Apparel Classification with Style

Lukas Bossard[1], Matthias Dantone[1], Christian Leistner[1,2],
Christian Wengert[1,3], Till Quack[3], Luc Van Gool[1,4]

[1]ETH Zürich, Switzerland [2]Microsoft, Austria [3]Kooaba AG, Switzerland
[4]KU Leuven, Belgium

Abstract. We introduce a complete pipeline for recognizing and classifying people's clothing in natural scenes. This has several interesting applications, including e-commerce, event and activity recognition, online advertising, *etc.* The stages of the pipeline combine a number of state-of-the-art building blocks such as upper body detectors, various feature channels and visual attributes. The core of our method consists of a multi-class learner based on a Random Forest that uses strong discriminative learners as decision nodes. To make the pipeline as automatic as possible we also integrate automatically crawled training data from the web in the learning process. Typically, multi-class learning benefits from more labeled data. Because the crawled data may be noisy and contain images unrelated to our task, we extend Random Forests to be capable of transfer learning from different domains. For evaluation, we define 15 clothing classes and introduce a benchmark data set for the clothing classification task consisting of over 80,000 images, which we make publicly available. We report experimental results, where our classifier outperforms an SVM baseline with 41.38 % vs 35.07 % average accuracy on challenging benchmark data.

1 Introduction

Clothing serves for much more than covering and protection. It is a means of communication to reflect social status, lifestyle, or membership of a particular

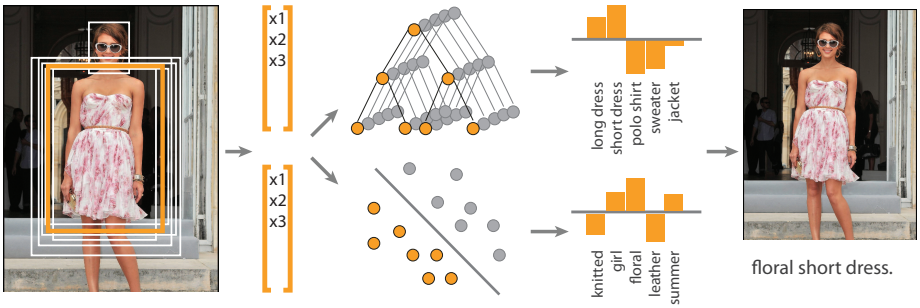


Fig. 1: Overview of our classification pipeline. First, an upper body detection algorithm is applied to the image. Then we densely extract a number of features. Histograms over the extracted features are used as input for a Random Forest (type classification) and for SVMs (attribute classification).

(sub-)culture. The apparel is also an important cue for describing other people. For example: "The man with the black coat", or "the girl with the red bikini". The objective of this paper is to detect, classify, and describe clothes appearing in natural scenes in order to generate such descriptions with a focus on upper body clothing. Typically this means not only recognizing the type of clothing a person is wearing, but also the style, color, patterns, materials, *etc.* An example of a desired outcome would be to label the clothing in Figure 1 as "girl wearing a summer dress with a floral pattern". Only such a combination of type and attributes comes close to the descriptions we use as humans.

Such a system has many potential applications, ranging from automatic labeling in private or professional photo collections, over applications in e-commerce, or contextual on-line advertising up to surveillance. Hence, systems for analyzing visual content may benefit significantly from autonomous apparel classification.

Enabling such robust classification of clothing in natural scenes is a non-trivial task that demands the combination of several computer vision fields. We propose a fully automated pipeline that proceeds in several stages (see Figure 1): First, a state-of-the art face and upper body detector is used to locate humans in natural scenes. The identified relevant image parts are then fed into two higher level classifiers, namely a random forest for classifying the type of clothing and several Support Vector Machines (SVMs) for characterizing the style of the apparel. In case of the random forest, SVMs are also used as split nodes to yield robust classifications at an acceptable speed.

Since the learning of classifiers demands large amounts of data for good generalization, but human annotation can be tedious, costly and inflexible, we also provide an extension of our algorithm that allows for the transfer of knowledge from corresponding data in other domains. E.g. knowledge from crawled web-data may be transferred to manually curated data from a clothing retail chain. We demonstrate this approach on 15 common types (classes) of clothing and 78 attributes. The benchmark data set for cloth classification is consists of over $80,000$ images.

In summary, the contributions of this work are:

- a pipeline for the detection, classification and description of upper body clothes in real-world images
- a benchmark data set for clothing classification
- an extension of Random Forests to transfer learning from related domains

The remainder of this paper is organized as follows. Section 2 discusses related work. An overview of our method is given in Section 3. In Section 4, the benchmark data set is introduced and in Section 5 our algorithms are evaluated. The paper ends with concluding remarks in Section 6.

2 Related Work

Classifying apparel or clothing is part of the wider task of classifying scenes. It is also related to detecting and describing persons in images or videos. Interestingly, in the past there has been little work on classifying clothing. Chen *et al.* [4] manually built a tree of composite clothing templates and match those to the

image. Another strand of work specifically focuses on segmentation of garments covering the upper body [14]. More recently Wang *et al.* [27] also investigated segmentation of upper bodies, where the individuals occlude each other. Retrieving similar clothes given a query image was addressed by Liu *et al.* [20] and Wang *et al.* [28]. In the latter work, the authors use attribute classifiers for re-ranking the search results. Song *et al.* [24] predict people's occupation incorporating information on their clothing. Information extracted from clothing has also been used successfully to improve face recognition results [13].

Very recently, detection and classification of apparel has gained some momentum in the computer vision community. For instance, Yamaguchi *et al.* [29] show impressive results, relying strongly on state-of-the-art body pose estimation and superpixel segmentation. Their work focuses on pixelwise annotation. A somewhat limiting factor of that work is, that occurring labels are supposed to be known beforehand.

In this paper, we do not focus on clothing segmentation or similarity search, but on classification, *i.e.*, the problem of describing what type of clothing is worn in an image. To do so, we build on top of existing work [14, 13, 28] for clothing segmentation as described in Section 3.1, to then fully focus on the classification task. Our work is also related to learning visual attributes, which also has gained importance in recent years. They have been applied in color and pattern naming [12], object description [11], and face verification [16]. Within the context of our proposed task, attributes are obviously suited for describing the visual properties of clothing. To this end, we follow the algorithm by Farhadi *et al.* [11] for semantic attributes and extend it with s-o-a techniques as described in the following section.

3 Classifying Upper Body Clothing

In this work we focus on identifying clothing that people wear on their upper bodies, in the context of natural scenes. This demands the combination of several robust computer vision building blocks, which we will describe in the sequel.

Our apparel classification mechanism consists of two parts: one part describes the overall type/style of clothing, *e.g.*, "suit", "dress", "sweater". The other part describes the attributes of the style, such as "blue", "wool". By combining the outputs of these parts the system can come up with detailed descriptions of the clothing style, such as "blue dress". This combination is crucial for a real-world applications, because the labeling with either only the type ("dress"), or only its attributes ("blue") would be quite incomplete. The combination is also important for higher level tasks, such as event detection. For instance the knowledge that a dress is white may refer to a wedding.

More specifically, our method carries out the following steps: the first stage consists of s-o-a upper body detection as will be described in Section 3.1. After identification of upper bodies, we extract a number of different features from this region with dense sampling as explained in Section 3.2. These features are then transformed into a histogram representation by applying feature coding and pooling.

These features build the basis for classifying the type of apparel (part 1 of the system, Section 3.3) and for classification of apparel attributes (part 2 of the system, Section 3.4).

3.1 Pre-Processing

Throughout this work we deal with real-world consumer images as they are found on the Internet. This entails multiple challenges concerning image quality, *e.g.*, varying lighting conditions, various image scales, *etc.* In a first pre-processing step, we address these variations by normalization of image dimensions and color distributions. This is achieved by resizing each image to 320 pixels maximum side length and by normalizing the histogram of each color channel.

As mentioned earlier, in order to identify clothing we need to identify persons first. One straightforward way to localize persons is to parametrize the upper body bounding box based on the position and scale of a detected face. In addition to this simple method, we also use the more sophisticated Calvin upper body detector [9], to generate additional bounding box hypotheses. All generated hypotheses are then combined through a non maximum suppression, in which hypotheses originating from the calvin upper body detector are scored higher than hypotheses coming only from the face position.

3.2 Features

In terms of feature extraction and coding, we follow a s-o-a image classification pipeline:

Feature extraction Within the bounding box of an upper body found in the previous step, we extract a number of features including SURF [1], HOG [6], LBP [21], Self-Similarity (SSD) [23], as well as color information in the L*a*b space. All of those features are densely sampled on a grid.

Coding For each of the feature types except LBP, a code book is learnt by using K-Means[1]. Subsequently all features are vector quantized using this code book.

Pooling Finally, the quantized features are then spatially pooled with spatial pyramids [18] and max-pooling applied to the histograms.

For each feature type this results in a sparse, high-dimensional histogram.

3.3 Apparel Type Learning

After person detection and feature extraction, we use a classifier for the final clothing type label prediction. Since we face a multi-class learning problem with high-dimensional input and many training samples, we use Random Forests [2] as our classification method. Random Forests (RF) are fast, noise-tolerant, and inherently multi-class classifiers that can easily handle high-dimensional data, making them the ideal choice for our task.

[1] We used $1,024$ words for SURF and HOG, 128 words for color and 256 words for SSD, respectively.

A RF is an ensemble of T decision trees, where each tree is trained to maximize the information gain at each node level, quantified as

$$\mathcal{I}(\mathcal{X}, \tau) = H(\mathcal{X}) - \left(\frac{|\mathcal{X}_l|}{|\mathcal{X}|} H(\mathcal{X}_l) + \frac{|\mathcal{X}_r|}{|\mathcal{X}|} H(\mathcal{X}_r) \right) \tag{1}$$

where $H(\mathcal{X})$ is the entropy for the set of samples $\mathcal{X}$ and τ is a binary test to split $\mathcal{X}$ into subsets $\mathcal{X}_l$ and $\mathcal{X}_r$. Class predictions are performed by averaging over the class leaf distributions as $p(c|L) = \frac{1}{T} \sum_{t=1}^{T} P(c|l_t)$ with $L = (l_1, \ldots, l_T)$ being the leaf nodes of all trees. The term *random* stems from the fact that during training time only a random subset over the input space is considered for the split tests τ and each tree uses only a random subset of the training samples. This de-correlates the trees and leads to lower generalization error [2].

Following the idea of Yao *et al.* [30], we use strong discriminative learners in the form of binary SVMs as split decision function τ. In particular, if $\boldsymbol{x} \in \mathcal{R}^d$ is a d-dimensional input vector and $\boldsymbol{w}$ the trained SVM weight vector, an SVM node splits all samples with $\boldsymbol{w}^T \boldsymbol{x} < 0$ to the left and all other samples to the right child node, respectively. In order to enable the binary classifier to handle multiple classes, we randomly partition these classes into two groups. While training, several of those binary class partitions are randomly generated. For each grouping, a linear SVM is trained for a randomly chosen feature channel. Finally the split that maximizes the multi-class information gain $\mathcal{I}(\mathcal{X}, \boldsymbol{w})$, measured on the real labels, is chosen as splitting function, *i.e.*, $\hat{\boldsymbol{w}} = \text{argmax}_{\boldsymbol{w}} \mathcal{I}(\mathcal{X}, \boldsymbol{w})$

Random forests are highly discriminative learners but they can also overfit easily to the training data if too few training samples are available [3], an effect that tends to intensify if SVMs are used as split nodes. Therefore, in the following, we propose two extensions to the random forest algorithms of [2] and [30] that shall improve the generalization accuracy but keep the discriminative power.

Large Margin While training, different split functions often yield the same information gain. Breaking such ties is often done by randomly selecting one split function out of the best performing splits. In this work we introduce an additional selection criterion to make more optimal decisions. It is inspired by Transductive Support Vector Machines (TSVM) [15], where the density of the feature space around the decision boundary is taken into account while solving the optimization problem for $\boldsymbol{w}$. Opposed to TSVMs however, we do not use this information while optimizing $\boldsymbol{w}$, but go after minimal feature density (or largest margin) as an additional optimality criterion for the split selection. In other words, if several split functions perform equally well, the density of the feature space within the margin is taken into account, estimated as:

$$\mathcal{I}^m(\mathcal{X}, \boldsymbol{w}) = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} \max\left(0, 1 - |\boldsymbol{w}^T \boldsymbol{x}|\right) \tag{2}$$

with the decision boundary $\boldsymbol{w}$ and training examples $\mathcal{X}$. Then the optimal split can be chosen by minimizing the above equation *w.r.t.* $\boldsymbol{w}$, *i.e.*, the optimal split function is given by $\hat{\boldsymbol{w}} = \text{argmin}_{\boldsymbol{w}}\ \mathcal{I}^m(\mathcal{X}, \boldsymbol{w})$.

Transfer Forests Another option to improve the generalization power of Random Forest is to use more training samples. However, it is often not easy to acquire more training samples along with good quality annotations. One way to achieve this is to outsource the labeling task to crowdsourcing platforms, such as Mechanical Turk [25]. Yet, this demands careful planning for an effectively designed task and an adequate strategy for quality control. It can also not be used to annotate confidential data. Therefore, previous work also studied the extension of RFs to semi-supervised learning [19, 5] in order to benefit from additional unlabeled data, which is usually cheap to collect.

For our task, we can use text-based image search engines to gather large amounts of images, such that the returned images come already with prior labels $\hat{c}$. For instance, we can type *cotton, black, pastel*, *etc.* to get clothing images that probably exhibit these attributes. Similarly, we can type *jacket, t-shirt, blouse*, *etc.* to get images containing our target type classes. On the downside, these images may contain high levels of noise and originate from variable source domains. Thus, not all samples might fit to our task and $\hat{c}$ cannot be considered to flawlessly correspond to the *real* label c.

Therefore, we extend Random Forests to *Transfer Learning* (TL) [22], which tries to improve the classification accuracy in scenarios where the training and test distributions differ. In particular, assume having access to M samples from the labeled target domain $\mathcal{X}^l$ (*e.g.* a manually labeled and quality controlled data set) along with their labels $\mathcal{C}$. Additionally, in TL one has access to N samples from an auxiliary domain $\mathcal{X}^a$ (*e.g.* Google image search) together with their labels $\hat{\mathcal{C}}$. The task of TL is to train a function $f : \mathcal{X} \rightarrow \mathcal{C}$ that performs better on the target domain via training on $\mathcal{X}^l \cup \mathcal{X}^a$ than solely relying on $\mathcal{X}^l$. There exist many approaches to TL (*c.f.* [22]) and its usefulness has also been demonstrated in various vision domains, *e.g.* [26, 17]. We present here a novel variant of transfer learning for Random Forests as this is our main learner.

To this end, we exploit the idea that although the source and target distributions might be different, some of the source samples $\boldsymbol{x}_i \in \mathcal{X}^a$ can still be useful for the task and should thus be incorporated during learning, while samples that may harm the learner should be eliminated. In order to accomplish such an *instance-transfer* approach (*c.f.* [22]) for Random Forests, we augment the information gain of Eq. 1 to become

$$\mathcal{I}^*(\mathcal{X}, \boldsymbol{w}) = (1 - \lambda) \cdot \mathcal{I}(\mathcal{X}^l, \boldsymbol{w}) + \lambda \cdot \mathcal{I}(\mathcal{X}^a, \boldsymbol{w}), \tag{3}$$

where the first term corresponds to Eq. 1 and $\mathcal{I}(\mathcal{X}^a, \boldsymbol{w})$ measures the information gain over the auxiliary data. The *overall* influence of $\mathcal{X}^a$ is controlled via the steering parameter $\lambda \in [0, 1]$.

The information gain $\mathcal{I}$ relies on the standard entropy measure $H(\mathcal{X}) = -\sum_c p_c \log(p_c)$ with $p_c = \frac{1}{|\mathcal{X}|} \sum_i \varphi(\boldsymbol{x}_i, \mathcal{X}_c)$, where $\varphi(\boldsymbol{x}_i, \mathcal{X}_c)$ is the indicator function and is defined as

$$\varphi_l(\boldsymbol{x}_i, \mathcal{X}_c) = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \in \mathcal{X}_c \\ 0 & \text{else,} \end{cases} \tag{4}$$

Colors	Patterns	Materials	Structures	Looks	Persons	Sleeves	Styles	
beige	animal print	cotton	frilly	black/white	child	long	20's	nerd
black	zebra	denim	knitted	colored	boy	short	50's	outdoor
blue	leopard	fur	ruffled	gaudy	girl	none	60's	preppy
brown	argyle	lace	wrinkled	pastel	female		70's	punk
gray	checkered	leather			male		80's	rock
green	dotted	silk					90's	romantic
orange	floral	tweed					bohemian	sports
pink	herringbone	wool					business	wedding
purple	houndstooth						casual	spring
red	paisley						dandy	summer
teal	pinstripes						hip hop	autumn
white	plaid						hippie	winter
yellow	print						mod	
	striped							
	tartan							

Table 1: List of attribute categories and the attributes therein.

with $\mathcal{X}_c$ representing the set of samples for class c. Note, the auxiliary dataset influences only the selection of the trained SVM for each node, but it is not used during the actual training of the SVM.

3.4 Clothing Attribute Learning

The slight differences in appearance of apparel are often orthogonal to the type of clothing, *i.e.*, the composition of colors, patterns, materials and/or cuttings often matter more than the information, that a particular cloth is *e.g.* a sweater. A common way to include such kind of information is to represent it by semantic attributes. We define eight attribute categories with in total 78 attributes as shown in Table 1. The training of the attributes happens for each of the eight attribute categories separately. Within each of those, the different attributes are considered mutually exclusive. Thus, within a category, we train for each attribute a one-vs-all linear SVM on the features described in Section 3.2.

4 Data Sets

For both tasks – classification of clothes and attribute detection – we collected two distinct data sets. Additionally, an auxiliary data set $\mathcal{X}^a$ was automatically crawled to be used for our transfer learning extension for Random Forests.

4.1 Apparel Type

To the best of our knowledge, there is no publicly available data set for the task of classifying apparel or clothing, respectively. The large variety of different clothing types and, additionally, the large variance of appearance in terms of colors, patterns, cuttings *etc.* necessitate that a large data set be used for training a robust classifier. However, assembling a comprehensive and high quality data set is a daunting task.

Category	Images	Boxes
Long dress	22,372	12,622
Coat	18,782	11,338
Jacket	17,848	11,719
Cloak	15,444	9,371
Robe	13,327	7,262

Category	Images	Boxes
Suit	12,971	7,573
Undergarment	10,881	6,927
Uniform	8,830	4,194
Sweater	8,393	6,515
Short dress	7,547	5,360

Category	Images	Boxes
Shirt	3,140	1,784
T-shirt	2,339	1,784
Blouses	1,344	1,121
Vest	1,261	938
Polo shirt	1,239	976
Total	145,718	89,484

Table 2: Main classes and number of images per class of the benchmark data set

Luckily, ImageNet [8], a quality controlled and human-annotated image database that is hierarchically organised according to WordNet, contains many categories (so called *synsets*) related to clothes. Nevertheless, a closer look at ImageNet's (or rather WordNet's) structure reveals that clothing synsets often do not correspond to the hierarchy a human would expect. Therefore we handpicked 15 categories and reorganized ImageNet's synsets accordingly. Due to how ImageNet is built, some images are ambiguous and quite a few are very small. As a cleaning step, we preprocess each image as described in Section 3.1. If no face or upper body can be detected, a centered bounding box is assumed as ImageNet also contains web shop images that show pieces of clothing alone. The resulting bounding boxes smaller than 91 pixels were discarded.

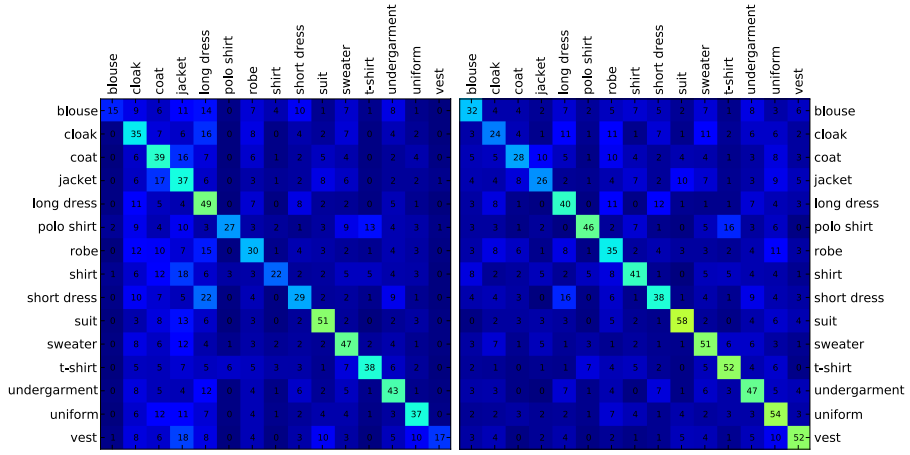
An overview over the categories can be found in Table 2. As a contribution of this paper, we make the details of the data set publicly available so that the community can use this subset of ImageNet as a benchmark for clothing classification.

4.2 Transfer Forest

For each of the clothing type classes, we collected the auxiliary data set $\mathcal{X}^a$ by querying Google image search multiple times with different word combinations for the same category (*e.g.* "sweater women", "sweater men" or *e.g.* "long dress formal", "long dress casual") such that the retrieved data contains some variation. We again restricted the result to photos of a minimum size of 300×400 pixels and performed no further supervision on the $42,624$ downloaded images.

4.3 Attributes

In order to train classifiers for visual attributes, we need a special training data set just for this task. While ImageNet provides images with attribute annotation, it only covers a small part of our defined attributes (*c.f.* Table 1). Moreover, ImageNet provides attribute annotation only for a subset of its synsets, thus making this data source not appropriate for learning our selection of attributes. Therefore we construct a third distinct data set by automatically crawling the Web. For each attribute, we let an automated script download at least 200 images using Google image search and restricted results to photos of a minimum size of 300×400 pixels. For each attribute, the script generates a query composed of the attribute label and one of the words "clothing", "clothes" or "look" as query keyword. No further supervision was applied to those $25,002$ images after downloading.



(a) (b)

Fig. 2: Confusion matrix of our clothing classes for the best performing SVM classifier on the left side and the proposed Transfer Forest on the right side.

5 Experiments

In this section we present experiments to evaluate our algorithm quantitatively. First we show the results for the apparel type part, then the results for the attribute part. An overview of the relevant results can be found in Table 3.

5.1 Apparel Types

We present three sets of numerical evaluations. First, using the apparel type data set introduced in Section 4.1, we trained a SVM as a baseline. Then, the results for Random Forest with SVMs as split nodes are shown. Finally, the effectiveness of the proposed Transfer Forest is demonstrated.

SVM Baseline As a baseline experiment we train a one-vs-all linear SVM for each clothing type category. We evaluated all possible feature combinations, and also L_2 regularized hinge as well as L_1 regularized logistic loss. For the evaluation of the feature combinations, the histograms of the different extracted features (*c.f.* Section 3.2) were simply concatenated. We used 80 % of the bounding boxes of each class for training and the remaining part for testing. Finally, L_1 regularized logistic loss using all available features yielded with 35.03 % average accuracy the best performance. The confusion matrix is shown in Figure 2a. There is a clear bias towards overrepresented classes.

Random Forest To evaluate the performance of the random forest framework we define the following protocol: again we use 80 % of the images of each type class of the data set for training and the remainder for testing. Each tree has

Learner	**Avg. Acc.** [%]
One vs. all SVM	35.03
RF	38.29
RF + large margin on $\mathcal{X}^l$	39.31
RF on $\mathcal{X}^l \cup \mathcal{X}^a$, naïve	36.27
RF on $\mathcal{X}^l \cup \mathcal{X}^a$, Daumé [7]	35.00
Transfer Forest	41.36

Table 3: Classification performance measured as average accuracy over all classes on our benchmark data set for different methods.

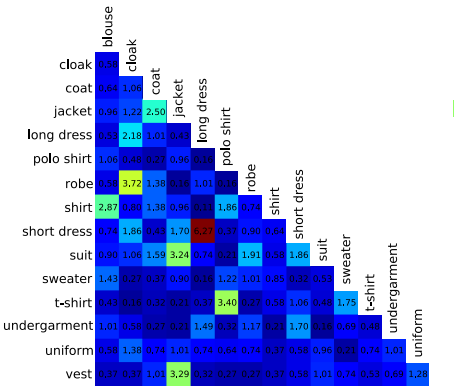


Fig. 3: Percentage of co-occurring classes in the deepest split nodes. Note how semantic similar classes often occur together.

been trained on a random subset of the training set, which contains 500 images for each class, thus $7,500$ images in total.

While training, we generate at each node 50 linear SVMs with the feature type and the binary partition of the class labels chosen at random. Other than what Yao *et al.* [30] propose, we do not randomly sample subregions within the bounding boxes, but use the spatially pooled histograms (*c.f.* Section 3.2) as input for the SVMs. Each tree is then recursively split until either the information gain stops increasing, the numbers of trainings examples drops below 10, or a maximum depth of 10 is reached. In total we trained 100 trees out of which we created 5 forests by randomly choosing 50 trees. The final result is then averaged over those 5 forests to reduce the variance of the results.

Baseline With 38.29 % average accuracy, our Random Forest with SVMs as split functions outperforms the plain SVM baseline (35.03 %) significantly. It handles the uneven class distribution much better as can be seen in Figure 2b. These results confirm our expectation that a Random Forest is a suitable learner for our task. Figure 3 shows the co-occurrences of the different classes at the deepest levels of the tree. Interestingly, semantic similar classes often occur together.

Large Margin Having strong discriminative learners as decision nodes renders the information gain as optimization criterion often as too weak a criteria: several different splits have the same information gain. In this case, choosing the split with the largest margin amongst the splits with the same information gain on the training data seems beneficial as performance increases about 1 % compared to the Random Forest baseline.

Transfer Forest To assess the performance of our approach, we follow the protocol defined in the baseline Random Forest evaluation. The parameter λ of Eq. 3 was varied between $0 < \lambda < 1$ in 0.05 steps. Unfortunately, no distinct

Category	Acc. [%]	Reg.	Loss	Surf	Hog	Color	Lbp	Ssd
Looks (4)	71.63	L_2	hinge	×		×	×	×
Sleeves (3)	71.52	L_1	logistic	×			×	×
Persons (5)	64.07	L_2	hinge	×	×	×		×
Materials (8)	63.97	L_1	logistic	×	×	×	×	×
Structure (4)	63.64	L_1	logistic					×
Colors (13)	60.50	L_2	hinge			×	×	×
Patterns (15)	49.75	L_1	logistic					×
Styles (25)	37.53	L_2	hinge					×

Table 4: Best average accuracy for each attribute category with the corresponding features. The number of attributes per category is denoted in in parentheses.

best choice for λ is obvious. Yet, our approach yields minimum and maximum improvement of 2.18 % and 3.09 % over the baseline Random Forest, respectively. On average, any choice of λ increased the performance about 2.45 % in that $0 < \lambda < 1$ interval.

To validate our assumption that transfer learning is beneficial in this case, we also trained a forest on the union of $\mathcal{X}^a$ and $\mathcal{X}^l$, thus treating the auxiliary images as they would stem from the regular data set. In this case, the performance significantly drops below that of the baseline Random Forest.

As a sanity check, we also compared to another domain adaptation method presented by Daumé [7], which comes at a cost of tripling the memory requirements and substantially longer training times, as the feature vectors that are passed on to the SVM are thrice as large. Moreover, also this approach does not improve the performance over that of the baseline Random Forest (see Table 3). This *(i)* highlights the importance of using transfer learning when incorporating data from different domains for our task and *(ii)* also shows that Random Forests are useful for transfer learning.

5.2 Attributes

For training and testing we assume that within a given attribute category (*e.g.* colors or patterns) attributes (*e.g.* red, white or zebra,dotted) are mutually exclusive. Furthermore attribute with the least samples constrains the number of samples for all other attributes in the same category. With this, out of the $25,002$ downloaded images, $16,155$ were used for testing and training the attributes. The data set was split in 75 % of samples for training and 25 % for testing.

We extract the features as described in Section 3.2 and train several linear one vs. all SVMs [10] with all possible feature combinations as well as with L_1 regularized logistic loss and L_2 regularized hinge loss. For the experiments, the cost was set at $C = 1$ as the classification performance stayed invariant in combination with max pooling. Results are shown in Table 4. The classification accuracy ranges between about 38 % and 71 % depending on the category. Of course it is expected that attribute categories with less possible values (*e.g. sleeves*) perform better than those with many (*e.g. patterns*). Nevertheless a classification task such as the sleeve length is not trivial and performs surprisingly well. On the other hand color and pattern classification could probably be

improved. It appears the classifier is distracted too much by background data present within the bounding box. A simple fix would be to sample data only from a small part from the center of the bounding box for categories such as colors or patterns. A large category such as *styles* with many "fuzzy" or "semantic" attribute values such as "punk" or "nerd" poses of course a challenge to even an advanced classifier.

5.3 Qualitative Results

In Figure 4 some example outputs of our full pipeline are shown. Note how we are able to correctly classify both style and attributes in many situations. This would allow a system to come up with the desired description combining attributes and style. For instance for the first example in the middle row a description such as "Girl wearing a pastel spring short dress without sleeves" could be generated. Also note how the random forest robustly handles slight variations in body pose for cloth classification (*e.g.*, in the top right example). Of course, accurate detection of the upper body is crucial for our method, and many of the failure cases are due to false upper body detections (example in the 3^{rd} row, 3^{rd} image). Another source for confusion are ambiguities in the ground truth (3^{rd} row, 1^{st} and 2^{nd} example). For attributes performance is mainly challenged by distracting background within the bounding box or lack of context in the bounding box (*e.g.*, 2^{nd} row, 2^{nd} example).

6 Conclusion

We presented a complete system, capable of classifying and describing upper body apparel in natural scenes. Our algorithm first identifies relevant image regions with state of the art upper body detectors. Then multiple features such as SURF, HOG, LBP, SSD and color features are densely extracted, vector quantized and pooled into histograms and fed into two higher level classifiers, one for classifying the type and one for determining the style of apparel. We could show that the Random Forest framework is a very suitable tool for this task, outperforming other methods such as SVM. Since there are many apparel images available on the web but they often come with noise or unrelated content, we extended Random Forests to transfer learning. While this improved the accuracy for the task at hand, we believe that also other vision applications using Random Forests might benefit from this algorithmic extension. We also introduced a challenging benchmark data set for the community, comprising more than $80,000$ images for the 15 clothing type classes. On this data set, our Transfer Forest algorithm yielded an accuracy of 41.36 %, when averaged over the type classes. This represents an improvement of 3.08 % compared to the base line Random Forest approach and an improvement of 6.3 % over the SVM baseline.

Acknowledgement. We thank Fabian Landau for his excellent work on segmentation. This project has been supported by the Commission for Technology and Innovation (CTI) within the program 12618.1.

Fig. 4: Some example output of our pipeline. The header denotes the ground truth class. Each example shows the detected bounding box and the output of the type classifier. On the left side of each example, the output of the most confident attribute classifier for each attribute group is shown.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. ICCV (2006)
2. Breiman, L.: Random forests. Machine Learning. (2001) 5–32
3. Caruana, R., Karampatziakis, N., Yessenalina, A.: An empirical evaluation of supervised learning methods in high dimensions. ICML (2008)
4. Chen, H., Xu, Z.J., Liu, Z.Q., Zhu, S.C.: Composite Templates for Cloth Modeling and Sketching. CVPR (2006)
5. Criminisi, A., Shotton, J., Konukoglu, E.: Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report MSR-TR-2011-114, Microsoft Research (2011)

6. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. CVPR (2005)
7. Daumé, H.: Frustratingly easy domain adaptation. Annual meeting-association for computational linguistics. Volume 45. (2007) 256
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. CVPR (2009)
9. Eichner, M., Ferrari, V.: CALVIN Upper-body detector for detection in still images
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. JMLR **9** (2008)
11. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. CVPR (2009)
12. Ferrari, V., Zisserman, A.: Learning visual attributes. NIPS (2008)
13. Gallagher, A.C.: Clothing cosegmentation for recognizing people. CVPR (2008)
14. Hu, Z., Yan, H., Lin, X.: Clothing segmentation using foreground and background estimation based on the constrained Delaunay triangulation. Pattern Recognition **41** (2008)
15. Joachims, T.: Transductive inference for text classification using support vector machines. ICML (1999)
16. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. ICCV (2009)
17. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. CVPR (2009)
18. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. CVPR (2006)
19. Leistner, C., Saffari, A., Santner, J., Bischof, H.: Semi-Supervised Random Forests. ICCV (2009)
20. Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., Yan, S.: Street-to-Shop: Cross-Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set. CVPR (2012)
21. Ojala, T., Pietikainen, M., Harwood, D.: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. ICOR (1994)
22. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE (2010)
23. Shechtman, E., Irani, M.: Matching Local Self-Similarities across Images and Videos. CVPR (2007)
24. Song, Z., Wang, M., Hua, X.s., Yan, S.: Predicting occupation via human clothing and contexts. ICCV (2011)
25. Sorokin, A., Forsyth, D.: Utility data annotation with amazon mechanical turk. Workshop on Internet Vision. (2008)
26. Stark, M., Goesele, M., Schiele, B.: A shape-based object class model for knowledge transfer. ICCV (2009)
27. Wang, N., Ai, H.: Who Blocks Who: Simultaneous clothing segmentation for grouping images. ICCV (2011)
28. Wang, X., Zhang, T.: Clothes search in consumer photos via color matching and attribute learning. MM, ACM Press (2011)
29. Yamaguchi, K., Kiapour, H., Ortiz, L., Berg, T.L.: Parsing Clothing in Fashion Photographs. CVPR (2012)
30. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. CVPR (2011)