Università degli Studi di Verona

Department of
Computer Science

PhD in
Computer Science
S.S.D. INF/01

Cycle / year
XXXIV / 2018

Title of doctoral thesis

Computational Aesthetics for Fashion

Doctoral Student
Dott. Christian Joppi

Coordinator
Prof. Massimo Merro
Tutor
Prof. Marco Cristani

Contents

List of Figures

List of Tables

Abstract

The online fashion industry is growing fast and with it, the need for advanced systems able to automatically solve different tasks in an accurate way. With the rapid advance of digital technologies, Deep Learning has played an important role in Computational Aesthetics, an interdisciplinary area that tries to bridge fine art, design, and computer science. Specifically, Computational Aesthetics aims to automatize human aesthetic judgments with computational methods. Understanding the appearance of clothes automatically is one of those. Deep Learning meets fashion in many tasks, such as classification (recognize different categories of clothes, describe clothes by semantic attributes), recommendation (understand the preferences of e-commerce customers), retrieval (find the desired clothes or similar ones in a huge catalog), generation (generate/edit clothes) and forecasting (predict sales of clothing items). In this thesis, we focus on applications of computer vision in fashion, and we discuss how Computational Aesthetics helps solve them accurately.

First, we introduce a new way to represent textures, based on a new paradigm that focuses on atomic components called *texels*, elements that are repeated within the pattern. Through simple statistics of texels, we generate a new descriptor made of interpretable and fine-grained aesthetic attributes, that plugged into both image retrieval and interactive image search systems, improves performances. We demonstrate the advantages of working on texels on ElBa dataset, introduced in this thesis. The dataset is composed of synthetic images of element-based textures, exploring a wide variety of colors, spatial patterns, and shapes.

In the second part of the thesis, we present a novel framework for the video-to-shop problem: find a clothing item portrayed in a video, within a huge catalog of shop images. This challenge is a natural extension of the widely explored street-to-shop problem, where the query item is a single image instead of a video. By extending to the time dimension, we are able to extract more information from the video, thanks also to an attention mechanism that focuses on the most salient frames. The framework is trained with a newly designed procedure, that does not require bounding box annotations, and still yields performances higher than existing approaches that require them. The model is trained on MovingFashion, a novel dataset collected from e-commerce and social networks, that we present in this thesis. This allows users to find in an online shop the desired clothing item worn in a video from a fashion influencer or from an ordinary person.

In the third part, we discuss a new challenge: New Fashion Product Performance Forecasting. The goal of this problem is to forecast the future of a new clothing product in terms of sales or popularity. We contribute to this problem, introducing VISUELLE, the first public dataset build upon sales data of a real fast fashion company. This dataset provides a benchmark for forecasting models and in this thesis, for a novel transformer-based architecture, dubbed GTM-Transformer. Compared to standard forecasting tasks, where the past observations are available, new products lack this information. In this thesis, we propose two different insights to fill this missing past. The first is using Google Trends as an exogenous signal, never used in practice in a new product forecasting setting. The second is POP signal, created following a new data-centric pipeline based on capturing the aesthetical similarity of the new product image with respect to the fashionable and unfashionable images with the same characteristics, uploaded on the web in the past. We demonstrate that both exogenous signals are of benefit for accurate performance estimation, especially POP signal that provides the best results.

The contributions introduced and shared in this thesis have many implications for fashion companies that aim to maximize profits while reducing waste, and for users that make e-commerces the main platforms for their purchases.

Chapter 1

Introduction

In recent years, the world of e-commerce has seen a considerable increase, accentuated even more by the arrival of the COVID pandemic [13]. One of the most affected markets was the fashion industry, with marketplaces such as Amazon, Zalando, or Asos. With the exponential growth of these markets, there is also the need to develop intelligent systems capable of improving the user's shopping experience through retrieval, recommendation, or virtual try-on systems. On the other hand, another objective is to predict the performance of clothing products, supporting brands in organizing and planning sales, with the aim of reducing waste while maximizing revenue. Artificial intelligence and Deep Learning have advanced the development of specialized systems that are increasingly able to meet these issues.

An interdisciplinary field that plays an important role in automatically understanding perceptual properties and the attractiveness of clothing images is that of Computational Aesthetics. Computational Aesthetics (CA) is defined as *"the research of computational methods that can make applicable aesthetic decisions in a similar fashion as humans can"* [106]. For the sake of this thesis, the goal of CA ranges from identifying aesthetic factors of texture images (e.g.:semantic attributes), to automatically finding out the human preference within a catalog, or predicting the fashionability of clothing images.

In this thesis, we discuss how Computational Aesthetics together with Deep Learning techniques help to propose solutions for some of the existing artificial intelligence challenges in fashion (FashionAI), discussed below.

First, we tackle the problem of automatically describing textures with perceptual and semantic attributes, a fundamendal component of clothes, introducing a novel framework able to generate a widely informative representation of element-based textures. These representations may be used as relative attributes [117] to improve the existing image search frameworks, which are often sequential and time-consuming. In this way, huge catalogs of images (e.g.: textures), present in the private databases of fabric companies or e-commerce platforms (Fig. 1.1), can be indexed and explored efficiently and effectively [72].

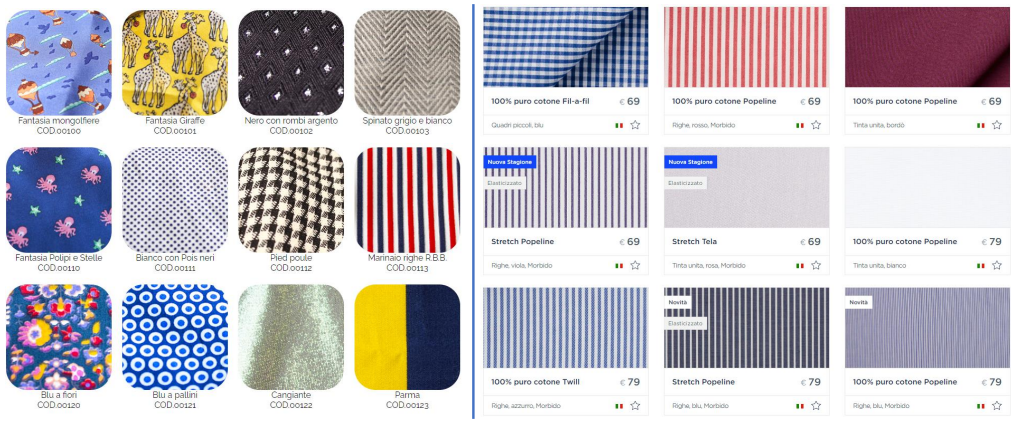

Figure 1.1: On the left, textures sampled from a shop that sells personalized shirts. On the right some of the textures available in a shop that directly sells patterned fabrics. Both of these are examples of shops that has a catalog of textures that a user has to search through.

Second, we consider the task of video-to-shop: starting from a video that depicts the desired clothing item, the system has to find out that particular garment in a gallery of images. It is a natural extension of the widely studied street-to-shop problem [48, 87, 42], where the starting data is a single image instead of a video. In fact, the existing approaches deal with single images instead of composition of frames, leading to worse performances. Other video-to-shop approaches require a lot of annotation data [23], such as bounding boxes for each frame of videos, that comprise a large amount of effort and costs. For this reason, we introduce a novel dataset and a new box-free video-to-shop framework, trained with weak annotation, easily and automatically gathered during dataset collection.

Last, we focus on the challenge of predicting the performance of new clothing products. Estimating product sales or product popularity is a crucial challenge for fashion companies [36]: a good forecast carried out prior to the target season will be helpful in selecting the right amount of items to be ordered, reducing losses, and increasing earnings. The existing forecasting methods [4, 3] follow the standard setup: predict the future given past observations. In this scenario, where the clothing item has been never seen in the market, past observations are not available. In order to face this challenge, we propose an innovative insight and solution, introducing a new deep-learning model that predicts clothing sales using different kinds of exogenous signals, to replace the absent past observation signals, created starting from aesthetic properties of the product image.

1.1 Texture Description

Every day, when we look at any object, we capture three aesthetic factors: color, shape, and texture. For this reason, texture analysis is one of the widely explored fields in computer vision and graphic, with many releases of datasets, which cover different materials [56, 19, 98, 125, 158], such as wood or fabric. In addition to the material, texture can be categorized by classes instead. In DTD [26], the authors separate the textures in 47 different labels, based on visual attributes (*banded, dotted, zigzagged, etc...*), that define more perceptive characteristics than material.

Standard techniques for textures analysis are based on local features, that capture fine-grained characteristics locally, in spatial regions, within the image. These are then aggregated forming a single high-informative descriptor [118, 166, 160, 27, 82]. The major weakness of these types of techniques is that they produce not-interpretable features. The state-of-the-art in texture analysis [118] on DTD Dataset is a descriptor built upon the concept of *histogram layer*, a neural network that locally models the distribution of features, creating a sequence of histograms, aggregated together by the network itself. The final descriptor is a vector of neural network features. Another example is the descriptor defined in [27], made of about 64K values, build upon a Fisher Vector aggregation of local convolutional features. These features are powerful, but not interpretable. One of the proposed methods to obtain interpretable descriptor is described in the same paper [27], by using the classification score for each class. This representation is less powerful but is related to a specific property (e.g. high *banded* and *dotted* scores mean that the textures are formed by both bands and dots).

Texture analysis is fundamental also for the fashion domain, where clothes have often the same shape but have different motifs and colors. Being able to describe textures in a proper way with interpretable descriptors, is important to give precise semantic content-based information that allows users to explore catalogs fastly, finding the preferred item to buy [133] or taking inspiration from it [72]. Attribute-based texture features [101, 14, 26, 83] are explicitly suited to give textures semantic yet discriminative descriptions. The 47 perceptually-driven texture attributes defined in Describable Texture Dataset (DTD) [26] is the most known, together with Tamura attributes [142]. It is worth noting a limitation of these attributes: they describe the properties of a texture image *as it was a whole atomic entity.* In Fig. 1.2a, *dotted* (left) and *banded* (right) attributes are considered. However, the images are strongly different: on the left, the number and the size of dots are clearly different, while on the right the thickness of bands changes dramatically. In Fig. 1.2b, we show an example where both the clothes have the same attribute (*checkered*), but with a visible change of squares' dimensions. In all cases, it is evident that we need attributes that are able to capture finer expressivity, focused on recognizable elements, dubbed *texels* [2], that organized according to specific spatial disposition, formed a particular type of textures called *Element-based* textures [62, 93, 92, 90](Fig. 1.2a-b). They differ from whose textures defined merely at a *micro* scale, *i.e.*, focusing on materials and material properties (see Fig. 1.2c).

(a) (b) (c)

Figure 1.2: (a) Examples of element-based textures in the DTD [26]: *dotted* (left) and *banded* (right) classes are examples where texels are dots and bands, respectively; (b) Zalando shows for each clothing a particular on the texture; (c) examples of DTD [26] textures which are *not* element-based: (*marbled* on top and *porous* on bottom); here is hard to find clearly nameable local entities.

To extract the above-mentioned attributes, in this thesis, we present *Texel-Att*, a fine-grained, attribute-based texture representation and classification framework for element-based textures. Texel-Att is structured as a pipeline that starts detecting the single texels, assigning them *individual attributes*. Subsequently, texels are grouped depending on the belonging class (circle, line, or polygon), and groups of texels receive *layout attributes*. Individual and layout attributes form the Texel-Att final descriptor of the texture, that can be used for classification and retrieval.

The detection core of *Texel-Att* is built upon Mask-RCNN [53] architecture, whose role is detecting Texels. The framework is trained and tested on a novel *El*ement-*Ba*sed texture dataset, *ElBa*. *ElBa* is composed of 30K procedurally-generated realistic renderings, where texels have different primitive shapes, colors, and layout distribution. The resulting attributes make it possible to describe textures with very a high level of detail and interpretabilty.

Working on this project led to another work: the first completely agnostic multi-class object counting approach called SIMCO [43](SIMilarity-based object COunting). The framework is built on Mask-RCNN [53] architecture, with the addition of a new branch called Similarity Head. The model is trained on *ElBa* and in particular, the new branch is trained with triplet loss, in order to embed similar shapes (type, color, and size) close to each other and far from different ones. Since each object is considered as a natural expansion of primitive shape, many objects can be detected by the system. The embedding is used in a clustering procedure to divide objects into groups and count them. The model is evaluated on RepTile [124], a dataset of real images captured in the wild. SIMCO reaches the state-of-the-art on the dataset and paves the way for different and completely class-agnostic applications, presented in the paper. Despite SIMCO works on repeated objects, it is not directly related to fashion, and then it is not presented in the thesis.
In chapter 3, we deeply explained *Texel-Att* framework and show experiments.
In detail we prove that:

- texels can be detected with high precision on *ElBa*, and even on textures in the wild;
- *Texel-Att* is better at ranking textures by their attributes, compared to other possible approaches;
- our attributes are able to discriminate easily between classes of texture that state-of-the-art descriptors are not able to distinguish;
- our attributes can be plugged into an interactive search system, making it faster than using other existing attribute representations;
- the attributes are more robust under difficult conditions (illumination changes, noise, and low resolution) compared to existing descriptors in the task of image retrieval.

1.2 Video-to-Shop

Retrieving clothes that are worn in social media videos is the latest frontier of e-fashion, referred to as "video-to-shop" in the computer vision literature. The aim is to match a social video (Instagram, TikTok) containing one or more given clothing item(s), against an image gallery (Fig. 1.3), potentially an e-commerce database. This procedure fits perfectly the key concept of Computational Aesthetics, as it replaces the human action of deciding when a desired garment shown in a video, in aesthetic terms, is the same or very similar to one in the e-commerce catalog. Understanding where the outfit of a celebrity, social influencer, or an ordinary person can be purchased, turn videos into priceless commercials, in a market where over a billion hours of video are uploaded and viewed on a daily basis [35], for around 3 hours per day [71, 113, 114]. The number of global users that will stream video regularly is estimated to reach 4.5 billion over the next five years [134], displaying the potential of video over static, generic images as a general marketing tool [34].

Video-to-shop is an extension of the *street-to-shop* problem, where the probe data is a single image [48]. On one hand, video-to-shop allows an increase of the available information by adding additional frames as probes. On the other hand, this information could be noisy due to challenging illumination, drastic zooming, human poses, missing data, and multiple people (dis)appearing in the video. Another issue is that a video-to-shop system needs training data with millions of bounding box annotations, linking each box with a shop item [23, 167].

We introduce MovingFashion, the very first publicly available video-to-shop dataset, composed of ∼15K different video sequences, each one related with at least one shop image. Even if some video-to-shop methods in the literature [23] have their code available, their training data are not, so MovingFashion is intended to fill this gap, representing a solid benchmark for the community.

The videos of MovingFashion are obtained from the fashion e-shop Net-A-Porter (10132 videos) and the social media platforms Instagram and TikTok (4723 videos),

Figure 1.3: Sketch of the video-to-shop problem. The input is a video that portrays a clothing item to find in a gallery. The result is a sequence of shop images sorted by their similarity with the input item.

and contain hundreds of frames per shop item, partitioned into a *Regular* and *Hard* setup.

Another contribution is the SElf-Attention Multiframe (SEAM) Match-RCNN, a video-to-shop baseline that individuates products and extracts features in a "street" video sequence by adopting a feature collection and aggregation mechanism, and then matching the products over a "shop" image gallery. SEAM Match-RCNN extends the popular Match-RCNN [42], state-of-the-art in the street-to-shop challenge, by applying image-to-video domain adaptation with the use of a novel Multi-frame Matching Head.

Technically, a pretraining on the image domain of the Match-RCNN enables it to provide initial pseudo-labels for a video sequence, individuating bounding boxes of a particular product. The training on the target domain exploits our Multi-frame Matching Head, which aggregates features by means of a non-local block [152] between different frames, which in turn applies a temporal self-attention mechanism [39] and a scoring function. In this way, an aggregation based on the attention score is used to create a single descriptor for a clothing item. In practice, SEAM Match-RCNN allows training on video data where only the pairs <*street* video,*shop* image> are available, without annotated ground-truth bounding boxes. This policy permits to alleviate an intense annotation effort, which in the case of MovingFashion would have required drawing ∼18M bounding boxes. In the experiments, SEAM Match-RCNN gives the best performances on MovingFashion, against multiple baselines and state-of-the-art techniques. Actually, few frames (10) of a social video are enough to individuate the correct product within the first 5 retrieved items with an accuracy of 80%, making SEAM Match-RCNN a proof of concept for a potential product in e-fashion.

In addition, our approach can be applied to multiple, unrelated street images of the same product, as it does not imply any temporal continuity between frames. This enlarges the range of applicability of the system. For example, on the popular DeepFashion2 dataset, some products have few multiple unrelated street images; by isolating these images in a subset (which we called *Multi* DeepFashion2) we define another scenario where SEAM Match-RCNN also overcomes all of the competitors in the state-of-the-art.

In chapter 4, we describe MovingFashion, SEAM Match-RCNN, the training and the inference procedures. We also show the power of the approach with a wide range of experiments. SEAM Match-RCNN provides a top-5 accuracy of 80% on the MovingFashion dataset, compared to the 73% of the LSTM + binary tree-based AsymNet, the

available state-of-the-art video-to-shop approach [23]. It also overcomes all of the other different comparative approaches and baselines.

Additional experiments are carried out on a subset of DeepFashion2 which we individuated, comprised of 11114 products where few independent takes are available for a shop item, which allows further exploration of our SEAM Match-RCNN via ablation studies and qualitative results, that in turn demonstrate the high interpretability of the proposed approach.

1.3 New Fashion Product Performance Forecasting

Forecasting the performance of fashion products is a typical forecasting application [25, 12]: driven by economic and financial reasons, the ability to anticipate the needs and behavior of customers can make a big difference for commercial activity, especially when large volumes of goods need to be managed.

Unfortunately, standard forecasting approaches require information on the past performance to provide a prediction of the future [4, 61, 75] and this information is available for evergreen products only (e.g., blue shirts), not for new ones (see Fig. 1.4a). In fact, fashion professionals are the only ones that can help, starting from photos or realistic renderings, which we call *probe* images, comparing them with trends as they surface and finally inferring their success [123][1].

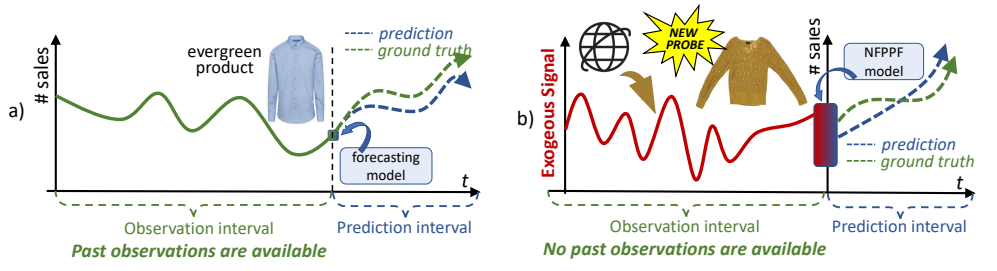

Figure 1.4: a) A standard forecasting setup, where an evergreen item has past observations to exploit, *e.g.,* # sales; b) New Fashion Product Performance Forecasting (NFPPF) problem, where no past observations are available and exogenous data must be considered.

The challenge of automatizing New Fashion Product Performance Forecasting (NFPPF) has started attracting attention in computer vision and machine learning [132, 36]: by exploiting no previous knowledge but the clothing attributes [132] or image data [36], zero-shot learning is essentially applied, under the rationale that new products will perform comparably to similar, older products.

[1]Commercial examples of this process are Trendstop `https://www.trendstop.com/` and its "Trend Platform Membership" service or WGSN `https://www.wgsn.com/en/`, with their study on the Shinkong Textile.

Another direction consists in using textual attributes associated with clothing images to query external sources, by checking the compatibility of the new product with emerging trends, producing exogenous signals.

To face the challenge of New Fashion Product Performance Forecasting, we first introduce a non-autoregressive transformer model dubbed GTM Transformer, which tries to estimate the performance behavior, by modeling the performance of new products based on information coming from several domains (modes): the product image and the textual descriptors of category, color and fabric which are exclusively embedded into the decoder. The above-mentioned exogenous signals are fed into the encoder. This last component is a crucial part of GTM-Transformer, since it introduces external information on item popularity into the reasoning. Intuitively, it models what people are interested in and proves important for forecasting performance.

Secondly, we introduce the first public dataset for new fashion product sales forecasting. VISUELLE is a repository built upon the data of a real fast fashion company, Nunalie[2] and is composed of 5577 new products and about 45M sales related to fashion seasons from 2016-2019. Each product in VISUELLE is equipped with multimodal information: its image, textual metadata and sales after the first release date. We use VISUELLE to compare GTM-Transformer with the few and recent alternatives in the state-of-the-art of new product sales forecasting.
In this thesis, we discuss the use of two different types of exogenous signals.

The first is extracted from the Google Trends API. While it has been already shown that Google Trends signals can be used to predict diverse types of economic activities [156, 18, 52, 49, 47, 10], such as real estate sales to inflation expectations, its adoption to clothing sales forecasting has only been suggested in [128] but never tried in practice, especially in a new product forecasting setting. Technically, we demonstrate that Google Trends are valuable when encoded appropriately. Thanks to the Cross-Attention weights of our model, we find that the most useful information is systematically located around the end of the previous year's same fashion season, i.e., seven to ten months before the product is planned for exposure.

The second embraces the saying "an image is worth a thousand words", comparing the probe image of a new product *directly* with images uploaded on the web in the past, driven by *text tags*, providing what we dub "POtential Performance" (POP) signal. In detail, we propose a *data-centric* [103] solution for the NFPPF problem, based on a cross-modal query expansion. The input is a single probe image of the product to be analyzed, or a photorealistic rendering[3], which can be available 5-6 months before the season's start date [1]. The approach first extracts some textual tags from the probe or directly considers the associated technical sheet. The tag set is expanded with some *positive* and *negative* tags, which are used to perform a *time-dependent* query on the web, that is, collecting images of fashionable and unfashionable items related to the probe, which have been uploaded during certain K_{past} intervals in the past. All

[2] `http://www.nunalie.it.`

[3] Many excellent tools are available nowadays, such as `https://www.tg3ds.com/3d-fashion-design-tools`.

these images are used to train a binary classifier to capture what is fashionable VS unfashionable. Since webly images have noisy labels, a confident learning [111] strategy is designed to prune noisy images and obtain a robust model. Subsequently, in the signal forming step, clean positive images are projected in an embedding space by the learned model, and compared against the (projected) initial probe image, providing the K_{past}-long POP signal. The POP signal indicates how performing the probe could have been if it were available earlier in the past, and how this performance evolved in time. After its formation, the POP signal can be used as an exogenous observation signal to forecast the performance of a new product. The approach should be cast in the field of *data-centric artificial intelligence* (DCAI) [103], since it automates the creation of high-quality training data for improved performance on a given classifier. It is also worth notice that the core of the pipeline, based on capturing what is aesthetically fashionable VS aesthetically unfashionable, is a typical computational aesthetic task.

The POP signal has also been customized to deal with fashion styles (*i.e.,* ensembles of clothing items) on the Fashion Forward benchmark [4]. Fashion Forward calculates a time series for any given style, allowing for standard forecasting.

In chapter 5 we describe the whole contribution introduced in this section, from the GTM-Transformer architecture to the pipeline followed creatin POP signal. Moreover, we present several experiments both with GoogleTrends and POP signals, with an accurate discussion of the results.

1.4 Contributions

In this thesis, we illustrate new datasets, techniques and frameworks able to face fashion domain challenges like clothing texture analysis, retrieval from video and forecasting of new fashion product performance. The industry of e-fashion is experiencing strong growth, and it is important to support companies and consumers with advanced systems. For this reason, the contributions in this thesis have an important industrial impact, paving the way for potential real application. Besides the industrial factors, we propose solutions that can provide interesting insights also for other different fields of Computer Vision, having so a scientific impact as well.
The main contributions of the thesis can be summarized as follow:

- We introduce a new way to represent texture, based on a new paradigm that focuses on the atomic components called *Texels*. From these elements we extract a set of attributes that are versatile as they can be used in multiple applications, including interactive search by comparison on attributes and image retrieval under strong degradation of the images. The pipeline used for feature extraction starts from the detection of texture repeated elements (texels) instead of dealing with raw pixels. This enables a higher-level representation, for which we show the effectiveness through extensive experiments that also include a user study and demos of different applications such as interactive image search and image retrieval. We introduce the ElBa synthetic dataset that we use for training the texel detector.

Experiments are performed on synthetic data and real data.

- We design an architecture to face the video-to-shop problem, a natural extension of the more studied street-to-shop problem. We contribute by releasing Moving-Fashion, the first publicly available dataset, together with SEAM Match-RCNN framework, which achieves state-of-the-art on multiple benchmarks. The idea is to perform clothes detection on a video sequence, and perform tracking to build sequences of detections from which features are extracted. Then, through the attention mechanism, a single descriptor is computed t to represent the whole sequence, discarding noisy detections and focusing on a diverse and high-quality set of frames. The proposed approach also offers an interesting insight as it is trained with weakly labeled data, avoiding a lot of effort in bounding-box annotations. The contributions introduced here, allow us to make a big leap forward compared to single image retrieval, very relevant nowadays, as social media, with promoting videos uploaded by an influencer, plays an important role in fashion advertisements.

- We propose a novel way to face the challenge of forecasting performance of new fashion products. We publish VISUELLE, the first dataset build upon data of a real fashion company. The dataset provides a benchmark for a novel non-autoregressive transformer model dubbed GTM Transformer, based on the standard encoder-decoder architecture. Besides the data and the novel architecture, we provide some interesting insights about how to replace the missed past observations that, for this particular scenario, are not available. We discuss how to use Google Trends as exogenous signals to fill the missed past, and we define a data-centric approach to create high-quality and expressive exogenous signals, based on esthetically fashionable and unfashionable images, uploaded on the web. The exogenous signals are fed into the encoder part of the framework, while the decoder takes as input information related to the new product (images, text tags and release date). The different contributions introduced here have a considerable impact in supporting companies during the process of bringing a new product to market, with a strong reduction in wasted money.

1.5 Outline

The thesis is organized as follows. In the next chapter, we analyze the related literature for each covered topic. In the chapter 3 we focus on attribute-based texture description, introducing and explaining Texel-Att, the framework based on *Texels*, the texture repeated elements (Sec. 3.1). In the same section we give more details about ElBa dataset. In section 3.3 we demonstrate the capability and the effectiveness of Texel-Att attributes with multiple experiments. We evaluate the detection and shape classification accuracy in section 3.3.1 and 3.3.2 respectively. Next, in section 3.3.3, we report the attribute ranking accuracy. Finally, we made use our descriptor for the task of interactive image

search in section 3.3.4 (where we validate its effectiveness with a user study) and for image retrieval in section 3.3.5.

In chapter 4 we present SEAM Match-RCNN. This architecture is trained on MovingFashion, the first publicly available Video-to-Shop dataset, described in section 4.1, showing statistics and detailed data processing. We then explain SEAM Match-RCNN in section 4.2, discussing the architecture and both training and inference procedures. To demonstrate its capabilities, in section 4.3 we test SEAM Match-RCNN on multiple benchmarks: MovingFashion in section 4.3.1 and MultiDeepFashion2 in section 4.3.2. Then we perform qualitative analysis of the attention mechanism in section 4.3.3 and of the retrieval results in section 4.3.4.

In chapter 5 we introduce and discuss the problem of New Fashion Product Performance Forecasting (NFPPF). We start sharing information about how VISUELLE dataset, the first dataset build upon data of a real fashion company has been built (Sec. 5.1). We start talking about exogenous signals in section 5.2, explaining how we use them to solve the problem of missing past observations. In section 5.3 we describe the data-centric pipeline followed to create the POP signal. In section 5.4 we show the GTM-Transformer architecture, explaining each of its components. In order to demonstrate the strengths of the architecture proposed, together with the correct insight of using exogenous signals to fill the missing past information, in sections 5.5.1 and 5.5.2, we show a wide range of experiments. In particular, in section 5.5.1 we report the performances of sales forecasting on VISUELLE using Google Trends as the exogenous signal, under two different setups. In the same section, we also conduce different ablative studies to investigate the contribution of each single input modality. In section 5.5.2 we show the results using POP signal, emphasizing that focusing on data instead of on model, achieve best results. In the same section, we ablate the different choices made in the pipeline design to create the signal. Moreover, in section 5.5.3, as a further contribution, we show how POP signal is able to deal with the Popularity Prediction of Fashion Styles task on the Fashion Forward benchmark.

Finally, in chapter 6, we discuss the overall impact of this thesis, considering also possible future works for all the challenges addressed.

1.6 Publications

Part of this thesis has been published in conference proceedings and submitted as journal contributions. The authors' list order of the papers reflects the contribution each person carried to the results. * means equal contribution.

- "Texel-Att: Representing and Classifying Element-based Textures by Attributes", *Marco Godi*, Christian Joppi*, Fabio Pellacini, Andrea Giachetti, Marco Cristani.* Oral at British Machine Vision Conference (BMVC) 2019. Presented in chapter 3.
- "Texture Retrieval in the Wild through detection-based attributes", *Christian Joppi*, Marco Godi*, Fabio Pellacini, Andrea Giachetti, Marco Cristani.* Oral at International Conference on Image Analysis and Processing (ICIAP) 2019. Presented in chapter 3.
- "SIMCO: SIMilarity-based object COunting", *Marco Godi*, Christian Joppi*, Andrea Giachetti, Marco Cristani.* Poster at International Conference on Pattern Recognition (ICPR) 2020.
- "MovingFashion: a Benchmark for the Video-to-Shop Challenge", *Marco Godi*, Christian Joppi*, Geri Skenderi*, Marco Cristani.* Oral and Poster at Winter Conference on Applications of Computer Vision (WACV) 2022. Presented in chapter 4.
- "Well Googled is Half Done: Multimodal Forecasting of New Fashion Product Sales with Image-based Google Trends", *Geri Skenderi*, Christian Joppi*, Matteo Denitto, Marco Cristani.* Submitted to Pattern Recongnition Journal (2021). Presented in chapter 5.
- "POP: Mining POtential Performance of new fashion products via webly cross-modal query expansion", *Christian Joppi, Geri Skenderi, Marco Cristani.* Submitted to European Conference on Computer Vision (ECCV 2022). Presented in chapter 5.

Chapter 2

State Of The Art

With the rise of Deep Learning techniques, the interest of the Computer Vision community in the fashion domain increased [22]. Earlier works focused on traditional problems such as clothing detection and classification [163, 162, 165, 129, 87, 169, 42, 67], landmarks detection [87, 42, 88], clothing attribute recognition [87, 88, 50, 150] and style classification [131, 70, 141, 57, 95]. ([22] for an extensive survey on computer vision and fashion). Thanks to the ability of Neural Network to learn more abstract concepts, more high-level tasks have been explored, such as content-based image retrieval [85, 48, 87, 42, 59, 151, 96, 73, 69, 79, 23], recommendation [137, 153, 136, 84, 32, 58, 65], virtual try-on [51, 149, 170, 105] and forecasting [4, 3, 36, 130, 161].

In the next sections, we analyze the state-of-the-art for each topic covered in the thesis.

2.1 Texture Description

Texture is an essential cue which characterizes materials and objects [145, 99, 81] and more important for the sake of the thesis, it is a fondamental part in describing clothes [15]. Considering its importance, e-commerce websites often provide a close-up image of the texture of a piece of clothing, as seen in Fig 2.1. Element-based textures [62, 93, 92, 90] are textures formed by nameable recognizable elements, also dubbed *texels* [2], organized according to specific spatial distributions (see Fig. 1.1a,b). They differ from those textures whose main characteristics are defined merely at a micro scale, i.e., focusing on materials properties (see Fig. 1.1c). Element-based textures are of particular importance in the field of fashion, with thousands of products stored in vast catalogs or websites that the user has to explore. In the defined scenario, describing textures and their compositional structure with interpretable and information-rich features, is of primary importance, in order to give a precise semantic content-based description. Texture can be represented by different types of features [81], such as bag of words (BoW)-based [76], CNN-based [41] and attribute-based [26, 142]. Despite the different ways to describe textures, attribute-based representation grew in interest in

the last years, due to its success in image search applications: thanks to attributes like *striped*, *dotted*, *banded*, etc., addressing precise types of textures has become possible in an effective and human-interpretable way, avoiding to rely to numerical-only codes like LBP [115, 97], SIFT [119] or recent approach based on neural network features [118, 166, 160, 27]. However, the existing attribute-based texture representation methods lack in the representation of details, representing the texture as an *atomic entity*. The proposed solution exploits the logic of texels to create a high-detailed descriptor, made by the Texel-Att framework.

Figure 2.1: Examples of an online fashion shop. On the left of each example, images and video provided for that particular item. Among different viewpoints, a zoomed shot of the texture is selected.

2.2 From Street to Video-to-Shop

Street-to-shop task can be defined as a retrieval problem, where, given a picture of a person wearing clothes, the aim is to find the corresponding garments in a set of product images. It is a challenging problem that can be dealt at image level [85, 87] or at instance level [42, 48]. The difference is that at the instance level the garment is first detected and then the retrieval is performed on the chosen detection, instead of that on the whole image. The main concept of the existing approaches, both image or instance level, is to build a descriptor of the street image that is as close as possible to the corresponding shop item. This is usually done in two different ways: by classifying whether two items are a match or not [48], or employing triplet loss [87]. Street-to-shop approaches employed single street images [42, 48, 87], paving the way for video-to-shop methods [23, 167]. AsymNet [23] aggregates frames by exploiting temporal continuity;

Dataset	#Videos	#Trajectories	#Shops	#Pairs	Publicly Available
AsymNet [23]	*526*	*26k*	*85k*	*39k*	✗
DPRNet [167]	*818*	*5k*	*21k*	***n.a.***	✗
MovingFashion	15k	15k	14k	15k	✓

Table 2.1: Comparison of Video-2-shop datasets. *n.a.* stands for *not available*.

it combines an LSTM and a binary tree, with each component requiring a separate training procedure. On the contrary, our SEAM Match-RCNN uses self-attention to learn a descriptor from a bunch of heterogeneous images, where temporal continuity is not required. DPRNet [167] manages the video-to-shop problem by treating it as street-to-shop, with a network that detects and tracks garments in the video, selecting automatically the frame with the highest quality (in terms of occlusions, blurring, etc). That detection is finally fed into an image-to-image retrieval module. SEAM Match-RCNN does not perform this kind of tracking, which could be prohibitive on social videos that have strong heterogeneous variations on a few frames.

Video-to-shop approaches share similarities with video person Re-ID [80], where the goal is to match a video snippet of a person's silhouette against a gallery of image identities taken from a different camera. Two recent approaches that do not make assumptions about the content of the data (such as employing pose estimation or human parsing like in [157, 140]). State-of-the-art approaches are VKD [120], NVAN [80] and MGH [164]. VKD proposes to learn using diverse views of the same target with a teacher-student framework, where the teacher educates a student who observes fewer views. NVAN is based on a non-local block self-attention module, embedded into the backbone CNN at multiple feature levels to incorporate both spatial and temporal characteristics of the pedestrian videos into the representation. Multi-Granular Hypergraph (MGH) is a novel graph-based framework that uses graph networks to cope with this problem.

Several datasets have been proposed for the task of street-to-shop. WTBI [48] and DARN [59] are collected from online shopping websites and they use metadata to extract category labels, making them noisy. The DeepFashion [87], CCP [165], ModaNet [169] and DeepFashion2 [42] datasets have higher quality labels as they are manually annotate.

Unfortunately, no video-to-shop datasets are publicly available. The above-quoted [23] and [167] use proprietary datasets, which have been not made open to the scientific community. We compare these datasets and their reported characteristics with our MovingFashion dataset in Table 2.1. It is visible that the datasets from AsymNet and DPRNet have a moderate number of sequences (526 and 818, respectively), while MovingFashion contains almost thirty times that amount (15K). In order to create more query data, DPRNet and AsymNet sample multiple sequences from the videos (generating 26K and 5K sub-trajectories, respectively). AsymNet contains 39K exact

street-shop pairs and 85K diverse shop items, so one may infer shop distractors are present (shop items not present in the street set) but no details are provided on this. DPRNet has 21K Shop items, with no mentions about the exact pairs. MovingFashion has a single item associated with a unique shop image for each video, for a total of 15K unique (video) street-shop pairs.

The DeepFashion2 dataset (DF2) [42] presents a particular scenario: among the street-to-shop datasets explicitly suited for single image analysis, DF2 is made for the street-to-shop challenge, but some shop items are related to more than one street image (coming from different sources), creating 11K pairings. This provides us with another experimental setting.

2.3 New Fashion Product Performance Forecasting

The New Fashion Product Performance Forecasting (NFPPF) problem has been deeply investigated in the fields of quantitative fashion design [122, 7, 66], marketing and social sciences [128, 40], but is relatively new in the machine learning community. In both [36] and [132], the main idea is that new products will sell comparably to similar, older products. In [132], a variety of boosting algorithms (XGBoost, Random Forest) and Neural Networks (MLP, LSTM) are taken into account, fed with textual attributes related to category and colors, and merchandising factors such as discounts or promotions. Notably, they do not make use of image features or exogenous information. The most related work with ours is [36], where the authors use an autoregressive RNN model that takes past sales, auxiliary signals like the release date and discounts, textual embeddings of product attributes, and the product image as input. The model uses soft-attention to understand which of the modalities is the most important to the sales. The model then embeds and combines all these attended features into a feature vector which is fed to a GRU [24] decoder and used to forecast the item sales. In contrast to our work, [36] do not make use of a "true exogenous" signal such as the Google Trends, the model is based on internal information available in the data. Additionally, the autoregressive nature of RNNs creates prediction curves that have a very common shape across products. Unfortunately, the dataset and the code are proprietary and were not released. With respect to the state-of-the-art, we focus on the additional direction of checking the past to look for predictive exogenous signals. In particular, we exploit two different signals. First, Google Trends, querying textual attributes related to the item to be forecast and embed the resulting trend into the encoder of the GTM-Transformer architecture. Second, we follow the idea of looking back to webly data as well but using web images to represent fashionable items, obtaining a richer exogenous signal. Predicting the success of new fashion styles has never been taken into account, with past works [4, 89, 94] focusing on the standard forecasting setup.

2.3.1 Datasets for fashion forecasting

Publicly available datasets to forecast fashion data take into account diverse applications, dissimilar from new product forecasting. The "Clothing, shoes and jewelry" dataset has been used in [108, 4] to forecast fashion styles, that is aggregates of products of multiple brands, in terms of popularity on Instagram. In our case the problem is different, since we are focusing on *single* products and not on groups of products, so we have definitely less data to reason on. In addition, we are *considering genuine sales data*, and not popularity trends. This makes our research more impactful on an industrial level. The Fashion Instagram Trends [94] adds geographical information to forecast trends in specific places. In our case, Nunalie has shops in two adjacent countries, Italy and Switzerland, and geographical information related to single cities is available in VISUELLE, which for simplicity have not been considered.

2.3.2 Data-centric AI.

Data-Centric AI [103] (DCAI) shifts the attention from the models to the data used to train and evaluate them. It is a topic whose importance is constantly growing in many AI communities [6, 112, 107][1], with important effects on CV & ML. In general, DCAI investigates methodologies for accelerating open-source dataset creation, in particular from low-quality resources. Consequently, it is tightly coupled with learning on noisy data, which aims at producing consistent and low noise data samples, or removing labeling noise or inconsistencies from existing data [111, 135, 154]. Our methodology is data-centric, since it automates the creation of training data from web resources while removing labeling noise. Notably, it represents a novelty in the DCAI panorama, since it creates *time-dependent* training data, *i.e.,* signals which are valid for a particular time interval, as it is required by NFPPF and in general by forecasting tasks.

[1] `https://datacentricai.org/`.

Chapter 3

Texel-based Texture Description

In this chapter, we introduce Texel-Att, a fine-grained, attribute-based representation and classification framework for element-based textures. The pipeline's framework starts individuating texels, describing them with individual attributes; subsequently, texels are clustered by shape class and characterized through layout attributes. The set of individual and layout attributes makes the Texel-Att representation. This representation can be used for classification, retrieval or plugged into an interactive image search system. It is worth noting that Texel-Att descriptor has no pre-defined dimensionality, as it depends on how many and which attributes one does use. In this thesis, we use a pre-established set just to illustrate the general framework. We evaluate Texel-Att on the first *El*ement-*Ba*sed texture dataset, *ElBa*. *ElBa* is composed of procedurally-generated realistic renderings, where we vary in a *continuous* way element shapes and colors and their distribution, to generate 30K texture images with different local symmetry, stationarity, and density of (3M) localized texels, whose attributes are thus known by construction.

3.1 Texel-Att Framework

In Fig. 3.1 a block diagram of the Texel-Att descriptor creation pipeline is shown. Briefly speaking, a customized region proposal method processes input images, extracting texels that subsequently are assigned with *individual* attributes, *i.e.*, labelled according to specific texel categories, and characterized with properties related to appearance and size. Individually labeled texels are then grouped, filtered (discarding non-repeated texels) and *layout* attributes describing the spatial layout of groups are estimated. Individual and layout attributes form the composite Texel-Att descriptor. In the following, each processing block is detailed.

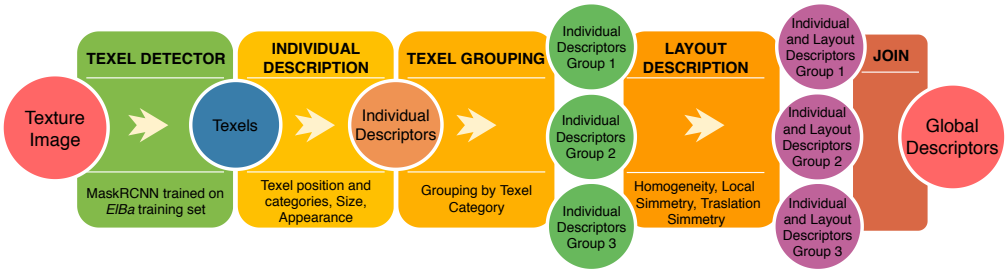

Figure 3.1: Block diagram of the formation of the Texel-Att element-based texture descriptor. On the bottom of each plate, the specific choices made in this thesis, which can be varied.

Texel Detector. The texel detection is built on the Mask-RCNN [53] model, which localizes and classifies objects providing bounding boxes and segmentation masks. We learn the model on the training partition of the *ElBa* dataset, allowing to detect and classify as *lines*, *circles*, *polygons* potentially each texel in a given image (see Sec. 3.2). The message here is that the texels, whose detection a few years ago was quite complicated and limited to specific scenarios (*i.e.*, lattices [46, 86]), are now easily addressable in whatsoever displacement.

Individual description of texels. Each detected texel is characterized with aesthetic attributes related to shape properties and human perception, and in particular:

(i) the *label* provided by the Mask-RCNN, indicating its shape;

(ii) main *color* given by a color naming procedure [146] (with 11 possible colors);

(iii) element *orientation*, if any;

(iv) element *size*, estimated as the area of the region mask.

Textures can be characterized by statistics computed on these features (averages or histograms, see in the following). It is worth specifying that different individual features and statistics could be adopted; in fact here we are not looking for "the best" set of features, but we are showing the portability and effectiveness of the general framework.

Texel Grouping. The goal is to cluster texels with the same appearance, to capture choral spatial characteristics via layout attributes. Here we simply group texels according to the assigned class labels (*circle*, *line* or *polygon*). Only groups including at least 10 texels are kept, the other detections are removed.

Layout description of texels. To describe spatial patterns of each texel group, we measured attributes related to the spatial distribution of the texels' centroids. Among the huge literature in statistics on spatial points patterns' analysis to evaluate randomness, symmetry, regularity and more [31, 148, 8], we selected a simple yet general set of measurements. They are:

Individual Attributes	
Attribute Name	**Dimensionality**
Label Histogram	3
Color	11
Orientation Histogram	3
Size	1
z	
Total	**18**

Layout Attributes	
Attribute Name	**Dimensionality**
Density	1
Homogeneity	1
Vector Orientation	3
Local Symmetry	1
Translation Symmetry	1
Background Color	11
Total	**18**

Table 3.1: Dimensionality of descriptor attributes. On the left, the attributes computed from the individual characterization of texels; on the right, attributes computed from statistics resulting from the spatial layout. The total dimensionality of the descriptor is 36.

(i) Point *density*, *e.g.* the number of texels per area unit (for circles and polygons) or line density, *e.g.* the number of lines/bands along the direction perpendicular to their principal orientation (for lines);

(ii) Quadratic counts-based *homogeneity* evaluation [64]: it amounts to divide the original image into 100 patches and perform a χ^2 test to evaluate the hypothesis of average point density in the sub-parts. Also in this case for lines, we estimated a similar 1D feature on the projection.

(iii) Point pair statistics [168]: we estimate the point pair vectors for all the texel centers and then use them to estimate the histogram of *vectors orientation*.

(iv) *Local symmetry*: for circles and polygons we considered the centroids' grid and measured average reflective self-similarity of 4-points neighborhoods of points after their reflection around the central point. The distance function used is the average point distance, normalized by neighborhood size.

(v) *Translational symmetry*: given the spatial distribution of the centroids of the detected dots and polygons, we characterize their average center-reflective and translational symmetry as follows. For the average reflective symmetry, we consider for each centroid a 4-points neighborhood, reflect the points of the neighborhood with respect to the center and estimate the distance from the closest centroid grid point. The average center-reflective symmetry score is

$$S(R) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{4}\left|\vec{R}_i(\vec{n}_{ij}) - \vec{c}_i\right|$$

where N is the total number of centroids in the group, $\vec{c}_i$ is a centroid, R_i is the reflection around centroid i, $\vec{n}_{ij}$ is the $j-th$ nearest neighbor of the centroid i (Fig. 3.2 left).

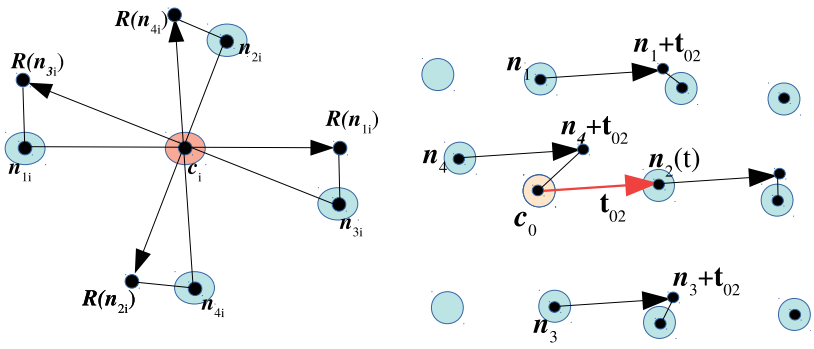

Figure 3.2: Symmetry scores are evaluated as an average of local self-similarity of elements' centroid patterns after translation of 4-point neighborhoods of point pairs vectors included in the neighborhood (left) and after reflection with respect to the central point (right).

For the average translational symmetry, we consider for each centroid a 4-points neighborhood, translate the points of the neighborhood of all the $\vec{t_j}$ vectors joining it with the neighbors and estimate the distance from the closest centroid grid point. The average translational symmetry score is

$$S(T) = \frac{1}{4N} \sum_{i=1}^{N} \sum_{j=1}^{4} \sum_{k=1}^{4} \left| \vec{n}_{ik} - \vec{c}_i + \vec{t}_j \right|$$

Where N is the total number of centroids in the group, $\vec{c}_i$ is a centroid, R_i is the reflection around centroid i, $\vec{n}_{ij}$ is the $j-th$ nearest neighbor of the centroid i (Fig. 3.2 right). Symmetry descriptors are computed on 1D projections for line texels.

The complete pattern descriptor is finally built joining texel attributes, spatial pattern attributes and the color attributes of the *background*. The dimensionality for each of these attributes is reported in Tab. 3.1. 1-dimensional attributes are averages of all the extracted values, while multi-dimensional ones are histograms. The Texel-Att descriptor is composed by concatenating the attributes and Z-normalizing each one of them.

3.2 Textures Datasets

While element-based textures are common and relevant to many practical applications (see Fig. 1.1), no public database focused on this texture domain is available. Existing databases as the DTD [26] include some examples of these textures (Fig. 1.1(a)), but these are mixed with other texture types. For these reasons, we build *ElBa*, the first dataset of element-based textures, and *E-DTD*, the element-based portion of the DTD dataset.

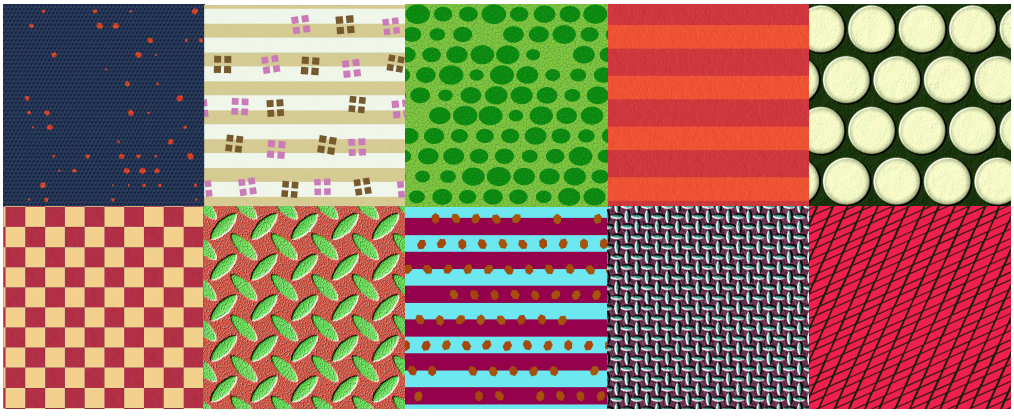

Figure 3.3: Images from the *ElBa* dataset.

3.2.1 ElBa dataset

ElBa includes synthetic photo-realistic images, like those shown in Fig. 3.3. The advantages of dealing with synthetic textures are the precise annotations for texels by construction, and the possibility to train adequately a deep classifier, since training with synthetic data is a common practice [144, 11]. In particular, we propose a parametric synthesis model where we vary both texel *individual* (addressing the single texel) and *layout* attributes (describing how groups of texels are mutually displaced).

As for individual attributes, we vary texel shape, size and orientation and color as follows. For the shape, inspired by the 2D shape ontology of [109], we consider general regular entities as *circles*, *lines*, *polygons* (squares, triangles, rectangles) since they can be thought of as approximations of more complicated shapes and because they encompass a large variety of geometric textiles. Size and orientation are varied linearly over a bounded domain. Colors are chosen from harmonized color palettes to better represent realistic use of colors. Texels are placed in 2D space based on a variety of layouts that can be described succinctly using symmetries. We consider both linear and grid-based layouts where the layout attributes are defined by one or two non-orthonormal vectors that define the translation between texels in the plane. This simple description represents several tiling of the plane and their corresponding patterns. We consider both regular and randomized distributions, where the randomization is performed by jittering the regular grid. By using jittering, we create a continuum between regular and non-regular distributions, and by varying the jitters per-texel we can change the stationarity.

Importantly, we take into account distributions of more than one element type arbitrarily combined in the plane. For example, we can create dotted+striped patterns. Each texel type has its own spatial layout attributes effectively creating arbitrary multi-class element textures (Fig. 3.3).

We generate the images of *ElBa* using state-of-the-art computer graphics tools. We

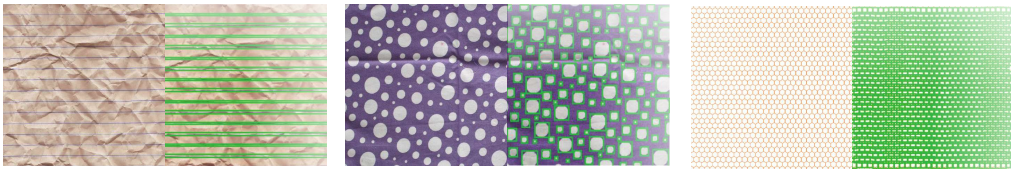

Figure 3.4: Examples of E-DTD annotations: ground truth green bounding boxes overlayed to images of the classes *lined, dotted, honeycombed*, respectively.

use Substance Designer for pattern generation [1]. Substance gives high-quality pattern synthesis, easy control and high-quality output including pattern antialiasing. High frequency patterns simulating realistic materials are added to the generated images.

This procedure led to a rough total of 30K of diverse textures rendered at a resolution of 1024×1024. The texture design process automatically provides ground-truth data for our analysis including texels masks, texels bounding boxes, and spatial distribution attributes. In total this amounts to around 3M annotated texels. It is very important to note that, differently from the other datasets used in the texture analysis domain, *ElBa* does not come with a rigid partition into classes: semantic labels that define relevant classification tasks like those used in our tests (Sec. 3.3.2) can be derived by texels' attributes or by experiments with subjects.

The complete dataset is split in a training part (90%), used to train the network model and a test part (10%), used to validate the classification and recovery experiments.

3.2.2 E-DTD dataset

To demonstrate our approach on real images, we created the Element-based DTD (E-DTD) as follows. From the DTD, we extracted textures that are element-based, i.e. characterized by a distribution or recognizable repeated texels with limited perspective distortions. We manually annotated the bounding boxes of each texel. E-DTD includes 1440 images belonging to 12 of the original DTD classes: *Banded, Chequered, Dotted, Grid, Honeycombed, Lined, Meshed, Perforated, Polka-dotted, Studded, Waffled.* These classes have been selected by 7 experts (3 graphic designers and 2 fashion experts and 2 computer scientists) with all of them agreeing on their inclusions. DTD classes with partial consensus have not been inserted into E-DTD. The annotation of texels was carried out by Mechanical Turk, borrowed from the three-phase ImageNet crowdsourcing annotation protocol [139]. The protocol consists of (1) a drawing phase, (2) a quality verification phase, where a second worker validates the goodness of the bounding boxes and (3) a coverage verification phase where a third worker verifies whether all object instances have bounding boxes. The annotation process produced around 900K texels annotations, some of which are shown in Fig. 3.4. It is important to note the very diverse types of bounding boxes, from very long and thin (addressing line texels) 745×5 pixels bounding boxes to very small 5×5 pixels bounding boxes (on tiny circles).

[1] `https://www.allegorithmic.com/`

3.3 Experiments

Experiments focus on five aspects:

1) *detection of texels*, where we show that finding texels is nowadays possible, with a Mask-RCNN trained on *ElBa*;

2) *classification*, where we point out the failure of state-of-the-art descriptors in distinguishing textures which are clearly diverse against our Texel-Att that instead is succeeding;

3) *ranking*, where we demonstrate that Texel-Att representation ranks texture w.r.t. expressive yet fine-grained attributes;

4) *image search*, where the Texel-Att attributes are exploited for accelerate human-in-the-loop image search [72] onto large image corpora.

5) *retrieval*, where we highlight the effectiveness of Texel-Att in retrieval under simulated real-world conditions.

3.3.1 Detection of texels

Detection performances have been computed on the testing partition of *ElBa* and on the whole E-DTD. The Mask-RCNN model used in these experiments has been trained on the training partition of *ElBa*. Fig. 3.10 reports some Texel-Att detection qualitative results, while Tab. 3.2 reports *per-image* average precision (AP): in practice, AP is computed *for each image*, and averaged over all the images, since we are interested in capturing how much *all of the texels of a single image* are detected, since it is crucial for computing the Texel-Att attributes afterwards. E-DTD dataset gives lower results, since it contains images with dramatic perspective deformation (see Fig. 3.10(a)), which was not a factor in the *ElBa* training data. Despite this, the next experiments show that such detection performance is enough to estimate attributes with high accuracy. Mask-RCNN trained on COCO gives dramatically low results (mAP = 1.75e-6), due to the completely different scenario, not reported in Tab. 3.2 for clarity. One may ask how Texel-Att detection works on a texture which is not element-based, like Fig. 1.2d. Few tests showed that the confidence of the detections, in that case, is definitely lower than in the element-based case.

Dataset	**mAP**	**AP50**	**AP75**
E-DTD	0.53	0.63	0.40
ElBa	0.91	0.92	0.90

Table 3.2: Detection *per-image* average precision (see text) on E-DTD and *ElBa* datasets.

3.3.2 Classification

On texel classification into *circles, polygons, lines* categories, Mask-RCNN scores a 99.85% of accuracy on almost 550K of correctly detected (IoU>0.5) texels. Class labels

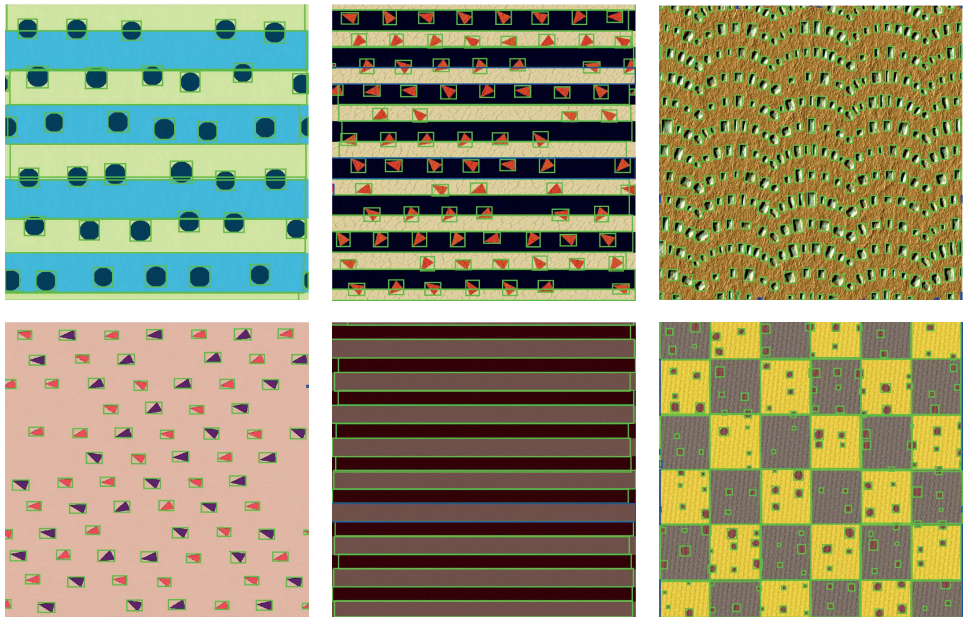

Figure 3.5: Detection results on *ElBa* dataset. Green color is used for the correct detections, blue for false negative detections and red for the false positive detections.

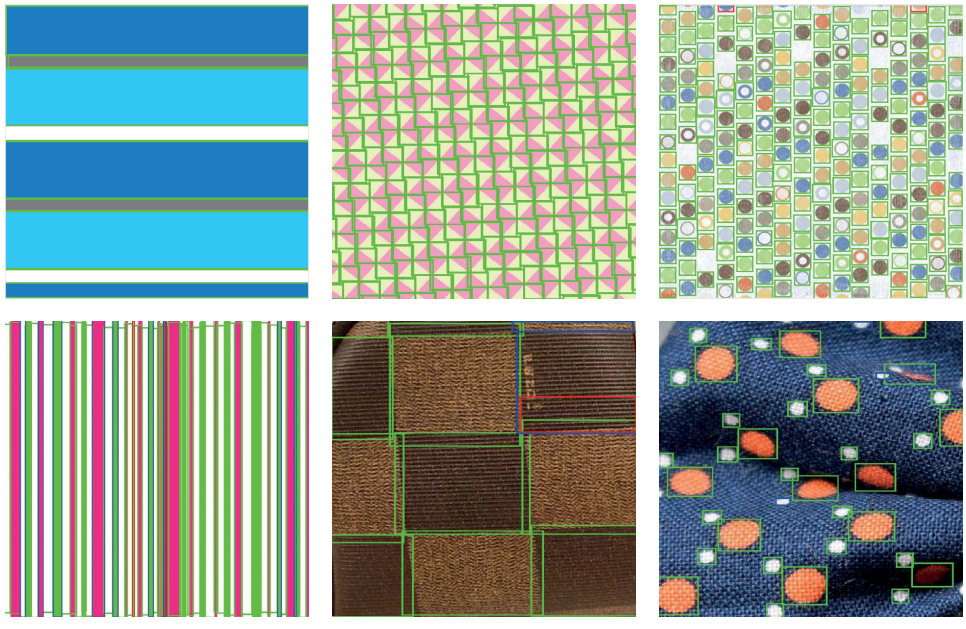

Figure 3.6: First column: Banded; Second column: Chequered; Third Column: Dotted. Green color is used for the correct detections, blue for false negative detections and red for the false positive detections.

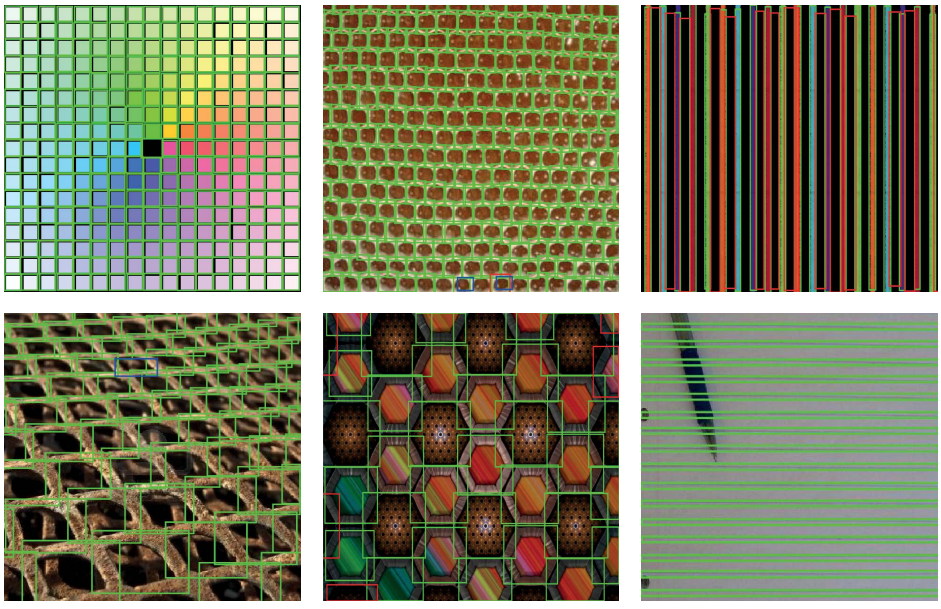

Figure 3.7: First column: Grid; Second column: Honeycombed; Third Column: Lined. Green color is used for the correct detections, blue for false negative detections and red for the false positive detections.

are used to organize texels into groups, as described in Sec. 3.1. Groups are described with layout attributes, and this completes the Texel-Att description. The Texel-Att description allows to communicate about visual aspects of a texture that are apparent yet unreached by the current literature. Three simple experiments on binary classification demonstrate the expressivity of Texel-Att, each one focusing on 200 *ElBa* images having strongly different attributes, that is *single-color VS bi-color circles*, *regularly VS randomly positioned circles* and *lines with uniform or non-uniform width* (see Fig. 3.11). Cross-validated 5-fold experiments compare the 36-dimensional Texel-Att description (see Tab. 3.1) against CNN+FV (65536-dimensional) and the Tamura [142] classic texture description. All of the three descriptors are fed into linear SVMs. Accuracies are shown in Tab. 3.3.Texel-Att obtains the best results as it captures higher visual semantics, i.e., the texels and how they are mutually related. FVs and Tamura features are not able to capture objects and spatial layout, focusing on filter outputs or directly on pixel values.

On E-DTD, Texel-Att description individuates strongly different textures within the same class, ideally defining further, finer-grained level of classes it can separate. For example the *dotted* and *banded* classes (see Fig. 1.2) are now further specified and classified considering big or small dots, regular or irregular bands.

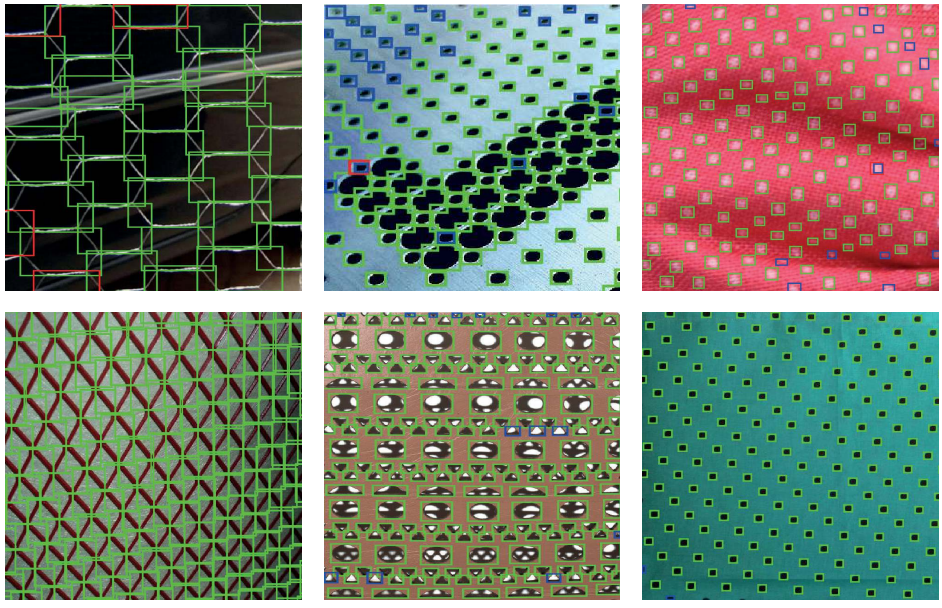

Figure 3.8: First column: Meshed; Second column: Perforated; Third Column: Polka-Dotted. Green color is used for the correct detections, blue for false negative detections and red for the false positive detections.

Classes	**Tamura [142]**	**FV-CNN** [27]	**Texel-Att**
Line uniformity	70.60	66.80	**85.30**
Circle positioning	54.86	53.01	**97.33**
Circle coloring	50.19	52.94	**93.42**

Table 3.3: Classification accuracy in three different binary tasks with three different approaches.

3.3.3 Ranking

Other than capturing fundamental properties of textures (for example, having regularly VS randomly placed texels, see the previous section), Texel-Att attribute values can be used to rank textures. Attributes that can be ranked are the basis for human-in-the-loop search strategies [72], so it is crucial that the ranking is reliable. Ideally, with the ground-truth texel detection the ranking via Texel-Att attributes will be perfect. In this experiment we evaluate how our detection step corrupts the ranking, and whether the ranking can be better estimated with learning based strategies, avoiding the detection step.

For simplicity, we consider here partial ranking; an attribute that induces partial ranking is said *relative* [117]; formally, given a set of images I=$\{i, j\}$ and an *ideal* Texel-

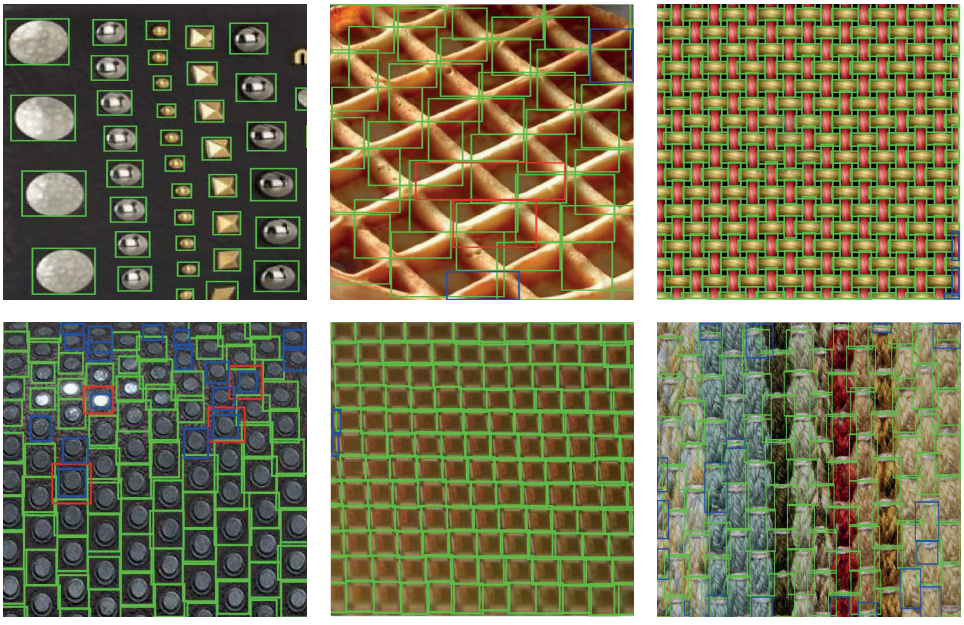

Figure 3.9: First column: Studded; Second column: Waffled; Third Column: Woven. Green color is used for the correct detections, blue for false negative and red for the false positive detections.

Att attribute a (*i.e.*, computed on the ground-truth texel detection) , there exists a partial order relation r_a^* such that $i > j \iff r_a^*(i) > r_a^*(j) \wedge |r_a^*(i) - r_a^*(j)| > \gamma_a$.

The goal of this experiment is to estimate a function r_a(i) as close as possible to r_a^*. Following the protocol of [117] the *ranking accuracy* of the function r_a is defined as the percentage of pairs correctly ordered by the r_a function over all the possible pairs in the set of images.

Two are the strategies we compare to estimate r_a: the first is the Texel-Att pipeline, which measures the attribute on top of the texel detections. The second is the relative attribute estimation of [117] using the FV-CNN [27] descriptor as input. For this second strategy we can assume x_i as the feature vector in $\mathbb{R}^n$ for the image i. In this case r_a is *estimated* by a ranking SVM, following the guidelines in [117]. We assume $\mathrm{r}_a = w_a^T x_i$ so that the output of the modeling is the unknown vector w. The model is trained using the set of ordered pairs of images $O_a = \{(i,j)\}$ where $(i,j) \in O_a \Rightarrow i \succ j$ and the set of un-ordered pairs of images $S_a = \{(i,j)\}$ where $(i,j) \in S_a \Rightarrow i \sim j$.

We perform this experiment on the *ElBa* dataset (using the partitioning defined in Sec. 3.2) and the E-DTD dataset (randomly choosing 90% of the images as training set and the rest as testing set). We consider one attribute at a time. Ground truth r_a^* (i.e. ordered and un-ordered pairs) are computed from the ground truth detections also used in Section 3.3.1. The ranking accuracy across all attributes is shown in Tab. 3.4. It can be clearly seen that computing explicitly texel detection (*i.e.*, following the Texel-Att pipeline) is the best strategy to rank textures according to the proposed attributes.

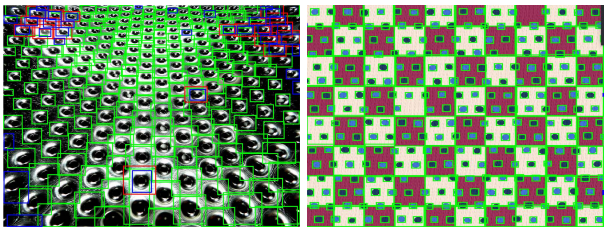

Figure 3.10: Texel-Att detection qualitative results on both E-DTD (left) and *ElBa* (right) datasets. In green the correct detections, in red the false positives (19 in the first, 0 in the second) and in blue the false negatives (35 in the first, 3 in the second). The AP(IoU=.50) are 0.81 and 0.99, respectively.

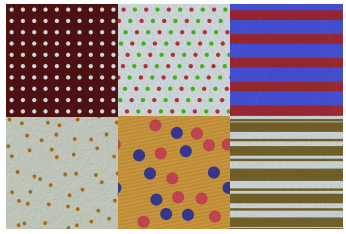

Figure 3.11: Images from the *ElBa* dataset, in columns: mono-colored regular and random circles, bi-colored regular and random circles, uniform and non-uniform lines.

E-DTD			ElBa		
Attributes	**Rank SVM [117]**	**Texel-Att**	**Attributes**	**Rank SVM [117]**	**Texel-Att**
% of Circle Texels	90.40	**100.00**	% of Circle Texels	72.71	**98.86**
% of Line Texels	89.66	**100.00**	% of Line Texels	81.56	**98.43**
% of Polygon Texels	86.61	**100.00**	% of Polygon Texels	71.66	**98.14**
Background Color	66.74	**90.71**	Background Color	63.02	**93.08**
Area	65.52	**85.71**	Area	70.16	**93.86**
Density	90.55	**95.14**	Density	81.86	**96.86**
Texel Orientation	66.32	**69.16**	Texel Orientation	63.47	**83.67**
Texel Color	66.74	**94.66**	Texel Color	62.91	**99.49**
Homogeneity	79.91	**90.29**	Homogeneity	77.06	**96.86**
Local Symmetry	54.72	**64.29**	Local Symmetry	71.06	**81.71**
Translational Symmetry	50.82	**59.57**	Translational Symmetry	63.57	**87.14**
Translation Histogram	57.47	**68.81**	Translation Histogram	68.95	**73.95**
Mean	67.56	**81.68**	Mean	65.95	**84.55**

Table 3.4: Ranking accuracy of relative attributes on E-DTD (left) and *ElBa* (right) datasets.

3.3.4 Texture Interactive Search

In this experiment we follow the Whittlesearch (WH) feedback scheme [72] to search a texture among a large repository. It can be considered a coarse-to-fine user-initiated and iterative search, with each iteration at time $t = 1, \ldots, T$ presenting on a GUI the *target* image, simulating the user's envisioned picture, and a *reference set* $\mathcal{T}_t$ of $n = 8$ images the user has to interact with by giving a feedback. At each iteration, top-ranked images are shown, until the target is ranked in the top n images, or the maximum iteration $T = 10$ is completed. The WH scheme enriches traditional binary relevance feedback mechanism [72] by allowing the user to whittle away specific irrelevant portions of the visual feature space, pinpointing *how* different one image in $\mathcal{T}_t$ is w.r.t. the target by using relative comparisons ("more", "equally", "less"), on a provided set of attributes. To prove that introducing Texel-Att attributes is beneficial to better describe textures, we set up a task of Interactive Image Search following the non-Active WhittleSearch

Figure 3.12: From texture with the smallest average texels' area (left) to the texture with the biggest one (right).

Figure 3.13: From texture with the vertical orientation (left) to the texture with the horizontal one (right).

variant [72], using our 36 attributes (see Table 3.1) estimated on top of the *detection* step. We compare with the attributes extracted from *ground-truth* annotations to understand how much a more accurate estimation leads to a better performance. We also compare with the 47 DTD attributes [26] employed here in their *relative* form (i.e., each attribute has a ranking function indicating how much it is expressed in the image) [117] and the 6 Tamura [142] attributes (*coarseness, contrast, directionality, linelikeness, regularity, roughness*). In particular we compare with three different variants: the 47 DTD attributes, the 6 Tamura attributes and the 47+6 DTD+Tamura attributes. The last combination is the most appealing, since the DTD attributes indicate the content of a

Figure 3.14: From texture with the most regular layout disposition (left) to the texture with the least regular one (right).

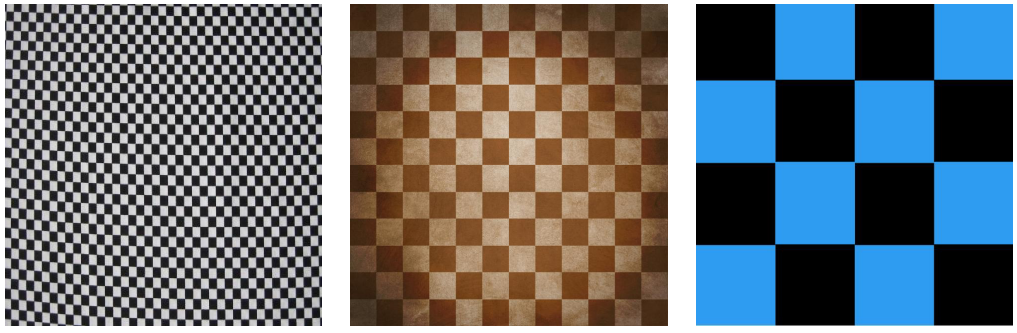

Figure 3.15: From texture with the smallest average texels' area (left) to the texture with the biggest one (right).

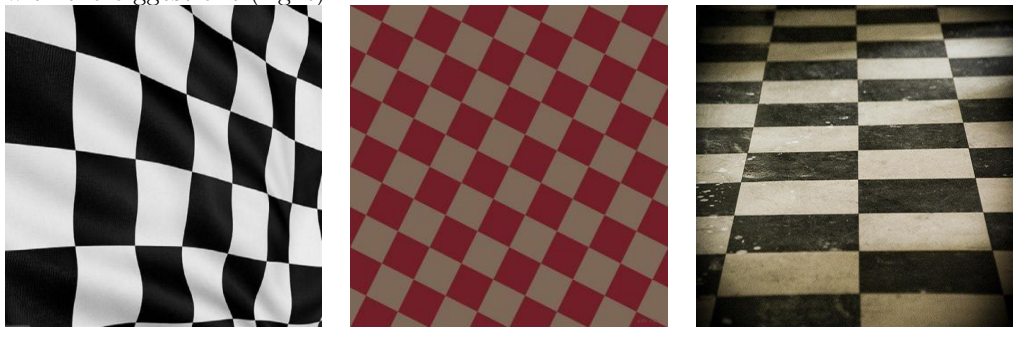

Figure 3.16: From texture with the vertical orientation (left) to the texture with the horizontal one (right).

texture (*i.e.*, dots) and the Tamura attributes models low-level characteristics similar in spirit to ours (*e.g.*, regularity) but computed on the pixels and not on texels. For all of these, each user is presented with a randomly chosen target image from the database (in this case, the *ElBa* dataset presented in Section 3.2) . The goal is to navigate the database until the target image is found (i.e. it becomes one of the top n most relevant images in the database). A total of 50 unacquainted users participated in the study (mean age: 24, std: 1), after having performed a brief individual training session on the use of the interface. Each user had three trials and performances are averaged. Users were partitioned equally among the five approaches taken into consideration in

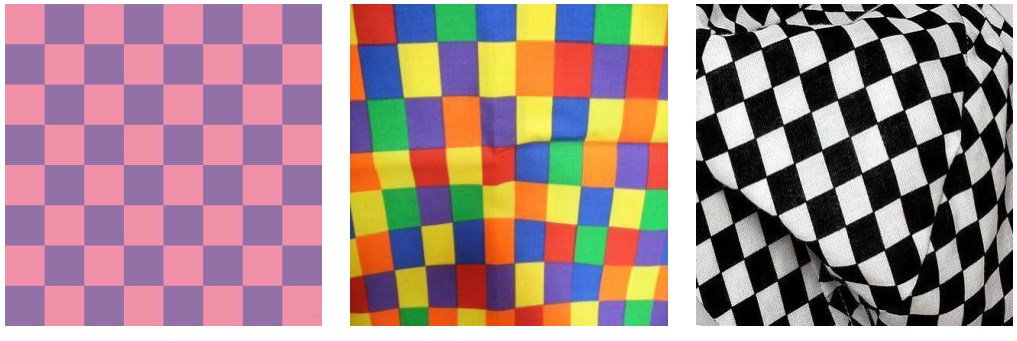

Figure 3.17: From texture with the most regular layout disposition (left) to the texture with the least regular one (right).

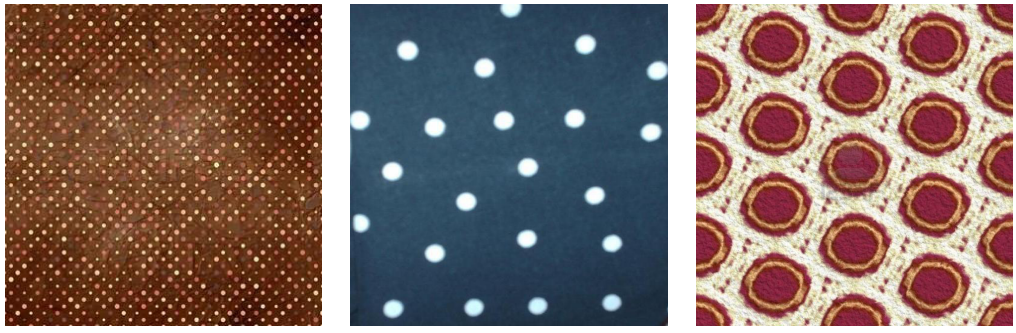

Figure 3.18: From texture with the smallest average texels' area (left) to the texture with the biggest one (right).

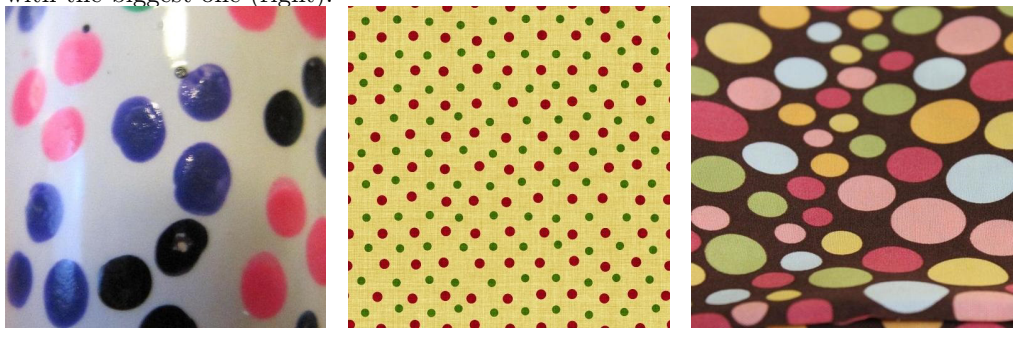

Figure 3.19: From texture with the vertical orientation (left) to the texture with the horizontal one (right).

this experiment.

Following [72], performance is measured using the percentile rank of the target image (i.e. the fraction of the database images ranked below the target) after a fixed number of interaction steps. The closer to 100%, the better the result. We also compute Search Accuracy: by considering 40 images as the size of a typical image search page [72], a texture is considered as "found" by the search if it is ranked among the first 40 images. Keeping this in mind we find that on the E-DTD dataset we are able to individuate within 10 iterations the desired texture in the 90% of cases while using the most performing variant's (DTD+Tamura) attributes accuracy drops down to 71%.

Figure 3.20: From texture with the most regular layout disposition (left) to the texture with the least regular one (right).

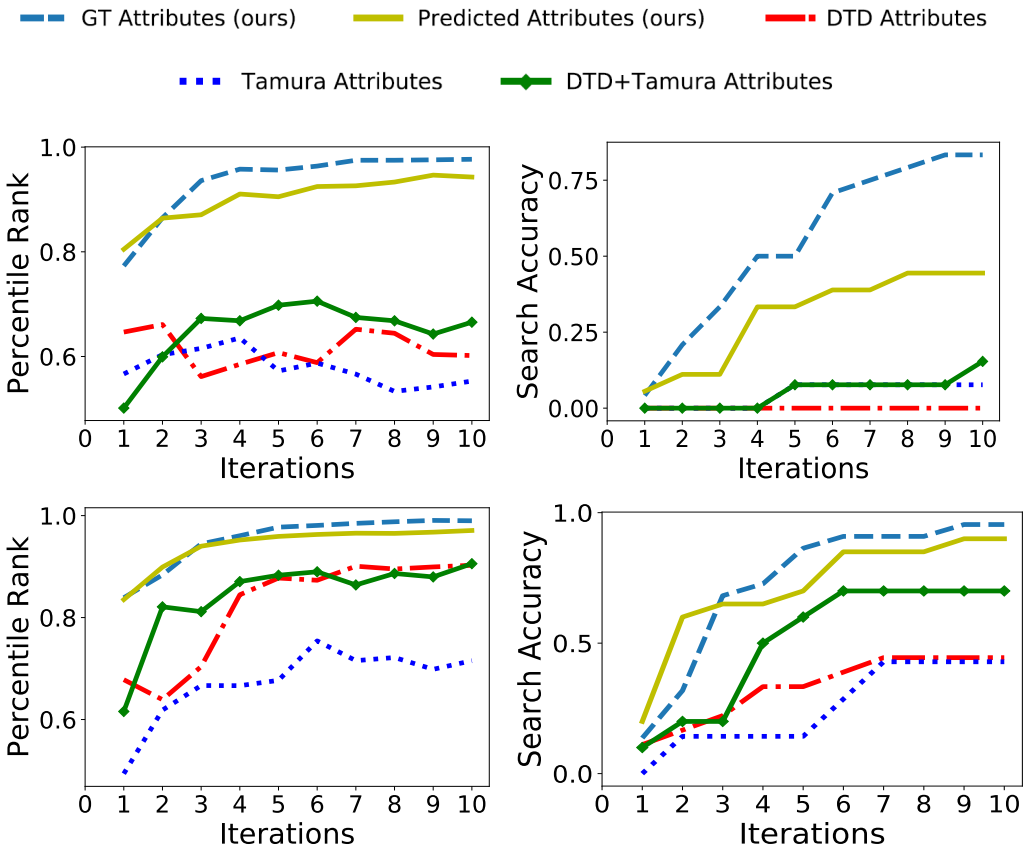

Figure 3.21: Texture Interactive Search (TIS) Percentile Rank and Search Accuracy results on *ElBa* (first row) and E-DTD (last row). On the x axis the number of feedback iterations. On the y axis the Percentile Rank index/Search accuracy score.

On the *ElBa* dataset, which is much more challenging on this task due to the larger size of the pool of images and the finer grained nature of the textures, we are able to reach 44% accuracy, while the DTD+Tamura attributes reach only 15%. The plot in Fig. 3.21 shows that Texel-Att has the best performance at any iteration. In addition, a good performance is preserved even in the case of predicted attributes, confirming the robustness of the approach against imperfect texel detection. Finally, on average we are able to individuate the desired texture more often with our approach than with other techniques.

3.3.5 Texture Retrieval

In this experiment, we highlight the effectiveness of Texel-Att in a retrieval task under simulated real-world conditions. The pipeline is as follows: a query image (e.g. a picture of a textured captured by a user) is processed by the Texel detector, allowing for the computation of individual and layout attributes and thus obtaining a descriptor. A standard distance function (such as cosine distance) is computed between every database image and the query image. The database set is then sorted according to the distance and the resulting ranking can be shown to the user for browsing. We compare our approach with both state-of-the-art texture descriptor FV-CNN [26] and Tamura attribute-based descriptor [142]. The *database set* for this retrieval experiment is the whole test partition of the *ElBa* dataset (composed of ∼3000 images). To simulate the real challenging conditions, we generated 6 variants of each image, down-sampling at one of 3 different resolutions (100x100, 200x200, 300x300) and up-sampling them back to the original image size (1024x1024). Then we apply one of the following distortions:

- impulsive noise with a pixel's probability of 0.2 over all the image;
- radial lighting effect, increasing the brightness on a random point on the image and gradually decreasing it more in each pixel the farther from the chosen point it is.

Some examples of these images are shown in Fig. 3.23. It can be seen that distorted images simulate pictures that could be captured by users wishing to employ a retrieval application. The lighting effect simulates the flash of a camera while impulsive noise simulates general defects in the image acquisition process.

We consider each of the 6 variants as *query set* and we test each one separately. Given a distorted image from the query set, the task is to retrieve the corresponding original one from the database set. The position of the correct match in the computed ranking is recorded. This process is repeated for every image in a query set.

To distance functions used for ranking is chosen according to the descriptor; for each descriptor we selected the best performing distance function between all of the ones available in the MATLAB software [100]. More specifically, for the FV-CNN descriptor and our descriptor we employ the cosine distance while for the Tamura descriptor the cityblock distance function performs best.

Distortions	Tamura [142]	FV-CNN [27]	Texel-Att
Down-Sampling (100x100) and Impulsive Noise (p=0.2)	0.1380	0.3304	**0.6618**
Down-Sampling (200x200) and Impulsive Noise (p=0.2)	0.2103	0.4811	**0.8011**
Down-Sampling (300x300) and Impulsive Noise (p=0.2)	0.2284	0.5640	**0.8560**
Down-Sampling (100x100) and Radial Lighting Effect	0.1611	0.4394	**0.6356**
Down-Sampling (200x200) and Radial Lighting Effect	0.1728	0.8001	**0.8746**
Down-Sampling (300x300) and Radial Lighting Effect	0.2708	0.8855	**0.9376**

Table 3.5: *AUC (Area Under Curve)* for each distortion variant. Texel-Att performs better on every one of them. The related CMC are shown in Fig. 3.22.

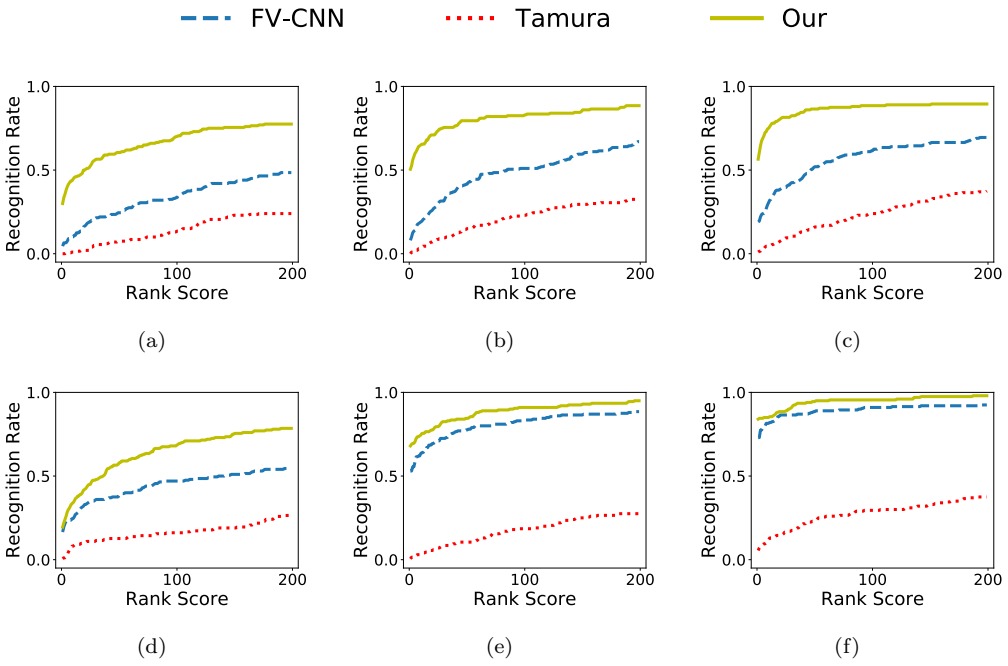

Figure 3.22: *CMC curves* on the retrieval experiments. Different plot for different variants of distortion: (a) 100x100 down-sampling and impulsive noise (b) 200x200 down-sampling and impulsive noise (c) 300x300 down-sampling and impulsive noise (d) 100x100 down-sampling and radial lighting effect. (e) 200x200 down-sampling and radial lighting effect. (f) 300x300 down-sampling and radial lighting effect. On the x axis the rank score (first 200 positions). On the y axis the recognition rate.

Table 3.5 shows the results of this experiment in all of the 6 variants previously described. In each case Texel-Att reaches the best results in terms of *AUC: Area Under Curve* index related to CMC (Cumulative Matching Characteristics) curves shown in the plots in Fig 3.22. We show only the first 200 positions for the CMC curve rank as we consider higher ranking positions less useful for a retrieval application (a user will rarely check results beyond 200 images).

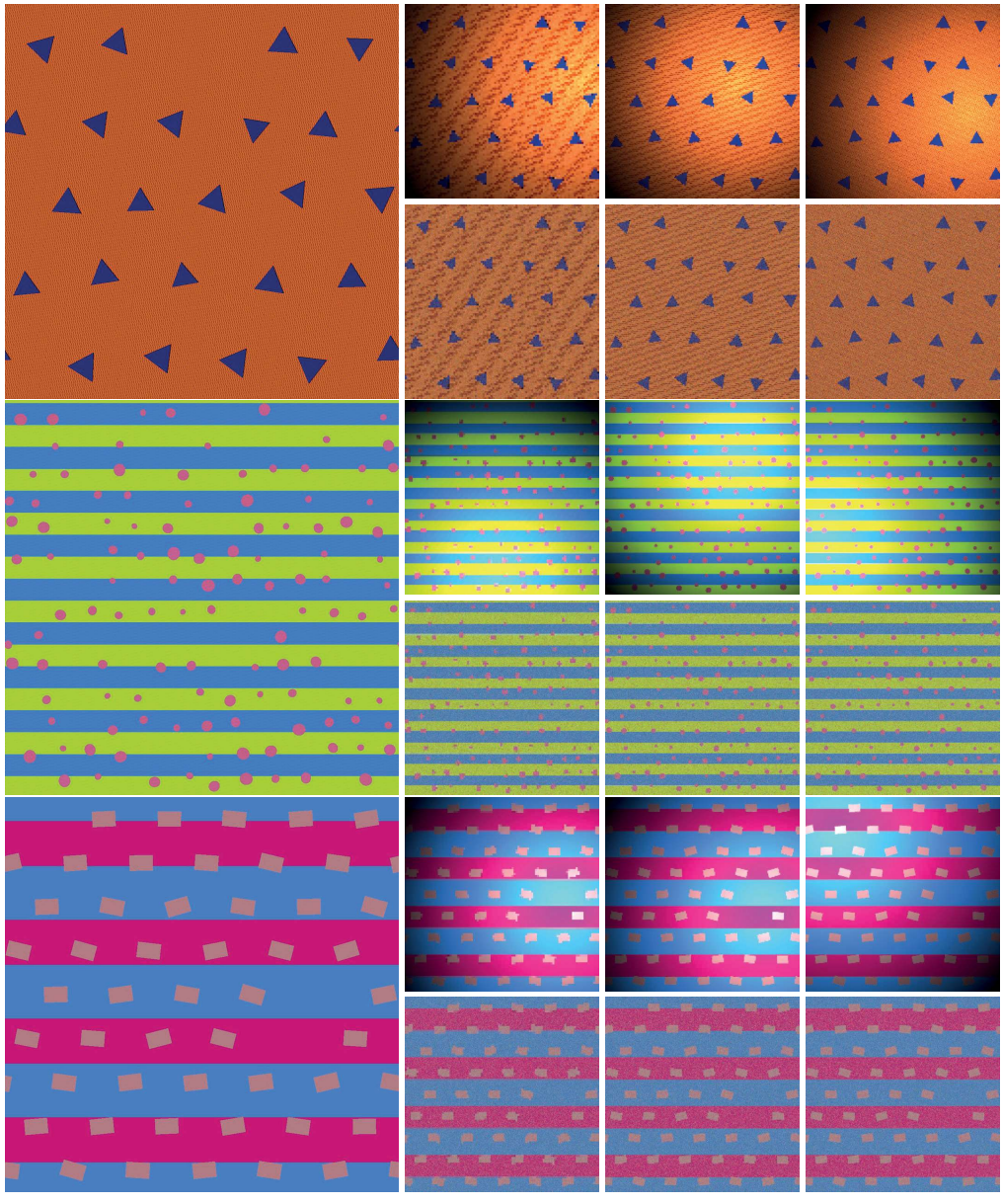

Figure 3.23: Three examples of distortions. For each one the biggest image is the original pattern. On the right, the first row depicts the radial lighting effect while the second one the impulsive noise distortion. The column are organized from the 100x100 down-sampling to 300x300 down-sampling.

Chapter 4

Video-To-Shop Retrieval

In this chapter, we present MovingFashion and SEAM Match-RCNN, respectively a novel dataset and a new framework for the video-to-shop problem: find the clothes worn in a video within a gallery of shop images. It is worth notice that the aim of this task is to automatize the sequential search process made by a human, in finding the garment that they like, worn by influencer in an Instagram story. It is a striking example of what the Computational Aesthetic want to deal with.

MovingFashion dataset contains 14866 videos and their corresponding shop images, partitioned in two different setups, *Regular* and *Hard*. The first is collected from the e-commerce website Net-A-Porter, taken under a controlled scenario. The latter is built with videos in the wild, taken from social networks such as Instagram and TikTok. Compare with Regular partition, it represents a harder scenario, with occlusions, different lightning, and other challenging conditions.

SElf Attention Multi-frame Match-RCNN (SEAM Match-RCNN) exploits a feature tracking and aggregation mechanism obtained by a non-local block temporal self-attention mechanism, mounted on the top of the Match-RNN architecture. The architecture is trained by domain adaptation, avoiding the need for tons of bounding-box annotations.

We compare SEAM Match-RCNN on MovingFashion and on the multi-frame street data excerpt of DeepFashion2, against multiple baselines and state-of-the-art techniques, achieving the new state-of-the-art on all the two datasets. On average, given a TikTok video from the Hard partition, we are able to individuate the correct garment within the first 5 retrieved items in a 1300+ elements gallery with an accuracy of 80%, representing an important proof of concept for a potential industrial application in e-fashion.

4.1 MovingFashion dataset

MovingFashion has 5.854M annotated frames, organized into 15045 video-shop *matching pairs*, i.e., each video is associated with a distinct *shop image*. In particular, there are 14.8K *unique* videos, among which some sequences (190 videos) have more than one

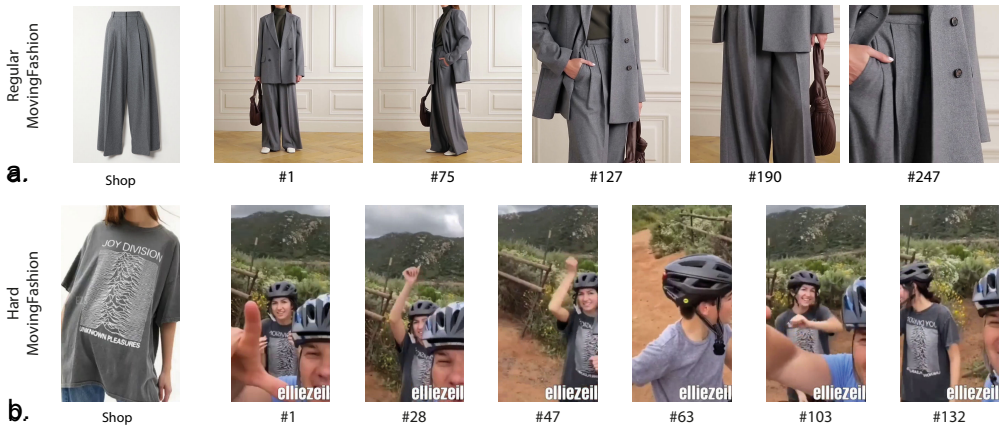

Figure 4.1: MovingFashion dataset samples. The top row contains a "Regular" sequence, the bottom row a "Hard" sequence.

associated shop item (*e.g.*, a t-shirt and trousers). The length of the videos is detailed in Fig. 4.2b, while the frame rate amounts to about 30FPS. Shop items are divided in classes, following the DeepFashion2 [42] taxonomy. The list of classes and the number of occurrences for each class in the dataset is reported in Fig. 4.2a.

4.1.1 Data sources

MovingFashion is formed by two subsets: *Regular* and *Hard*.

Regular MovingFashion: Regular MovingFashion consists of 10132 videos downloaded from the e-commerce website Net-A-Porter [1]: in the street video a single person is wearing the shop item in an indoor scenario (which can vary), and the corresponding shop image consists in the shop item captured over a plain background. This is the canonical shop image we have used in our experiments. Additionally, we have collected: a *front* shop image captured in the same background of the street video and worn by the same model in a frontal pose; a *rear* view image and a detail of the *fabric*. These last three were not used in the experiments. An example of Regular MovingFashion is showed in Fig. 4.1a. All of the videos in Net-A-Porter have been designed to promote a clothing item, which made the data collection process simpler. Cleaning was necessary only to remove classes not compliant with the taxonomy of DeepFashion2 [42], in particular *shoes* (deserving of a specific fashion taxonomy) and *jewelry* (due to the lack of a shared and widely accepted aesthetical taxonomy). For the remaining classes, the association to the specific DF2 taxonomy was direct.

Hard MovingFashion: Hard MovingFashion consists of 4723 videos from the social platforms Instagram and TikTok. In this case, shop images have been obtained either

[1] `https://www.net-a-porter.com`

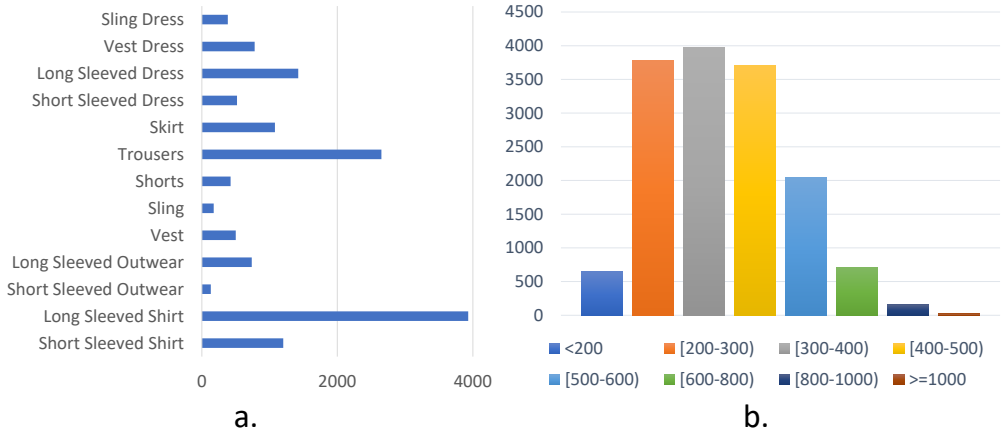

Figure 4.2: MovingFashion statistics; a) Cardinality of each clothing item class; b) Histogram of the number of frames for the street sequences.

by downloading images associated to the video as multiple images of the Instagram post or as part of the video itself (the spatial layout of some raw videos was organized in two halves, one being a still picture of the "shop" item, the other with the "street" video). Hard MovingFashion represents the hardest challenge, since all of the critical conditions listed in the introduction are present here, as also visible in Fig. 4.1b.

Instagram and TikTok videos required a lot of work, starting with the search for the street videos and their shop counterparts using the available API, up to the careful scraping of hashtags and profiles. In order to to download the data, the Instaloader[2] tool was employed. We manually selected a list of hashtags and profiles with a lot of content, i.e. a lot of videos paired with fashion products for sale. Through the use of the tool, we downloaded posts containing videos only based on the previously mentioned hashtags and profiles. The layout of these videos was standard for the vast majority of them: the frame was divided vertically in two parts, one with just a still picture of the shop product and one with the video itself.

We manually annotated these videos by following these steps:

- We checked that the product actually appears in the video, since in some cases the item never appears or appears very briefly in the frame; sometimes the item is in a different color than the one in the shop image.
- We drew a bounding box around the area of the shop item(s), taking care to include as few other items as possible.
- We drew another bounding box around the area of the video.

Using these annotations we crop the street videos and shop images. This results in pairings, where in some cases we have more than one shop item associated with a street video.

[2] `https://instaloader.github.io/`

Next, we dealt with duplicates of shop products. In some cases the same product is showcased in multiple videos by different users, but fortunately, the shop image used in such videos is the same. We leveraged this fact to perform a duplicate search for all the shop images. Products that were found to be duplicates were merged, creating pairings where for one shop product multiple videos are associated. To perform this search, for each product we searched for duplicates using a pre-existing tool[3] that employs Perceptual Hash. However we found out that in order to have a very high recall, this process also includes a lot of false positives. To perform a more thorough search, we tried an Image Registration technique using the RANSAC algorithm between each shop image and the duplicate candidates found using the tool. We tried to estimate a Similarity Transform, to account for translations and scaling (as is the case for these images). We then put a threshold on average pixel difference to separate between duplicates and non duplicates. Since no Python libraries that implement RANSAC are available, it was performed using a custom script.

To make sure that MovingFashion respects the privacy of social media users, we have rendered any face in the videos blurred using a publicly available, face blurring tool[4].

After collection and preprocessing, we split the data into a training and a testing partition, taking care of applying the same split for each single class. We perform a 90/10 train/test split for each partition before merging. Bounding boxes are extracted using a clothing detector. We then utilize the training data to train our SEAM Match-RCNN following the unsupervised procedure shown in Sec. 4.2.4. Since video sequences contain more than one item, to evaluate SEAM Match-RCNN and all the comparative approaches we create a tracklet containing the correct item for each street video sequence. In order to create them:

- Our SEAM Match-RCNN is trained on the data *using only video-image pairing annotations*. This results in a model where the Single-frame Matching Head can be effectively used for precisely tracking each item.
- We use the trained model to build a set of tracklets for each video.
- We manually go over each video and select the tracklets that contain the paired shop item, merging them if they are disjointed (this happens when an item is occluded completely or disappears from the frame and two separate tracklets are built).

The resulting tracklets are then saved. While for our approach, no tracklet annotations are used during training, they are used for all the comparative approaches. They are considered as equivalent to ours (the detector and the tracker are the same). It can be argued that they are actually better than ours as they are produced after the last epoch of training, while for our approach we start with a tracker that has not been trained yet. For the Person Re-ID approaches, the annotations are used to crop out

[3] `https://github.com/umbertogriffo/fast-near-duplicate-image-search`
[4] `https://github.com/ORB-HD/deface`

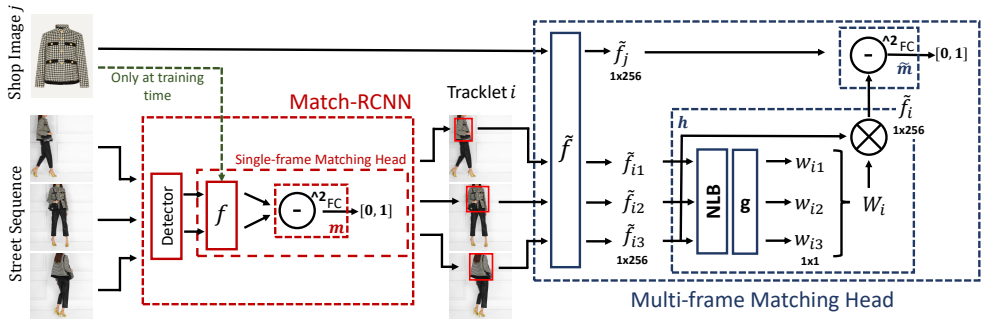

Figure 4.3: Architecture of our SEAM Match-RCNN system. Images are first processed by the Match-RCNN to extract bounding boxes and convolutional features. After tracking a clothing item across frames, its features are further processed by the Multi-frame Matching Head producing a final matching score between the street video sequence and the shop image.

part of the image according to the extracted bounding box. All of the comparative approaches shown in Sec. 4.3 use these *ground truth tracklets* for training and testing.

4.2 SEAM Match-RCNN Framework

SEAM Match-RCNN takes as input a sequence of street images $i_1 ... i_N$, and compares it with the gallery of K shop images providing a list of matching scores as output. Once the model has learned, the retrieval happens by means of three procedures:

1. *Tracklet creation*;
2. *Feature aggregation*;
3. *Video-to-shop matching.*

Going through these steps will allow us to present the structure of the network, detailed in Fig. 4.3.

4.2.1 Tracklet creation

On the input video sequence we need to locate a set of consecutive detections which refer to the same object, dubbed here *tracklet*. Since multiple objects might be on the video, multiple tracklets are expected. The module that deals with this is the **Match-RCNN**, which is composed of three functions:

1. A clothing detector which provides convolutional features $c_{i,t,k}$ with i indicating the i-th tracklet, t indicating the frame, k the k-th detection in that frame;
2. A 256-d feature extractor $f_{i,t,k} = f(c_{i,t,k}) \in \mathbb{R}^{256}$ which performs embedding of the convolutional features;

3. A matching score function $m(f_{i,t,k}, f_{i,t',k'}) \in [0,1]$, comparing different embeddings.

f and m together form the **Single-frame Matching Head**.

The tracklet extraction procedure is performed in an iterative fashion, following a two-step process:

1. Determining the *pivot* bounding box: This represents the most confident detection $f_{i,t_{best},k_{best}}$ in the sequence and acts as the central reference based on which the tracklet will be built.
2. Performing *propagation* based on the *pivot*: By comparing the embedding of the pivot $f_{i,t_{best},k_{best}}$ with all of the detections in every frame, the tracklet i can be built. In particular, a detection joins the tracklet if its matching score (matching function m of the Single-Frame Matching Head) is above a certain threshold, to avoid considering frames where the item is not visible.

Once the tracklet i is built, its detections are removed, and another tracklet focusing on a different item can be built.

4.2.2 Feature aggregation

The next step is aggregating the information of a tracklet and condensing it into a single multi-frame descriptor. The module that deals with the feature aggregation procedure is the **Multi-frame Matching Head** and it is composed of the following functions and modules:

1. A 256-d feature extractor $\tilde{f}_{i,t} = \tilde{f}(c_{i,t}) \in \mathbb{R}^{256}$ operating on the bounding box at time t of the tracklet i, i.e., $c_{i,t}$.
2. A non-local block [152] module which applies self-attention, enriching $\{\tilde{f}_{i,t}\}_t$ with information coming from all the other bounding boxes related to the object tracklet i.
3. An attention module $g\colon \mathbb{R}^{N\times 256} \mapsto \mathbb{R}^N$ that for each detection in a tracklet computes an importance score w_t such that $\sum_t w_t = 1$.
4. An aggregation module, which fuses $\{\tilde{f}_{i,t}\}_t$ into a joint descriptor $\tilde{f}_i$ by a weighted average over the attention scores $\{w_t\}$: $h(x) = g(NLB(x)) \cdot x$, $x \in \mathbb{R}^{N\times 256}$.
5. A matching score function $\tilde{m}(\tilde{f}_j, \tilde{f}_i) \in [0,1]$, which compares the aggregated descriptor of street sequence i ($h(\{\tilde{f}_{i,t}\}_t)$ as $\tilde{f}_i$) with the the shop descriptor of image j ($\tilde{f}_j$).

The aggregation procedure starts with the feature extractor $\tilde{f}$, which creates the initial descriptors for each box in a sequence. Then, self-attention is computed by the non-local block module and afterwards the attention module calculates the attention weights for each descriptor. The aggregation module puts all of the above together, producing the single multi-frame descriptor $\tilde{f}_i$. Note that we discard temporal continuity *by design*. Social network videos usually have dramatic zooms, very fast pose dynamics

and occlusions; moreover, elaborated videos may have shot changes which can fragment temporal continuity.

4.2.3 Video-to-shop matching

Following the feature aggregation procedure described above, we obtain a single multi-frame descriptor $\tilde{f}_i$ of the street tracklet i. In this final procedure, the matching score function $\tilde{m}$ of the Multi-frame Matching Head is used to match the aggregated multi-frame description with the single [5] shop descriptor of image j, $\tilde{f}_j$ (which can be considered as a tracklet composed by a single frame), under the assumption that a single item is portrayed in the shop image. We use the matching function $\tilde{m}$ on all the images in the shop gallery, producing in this way a list of matching scores between the street tracklet and all the images in the shop gallery, sorted in descending order, creating thus a *ranking*.

4.2.4 Model Training

To avoid the need of bounding boxes annotations, a time-consuming procedure especially for videos, SEAM Match-RCNN is trained by domain adaptation, through two phases: pretraining on the source image domain and training on the target video domain.

Pretraining on Source domain. The Match-RCNN part of SEAM Match-RCNN (detector and Single-frame Matching Head) is pretrained on an image street-to-shop dataset (e.g. DeepFashion2).

The purpose of this phase is to train a model that is able to estimate bounding boxes and matching scores in the target domain (even with noisy predictions due to the domain gap). Such predictions are used to generate tracklets and pseudo-labels to train the Multi-frame Matching Head.

Training on Target domain. The training procedure estimates the parameters for the Multi-frame Matching Head of the SEAM Match-RCNN, whose structure is detailed in Sec. 4.2.2, and fine-tunes the Single-frame Matching Head, while the detector's weights are frozen. The weights of the features extractor $\tilde{f}$ and matching score function $\tilde{m}$ are initialized copying those of f and m from the pretrained Single-frame Matching Head. Conversely, the attention modules of h are randomly initialized. During training, image and street video sequence pairs (thanks to the MovingFashion pairing annotations) are sampled, which are leveraged in the tracking procedure (Sec. 4.2.1): the pivot selection is done by selecting the detection that matches the shop product the most in the whole video if the matching score inferred from the matching function m of the Single-frame Matching Head is over a certain threshold. The propagation step remains the same as in Sec. 4.2.1.

[5] In principle, this model can be easily extended to deal with multiple frames in the case of a "shop" sequence. We did not consider this variant though, because usually clothing items are represented by a single image in e-commerce.

With this *training tracking procedure* a tracklet is built such that, with a certain confidence, it contains the correct shop item due to the pivot selection starting from the ground truth shop image. This is considered as a positive match during training (i.e. we set 1 as a pseudo-label for the tracklet). For what concerns the Single-frame Matching Head fine-tuning, each detection that composes the tracklet is considered as positive match as well.

The tracklet is then passed as input to the Multi-frame Matching head, which computes a singular multi-frame descriptor $\tilde{f}_i$ thanks to the aggregation procedure described in Sec. 4.2.2. In the end, this singular multi-frame descriptor $\tilde{f}_i$ is compared with the corresponding shop descriptor $\tilde{f}_j$ (obtained by using the feature extractor $\tilde{f}_j = f(c_j)$) utilizing the matching score function $\tilde{m}$. This produces a matching score in the range [0, 1].

Training is done by Stochastic Gradient Descent using Cross-Entropy loss for the classification of street videos and shop images as positive/negative matches. Positive pairings are built using the aforementioned procedure. All of the other combinations between tracklets and shop image descriptors are considered negative pairings (i.e. pseudo-label of 0) for the Single-frame Matching Head and the Multi-frame Matching Head.

We train the model on a single NVIDIA Titan RTX for 50 epochs. The total time of training is about 40 hours. On the same GPU, inference for a single image takes 50ms. For a sequence of 10 frames it takes about a second to compute detections, build tracklets and compute aggregated descriptors.

4.3 Video-To-Shop Experiments

For the retrieval performance evaluation, we follow the testing protocol of DeepFashion2 [42] for evaluating a street image probe against a shop image gallery, with some modifications in order to cope with videos. In DeepFashion2, a street image generates multiple detections: each *street* detection can generate a *proper* matching with some shop image, if it overlaps by a threshold with the corresponding ground truth street bounding box and if its item class is correct, otherwise the matching score is 0.

On MovingFashion, we compute detections on every street image and we build tracklets using the *tracking procedure* detailed in Sec. 4.2.1. Then, we compute the average IoU between each street tracklet and the *ground truth tracklet*. The one with the highest average IoU is chosen and used as a query.

In order to guarantee fairness in experiments, all baselines and comparative methods have been pretrained on two different street-to-shop datasets: DeepFashion2 and Exact Street2Shop [48]; the former has 873K probe-gallery pairs, while the latter 39K pairs only. Detailed results are reported for the first case, since performances were higher, while in the second case we show the main retrieval results, where our SEAM Match-RCNN remains the best performing approach.

Method	MovingFashion				Regular-MovingFashion				Hard-MovingFashion			
	T-1	T-5	T-10	T-20	T-1	T-5	T-10	T-20	T-1	T-5	T-10	T-20
Max Confidence [42]	0.29	0.59	0.72	0.83	0.31	0.63	0.76	0.86	0.21	0.46	0.60	0.71
Max Matching [23]	0.26	0.60	0.74	0.85	0.29	0.65	0.79	0.88	0.17	0.44	0.58	0.73
NVAN (2019) [80]	0.38	0.62	0.70	0.80	0.47	0.73	0.81	0.90	0.11	0.28	0.37	0.48
VKD (2020) [120]	0.40	0.49	0.58	0.65	0.49	0.59	0.68	0.75	0.13	0.20	0.27	0.34
AsymNet (2017) [23]	0.42	0.73	0.86	0.92	0.49	0.81	0.93	0.96	0.22	0.47	0.65	0.74
AsymNet [AVG]	0.39	0.66	0.83	0.90	0.46	0.78	0.90	0.96	0.19	0.44	0.62	0.73
AsymNet [MAX]	0.40	0.71	0.81	0.90	0.47	0.80	0.91	0.95	0.20	0.42	0.61	0.73
Average Match-RCNN [23]	0.39	0.73	0.84	0.91	0.43	0.79	0.88	0.94	0.24	0.56	0.70	0.81
SEAM Match-RCNN w/o NLB, g	0.37	0.73	0.86	0.93	0.42	0.78	0.90	0.95	0.21	0.57	0.75	0.85
SEAM Match-RCNN w/o NLB	0.41	0.73	0.83	0.91	0.47	0.79	0.89	0.95	0.21	0.54	0.66	0.79
SEAM Match-RCNN	**0.49**	**0.80**	**0.89**	**0.94**	**0.55**	**0.86**	**0.94**	**0.97**	**0.30**	**0.62**	**0.76**	**0.87**

Table 4.1: Video-to-Shop retrieval results on MovingFashion. Note: T-K means Top-K Accuracy.

Method	T-1	T-5	T-10	T-20
NVAN [80]	0.07	0.20	0.29	0.42
VKD [120]	0.16	0.24	0.31	0.38
MGH [164]	0.15	0.23	0.30	0.41
AsymNet [23]	0.09	0.26	0.37	0.49
SEAM Match-RCNN	**0.21**	**0.41**	**0.53**	**0.62**

Table 4.2: Top-K accuracy on MovingFashion, pretraining on S2S [48]

4.3.1 Experiments on MovingFashion

We compare our technique with three types of approaches:

Multi-frame baselines. They are extensions of single-frame techniques to multi-frame. The *Max Confidence* [42] keeps the most confident detection in a tracklet and uses it for Single-frame Matching, employing a Match-RCNN. The *Max Matching* is inspired from [23] and considers the max matching score between the tracklet's street frames and each shop image. These two baselines are actually working with a single image, which is selected by looking at the entire pool of frames in a tracklet. They are also useful to validate the performance boost that comes when using multiple frames instead of single ones.

The *Average Distance* is inspired by [23] and consists in averaging single-image matching scores of the tracklet street frames and each shop image. The *SEAM Match-RCNN w/o NLB,g* is obtained by averaging *descriptors* (and not matching scores) together by average pooling, removing in practice the NLB self-attention block and the attention scoring function g from the SEAM Match-RCNN (see the scheme in Fig. 4.3). Finally, *SEAM Match-RCNN w/o NLB* keeps the attention score, without the self-attention. These last three are proper multi-frame baselines, in the sense that they merge information coming from multiple frames.

Figure 4.4: Failure cases results of SEAM Match-RCNN for the MovingFashion dataset. On the left, we show 3 frames sampled from the 10 frames used for aggregation. On the right the shop images retrieved with the corresponding rank. The correct matches are with a green border.

Video Re-Identification approaches. Video Re-Id approaches share similarities with Video-to-shop, in that they look for the best way to aggregate multi-frame information to match a person in a disjoint multi-camera setting. In practice, we consider each shop clothing item the equivalent of a Person Re-Identification Identity. The main differences between video-to-shop and Person Re-ID are that in our case less information is available in terms of pixels, since face and hair need to be discarded, focusing only on the clothing.

Here we consider the SoA approaches of NVAN [80], VKD [120] and MGH [164][6].

Video-to-shop approaches. We consider the *AsymNet* [23] approach[7], and its modifications AsymNet[AVG] and AsymNet[MAX], in which the aggregations are made respectively by the average and the max of the similarity score nodes' outputs instead of using the fusion nodes binary tree. Asymnet exploits temporal continuity, yet it does not reach our results.

We set the sequence length to $T = 10$ for both training and testing, picking the frames using the Restricted Random Sampling strategy [78], thus ensuring coverage of the entire sequence length. To analyze variability in the results, we analyze the testing samples by sub-sampling them into pool of 800, 20 times, averaging the rankings.

Table 4.1 reports the results. Three facts become apparent:

1. As expected, single-frame approaches (Max Confidence, Max Matching) are definitely inferior (<15% on average) than multi-frame approaches;

2. The considered re-identification approaches, apart from top-1 scores, are inferior to genuine video-to-shop methods;

3. Our SEAM Match-RCNN surpasses all the competitors, including AsymNet, which gives a better aggregation than the AVG-distance in its [AVG] version and the MAX-distance in its [MAX] version.

[6] At the moment of writing, the MGH approach is state-of-the-art in the MARS Video Person Re-Identification dataset, followed closely by VKD and NVAN.

[7] The code is available at `https://github.com/kyusbok/Video2ShopExactMatching`.

Figure 4.5: Qualitative retrieval results of SEAM Match-RCNN for the MovingFashion dataset. On the left, we show 3 frames sampled from the 10 frames used for aggregation. On the right the shop images retrieved with the corresponding rank. The correct matches are with a green border, otherwise red.

By looking at the ablative versions of SEAM Match-RCNN, one can note that the self-attention gives the strongest performance boost, followed by the attention layer. Their cooperation, i.e., the complete SEAM Match-RCNN, reaches the highest score.

By looking at the results within the Regular and Hard MovingFashion partitions, it is quite easy to note the general decrease in performance when it comes to the hard partition. To understand the performance qualitatively, Fig. 4.5 shows retrieval results from Regular (Fig. 4.5a) and Hard (Fig. 4.5b). Actually, even if Regular is apparently harder due to many shop alternatives which differ by fine grained results (see the flared jeans), the dramatic changes of poses and backgrounds of the Hard partition play a stronger role.

Failure cases arise when the original video has discriminant parts of the clothing item covered for most of the sequence, for instance the logo of the light blue sweatshirt (Fig. 4.4a). In this case, self-attention overlooks such important details. Complex visual patterns like the hard-rock band logo (Fig. 4.4b), seem to be not well characterized, meaning that the best match is attributed considering the shape of the logo rather than its content (the "Metallica" logo has the same shape of the probe logo).

In Table 4.3, we show the results of Single-frame baselines built on top of the Match-RCNN (the main building block of our SEAM Match-RCNN). In particular, SFM-1qrt uses the frame at the first quartile of all the available frames of that sequence, SFM-median uses the median frame and so on. SFM stands for Single-frame match and is a short term for Match-RCNN.

The correspondent baselines are shown, adopting the Deep Kronecker-Product Matching (KPM) [127] and the Easy Positive Triplet Mining approach (EPHN) [159]. The rationale of this choice was to focus on Single-frame Re-Identification approaches and compare them to the Match-RCNN. This was done to enlarge the spectrum of possible comparative approaches, which have open-source code available. The idea of considering Re-ID approaches against street-to-shop techniques was also presented in the DPRNet

Method	MovingFashion				Regular-MovingFashion				Hard-MovingFashion			
	T-1	T-5	T-10	T-20	T-1	T-5	T-10	T-20	T-1	T-5	T-10	T-20
SFM-First	0.20	0.43	0.52	0.63	0.21	0.44	0.53	0.64	0.16	0.41	0.52	0.62
SFM-1qrt	0.25	0.53	0.66	0.77	0.29	0.58	0.71	0.82	0.15	0.37	0.51	0.63
SFM-Median	0.23	0.48	0.61	0.75	0.26	0.53	0.66	0.79	0.17	0.33	0.47	0.65
SFM-3qrt	0.21	0.47	0.60	0.72	0.24	0.53	0.66	0.77	0.13	0.29	0.42	0.57
SFM-Last	0.11	0.31	0.41	0.53	0.14	0.35	0.46	0.58	0.05	0.18	0.27	0.36
EPHN-First (2020) [159]	0.15	0.34	0.44	0.53	0.16	0.36	0.46	0.55	0.11	0.27	0.37	0.47
EPHN-1qrt	0.24	0.45	0.55	0.65	0.28	0.51	0.62	0.72	0.13	0.24	0.32	0.42
EPHN-Median	0.27	0.49	0.58	0.66	0.32	0.57	0.67	0.74	0.10	0.24	0.32	0.42
EPHN-3qrt	0.24	0.47	0.55	0.65	0.29	0.55	0.64	0.74	0.09	0.21	0.29	0.40
EPHN-Last	0.17	0.33	0.41	0.49	0.20	0.39	0.47	0.56	0.07	0.15	0.19	0.27
KPM-First (2019) [127]	0.19	0.40	0.51	0.61	0.22	0.45	0.56	0.67	0.09	0.26	0.33	0.45
KPM-1qrt	0.27	0.48	0.60	0.71	0.32	0.56	0.69	0.80	0.12	0.24	0.33	0.45
KPM-Median	0.24	0.48	0.59	0.69	0.27	0.55	0.67	0.78	0.12	0.25	0.35	0.43
KPM-3qrt	0.23	0.46	0.56	0.69	0.27	0.53	0.65	0.76	0.10	0.22	0.28	0.39
KPM-Last	0.16	0.35	0.45	0.55	0.20	0.41	0.53	0.65	0.05	0.14	0.19	0.23
SEAM Match-RCNN	**0.49**	**0.80**	**0.89**	**0.94**	**0.55**	**0.86**	**0.94**	**0.97**	**0.30**	**0.62**	**0.76**	**0.87**

Table 4.3: SEAM Match-RCNN retrieval results on MovingFashion compared with Single-frame approaches. Note: T-K means Top-K Accuracy.

Method	MovingFashion				Regular-MovingFashion				Hard-MovingFashion			
	T-1	T-5	T-10	T-20	T-1	T-5	T-10	T-20	T-1	T-5	T-10	T-20
Max Confidence	0.29	0.59	0.72	0.83	0.31	0.63	0.76	0.86	0.21	0.46	0.60	0.71
Max Matching	0.26	0.60	0.74	0.85	0.29	0.65	0.79	0.89	0.17	0.44	0.58	0.74
Average Match-RCNN [23]	0.39	0.73	0.84	0.91	0.43	0.79	0.88	0.94	0.24	0.56	0.70	0.81
Average Descriptor	0.37	0.72	0.86	0.93	0.42	0.78	0.90	0.95	0.21	0.57	0.75	0.85
EPHN-MaxConf (2020) [159]	0.22	0.43	0.55	0.65	0.26	0.50	0.61	0.71	0.10	0.22	0.34	0.44
EPHN-MaxMatching	0.35	0.59	0.67	0.74	0.42	0.68	0.76	0.81	0.14	0.32	0.41	0.52
EPHN-AvgMatching	0.31	0.55	0.64	0.73	0.37	0.64	0.73	0.81	0.11	0.28	0.37	0.48
EPHN-AvgDescriptor	0.22	0.43	0.52	0.61	0.26	0.49	0.58	0.68	0.10	0.24	0.33	0.43
KPM-MaxConf (2019) [127]	0.25	0.47	0.57	0.68	0.30	0.54	0.65	0.77	0.11	0.25	0.32	0.43
KPM-MaxMatching	0.30	0.54	0.66	0.75	0.36	0.61	0.73	0.82	0.13	0.32	0.42	0.53
KPM-AvgMatching	0.34	0.58	0.68	0.77	0.40	0.68	0.78	0.86	0.15	0.28	0.38	0.48
KPM-AvgDescriptor	0.34	0.58	0.69	0.77	0.40	0.68	0.78	0.86	0.15	0.28	0.38	0.48
SEAM Match-RCNN	**0.49**	**0.80**	**0.89**	**0.94**	**0.55**	**0.86**	**0.94**	**0.97**	**0.30**	**0.62**	**0.76**	**0.87**

Table 4.4: SEAM Match-RCNN retrieval results on MovingFashion compared with Multi-frame approaches. Note: T-K means Top-K Accuracy.

Categories	**NVAN** [80]	**VKD** [120]	**MGH** [164]	**AsymNet** [23]	**SEAM M-RCNN**
Short Sleeve Shirt	**0.46**	0.20	0.39	0.35	0.43
Long Sleeve Shirt	0.38	0.44	0.41	0.45	**0.44**
Short Sleeve Outwear	0.34	0.23	0.33	0.35	**0.42**
Long Sleeve Outwear	0.40	0.43	0.42	0.36	**0.46**
Vest	**0.42**	0.10	0.24	0.27	0.31
Sling	0.30	0.16	0.33	0.32	**0.36**
Shorts	0.19	0.27	0.22	0.25	**0.39**
Trousers	0.37	0.28	0.35	**0.45**	0.39
Skirt	0.40	0.52	0.47	0.39	**0.56**
Short Sleeve Dress	0.34	0.54	0.35	0.45	**0.73**
Long Sleeve Dress	0.37	0.63	0.36	0.57	**0.68**
Vest Dress	0.39	0.49	0.37	0.42	**0.64**
Sling Dress	0.42	0.39	.42	.32	**0.69**
All Classes	0.38	0.40	0.40	0.42	**0.49**

Table 4.5: Top-1 retrieval accuracy on MovingFashion for the 14 different item classes.

paper [167].

The inferiority of these baselines with respect of the Multi-frame of Table 4.1, and in particular with SEAM Match-RCNN, is evident and fully understandable.

Notably, in almost all of the MovingFashion partitions (apart the regular one with EPHN), the ·-1qrt baseline gives the higher results, which seems to be in accord with the best practices in social media video editing, that is, that videos have to deliver their main message within approximately 6 seconds [44].

As additional Multi-frame approaches, Table 4.4 shows Max Confidence, Max Matching and Average Matching scores when considering the KPM [127] and the EPHN [159] as Single-frame method ingredients, in the same way that Match-RCNN was used to calculate Max Confidence, Max Matching and Average Matching from Table 4.1.

Even in this case, SEAM Match-RCNN gives the best performance, showing an overall superiority of Match-RCNN as a Single-frame tool to aggregate visual clothing information.

The results w.r.t the single clothing classes of MovingFashion are reported in Table 4.5, where it is possible to observe our advantage in all but three classes. Interestingly, we found that the simpler the clothing in terms of texture, the lower the retrieval performance. This is reasonable, since texture adds discriminative details, and this is why classes with simpler texture like vest, sling, shorts and trousers performed worse. We computed textureness by gray-level co-occurrence matrix contrast; quantitatively speaking, textureness and top-1 accuracy in retrieval are found to be correlated (Spearman $\phi = 0.72$, $p - value \leq 0.05$).

Another experiment regards the length of the sequences. Fig. 4.6 reports, with the associated error bars, the performance of SEAM Match-RCNN when increasing the number of frames from 1 to 20. As expected, the curves for both partitions, at both the top-1 and top-20 are increasing, with the "Hard" partition showing a plateau after 10 frames, while the "Regular" partition seem to benefit systematically. The reason could be that "Hard" sequences are dramatically noisy, and adding more frames will

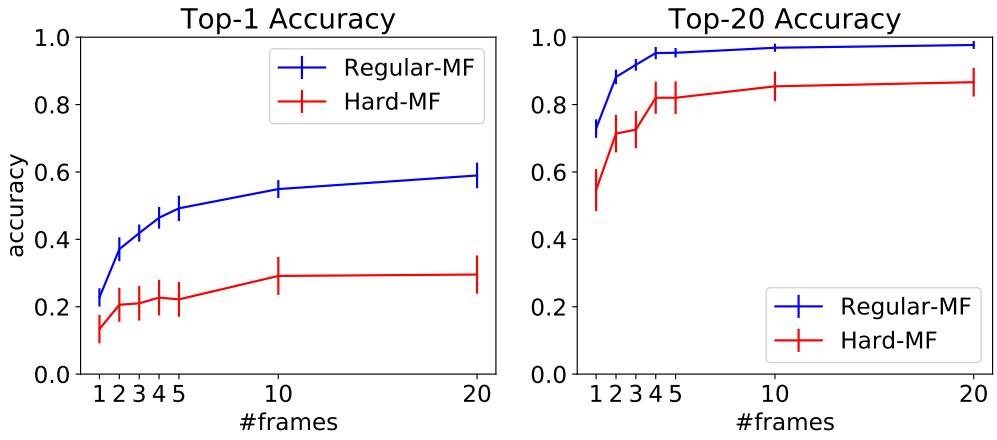

Figure 4.6: Plot of the SEAM Match-RCNN retrieval accuracy (y-axis) using different numbers of frames (x-axis) for aggregation. Error bars represent standard deviation of the accuracy.

Method	**5 Frames**	**10 Frames**	**20 Frames**
NVAN [80]	0.35	0.38	0.39
VKD [120]	0.36	0.40	0.43
MGH [164]	0.36	0.38	0.40
AsymNet [23]	0.37	0.42	0.44
SEAM Match-RCNN	**0.43**	**0.49**	**0.52**

Table 4.6: Top-1 accuracy on MovingFashion, with different number of frames.

augment the clutter we need to deal with, while the "Regular" ones benefit because of the fine grained details which characterize the partition. Comparative performances when varying the sequence's length against other approaches are in Tab. 4.6. Notably, Asymnet [23] does not reach our results *even when doubling the number of input frames.*

4.3.2 Experiments on unrelated sets of images

MovingFashion has street videos which depict clothing items in a variety of scenarios: indoor, outdoor, etc. We are interested in bringing this variety to the extreme, answering the following question: how does SEAM Match-RCNN behave when the street video sequence is formed by a few totally unrelated frames?

In order to perform these experiments, we build Multi-DeepFashion2 from DeepFashion2 using the pairings between shop images and street sequences composed of multiple corresponding street images (Fig. 4.7).

The total pairings amount to 11K, each one composed of an image sequence (6 frames on average) sampled from different sources, along with the corresponding shop image.

Results are in Tab. 4.7. Please note that, in order to be consistent with the 10-frames street sequence length we generate random repetitions for all the approaches given the

Figure 4.7: Three street images and their paired shop image taken from MultiDeepFashion2.

smaller set of diverse images. The numbers indicate a decrease in general performance (less distinctive frames, more shop items); even in this case, we perform better than AsymNet. The ground truth versions (where ground truth bounding boxes are used for training and testing instead of detections) give an upper bound both for SEAM Match-RCNN and Asymnet, providing a limited 2% boost on average. This bears witness to a consistent resistance to detection irregularities and confirms our superiority.

We also investigate other multi-frame policies (Max Confidence, Matching, Avg Matching and Descriptor), since single-frame policies do not have much sense, as the single-frames are not part of a single sequence. Even in this case, SEAM Match-RCNN is the best alternative (Table 4.7).

Method	**T-1**	**T-5**	**T-10**	**T-20**
Max-Confidence	0.19	0.44	0.54	0.66
Max Matching [23]	0.14	0.45	0.61	0.75
Average Match-RCNN [23]	0.22	0.49	0.63	0.74
Average Descriptor	0.20	0.48	0.60	0.71
NVAN (2019) [80]	0.22	0.37	0.43	0.49
VKD (2020) [120]	0.21	0.27	0.33	0.38
MGH (2020) [164]	0.22	0.34	0.39	0.45
EPHN-MaxConf (2020) [159]	0.11	0.19	0.24	0.29
EPHN-MaxMatching	0.11	0.21	0.26	0.33
EPHN-AvgMatching	0.16	0.29	0.34	0.41
EPHN-AvgDescriptor	0.12	0.22	0.27	0.33
KPM-MaxConf (2019) [127]	0.09	0.20	0.25	0.30
KPM-MaxMatching	0.08	0.16	0.21	0.28
KPM-AvgMatching	0.10	0.20	0.25	0.32
KPM-AvgDescriptor	0.13	0.25	0.33	0.40
AsymNet [GT] [23]	0.21	0.50	0.62	0.74
AsymNet (2017) [23]	0.18	0.43	0.57	0.70
AsymNet [AVG]	0.16	0.41	0.54	0.68
AsymNet [MAX]	0.15	0.42	0.56	0.70
Average Distance [23]	0.22	0.49	0.63	0.74
SEAM Match-RCNN w/o NLB, g	0.20	0.47	0.60	0.71
SEAM Match-RCNN [GT]	0.30	0.58	0.67	0.76
SEAM Match-RCNN	**0.28**	**0.54**	**0.66**	**0.76**

Table 4.7: Video-to-Shop retrieval results on MultiDeepFashion2. Note: T-K means Top-K Accuracy.

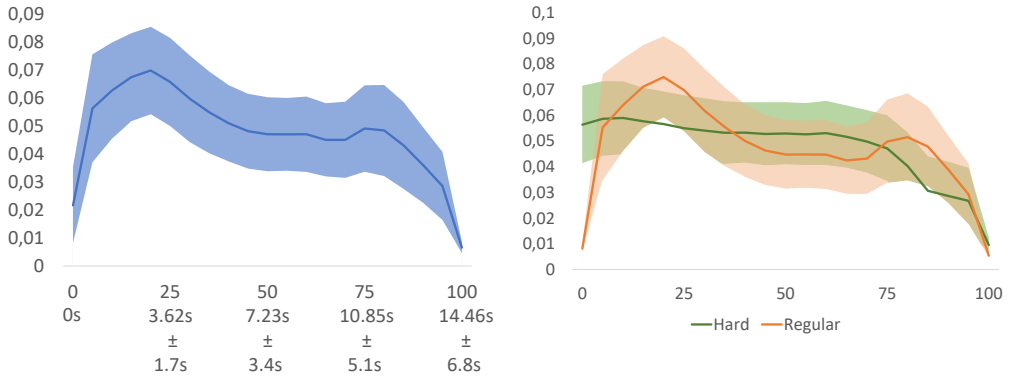

Figure 4.8: Mean attention score every 5 percentiles of the video length. For each video we sampled 21 equally spaced frames. On the left we report the average attention (y-axis) and frame-timing information (x-axis labels) for the whole MovingFashion dataset. On the right for the Regular and Hard subsets. We show error bands for the standard deviation.

4.3.3 Experiments on the attention mechanism

The ablation studies of Table 4.1 clearly show that the attention layers play a crucial role for the SEAM Match-RCNN performance. Here we explain their role qualitatively and quantitatively. In Fig. 4.9 we report the attention values obtained after the application of the attention layer g to the output of the self-attention layer NLB of Sec. 4.2.2, i.e., $g(NLB(x))$. On row a), one can note that the attention is high when the heart logo is visible (0.31, 0.23 in the first two frames) and it goes down when it vanishes, despite the light blue shirt (last frame) being very similar area-wise. This means that the mechanism considers the heart logo as important for retrieval. On the second row b), the effect of an occlusion in the attention score (last frame). On the third row c), a white top with a logo gives a stable attention score (around 0.28). We manually cover the logo in the third frame, causing a clear decrease in the attention, uniformly increasing the ones highlighting the logo.

Finally, driven by best practices in social video editing [44], which state that a video message has to deliver its main content in the first 6 seconds to trigger the observers' attention, we calculate the attention every 5 percentiles on all the MovingFashion sequences, producing the curves in Fig. 4.8a) (on the whole MovingFashion dataset) and on the separate partitions Fig. 4.8b. Surprisingly, the data confirms this rule, showing a clear (Fig. 4.8a) peak around the first quartile (definitely within 6 seconds), then a decrease and a later increase with a local maximum on the fourth quartile. The same holds for the two separate partitions (Fig. 4.8b)), with less emphasis on the "Hard partition". The reason lies in the nature of the Net-A-Porter videos, which in many cases show the entire clothing item in the beginning of the sequence, with the model that moves subsequently, zooming up to critical detail (the belt for the shorts) towards the end (second peak). On the "Hard" partition, the attention for the clothing items is higher in the beginning, since the actors present their outfit and then exhibit their

Figure 4.9: Qualitative observations on the attention behaviour. On the left, for each video sequence we show the detection bounding boxes and the computed attention score. On the right the paired shop item.

performance (dancing, gymnastics etc.), concluding in both the cases with uninteresting details clothing wise.

4.3.4 Qualitative Results

As additional *qualitative* results, on Fig. 4.10 results of SEAM Match-RCNN for the Hard-MovingFashion dataset are shown. Two types of considerations can be drawn: the first one is the variability of the videos, which here can be appreciated with more examples. Second, the retrieval results on the right display that SEAM Match-RCNN is capable of finding similar images, among a shop gallery that in some cases contains highly similar items (see for example the light gray trousers).

On Fig. 4.11 results of SEAM Match-RCNN for the Regular-MovingFashion dataset are shown. Here, on street frames which exhibit more regularities, the shop items are vice versa more insidious than the TikTok ones, since they exhibit a lower variability, see for example the black female dresses of row 6. The same rationale holds for the white shirts and the black paints. Finally, on Fig. 4.12 retrieval results on MultiDeepFashion2 are shown. Looking at the retrieval results, one can notice that shop items are way less regular/neutral than the ones on the MovingFashion (which anyway represent a more genuine excerpt of an e-commerce website): at the same time, street frames are often zoomed captures of the object of interest, in general offering a retrieval challenge different than the one on MovingFashion. The strong results obtained by SEAM Match-RCNN prove its versatility in working on a broader set of scenarios.

Figure 4.10: Qualitative retrieval results of SEAM Match-RCNN for the Hard-MovingFashion dataset. On the left, we show 3 frames sampled from the 10 frames used for aggregation. On the right the shop images retrieved starting from the closest match (left). The correct matches are represented with a green border.

Figure 4.11: Qualitative retrieval results of SEAM Match-RCNN for the Regular-MovingFashion dataset. On the left, we show 3 frames sampled from the 10 frames used for aggregation. On the right the shop images retrieved starting from the closest match (left). The correct matches are represented with a green border.

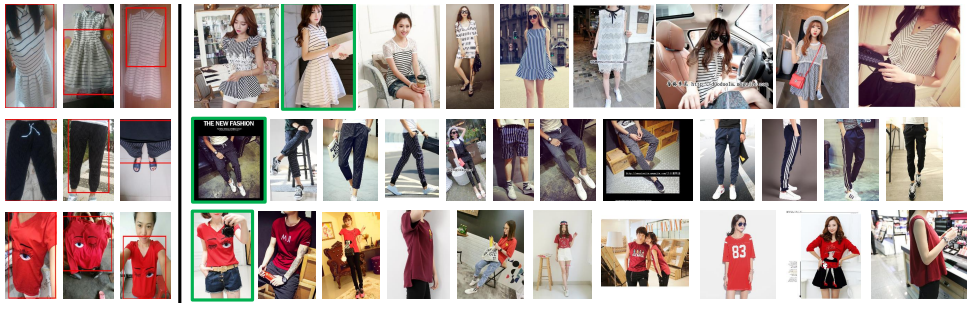

Figure 4.12: Qualitative retrieval results of SEAM Match-RCNN for the MultiDeepFashion2 dataset. On the left, we show 3 frames sampled from the 10 frames used for aggregation. On the right the shop images retrieved starting from the closest match (left). The correct matches are represented with a green border.

Chapter 5

New Fashion Product Performance Forecasting

In this chapter, we tackle a challenge that has started attracting attention in computer vision and machine learning: New Fashion Product Performance Forecasting (NFPPF). This challenge aims at predicting the success of a brand-new clothing probe with no available past observations. The success is assessed through various performance indices such as the number of sales or popularity. To tackle this issue, we first introduce VISUELLE, the first publicly available dataset for the task of new fashion product sales forecasting, containing the sales of 5577 new products sold between 2016-2019, derived from genuine historical data of Nunalie, an Italian fast-fashion company. Our dataset is equipped with images of products, metadata and related sales. By exploiting clothing attributes or image data, zero-shot learning is essentially applied, under the rationale that new products will perform comparably to aesthetically similar, older products. The second contribution is GTM-Transformer, whose encoder works on the representation of the exogenous time series, while the decoder forecasts the sales using the Google Trends encoding, and the available visual and metadata information. The model works in a non-autoregressive manner, avoiding the compounding effect of the first-step errors.

To fill the missing past observations we introduce two different exogenous signals. The former is Google Trends signals, from which the framework takes its name (Google Trends Multimodal Transformer), investigating the effectiveness of systematically querying the Google Trends API through textual translations of aesthetic aspects, generating exogenous knowledge. The latter embraces the saying *an image is worth a thousand words*, comparing the probe image of a new product directly with images uploaded on the web in the past and indexed from Google Images. This signal is built following a data-centric pipeline that starts expanding textual tags associated with a probe image so as to query fashionable or unfashionable images related to it, which have been uploaded on the web in the recent past. A binary classifier is robustly trained on these web images by confident learning, to capture what was fashionable at that time, and how much the probe image is conforming. Such compliance produces the POtential Perfor-

Figure 5.1: Examples of Images Per Category

mance (POP) time series. It is important to emphasize that, being able to distinguish fashionable VS unfashionable images, is a typical computational aesthetic task.

We compare GTM-Transformer on VISUELLE against different approaches, encapsulating both Google Trends and POP signals as exogenous time-series. These experiments prove that our framework is more accurate in terms of both percentage and absolute error, especially with the addition of exogenous knowledge that boosts the performance. In particular, POP ameliorates the sales curve prediction of all the state-of-the-art NFPPF models and is also predictive for the popularity of new styles (ensembles of clothing items) on the Fashion Forward benchmark.

5.1 VISUELLE dataset

VISUELLE describes the sales between October 2016 and December 2019 of 5577 products in 100 shops of Nunalie[1], an Italian fast-fashion company funded in 2003. The sales of 2020 and 2021 are also available, but ignored here because of the COVID pandemic. Only sales of 2021 is used in an additional experiments, to stress out the framework on an extremly difficult scenario. For each product, multimodal information is available, which will be detailed in the following subsections, giving more emphasis to sales data, Google Trends and POP signals.

[1] www.nunalie.it

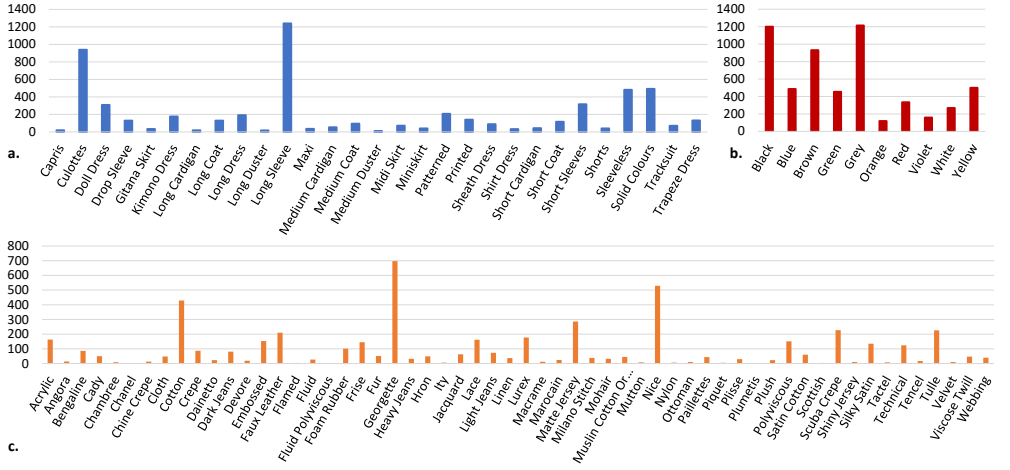

Figure 5.2: Cardinalities of the dataset per categories (a), color (b) and fabric (c)

5.1.1 Image data

Each product is associated with an RGB image, of resolution which varies from 256 to 1193 (width) and from 256 to 1172 (height) with median values 575 (w) 722 (h) . Images have been captured in a controlled environment, in order to avoid color inaccuracies and potential biases in the predictions [110]. Each image portrays the clothing item on a white background, with no person wearing it. Additionally, a binary foreground mask is provided.

5.1.2 Text data

Each product has multiple associated tags, which have been extracted with diverse procedures detailed in the following, and carefully validated by the Nunalie team.

The first tag is the *category*, taken from a vocabulary of 27 categories, visualized in Fig. 5.2a; the cardinality of the products shows large variability among categories overall, due to the fact that some categories (e.g. long sleeves) cost less and ensure higher earnings. The "color" tag represents the most dominant color, and is extracted from the images with a proprietary pixel clustering algorithm, keeping the color with the most belonging pixels, and validated for each product by two human operators that must agree on it. The final vocabulary is made of 10 elements. The cardinality per color is reported in Fig. 5.2b. The *fabric* tag describes the material from which clothes are made, and comes directly from the technical sheets of the fashion items. This tag comes from a vocabulary of 58 elements, visualized in Fig. 5.2c; A product is sold during a particular season, and within a season, released on the market at a precise day. This *temporal information* is recorded as a text string. Holidays and sales periods are supplementary information which we plan to deliver for a second version of the dataset.

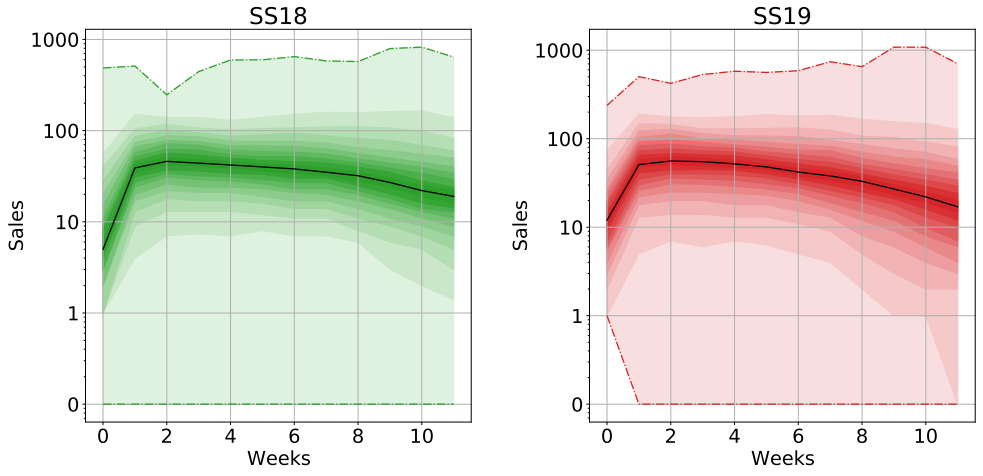

Figure 5.3: 25-percentile density plots of the SS18 and SS19 seasons.

5.1.3 Sales data

The sales time series have a weekly frequency and contain 12 observations each, which corresponds to the permanence of an item in the shops during a fashion season (Autumn-Winter, AW and Spring-Summer, SS). Fig. 5.3 contains a log-density plot of the sales of all the products, merging together different categories, across corresponding seasons (SS18 and SS19 were used for clarity). This is useful to show that there are general "mean curves" where the sales peak occurs after a week and that as the weeks go by, the sales are characterized by a higher variability. An increase of the sales during the years is visible, showing that the company seems to perform well. Notably, from the release moment until 6 weeks, no external action is done by the company owners (discounts, pre/sales, additional supplying) and they had never sold out products, so we can state that the signal variability is given by the product attractiveness.

5.2 Google Trends

Extracting Google Trends to discover the popularity of textual term describing visual data poses a paradox: the more specific the text, the least informative the signal (due to sparsity), and vice-versa. We collect, for each product, a Google Trends time-series for each of its three associated attributes: $color, category, fabric$. The trends are downloaded starting from the release date and going back 52 weeks, essentially anticipating the release of each single item by one year. Each signal gives percentages, reaching 1 (100%) in the moment in time when the particular attribute had the maximum search volume on Google, depending on the search interval.

Fig.5.4 contains examples of Google Trends in the interval 2016-2019. As visible, the nature of these signals is highly variable, spanning from highly structured to more noisy. To make the Google Trends signal more reliable, we follow the "multiple sampling"

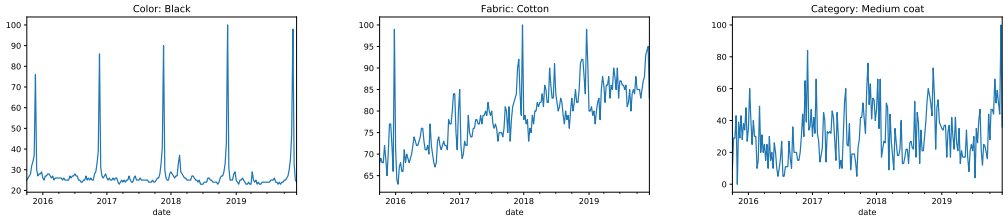

Figure 5.4: Examples of Google Trends time-series spanning multiple years.

strategy discussed in [102]. Google normalizes the search results of a query by the total searches of the location and time range chosen by the user. Then, the resulting numbers are scaled from 0 to 100, in order to represent the relative popularity. The problem is of course, because of the high amount of search queries that Google processes each day, the query results are always a sub-sample of the "true" ones and this sample may not always be the same. So to avoid sampling bias, we download each Google Trend 10 times and use the mean to create a more representative signal.

5.3 POP Signal

POP Signal is built following a completely novel pipeline. The input of the pipeline is the probe image $\mathbf{z}^{(t)}$, where $\mathbf{z}$ represents the new clothing item and t the *observation time*, which is the date from when we begin looking into the past. The output is the POP signal $S_{\mathbf{z}}^{(t)} = s_{\mathbf{z}}^{(t-K_{past})}, \ldots, s_{\mathbf{z}}^{(t-k)}, \ldots, s_{\mathbf{z}}^{(t-1)}$, defined for K_{past} time steps preceding t, where $k = 1, \ldots, K_{past}$ and $s_{\mathbf{z}}^{(t-k)} \in \mathbb{R}$. In this thesis, we describe the observation times in terms of weeks and set $K_{past} = 52$. This means we look one year before the observation time t, since market analysis for fashion products typically begins nearly a year in advance [138]. The next sections will sequentially detail the general pipeline of our approach, reported in Fig. 5.5.

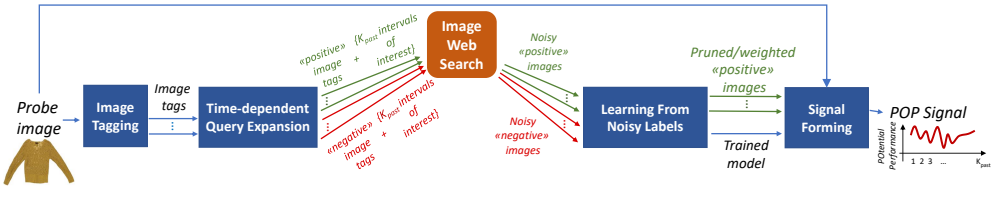

Figure 5.5: Schematic pipeline of our approach; we start with a probe image and obtain the POtential Performance (POP) signal at the end. Along this pipeline, we sequentially process information in different modalities, thereby creating a *cross-modal signal*.

5.3.1 Image Tagging

The first operation is the extraction of textual tags $\{a_{\mathbf{z}}^{(j)}\}_{j=1,\ldots,J}$ associated to $\mathbf{z}$. These tags should represent the clothing item with sufficient generality, capturing at least

categorical information (*e.g* "long sleeve") and a dominant color (*e.g* "yellow"). Empirically, we found these types of tags to work well with our approach. Category and color can be automatically extracted with high accuracy[87]; at the same time, these are tags usually contained in the technical data sheet accompanying the product which we exploit here, as discussed in Sec. 5.5.2, to avoid early errors that might compromise the downstream pipeline.

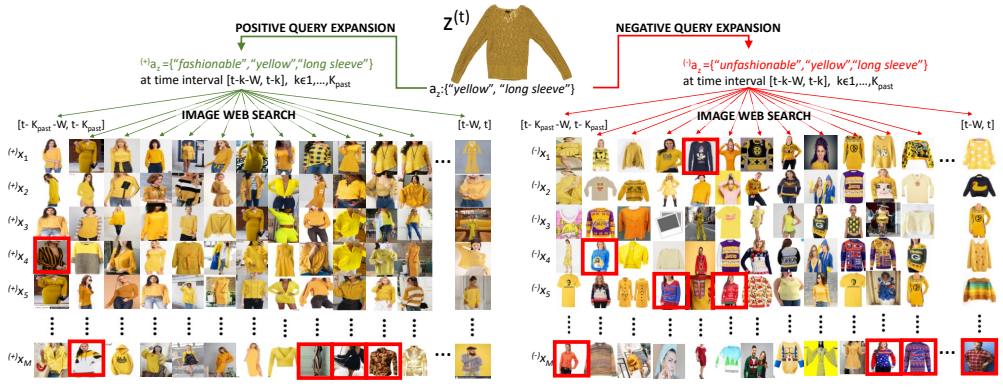

Figure 5.6: Pipeline insights on *Time-dependent Query Expansion* (Sec. 5.3.2), *Image Web Search* (Sec. 5.3.3) and *Learning From Noisy Labels* (Sec. 5.3.4) steps. This figure reports a real world excerpt of the download and processing of N=2600 images ($N = 2(M \times K_{past})$, $M = 25$, K_{past}=52).

5.3.2 Time-dependent Query Expansion

The second operation (detailed in Fig. 5.6 on a real example) performs two different textual query expansions, generating *positive expansions*, $\{a_{\mathbf{z}}^{(j)}\}_{j=1,...,J} \cup J^{(+)}$ where the additional $J^{(+)}$ tags indicate attractive clothing items, and conversely for *negative expansions*. In this thesis, we found the tags $J^{(+)}$ = "fashionable" and $J^{(-)}$ = "unfashionable" to be the most effective for positive and negative expansions, respectively. Alternatives as "best seller" and "unattractive" were considered, returning similar results.

Each expansion, either positive or negative, is associated to a particular $k = 1, ..., K_{past}$ for the time interval $[t-k-W, t-k]$, where W is a temporal window we wish to consider for the image search, also expressed in weeks. In our experiments we set $W = 4$, which translates to having a sliding window of size 4 and stride of 1 over the temporal axis. This allows the pool of downloaded images to disclose what are newly indexed items in relation to previous time steps, developing a temporal locality in the data pool. The precise value of W was chosen after an empirical evaluation over the range $1, ..., 12$.

5.3.3 Image Web Search

A given expanded textual query along with a time interval is fed into a web API request to gather M representative fashionable and unfashionable images ${}^{(+)}\{\mathbf{x}_i\}_{i=1,\dots,M}^{(t-k)}$; ${}^{(-)}\{\mathbf{x}_i\}_{i=1,\dots,M}^{(t-k)}$ that have been uploaded in the interval $[t-k-W, t-k]$, for $k = 1,\dots,K_{past}$. In particular, we adopt Google Images search, selecting the first $M = 25$ images returned, assuming the ordering of Google Images perfectly mirrors a genuine image relevance [68]. After the image web search phase, $M \times K_{past}$ fashionable and unfashionable images are collected respectively (as shown in Fig. 5.6). These images are then used to train a binary classifier θ, aimed at distinguishing fashionable from unfashionable images. Webly learning and supervision based on Google Images has been constantly considered in computer vision throughout the years, especially for image classification and object detection [37, 21, 77]. POP goes one step further, merging visual and textual search while adding a time-dependent query expansion to create more discriminative image sets. Nevertheless, the labels assigned to the images from the query expansions might be noisy, therefore we apply a confident learning method.

5.3.4 Learning From Noisy Labels

In the following, we adapt the confident learning (CL) methodology specifically for our binary problem. For a broader overview, readers may refer to [111]. Let $\mathbf{X} = \{\mathbf{x}_i, \tilde{y}_i\}_{1\dots N}$ be our set of $N = 2(M \times K_{past})$ images with associated observed noisy binary labels $\tilde{y}_i \in \{\text{“fashionable”}, \text{“unfashionable”}\}$. CL assumes that a true, latent label $y_i^* \in \{\text{“fashionable”}, \text{“unfashionable”}\}$ exists for every sample. CL requires two inputs: 1) the out-of-sample $N \times 2$ matrix $\hat{\mathbf{P}}$ of predicted probabilities where $\hat{\mathbf{P}}_{i,h} = \hat{p}(\tilde{y}_i = h; \mathbf{x}_i, \theta)$ with θ a generic (binary) classifier initially trained on $\mathbf{X}$; 2) the set of noisy labels $\{\tilde{y}_i\}$. Subsequently, a robust 2×2 confusion matrix, called the *confident joint* matrix $\mathbf{C}_{\tilde{y},y^*}$, is computed[2]:

$$\begin{aligned} &\mathbf{C}_{\tilde{y},y^*}(h,l) = |\hat{\mathbf{X}}_{\tilde{y}=h,y^*=l}|, \text{with} \\ &\hat{\mathbf{X}}_{\tilde{y}=h,y^*=l} = \left\{ \mathbf{x} \in \mathbf{X}_{\tilde{y}=h} : \hat{p}(\tilde{y}=l; \mathbf{x}, \theta) \geq t_l \right\} \end{aligned} \tag{5.1}$$

where t_l is a threshold that represents the expected self confidence value for each class:

$$t_l = \frac{1}{|\mathbf{X}_{\tilde{y}=l}|} \sum_{x \in \mathbf{X}_{\tilde{y}=l}} \hat{p}(\tilde{y}=l; x, \theta) \tag{5.2}$$

In practice, $\mathbf{C}_{\tilde{y},y^*}$ counts only those elements which have been confidently classified in a particular class, where the term “confident” means with a probability that is higher than the average probability of an element belonging to that class. In simpler words, if samples labeled as belonging to class h tend to have higher probabilities because the model is over-confident about class h, then t_h will be proportionally larger. It also worth

[2]We drop the index i for clarity.

noting that Eq. 5.1 corresponds to a simplified version of the general building procedure of the confident joint matrix $\mathbf{C}_{\tilde{y},y^*}$ of [111], which nonetheless in our case is perfectly fine since we deal with binary classification and no *label collision* may happen, *i.e.,* the fact that a noisy label can correspond to a more than a single alternative class.

On this robust confusion matrix, we estimate label errors from the off diagonal elements of $\mathbf{C}_{\tilde{y},y^*}(h,l)$. This is shown to be highly performing and theoretically grounded in [111]. Wrongly labeled images are therefore pruned (indicated by the red boxes in Fig. 5.6), obtaining the cleaned fashionable and unfashionable images ${}^{(+)}\{x'_i\}^{(t-k)}_{i=1,\dots,M'^{(t-k)}}$; ${}^{(-)}\{x'_i\}^{(t-k)}_{i=1,\dots,M''^{(t-k)}}$, where $M'^{(t-k)}$ and $M''^{(t-k)}$ indicate that we can have a different number of positive and negative images, respectively, related to each $t-k$ time step, due to the noisy sample elimination. The classifier is retrained on the cleaned data, obtaining a robust trained model θ'. This procedure purely data-centric and model agnostic; the specific θ used in this work is described in Sec. 5.5.2.

5.3.5 Signal Forming

The POP signal $S^{(t)}_{\mathbf{z}} = s^{(t-K_{past})}_{\mathbf{z}}, \dots, s^{(t-k)}_{\mathbf{z}}, \dots, s^{(t-1)}_{\mathbf{z}}$, is computed by considering the cleaned fashionable images ${}^{(+)}\{\mathbf{x}'_i\}^{(t-k)}_{i=1,\dots,M^{(t-k)}}$, the robust model θ', and the image $\mathbf{z}$, as follows:

$$s^{(t-k)}_{\mathbf{z}} = \frac{1}{M^{(t-k)}} \sum_{i=1}^{M^{(t-k)}} \frac{\langle \theta'\left({}^{(+)}\mathbf{x}'^{(t-k)}_i\right) \cdot \theta'(\mathbf{z}) \rangle}{\| \theta'\left({}^{(+)}\mathbf{x}'^{(t-k)}_i\right) \| \| \theta'(\mathbf{z}) \|} \tag{5.3}$$

where $\theta'(\mathbf{z})$ indicates the extracted features of $\mathbf{z}$ from θ', and $\langle\cdot\rangle$ indicates the scalar product. In practice, the signal value $s^{(t-k)}_{\mathbf{z}}$ is the average cosine similarity between the embedding of the probe image $\mathbf{z}$ and each webly image $\mathbf{x}'_{i^{(t-k)}}$, computed over $M^{(t-k)}$ downloaded images.

5.4 GTM-Transformer

The structure of the proposed model is depicted in Fig. 5.7: GTM-Transformer is based on the Transformer model [147], yet we deviate from the canonical form by considering a non-autoregressive variant [45], motivated by two reasons: i) to avoid the compounding of errors caused by wrong initial predictions; ii) to generate the forecasted time series in one go, without any recurrence mechanism, allowing for faster training and inference. In particular, GTM-Transformer learns different representations for each input type and then projects such representations in a novel latent space to non-autoregressively forecast the sales. The different components of the model are explained in detail below:

The transformer encoder takes as input the exogenous time series of the product, 3 series for Google Trends (one for each attribute), 1 serie for POP signals. The series are projected into a higher dimensional space R^D enriched with a positional encoding. This signal is then processed by the standard encoder block of [147], by applying Scaled Dot-product Self-Attention. We employ *masking* which enforces localized Attention on the time series [121]. The encoder outputs $\psi_t \in R^D$: a representation of the exogenous

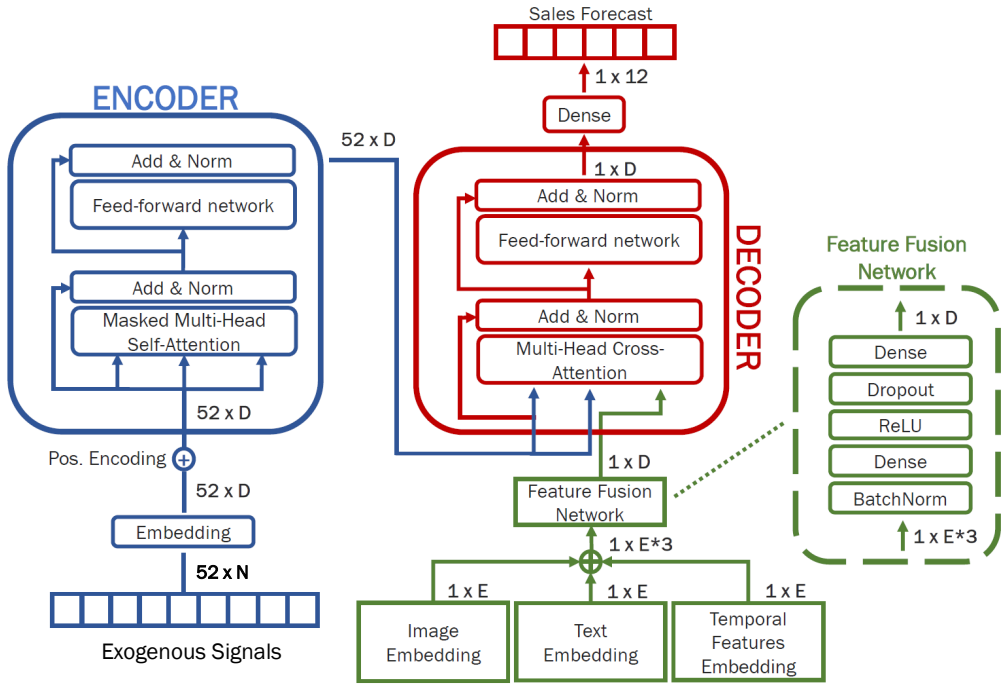

Figure 5.7: GTM-Transformer architecture. The encoder processes the exogenous series. The decoder takes as input a multimodal embedding created from the Feature Fusion Network and attends to the encoder's output. The output of the transformer model is then passed through a dense layer, to *generate* the sales forecasts.

time series enriched with information about which portions of itself are more important. This information is then fed to the decoder, acting as a type of prior knowledge on the popularity of the product.

The image embedding module uses a ResNet-50 model [54] pre-trained on ImageNet [29] to extract 2D convolutional features $\phi_{i_{resnet}} \in R^{CxWxH}$ (where $C = 2048$ is the number of final feature channels, W represents the image width and H the image height). Finally, Average Pooling with a square kernel of size 1 is applied, followed by a Dense layer, creating a compact representation of the image $\phi_i \in R^E$.

The text embedding module consists of a BERT model [30] pre-trained on a large corpus comprising the Toronto Book Corpus and Wikipedia. This module takes as input the textual tags, i.e $color, category, fabric$ and produces an embedding $\phi_{t_{bert}} \in R^{768}$ of the words. BERT adopts particular, reserved tokens when trained like *[CLS]* and *[SEP]*. Because this module is not fine-tuned, after the tokenization process, we discard the embeddings for these reserved tokens. By exploiting a pre-trained model, our model can obtain a representation for any textual tag that it might have never seen before, while also obtaining additional context from the textual information. The module averages the embeddings for each attribute and then uses a Dense layer to create a compact representation of the text $\phi_t \in R^E$.

The temporal features embedding module, is a feed-forward network that creates a joint embedding of the temporal features available for each product: the day of the week, the week of the year, the month and the year. An individual embedding $\phi_j \in R^E$ is created for each one of these features. Afterwards, these embeddings are concatenated and merged together through a dense layer, creating a final representation $\phi_{temp} \in R^E$ of all these temporal features.

The feature fusion network is another feed-forward network that merges the separate multimodal embeddings together, creating a learned representation $\psi_f = f(\phi_i, \phi_t, \phi_{temp}), \psi_f \in R^D$ where $f(\phi_i, \phi_t, \phi_{temp}) = W_{d2} * RELU(W_{d1} * [\phi_i; \phi_t; \phi_{temp}]) + B_{d2}$.

The transformer decoder is the component which performs the forecasting. Alternatively to the decoder block of [147], we remove the Self-Attention segment, since the input coming from the feature fusion network is a single representation and not a sequence. The input is fed to the Multi-Head Cross-Attention attention segment as the query, producing a product embedding $\psi_p \in R^D$ containing information about the exogenous trends of the solar year before the product's release date. Therefore, ψ_p is a compact representation of four different modalities: $[\psi_t, \phi_i, \phi_t, \phi_{temp}]$. After passing through the decoder's feed-forward segment, a dense layer projects ψ_p into $R^{horizon}$ in order to produce the desired forecasts based on the specified forecast horizon.

Summarizing, GTM-Transformer works by feeding to the decoder the exogenous embedding (produced by the encoder) and the multimodal embedding (produced by the Feature Fusion Network) to generate the forecasts.

5.5 Task 1: New Fashion Product Sales Curve Prediction

In this section we discuss the experiments carrierd on the first task different task: "New Fashion Product Sales Curve Prediction". The prediction of sales curves for a probe clothing item $\mathbf{z}$ provides as output a time series $O_{\mathbf{z}}^{(st)} = o_{\mathbf{z}}^{(st+1)}, \ldots, o_{\mathbf{z}}^{(st+k)}, \ldots, o_{\mathbf{z}}^{(st+K_{fut})}$ of how many pieces of z will be sold in a given season, starting at a particular time step st (the start of the season), for the next K_{fut} time steps.

Experimental Protocol

On VISUELLE we define an experimental protocol that simulates how a fast fashion company deals with new products, focusing on two particular moments: i) the *first order setup*, which is when the company orders the first stock of products to be distributed in the shops, usually two months before the starting season; ii) the *release setup*, which is right before the start of the season, and is useful to obtain the best forecast by using all of the exogenous information at hand, so to have a preliminary idea of when to do the stock replenishment. For these two moments we use 28 and 52 timesteps long exogenous signals, respectively.

As forecast horizon, we consider 6 weeks, as it is the period where no interventions are made by the company, such as reordering or retirements of products (if they perform very poorly). In any case, all models classifiers have been trained assuming a 12-week prediction, and shorter horizons have been taken into account for the evaluation. This procedure maximized the performances of all the approaches. Nonetheless results at different horizons will be shown here as for our approach. To perform the experiments, we divide the data into a training and testing partition, where the testing products are composed of the 497 most recent products. The rest of the dataset (5080 products) is used for training.

We utilize the *Weighted Absolute Percentage Error* [60] as the primary error measure. It expresses the forecasting accuracy as a ratio:

$$\text{WAPE} = \frac{\sum_{t=1}^{T} |y_t - \hat{y}_t|}{\sum_{t=1}^{T} y_t} \tag{5.4}$$

where T is the forecasting horizon. WAPE is always nonnegative, and a lower value indicates a more accurate model. Even though it is a percentage-based metric, it is not bounded by 100.

For a more articulated understanding of our approach, we compute the *Mean Absolute Error* (MAE), also known as Mean Average Devation (MAD):

$$\text{MAE} = \frac{\sum_{t=1}^{T} |y_t - \hat{y}_t|}{T} \tag{5.5}$$

MAE describes the mean quantity by which the forecast misses the values on their respective scale.

Forecasting bias [17] is another aspect to take into account, measuring systematic over- or underestimation of the forecast w.r.t. the correct value. Even if a slight forecast bias might not have a notable effect on store replenishment, it can lead to over- or undersupply at the central warehouse. To measure the forecasting bias, we adopt the *tracking signal* (TS) measure [17, 104]:

$$\text{TS} = \frac{\sum_{t=1}^{T} y_t - \hat{y}_t}{MAE} \tag{5.6}$$

which is basically the signed difference between actual and prediction value, divided by the MAE. The sign of the tracking signal communicates if we have an overestimation (if negative) or an underestimation (if positive). The closer to zero, the more unbiased the forecast. In the literature, a forecasting approach is considered to be consistently biased if the tracking error is above 3.75 or below -3.75 [17, 104].

Finally, we focus on the capability in providing a forecasting curve which resembles the ground truth, as a way to highlight whether the model has properly captured the actual signal dynamics. To this end, we exploit the Edit distance with Real Penalty (ERP) [20] which borrows from the classical Edit Distance (ED). ED works on discrete sequences, counting the number of edit operations (insert, delete, replace) that are necessary to transform one series into the other. ERP uses the following algorithm: if

the Euclidean distance between prediction $\hat{y}_t$ and y_t is smaller than a penalty ϵ, they are considered equal (d=0) and if not they are considered different (d=1). Summing over differences along the time axis gives the final distance. Because we are dealing with continuous values, a threshold ϵ=0.03 is used to decide if values are assumed to be different. ERP is a dissimilarity $\in \mathbb{R}_+$, so the closer to 0 the better.

Comparative results

Comparing GTM-Transformer with other approaches in the literature requires particular care, since we are the first to exploit Google Trends and to create a data-centric signal as exogenous variables to forecast sales for new products. For this reason, together with considering state-of-the-art alternatives in their original form, we adapt them by injecting Google Trends and POP wherever this modification is natural, for example on models which already do process exogenous data. All the code, including the one for the competitors will be made publicly available, for the sake of fairness. To ease the reading, the name of the approaches will be followed by a square parenthesis indicating the type of information exploited within: T for textual data (category, color, fabric and release date), I for image data, G for exogenous signals. Additionally, the name of the approaches which have been augmented with the exogenous signals will be followed by a "+G". More in the detail, we consider:

kNN models. These non-parametric methods methods are proposed in [36], and follow a common guideline for fast fashion companies: sales of new products will be similar to older, similar products they have already commercialized [143]. The idea is to define a similarity metric between products and then forecast the sales of the new product by averaging the sales of the k most similar products that have sold before. Let P be set of all products and let $d(x_{p_i}, x_{p_j}), \forall x \in P$ be the distance between any two products. We can then obtain the set of k nearest neighbors to a product $K = \{x_1..x_k|P, d\}$. We can then estimate the sales of the a product x_p using a weighted average the sales of its neighbors $\sum_{k=1}^{K} \frac{d(x_p,x_k)}{\sum_{k=1}^{K} d(x_p,x_k)} y_k$, where y is the sales time series. The three KNN alternatives proposed in [36] are all considered here, which depend on the data they consider to capture the similarity: i) between product attributes (color + category + fabric), *Attribute KNN*; ii) Between product images (*Image KNN*); iii) Between the product attributes *and* images *Attribute + Image KNN*. In our experiments, we use the cosine distance and set $k = 11$. These models are used only without exogenous signals since possible modification would be not natural.

Gradient Boosting [38]. This fundamental technique has been used in time series forecasting either as solitary models [55] and recently as components of more elaborate architectures [63]. Gradient Boosting is an ensemble model which aggregate the results from multiple Decision Trees, where we assume Gradient Boosted Trees. Decision Trees are simple, tree-like diagrams for decision making. Gradient Boosted Trees build trees one after the other, such that each new tree helps correct the errors made by the previous

one. This is done by fitting the trees on the negative of the gradient of a particular loss function (similarly to Backpropagation through SGD in Neural Networks). We use 500 trees and set least squares as the optimization problem. When using this model, the additional features, both exogenous and not, are concatenated together and fed to the model.

Multimodal Encoder-Decoder RNNs. It is proposed as most advanced techniques in [36]. The idea is to perform sequence learning in a two-step process, where an Encoder module takes the available information and produces a learned feature representation of the various modalities. This is then fed to an GRU[24] network that acts a Decoder, which autoregressively performs the forecasting. The authors augment their architecture with Bahdanau Attention[9], using the last produced decoder hidden state to learn, at each prediction step, which one of the various modalities provides more important information to the forecast. In particular, we consider the two best performing techniques from the original paper, that is the *Concat Multimodal RNN*, which which learns joint embeddings derived by concatenating embeddings of individual input modalities and the *Cross-Attention RNN*, which learns multimodal attention weights and temporal attention weights to create an improved joint embedding. Both these architectures natively accomodate the use of Google Trends, so we feed the trends in the exogenous data module as depicted in [36].

We train all the neural networks for 200 epochs with a batch size of 128 and MSE (Mean Squared Error) loss function, using the AdaFactor [126] optimizer, on an NVIDIA Titan RTX GPU.

5.5.1 New Fashion Product Sales Curve Prediction: Google Trends

In this section we discuss the experiments with Google Trends, starting with a preliminary study on how Google Trends correlate with the sales. Next, we analyze the first results about how our approach does perform against 9 comparative approaches covering the emerging literature of the new product sales forecasting. Subsequently, an ablation study investigates the role of the different modalities we take into account, namely textual data, image data and the Google Trends (see Sec. 5.1). The analysis of the performance on the single categories is showed in the next section, while the analysis on different time horizons completes the series of experiments in the last section.

Correlation analysis with Google Trends

The goal is to check the strength and direction of monotonic association between the sales time series and the Google Trends, motivating their use in our framework. As a preprocessing step, we test the time series for stationarity using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test [74], to make sure that the potential correlations will not be simply due to the dependency on time, resulting in spurious correlations [5].

First Order Setup, G.Trends: 28 weeks					
Method	**Input**	**WAPE**	**MAE(25%;75%)**	**TS**	**ERP**
Attribute KNN [36]	[T]	59,8	32,7(18;39)	-0,88	0,40
ImageKNN [36]	[I]	62,2	34,0(19;42)	-1,09	0,43
Attribute + Image KNN [36]	[T+I]	61,3	33,5(19;39)	-1,10	0,41
Gradient Boosting [38]	[T+I]	64,1	35,0(21;41)	-1,58	0,43
Gradient Boosting+G [38]	[T+I+G]	64,3	35,1(21;41)	-1,71	0,43
Concat Multimodal RNN [36]	[T+I]	63,3	34,4(18;44)	-0,67	0,42
Concat Multimodal RNN+G [36]	[T+I+G]	64,1	34,8(18;43)	-0,21	0,43
Cross-Attention RNN [36]	[T+I]	59,5	32,3(16;39)	-0,32	0,38
Cross-Attention RNN+G [36]	[T+I+G]	58,7	31,9(16;39)	-0,88	0,38
GTM-Transformer	[T+I+G]	**56,8**	**31,0(15;38)**	0,90	**0,35**
GTM-Transformer**	[T+I+G+Extra]	**54,4**	**29,7(14;36)**	0,44	**0,31**

Table 5.1: Results on VISUELLE with *first order setup*. Forecasting horizon = 6 weeks.

34% of sales time series are found to be non-stationary and are not considered for the analysis.

For each product, we utilize its associated 52-week Google Trends, based on the textual attributes. We calculate the Spearman correlation coefficient against the 12-week sales, using a sliding window protocol with window length $w = 12$ and stride of one step. Even though the small sample size does not encourage the use of correlation analysis [28], we wish to investigate the distribution of significant correlations and in particular if they are located on specific periods of the trends. In other words, we are more interested in where the correlations are located across the trends, rather than their values.

The results give statistically significant ρ correlation coefficient in 86% of the total cases. On this selection, the strongest correlations were found to be positive, with 19% of all coefficients in the range [0.75,1]. The lags that contain the strongest correlations are contained mostly (54% of the cases) in the range [-42,-32].

These findings are quite interesting, since they state that the period which is most correlated to the sales is seven to ten months before the product's release date, which corresponds loosely to the end of the same fashion season from the previous year. This preliminary analysis provides further motivation for the use of the Google Trends and is later confirmed by the cross-attention weights of GTM-Transformer in Sec. 5.5.1.

Results on New Fashion Product Sales Curve Prediction

Tables 5.2 and 5.1 reports the results, where the following facts can be pointed out:

- The use of Google Trends boosts the performance of all the models, except Concat Multimodal, where the Google Trends have been simply concatenated as static data.
- Our GTM-Transformer gives the best results in both setups (first order and release setup), with the best MAE and WAPE and the second best Tracking Signal, displaying a good balance between over and underestimation; also, we have the best ERP, which indicates that the shapes of our forecasting curves better resemble the actual sales (as also seen in Fig. 5.8).

Release Setup, G.Trends: 52 weeks					
Method	**Input**	**WAPE**	**MAE(25%;75%)**	**TS**	**ERP**
Attribute KNN [36]	[T]	59,8	32,7(18;39)	-0,88	0,40
ImageKNN [36]	[I]	62,2	34,0(19;42)	-1,09	0,43
Attribute + Image KNN [36]	[T+I]	61,3	33,5(19;39)	-1,10	0,41
Gradient Boosting [38]	[T+I]	64,1	35,0(21;41)	-1,58	0,43
Gradient Boosting+G [38]	[T+I+G]	63,5	34,7(20;41)	-1,55	0,42
Concat Multimodal RNN [36]	[T+I]	63,3	34,4(18;44)	-0,67	0,42
Concat Multimodal RNN+G [36]	[T+I+G]	65,9	35,8(19;45)	-0,41	0,44
Cross-Attention RNN [36]	[T+I]	59,5	32,3(16;39)	-0,32	0,38
Cross-Attention RNN+G [36]	[T+I+G]	59,0	32,1(17;38)	-0,18	0,38
GTM-Transformer	[T+I+G]	**55,2**	**30,2(15;36)**	0,41	**0,33**
GTM-Transformer**	[T+I+G+Extra]	**54,2**	**29,6(14;35)**	0,56	**0,33**

Table 5.2: Results on VISUELLE with *release setup*. Forecasting horizon = 6 weeks.

- The tracking signal indicates persistent forecasting bias if its value is above (below) 3.75 [17, 104]. Not one of the methods used has this problem, including our GTM-Transformer. This shows that even though the models have gotten much more complex, we are still able to maintain a strong balance between positive and negative errors. GTM-Transformer remains balanced even with 28-week Google Trends.
- Using shorter Google Trends (28-week, Table 5.1) gives performances which in general are just slightly worse, proving once again their usefulness. An explanation for this can be inferred when looking at the attention weights, which are explored in Sec. 5.5.1

To explore the generalization of the model to additional types of visual attributes, we consider the tags from Fashion IQs [155]: they represent a widely-known approach to describe fashion items for automated retrieval purposes. We apply the attribute extraction code directly to our data, focusing on the "shape" attribute, which describes fine-grained aspects of the structure of the product (v-neck, hem, ...). We discard the other types of attributes, since they consistently overlap with ours (such as the "fabric" attribute) or do not fit very well with VISUELLE, because in Fashion IQ clothes are worn by models. After the attribute extraction, we download the related Google Trends as described in Sec. 5.1. We dub this model in Tables 5.2 and 5.1 as GTM-Transformer**. Interestingly, adding complementary information boosts further the model, promoting once again the use of the Google Trends.

Additional insight can be inferred by some qualitative results, showing two 12-week predictions (Fig. 5.8): Attribute KNN gives reasonable estimates, trying to capture the scarce performance of the first 6 weeks portrayed in the second plot. Gradient Boosting overestimates both the cases, offering a graphical demonstration of its high tracking signal TS=-1.58 (Table 5.1). The RNN-based approaches Concat Multimodal+G, Cross Attention RNN+G seems to have a very regular slope, irrespective of the real structure of the sale signal: this is likely due to the nature of the autoregressive approach, which has learned the general sale curve dynamics and struggles with trajectories which deviate from it. With the GTM-Transformer the role of the Google Trends appears to be clear,

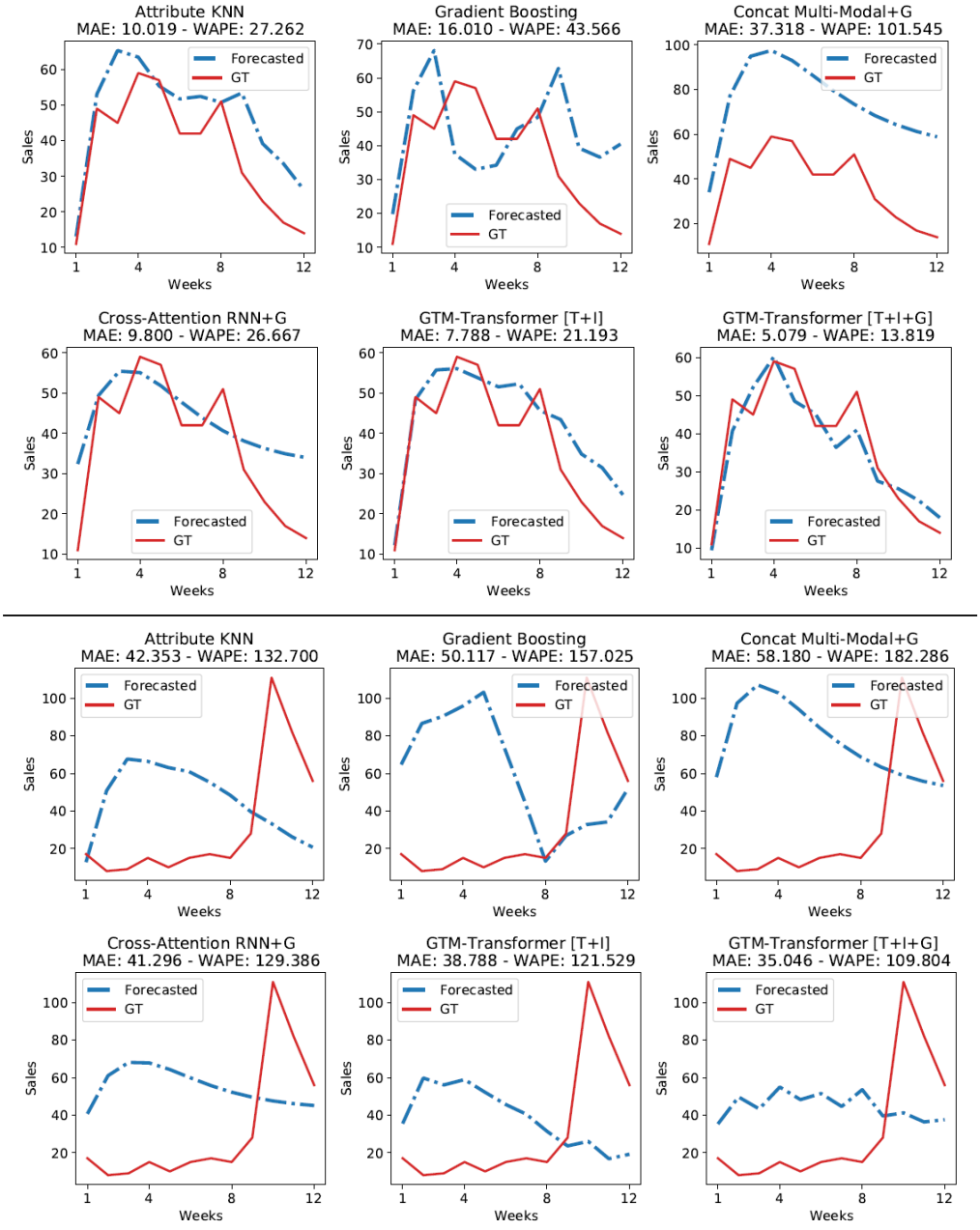

Figure 5.8: Qualitative Results

being capable of giving more structure to the final forecast (above), lowering down the forecasting thus predicting a scarce performance (below).

Ablation study

Ablative results refer to the 6-week forecasting horizon, using the full 52-week Google Trends, and are reported in Tab. 5.3.

GTM ablations	6 Weeks			
	WAPE	**MAE (25%;75%)**	**TS**	**ERP**
[I]	56,4	30,8(16;36)	-0,34	0,36
[T]	62,6	34,2(19;43)	-1,42	0,43
[G]	58,2	31,8(17;37)	-0,89	0,38
[I+T]	56,7	30,9(16;38)	-0,32	0,37
[T+G]	56,8	31,0(14;38)	1,63	0,33
[I+G]	55,7	30,4(13;32)	1,45	0,30
[T+I+G]	**55,2**	**30,2(15;36)**	0,41	**0,33**
[AR]	59,6	32,5(14;36)	1,18	0,32

Table 5.3: 6 weeks ablative results on VISUELLE with *release setup*.

The first ablation is our model without the Google Trends, so removing the encoder module in Fig. 5.7 (row [T+I]). The much higher WAPE highlights the net role of the exogenous data, and is one of the main results of our study. It is worth noting that the performances are better than all of the approaches using the same kind of information (see Tab. 5.2), proving the good design of our architecture. The two-modality combos text + Google Trends ([T+G]) and image + Google Trends ([I+G]) give WAPE scores both around 57%, demonstrating that text and images carry complementary information which the complete GTM-Transformer is capable of combining and exploiting. Single modalities ablations instead demonstrate that the image alone [I] has the best performance, and this obviously states that it is the appearance of the product which allows for the most discrimination. Surprisingly, Google Trends [G] alone gives the second best results, while text attributes [T] alone gives the worst results, indicating once again the net value of this exogenous signal.

Finally, the *[AR]* row indicates the complete model, but in its autoregressive version: the performance is 4.4% worse than our GTM-Transformer, showing the benefit of the non-autoregressive design.

Single category analysis

Is interesting to check how GTM-Transformer performs on different categories. Figure 5.9 contains the separate WAPEs, where the marker size represents the cardinality of the category (Fig. 5.2a). The results confirm the fact that performances are more stable for categories with a large number of products such as “Long sleeve” or “Culottes”, as the amount of data available for training over these products is larger.

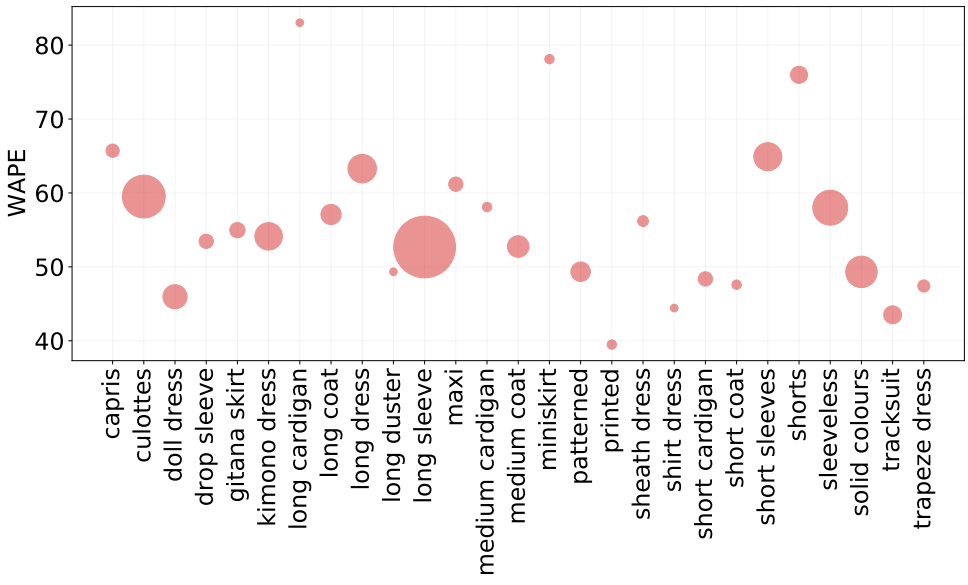

Figure 5.9: Category Results

Varying the forecasting horizon

In this section we demonstrate the effect of the forecasting horizon on the performance. Figure 5.10 contains the WAPE for 1, 2, 4, 6, 8 and 12 week forecasts. GTM-Transformer remains the best performing approach for all horizons, on pair at 2 weeks with Cross-Attention RNN+G. Most of the slopes show a minimum error at 6 weeks, except the Gradient Boosting which shows the second best performance at 1 week. The first 6 weeks performance varies greatly, with Attribute + Image KNN performing the worst. After 6 weeks, all the approaches have a decrease in the performance, which is natural, since the sale signal becomes more dependent on external choices (replenishments, discounts) we are not modeling here.

Model interpretability: unveiling the Google Trends

To understand the role of Google Trends in GTM-Transformer we exploit the interpretability of the Attention mechanism. To this sake we calculate where in the Google Trend the decoder assigns the highest Cross-Attention weight, to find if there are any systematical tendencies as to where the model looks at when making the prediction. Table 5.4 contains the results, where it can be seen that the initial period of the Google Trend seems to be the most crucial, as also hinted by the correlation analysis in section 5.5.1.

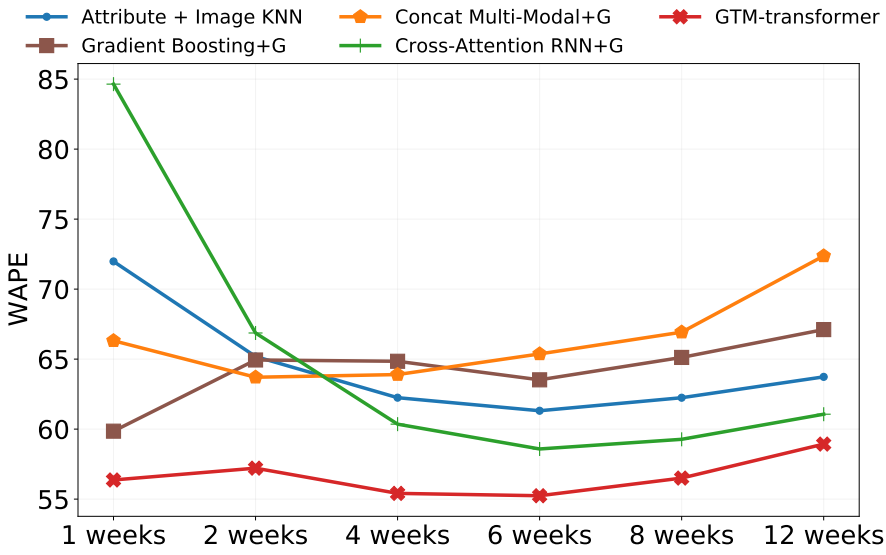

Figure 5.10: Different forecasting horizon results

Lag	*-52 – -42*	*-42 – -32*	*-32 – -22*	*-22 – -12*	*-12 – -0*
#Highest W	145	231	42	46	33

Table 5.4: Points of the Google Trends time series with the highest Cross-attention weights

A very practical use of our model: the *first-order* problem

Accurate new product forecasting is highly desirable for many reasons, as explained in the introduction: understand tendency in the sales, deciding when to replenish the warehouses, and how many products per reference to buy before the season starts. This is known as the *first-order* problem [33], and it can be accurately simulated with the real data of VISUELLE. The goal is to order a number of products that matches the sum of future sales until the sixth week, without exceeding or underestimating. During the first six weeks then, sales will help with more predictive power in suggesting how to behave with the remaining weeks, for example deciding whether to order again or not.

A general protocol to deal with the first order problem is to consider the sum of the sold products of the same period in the previous correspondent season, adding a percentage which mirrors the expected growth, and make the first order. In our case, the policy adopted by the company is to increase the orders for a product of a particular category, color and fabric by 60% of the previous average sum of sold products in the first six weeks for those attributes. We call this the 60% policy. For example, if we want to do the first order for SS19 season of a new white cotton cardigan, we take the average of sold white cotton cardigans of the previous SS18 and add the 60%.

To compute the first order error, we simply calculate the integral of the forecasting

Method	6 Weeks	
	MAE	US $ discr. ↓
60% Policy	313,6	4.390.540 $
Attribute KNN [36]	271,0	3.770.886 $
ImageKNN [36]	279,7	3.892.271 $
Attribute + Image KNN [36]	271,9	3.783.854 $
Gradient Boosting+G [38]	297,2	4.135.547 $
Concat Multimodal+G [36]	359,7	5.035.494 $
Cross-Attention RNN+G [36]	271,5	3.800.938 $
GTM-Transformer	**262,3**	**3.625.163** $

Table 5.5: First-order results on VISUELLE.

and ground truth curves for the first 6 weeks and compare them with each other, for each considered approach, including the 60% policy. To evaluate the performance, we compute the mean of all the absolute errors over all products. This tells us by how much, on average, the model is mistaken about the total sold amount and therefore the quantity of the first order. To show the real impact of such a problem, in Table 5.5 we report also the monetary discrepancy in US dollars, assuming that each reference has a cost of $28 (the average cost of a fast fashion product). In a market of around 13M dollars, the 60% policy is clearly ineffective, and all the forecasting approaches lower the discrepancy considerably, with GTM-Transformer lowering it the most.

5.5.2 New Fashion Product Sales Curve Prediction: POP Signal

In this section we show the experiments with POP signal. In line with the general requirements of DCAI [103], we show how our pipeline for creating training data for a specific model ψ solving a given task γ will give better performances than alternative pipelines. In this section, we extensively evaluate our approach with POP Signal on different classifiers on "new fashion product sales curve prediction task", showing also ablative studies.

The binary classifier θ for learning on noisy data (see Sec. 5.3.4) is based on a ResNet50 [54], pre-trained on ImageNet [29], with 2 additional fully connected layers. During the confident learning procedure, we train its last 6conv+2fc layers for 50 epochs with a batch size of 64, using cross-entropy loss, following a 5-fold cross validation protocol. AdamW [91] has been used as optimizer with $lr = 1e - 4$.

The GTM-Transformer is trained with the same setup described in Sec.5.5.

As comparative models, we consider 5 algorithms (from the oldest to newest): *Gradient Boosting* for forecasting [63], Concat Multi-Modal RNN [36] (*Concat MM RNN* in the tables), Residual Multi-Modal RNN [36] (*Residual MM RNN*), Cross-Attention RNN [36] (*X-Attention RNN*)[3] and our GTM Transformer (*GTM Transf.*).

[3]considering the code in `https://github.com/HumaticsLAB/AttentionBasedMultiModalRNN`

First Order Setup (K_{best} = 28 weeks)															
Exogenous Signal	***Gradient Boosting* [63] 2020**			***Concat MM RNN* [36] 2020**			***Residual MM RNN* [36] 2020**			***X-Attention RNN* [36] 2020**			***GTM Transformer***		
	W	M	ERP	W	M	ERP	W	M	ERP	W	M	ERP	W	M	ERP
No Signal	64.10	35.02	0.43	63.31	34.41	0.42	64.26	34.92	0.44	59.49	32.33	0.38	56.62	30.93	0.37
Google Trends	64.29	35.12	0.43	64.11	34.84	0.43	68.11	37.02	0.47	58.70	31.90	0.38	56.83	31.05	0.35
POP Signal	**63.75**	**34.83**	**0.42**	**58.09**	**31.73**	**0.39**	**58.88**	**32.16**	**0.39**	**57.78**	**31.56**	**0.38**	**53.41**	**29.18**	**0.32**

Table 5.6: Results on VISUELLE with *first order setup*; "W" stands for WAPE, "M" for MAE. Lower is better for all metrics.

Release Setup (K_{best} = 52 weeks)															
Exogenous Signal	***Gradient Boosting* [63] 2020**			***Concat MM RNN* [36] 2020**			***Residual MM RNN* [36] 2020**			***X-Attention RNN* [36] 2020**			***GTM Transformer***		
	W	M	ERP	W	M	ERP	W	M	ERP	W	M	ERP	W	M	ERP
No Signal	64.10	35.02	0.43	63.31	34.41	0.42	64.26	34.92	0.44	59.49	32.33	0.38	56.62	30.93	0.37
Google Trends	63.52	34.70	0.42	65.87	35.80	0.44	68.46	37.21	0.48	59.02	32.08	0.38	55.24	30.18	0.33
POP Signal	**63.38**	**34.62**	**0.42**	**57.43**	**31.37**	**0.36**	**58.38**	**31.89**	**0.39**	**57.36**	**31.33**	**0.36**	**52.39**	**28.62**	**0.29**

Table 5.7: Results on VISUELLE with *release setup*; "W" stands for WAPE, "M" for MAE. Lower is better for all metrics.

Results on New Fashion Product Sales Curve Prediction

The results are shown in Table 5.6 for the *first order setup* and in Table 5.7 for the *release setup*. As reference, we also report results *without* Google Trends or POP, to show the net value of the two different exogenous training signals.

For the experiments with Google Trends (Sec. 5.5.1) we also include approaches not coping with exogenous signal, whose results are reported here as reference, related to the best no-exogenous approach (*Attribute KNN*), with the following performances: WAPE of 59.8, MAE of 32.7 and ERP of 0.40.

As visible, for all the algorithms and the two setups, the POP signal boosts the performances considering all the metrics, notably *reaching the absolute best with GTM Transformer*. On average, in the *first order setup*, we improve by 3.42% over the Google Trends and by 3.21% over no exogenous signals. In the *release setup* we improve by 2.85% over the Google Trends and by 4.23% over no exogenous signals. These results demonstrate that the Google Trends attain lower forecasting performance when they are shorter while our POP signal leads to consistent improvement. These boosts have an important economical impact, as discussed in Sec. 5.5.4. Briefly speaking, considering the average item cost of Nunalie (28 US dollars), the fast-fashion brand whose data was used to construct VISUELLE, 1% more WAPE translates to more than 86K US dollars lost.

In Fig. 5.12 we show the WAPE *per category*. It is possible to note that in most of the cases we perform better than the other training alternatives, yet some particular

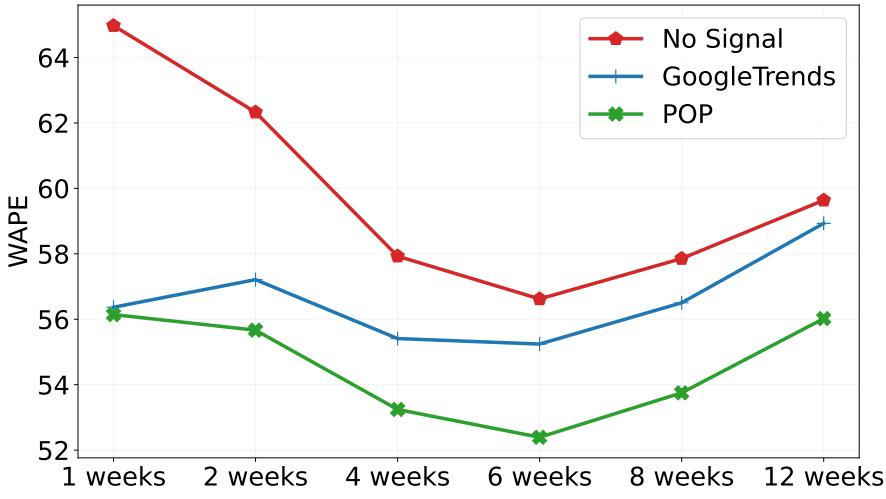

Figure 5.11: WAPE for different forecasting horizons and exogenous signals, using GTM-Transformer on the VISUELLE dataset. After six weeks there is a long enough history to model tendencies in the sales without considering product discounts or replenishments, unlike longer horizons. This is also reflected in the WAPE values, which keep increasing for forecasting horizons longer than six weeks. POP improves the forecasts for any horizon.

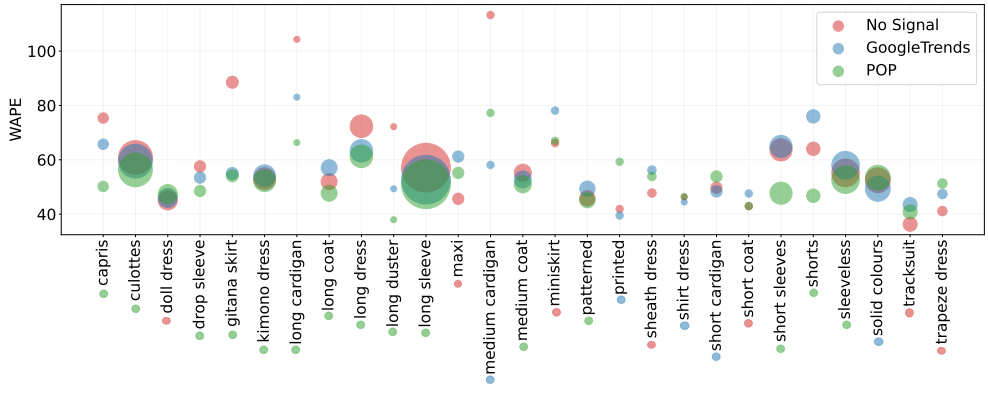

Figure 5.12: Forecasting WAPE results per clothing category; the larger the blob, the higher the # of items in that category; the color below each category name indicates the type of training setup which gives the best WAPE.

categories display limitations of our approach. These limitations are qualitatively shown in Fig. 5.13 and discussed later. Qualitative results on VISUELLE are shown in Fig 5.14.

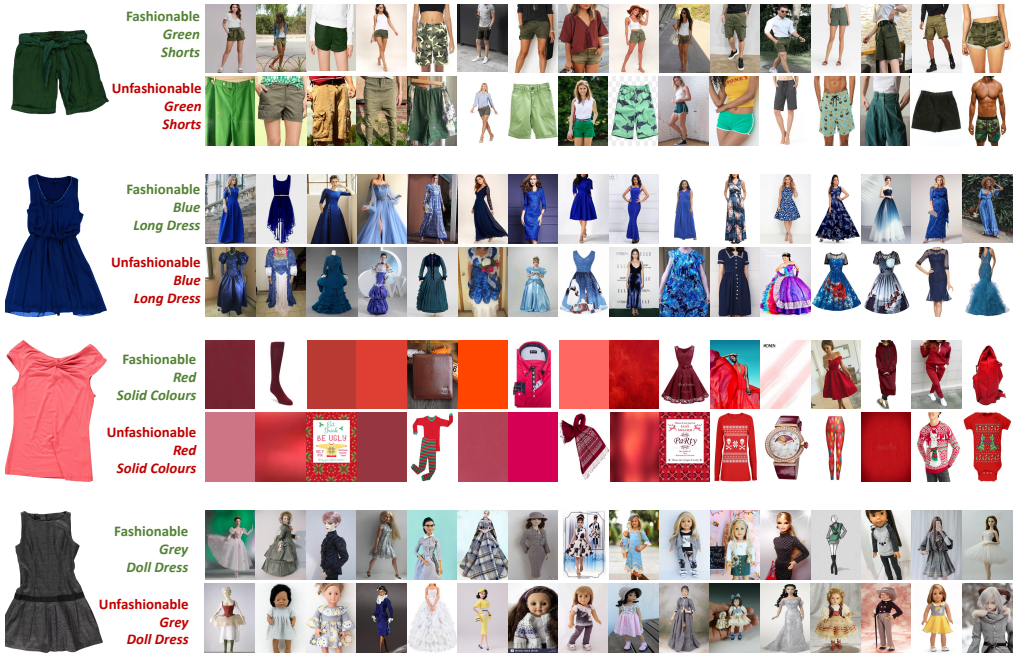

Figure 5.13: Examples of VISUELLE items (seasons SS17, SS18, SS19 and AI19, respectively) and the correspondent fashionable/unfashionable images from the web. As discussed in Sec. 5.5.4, some web images are misleading, due to some questionable category names of the VISUELLE dataset ("solid colours", "doll dress").

Alternative versions and ablative studies

Here we focus on alternative and ablated versions of our proposed pipeline, focusing on the specific modules which are illustrated in Fig. 5.5. The results are all shown in Table 5.8.

Time dependent query expansion.

- *No expansion:* we query images with the original tags collected in the Image Tagging phase, without generating positive or negative expansions. By doing this, we are essentially searching based on the "color + category" query string. This has an impact on the learning step too, since no positive or negative classes are available to learn, therefore we use our backbone ResNet50 to extract image features. For each image $\mathbf{z}$ observed at t the web images $\{\mathbf{x}_i\}_{i=1,\dots,M}^{(t-k)}$ that have been uploaded in the interval $[t-k-W, t-k]$, for $k = 1, \dots, K_{past}$ are collected. The signal forming Eq. 5.3 changes accordingly, using all the M downloaded images;

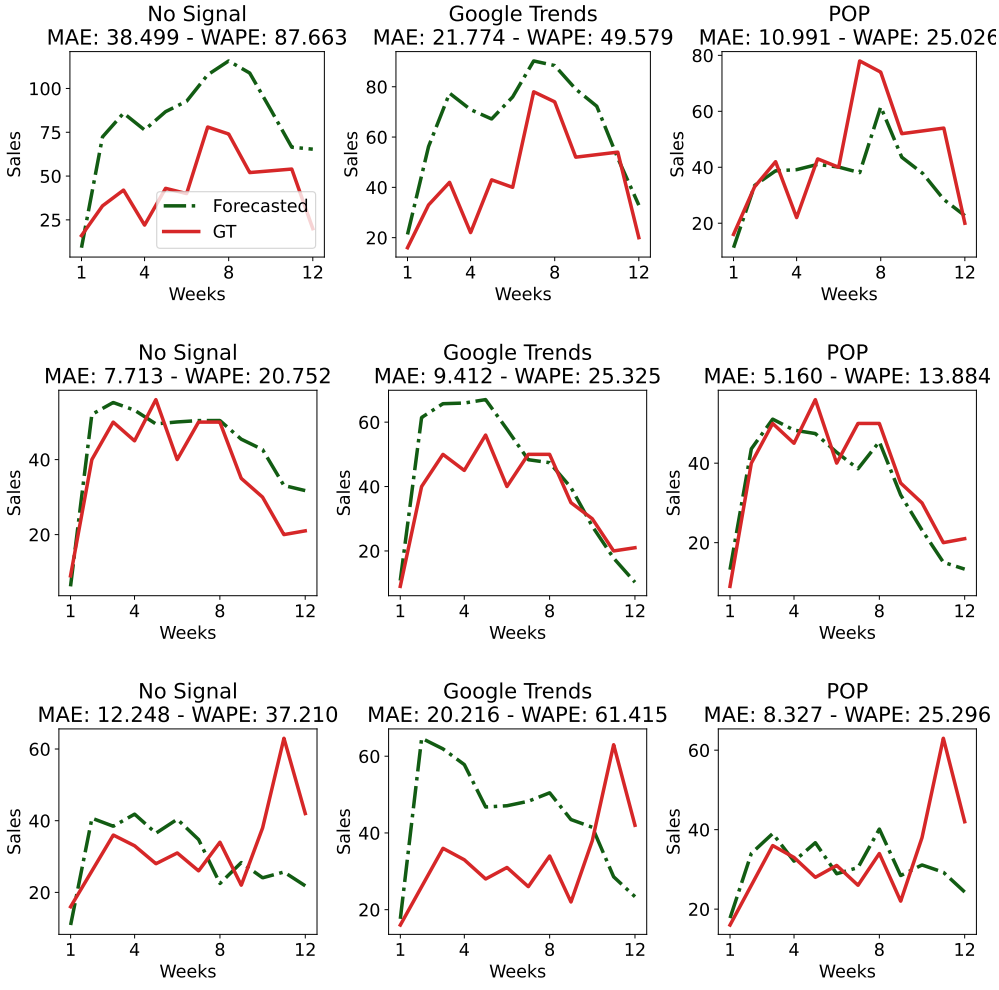

Figure 5.14: Qualitative results on VISUELLE, considering all the 12 time-steps. In all the cases POP outperforms the competitors. In the bottom plot, we show a failure case where the product is discounted in its final week of sales.

Time Dependent Query Expansion				
	Release Setup		***First Order Setup***	
Strategy	**W**	**M**	**W**	**M**
No Expansion	53.12	29.02	54.47	29.77
Misaligned past	53.02	28.96	53.63	29.30
Learning With Noisy Labels				
	Release Setup		***First Order Setup***	
Strategy	**W**	**M**	**W**	**M**
No Learning	53.03	28.97	53.83	29.41
No Robust Learning	52.81	28.85	53.59	29.28
Symmetric Cross Entropy [154]	52.63	28.75	53.58	29.27
SELFIE [135]	52.56	28.71	53.51	29.23
Signal Forming				
	Release Setup		***First Order Setup***	
Strategy	**W**	**M**	**W**	**M**
Negative	52.68	28.78	53.90	29.44
Positive and Negative	52.97	28.94	54.35	29.69
POP	**52.39**	**28.62**	**53.41**	**29.18**

Table 5.8: Alternative versions of our pipeline (Fig. 5.5) on both the *release Setup* and *first order setup*; "W" stands for WAPE, "M" for MAE. Lower is better for all metrics.

- *Misaligned past:* we modify the query expansions by looking one year earlier than the "correct" past; given the observation time t of the probe $\mathbf{z}^{(t)}$, instead of looking backwards from $t-1$ weeks to $t-K_{past}$, we go from $t-1-K_{past}$ to $t-2\cdot K_{past}$.

With respect to all the alternative versions in this study, the *No expansion* ablation gives the worst result. Even though Google Images provides images ranked by their PageRank index [116] and this guarantees in some sense to collect nice clothing images, POP provides an improvement of 0.73% and 1.06% WAPE for the *first order setup* and *release setup*, respectively. The *Misaligned past* provides slightly better results, but still performs worse than POP by 0.63% and 0.22% WAPE for the *first order setup* and *release setup*, respectively. This confirms that fashion has an evolution which, year after year, does change, and we need a proper synchronization.

Learning with noisy data. We consider alternative strategies to accomplish this phase:

- *No learning*: A predefined image classification network is used to compute the distance among embeddings of the probe image with the positive, downloaded images. This is equivalent to ablating the "Learning from Noisy Data" phase of Fig. 5.5. It will highlight the importance of dealing with distances among embeddings which are specifically learned against distances coming from a general purpose network. We utilise the backbone of our binary classifier specified in Sec. 5.3.4;
- *No robust learning*: All of the downloaded positive and negative images are used to learn our binary classifier without pruning noisy data by confident learning;
- *Symmetric cross entropy* [154]: SCE is a robust classification loss; it adds to the standard cross entropy loss a *reverse cross entropy* term which assumes the predicted labels as ground truth, and the original labels as possibly faulty. In practice, it penalizes noisy labels, without removing any associated training data;
- *SELFIE* [135]: the key idea is to correct the label of noisy *refurnishable* samples with high precision, with the help of clean data which is defined as those samples within a mini-batch creating a small loss. Repeated training runs (dubbed "restarts") allow to use more training data, *i.e.,* noisy samples which have been corrected in their labels. In particular, we use 3 restarts, after which 1.1% of both fashionable and unfashionable items have been removed from the training data.

The results in Table 5.8 show slightly different performances, promoting the general idea of learning from webly data. Nonetheless, no learning gives the worse performance, indicating that a fine tuning on the web data is beneficial (53.03 and 53.83 WAPE); when learning is done on the web data, there is some increase (52.81 and 53.59 WAPE); when learning is robust to noisy data, with SCE, performances are better (52.63 and 53.58 WAPE); removing some outliers with SELFIE gives a further help (52.56 and 53.51 WAPE). Confident learning remains the best solution, with 52.39 and 53.41 WAPE, removing the 0.8% and 1.1% of fashionable and unfashionable items respectively.

Signal forming. We change the way the POP signal is created, given the embeddings of the cleaned images by CL, and the embedding of the probe $\mathbf{z}$ by the robust model θ'.

- *Negative*; it indicates the average distance of $\mathbf{z}$ with the pruned unfashionable images ${}^{(-)}\{\mathbf{x}'_i\}^{(t-k)}_{i=1,\dots,M^{(t-k)}}$, substituting the positive ones in Eq. 5.3;
- *Positive and Negative*; here we fed into the forecasting approach two signals, the original POP and the Negative one.

The results show that the *Negative* approach gives some boost, probably accounting for how much the probe has to be dissimilar to unfashionable items. On the other hand *Positive and Negative* shows a decrease, probably because the two signals are complementary.

Qualitative results

In this section we report a qualitative analysis of the POP signal, in addition to the Fig. 5.6. These results give additional insight on the significance of our time-dependent, data-centric approach. In Fig. 5.15 and Fig. 5.16 we report two examples of the (automatically) downloaded images used for the formation of the POP signal. In both figures, the probe images from which we extract the textual attributes to index the search are depicted. The analysis for each figure is reported in the corresponding caption. We also report in the figures some pruned images by the confident learning step, marked by a red cross.

In Tab. 5.8 various ablation studies on POP. The obtained results suggest that exploiting (un)fashionable images not related to the date of delivery on the market gives worse results in terms of forecasting. Fig. 5.17 qualitatively demonstrates why this is the case. As it is visible, what made a garment of a particular type and color fashionable in 2017 (Fig. 5.17, top) does not correspond to the same visual elements that can be found in 2019 (Fig. 5.17, bottom). More specifically, throughout the spring/summer season of 2017, the green kimonos tend to be heavily associated with white patterns and the color white in general. In 2019, the kimonos are almost all in different shades of green or even dark green.

Figure 5.15: Examples of images downloaded for the query 'Grey Long Sleeves" (after pruning by confident learning). One may note that mismatching images are very few, intended as those images which are not containing any "Grey Long Sleeves". An example would be the green sleeve + blue jeans in the bottom row. It is worth noting how most of the fashionable items have no printed logos, texture or tight sleeves. On the contrary,"Unfashionable Grey Long Sleeves" have big logo on them, with a winter theme, and many colors accompanying a gray background. In some cases, the gray color actually covers a small portion of the clothing item. Pruned images are marked with a red cross.

Figure 5.16: Examples of images downloaded for the query 'Violet Long Sleeve" (after pruning by confident learning). The "Fashionable Violet Long Sleeve" items seem to have a darker tone in most cases. Very long sleeves fade into dresses, indicating the length of the garment as an important aspect for making it fashionable. Curiously, "Unfashionable Violet Long Sleeve" contain brighter colors, short garments (like pyjamas) with writings or printed images. Pruned images are marked with a red cross.

Figure 5.17: Examples of Fashionable downloaded images for particular time-depended queries. In this particular case, for the query "green kimono dress", it can be seen how the notion of fashionability can have significant variations over time. Notably, green kimonos in 2017, as seen in the latter half of the first figure, tend to be heavily associated with white patterns and the color white in general. In 2019, this trend appears to be dying out, with the kimonos being of different shades of green or even dark green.

5.5.3 Task 2: Popularity Prediction Of Fashion Styles

In this section we discuss the experiments carrierd on the first task different task: "Popularity Prediction Of Fashion Styles". The style popularity prediction task [4] is different from product sales forecasting in that it considers a measurement based on multiple clothing items, which form a style. In this case, the style is defined as a latent property of a set of images, which share some common attributes and therefore visual features. Concretely, in Fashion Forward (FF) [4], the authors apply Non-negative Matrix factorization in order to extract K styles from the attribute extraction features [87] of all the product images. The matrix to be factorized is $\mathbf{A} \in \mathbb{R}^{M \times N}$, indicating the confidence that each of the N images contains each of the M visual attributes. A can be factorized into two matrices with non-negative entries, as follows:

$$\mathbf{A} \approx \mathbf{WH}, \mathbf{W} \in \mathbb{R}^{M \times K} \quad \text{and} \quad \mathbf{H} \in \mathbb{R}^{K \times N} \tag{5.7}$$

The popularity signal y_t for a style k is built by considering the sales of all the items $\{\mathbf{z}\}$ at time t, weighted by their membership $p(k|\mathbf{z})$, which can be obtained by row-wise normalizing $\mathbf{H}$. For all the details of this procedure, we refer to the original paper [4].

To extend this problem to a NFPPF setup, we have to imagine we are evaluating the performance of a fresh new style, therefore the purpose of POP is to replace the original style popularity series and be used directly as the *only input* to the forecasting model. To this end, as textual tags we consider for each style k in FF the 2 textual attributes [87] (extracted from $\mathbf{W}$) with the highest confidence scores as textual tags, and use them to drive the time dependent query expansion. FF provides the only dataset for style forecasting where both images and product metadata are available, and where it is required to predict a popularity score on a yearly basis. The data ranges from $[2008-2013]$, but since Google Images returns little to no images for queries before 2010, the range $[2010-2013]$ is used in our experiments. Therefore we set $K_{past} = 208$, meaning that we investigate 4 years into the past. In this way we can create weekly series for each year and use the average as the value representing the popularity for that year. As probe image to create our POP signal, we consider the top 10 images $\{\mathbf{z}\}$ that represent a style (based on their membership weight $p(k|\mathbf{z})$). Each image will lead to a POP signal, which we average together to obtain the POP style signal. This process is repeated for all the dataset partitions presented in FF.

To remain faithful to the original work, we adopt their most performing statistical forecasting techniques [61], which are divided into three major groups:

1. **Naive.** These methods infer by utilizing general information from the training data. *Mean* forecasts the future as the mean of past observations, while *Last* as the last observed value. *Drift* is the same as *Last*, but the forecasts change over time based on the global trend of the series;

2. **Autoregressive.** These methods forecast using the past observations in a linear regression framework. The *AutoRegressive* (AR) model forecasts purely as described above, while the *AutoRegressive Integrated Moving Average* (ARIMA)

Global Average												
Signals	**Mean**		**Last**		**Drift**		**AR**		**ARIMA**		**SES**	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
Oracle	*0.136*	*0.170*	*0.093*	*0.114*	*0.174*	*0.222*	*0.271*	*0.403*	*0.136*	*0.167*	*0.094*	*0.116*
GoogleTrends	0.846	1.000	0.846	1.000	0.846	1.000	0.846	1.000	0.846	1.000	0.846	1.000
POP	**0.152**	**0.192**	**0.116**	**0.144**	**0.182**	**0.229**	**0.281**	**0.418**	**0.235**	**0.293**	**0.125**	**0.156**
Dresses												
Signals	**Mean**		**Last**		**Drift**		**AR**		**ARIMA**		**SES**	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
Oracle	*0.155*	*0.197*	*0.130*	*0.158*	*0.203*	*0.263*	*0.307*	*0.409*	*0.173*	*0.209*	*0.129*	*0.157*
GoogleTrends	0.849	1.000	0.849	1.000	0.849	1.000	0.849	1.000	0.849	1.000	0.849	1.000
POP	**0.119**	**0.157**	**0.108**	**0.127**	**0.173**	**0.216**	**0.229**	**0.334**	**0.162**	**0.193**	**0.109**	**0.130**
Shirts												
Signals	**Mean**		**Last**		**Drift**		**AR**		**ARIMA**		**SES**	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
Oracle	*0.122*	*0.149*	*0.075*	*0.097*	*0.148*	*0.190*	*0.301*	*0.371*	*0.126*	*0.159*	*0.080*	*0.103*
GoogleTrends	0.840	1.000	0.840	1.000	0.840	1.000	0.840	1.000	0.840	1.000	0.840	1.000
POP	**0.144**	**0.175**	**0.109**	**0.152**	**0.166**	**0.215**	**0.274**	**0.336**	**0.139**	**0.189**	**0.111**	**0.151**
Tops&Tees												
Signals	**Mean**		**Last**		**Drift**		**AR**		**ARIMA**		**SES**	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
Oracle	*0.132*	*0.165*	*0.074*	*0.087*	*0.172*	*0.212*	*0.206*	*0.429*	*0.108*	*0.133*	*0.073*	*0.087*
GoogleTrends	0.848	1.000	0.848	1.000	0.848	1.000	0.848	1.000	0.848	1.000	0.848	1.000
POP	**0.193**	**0.245**	**0.131**	**0.153**	**0.206**	**0.257**	**0.341**	**0.585**	**0.405**	**0.497**	**0.156**	**0.186**

Table 5.9: Results across all the Fashion Forward [4] datasets.

models adds the additional information of integration for time series stationarity, and a moving average component [16];

3. **SES.** Stands for simple exponential smoothing, a method that forecasts using weighted averages of previous observations, where the weights decrease exponentially as observations come from further in the past.

Following the protocol of [4], the goal is to train the models on all but the last timesteps, and test the models on the last one. We are also interested in verifying how similar the generated POP series is to the ground-truth style popularity series, essentially testing if it could be used as a replacement, solving the NFPPF setup. We utilise the mean absolute percentage error (MAPE) and the mean absolute error (MAE) to evaluate the forecasting accuracy, as in [4]. In order to provide a comparison for both cases, we show the results using Google Trends as the substitute popularity time series. Note that to obtain fair and comparable results, we rescale all the signal values (both POP and ground-truth FF series) in the range [0,1] using min-max normalization. The globally averaged results and per partition results are shown in Table 5.9, where *Oracle* refers to the original ground-truth style popularity series forecast.

As demonstrated in Table 5.9, POP acts as a natural substitute to the ground-truth style popularity time series. As a matter of fact using POP allows for a better forecast

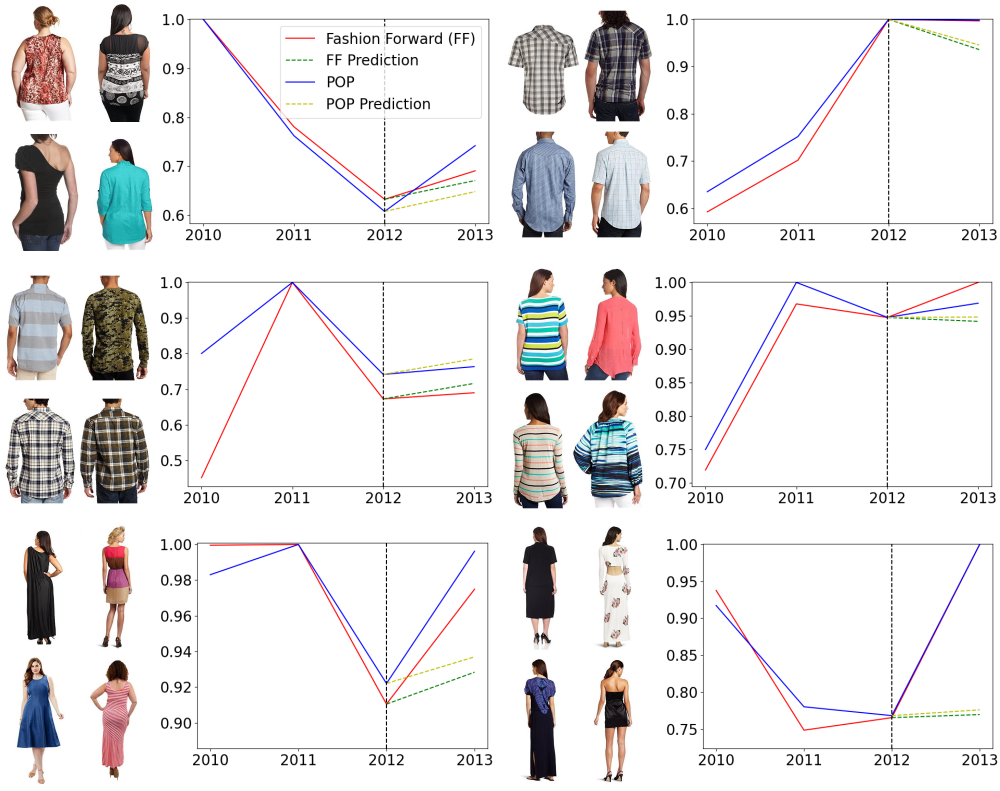

Figure 5.18: Qualitative results for the forecasting performed on six different styles from FF represented by the four images shown besides the plots. In the two topmost rows, POP and the ground-truth signals are substantially similar, while on the bottom row two relatively similar series are displayed, along with a forecasting failure case in the bottom-right plot.

of the future popularity than the ground truth signal itself in the *Dresses* partition. The qualitative results in Fig. 5.18 verify this claim, showing how structurally similar the potential performance generated series and the ground-truth FF series are. As further proof of their similarity, the ERP (if the full length series are similar in form *i.e.,* if one can replace the other) between POP and FF popularity signals are 0.31, 0.23, 0.23 for the Dresses, Shirts and Tops&Tees partitions respectively, for an average of 0.26 over all three datasets. On the other hand, Google Trends are not able to convey such similarities, partially because searching only with textual tags might not provide meaningful series.

5.5.4 Discussion

Overall, learning trend signals driven by cross-modal queries outperform Google Trends, where the image is simply used for textual attribute extraction. It is also worth noting that the tag data which we use are the same, making the comparison completely fair.

These tags are coming from the technical sheet of the probe image, which is given by default in the fashion supply chain. Our cross-modal pipeline, in all of its variations shown in Sec. 5.5.2 and Tab. 5.8 achieves the best performance when compared not only to another signal, such as the Google Trends, but also to other robust learning approaches. As further proof of the capabilities of POP, it has proven to be beneficial also in a challenging scenario like the current one, under the COVID pandemic: on sales in 2021, the forecasting accuracy reaches a 69.11 WAPE and 31.83 MAE. The experiments on the Fashion Forward datasets show that our approach is portable to different fashion domains: in Fashion Forward clothes are worn by models, in VISUELLE clothes are shown without models.

Limitations.

A potential limitation of our pipeline is that the Image Tagging phase is assumed as flawless since we are sticking with the technical sheet accompanying the probe image to extract the color and category tags. The results per category (Fig. 5.12) display how possibly mislabeled categories, or categories labeled in a general manner like ("solid colours","doll dress") may bring to misleading web images. As visible in Fig. 5.13, the related images from the web, both fashionable and not, are completely useless, since the tag of the category itself is misleading. This happened because we decide to use the category tag given by the VISUELLE dataset; it is clear that in such cases a robust automated category extraction could potentially bring better results.

Ethical Aspects And Societal Impact.

Ethical implications could *in principle* arise from the web image search: observed images can contain copyrighted images. At the same time, just as a normal user will use Google Images to gather an opinion of what could be a trend in fashion, so do we, albeit automatically. In particular, we do not need to personally look at the web images (apart from the ones reported in the thesis for explanatory purposes), since the POP signal is just a numerical time series. As for the societal impact, our approach is highly beneficial for fast fashion, which is the third most polluting industry in the world. The problem is that many clothing items remain unsold as stock, while lacking refurnishment for highly desired items causes the reactivation of the supply chain, causing more pollution. Having a precise estimation of sales or popularity has a role in amelioration the situation, and our pipeline can play a leading part. To put this discussion in quantitative terms, our signal, fed into the best performing forecasting model on the VISUELLE scenario, allows to spare 21% w.r.t. ordinary guidelines for new fashion products, reducing a loss of $4.390.400 US dollars to $3.491.600 US dollars, assuming a general price of 28$ per piece for all products, independently on the category.

Chapter 6

Conclusions

Fashion is a fast-growing industry, accentuated by the rapid spread of online markets. With this advancement, there is a corresponding, ever-increasing mass of multimedia data (images, videos and text). This huge amount of valuable information can be exploited to build advanced systems that can deal with such data for a variety of tasks, helping users and industries alike. The main purpose of these systems is to automatize the human processes, that require too much manual effort. All these processes are often based on aesthetic judgments, such as choosing the most similar garment within a gallery of images, describing clothes or patches thereof by attributes, discriminating fashionable against unfashionable clothes. Automatizing the human aesthetic decision is the key concept of the interdisciplinary field called Computational Aesthetics. In this thesis we address some of the existing open challenges, discussing how Computational Aesthetics and Deep Learning help in proposing solutions to cope with them.

6.1 Texel-based Texture Descriptor

In this thesis we present a new way to aesthetically describe textures, adopting attributes that focus on texels. The proposed framework, Texel-Att, can successfully describe and classify patterns that are not well-handled by the existing texture analysis approaches, as demonstrated by the experimental results, supported also by two new datasets, ElBa and E-DTD. In addition, Texel-Att is shown to be highly effective for image search, paving the way to fashion and graphic design applications. The current implementation has much room for improvement, being trained with few texel types (circles, lines, polygons) and 2D patterns with limited distortions. In fact, the modular design of the framework makes it easy to customize to handle different kinds of element-based textures, as it is just a matter of changing the detector and group different texel types, and changing the invariance properties of the layout attributes to handle larger distortions.

We show the effectiveness of Texel-Att in different applications, such as retrieval and interactive image search. For the first task, we prove that Texel-Att outperforms all the competitors in finding patterns inside large databases even under simulated real-

world factors such as poor resolution, noise and lighting conditions. Next, we show the potential of Texel-Att in an interactive image search system. We integrate it into the WhittleSearch framework, where we demonstrate through a user study its much higher performance in searching in a huge catalog of textures thanks to the relative attribute formulation. By describing texel related attributes we are able to perform very fine-grained searches that are simply unfeasible with existing texture attribute representations. This is of real importance in an industry where there are many products that can be similar at a first glance but different in the details (e.g. a dotted texture with small dots or big dots).

Texel-Att has many strengths, but it can certainly be improved in future works. More in detail, detections of texels suffers from a limited variety of shapes, currently defined by primitives like circle, line and polygons. A more flexible detector that is able to find repeated patterns not necessarily related to a specific primitive could enable the analysis of a wider variety of textures. Moreover, it is currently not able to deal with 3D deformation, since the layout of the pattern changes completely. Another point of potential improvement is the definition of the attributes. They are computed using well-known statistical measures, but they could also be learned from data to maximize the expressiveness of the attribute set.

6.2 Video-to-Shop Retrieval

One of the most important applications for e-fashion is that of retrieval. It enables users to search the desired garment inside a gallery starting from a taken photo or a recorded video (query item) in a fast way. This action replaces both the human process of manually scrolling each page of the website and the human decision in defining when the gallery item is aesthetically the same or a similar one to the one sought and desired. Moreover, this retrieval is useful also for industries: they could analyze social network pictures to find out which brand or which collection is trending, providing very valuable marketing data.

In this thesis, we introduce SEAM-Match-RCNN, trained on the new MovingFashion dataset, and show that video-to-shop matching can be performed on videos in the wild, such as TikToks, possibly unveiling fashion trends directly from social platforms and consequently attracting big fashion players. To fill the gap of publicly available benchmarks, we created MovingFashion, a dataset composed of 15K videos and their corresponding shop images, collected from both e-commerce websites and social networks.

SEAM Match-RCNN, which relies on an attention mechanism, achieves better results with respect to all the available competitors, displaying a natural improvement *wrt* the classical single-image setup. Interestingly, the attention modules' values seem to be in accord with the best practices in social media video editing, that is, that videos have to deliver their main message within approximately 6 seconds [44]. It is within the 2-6sec interval where the attention score gives the highest value for most of the videos

and it is within that same interval that the images most effective for recognition are contained. As a counter-proof, this interval identifies also the sequence frames which are most successful to match the shop image.

This setup can be attractive for many scenarios, for example:

1. a *casual user* can match a video snippet of a nice outfit he/she has captured with a gallery of products (e.g. Zalando, Amazon, etc.);

2. a *fast fashion company* can measure the similarity of clothing items contained in a viral video, or fashion show, with the items of its catalog, deciding which item to promote the most;

3. Youtube videos can be automatically processed by *video sharing platforms* to build valuable statistics of popular outfits and discover emerging trends;

In addition, this architecture can also be exploited for other tasks such as *attribute prediction*, extending it from an image-based problem to a video-based one, with potentially high-performance improvement of existing methods.

Failure cases in SEAM Match-RCNN arise when discriminant details are covered or missing in most of the video sequence. In those cases, self-attention doesn't properly focus on them as they are not consistently present in the sequence. Also, performance on complex textural patterns, such as writing, is low. Fine-grained details are ignored in favor of the general shape and color of that pattern. This causes failure when in the gallery there are multiple similar clothes with different prints on them.

6.3 New Fashion Product Performance Forecasting

Forecasting the performance of new clothing items is a crucial challenge for fashion companies. A good forecast in terms of predicted sales, or product popularity, carried out prior to the target season will be helpful in selecting the right amount of items to be put on the market, optimizing the entire supply chain. Unfortunately, standard forecasting approaches require information on the past performance to provide a prediction of the future and this information is available for evergreen products only, not for new ones.

The solutions for New Fashion Product Performance Forecasting challenge work on the logic that new products will perform comparably to older, aesthetically similar products (a solid yellow t-shirt put on the market from spring-summer 2019 will sell similarly to a solid yellow t-shirt sold in spring-summer 2018). This is how professionals work to estimate sales of new products.

In this thesis, we address the challenge of automatizing the New Fashion Product Performance Forecasting task, up to now made by professionals. We collect VISUELLE, a novel dataset build upon genuine data of an Italian fashion company, made of 5577 new products sold between 2016-2019. It is equipped with images of the products together with metadata and related sales. The dataset provides a new benchmark for the New Fashion Product Performance Forecasting challenge and in particular for

GTM-Transformer, a novel non-autoregressive transformer model based on the standard encoder-decoder architecture. The encoder works on the representation of exogenous time series, used to fill the missing past performance observations, whilst the decoder forecasts the sales using the encoded exogenous signals and the available visual and metadata information related to the new product, for which we want to estimate the performance.

To fill the missing past information, we adopt two different exogenous signals. The first is Google Trends, directly collected from the corresponding API and never used in practice in a new product forecasting setting. The second, dubbed "POtential Performance" (POP) signal, is made with a new interesting data-centric pipeline, based on capturing the aesthetical similarity of the probe image against fashionable and unfashionable images, uploaded on the web in the past. Metaphorically, the pipeline performs somewhat of a "time travel": it sends a fashion probe image in the past, before its launch in the market. This past is modeled by highly ranked web images, queried by using general textual tags related to the probe. The probe similarity with the past is then shown to be a good predictor for future performance.

We show that both Google Trends and POP signals are beneficial to forecasting and help augment the model's reasoning, even in presence of attributes that are automatically extracted from the raw product image. Moreover, the signals have been proved to be informative even in the case of challenging scenarios such as sales of 2021, where the CODIV pandemic introduced a hard to manage variable in forecasting sales. All of this was possible thanks to a multimodal framework based on the Transformer, made non-autoregressive in order to deal with the high dynamics that sales data exhibit, by effectively ingesting the exogenous data. In particular, POP signal outperforms Google Trends, demonstrating that focusing on how to collect and process data, instead of focusing on models is more important.

Moreover, POP has proved to be informative also for the Popularity Prediction of Fashion Styles task on the FashionForward benchmark.

Despite the capabilities demonstrated by several experiments, the New Fashion Product Performance Forecasting problem still leaves many doors open in the field of research. In particular, failure cases arise when the model is not able to deal with external variables such as human interventions in terms of discounts and rearrangements of stock. As future work, we plan to introduce such external factors in VISUELLE, also extending the architecture in order to manage them.

Other failure cases occur when trying to estimate the performance of products with rarely seen tags, since in a real scenario, new categories of garments may be introduced in the catalog at any time. In such a case, exploiting the semantic relation between different clothing categories could represent a possible solution, similarly to zero-shot or few-shot learning models.

Regarding exogenous signals, the principal limitations are due to the choice of the tag set. By querying the Google Trends API, the choice and the order of the tags is really important, since a wrong combination of them leads to the collection of not

meaningful or empty series. In this thesis, we have chosen to use the tags detailed in the technical sheet instead of extracting them automatically from images. As discussed in Sec. 5.5.4, many times, products are labeled confusingly, generating tags that are too general or unusable (e.g. "solid colours"). This issue causes the collection of signals or images that are not representative for our task. A possible solution is the use of a robust attribute predictor, extracting tags automatically following a meaningful taxonomy.

Bibliography

[1] M. A. Aftab, Q. Yuanjian, N. Kabir, and Z. Barua, "Super responsive supply chain: The case of spanish fast fashion retailer inditex-zara," *International Journal of Business and Management*, vol. 13, no. 5, 2018.

[2] N. Ahuja and S. Todorovic, "Extracting texels in 2.1 d natural textures," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[3] Z. Al-Halah and K. Grauman, "From paris to berlin: Discovering fashion style influences around the world," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 136–10 145.

[4] Z. Al-Halah, R. Stiefelhagen, and K. Grauman, "Fashion forward: Forecasting visual style in fashion," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 388–397.

[5] J. Aldrich, "Correlations genuine and spurious in pearson and yule," *Statistical science*, pp. 364–376, 1995.

[6] A. I. Anik and A. Bunt, "Data-centric explanations: Explaining training data of machine learning systems to promote transparency," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

[7] M. Arvan, B. Fahimnia, M. Reisi, and E. Siemsen, "Integrating human judgement into quantitative forecasting methods: A review," *Omega*, vol. 86, 2019.

[8] A. Baddeley, E. Rubak, and R. Turner, *Spatial point patterns: methodology and applications with R*. Chapman and Hall/CRC, 2015.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[10] P. F. Bangwayo-Skeete and R. W. Skeete, "Can google data improve the forecasting performance of tourist arrivals? mixed-data sampling approach," *Tourism Management*, vol. 46, 2015.

[11] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis, "Looking beyond appearances: Synthetic training data for deep cnns in re-identification," *Computer Vision and Image Understanding*, vol. 167, pp. 50–62, 2018.

[12] S. Beheshti-Kashi, H. R. Karimi, K.-D. Thoben, M. Lütjenband, and M. Teucke, "A survey on retail sales forecasting and prediction infashion markets," *Systems Science and Control Engineering*, vol. 3, 2015.

[13] A. Bhatti, H. Akram, H. M. Basit, A. U. Khan, S. M. Raza, and M. B. Naqvi, "E-commerce trends during covid-19 pandemic," *International Journal of Future Generation Communication and Networking*, vol. 13, no. 2, pp. 1449–1452, 2020.

[14] R. Bormann, D. Esslinger, D. Hundsdoerfer, M. Haegele, and M. Vincze, "Robotics domain attributes database (rdad)," 2016, http://wiki.ros.org/ipa_texture_classification.

[15] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, "Apparel classification with style," in *Asian conference on computer vision*. Springer, 2012, pp. 321–335.

[16] G. Box, G. Jenkins, G. Reinsel, and G. Ljung, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[17] R. G. Brown, *Smoothing, forecasting and prediction of discrete time series*. Courier Corporation, 2004.

[18] L. Bulut, "Google trends and the forecasting performance of exchange rate models," *Journal of Forecasting*, vol. 37, no. 3, 2018.

[19] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1597–1604.

[20] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 792–803.

[21] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2015, pp. 1431–1439. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.2015.168

[22] W.-H. Cheng, S. Song, C.-Y. Chen, S. C. Hidayati, and J. Liu, "Fashion meets computer vision: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–41, 2021.

[23] Z.-Q. Cheng, X. Wu, Y. Liu, and X.-S. Hua, "Video2shop: Exact matching clothes in videos to online shopping images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4048–4056.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[25] T.-M. Choi, C.-L. Hui, and Y. Yu, *Intelligent Fashion Forecasting Systems: Models and Applications*. Springer, 2013.

[26] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606–3613.

[27] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3828–3836.

[28] J. C. de Winter, S. D. Gosling, and J. Potter, "Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data." *Psychological methods*, vol. 21, no. 3, p. 273, 2016.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[31] P. J. Diggle *et al.*, *Statistical analysis of spatial point patterns*. Academic press, 1983.

[32] X. Dong, X. Song, F. Feng, P. Jing, X.-S. Xu, and L. Nie, "Personalized capsule wardrobe creation with garment and user modeling," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 302–310.

[33] K. L. Donohue, "Efficient supply contracts for fashion goods with forecast updating and two production modes," *Management science*, vol. 46, no. 11, 2000.

[34] E. Dopson, "Videos vs. images: Which drives more engagement in facebook ads?" 2020, available online: https://databox.com/videos-vs-images-in-facebook-ads [Accessed: 10 November 2020].

[35] R. Duffett, "The youtube marketing communication effect on cognitive, affective and behavioural attitudes among generation z consumers," *Sustainability*, vol. 12, no. 12, p. 5075, 2020.

[36] V. Ekambaram, K. Manglik, S. Mukherjee, S. S. K. Sajja, S. Dwivedi, and V. Raykar, "Attention based multi-modal new product sales time-series forecasting," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3110–3118.

[37] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 2005, pp. 1816–1823 Vol. 2.

[38] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *The Annals of Statistics*, vol. 29, no. 5, 2001. [Online]. Available: https://doi.org/10.1214/aos/1013203451

[39] J. Gao and R. Nevatia, "Revisiting temporal modeling for video-based person reid," *arXiv preprint arXiv:1805.02104*, 2018.

[40] C. C. Garcia, "Fashion forecasting: an overview from material culture to industry," *Journal of Fashion Marketing and Management: An International Journal*, 2021.

[41] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Advances in neural information processing systems*, 2015, pp. 262–270.

[42] Y. Ge, R. Zhang, X. Wang, X. Tang, and P. Luo, "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5337–5345.

[43] M. Godi, C. Joppi, A. Giachetti, and M. Cristani, "Simco: Similarity-based object counting," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 47–52.

[44] M. Gollin, "Facebook video ads: Best practices for 2019," *Falcon. io*, 2018.

[45] J. Gu, J. Bradbury, C. Xiong, V. O. Li, and R. Socher, "Non-autoregressive neural machine translation," in *International Conference on Learning Representations*, 2018.

[46] Y. Gui, M. Chen, L. Ma, and Z. Chen, "Texel based regular and near-regular texture characterization," in *2011 International Conference on Multimedia and Signal Processing*, vol. 1. IEEE, 2011, pp. 266–270.

[47] G. Guzman, "Internet search behavior as an economic forecasting tool: The case of inflation expectations," *Journal of economic and social measurement*, vol. 36, no. 3, pp. 119–167, 2011.

[48] M. Hadi Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, "Where to buy it: Matching street clothing photos in online shops," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3343–3351.

[49] A. Hamid and M. Heiden, "Forecasting volatility with empirical similarity and google trends," *Journal of Economic Behavior & Organization*, vol. 117, 2015.

[50] X. Han, Z. Wu, P. X. Huang, X. Zhang, M. Zhu, Y. Li, Y. Zhao, and L. S. Davis, "Automatic spatially-aware fashion concept discovery," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1463–1471.

[51] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "Viton: An image-based virtual try-on network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7543–7552.

[52] C. Hand and G. Judge, "Searching for the picture: forecasting uk cinema admissions using google trends data," *Applied Economics Letters*, vol. 19, no. 11, pp. 1051–1055, 2012.

[53] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[55] J. Henzel and M. Sikora, "Gradient boosting application in forecasting of performance indicators values for measuring the efficiency of promotions in fmcg retail," 2020.

[56] I. Hersey, "Textures: A photographic album for artists and designers by phil brodatz," *Leonardo*, vol. 1, no. 1, pp. 91–92, 1968.

[57] W.-L. Hsiao and K. Grauman, "Learning the latent" look": Unsupervised discovery of a style-coherent embedding from fashion images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4203–4212.

[58] W. Hsiao and K. Grauman, "Creating capsule wardrobes from fashion images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7161–7170.

[59] J. Huang, R. S. Feris, Q. Chen, and S. Yan, "Cross-domain image retrieval with a dual attribute-aware ranking network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1062–1070.

[60] R. Hyndman, A. B. Koehler, J. K. Ord, and R. D. Snyder, *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[61] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2021.

[62] T. Ijiri, R. Mech, T. Igarashi, and G. Miller, "An example-based procedural system for element arrangement," in *Computer Graphics Forum*, vol. 27. Wiley Online Library, 2008, pp. 429–436.

[63] I. Ilic, B. Görgülü, M. Cevik, and M. G. Baydoğan, "Explainable boosted linear regression for time series forecasting," *Pattern Recognition*, p. 108144, 2021.

[64] J. Illian, A. Penttinen, H. Stoyan, and D. Stoyan, *Statistical analysis and modelling of spatial point patterns*. John Wiley & Sons, 2008, vol. 70.

[65] T. Iwata, S. Watanabe, and H. Sawada, "Fashion coordinates recommender system using photographs from fashion magazines," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[66] Y. Jeon, S. Jin, B. Kim, and K. Han, "Fashionq: An interactive tool for analyzing fashion style trend with quantitative criteria," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.

[67] M. Jia, M. Shi, M. Sirotenko, Y. Cui, C. Cardie, B. Hariharan, H. Adam, and S. Belongie, "Fashionpedia: Ontology, segmentation, and an attribute localization dataset," in *European conference on computer vision*. Springer, 2020, pp. 316–332.

[68] Y. Jing and S. Baluja, "Visualrank: Applying pagerank to large-scale image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1877–1890, 2008.

[69] Y. Kalantidis, L. Kennedy, and L.-J. Li, "Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos," in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, 2013, pp. 105–112.

[70] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg, "Hipster wars: Discovering elements of fashion styles," in *European conference on computer vision*. Springer, 2014, pp. 472–488.

[71] M. E. Koponen, "How to create engaging mobile-optimised video ads for social media." 2020.

[72] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Image search with relative attribute feedback," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2973–2980.

[73] Z. Kuang, Y. Gao, G. Li, P. Luo, Y. Chen, L. Lin, and W. Zhang, "Fashion retrieval via graph reasoning networks on a similarity pyramid," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3066–3075.

[74] D. Kwiatkowski, P. C. Phillips, P. Schmidt, and Y. Shin, "Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?" *Journal of Econometrics*, vol. 54, no. 1, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/030440769290104Y

[75] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *International Journal of Neural Systems*, vol. 31, no. 03, p. 2130001, 2021.

[76] T. Leung and J. Malik, "Representing and recognizing the visual appearance of materials using three-dimensional textons," *International journal of computer vision*, vol. 43, no. 1, pp. 29–44, 2001.

[77] J. Li, Y. Song, J. Zhu, L. Cheng, Y. Su, L. Ye, P. Yuan, and S. Han, "Learning from large-scale noisy web data with ubiquitous reweighting for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1808–1814, 2021.

[78] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 369–378.

[79] Z. Li, Y. Li, W. Tian, Y. Pang, and Y. Liu, "Cross-scenario clothing retrieval and fine-grained style recognition," in *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2016, pp. 2912–2917.

[80] C.-T. Liu, C.-W. Wu, Y.-C. F. Wang, and S.-Y. Chien, "Spatially and temporally efficient non-local attention network for video-based person re-identification," in *British Machine Vision Conference*, 2019.

[81] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikainen, "A survey of recent advances in texture representation," *arXiv preprint arXiv:1801.10324*, vol. 3, 2018.

[82] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "From bow to cnn: Two decades of texture representation for texture classification," *International Journal of Computer Vision*, vol. 127, no. 1, pp. 74–109, 2019.

[83] L. Liu, J. Chen, P. W. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "A survey of recent advances in texture representation," *CoRR*, vol. abs/1801.10324, 2018. [Online]. Available: http://arxiv.org/abs/1801.10324

[84] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *Proceedings of the 20th ACM international conference on Multimedia*, 2012, pp. 619–628.

[85] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3330–3337.

[86] S. Liu, T.-T. Ng, K. Sunkavalli, M. N. Do, E. Shechtman, and N. Carr, "Patchmatch-based automatic lattice detection for near-regular textures," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 181–189.

[87] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1096–1104.

[88] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang, "Fashion landmark detection in the wild," in *European Conference on Computer Vision*. Springer, 2016, pp. 229–245.

[89] L. Lo, C. Liu, R. Lin, B. Wu, H. Shuai, and W. Cheng, "Dressing for Attention: Outfit Based Fashion Popularity Prediction," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, iSSN: 2381-8549.

[90] H. Loi, T. Hurtut, R. Vergne, and J. Thollot, "Programmable 2d arrangements for element texture design," *ACM Trans. Graph.*, vol. 36, no. 4, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3072959.2983617

[91] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[92] C. Ma, L.-Y. Wei, S. Lefebvre, and X. Tong, "Dynamic element textures," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 90:1–90:10, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2461912.2461921

[93] C. Ma, L.-Y. Wei, and X. Tong, "Discrete element textures," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 62:1–62:10, Jul. 2011. [Online]. Available: http://doi.acm.org/10.1145/2010324.1964957

[94] Y. Ma, Y. Ding, X. Yang, L. Liao, W. K. Wong, and T.-S. Chua, "Knowledge Enhanced Neural Fashion Trend Forecasting," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*. Dublin Ireland: ACM, Jun. 2020, pp. 82–90. [Online]. Available: https://dl.acm.org/doi/10.1145/3372278.3390677

[95] Y. Ma, X. Yang, L. Liao, Y. Cao, and T.-S. Chua, "Who, where, and what to wear? extracting fashion knowledge from social media," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 257–265.

[96] Z. Ma, J. Dong, Z. Long, Y. Zhang, Y. He, H. Xue, and S. Ji, "Fine-grained fashion similarity learning by attribute-specific embedding network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 741–11 748.

[97] T. Maenpaa, "The local binary pattern approach to texture analysis: Extensions and applications." 2004.

[98] P. Mallikarjuna, M. Fritz, A. T. Targhi, E. Hayman, B. Caputo, and J. Eklundh, "The kth-tips and kth-tips2 databases," 2006.

[99] B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 6, pp. 703–715, 2001.

[100] MATLAB, *version R2019a*. Natick, Massachusetts: The MathWorks Inc., 2019.

[101] T. Matthews, M. S. Nixon, and M. Niranjan, "Enriching texture analysis with semantic data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1248–1255.

[102] M. C. Medeiros and H. F. Pires, "The proper use of google trends in forecasting models," 2021.

[103] M. Motamedi, N. Sakharnykh, and T. Kaldewey, "A data-centric approach for training deep neural networks with less data," *arXiv preprint arXiv:2110.03613*, 2021.

[104] S. Nahmias and Y. Cheng, *Production and operations analysis*. McGraw-hill New York, 2009, vol. 6.

[105] A. Neuberger, E. Borenstein, B. Hilleli, E. Oks, and S. Alpert, "Image based virtual try-on network from unpaired data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5184–5193.

[106] L. Neumann, M. Sbert, B. Gooch, W. Purgathofer *et al.*, "Defining computational aesthetics," *Computational aesthetics in graphics, visualization and imaging*, pp. 13–18, 2005.

[107] A. Ng, "A chat with andrew on mlops: From model-centric to data-centric ai," https://www.youtube.com/watch?v=06-AZXmwHjo, May 2021.

[108] J. Ni, J. Li, and J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 188–197. [Online]. Available: https://www.aclweb.org/anthology/D19-1018

[109] M. Niknam and C. Kemke, "Modeling shapes and graphics concepts in an ontology." in *SHAPES*, 2011.

[110] P. S. Nitse, K. R. Parker, D. Krumwiede, and T. Ottaway, "The impact of color in the e-commerce marketing of fashions: an exploratory study," *European Journal of Marketing*, 2004.

[111] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, 2021.

[112] C. G. Northcutt, M. ChipBrain, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *stat*, vol. 1050, 2021.

[113] Oberlo, "10 social media statistics that you need to know in 2020," 2020.

[114] ——, "10 tiktok statistics that you need to know in 2020," 2020.

[115] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[116] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[117] D. Parikh and K. Grauman, "Relative attributes," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 503–510.

[118] J. Peeples, W. Xu, and A. Zare, "Histogram layers for texture analysis," *arXiv preprint arXiv:2001.00215*, 2020.

[119] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[120] A. Porrello, L. Bergamini, and S. Calderara, "Robust re-identification by multiple views knowledge distillation," in *The European Conference on Computer Vision (ECCV)*, 2020.

[121] J. Rae and A. Razavi, "Do transformers need deep long-range memory?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.672

[122] S. Ren, H.-L. Chan, and P. Ram, "A comparative study on fashion demand forecasting models with multiple sources of uncertainty," *Annals of Operations Research*, vol. 257, no. 1, 2017.

[123] S. Ren, H.-L. Chan, and T. Siqin, "Demand forecasting in retail operations for fashionable products: methods, practices, and real case study," *Annals of Operations Research*, vol. 291, no. 1, 2020.

[124] F. Setti, D. Conigliaro, M. Tobanelli, and M. Cristani, "Count on me: learning to count on a single image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1798–1806, 2017.

[125] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009.

[126] N. Shazeer and M. Stern, "Adafactor: Adaptive learning rates with sublinear memory cost," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018. [Online]. Available: https://proceedings.mlr.press/v80/shazeer18a.html

[127] Y. Shen, T. Xiao, S. Yi, D. Chen, X. Wang, and H. Li, "Person re-identification with deep kronecker-product matching and group-shuffling random walk," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[128] E. S. Silva, H. Hassani, D. Ø. Madsen, and L. Gee, "Googling fashion: forecasting fashion consumer behaviour using google trends," *Social Sciences*, vol. 8, no. 4, p. 111, 2019.

[129] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun, "A high performance crf model for clothes parsing," in *Asian conference on computer vision*. Springer, 2014, pp. 64–81.

[130] E. Simo-Serra, S. Fidler, F. MorenoNoguer, and R. Urtasun, "Neuroaesthetics in fashion: Modeling the perception of fashionability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 869–877.

[131] E. Simo-Serra and H. Ishikawa, "Fashion style in 128 floats: Joint ranking and classification using weak data for feature extraction," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 298–307.

[132] P. K. Singh, Y. Gupta, N. Jha, and A. Rajan, "Fashion Retail: Forecasting Demand for New Items," *arXiv:1907.01960 [cs]*, Jun. 2019. [Online]. Available: http://arxiv.org/abs/1907.01960

[133] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 1349–1380, 2000.

[134] K. Smith, "57 fascinating and incredible youtube statistics," p. 39, 2018.

[135] H. Song, M. Kim, and J.-G. Lee, "Selfie: Refurbishing unclean samples for robust deep learning," in *International Conference on Machine Learning*. PMLR, 2019.

[136] X. Song, F. Feng, J. Liu, Z. Li, L. Nie, and J. Ma, "Neurostylist: Neural compatibility modeling for clothing matching," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 753–761.

[137] X. Song, X. Han, Y. Li, J. Chen, X.-S. Xu, and L. Nie, "Gp-bpr: Personalized compatibility modeling for clothing matching," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 320–328.

[138] R. Sorger and J. Udale, *The fundamentals of fashion design*. Bloomsbury Publishing, 2017.

[139] H. Su, J. Deng, and L. Fei-Fei, "Crowdsourcing annotations for visual object detection," in *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[140] A. Subramaniam, A. Nambiar, and A. Mittal, "Co-segmentation inspired attention networks for video-based person re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[141] M. Takagi, E. Simo-Serra, S. Iizuka, and H. Ishikawa, "What makes a style: Experimental analysis of fashion prediction," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 2247–2253.

[142] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, man, and cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[143] S. Thomassey, "Sales forecasting in apparel and fashion industry: A review," *Intelligent fashion forecasting systems: Models and applications*, 2014.

[144] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 969–977.

[145] M. Tuceryan and A. K. Jain, "Texture analysis," in *Handbook of pattern recognition and computer vision*. World Scientific, 1993, pp. 235–276.

[146] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1512–1523, 2009.

[147] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[148] E. Velázquez, I. Martínez, S. Getzin, K. A. Moloney, and T. Wiegand, "An evaluation of the state of spatial point pattern analysis in ecology," *Ecography*, vol. 39, no. 11, pp. 1042–1055, 2016.

[149] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward characteristic-preserving image-based virtual try-on network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 589–604.

[150] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu, "Attentive fashion grammar network for fashion landmark detection and clothing category classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4271–4280.

[151] X. Wang and T. Zhang, "Clothes search in consumer photos via color matching and attribute learning," in *Proceedings of the 19th ACM international conference on Multimedia*, 2011, pp. 1353–1356.

[152] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[153] X. Wang, B. Wu, and Y. Zhong, "Outfit compatibility prediction and diagnosis with multi-layered comparison network," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 329–337.

[154] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[155] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, "Fashion iq: A new dataset towards retrieving images by natural language feedback," 2020.

[156] L. Wu and E. Brynjolfsson, "3. the future of prediction: How google searches foreshadow housing prices and sales," in *Economic analysis of the digital economy*. University of Chicago Press, 2015.

[157] Y. Wu, O. E. F. Bourahla, X. Li, F. Wu, Q. Tian, and X. Zhou, "Adaptive graph representation learning for video person re-identification," *IEEE Transactions on Image Processing*, vol. 29, pp. 8821–8830, 2020.

[158] Y. Xu, H. Ji, and C. Fermüller, "Viewpoint invariant texture description using fractal analysis," *International Journal of Computer Vision*, vol. 83, no. 1, pp. 85–100, 2009.

[159] H. Xuan, A. Stylianou, and R. Pless, "Improved embeddings with easy positive triplet mining," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 2474–2482.

[160] J. Xue, H. Zhang, and K. Dana, "Deep texture manifold for ground terrain recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 558–567.

[161] K. Yamaguchi, T. L. Berg, and L. E. Ortiz, "Chic or social: Visual popularity analysis in online fashion networks," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 773–776.

[162] K. Yamaguchi, M. Hadi Kiapour, and T. L. Berg, "Paper doll parsing: Retrieving similar styles to parse clothing items," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3519–3526.

[163] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg, "Parsing clothing in fashion photographs," in *2012 IEEE Conference on Computer vision and pattern recognition*. IEEE, 2012, pp. 3570–3577.

[164] Y. Yan, J. Qin, J. Chen, L. Liu, F. Zhu, Y. Tai, and L. Shao, "Learning multi-granular hypergraphs for video-based person re-identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[165] W. Yang, P. Luo, and L. Lin, "Clothing co-parsing by joint image segmentation and labeling," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3182–3189.

[166] H. Zhang, J. Xue, and K. Dana, "Deep ten: Texture encoding network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 708–717.

[167] H. Zhao, J. Yu, Y. Li, D. Wang, J. Liu, H. Yang, and F. Wu, "Dress like an internet celebrity: Fashion retrieval in videos," in *proceedings of the International Joint Conferences on Artificial Intelligence*, 07 2020, pp. 1054–1060.

[168] P. Zhao and L. Quan, "Translation symmetry detection in a fronto-parallel view," in *CVPR 2011*. IEEE, 2011, pp. 1009–1016.

[169] S. Zheng, F. Yang, M. H. Kiapour, and R. Piramuthu, "Modanet: A large-scale street fashion dataset with polygon annotations," in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1670–1678.

[170] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy, "Be your own prada: Fashion synthesis with structural coherence," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1680–1688.