

# A Strong Baseline for Fashion Retrieval with Person Re-Identification models

Mikolaj Wieczorek<sup>1</sup>, Andrzej Michalowski<sup>1</sup>, Anna Wroblewska<sup>1,2</sup>[0000-0002-3407-7570], and Jacek Dabrowski<sup>1</sup>[0000-0002-1581-2365]

<sup>1</sup> Synerise

<sup>2</sup> Faculty of Mathematics and Information Science, Warsaw University of Technology

**Abstract.** Fashion retrieval is the challenging task of finding an exact match for fashion items contained within an image. Difficulties arise from the fine-grained nature of clothing items, very large intra-class and inter-class variance. Additionally, query and source images for the task usually come from different domains - street photos and catalogue photos respectively. Due to these differences, a significant gap in quality, lighting, contrast, background clutter and item presentation exists between domains. As a result, fashion retrieval is an active field of research both in academia and the industry.

Inspired by recent advancements in Person Re-Identification research, we adapt leading ReID models to be used in fashion retrieval tasks. We introduce a simple baseline model for fashion retrieval, significantly outperforming previous state-of-the-art results despite a much simpler architecture. We conduct in-depth experiments on *Street2Shop* and *DeepFashion* datasets and validate our results. Finally, we propose a cross-domain (cross-dataset) evaluation method to test the robustness of fashion retrieval models.

**Keywords:** clothes retrieval, quadruplet loss, person re-identification, deep learning in fashion

## 1 Introduction

Fashion image retrieval,<sup>3</sup> commonly associated with visual search, has received a lot of attention recently. This is an interesting challenge from both a commercial and academic perspective. The task of fashion retrieval is to find an exact or very similar products from the vendor’s catalogue (gallery) to the given query image. Creating a model that can find similarities between content of the images is essential for the two basic visual-related products for the fashion industry: recommendations and search.

Visual recommendations (VR) - visual similarity needs to be found between a currently viewed product and other products in the database. Retrieval is done among the images from a single domain, which is a domain of catalogue photos.

<sup>3</sup> Image retrieval pertains to finding similar images as a whole, while in the case of fashion, the task is an instance retrieval task, as we want to find a match to a single item contained in the image. In this work we use those two terms interchangeably.

Visual search (VS) - visual similarity between user taken/uploaded photo and the products' photos in the vendor's database. This case is a cross-domain retrieval as the photos from catalogue contains generally a single item and are taken by professionals using high-quality equipment, assuring proper background (usually solid white) and lighting conditions. On the other hand, photos taken by users are taken with a smartphone camera in uncontrolled lighting conditions and are of lower quality, noisy, with multiple persons and garments.

From a business perspective, especially for e-commerce retailers, these tools still have a lot of untapped potential. VS in particular can be used in various ways to enrich the customer experience and facilitate a more convenient search, since it may be easier and more natural for users to take a photo of a product and upload it directly via an app rather than to use textual search.

During work on VS solutions, we found that the task of visual search for clothes is in many ways analogous to Person Re-Identification problem (ReID). Yet we have not encountered any work that examines models specific to Person ReID tasks in the context of fashion image retrieval tasks. Thus, we decided to modify and apply approaches from Person ReID field to fashion retrieval.

There are four main differences between the ReID and VS problems. Firstly, in ReID tasks data may be regarded as homogeneous, since the query and gallery images are both taken by CCTV cameras, therefore they are from the same domain. The cameras, resolution, view angles, and other factors may vary within the domain. In fashion visual search tasks images come from two domains. One domain contains store catalogue photos taken by professionals at the same studio settings. The second domain consists of photos taken by users, which may vary due to lighting conditions, the quality of the camera, angles and focal length. Additionally, the content of the image may be photographed differently. Catalogue photos are often well aligned, the background is usually solid white or monochromatic and the fashion item is fully visible. In contrast, user images often contain multiple objects, a cluttered background and the fashion item is often occluded or cropped.

Secondly, ReID problems may use additional temporal information to narrow down the search space, e.g. [32], which adds an information stream unavailable in the fashion domain.

Moreover, fashion items, especially clothes, are highly deformable goods resulting in a high intra-class variance in appearance, further complicating appearance comparisons.

Finally, in ReID, the majority of images contain a whole person and can be relatively easily aligned. User images in fashion may differ both in the alignment, orientation and content. They may contain a whole person with multiple clothing items, upper or lower half of a person, clothes laying on the sofa etc.

We also pinpointed four major similarities between ReID and clothes retrieval. In both domains, the core problem is an instance retrieval task that relies heavily on fine-grain details, which makes both tasks complex and challenging. Secondly, both domains must implicitly consider a common subset of clothes & materials deformations specific to human body poses and shapes. Moreover, in



**Fig. 1.** Examples of images from ReID dataset, *DeepFashion* and *Street2Shop*. The difference in alignment, background and quality can be seen between ReID and clothes retrieval datasets, as well as between street and shop photos within a single dataset.

both domains the models have to deal with occlusion, lighting, angle and orientation differences. Finally, clothes are also an important factor in ReID task, especially when faces are not visible.

Due to the similarities and despite the differences of the two aforementioned tasks we decided to investigate how and which ReID solutions can be effectively applied to the fashion retrieval task. Our main contributions are:

- Recognition of deep analogies between Person ReID and Fashion Retrieval tasks.
- A review of ReID models and their applicability to Fashion Retrieval tasks.
- Adjustments of ReID models for their application to the fashion domain.
- Thorough evaluation of the adapted ReID models on two commonly used fashion datasets.
- A simple baseline model for fashion retrieval significantly outperforming state-of-the-art results for *DeepFashion* and *Street2Shop* datasets.
- A cross-domain (cross-dataset) evaluation method to test robustness of the solutions.

In the following sections we provide a review of Person Re-Identification models and solutions used in Fashion Retrieval domain. We show the baselines for our comparisons (Section 2). Then, we provide an argument for our selection of applicable ReID models (Section 3). Subsequently, we describe our results, compare them with previous state-of-the-art approaches and provide a discussion (Section 4) of our findings. We conclude with our best achievements (Section 5). In supplement material we provide statistics of datasets used for training and evaluation, as well as additional samples from model outputs.

## 2 Related Work

This section focuses mainly on the problems of extracting features from images and the retrieval process itself. Instance retrieval is basically a two-stage process: (1) encoding image/item into a  $n$ -dimensional embedding vector; (2) searching for the most similar product embedding to the query in the  $n$ -dimensional space under a chosen distance metric.

## 2.1 Feature Extraction

The process of extracting features from an image may be defined as passing the pixel matrix through a function to obtain an embedding vector that encompasses all necessary information about the image in the encoded form. It is the first and most indispensable step towards image retrieval. Most research is devoted to improving the performance of this stage.

Over the years, the functions used for encoding image content have evolved. At first, hand-crafted features describing global image characteristics (e.g. histogram of oriented gradients - HOG) or local dependencies like SIFT (scale-invariant feature transform) were used. These methods did work, however subsequently convolutional neural networks (CNNs) surpassed these methods in computer vision tasks and dominated the field.

One of the first works using CNN for an image retrieval task was [24] that also managed to surpass the state-of-the-art of approaches based on SIFT, HOG and other hand-crafted features. In [6] authors used CNNs extracted features and trained a model using a triplet loss function, which was widespread by [26] for face recognition task. By 'CNN extracted features' we mean the final output of either one or many convolutional layers that are arranged in various combinations (e.g. pyramid, sequence of layers). All of these combinations work as various learned filters for an image matrix.

## 2.2 Image Retrieval

Image retrieval may be defined as a task of matching images based on their feature vectors/embeddings under a chosen distance metric. There are three main approaches when dealing with image retrieval using deep learning embeddings: direct representation, classification loss training and deep metric learning. We explain them in the following paragraphs.

*Direct representation* This is the simplest approach as it does not involve any training on the target dataset. In most cases, an ImageNet pre-trained neural network is used. To obtain the feature vector, an image is fed into the network and the output from one or more layers is treated as its embedding. To find the most similar images to the query, a cosine or other similarity metric is applied to the set of embeddings. Such an approach was used in [14].

*Classification loss training* Another approach is to again take a pre-trained CNN and train it on the target dataset using a classification loss function. The difference to the direct representation is that the network is further trained (fine-tuned) on a target dataset. Thus, the embeddings should be optimized in a way to allow correct classification and in turn indirectly improve the retrieval quality.

This approach was also present in previously mentioned reference [14], where the authors used pre-trained AlexNet to learn the similarity between street and shop photos using cross-entropy loss over pair of images, classifying them as either matching or non-matching.



*Deep metric learning* Image retrieval formulated as a classification problem is not an optimal approach, as the classification is done mainly in the last fully connected (FC) layer of the neural network (NN), which is not used during the inference stage. Moreover, the gradient flow does not explicitly optimize the feature embeddings to be similar/dissimilar under the distance metric used for retrieval.

Deep metric learning treats the image retrieval problem somewhat as a clustering or ranking problem and optimizes the mutual arrangement of the embeddings in space. One of the most widely used approaches that pulls the same class closely and pushes away other class embeddings is using a triplet loss for training neural network. The triplet loss and its' modifications are described in our supplement in more details.

### 2.3 Fashion Image Understanding

Fashion image understanding has been an area of research for a while. The papers that introduced two modern large public datasets [38], [13], gave the start for a rapid development of machine learning applications in the fashion field. Over the years numerous fashion datasets were released by various authors: [14], [43], [47], [11], [21], [4], [7] accelerating the research and allowing a wider range of tasks to be tackled.

Clothes retrieval seems to be the most popular task [11], [4], [43], [39], [16]. However, it is often combined with another task to attain a synergy effect, either with landmark detection (detecting keypoints of clothes) [20], attribute prediction [18], [20], or object detection [11], [4], [16], [14].

### 2.4 Re-identification

The problem of Person Re-Identification, being an instance retrieval task, has undergone a similar evolution as the image retrieval domain, starting from using hand-crafted features [17] and matching their embeddings under chosen distance metric. Currently CNN extracted features are the dominant approach.

Similarly to Section 2.2 a variety of loss functions are used in ReID problems: classification loss [42], verification loss [5] and triplet loss [19]. Triplet Loss performs best [9] in most cases by a significant margin.

Due to the fact that the ReID problem focuses on images of people from CCTV cameras, who are mostly in a standing position, some works exploited this fact by adding this information during training. In [41] the authors used horizontal pooling of image stripes, while [23] modified a network architecture search algorithm specifically for ReID task, that also integrated the human body in an upright position during training. [29], [25] proposed using human/skeleton pose information along with global and local features for better alignment of features, thus, increasing retrieval performance.

Moreover, Person Re-Identification problem settings allow to implicitly or explicitly use spatio-temporal information, as time constraint allows to eliminate

a portion of irrelevant images during retrieval stage [2], [12]. Currently spatio-temporal approach [32] tops the leaderboard in Person ReID task on *Market1501* - common dataset for Person Re-Identification task.

### 3 Our Approach

The aim of our work was to investigate if and how ReID models can be successfully applied to a fashion retrieval task. Based on the performance of various models in Person Re-Identification task<sup>4</sup> and their suitability for the fashion retrieval problem, we selected the most appropriate models. The motivation behind our final choice was threefold:

1. The chosen models should exhibit top performance on ReID datasets.
2. The chosen models should cover different approaches with regards to the network architecture, training regime etc.
3. The code had to be publicly available.

#### 3.1 ReID Models

Two models were chosen based on the above criteria.

First, [22] presents an approach that combines the most efficient training approaches and sets a strong baseline for other ReID researchers, hence its name: *ReID Strong Baseline* (RST). Through a thorough evaluation it showed that by combining simple architecture, global features and using training tricks (warm-up learning rate, random erasing augmentation, label smoothing, last stride, BNNeck, center loss; for more details see [22]) one can surpass state-of-the-art performance in the ReID domain.

The second choice - *OSNet-AIN* [45] uses features extracted from different scales, which seems to be a well-suited approach in a fine-grained instance retrieval. The authors devised a flexible fusion mechanism that allows the network to focus on local or global features that improves performance. This is a different approach from the first one, as both global and local features are used and the whole neural architecture was purpose-built for ReID task.

It is worth mentioning that some top scoring papers from website *Paperswithcode* were not chosen as they either required temporal data [32] or made an implicit assumptions of spatial alignment of objects in images [23]. Neither approach was suitable to the clothes retrieval task and the datasets we used for evaluation.

#### 3.2 Settings for Our Experiments

To fully explore ReID models and their application to the fashion retrieval task we conducted a number of experiments with the two selected approaches. We explored several aspects with potential to influence the performance in the fashion domain:

---

<sup>4</sup> <https://paperswithcode.com/task/person-re-identification>

- backbone architecture (a feature extractor part of a model; see Figure 2)
- triplet and quadruplet loss functions in RST
- size of input images
- re-ranking (post-processing methods to improve accuracy of retrieval)
- cross-domain evaluation (training on one dataset and testing on another)

Using the selected ReID models we dealt with three loss functions. As for the *OSNet-AIN*, we followed the training regime proposed by its authors and we trained the model using only a classification loss. For the RST model we used a loss function that consisted of three elements: a classification loss, a triplet loss and a center loss. Additionally, we tested if by replacing the triplet loss with a quadruplet loss one could improve the performance.<sup>5</sup>

### 3.3 SOTA Models for Clothes Retrieval vs RST Model

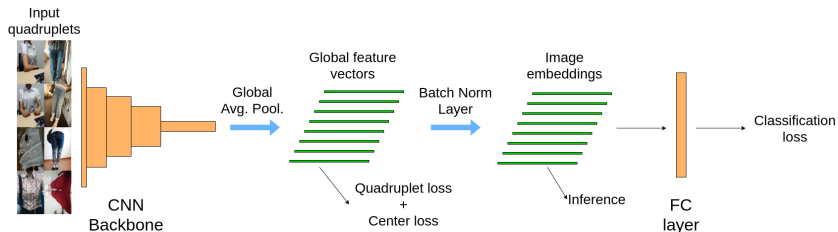
A *detect-then-retrieve model for multi-domain fashion item retrieval* [16] showed SOTA performance for *Street2Shop* dataset. The model consists of Object Detector (Mask-RCNN), which is used to crop clothing items from shop images and a Retrieval Model that is built upon Resnet-50 backbone, but incorporates RMAC pipeline [6]. The RMAC is a global representation of images that uses CNN as a local feature extractor. Local features are extracted and max-pooled from multiple multi-scale overlapping regions covering whole image, creating a single feature vector for each region. These region features are aggregated, normalized, transformed and then normalized again. Moreover, in [16] an ensemble model was also created that consisted of two single models, one trained using triplet loss and other one using AP loss. Input image sizes used in [16] were also much larger than used in our RST model as the images were resized to 800 pixels, which caused problem in processing them through the network, since they were forced to process either a single image (AP loss) or a single triplet (triplet loss) at a time. For more details see [16].

*Fashion Retrieval via Graph Reasoning Networks on a Similarity Pyramid* [15] achieved SOTA performance on *DeepFashion* dataset. They based their solution on graphs and graph convolution. The architecture consists of three parts. First is a CNN network, which extracts multi-scale (pyramid) features from numerous, overlapping windows that cover the whole image. What is important two images of the same item are fed into the network – street and shop images and they are processed together as a pair. Second part is the Similarity Computation, which computes region similarity between street and shop images for all possible local feature combinations at the same pyramid scale. Next, a graph is built. Computed region similarities are the nodes, the relations between region similarities are the edges. Scalar edges weights are computed automatically based on the node, incoming and outgoing edges. Finally, the convolution operations are applied to the graph and classification loss is used for training. The aim of the

<sup>5</sup> Triplet and quadruplet loss functions and their modifications are described in our supplementary material in more detail.

network is to classify the pair of images as depicting the same clothes or not. Input size of images was 224x224, which is slightly smaller than used in our best performing RST model. For more details see [15].

*RST model* Compared to the two models described above, the RST model can be characterised by its simplicity. Even though, the architecture is much simpler, the performance exceeds significantly current SOTA. The RST model consists of 3 parts. First, CNN backbone extracts features from images and global average pooling is applied to create global feature vectors. They are used for computing quadruplet and center loss. Next, the global feature vectors are normalized and we call them *Images embeddings* (see Figure 2). *Images embeddings* are used during training as input to a fully connected layer, while during inference stage (retrieval) they are used to compute similarity distance. In Figure 2 we presents the RST model, which shows the pipeline and all parts of the architecture. It can be deemed as strikingly simple, yet it substantially exceeds current SOTA results.



**Fig. 2.** RST model architecture

## 4 Experiments

In this section we describe our methodology and details of experiments we ran. We conducted evaluation of our approach and its settings on *Street2Shop* and *DeepFashion* datasets.<sup>6</sup>

For both *Street2Shop* and *DeepFashion* we cropped the clothing items of interest from the whole images, as both datasets contain ground-truth bounding boxes for street photos. Additionally *DeepFashion* contains bounding boxes for shop images, so we cropped them as well. *Street2Shop* does not provide bounding boxes for shop images, therefore we used the original, un-cropped shop images for the gallery set. These settings for street images are in line with those used

<sup>6</sup> These datasets, used metrics and many visualizations are presented very precisely in our supplement.

in [16] and [15], which we consider as the current SOTA for *Street2Shop* and *DeepFashion* respectively.

## 4.1 Training

To adapt the RST model to fashion datasets we introduced some changes to the code, which were mostly required due to significant difference in size between ReID and clothes retrieval datasets. The size of *DeepFashion* and *Street2Shop* caused RAM and video RAM overflow, which was solved by using more memory-efficient methods. Moreover, we added some new functionalities and scripts. Our improvements are described in detail in the supplement.

Unfortunately, evaluation with re-ranking and without category constraint for *Street2Shop* was impossible despite improving the code efficiency, therefore we decided to conservatively estimate the results. The estimated results are marked with asterisk (\*). For more details see the supplement.

## 4.2 ReID Models Comparison

In the first experiment we trained both ReID models to compare their performance on *DeepFashion* and *Street2Shop* datasets. For training we used 256x128 input images. Table 1 contains the results of the models and current state-of-the-art approach. It can be seen that the RST model surpasses the current SOTA model and the *OSNet-AIN* performs worse than the other two approaches. The reason for poor performance of *OSNet* may be the fact that it is a lightweight network and built very specifically for ReID tasks.

Due to the large performance gap between RST and *OSNet* models, we decided to conduct further experiments using only the more promising RST model.

**Table 1.** Comparison of performance of models on *Street2Shop* and *DeepFashion* data. Best performance across models on a given dataset is in bold

Model	<i>Street2Shop</i>				<i>DeepFashion</i>				
	mAP	Acc@1	Acc@10	Acc@20	mAP	Acc@1	Acc@10	Acc@20	Acc@50
RST	<b>37.2</b>	<b>42.3</b>	<b>61.1</b>	<b>66.5</b>	<b>35</b>	<b>30.8</b>	<b>62.3</b>	<b>69.4</b>	<b>78</b>
OSNet-AIN	18.9	25.3	40.8	45.4	20.1	17.5	40.2	46.9	53.8
Current SOTA	29.7	34.4	-	60.4	-	27.5	-	65.3	75

## 4.3 Backbone Influence

In this section we inspect the influence of the used backbone on the results. The results of our experiments are shown in Table 2. All runs during this experiment were performed on input images of size 256x128, using an RST approach with all tricks enabled and using quadruplet loss. All models were trained for 120 epochs

on 1 GPU, with base learning rate 0.0001 and batch size 128. *SeResNext-50* [10], *EfficientNet-b2* and *EfficientNet-b4* [30] were trained on 2 GPUs using ModelParallel mode due to their size.

Our findings are in line with backbone performance presented by the authors in [22], i.e. *ResNet50-IBN-A* [37] is the best performing backbone. Regarding the results, we infer that such a large advantage in performance for *Resnet50-IBNs* may be caused by instance normalization, which reduces the impact of variations in contrast and lighting between street and shop photos.

**Table 2.** Results achieved by different backbones on *Street2Shop* and *DeepFashion* datasets compared to the current SOTA.

Backbone	<i>Street2Shop</i>				<i>DeepFashion</i>				
	mAP	Acc@1	Acc@10	Acc@20	mAP	Acc@1	Acc@10	Acc@20	Acc@50
ResNet-50	32.0	36.6	55.3	60.6	32.4	28.1	58.3	65.5	74.2
SeResNet-50	30.5	34.6	53.1	58.7	31.3	27.0	57.8	65.4	74.4
SeResNeXt-50	31.9	36.9	54.5	59.7	32.2	27.8	58.5	66.0	74.6
ResNet50-IBN-A	<b>37.2</b>	<b>42.3</b>	<b>61.1</b>	<b>66.5</b>	<b>35.0</b>	<b>30.8</b>	62.3	<b>69.4</b>	<b>78.0</b>
ResNet50-IBN-B	36.9	41.9	60.6	65.1	32.2	28.1	58.4	65.8	74.7
EfficientNet-b1	28.8	35.1	52.1	56.7	28.5	23.4	49.8	56.8	66.1
EfficientNet-b2	29.4	34.0	50.8	56.2	24.1	20.4	45.9	53.2	62.6
EfficientNet-b4	31.8	38.2	55.6	60.2	26.8	23.1	59.1	57.4	66.7
Current SOTA	29.7	34.4	-	60.4	-	25.7	<b>64.4</b>	-	75.0

#### 4.4 Influence of Loss Functions

The results for the RST model using triplet and quadruplet loss are shown in Table 3. It can be seen that the quadruplet loss in our test settings performed marginally better than the triplet loss, yet it brings an improvement at almost no cost.

**Table 3.** Our results on *Street2Shop* and *DeepFashion* datasets achieved with triplet and quadruplet loss functions

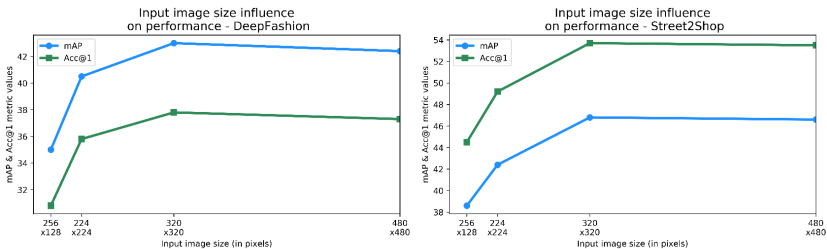
Loss function	<i>Street2Shop</i>				<i>DeepFashion</i>			
	mAP	Acc@1	Acc@10	Acc@20	mAP	Acc@1	Acc@10	Acc@20
Quadruplet	<b>37.2</b>	<b>42.3</b>	<b>61.1</b>	<b>66.5</b>	<b>35</b>	<b>30.8</b>	62.3	69.4
Triplet	37.1	41.8	60.4	65.7	34.8	30.5	<b>62.4</b>	<b>69.5</b>

## 4.5 Influence of Input Image Size

Even though the results achieved with 256x128 input images had already surpassed the current SOTA performance for many backbones (See Table 2), we decided to test if larger images would result in even higher performance. Outcomes of our experiments are presented in Table 4. It can be seen that using larger images allows to further boost performance. In our settings, we achieved best results for input images of size 320x320. Using larger images (480x480) did not bring any advantage. Additional plots are available in the supplement.

**Table 4.** Comparison of performance of clothes retrieval with different input image sizes. All experiments were performed using *Resnet50-IBN-A* RST model with all tricks enabled, quadruplet loss function and without re-ranking

Input size	<i>Street2Shop</i>				<i>DeepFashion</i>				
	mAP	Acc@1	Acc@10	Acc@50	mAP	Acc@1	Acc@10	Acc@20	Acc@50
256x128	38.6	44.5	62.5	67.2	35.0	30.8	62.3	69.4	78.0
224x224	42.4	49.2	66.2	70.9	40.5	35.8	68.5	75.1	82.4
320x320	<b>46.8</b>	<b>53.7</b>	<b>69.8</b>	<b>73.6</b>	<b>43.0</b>	<b>37.8</b>	<b>71.1</b>	<b>77.2</b>	<b>84.1</b>
480x480	46.6	53.5	69.0	72.9	42.4	37.3	69.4	75.4	82.2
Current SOTA	29.7	34.4	-	60.4	-	27.5	-	65.3	76.0



**Fig. 3.** Two plots presenting influence of input image size on mAP and Acc@1 metric for *DeepFashion* and *Street2Shop* datasets respectively.

## 4.6 Influence of Re-ranking

One of the methods boosting performance tested in [22] was re-ranking during test stage. Re-ranking is a post-processing method applied to raw retrieval results in order to improve the accuracy of the results. In our experiments we used k-reciprocal encoding proposed by [44] similarly to [22].

Table 5 presents the comparison of RST models with and without using re-ranking. It can be seen that the results with re-ranking post-processing improved significantly for both datasets. The improvement is greatest for mAP and Acc@1, thus suggesting that re-ranking ‘pushes’ the positive samples closer to the start of the results list. Such behaviour seems desirable for real-world applications where a visual search user will obtain more relevant items at the top

**Table 5.** The comparison of performance for the model without and with re-ranking. For results marked with \* see Section 4.1

	Street2Shop				DeepFashion			
	mAP	Acc@1	Acc@10	Acc@20	mAP	Acc@1	Acc@10	Acc@20
No re-ranking	46.8	53.7	69.8	<b>73.6</b>	43.0	37.8	71.1	77.2
Re-ranking	<b>54.8*</b>	<b>57.1*</b>	<b>69.9*</b>	72.9*	<b>47.3</b>	<b>40.0</b>	<b>73.5</b>	<b>79.0</b>

#### 4.7 Comparison to the State-of-the-art

In Table 6 we compare our results to the state-of-the-art on *Street2Shop* dataset [16]. We list the best performing ensemble model from the aforementioned paper, as it achieved better results than their best single model. In comparison, we show the results of our best performing single model, which was trained using an RST approach, *Resnet50-IBN-A* backbone using quadruplet loss during training. It can be seen that generally our approach performs better by a large margin despite using only 320x320 input image size compared to 800x800 used in [16].

In all categories and in overall performance our model clearly outperforms current state-of-the-art by a large margin. In two categories our model performs marginally worse, but it may be attributed to the fact that we use much smaller images. The categories with inferior performance consist of small items such as belts or eye-wear, where fine-grained details are more important and a high image resolution is beneficial. In our case cropped items are of very small dimensions, which may limit performance.

It is important to note that the mAP, Acc@1, Acc@20 reported by [16] as overall performance are just average values across all categories for each metric. There, the retrieval stage was limited to products only from a specific category, thus limiting the choice of the model and making retrieval less challenging. We also report the performance in these settings for our model and for [16] in the *Average over categories* row of Table 6 to allow fair comparison. Additionally, we propose an unconstrained evaluation, where we conduct retrieval from all gallery images, with no restrictions. Our results are in the *Unconstrained retrieval* row of Table 6.

The large gap between *Average over categories* and *Unconstrained retrieval* derives from the fact that the well performing categories such as *skirts* and *dresses* are the most numerous ones, therefore they weigh more in the final



results than categories that have few queries, such as *belts*. Hence, unconstrained retrieval is closer to a weighted average than the simple average named by us as *Average over categories*, where each category has equal weight when calculating the final results.

**Table 6.** Comparison of performance on *Street2Shop* dataset using mAP, Acc@1, and Acc@20 metrics. Best performance for each category is presented in bold per metric. For results marked with \* see Section 4.1

Category	Current SOTA [16]			Our Model			Our Model Re-ranking		
	mAP	Acc@1	Acc@20	mAP	Acc@1	Acc@20	mAP	Acc@1	Acc@20
bags	23.4	36	62.6	32.2	44.2	<b>74.6</b>	<b>39</b>	<b>45.7</b>	69.6
belts	9.4	9.5	<b>42.9</b>	<b>11.3</b>	<b>12.2</b>	46.3	10.5	7.3	29.3
dresses	49.5	56.4	72.0	65.8	73.7	<b>85.9</b>	<b>75.3</b>	<b>76.7</b>	85.8
eyewear	26.7	36.2	91.4	27	31.4	76.5	<b>27.5</b>	<b>37.3</b>	<b>80.4</b>
footwear	11.0	14.8	34.2	34.2	37.9	<b>65.4</b>	<b>44.3</b>	<b>43.3</b>	<b>65.4</b>
hats	32.8	30.8	70.8	38.5	37.5	85.9	<b>52.7</b>	<b>53.1</b>	<b>90.6</b>
leggings	18.2	20.5	49.0	30.8	37.3	70.7	<b>36.9</b>	<b>39.9</b>	<b>73.7</b>
outerwear	28.1	30.5	47.9	36.8	43.5	68.0	<b>45.3</b>	<b>51.9</b>	<b>70.2</b>
pants	28.2	<b>33.33</b>	<b>51.5</b>	23.9	27.3	42.4	<b>30.4</b>	<b>33.3</b>	48.5
skirts	62.3	68.0	80.2	64.5	71.2	<b>86.5</b>	<b>73.3</b>	<b>75.1</b>	85.5
tops	36.9	42.7	61.6	46.8	52.7	71.9	<b>56.5</b>	<b>57.9</b>	<b>72.9</b>
Average over categories	29.7	34.4	60.4	37.4	42.6	<b>70.4</b>	<b>44.7</b>	<b>47.4</b>	70.2
Unconstrained retrieval	-	-	-	46.8	53.7	<b>73.6</b>	<b>54.8*</b>	<b>57.1*</b>	72.9*

In Table 7 results on *DeepFashion* dataset are presented. Our model outperforms the results of [3] in terms of Accuracy for all  $k$  values. Especially Acc@1 noted largest relative boost.

**Table 7.** Comparison of performance on *DeepFashion* dataset using Acc@1, Acc@20 and Acc@50 metrics. Best performance for each metric is presented in bold

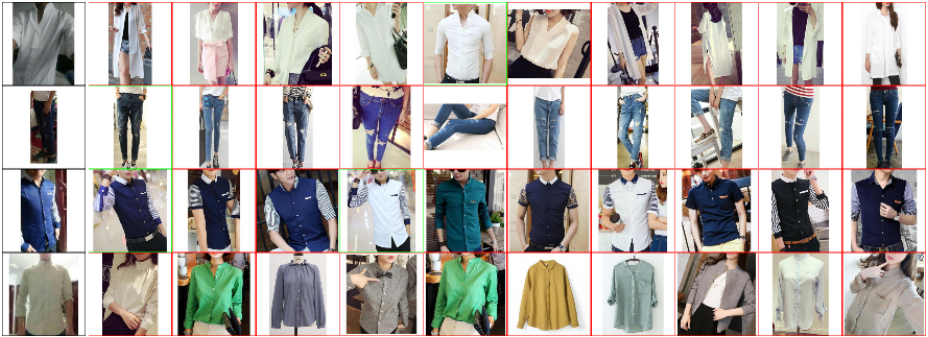
Current SOTA [15]			Our Model			Our Model Re-ranking		
Acc@1	Acc@20	Acc@50	Acc@1	Acc@20	Acc@50	Acc@1	Acc@20	Acc@50
25.73	64.38	75	37.8	77.2	84.1	<b>40</b>	<b>79</b>	<b>85.5</b>

## 4.8 Cross-domain Evaluation

Inspired by the Person Re-Identification domain, where cross-domain evaluation is often performed to measure robustness of the models [22], [46], we decided to conduct such a test using our best performing RST model.



**Fig. 4.** Examples of retrieval on the *Street2Shop* dataset produced by our best model on 320x320 images. The images in the first column are query images, while the images on their right are the retrieval results with decreasing similarity towards the right side. Retrieval images with green border are the true match to the query. The top 10 most similar retrieval images are shown. It is worth noting that the retrieval was performed on the whole gallery dataset, with no pruning to the query item’s category.



**Fig. 5.** Examples of retrieval on *DeepFashion* dataset produced by our best model on 320x320 images. Retrieval and visualization settings are identical as in Figure 4

Cross-domain evaluation consists of training a model on Dataset A and evaluation of its performance on Dataset B. In such settings training and test data distributions are different.

In Table 8 we present the results of our cross-domain experiments, both with and without re-ranking. It can be seen that cross-domain performance of the RST model on both datasets is much lower than when model is trained and tested on the same dataset - same domain (See Table 6 and Table 7). We suppose that such a large gap in performance between cross and same-domain evaluation is caused by a significant difference in data distribution of the two datasets. For example *DeepFashion* contains only three categories: lower, upper and full-body, while *Street2Shop* contains 11 fine-grained clothing categories.

Even though cross-domain evaluation results are far from our best performing models, they are on a reasonable level when compared to the current state-of-the-art results for each dataset, thus, indicating that the RST model with *Resnet50-IBN-A* backbone can learn meaningful representation of garments that can be transferred between domains to a large extent.

**Table 8.** Comparison of the performance of the RST on cross-domain evaluation. DF  $\rightarrow$  S2S means that the model was trained on *DeepFashion* dataset and tested on the *Street2Shop* test set. For results marked with \* see Section 4.1

	DF $\rightarrow$ S2S				S2S $\rightarrow$ DF			
	mAP	Acc@1	Acc@10	Acc@20	mAP	Acc@1	Acc@10	Acc@20
No re-ranking	26.2	37.7	49.5	53.3	20.9	18.6	40.7	47.2
Re-ranking	27.9*	37.4*	48.8*	51.7*	23.5	20.5	43.7	49.6

## 5 Conclusions

In this paper we examined the analogies and differences between Person Re-Identification research field and Fashion Retrieval, then examined the transferability of ReID models, and performed adjustments necessary to make them work for fashion items.

Using the approach proposed by [22] using only global features, a number of improvements and learning optimizations we achieved 54.8 mAP, 72.9 Acc@20 on *Street2Shop* and 47.3 Acc@1, 79.0 Acc@20 on *DeepFashion* dataset, establishing new state-of-the-art results for both. The performance on *Street2Shop* seems particularly robust, compared to previous state-of-the-art as our results were achieved on images several times smaller than the previous top-scoring models.

By analogy to [22] we consider the results achieved by our model as a strong baseline for further clothes retrieval research. Despite a simple architecture, the baseline model outperforms previous solutions dedicated and customized specifically for fashion retrieval systems. Our results are a clear indica-

tion that field of fashion retrieval can benefit from research established for Person Re-Identification. Additionally, we performed supplementary experiments and showed that quadruplet loss may bring further improvements at a negligible cost, and that re-ranking can further boost performance. Finally, we introduced cross-domain evaluation into clothes retrieval research to test robustness of fashion retrieval models.

## References

1. Chen, W., Chen, X., Zhang, J., Huang, K.: Beyond triplet loss: a deep quadruplet network for person re-identification. arXiv:1704.01719 [cs] (Apr 2017), <http://arxiv.org/abs/1704.01719>, arXiv: 1704.01719
2. Cho, Y.J., Kim, S.A., Park, J.H., Lee, K., Yoon, K.J.: Joint person re-identification and camera network topology inference in multiple cameras. *Computer Vision and Image Understanding* **180**, 34–46 (Mar 2019). <https://doi.org/10.1016/j.cviu.2019.01.003>, <http://www.sciencedirect.com/science/article/pii/S1077314219300037>
3. Dodds, E., Nguyen, H., Herdade, S., Culpepper, J., Kae, A., Garrigues, P.: Learning Embeddings for Product Visual Search with Triplet Loss and Online Sampling. arXiv:1810.04652 [cs] (Oct 2018), <http://arxiv.org/abs/1810.04652>, arXiv: 1810.04652
4. Ge, Y., Zhang, R., Wu, L., Wang, X., Tang, X., Luo, P.: DeepFashion2: A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images. arXiv:1901.07973 [cs] (Jan 2019), <http://arxiv.org/abs/1901.07973>, arXiv: 1901.07973
5. Geng, M., Wang, Y., Xiang, T., Tian, Y.: Deep Transfer Learning for Person Re-identification. arXiv:1611.05244 [cs] (Nov 2016), <http://arxiv.org/abs/1611.05244>, arXiv: 1611.05244
6. Gordo, A., Almazan, J., Revaud, J., Larlus, D.: End-to-end Learning of Deep Visual Representations for Image Retrieval. arXiv:1610.07940 [cs] (May 2017), <http://arxiv.org/abs/1610.07940>, arXiv: 1610.07940
7. Guo, S., Huang, W., Zhang, X., Srikhanta, P., Cui, Y., Li, Y., Scott, M.R., Adam, H., Belongie, S.: The iMaterialist Fashion Attribute Dataset. arXiv:1906.05750 [cs] (Jun 2019), <http://arxiv.org/abs/1906.05750>, arXiv: 1906.05750
8. Harwood, B., G, V.K.B., Carneiro, G., Reid, I., Drummond, T.: Smart Mining for Deep Metric Learning. arXiv:1704.01285 [cs] (Jul 2017), <http://arxiv.org/abs/1704.01285>, arXiv: 1704.01285
9. Hermans, A., Beyer, L., Leibe, B.: In Defense of the Triplet Loss for Person Re-Identification. arXiv:1703.07737 [cs] (Nov 2017), <http://arxiv.org/abs/1703.07737>, arXiv: 1703.07737
10. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-Excitation Networks. arXiv:1709.01507 [cs] (May 2019), <http://arxiv.org/abs/1709.01507>, arXiv: 1709.01507
11. Huang, J., Feris, R., Chen, Q., Yan, S.: Cross-Domain Image Retrieval with a Dual Attribute-Aware Ranking Network. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1062–1070. IEEE, Santiago, Chile (Dec 2015). <https://doi.org/10.1109/ICCV.2015.127>, <http://ieeexplore.ieee.org/document/7410484/>
12. Huang, W., Hu, R., Liang, C., Yu, Y., Wang, Z., Zhong, X., Zhang, C.: Camera Network Based Person Re-identification by Leveraging Spatial-Temporal Constraint and Multiple Cameras Relations. In: Tian, Q., Sebe, N., Qi, G.J., Huet, B., Hong, R., Liu, X. (eds.) *MultiMedia Modeling*. pp. 174–186. Lecture Notes in Computer Science, Springer International Publishing, Cham (2016). [https://doi.org/10.1007/978-3-319-27671-7\\_15](https://doi.org/10.1007/978-3-319-27671-7_15)
13. Jagadeesh, V., Piramuthu, R., Bhardwaj, A., Di, W., Sundaresan, N.: Large Scale Visual Recommendations From Street Fashion Images. arXiv:1401.1778 [cs] (Jan 2014), <http://arxiv.org/abs/1401.1778>, arXiv: 1401.1778

14. Kiapour, M.H., Han, X., Lazebnik, S., Berg, A.C., Berg, T.L.: Where to Buy It: Matching Street Clothing Photos in Online Shops. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3343–3351. IEEE, Santiago, Chile (Dec 2015). <https://doi.org/10.1109/ICCV.2015.382>, <http://ieeexplore.ieee.org/document/7410739/>
15. Kuang, Z., Gao, Y., Li, G., Luo, P., Chen, Y., Lin, L., Zhang, W.: Fashion Retrieval via Graph Reasoning Networks on a Similarity Pyramid. arXiv:1908.11754 [cs] (Aug 2019), <http://arxiv.org/abs/1908.11754>, arXiv: 1908.11754
16. Kucer, M., Murray, N.: A detect-then-retrieve model for multi-domain fashion item retrieval. In: CVPR Workshops (2019)
17. Kstinger, M., Hirzer, M., Wohlhart, P., Roth, P.M., Bischof, H.: Large scale metric learning from equivalence constraints. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2288–2295 (Jun 2012). <https://doi.org/10.1109/CVPR.2012.6247939>, ISSN: 1063-6919
18. Liao, L., He, X., Zhao, B., Ngo, C.W., Chua, T.S.: Interpretable Multimodal Retrieval for Fashion Products. In: 2018 ACM Multimedia Conference on Multimedia Conference - MM '18. pp. 1571–1579. ACM Press, Seoul, Republic of Korea (2018). <https://doi.org/10.1145/3240508.3240646>, <http://dl.acm.org/citation.cfm?doid=3240508.3240646>
19. Liu, J., Zha, Z.J., Tian, Q., Liu, D., Yao, T., Ling, Q., Mei, T.: Multi-Scale Triplet CNN for Person Re-Identification. In: Proceedings of the 24th ACM international conference on Multimedia. pp. 192–196. MM '16, Association for Computing Machinery, Amsterdam, The Netherlands (Oct 2016). <https://doi.org/10.1145/2964284.2967209>, <https://doi.org/10.1145/2964284.2967209>
20. Liu, J., Lu, H.: Deep Fashion Analysis with Feature Map Upsampling and Landmark-Driven Attention. In: Leal-Taix, L., Roth, S. (eds.) Computer Vision ECCV 2018 Workshops, vol. 11131, pp. 30–36. Springer International Publishing, Cham (2019). [https://doi.org/10.1007/978-3-030-11015-4\\_4](https://doi.org/10.1007/978-3-030-11015-4_4), [http://link.springer.com/10.1007/978-3-030-11015-4\\_4](http://link.springer.com/10.1007/978-3-030-11015-4_4)
21. Liu, Z., Luo, P., Qiu, S., Wang, X., Tang, X.: DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1096–1104. IEEE, Las Vegas, NV, USA (Jun 2016). <https://doi.org/10.1109/CVPR.2016.124>, <http://ieeexplore.ieee.org/document/7780493/>
22. Luo, H., Jiang, W., Gu, Y., Liu, F., Liao, X., Lai, S., Gu, J.: A Strong Baseline and Batch Normalization Neck for Deep Person Re-identification. arXiv:1906.08332 [cs] (Jun 2019), <http://arxiv.org/abs/1906.08332>, arXiv: 1906.08332
23. Quan, R., Dong, X., Wu, Y., Zhu, L., Yang, Y.: Auto-ReID: Searching for a Part-aware ConvNet for Person Re-Identification. arXiv:1903.09776 [cs] (Aug 2019), <http://arxiv.org/abs/1903.09776>, arXiv: 1903.09776 version: 4
24. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 512–519. IEEE, Columbus, OH, USA (Jun 2014). <https://doi.org/10.1109/CVPRW.2014.131>, <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6910029>
25. Sarfraz, M.S., Schumann, A., Eberle, A., Stiefelhagen, R.: A Pose-Sensitive Embedding for Person Re-Identification with Expanded Cross Neighborhood Re-Ranking. arXiv:1711.10378 [cs] (Apr 2018), <http://arxiv.org/abs/1711.10378>, arXiv: 1711.10378

26. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: A Unified Embedding for Face Recognition and Clustering. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 815–823 (Jun 2015). <https://doi.org/10.1109/CVPR.2015.7298682>, <http://arxiv.org/abs/1503.03832>, arXiv: 1503.03832
27. Shrivastava, A., Gupta, A., Girshick, R.: Training Region-Based Object Detectors With Online Hard Example Mining. pp. 761–769 (2016), [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/html/Shrivastava\\_Training\\_Region-Based\\_Object\\_CVPR\\_2016\\_paper.html](https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Shrivastava_Training_Region-Based_Object_CVPR_2016_paper.html)
28. Sohn, K.: Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In: NIPS (2016)
29. Su, C., Li, J., Zhang, S., Xing, J., Gao, W., Tian, Q.: Pose-driven Deep Convolutional Model for Person Re-identification (Sep 2017). <https://doi.org/10.1109/ICCV.2017.427>, <https://arxiv.org/abs/1709.08325v1>
30. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. ICML **abs/1905.11946** (2019), <http://arxiv.org/abs/1905.11946>
31. Vishvakarma, A.: MILDNet: A Lightweight Single Scaled Deep Ranking Architecture. arXiv:1903.00905 [cs] (Mar 2019), <http://arxiv.org/abs/1903.00905>, arXiv: 1903.00905
32. Wang, G., Lai, J., Huang, P., Xie, X.: Spatial-Temporal Person Re-identification. arXiv:1812.03282 [cs] (Dec 2018), <http://arxiv.org/abs/1812.03282>, arXiv: 1812.03282 version: 1
33. Wang, J., song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning Fine-grained Image Similarity with Deep Ranking. arXiv:1404.4661 [cs] (Apr 2014), <http://arxiv.org/abs/1404.4661>, arXiv: 1404.4661
34. Wang, X., Hua, Y., Kodirov, E., Hu, G., Garnier, R., Robertson, N.M.: Ranked List Loss for Deep Metric Learning. arXiv:1903.03238 [cs] (Aug 2019), <http://arxiv.org/abs/1903.03238>, arXiv: 1903.03238
35. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A Discriminative Feature Learning Approach for Deep Face Recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision ECCV 2016, vol. 9911, pp. 499–515. Springer International Publishing, Cham (2016). [https://doi.org/10.1007/978-3-319-46478-7\\_31](https://doi.org/10.1007/978-3-319-46478-7_31), [http://link.springer.com/10.1007/978-3-319-46478-7\\_31](http://link.springer.com/10.1007/978-3-319-46478-7_31)
36. Wu, C.Y., Manmatha, R., Smola, A.J., Krhenbhl, P.: Sampling Matters in Deep Embedding Learning. arXiv:1706.07567 [cs] (Jan 2018), <http://arxiv.org/abs/1706.07567>, arXiv: 1706.07567
37. Xingang Pan, Ping Luo, J.S., Tang, X.: Two at once: Enhancing learning and generalization capacities via ibn-net. In: ECCV (2018)
38. Yamaguchi, K., Kiapour, M.H., Berg, T.L.: Paper Doll Parsing: Retrieving Similar Styles to Parse Clothing Items. In: 2013 IEEE International Conference on Computer Vision. pp. 3519–3526. IEEE, Sydney, Australia (Dec 2013). <https://doi.org/10.1109/ICCV.2013.437>, <http://ieeexplore.ieee.org/document/6751549/>
39. Yang, W., Luo, P., Lin, L.: Clothing Co-Parsing by Joint Image Segmentation and Labeling. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 3182–3189 (Jun 2014). <https://doi.org/10.1109/CVPR.2014.407>, <http://arxiv.org/abs/1502.00739>, arXiv: 1502.00739
40. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (Oct 2016). <https://doi.org/10.1109/LSP.2016.2603342>

41. Zhang, X., Luo, H., Fan, X., Xiang, W., Sun, Y., Xiao, Q., Jiang, W., Zhang, C., Sun, J.: AlignedReID: Surpassing Human-Level Performance in Person Re-Identification. arXiv:1711.08184 [cs] (Jan 2018), <http://arxiv.org/abs/1711.08184>, arXiv: 1711.08184
42. Zheng, L., Bie, Z., Sun, Y., Wang, J., Su, C., Wang, S., Tian, Q.: MARS: A Video Benchmark for Large-Scale Person Re-Identification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision ECCV 2016. pp. 868–884. Lecture Notes in Computer Science, Springer International Publishing, Cham (2016). [https://doi.org/10.1007/978-3-319-46466-4\\_52](https://doi.org/10.1007/978-3-319-46466-4_52)
43. Zheng, S., Yang, F., Kiapour, M.H., Piramuthu, R.: ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations. arXiv:1807.01394 [cs] (Jul 2018), <http://arxiv.org/abs/1807.01394>, arXiv: 1807.01394
44. Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking Person Re-identification with k-Reciprocal Encoding. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3652–3661. IEEE, Honolulu, HI (Jul 2017). <https://doi.org/10.1109/CVPR.2017.389>, <http://ieeexplore.ieee.org/document/8099872/>
45. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Learning Generalisable Omni-Scale Representations for Person Re-Identification. arXiv:1910.06827 [cs] (Oct 2019), <http://arxiv.org/abs/1910.06827>, arXiv: 1910.06827
46. Zhou, K., Yang, Y., Cavallaro, A., Xiang, T.: Omni-Scale Feature Learning for Person Re-Identification. arXiv:1905.00953 [cs] (Dec 2019), <http://arxiv.org/abs/1905.00953>, arXiv: 1905.00953
47. Zou, X., Kong, X., Wong, W., Wang, C., Liu, Y., Cao, Y.: FashionAI: A Hierarchical Dataset for Fashion Understanding p. 9 (2018)



# Supplementary material for: A Strong Baseline for Fashion Retrieval with Person Re-Identification models

Mikolaj Wieczorek<sup>1</sup>, Andrzej Michalowski<sup>1</sup>, Anna Wroblewska<sup>1,2</sup>[0000–0002–3407–7570], and Jacek Dabrowski<sup>1</sup>[0000–0002–1581–2365]

<sup>1</sup> Synerise

<sup>2</sup> Faculty of Mathematics and Information Science, Warsaw University of Technology

In the following sections of supplementary material we provide loss function definitions used in fashion retrieval domain (Section 1) along with metrics used (Section 2). Then we describe two open datasets used in our tests which are commonly used in the domain (Section 3). We also demonstrate a few problems with the datasets along with our adjustments for the community to use them in more convenient way using the popular COCO format.

Finally we list a few examples of outputs of our models, and also examples with re-ranking technique and without them (Section 5).

In the following sections of supplementary material we provide loss function definitions used in fashion retrieval domain (Section 1) along with metrics used (Section 2). Then we describe two open datasets used in our tests which are commonly used in the domain (Section 3). We also demonstrate a few problems with the datasets along with our adjustments for the community to use them in more convenient way using the popular COCO format.

Finally we list a few examples of outputs of our models, and also examples with re-ranking technique and without them (Section 5).

## 1 Loss functions

In the image retrieval task there are two loss functions commonly used: classification and triplet loss. Therefore, prevailing number of works in the image retrieval domain use a combination of a classification and a triplet loss for training deep learning models. Classification loss function is used to identify exact id of a person/garment (i.e. images of the same person/garment have the same id). It is a standard loss in classification tasks and in our case it is cross-entropy loss. Also all of the considered models use either of these two loss functions.

Deep metric learning treats the image retrieval problem somewhat as a clustering or ranking problem and optimizes the mutual arrangement of the embeddings in space. One of the most widely used approaches that pulls the same class closely and pushes away other class embeddings is using a triplet loss [26] for training neural network.

Triplet loss is formulated as follows:

$$\mathcal{L}_{triplet} = \left[ \|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha \right]_+ \quad (1)$$

where  $[z]_+ = \max(z, 0)$  and  $f$  denotes learnt embedding function applied to all data points.

A triplet loss consists of an anchor image (a query)  $A$ , a positive example  $P$  – the other image of the same object (in this paper – clothing item) present in the  $A$  image – and negative sample  $N$ , which is an image of a different object from that shown in the image  $A$ .

Learning NN with triplet loss minimizes the intra-class distance, between anchor and positive samples, and maximizes inter-class distance, between anchor and negative samples. The triplet loss proved to achieve state-of-the-art performance and became a standard in similarity learning tasks [33], [26], [9].

In triplet loss strategy, the method of creating triplets is an important part of the training and influences the model performance immensely [36]. In [26] authors used semi-hard triplets, where negative samples are further away from the anchor than the positive samples, but still the loss value is positive, thus, it allows learning.

Most of the works we examined use online hard negative sampling to form a training triplet. This methods select such data points, so that the negative sample is closer to the anchor than the positive sample. As a results, the neural network is given only the triplets that maximize the value of the loss function, therefore it is called 'hard'. This method of creating triplets proved to perform better than other sampling methods and is used in numerous works [27], [40], [22], [31].

To further improve the triplet loss some authors either extends the number of tuples in the loss [1], [28], [34] or/and propose novel sampling methods [8], [36]. However, the reported improvements are not high, thus, we did not use them in our experiments.

Triplet, and in general n-tuple-loss, aims to properly arrange embeddings in an n-dimensional space. While the triplet loss is a common choice in retrieval/ranking tasks, we also examined the quadruplet loss and its influence on the performance. Our implementation of the quadruplet loss follows one found in [1]:

$$\mathcal{L}_{quad} = \left[ \|f(A) - f(P)\|_2^2 - \|f(A) - f(N_1)\|_2^2 + \alpha_1 \right]_+ + \left[ \|f(A) - f(P)\|_2^2 - \|f(N_2) - f(N_1)\|_2^2 + \alpha_2 \right]_+ \quad (2)$$

where the first term is the same as in Equation 1, thus, it takes care of the pull-push relation between anchor, positive and negative samples. The second term demands the intra-class distance to be smaller than the maximum inter-class distance in respect to a different probe -  $N_2$  ( $N_2 \neq N_1 \Rightarrow N_2$  and  $N_1$  represents different garments/IDs).  $\alpha_1$  and  $\alpha_2$  are the margin values, which are set dynamically as in [1].

In [22] additionally center loss [35] is used as one of the training tricks. It aims to pull same class embeddings together as the n-tuple-loss considers only relative distance between embeddings neglecting the distance absolute values. Center loss alleviate this problem by penalizing the distance between embeddings and their

id/class center. Formula for center loss is as follows:

$$\mathcal{L}_{center} = \frac{1}{2} \sum_{j=1}^B \left\| \mathbf{f}_{t_j} - \mathbf{c}_{y_j} \right\|_2^2 \quad (3)$$

where  $y_j$  denotes label of  $j$ -th image in the mini-batch.  $B$  is the batch size,  $\mathbf{f}_{t_j}$  is an embedding of  $j$ -th image and  $\mathbf{c}_{y_j}$  is the center of  $y_j$ -th class features center.

## 2 Metrics in fashion retrieval

To evaluate the performance of our approach we used metrics that we found most often in the related papers. Most widely used metric in retrieval tasks is *Accuracy@k* ( $Acc@k$ ), formulated as:

$$Acc@k = \frac{1}{N} \sum_{i=1}^N 1 [S_q^+ \cap S_q^K], \quad (4)$$

where  $N$  is the number of queries and  $1 [S_q^+ \cap S_q^K]$  is an indicator function, which evaluates to 1 if the ground-truth image is within top- $k$  retrieved results.  $k$  is usually set from 1 to 20. The metric measures if the retrieved item was among top- $k$  proposals.

Second metric that we encountered in the papers was *mAP*, which is a mean average precision, that shows how well the retrieval is done on average. Though *mAP* values were rarely reported in clothes retrieval papers we decided to use this metric in our experiments along  $Acc@k$ .

## 3 Datasets

In this section we describe datasets used for evaluation. Apart from describing their statistics, we also explain the process of reformatting them and how they were processed during our experiments.

### 3.1 *Street2Shop*

The dataset was introduced by [14] and became one of the most widely used datasets for evaluating clothes retrieval solutions. Therefore there is an abundance of works that present their results on the dataset, thus, providing a strong benchmark for our methods. It contains 404,683 shop photos and 20,357 street photos depicting 204,795 distinct clothing items [14]. To allow compatibility across datasets and models we tested, we transformed the *Street2Shop* dataset to COCO-format, while keeping original train/test split and categorization. Annotations in COCO-format are available on our GitHub<sup>3</sup>.

<sup>3</sup> Link will be released in the final submission

In contrast to some authors [15] we decided not to perform any data cleaning or hand-curating images/annotations, even though we encountered some erroneous annotations such as multiple annotations for a single-item image or bounding boxes placed in 'random' places (see examples in Fig. ).

We made such decision to allow a fair comparison with [16], which we found to have best performance on *Street2Shop* dataset, while it does not mention any data cleaning.

### 3.2 DeepFashion

The dataset contributed by [21] contains over 800,00 images, but for our task we only used *Consumer-to-shop Clothes Retrieval* subset that consists of 33,881 distinct clothing items and total of 239,557 images, creating 195,540 pairs. We used results found in [3] as our benchmark, since their were the best we found.

Similarly to *Street2Shop* dataset, *DeepFashion* is also not free from some defects, which we show in Figures 1, 2, 3, 4.

In *Consumer-to-Shop* subset of *DeepFashion*, we found out that the same products and even the same images were assigned different product identifiers. As a result, the retrieval performance is falsely understated compared to the real performance. Two examples of such errors are presented in the supplementary materials to this paper



**Fig. 1.** A collection of photos that depict, we believe, the same clothing item - dress visible in the left most photo. Above each photo there are four pieces of information; from the top: item id, file name, category, subset name. Despite the fact that item ids should be unique for distinct garments, it seems that the same item have various ids assigned, which results in erroneous retrieval results presented in Figure 2

## 4 Code improvements

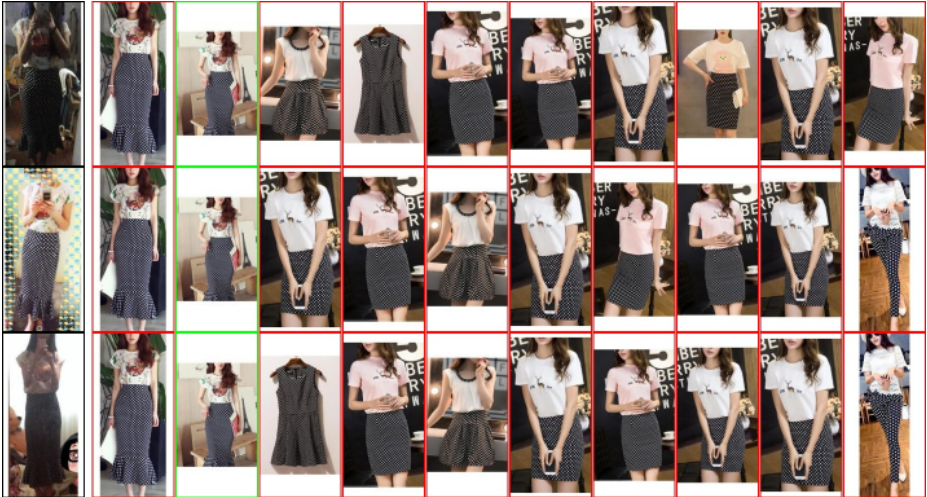
As we mentioned in the main text, we encountered problems with vRAM and RAM overflow caused by the size of the datasets we tested. The RST model contains fully connected layers used for classification of identities/clothes. While ReID datasets the RST model was tested on *Market-1501*, *DukeMTMC-reid* contain 1501 and 1812 unique identities, the fashion datasets have an order of



**Fig. 2.** Examples of retrieval for the query images with product ids from presented in Figure 1 produced by our best model on 320x320 images. The images in the first column are query images, while the images on their right are the retrieval results with decreasing similarity towards the right side. Retrieval images with green border are the true match to the query. The top 10 most similar retrieval images are shown. It can be seen that some images that are just mirrored copies of the same image, yet only one of them is deemed as a true match. We believe it is an error in data annotation, which understates real retrieval performance.



**Fig. 3.** A collection of photos that depict, we believe, the same clothing item - top visible in the left most photo. Above each photo there are four pieces of information; from the top: item id, file name, category, subset name. Despite the fact that item ids should be unique for distinct garments, it seems that the same item have various ids assigned, which results in erroneous retrieval results presented in Figure 4. Interestingly, the right most photos depicts a top that is plain white, which also seems to be incorrect compared the rest.



**Fig. 4.** Examples of retrieval for the query images with product ids from presented in Figure 3 produced by our best model on 320x320 images. The images in the first column are query images, while the images on their right are the retrieval results with decreasing similarity towards the right side. Retrieval images with green border are the true match to the query. The top 10 most similar retrieval images are shown. It can be seen that all results have first and second image the same, while in all cases only the latter is correct even though the former is also from 'top' category, thus, it seems to be pertaining exactly the same garment.

magnitude more identities (clothes) roughly 10000-15000. As a result, the FC layer needs thousands neurons instead of hundreds, thus, it requires much more video RAM (vRAM. To address this problem we introduced two independent solutions:

1. Gradient accumulation - it allows to use smaller mini-batches in constrained vRAM settings when the mini-batch either do not fit into GPU memory or is too small causing the gradient descent to be volatile and prevent model from converging. Gradient accumulation splits original mini-batch into sub-mini-batches, feeds them into network, compute gradients, but the model weights are updated after all sub-mini-batches.
2. ModelParallel mode - it is a distributed training technique that splits a single model across different devices. The data and gradients are moved between devices during forward and backward pass. It was implemented to allow us to test larger backbones that would not fit into a single GPU, use larger mini-batches and, thus speed up the training.

In the code version we used, Resnet-50-IBNs were not yet implemented, therefore we implemented both A and B variants by ourselves based on original implementation. Additionally, we expanded available backbones with the whole EfficientNet family based on the implementations available here.<sup>4</sup>

During evaluation step, especially when performing re-ranking, we encountered problem with RAM consumption. Again, the problem arises from the size of the clothes retrieval datasets we tested. To tackle the problem we introduced three solutions:

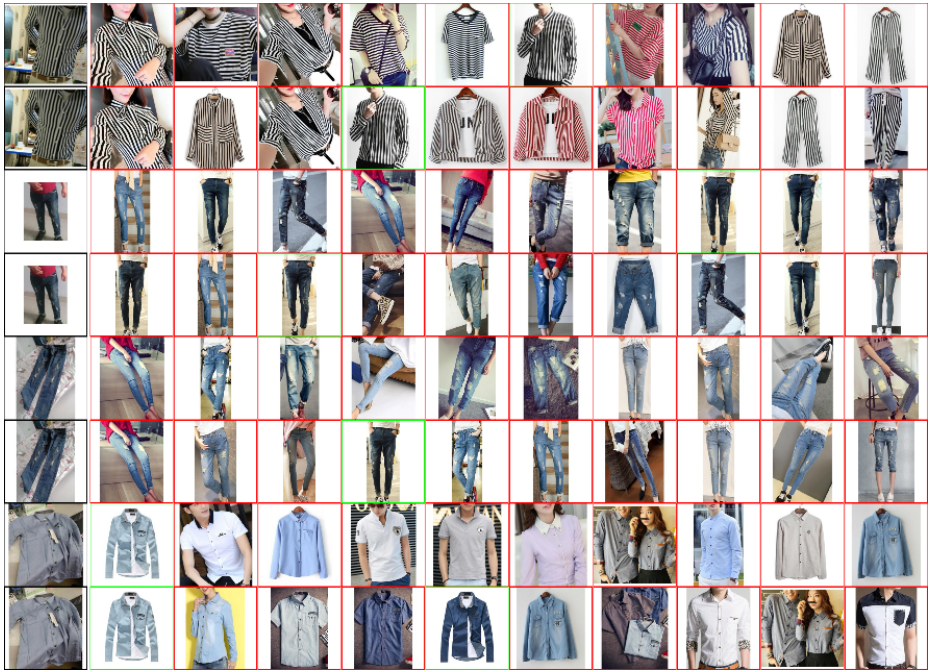
1. We introduced batch processing during both creating images' embeddings, to avoid vRAM overflow, and during computation of distance matrix for tens of thousands images. Originally, both operations were performed in one go.
2. Conducting evaluation with re-ranking for single categories for *Street2Shop* was still problematic due to large RAM requirements, so we used batch processing again, but, we appended intermediate results from batch computations of distance matrix to a file in a hard drive. During re-ranking itself we used Numpy function *memmap* to avoid reading the whole matrix into RAM, while still allowing RAM-like processing.
3. Unfortunately, evaluation with re-ranking and without category constraint was still impossible for *Street2Shop*, as the whole distance matrix over 400,000x400,000 floats, was again too big even when using memap function. We decided to conservatively estimate the results by calculating weighted average over categories and deducting a penalty term. The penalty term was computed for each metric separately using results from the model variant without re-ranking, as the maximum difference, between the weighted average over categories and *Unconstrained Retrieval* values.

Finally, we added *Accuracy@k* computation for specified *k* and a script that at the end of evaluation creates visualization of the results. It plots query image and top-k retrieved images.

<sup>4</sup> <https://github.com/lukemelas/EfficientNet-PyTorch>



## 5 Our result examples

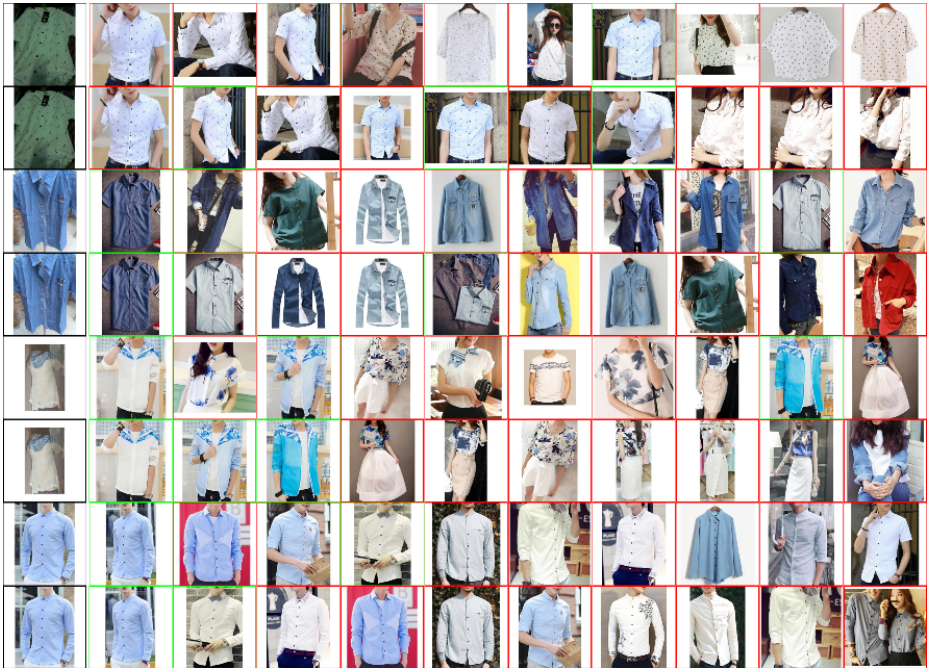


**Fig. 5.** Examples of retrieval on *DeepFashion* dataset produced by our best model on 320x320 images. The images in the first column are query images, while the images on their right are the retrieval results with decreasing similarity towards the right side. Retrieval images with green border are the true match to the query. The top 10 most similar retrieval images are shown. Two retrieval results are shown for each query image, one without and one with re-ranking. The top result from a pair is without re-ranking.





**Fig. 6.** More retrieval results on *DeepFashion* dataset without and with re-ranking.



**Fig. 7.** More retrieval results on *DeepFashion* dataset without and with re-ranking.



**Fig. 8.** Examples of retrieval on *Street2Shop* dataset produced by our best model on 320x320 images. The images in the first column are query images, while the images on their right are the retrieval results with decreasing similarity towards the right side. Retrieval images with green border are the true match to the query. The top 10 most similar retrieval images are shown. Two retrieval results are shown for each query image, one without and one with re-ranking. The top result from a pair is without re-ranking.



Fig. 9. More retrieval results on *Street2Shop* dataset without and with re-ranking.



Fig. 10. More retrieval results on *Street2Shop* dataset without and with re-ranking.