

Studio2Shop: From Studio Photo Shoots to Fashion Articles

Julia Lasserre*, Katharina Rasch* and Roland Vollgraf

Zalando Research, Muehlenstr. 25, 10243 Berlin, Germany

Keywords: Computer Vision, Deep Learning, Fashion, Item Recognition, Street-to-shop.

Abstract: Fashion is an increasingly important topic in computer vision, in particular the so-called *street-to-shop* task of matching street images with shop images containing similar fashion items. Solving this problem promises new means of making fashion searchable and helping shoppers find the articles they are looking for. This paper focuses on finding pieces of clothing worn by a person in full-body or half-body images with neutral backgrounds. Such images are ubiquitous on the web and in fashion blogs, and are typically studio photos, we refer to this setting as *studio-to-shop*. Recent advances in computational fashion include the development of domain-specific numerical representations. Our model Studio2Shop builds on top of such representations and uses a deep convolutional network trained to match a query image to the numerical feature vectors of all the articles annotated in this image. Top- k retrieval evaluation on test query images shows that the correct items are most often found within a range that is sufficiently small for building realistic visual search engines for the studio-to-shop setting.

1 INTRODUCTION

Online fashion is a fast growing field which generates massive amounts of (largely unexplored) data. The last five years have seen an increasing number of academic studies specific to fashion in top conferences, and particularly in the computer vision community, with topics as varied as article tagging (Chen et al., 2012; Bossard et al., 2013; Chen et al., 2015), clothing parsing (Wang and Haizhou, 2011; Yamaguchi et al., 2012; Dong et al., 2013; Yamaguchi et al., 2013; Liu et al., 2014), article recognition (Wang and Zhang, 2011; Fu et al., 2013; Liu et al., 2012b; Kalantidis et al., 2013; Huang et al., 2015; Liu et al., 2016b), style recommenders (magic mirrors or closets) (Liu et al., 2012a; Di et al., 2013; Kiapour et al., 2014; Jagadeesh et al., 2014; Vittayakorn et al., 2015; Yamaguchi et al., 2015), and fashion-specific feature representations (Bracher et al., 2016; Simo-Serra and Ishikawa, 2016). Our company Zalando is Europe’s leading online fashion platform and, like other big players on the market, benefits from large datasets and has a strong interest in taking part in this effort.¹

Street-to-shop (Liu et al., 2012b) is the task of retrieving articles from a given assortment that are similar to articles in a query picture from an unknown

*These authors contributed equally.

¹This paper is best viewed in colour.

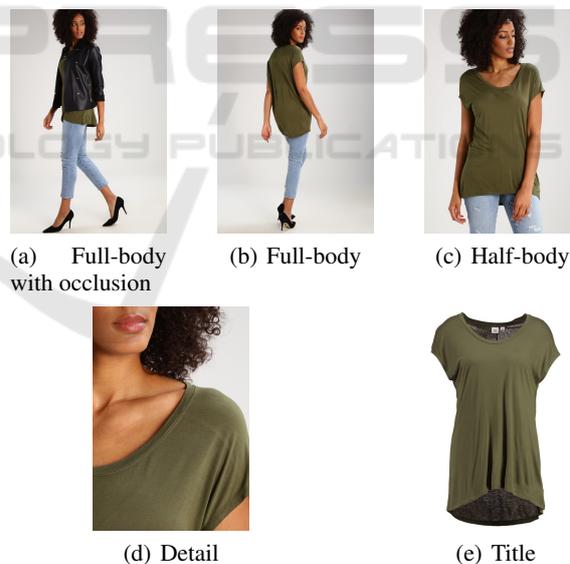


Figure 1: Examples of images in our dataset. Image types (a-d) are query images featuring models, image type (e) represents the articles we retrieve from.

source (a photo taken on the street, a selfie, a professional photo). In this study, we follow the *exact-street-to-shop* variant (Kiapour et al., 2015), where query images are related to the assortment and therefore the exact-matching article should be retrieved. In particular, we focus on a setting we call *studio-to-shop* where the query picture is a photo shoot image of a model

Table 1: Overview of street-to-shop studies. CV stands for classical computer vision, CNN for convolutional neural network.

dataset	focus	domain 1			domain 2			method
		background	assumptions	focus	background	assumptions		
Street-To-Shop (Liu et al., 2012b)	Street-To-Shop	half + full	yes	body-part detector upper vs lower	half + full	no	body-part detector upper vs lower	CV
Where-To-Buy-It (Kiapour et al., 2015)	Exact-Street-To-Shop	half + full	yes	category of item bounding box	half + full + title	no		CV
DARN (Huang et al., 2015)	DARN dataset	half, frontal view	yes	upper clothing frontal view clothing detector to crop	half + title, frontal view	yes+no	upper clothing frontal view clothing detector to crop	siamese CNN
Wang et al (Wang et al., 2016)	Exact-Street-To-Shop AliBaba	half + full	yes	none	half + full + title	no	none	siamese CNN shared lower layers
DeepFashion (Liu et al., 2016a)	In-Shop	half + full + title	no	landmark (training)	<i>same as domain 1</i>			single CNN
DeepFashion (Liu et al., 2016a)	Consumer-To-Shop	half + full + title	yes	landmark (training)	<i>same as domain 1</i>			single CNN
Studio2Shop	Zalando	half + full	no	none	title	no	none	siamese CNN second input as features

wearing one or several fashion items in a controlled setting with a neutral background, and where the target is an article of our assortment. Such images are ubiquitous on the web, for example in fashion magazines or on online shopping websites, and solving our task would allow our customers to search for products more easily. In addition, this setting can have many internal applications such as helping trend scouts match blog images with our products, or annotating all our catalogue images with all contained articles automatically.

Recognising fashion articles is a challenging task. Clothing items are not rigid and usually undergo strong deformations in images, they may also be partially occluded. They vary in most of their physical attributes (for example colour, texture, pattern), even within the same category, and can contain details of importance such as a small logo, so that recovering a few basic attributes may not always be enough. Our query images contain a wide variety of positions and views, with full-body and half-body model images, as well as images of details (see Figures 1(a-d)).

In the literature, street-to-shop is typically approached as an isolated problem, taking pairs of query/target images together with optional article meta-data as input. So far, both query and target inputs have had the same form, namely a feature vector or an image. By putting forward our model Studio2Shop, we propose instead to build on existing feature representations of fashion articles, and to use images for queries and static feature vectors for targets. Indeed, many online shops, including Zalando, already have a well-tuned feature representation of their articles that can be used across a wide array of applications, including recommender systems and visual search. Moreover, such features are also publicly available (e.g. AlexNet’s fc6 (Krizhevsky et al., 2012), VGG16’s fc14 (Simonyan and Zisserman, 2014), fashion in 128 floats (Simo-Serra and Ishikawa, 2016)).

Breaking the symmetry between query and target allows us to use more complete feature representa-

tions, since static features have usually been trained on massive datasets. The representation we use in this study was trained on hundreds of thousands of articles, including categories of articles that are not directly relevant to our task, and on more attributes than we could process with an end-to-end framework. Note that only title images were used during training, i.e. images of the article without any background as shown in Figure 1(e), and no model images were seen.

The contribution of our work is three-fold:

- We naturally handle all categories at hand and make no assumptions about the format of the model image (full body, half body, detail), nor do we require additional information such as the category, a bounding box or landmarks.
- We show that in our setting, end-to-end approaches are not necessary, and that using a static feature representation for the target side is effective, especially when combining it with a non-linear query-article matching model.
- We show that we can achieve reasonable results on an extra (publicly available) dataset with similar properties, without even fine-tuning our model.

The remainder of this paper is organised as follows: we review the related work in Section 2, describe our dataset and approach in detail in Sections 3 and 4, and evaluate our approach in Section 5.

2 RELATED WORK

Data. The street-to-shop task was defined in (Liu et al., 2012b) and formulated as a domain transfer problem. Since then, many groups have contributed their own dataset with their own set of assumptions. There is a lot of variation in the literature regarding types of street (query) and shop (target) images. In order to reduce background, street images are often assumed to be cropped around the article (Kiapour et al., 2015;

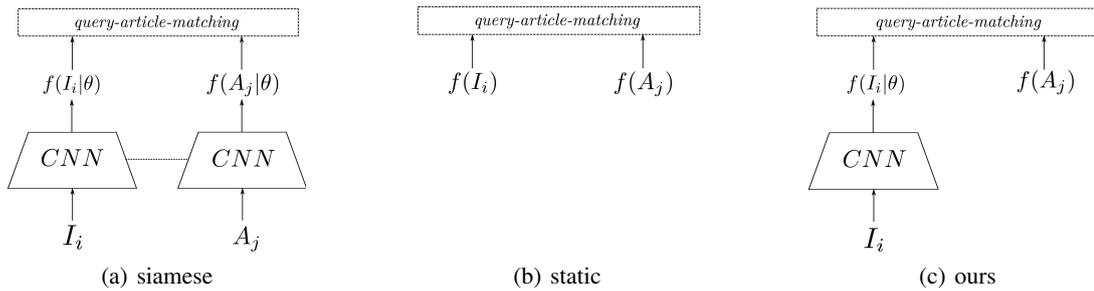


Figure 2: Variations of street-to-shop architectures in the literature. θ represents the joint set of parameters of the two legs. (a) The most common architecture in recent studies, based on image pairs. (b) Only static features, as found in (Kiapour et al., 2015). (c) Our approach.

Huang et al., 2015) and sometimes come with the information of the category of the article (Kiapour et al., 2015). Shop images vary across and within datasets, often mixing full-body model images, half-body model images and title images. In some works, shop images resemble high quality street images (Huang et al., 2015; Liu et al., 2016a). We classify the state-of-the-art in Table 1 following the types of images used in both domains according to three factors: *focus* of the image (full-body model image, half-body model image and title image), *background* (yes/no), and the assumptions made. Note that images from domain 2 are typically professional, while images from domain 1 that contain background can be amateur shots.

Approaches. Early works on street-to-shop or more generally article retrieval (Wang and Zhang, 2011; Fu et al., 2013; Liu et al., 2012b; Kalantidis et al., 2013; Yamaguchi et al., 2013) are based on classical computer vision: body part detection and/or segmentation, and hand-crafted features. For example, (Liu et al., 2012b) locate 20 upper and 10 lower body parts and extract HOG (Dalal and Triggs, 2005), LBP (Ojala et al., 2002) and colour features for each body part, while (Yamaguchi et al., 2013) use clothes parsing to segment their images. In recent years, attention in the fashion-recognition domain has shifted towards using deep learning methods. Multiple recent studies follow a similar paradigm: the feature representation of both types of images is learnt via attribute classification, while the domain transfer is performed via a ranking loss which pushes matching pairs to have higher scores than non-matching pairs (Huang et al., 2015; Wang et al., 2016; Liu et al., 2016a; Simo-Serra and Ishikawa, 2016). In this context, siamese architectures seem quite natural and have given promising results (Huang et al., 2015; Wang et al., 2016). When no domain transfer is needed, *i.e.* when query images and assortment images are of the same kind, a single network for the two branches performs just as well (Liu et al., 2016a).

3 DATA

Images. Our dataset contains images of articles from the categories "dress", "jacket", "pullover", "shirt", "skirt", "t-shirt/top" and "trouser" that were sold roughly between 2012 and 2016. The distribution of categories is 35% t-shirts/tops, 20% trousers, 14% pullovers, 14% dresses, 12% shirts, 3% skirts and 2% jackets. Note that our data is restricted to the categories aforementioned not because of limitations of our approach, but because the selected categories contain the most samples and are likely to be visible. As an example, most of our shoes and socks have no dedicated model images and socks are not visible on other model images.

We have approximately 1.15 million query images in size 224x155 pixels with neutral backgrounds of the type shown in Figures 1(a-d). About 250,000 of those are annotated with several articles, while the others are annotated with one article only, even if several articles can be seen. In addition to these query images, our dataset contains approximately 300,000 title images in size 224x155 pixels (see Figure 1(e)). These 300,000 images represent the assortment we want to retrieve from.

FashionDNA. FashionDNA (fDNA) is Zalando's feature representation for fashion articles. These features are obtained by extracting the activations of a hidden fully connected layer in a static deep convolutional neural network that was trained using title images and article attributes. In our case, these activations are of size 1536, which we reduce to 128 using PCA to decrease the number of parameters in our model. fDNA is not within the scope of this study, but full details about the architecture of the network and the attributes used are given in Appendix.

Because our article features are based on title images, we are slightly different to the other papers which allow model images in their shop images. Our task involves complete domain transfer, while their

Table 2: Variations of the main architecture, the features used, and the matching method in our evaluation.

	global architecture	features	query-article-matching	loss	inspiration
static-fc14-linear	static	fc14	linear	none	
static-fc14-non-linear	static	fc14	non-linear	cross-entropy	(Kiapour et al., 2015)
fc14-non-linear	ours	fc14	non-linear	cross-entropy	
fDNA-linear	ours	fDNA	linear	cross-entropy	
fDNA-ranking-loss	ours	fDNA	linear	triplet ranking	
all-in-two-nets	siamese	learnt	linear	triplet ranking + attributes cross-entropy	(Huang et al., 2015; ?)
Studio2Shop	ours	fDNA	non-linear	cross-entropy	

task is more related to image similarity, especially in the case of (Liu et al., 2016a). This makes comparisons rather difficult. Nevertheless, it was possible to isolate title images in two external datasets which we used as additional test sets.

Attributes. The siamese method implemented in this study requires article attributes. We used category (7 values, available for 100% of articles), main colour (82 values, 100% coverage), pattern (19 values, 53% coverage), clothing length (12 values, 50% coverage), sleeve length (9 values, 32% coverage), shirt collar (27 values, 30% coverage), neckline (12 values, 23% coverage), material construction (14 values, 20% coverage), trouser rise (3 values, 14% coverage).

4 MATCHING STUDIO PHOTOS TO FASHION ARTICLES OF A GIVEN ASSORTMENT

4.1 Studio2Shop

Building on Existing Feature Representations. Most recent deep learning approaches to street-to-shop (Huang et al., 2015; Wang et al., 2016; Liu et al., 2016a) can be summarised by Figure 2a. In a forward pass, the left leg of the network processes query images I_i (query-feature module), the right leg processes target images A_j (target-feature module), and a query-article-matching submodel matches the two feature vectors produced. Both inputs are images, and the two legs are trained with varying amounts of weights shared between them (0% sharing for (Huang et al., 2015), 100% for (Liu et al., 2016a)). On the other end of the spectrum, (Kiapour et al., 2015) use static features on both sides, as sketched in Figure 2b.

In contrast, Studio2Shop builds on top of existing feature representations of fashion articles, for example fDNA. This means that the right leg of the network, as shown in Figure 2c, takes as input the pre-

computed features of the articles given their title images.

Non-linear Matching of Query Features to Article Features. Most street-to-shop methods relying on an architecture of the type shown in Figure 2a use a triplet ranking loss directly on the feature vectors produced (Huang et al., 2015; Wang et al., 2016; Liu et al., 2016a).

In contrast, Studio2Shop follows (Kiapour et al., 2015) and uses a more sophisticated submodel. We concatenate the two feature vectors, add on top a batch normalisation layer followed by two fully connected layers with 256 nodes and ReLU activations and one logistic regression layer. We have not come across this combination of a feature learning module and a deep matching module in the literature.

Details of the Query-feature Submodel. In the left leg of our network, for the CNN denoted in Figure 2c, we use as a basis the publicly available VGG16 (Simonyan and Zisserman, 2014) pre-trained on ImageNet ILSVRC-2014 (Russakovsky et al., 2015), but only keep the 13 convolutional layers. On top of these layers we add two fully connected layers with 2048 nodes and ReLU activations, each followed by a 50% dropout. We finally add a fully connected layer of size 128 which outputs the fDNA of the input query image. All details are given in Appendix.

Backward Pass. Approaches of the type shown in Figure 2a (Huang et al., 2015; Wang et al., 2016; Liu et al., 2016a) use article meta-data or attributes on each side of the two-legged network. θ denotes the joint set of parameters of the two legs. On top of the layer generating the feature vector $f(\cdot|\theta)$ sits, for each attribute, a sub-network ending with softmax activations. A categorical cross-entropy loss is used for each attribute-specific branch. Finally a triplet ranking loss joins the two feature representations.

In contrast, we disregard attributes and use a cross-entropy loss for our query-article-matching submodel. The loss is given in Equation 1, where N is

the number of query images, M the number of articles, p_{ij} the output of the model given the input query image I_i and target article A_j , and y_{ij} is the ground truth label: 1 if the target article A_j belongs to the query model image I_i , 0 otherwise. Note that the loss is back-propagated to all layers of the network, even to layers that were pretrained on ImageNet.

$$\mathcal{L} = \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log p_{ij} + (1 - y_{ij}) \log (1 - p_{ij}) \quad (1)$$

For practical reasons, we split our query images in mini-batches of size 64, and compare them with 50 articles only. These 50 articles contain the (at least 1) annotated matching articles and (at most 49) articles drawn randomly without replacement and temporarily labelled as negative matches. The negative articles are constantly resampled (for each query image in the batch, for each batch, and for each epoch), which enforces diversity. Our images are not fully annotated and some of the data supplied might be erroneously labelled as negative, however with so many articles available and so few articles actually present in query images, the probability that such an incorrect negative label occurs is relatively low. The loss over a mini-batch is shown in Equation 2, where $c(i)$ is the i^{th} query image in the batch, and $c(i, j)$ the j^{th} article for this image in the batch.

$$\mathcal{L} = \sum_{i=1}^{64} \sum_{j=1}^{50} y_{c(i)c(i,j)} \log p_{c(i)c(i,j)} + (1 - y_{c(i)c(i,j)}) \log (1 - p_{c(i)c(i,j)}) \quad (2)$$

4.2 Other Approaches

We compare ourselves with several alternatives, some of which are inspired by the literature.

- As an alternative to fDNA, we also use features extracted from layer fc14 of a VGG16 network pre-trained on ImageNet with their dimensionality reduced to 128 by PCA for comparability (denoted fc14), and from (Simo-Serra and Ishikawa, 2016) (denoted 128floats).
- We implement the architecture variations mentioned in Figure 2.
- We implement several alternative query-article-matching and loss strategies. The various combinations are listed in Table 2.

In this study, the triplet ranking loss is given by $l_{ijk} = \sigma(f(I_i|\theta)^T (f(A_k|\gamma) - f(A_j|\gamma)))$ where I_i is a query image, A_j an article present in I_i , A_k an article

which is not known to be present in I_i , θ the set of parameters of the model, γ is θ for a siamese model and empty otherwise, $f(\cdot|\cdot)$ the feature vector of an image given parameters and σ the sigmoid function. For each query image of each batch, one positive article is sampled as A_j , and 50 negative articles are sampled as A_k , giving 50 triplets at a time. The 50 losses are summed, as shown in Equation 3, where $c(i)$ is the i^{th} query image in the batch, $c(i, +)$ the sampled positive article for this image, and $c(i, j)$ the j^{th} negative article.¹

$$\mathcal{L} = \sum_{i=1}^{64} \sum_{j=1}^{50} l_{c(i)c(i,+)c(i,j)} \quad (3)$$

In our all-in-two-nets model, the right leg has the same architecture as the left leg described in Section 4.1 and is also pretrained on ImageNet, but no parameters are shared between the two legs, as in (Huang et al., 2015). All layers are trainable, even the pretrained ones.

5 RESULTS

5.1 Retrieval on Test Query Images

Experimental Set-up. We randomly split the dataset into training and test set, with 80% of query images and articles being kept for training. Many of our full-body images are annotated with several articles that may be shared across images, making a clean split of articles impossible. To avoid using training articles at test time, we discard them from the retrieval set, and discard query images that were annotated with such articles. As a result, our pool of test queries is slightly biased towards half-body images.

We run tests on 20000 randomly sampled test query images against 50000 test articles, which is roughly the number of articles of the selected categories at a given time in Zalando’s assortment. For each test query image, all 50000 possible (image, article) pairs are submitted to the model and are ranked according to their score.

Retrieval Performance. Table 3 shows the performance of the various models (a plot of these results can also be found in Figure 8 in Appendix). We use as performance measure top- k retrieval, which gives the proportion of query images for which the correct article was found at position k or below. We add a top-1% measure which here means top-500, so that research groups with a different number of articles may compare their results to ours more easily. Average refers

Table 3: Results of the retrieval test using 20000 query images against 50000 Zalando articles. Top- k indicates the proportion of query images for which the correct article was found at position k or below. Average and median refer respectively to the average and median position at which an article is retrieved. The best performance is shown in bold. A plot of these results can also be found in Figure 8 in Appendix.

	top-1	top-5	top-10	top-20	top-50	top-1%	average	median
static-fc14-linear	0.005	0.015	0.022	0.033	0.054	0.166	13502	8139
static-fc14-non-linear	0.030	0.083	0.121	0.176	0.271	0.609	1672	258
fc14-non-linear	0.131	0.317	0.423	0.539	0.684	0.926	230	15
128floats-non-linear	0.132	0.319	0.426	0.540	0.677	0.909	274	15
fDNA-ranking-loss	0.091	0.263	0.372	0.494	0.660	0.933	177	20
fDNA-linear	0.121	0.314	0.423	0.547	0.700	0.936	178	15
all-in-two-nets	0.033	0.115	0.181	0.277	0.438	0.838	620	69
Studio2Shop	0.238	0.496	0.613	0.722	0.834	0.970	89	5



Figure 3: Random examples of the retrieval test using 20000 queries against 50000 Zalando articles. Query images are in the left-most column. Each query image is next to two rows displaying the top 50 retrieved articles, from left to right, top to bottom. Green boxes show exact hits.

to the average position (or rank) at which an article is retrieved. We expect the distribution of retrieval positions to be heavy-tailed, so we include the median position as a more robust measure. All our models are assessed on the exact same (query, article) pairs.

Studio2Shop outperforms the other models, with a median article position of 5, mostly because of the non-linear matching module. In general, fc14 with our architecture performs surprisingly well with a median index of 15 and would already be suited for practical applications, which supports our one-leg approach. It is surpassed however by fDNA, indicating that having a fashion-specific representation does help. 128floats is pretrained on ImageNet and finetuned using a fashion dataset, but this dataset seems too limited in size to make a difference with fc14 on this

task.

A second observation is that learning our feature representation from scratch does not perform as well as a pre-trained feature representation. The reason is two-fold. Firstly, fDNA was trained on many more articles than we have in this study, including articles of other categories such as shoes for example, or swimming suits. Secondly, having a pre-trained feature representation heavily reduces the model complexity, which is desirable in terms of computational time, but also in the presence of limited datasets. Extensive architecture exploration may lead to a siamese architecture that outperforms Studio2Shop. We do not conclude that siamese architectures are not performant, only that a one-leg architecture is a viable alternative.

A third observation is that a ranking loss does not

Table 4: Results of the retrieval test on DeepFashion In-Shop-Retrieval (Liu et al., 2016a) using 2922 query images against 683 articles, and on LookBook (Yoo et al., 2016) using 68820 query images against 8726 articles. Top- k indicates the proportion of query images for which the correct article was found at position k or below. Average and median refer respectively to the average and median position at which an article is retrieved.

		top-1	top-5	top-10	top-20	top-50	top-1%	average	median
DeepFashion	fc14-non-linear	0.159	0.439	0.565	0.689	0.838	0.475	32	6
	Studio2Shop	0.258	0.578	0.712	0.818	0.919	0.619	17	3
LookBook	fc14-non-linear	0.009	0.031	0.050	0.078	0.134	0.181	1657	797
	Studio2Shop	0.013	0.044	0.070	0.107	0.182	0.241	1266	466

seem necessary. In our case, it performed worse than the cross-entropy loss. It is also slower to train as it requires triplets of images instead of pairs and, because it is trained to rank and not to predict a match, the scores produced are meaningless.

The total time needed for (naive) retrieval using Studio2Shop is given in Figure 9 in Appendix. While features generated by the query-feature submodel can be pre-computed for efficiency, the query-article-matching submodel still needs to be run and its non-linearity could be a limitation in a realistic scenario. A possibility could be to use fDNA-linear (which achieves the second-best performance and is extremely fast since only a dot product is needed) to identify a subset of candidate articles on which it is worth running the non-linear match of Studio2Shop.

Article Retrieval. Figure 3 shows randomly selected examples of retrievals, with the query image shown on the left, followed by the top 50 articles. Even if the article is not retrieved at the top position, style is conserved, suggesting that the model has indeed generalised beyond simple attributes. Moreover, we are able to find more than one category when they are sufficiently visible, for example similar trousers are also found early in the upper two images, and in the top image t-shirts also appear, though there seems to be some degree of confusion as to which colour belongs to which garment.

5.2 Results on External Datasets

The Datasets. To assess the usability of our model, we run tests on two external datasets, namely *DeepFashion In-Shop-Retrieval* (Liu et al., 2016a), which is the closest to ours and contains 683 title images and 2922 matching shop images, and *LookBook* (Yoo et al., 2016), which contains 8726 title images and 68820 matching shop images with backgrounds. For both datasets, the background is different to ours, and the aspect ratio of the people in query images may also vary due to image resizing. As a result, the

image distributions deviate from ours, both for queries and for targets. Note that publicly available datasets such as *Exact-Street-To-Shop* (Kiapour et al., 2015) or *DeepFashion* (Liu et al., 2016a) have a mixture of image types as targets and can therefore not be used directly. Nevertheless, for *DeepFashion In-Shop-Retrieval* and *Lookbook*, it was easily possible to isolate title images and to restructure the dataset for our needs.

LookBook was originally not meant for street-to-shop, many articles have more than one ID and are treated as different so we are not rewarded for finding them with the wrong ID. The images are very different from ours and we do not really expect any good results, but we use it to test the boundaries of *DeepArticleFinder*.

Retrieval Test. We compute the static features of the title images using the pre-existing feature representation of interest (fc14 or fDNA) and simply apply our model to their query images. It is critical to note here that our model is not fine-tuned to this new data. The results are shown in Table 4. Figures 4 and 5 show the retrieval outcomes for 5 randomly selected query images from *DeepFashion In-Shop-Retrieval* and *LookBook* respectively. Note that, in *DeepFashion In-Shop-Retrieval*, where full bodies can be seen, more than one category can be retrieved. *LookBook* has the added difficulty of containing backgrounds. However, when these remain understated, our model can find adequate suggestions.

Finding Similar Articles in Zalando’s Assortment. While the previous tests allow us to assess performance, in practice we would like to suggest articles from our own assortment. We run qualitative tests on query images from *DeepFashion In-Shop-Retrieval* and *LookBook* using our own test articles as retrieval set. Figures 6 and 7 show the retrieval outcomes for 5 randomly selected query images from *DeepFashion In-Shop-Retrieval* and *LookBook* respectively, and indicate that our model can make very appropriate

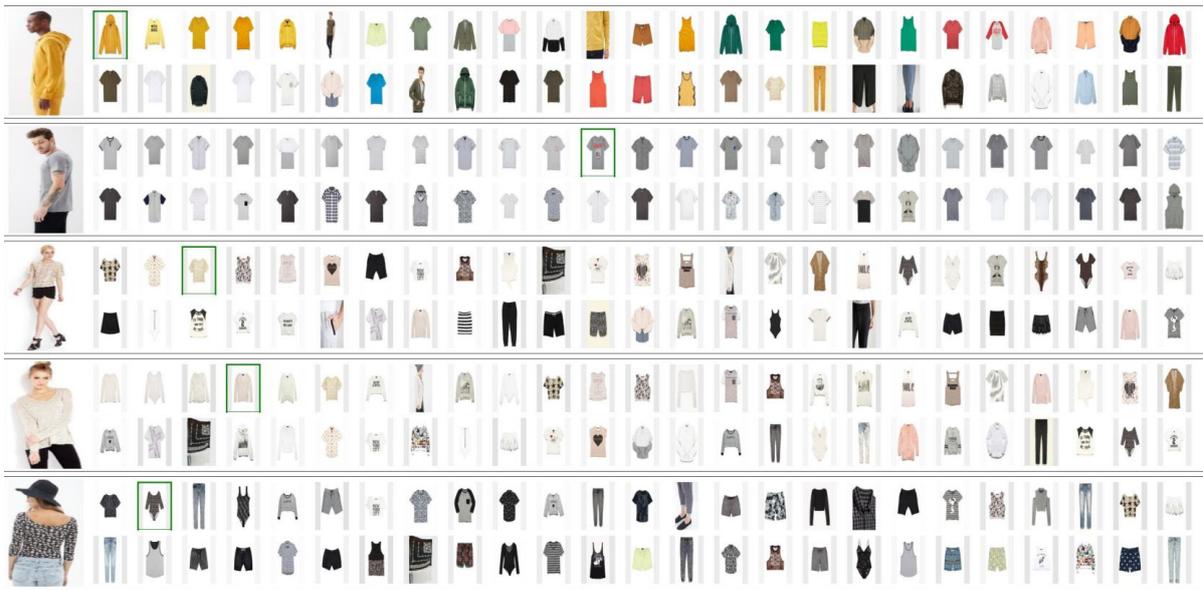


Figure 4: Random examples of outcomes of the retrieval test on query images from DeepFashion In-Shop-Retrieval (Liu et al., 2016a). Query images are in the left-most column. Each query image is next to two rows displaying the top 50 retrieved articles, from left to right, top to bottom. Green boxes show exact hits.



Figure 5: Random examples of outcomes of the retrieval test on query images from LookBook (Yoo et al., 2016). Query images are in the left-most column. Each query image is next to two rows displaying the top 50 retrieved articles, from left to right, top to bottom. Green boxes show exact hits.

suggestions for DeepFashion In-Shop-Retrieval, and to a certain extent for LookBook when background is understated.

6 CONCLUSION

We have presented Studio2Shop, a model for article recognition in fashion images with neutral backgrounds. Instead of solving the problem from scratch as most recent studies have done, Studio2Shop builds on top of existing feature representations for fashion articles, and projects query images onto this



Figure 6: Random examples of outcomes of the retrieval test on query images from DeepFashion In-Shop-Retrieval (Liu et al., 2016a) against 50000 Zalando articles. Query images are in the left-most columns. Each query image is next to two rows showing the top 50 retrieved articles, from left to right, top to bottom.

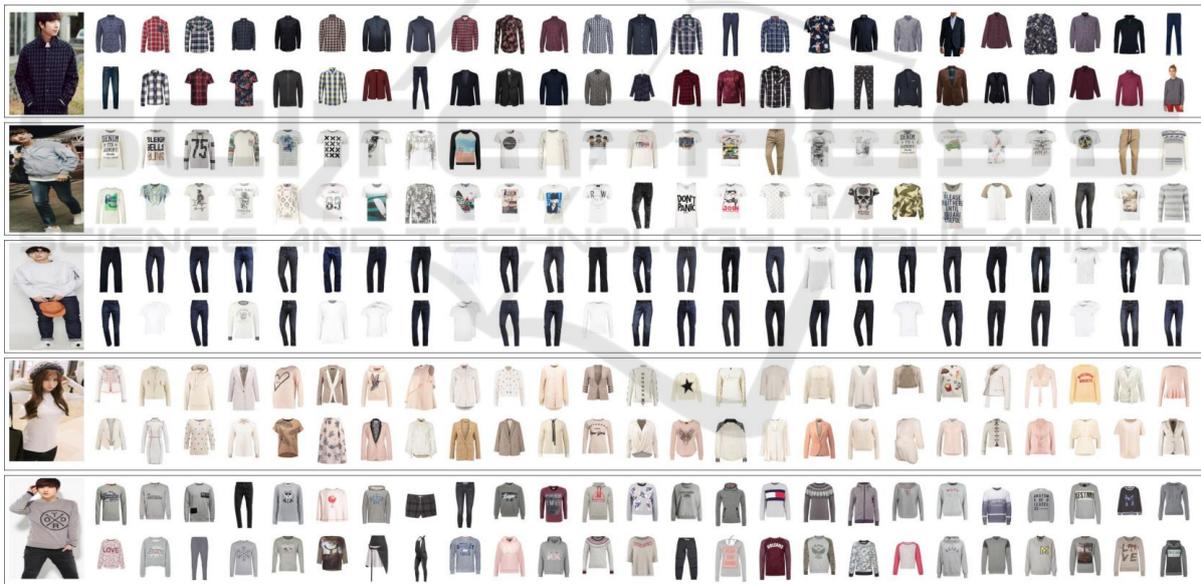


Figure 7: Random examples of outcomes of the retrieval test on query images from LookBook (Yoo et al., 2016) against 50000 Zalando articles. Query images are in the left-most columns. Each query image is next to two rows showing the top 50 retrieved articles, from left to right, top to bottom.

fixed feature space using a deep convolutional neural network. We show that our approach is most often able to find correct articles within a range that is sufficiently small for building realistic visual search engines in the studio-to-shop setting.

Our method is easy to implement and only requires article features and matches between query images and articles. We find that we achieve satisfactory results without having to use additional meta-data for

our query images or articles, and that, in the absence of specific features such as FashionDNA, the publicly available features fc14 or 128floats are already quite powerful.

While our dataset does not contain street images, many of the obstacles in computer vision for fashion such as occlusion and deformation remain, yet our results are very promising. We are currently working on extending our approach to street images, making use

of state-of-the-art image segmentation techniques and external datasets.

ACKNOWLEDGEMENTS

The authors would like to thank Sebastian Heinz and Christian Bracher for their help with FashionDNA.

REFERENCES

- Bossard, L., Dantone, M., Leistner, C., Wengert, C., Quack, T., and Van Gool, L. (2013). Apparel classification with style. In *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part IV, ACCV'12*, pages 321–335.
- Bracher, C., Heinz, S., and Vollgraf, R. (2016). Fashion DNA: merging content and sales data for recommendation and article mapping. *CoRR*, abs/1609.02489.
- Chen, H., Gallagher, A., and Girod, B. (2012). Describing clothing by semantic attributes. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III, ECCV'12*, pages 609–623.
- Chen, Q., Huang, J., Feris, R., Brown, L. M., Dong, J., and Yan, S. (2015). Deep domain adaptation for describing people based on fine-grained clothing attributes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05*, pages 886–893, Washington, DC, USA. IEEE Computer Society.
- Di, W., Wah, C., Bhardwaj, A., Piramuthu, R., and Sundaresan, N. (2013). Style finder: Fine-grained clothing style detection and retrieval. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Dong, J., Chen, Q., Xia, W., Huang, Z., and Yan, S. (2013). A deformable mixture parsing model with parselets. In *ICCV*, pages 3408–3415.
- Fu, J., Wang, J., Li, Z., Xu, M., and Lu, H. (2013). *Efficient Clothing Retrieval with Semantic-Preserving Visual Phrases*, pages 420–431.
- Huang, J., Feris, R. S., Chen, Q., and Yan, S. (2015). Cross-domain image retrieval with a dual attribute-aware ranking network. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1062–1070.
- Jagadeesh, V., Piramuthu, R., Bhardwaj, A., Di, W., and Sundaresan, N. (2014). Large scale visual recommendations from street fashion images. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14*, pages 1925–1934.
- Kalantidis, Y., Kennedy, L., and Li, L.-J. (2013). 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, ICMR '13*, pages 105–112.
- Kiapour, M. H., Han, X., Lazebnik, S., Berg, A. C., and Berg, T. L. (2015). Where to buy it: Matching street clothing photos in online shops. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3343–3351.
- Kiapour, M. H., Yamaguchi, K., Berg, A. C., and Berg, T. L. (2014). *Hipster Wars: Discovering Elements of Fashion Styles*, pages 472–488.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.
- Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., and Yan, S. (2014). Fashion parsing with weak color-category labels. *IEEE Trans. Multimedia*, 16(1):253–265.
- Liu, S., Feng, J., Song, Z., Zhang, T., Lu, H., Xu, C., and Yan, S. (2012a). Hi, magic closet, tell me what to wear! In *Proceedings of the 20th ACM International Conference on Multimedia, MM '12*, pages 619–628.
- Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., and Yan, S. (2012b). Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3330–3337.
- Liu, Z., Luo, P., Qiu, S., Wang, X., and Tang, X. (2016a). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Liu, Z., Yan, S., Luo, P., Wang, X., and Tang, X. (2016b). Fashion landmark detection in the wild. In *European Conference on Computer Vision (ECCV)*.
- Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.
- Simo-Serra, E. and Ishikawa, H. (2016). Fashion Style in 128 Floats: Joint Ranking and Classification using Weak Data for Feature Extraction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.
- Vittayakorn, S., Yamaguchi, K., Berg, A. C., and Berg, T. L. (2015). Runway to realway: Visual analysis of

fashion. In *2015 IEEE Winter Conference on Applications of Computer Vision*, pages 951–958.

- Wang, N. and Haizhou, A. (2011). Who blocks who: Simultaneous clothing segmentation for grouping images. In *Proceedings of the International Conference on Computer Vision, ICCV'11*.
- Wang, X., Sun, Z., Zhang, W., Zhou, Y., and Jiang, Y.-G. (2016). Matching user photos to online products with robust deep features. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICMR '16*, pages 7–14.
- Wang, X. and Zhang, T. (2011). Clothes search in consumer photos via color matching and attribute learning. In *Proceedings of the 19th ACM International Conference on Multimedia, MM '11*, pages 1353–1356.
- Yamaguchi, K., Kiapour, M. H., and Berg, T. L. (2013). Paper doll parsing: Retrieving similar styles to parse clothing items. In *2013 IEEE International Conference on Computer Vision*, pages 3519–3526.
- Yamaguchi, K., Kiapour, M. H., Ortiz, L., and Berg, T. (2012). Parsing clothing in fashion photographs. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3570–3577.
- Yamaguchi, K., Okatani, T., Sudo, K., Murasaki, K., and Taniguchi, Y. (2015). Mix and match: Joint model for clothing and attribute recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 51.1–51.12.
- Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. (2016). Pixel-level domain transfer. *CoRR*, abs/1603.07442.

APPENDIX

FashionDNA: Implementation Details

fDNA features are extracted from the deep convolutional neural network described in Table 5. This net is trained on title images to predict a number of article attributes, summarised in Table 6. Training is done via multiple categorical cross-entropy losses, one for each attribute.

Table 5: Architecture of the network used for the fDNA features of title images.

name	type	kernel	activation	output size	# params
	InputLayer			256x177x3	0
	Conv2D	11x11	ReLU	62x42x96	34.944
	MaxPooling2D			31x21x96	0
	LRN			31x21x96	0
	Conv2D	5x5	ReLU	31x21x256	614.656
	MaxPooling2D			15x10x256	0
	LRN			15x10x256	0
	Conv2D	3x3	ReLU	15x10x384	885.120
	Conv2D	3x3	ReLU	15x10x384	1.327.488
	Conv2D	3x3	ReLU	15x10x256	884.992
	MaxPooling2D			7x5x256	0
	Flatten			8960	0
fDNA	Dense		ReLU	1536	13.764.096
	Dense		ReLU	1024	1.573.888
commodity	Dense		softmax	1448	1.484.200
	Dense		ReLU	890	1.367.930
article_number	Dense		softmax	445	396.495
	Dense		ReLU	160	245.920
silhouette	Dense		softmax	80	12.880
	Dense		ReLU	1024	1.573.888
brand	Dense		softmax	3719	3.811.975
	Dense		ReLU	306	470.322
target_group	Dense		softmax	153	46.971
	Dense		ReLU	40	61.480
pattern	Dense		softmax	19	779
	Dense		ReLU	72	110.664
material	Dense		softmax	36	2.628
	Dense		ReLU	244	375.028
main_colour	Dense		softmax	122	29.890
	Dense		ReLU	140	215.180
second_colour	Dense		softmax	70	9870
					29.301.284

Table 6: Attributes used for training FashionDNA. Commodity group, statistical article number and silhouette all describe article categories on different levels of granularity. The target group attribute describes combinations of age and gender.

attribute	# values	coverage [%]
commodity group	1448	100
statistical article number	445	76
silhouette	80	100
brand	3719	98
target group	153	100
pattern	19	1
material	36	1
main colour	122	100
second colour	70	7

Architecture of Studio2Shop

Table 7: Architecture of Studio2Shop.

name	type	activation	output size	# params
<i>query-feature submodule</i>				
query_input	InputLayer		224x155x3	0
	VGG16		7x4x512	14,714,688
	Flatten		14436	0
	Dense	ReLU	2048	29,362,176
	Dropout(0.5)		2048	0
	Dense	ReLU	2048	4,196,352
	Dropout(0.5)		2048	0
query_fDNA	Dense		128	262,272
<i>query-article-matching submodule</i>				
article_fDNA	InputLayer		128	0
query_fDNA	InputLayer		128	0
	Concatenate		256	0
	BatchNorm		256	512
	Dense	ReLU	256	65792
	Dense	ReLU	256	65792
	Dense	sigmoid	1	257
				48,667,841

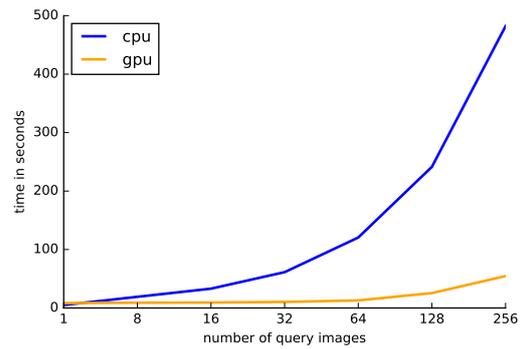


Figure 9: Time needed to test query images against 50000 articles with a fixed batch size of 64 and using 50 articles at a time. Model loading is not part of the time calculation. The experiments are run from scratch (apart from model loading) for each number of images and averaged over 10 repetitions. To process only one image, it takes about 5s on a CPU and 8s on a GPU.

Top-k Retrieval Results

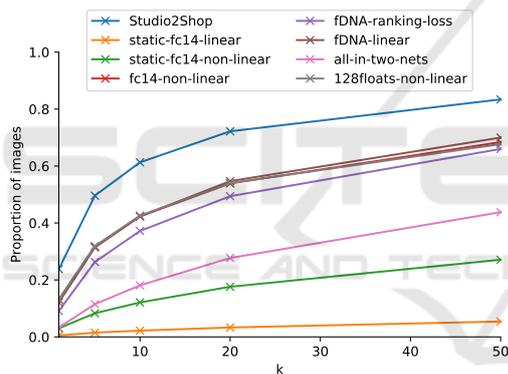


Figure 8: Top-k results of the retrieval test using 20000 query images against 50000 Zalando articles. The top-k metric gives the proportion of query images for which the correct article was found at position k or below.

Timings

Figure 9 shows the time needed, both for a CPU (Intel Xeon processor with 3.5 GHz) and for a GPU (Nvidia K80), to test query images against 50000 articles with a fixed batch size of 64, which is rather small but common. First, the features of a query image are extracted using the query-feature submodel, then the query-article-matching submodel is applied to all images against all the articles, using 50 articles at a time. These two steps (query-feature + query-article-matching) are part of the calculation, however model loading and image resizing are not. These numbers are rough but are averaged over 10 repetitions and are meant to give an order of magnitude.

To process only one image, it takes about five seconds on a CPU, 8 on a GPU (the difference is explained by the overhead inherent to GPU calculations that becomes very quickly negligible as the number of images increases). Note that our implementation is not optimised for speed. For example, there is no parallelisation. Moreover, we use the exact same architecture as for training and therefore only test against 50 articles at a time. For production, retrieval time could be substantially decreased by testing instead against hundreds of articles at a time. Further speed-ups could be achieved by increasing the batch size when running on a GPU.