be evaluated on these tasks. Results will be conducted in an ablation study consisting of the models and their results on the respective tasks, as well as their results for Fill-in-the-blank on the Polyvore Outfits dataset. Through this ablation study, it will be evident which

model generalizes best to the novel task of matching stylistic collections of fashion items, and as well it will grant an understanding of the relation between this task and the outfit matching task by analyzing the discrepancy in results between the two datasets.

4.3.1 Fill-in-the-blank (FITB)

First presented by Han et al. [7], Fill-in-the-blank (FITB) has become a standard evaluation task for fashion compatibility modelling and for the outfit matching task [25, 5, 52, 6]. The core idea is to remove a random item from each collection of compatible fashion items (positive sample) in the test data, also denoted as the question set, and the objective of the model is to pick the correct missing item from four options (answer set), the other three being randomly sampled items from a non-compatible collection (negative sample). This task is then quantitatively evaluated by simply drawing the accuracy from all FITB iterations of the experiment, that is the percentage of correct picks.

Two primary variants of the FITB evaluation have been utilized in earlier research within the field. One of them uses mixed item categories in the answer set, and the other instead use a single category, which means that all non-compatible samples in that case have to be sampled from the same item category as the removed compatible item. For this project, both variants have been used.

Figure 4.8: Example of question set and answer set in one iteration of the FITB experiment presented in [7].

4.3.2 Style Retrieval

The task of style retrieval is inspired and hence is a slightly altered version of the Outfit Complementary Item Retrieval task presented in [5]. The core idea is to take a collection of compatible items (a positive sample) and perform recurrent retrieval from a large set of items both containing items from the same special store (i.e. compatible items with the collection in question) but also non-compatible items. By performing recurrent retrievals, and aggregating the retrieved results for all items in the compatible collection by average fusion, the final result will be a rank of retrieved items based on what the model considers to be the most to least compatible with the initial collection.

Within this rank, it will then be evaluated how many of the top-ranking items that are compatible according to the ground truth, which as mentioned is the stylistic collection that the item belongs to. As such, performance in the evaluation test will be measured using $recall\ @\ top\ k$, with distinct recall values for a number of different k-values. The data being retrieved, which has been indexed ahead of the experiment using the output embedding representation of the item, is queried for utilizing kNN-search, for any given k.

4.3.3 t-SNE Visualization

In addition to the above-mentioned quantitative evaluation methods, the models will as well be evaluated in a qualitative fashion. A method of qualitative evaluation, which has been used in the research field [53], is to visualize the extracted item embeddings using the t-SNE algorithm [54]. This will enable a relatively large collection of item embeddings to be visualized in a two-dimensional space, where the aim and idea is that more compatible items will be placed closer to each other in the space. If a stylistic resemblance can be detected by fashion experts in garments and fashion accessories closely located, it indicates that the model has been successful in modelling style compatibility.

4.4 Experiments

To estimate the relation between the outfit matching task and the adapted novel task of stylistic compatibility modelling, and as well to gauge an

understanding of how well the reimplemented models perform in this setting of matching stylistic collection rather than outfits, a main FITB experiment will be performed. In it, the reimplemented state-of-the-art models will be tested on newly assembled Sellpy data using the same parameter settings in the training and evaluation process as the original implementation of the models, for a fair comparison. The models will as well be evaluated on Polyvore Outfits data in an outfit matching setting, to assure that the reimplementation is configured correctly. The results of these experiments will be presented in section 5.1.1.

For the purpose of further exploring the capability of the models and further optimize the parameter settings, a series of ablation studies will be performed where selected hyperparameters will be varied and evaluated using FITB to find an optimal tuning. These hyperparameters include the dimension of the embedding representation of the image data, the number of subspaces used in the models, and the margin value of the respective ranking loss functions. The concluding results of these studies will be presented in section 5.1.2.

The Style Retrieval evaluation will consist of a series of experiments partly inspired by the experimental retrieval performed in [5] but adapted to suit the retrieval of stylistic collections rather than matching outfits. A ranked selection of compatible garments will be retrieved based on both a single sample, but also a smaller collection of samples as a query. Further, category-specific retrieval will be performed where only a given category of indexed samples will be considered for retrieval. This aims to simulate the setting where a specific target category is desired for a given stylistic collection. Results from these experiments will be presented in section 5.2.

For the evaluation utilizing t-SNE visualizations, selected masked subspace embeddings of the respective state-of-the-art models on a curated subset of items will be visualized. The visuals will then be qualitatively evaluated to gauge if they have learned different notions of similarity among the selected items during training. Visualizations will both be performed using a diverse selection of categories as well as in the case of a singular category. The resulting visualizations will be presented in section 5.3.

4.5 Software Implementation Details

All software for the thesis was written in Python 3.8.2, using the TensorFlow 2.4 library and the Keras interface for modelling. Models were trained as batch computing jobs at Amazon Web Services using a Tesla K80 GPU with 16GB of RAM. To run the t-SNE algorithm, the modular Python implementation openTSNE [55] was utilized. For indexing and retrieval of output representation embeddings, the nearest neighbor search library Annoy was used [56].

Chapter 5

Results

In this chapter, results from experiments described in section 4.4 using evaluation methods detailed in section 4.3, will be reported and analyzed. First, a main comparison to the outfit matching task will be made by contrasting the FITB results of the reimplemented models on Sellpy data with respective FITB results in their initial implementation on Polyvore Outfits data. Second, a series of ablation studies with the aim of optimizing specific hyperparameters of the SCE and CSA model respectively will be conducted, in an FITB setting. Subsequently, the Style Retrieval experiments detailed in section 4.3.2 will be performed, both in a general retrieval setting as well as a category-specific one. Lastly, a set of t-SNE visualizations will be displayed for various subsets of the evaluation data.

5.1 FITB

5.1.1 Performance Comparison to Outfit Matching Task

Table 5.1 displays the results of a main comparative FITB experiment, with the main purpose of producing a performance comparison to the outfit matching task. Standard hyperparameters of all models, stated in sections 4.2.1, 4.2.2, and 4.2.3 respectively, were used in the experiment. Both state-of-the-art models outperformed the baseline model for both the outfit matching task using Polyvore Outfits data, as well as the two style compatibility tasks using Sellpy Evaluation data. In all three tests, CSA-Net slightly outperformed the SCE-Net.

The more unconstrained task of style compatibility matching proved significantly

more difficult for all models, as performance on Polyvore Outfits data was higher. Between the two style compatibility tasks, Sellpy Evaluation Mixed having mixed item categories in the answer set of all FITB iterations, and Sellpy Evaluation One having a singular category, no significant difference in performance was noted.

Model $Name$	**Polyvore Outfits** $\%\ accuracy$	**Sellpy Evaluation Mixed** $\%\ accuracy$	**Sellpy Evaluation One** $\%\ accuracy$
Baseline	36.72	31.02	30.88
SCE	61.60	46.44	46.21
CSA	63.73	48.91	48.97

Table 5.1: FITB accuracy for all models when comparing the outfit matching task on Polyvore Outfits data to the style compatibility task using Sellpy data.

To investigate whether the SCE-Net and CSA-Net could generalize well to previously unseen classes, or in this case styles, an additional FITB evaluation was performed solely using question sets from two stylistic collections not included in the training or validation splits of the data. Results can be seen in Table 5.2, which suggests that the models are able to generalize to new styles given that the decrease in FITB accuracy is very slim.

Model $Name$	**Sellpy Evaluation Unseen** $\%\ accuracy$
SCE	46.02
CSA	47.94

Table 5.2: FITB accuracy for SCE-Net and CSA-Net on Sellpy Evaluation data with unseen stylistic collections making up all question sets.

Examples of FITB iterations can be seen in Figure 5.1 and 5.2 respectively. Figure 5.1 highlights two successful FITB iterations using answer sets of mixed categories, while Figure 5.2 show examples of unsuccessful iterations.

Figure 5.1: Two examples of successful FITB iterations, i.e. when the correct answer was chosen. Above is an example using SCE-Net, and below CSA-Net.

Figure 5.2: Two examples of unsuccessful FITB iterations, i.e. when an incorrect answer was chosen. Above is an example using SCE-Net, and below CSA-Net.

5.1.2 Ablation Studies

Three separate ablation studies were performed for the SCE-Net and CSA-Net respectively, each varying a specific hyperparameter to further explore and understand the dynamics of the models. For all experiments, standard hyperparameters of the respective models were used apart from the varying parameter. Table 5.3 shows results of varying the subspace dimension of the respective models, that is the number of learnable masks and thus the different notions of similarity that the models incorporated. There was not a significant difference in performance when varying the subspace dimension, and while 10 subspaces for the CSA-Net reached the highest FITB accuracy, results

showed that more subspaces did not necessarily equate to higher performance. Between the two models, CSA-Net slightly outperformed SCE-Net.

Subspace Dimension $\#$	**SCE** $\%\ accuracy$	**CSA** $\%\ accuracy$
3	46.12	48.62
4	45.55	48.95
5	46.44	48.91
7	45.07	46.76
10	44.87	49.93

Table 5.3: FITB accuracy results for SCE-Net and CSA-Net in an ablation study varying the number of subspace dimensions in the respective models.

In a subsequent study, the dimension of the embedding representation within each model was varied. Table 5.4 displays the results, which show an evident increase in performance for embeddings of higher dimensionality, with both state-of-the-art models reaching their respective highest accuracy for embeddings of 256 dimensions. A before, CSA-Net slightly outperformed SCE-Net across all tests.

Embedding Dimension $\#$	**SCE** $\%\ accuracy$	**CSA** $\%\ accuracy$
32	42.02	43.14
64	46.44	48.91
128	48.17	49.48
256	50.12	51.77

Table 5.4: FITB accuracy results for SCE-Net and CSA-Net in an ablation study varying the dimension of the output embedding in the respective models.

Table 5.5 shows the results of the final ablation study, where the loss margins μ of the respective ranking loss functions were varied. Once again, CSA-Net marginally outperformed SCE-Net when varying the margin parameter. The most significant difference between the two models was reached when using a small margin of 0.1, where SCE-Net comparatively attained its lowest accuracy, and CSA-Net reached its highest. Nevertheless, it could not be concluded that CSA-Net always benefitted from lower margins, as margins

of 0.3 and 0.4 respectively attained higher performance than 0.2.

Loss Margin	**SCE**	**CSA**
$\#$	$\%\ accuracy$	$\%\ accuracy$
0.1	44.96	51.10
0.2	46.44	46.98
0.3	47.34	48.91
0.5	45.91	47.77

Table 5.5: FITB accuracy results for SCE-Net and CSA-Net in an ablation study varying the margin of the loss function in the respective models.

As a final quantitative FITB experiment, the difference in the average distance between the chosen answer in each FITB iterations to the whole question set, and between the second closest item in the answer set to the whole question set, was calculated. Standard hyperparameters for both models, as described in sections 4.2.2 and 4.2.3 respectively, were used. The average difference in distance was, as can be seen in Table 5.6, divided into two groups based on if the answer had been correct or not. It could be concluded that for both SCE-Net and CSA-Net, the distance was slightly higher for correct answers, which indicates that the respective models were somewhat more confident of their decisions when the correct answer was selected.

Model	**Correct Answers**	**Incorrect Answers**
$Name$	avg	avg
SCE	0.008 71	0.007 76
CSA	0.007 87	0.006 69

Table 5.6: Average difference between the chosen answer and the second closes item in an FITB setting for the SCE-Net and CSA-Net.

5.2 Style Retrieval

As described in section 4.4, two primary experiments were performed for the Style Retrieval evaluation. For both experiments, the $recall\ @\ top\ k$ for k-values of 10, 30, and 50 respectively were tested for both SCE-Net and CSA-Net. Results from the first experiment, which performed retrieval from a

mixed set of all included item categories, can be seen in Table 5.7. For both models, the percentual share of what was considered a relevant item increased for higher k-values. Similar to the FITB experiments, CSA-Net somewhat outperformed SCE-Net. Examples of retrieval results from this experiment can be seen in Figure 5.3.

Recall @ top k	**SCE**	**CSA**
k	%	%
10	26.65	29.30
30	35.01	36.77
50	42.49	45.25

Table 5.7: Recall @ top k for k-values of 10, 30 and 50 respectively for SCE-Net and CSA-Net.

Figure 5.3: Example results of performing automatic style retrieval based on a query set of compatible items.

Results from the second quantitative Style Retrieval experiments, where retrieval was performed from a specific selection that only included a singular item category, are displayed in Table 5.8. The final result was averaged for all included categories. Similar to the results in Table 5.7, the proportion of relevant items increased with a larger k-value, and similar to before, CSA-Net outperformed SCE-Net for all values of k. Examples of retrieval results from this experiment can be seen in Figure 5.4.

Recall @ top k	**SCE**	**CSA**
k	%	%
10	34.83	39.76
30	43.39	48.42
50	49.03	54.30

Table 5.8: Recall @ top k for k-values of 10, 30 and 50 respectively for SCE-Net and CSA-Net in a category-specific retrieval setting.

Figure 5.4: Example results of performing category-specific automatic style retrieval based on a query set of compatible items.

To investigate how the respective models would perform when retrieving based on a singular item query, a final qualitative Style Retrieval experiment was performed in this setting. Examples from this experiment can be seen in Figure 5.5.

Figure 5.5: Example results of performing automatic style retrieval based on a query set of just a single item.

5.3 t-SNE

As described in section 4.3.3, a qualitative evaluation of the models was performed using the t-SNE algorithm. In Figure 5.6 and 5.7, individual subspaces have been visualized in a low-dimensional space for both respective state-of-the-art models given a selected set of items. In Figure 5.8 and 5.9, results can be seen given another set of items, this time all sampled from the same item category.

Figure 5.6: Visualization utilizing the t-SNE algorithm of 150 randomly selected items of the Sellpy Evaluation data, with mixed categories for SCE-Net.

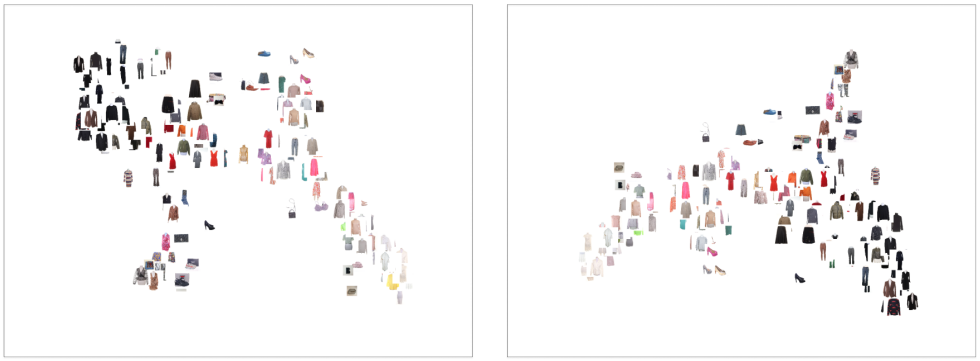


Figure 5.7: Visualization utilizing the t-SNE algorithm of 150 randomly selected items of the Sellpy Evaluation data, with mixed categories for CSA-Net.

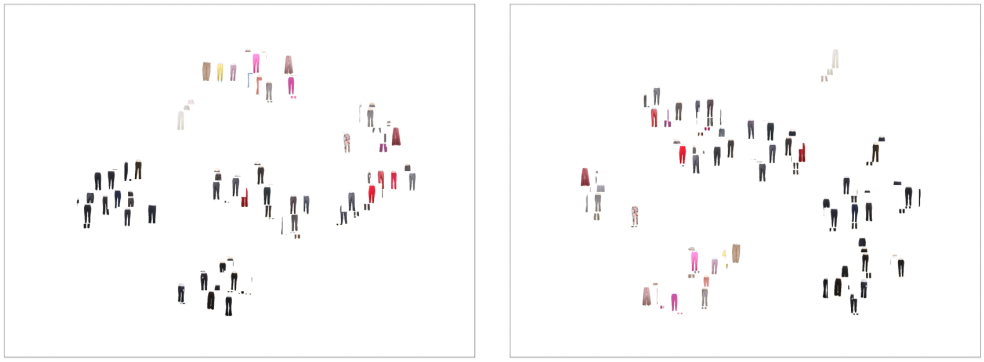


Figure 5.8: Visualization utilizing the t-SNE algorithm of 150 randomly selected items of the Sellpy Evaluation data, with a single category for SCE-Net.

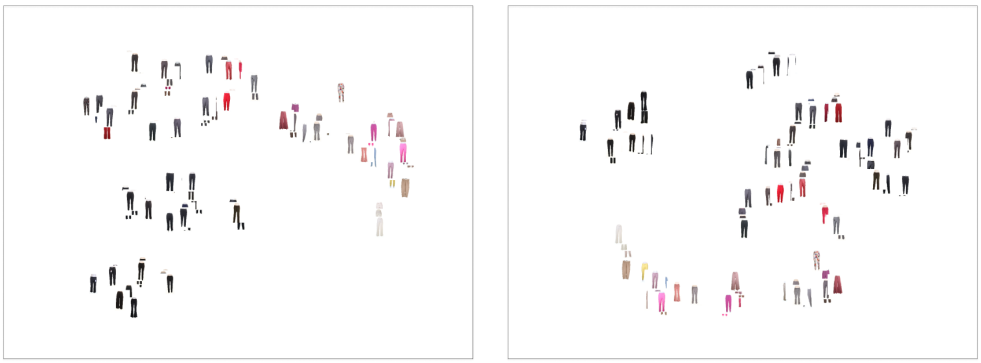


Figure 5.9: Visualization utilizing the t-SNE algorithm of 150 randomly selected items of the Sellpy Evaluation data, with a single category for CSA-Net.

Chapter 6

Discussion

This chapter will be a discussion of the results presented in Chapter 5, as well as a critical evaluation of the methodology presented in Chapter 4. Section 6.1 gives an analytic summary of the results and findings, quantitative as well as qualitative, and in sections 6.2 and 6.3 respectively, more detailed analysis of specific aspects of the findings in relation to the research questions presented in section 1.2.2, will be presented. In section 6.4, limitations within the chosen methodology will be analyzed and discussed, and the final section of the chapter will give a reflection of the ethical and sustainable aspects of the thesis work.

6.1 Summary of Results & Findings

Given the configuration of the FITB evaluation, one could assume that a completely random model would achieve an accuracy of 25%, since at each iteration the answer set contains four separate options. Based on this, we can conclude that the baseline model performed just slightly above random in the main comparative FITB experiment presented in section 5.1.1. This result was largely expected, as the model was designed to be simplistic with the anticipation of it not being complex enough. That being said, it would be interesting to train additional layers of the ResNet18 CNN from the point of their ImageNet pre-training, rather than as in this case just tweak the parameters of the final layer, and investigate if that potentially could yield higher performance.

The baseline model was clearly outperformed by the two reimplemented state-of-the-art models, that yielded results significantly above random which

suggests a better understanding of the problem being presented, and may indicate that the models to some extent have learned to recognize stylistic resemblance between fashion items. This indication could also be reinforced by the quantitative results attained in the Style Retrieval evaluation presented in section 5.2, which showed the models clear preferences towards retrieving relevant items at top rankings, meaning items from the same stylistic collection as the query. Given that 27 separate collections, with an equal amount of items, made up the set of possible items to retrieve, one would expect a random model to attain on average a $recall\ @\ top\ k$ of 3.7%, for any value of k. SCE-Net and CSA-Net achieved an average score of 26.65% and 29.30% respectively for top 10 retrieval rankings, a number which also grew for larger k-values. This demonstrates a strong preference to retrieve items from the same stylistic collection at the highest ranking positions, for both SCE-Net and CSA-Net.

Furthermore, both SCE-Net and CSA-Net showed ability in generalizing well to previously unseen styles, given the results in Table 5.2. When exclusively using two stylistic collections not present in the training or validation splits for the data for all FITB iterations of the experiment, accuracy only dropped 0.42 and 0.97 percentage points for SCE-Net and CSA-Net respectively, a decrease modest enough to not be considered especially substantial in terms of the models capabilities. This is an interesting finding, as it indicates that even though a diversity of styles is desirable within the data, a lack of diversity doesn't necessarily mean that practical results will suffer from that.

From the ablation studies presented in section 5.1.2, it was evident that by increasing the dimension of the embedding representation, a higher accuracy could be achieved. This was expected, as a larger embedding vector is able to store more information. Further, by varying the subspace dimension of the models in subsequent tests, it was difficult to analyze the results presented in Table 5.3. Similar to the embedding dimension, a reasonable prediction ahead of testing may be that more subspaces lead to higher performance, as additional information can be gathered, but this was not necessarily the case given the results. While the CSA-Net reached the highest accuracy using the most amount of subspaces, more specifically 10, it also attained a lower score when utilizing 7 subspaces compared to 5, 4, and 3. For SCE-Net, the highest accuracy was achieved using the standard configuration of 5 subspaces, and the second highest score when utilizing only 3. In [6], a similar ablation study was performed for the SCE-Net which as well indicated a preference for fewer subspaces in the model and concluded a subspace dimension of 5 to be optimal

for SCE-Net. As such, it may not have been surprising that this configuration yielded the best result. However, the lack of a clear pattern in the results was what made it difficult to draw conclusions.

Visualizing the separate subspaces using the t-SNE algorithm, with results presented in section 5.3, meant to shed light on how different subspaces within the model reason, and how the different subspace condition are learned throughout training under the influence of the condition weight branch. As the evaluation was purely qualitative, definite conclusions was difficult to draw. However, the idea of individual subspaces learning different notions of similarity, which together constitutes a measure of compatibility, may very well hold true given these results. By inspecting the visualizations, it's clear that similar items in terms of color and item category appeared close in the two-dimensional space. When comparing two separate subspaces for the same model, similar clusters of items can be found, albeit in different positions in the graph. This could indicate that while different subspaces learn separate notions of similarity, an overlap between them does exist.

6.2 Style Compatibility

One main aim of the thesis was to investigate if, and how well, models designed for the outfit matching task within fashion compatibility modelling could be adapted to a more unconstrained task of matching stylistic resemblance and thus perform style compatibility modelling on fashion data. From the main comparative results presented in section 5.1.1, it's evident that the models did perform significantly better on the outfit matching task using the Polyvore Outfits data. The question on why that is may have two possible answers, as the models were not only introduced to a new and more unconstrained task but also to completely new data. As per the task, it's reasonable to presume that matching fashion styles is comparatively more difficult than matching outfits, seeing as more combinations of matched compositions of items are possible, due to duplicate item categories. Additionally, the style compatibility task presented in this thesis does not put any gender constraint on the stylistic collections. As such a collection can contain both menswear and womenswear, as well as unisex items, which stands in contrast to Polyvore Outfits where each outfit is categorized by gender. It should be noted however that a majority of the collections in the Sellpy data do carry womenswear predominantly, but this does not stand true for all stylistic collections, and as such it may have affected the increased difficulty of the style compatibility task.

In terms of the data, there are some key differences in the fashion items themselves between the Sellpy data and the Polyvore Outfits data. At Sellpy, almost all fashion items are sold to the company by individuals. As such, the accumulated data contains a large portion of everyday clothing and basic garments. Even in the curated stylistic collections, which one could argue contains fewer everyday items due to the more selective procedure in curation, this still holds true to a large extent. Certain stylistic collections used in the Sellpy data for this thesis even has an explicit focus on basic garments. This is quite opposed to the Polyvore Outfits data, where most outfits predominantly accommodates eye-catching garments and statement pieces, which could reasonably be described as extravagant rather than basic. A presumption could be made that the typically more eye-catching nature of Polyvore Outfits gave the outfits within that dataset a more distinct, and visually more definite style. It remains unclear whether or not this difference affected the models, and as such made it more difficult to learn from Sellpy data, but it's not unreasonable to suggest that the more subtle styles carried by the everyday clothing at Sellpy had an effect.

6.3 Model Comparison

When analyzing the performance of SCE-Net and CSA-Net, as per the results presented in chapter 5, it's evident that CSA-Net achieves slightly better results across all experiments. While some presumptions can be made, it's not apparent as to why that is the case. In practice, there are two aspects that separate the models. The first one is the input to the conditional weight branch of the respective models, where concatenations of image embedding pairs acted as input for SCE-Net, while CSA-Net utilized one-hot encodings of the item category vectors. The other differentiating factor is the loss function, where SCE-Net is trained using the triplet loss and CSA-Net uses the outfit ranking loss. In [5], an experiment was made using a hybrid of these models, utilizing triplet loss with one-hot encoded category vectors. This configuration was inferior compared to the regular CSA-Net, which makes it reasonable to assume that it's in fact the outfit ranking loss function that produces a somewhat more accurate model.

Lin et al. [5] argued for the outfit ranking loss being able to consider item relationships within a complete set of items rather than just item pairs, which in turn would yield a higher capability in modelling compatibility. Given the

results presented in this thesis, this seems to hold true. It should be noted however that the outfit ranking loss function is more complex than the triplet loss, and as such it's as well more computationally intensive. It could therefore be argued that the slight increase in performance does not completely justify the upsurge in computational complexity.

6.4 Limitations in Methodology

6.4.1 Data

As mentioned before, the Sellpy data constructed for this thesis was based on previously curated stylistic collections. While these collections explicitly revolves around an intended style, it's important to have a critical approach when assessing the validity of the stylistic closeness between items in a given collection, especially when considering all items within that collection. When transforming the data to create the triplet, style ranking, and evaluation samples used for model training and testing, random items from the same collection were paired up and put together to form samples under the assumption that they shared a stylistic resemblance. On a whole, each collection may have expressed a fairly distinct style, but it's unclear how valid that still is when investigating every possible combination of item pairs or smaller subsets of the collection. With more resources, a calculated and selective manual process of pairing transformed data samples would have been preferred, however as this was not possible, it's plausible that certain samples suffered from a faulty pairing.

In a broader sense, similar critical questions can be asked of the Polyvore Outfits dataset and the related work within fashion compatibility. How can the composition of a fashion outfit or the curation of a stylistic fashion collection be judged objectively? To some extent, what is given to the model as ground-truth does carry some level of subjectivity and ambiguity. A set of items that one person considers to be compatible, may not be thought of as such by someone else.

6.4.2 Evaluation

Within fashion compatibility modelling, certain experimental evaluation tests have become standardized methods of measuring the success of a given models ability to learn compatibility between fashion items. The FITB

evaluation, which was used in this thesis, is one of them. Because these models, in the end, are meant to exist in some type of recommendation system, for example in an item retrieval setting, it's important to reason about how well the scores on these standardized tests, for example the FITB accuracy, translates to the practical setting of providing complementary recommendations through relevant item retrieval. CSA-Net outperformed SCE-Net on both the FITB evaluation and for recall values at top rankings in the custom retrieval experiment with results presented in section 5.2, but does this improvement really manifest itself in the practical production of better recommendations? It's clear that CSA-Net performed marginally better in retrieving relevant items by the definition of the evaluation test, but as stated in previous work [5], recommendations not considered relevant per the ground-truth may still be complementary and share a stylistic resemblance with the retrieval query.

By analyzing the qualitative results of the Style Retrieval experiments, as can be seen in Figure 5.3, 5.4, 5.5, and as well the visualized subspaces in Figure 5.6, 5.7, 5.8, and 5.9, it's difficult to say that CSA-Net provided better and more complementary recommendations compared to SCE-Net. Seemingly, they both do an adequate job of finding compatible items and it may be very hard to judge how the quantitative difference in performance translates to this practical setting. If one would conclude that the actual quality of the item retrieval was quite high, then that could bring an increasing doubt over the legitimacy of using FITB as a standard benchmark evaluation to measure compatibility modelling and ability to perform complementary recommendations, as both state-of-the-art models did not attain particularly high accuracies in those tests. While it remains unclear, it is possible that there exists a mismatch between the attained accuracy in the FITB setting, and the practical quality of the actual complementary recommendations in a retrieval setting.

6.5 Ethics & Sustainability

When developing models that use large amount of data, ethical and socially sustainable considerations are important. Regardless of configuration, the model will inherently be biased towards the data it has been trained on, and in most cases, multiple biases exist within that data. Most fashion datasets, including Polyvore Outfits and the Sellpy data constructed for this thesis, are dominated by womenswear. Additionally, selected sizes are significantly more common than others for both clothing garments and shoes. If we are not

considerate of these biases and take them into account when producing AI models, we run a risk of developing models that acts on a non-inclusive, and possibly even discriminatory basis.

Aspects of environmental sustainability are as well very important to recognize and uphold. Machine learning models, and especially large end-to-end deep learning architectures require increasingly substantial demands for computing power and energy usage, an increase that is currently outpacing energy efficiency improvements in new hardware [57]. The growing carbon footprint of deep learning, as a result of this development, is troubling. As such, considerations of energy usage and its impact on the environment are therefore critical when training deep architectures. In relation to this project, sustainable aspects could be considered when deciding if the supposed increase in performance of the outfit ranking loss function is justified given its added computational complexity in comparison to the triplet loss.

The application of the deep learning model in production should also be considered from ethical and sustainable aspects. As an online consignment store of second-hand fashion, Sellpy does provide a climate positive solution, which saves greenhouse gas emissions and water for every sold item compared to buying an equivalent new item. Because of this, applications based on the work in this thesis could provide a positive contribution from an environmentally sustainable aspect. However, one should be mindful that research within fashion recommendation and fashion compatibility modelling could be used to increase sales within traditional fast fashion, which is harmful from an environmental perspective.

Chapter 7

Conclusion

In this thesis, two state-of-the-art deep neural network models, SCE-Net and CSA-Net, originally used for the outfit matching task within fashion compatibility, have been implemented, trained, and evaluated on a newly constructed industrial fashion dataset. With this data, the aim was to investigate if the models could be successfully adapted to an altered and more unconstrained task of matching fashion styles, where duplicate item categories as well as mixed demographies were allowed. The standardized FITB evaluation was utilized for experiments, as well as a custom retrieval test and qualitative evaluation by visualizing subspaces in the models using the t-SNE dimensionality reduction algorithm.

Results could conclude that both models showed a strong preference of retrieving relevant items in the custom style retrieval experiment, with recall @ top 10 values of 26.65% for SCE-Net and 29.30% CSA-Net at mixed category retrieval, and recall @ top 50 of 42.49% and 45.25% respectively for the models, with a random model attaining 3.70% at the same tests. FITB results showed the models struggling to reach the same performance as on the outfit matching task on Polyvore Outfits data. SCE-Net reached an accuracy of 46.44%, and CSA-Net 48.91% accuracy using standard hyperparameters, which is inferior to the respective accuracies attained for outfit matching of 61.60% for SCE-Net and 63.73% for CSA-Net. However, this was deemed as reasonable given the probable increase in difficulty for style compatibility modelling due to fewer constraints and thus higher diversity in data.

Further, results indicated both models to be capable of generalizing well to previously unseen fashion styles, as the FITB accuracy did not decrease with

any significance when evaluating using stylistic collections not included in the training split of the data. Through the result of an ablation study, it could also be established that both models performed significantly better in an FITB setting as the dimension of the vector embedding representation increased.

7.1 Future Work

Given the results and analysis of this thesis, further research on the topic of style compatibility modelling and matching fashion styles could potentially yield several improvements and derive new experiments based on findings in this work. Efforts could be made to improve on probable shortcomings in the data, and assure data samples to be distinct in style, either through manual curation by fashion experts or a clever mining technique. Moving beyond solely visual image data, and incorporating more semantic textual metadata in the embedding representation, may as well produce better results and create a solution more capable of differentiating between fashion styles. Suggested product metadata would be price, brand, material composition, and to further utilize the item category in the embedding representation as well.

In terms of model architecture, a more general fine-tuning of the parameters in the chosen CNN backbone would be of interest, rather than solely tune the final layers (top) of the model. In addition, future work could experiment with various backbone architectures, including other pre-trained CNN models, as well as the Vision Transformer (ViT) [58]. As it could be concluded that there exists subjectivity to a certain extent in the ground-truth when training models for style compatibility modelling in a weakly supervised setting, using an altogether unsupervised approach such as utilizing association rules, may produce favorable results.

Finally, as the task of matching fashion styles and providing means to achieve strong models capable of modelling style compatibility gains attention, it would be of great interest to collect and construct a well-functioning, open-source benchmark dataset for the task, where an emphasis is put on collecting a diversity of fashion styles.

References

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[5] Yen-Liang Lin, Son Tran, and Larry S Davis. Fashion outfit complementary item retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3311–3319, 2020.

[6] Reuben Tan, Mariya I Vasileva, Kate Saenko, and Bryan A Plummer. Learning similarity conditions without explicit supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10373–10382, 2019.

[7] Xintong Han, Zuxuan Wu, Yu-Gang Jiang, and Larry S Davis. Learning fashion compatibility with bidirectional lstms. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1078–1086, 2017.

[8] Major, John S. and Steele, Valerie. "Fashion industry". Encyclopedia Britannica, 23 oct. 2020,. https://www.britannica.com/art/fashion-industry. Accessed: 2021-02-07.

[9] Fred Davis. *Fashion, culture, and identity*. University of Chicago Press, 2013.

[10] Fashion Revolution, Fashion transparency index. https://www.fashionrevolution.org/wp-content/uploads/2016/04/FR_FashionTransparencyIndex.pdf. Accessed 2021-03-01.

[11] Shopify Fashion Industry Report, Shopify. https://www.shopify.com/plus/industry-reports/fashion-and-apparel?itcat=plusblog&itterm=ecommerce-fashion-industry. Accessed: 2021-02-11.

[12] Euromonitor. International apparel footwear, volume sales trends 2005-2015, 2016.

[13] House of Commons Environmental Audit Committee et al. Fixing fashion: Clothing consumption and sustainability. *London: House of Commons*, 2019.

[14] Fundación Ellen MacArthur. A new textiles economy: redesigning fashion's future. *Ellen MacArthur Foundation*, 2017.

[15] INDUSTRIALL, Written evidence submitted by IndustriALL Global Union. http://data.parliament.uk/WrittenEvidence/CommitteeEvidence.svc/EvidenceDocument/Environmental%20Audit/Sustainability%20of%20the%20fashion%20industry/Written/92228.html, 2018. Accessed 2021-02-18.

[16] Pauliina Isokangas et al. Global governance in the fashion industry–an analysis of the fashion industry charter for climate action as an instrument of transnational regulation. 2020.

[17] UNFCCC 2015 Paris Agreement. https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en. Accesses 2021-03-05.

[18] Johan Falk, Owen Gaffney, Avit K Bhowmik, Pernilla Bergmark, Victor Galaz, Nick Gaskell, Stefan Henningsson, Mattias Höjer, Lisa Jacobson, Krisztina Jónás, et al. Exponential roadmap 1.5. *Future earth, SITRA*, 2019.

[19] ThredUp. Thredup resale report 2020. https://www.thredup.com/resale/. Accessed 2021-02-22.

[20] Dirk Bollen, Bart P Knijnenburg, Martijn C Willemsen, and Mark Graus. Understanding choice overload in recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 63–70, 2010.

[21] Nima Dokoohaki. *Fashion Recommender Systems*. Springer, 2020.

[22] Hang Yu, Lester Litchfield, Thomas Kernreiter, Seamus Jolly, and Kathryn Hempstalk. Complementary recommendations: A brief survey. In *2019 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS)*, pages 73–78. IEEE, 2019.

[23] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015.

[24] Andreas Veit, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4642–4650, 2015.

[25] Mariya I Vasileva, Bryan A Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. Learning type-aware embeddings for fashion compatibility. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 390–405, 2018.

[26] Wen-Huang Cheng, Sijie Song, Chieh-Yun Chen, Shintami Chusnul Hidayati, and Jiaying Liu. Fashion meets computer vision: A survey. *arXiv preprint arXiv:2003.13988*, 2020.

[27] Kartik Hosanagar, Daniel Fleder, Dokyun Lee, and Andreas Buja. Will the global village fracture into tribes? recommender systems and their effects on consumer fragmentation. *Management Science*, 60(4):805–823, 2014.

[28] Nicholas Walliman. *Research methods: The basics*. Routledge, 2017.

[29] Martin Höst, Björn Regnell, and Per Runeson. *Att genomföra examensarbete*. Studentlitteratur AB, 2006.

[30] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[31] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[35] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[36] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[38] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.

[39] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[40] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[41] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6:737–744, 1993.

[42] Wei Chen, Tie-Yan Liu, Yanyan Lan, Zhi-Ming Ma, and Hang Li. Ranking measures and loss functions in learning to rank. *Advances in Neural Information Processing Systems*, 22:315–323, 2009.

[43] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 830–838, 2017.

[44] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 403–412, 2017.

[45] Si Liu, Zheng Song, Guangcan Liu, Changsheng Xu, Hanqing Lu, and Shuicheng Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3330–3337. IEEE, 2012.

[46] M Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. Where to buy it: Matching street clothing photos in online shops. In *Proceedings of the IEEE international conference on computer vision*, pages 3343–3351, 2015.

[47] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, pages 737–737, 1994.

[48] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[49] Kazuya Kawakami. Supervised sequence labelling with recurrent neural networks. *Ph. D. thesis*, 2008.

[50] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[52] Guillem Cucurull, Perouz Taslakian, and David Vazquez. Context-aware visual compatibility prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12617–12626, 2019.

[53] Maryam Moosaei, Yusan Lin, and Hao Yang. Fashion recommendation and compatibility prediction using relational network. *arXiv preprint arXiv:2005.06584*, 2020.

[54] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[55] Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. opentsne: a modular python library for t-sne dimensionality reduction and embedding. *bioRxiv*, 2019.

[56] ANNOY library. `https://github.com/spotify/annoy`. Accessed: 2021-04-05.

[57] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

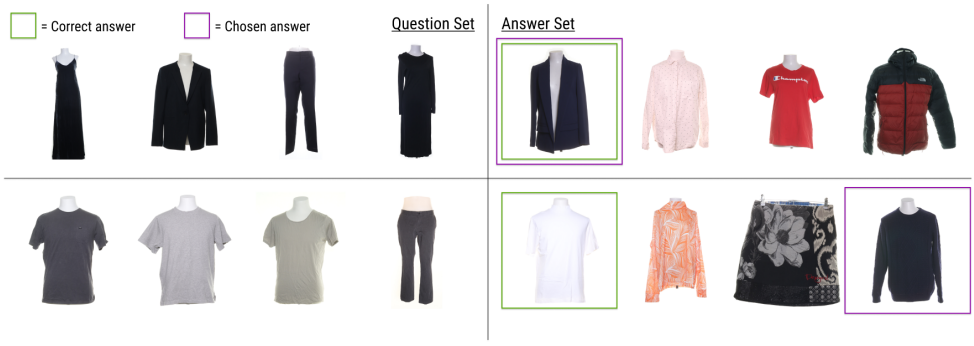
[58] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Appendix A

FITB Additional Results

A.1 SCE-Net

A.2 CSA-Net



Appendix B

Style Retrieval Additional Results

B.1 SCE-Net

TODO

B.2 CSA-Net

TODO

For DIVA


```
{
"Author1": {
            "Last name": "FRÖSSLUND",
            "First name": "LUKAS",
            "Local User Id": "u100001",
            "E-mail": "lukasfro@kth.se",
            "ORCiD": "0000-0002-00001-1234",
            "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                             }
          },
"Degree": {"Educational program": "Masters Programme in Computer Science"},
"Title": {
            "Main title": "Learning Style Compatibility on Fashion Data",
            "Language": "eng" },
"Alternative title": {
            "Main title": "Lärande av Stilkompatibilitet på Modedata",
            "Language": "swe"
},
"Supervisor1": {
                "Last name": "Aghanavesi",
                "First name": "Somayeh",
                "Local User Id": "u100003",
                "E-mail": "somagh@kth.se",
                "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                                 "L2": "Computer Science" }
              },
"Supervisor2": {
                "Last name": "Persson",
                "First name": "Therese",
                "E-mail": "therese.persson@sellpy.se",
                "Other organisation": "Sellhelp AB, Data Engineer"}
              },
"Examiner1": {
                "Last name": "Chiesa",
                "First name": "Marco",
                "Local User Id": "u100004",
                "E-mail": "mchiesa@kth.se",
                "organisation": {"L1": "School of Electrical Engineering and Computer Science ",
                                 "L2": "Computer Science" }
              },
"Cooperation": { "Partner_name": "Sellhelp AB"},
"Other information": {
"Year": "2021", "Number of pages": "??,62"}
}
```