Look Closely: Learning Exemplar Patches for Recognizing Textiles from Product Images

Quoc Huy Phan[1], Hongbo Fu[1], Antoni B. Chan[2]

[1]School of Creative Media
[2]Department of Computer Science
City University of Hong Kong

Abstract. The resolution of product images is becoming higher dues to the rapid development of digital cameras and the Internet. Higher resolution images expose novel feature relationships that did not exist before. For instance, from a large image of a garment, one can observe the overall shape, the wrinkles, and the micro-level details such as sewing lines and weaving patterns. The key idea of our work is to combine features obtained at such largely different scales to improve textile recognition performance. Specifically, we develop a robust semi-supervised model that exploits both *micro textures* and *macro deformable shapes* to select representative patches from product images. The selected patches are then used as inputs to conventional texture recognition methods to perform texture recognition. We show that, by learning from human-provided image regions, the method can suggest more discriminative regions that lead to higher categorization rates (+5-7%). We also show that our patch selection method significantly improves the performance of conventional texture recognition methods that usually rely on dense sampling. Our dataset of labeled textile images will be released for further investigation in this emerging field.

1 Introduction

Online shopping is changing the way people buy goods. It offers consumers the quickest way to check out a product's price and appearance without visiting an actual shop. In recent years, online fashion stores have provided high-resolution images to advertise their products, since users often pay attention to every detail, like materials, sewing lines, weaving quality, decorators, etc. Zoom-in functions may also be available to enable easy examination of the fine details of products. From a pattern recognition perspective, these high quality images are potentially useful for multiple tasks. For instance, one can use edge feature at macro scales to match products' global shape, while features at micro scales can be used for recognizing textures and details. More interestingly, one may combine features from different scales to reliably recognize objects.

One problem that might benefit from using a multi-scale approach is that of real-world textile recognition. Considering the case of leather, the best cue to identify its instances is the macro shapes of the wrinkles, since leather has relatively smooth surface with few color patterns. Whereas, fur and fleece have

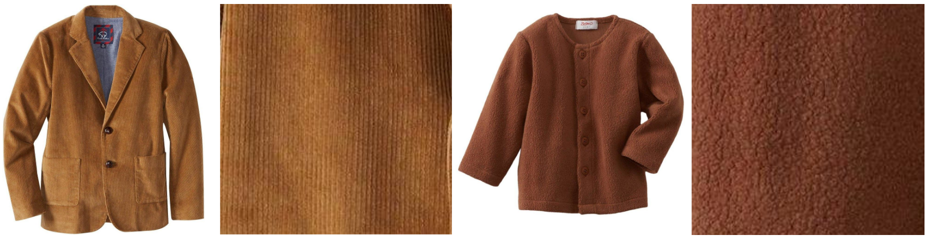

Fig. 1. Different scales reveal different characteristics of textiles. In this case, the micro scale exposes more discriminative features. Columns 1 and 2: Corduroy. Columns 3 and 4: Fleece.

rather rough micro structures with few or no wrinkles. To recognize these textiles, a better way should be to examine their texture features. See Figure 1 for such an example, where macro shapes are largely similar while micro textures are very different. Motivated by this observation, we design a patch selection method that takes into account the evidences from two different scales (micro- and macro-scale) to identify discriminative patches in textile images. Having good patches identified, a number of related tasks such as retrieval and classification can be reliably carried out.

Despite the usefulness of higher resolution product images discussed above, existing works in the field of garment recognition (e.g., [1–4]) typically take small-sized images as input and thus ignore the information at micro scales. This limitation seriously restricts the understanding of garment textile, which is important to customers. On the other hand, texture recognition has been studied extensively in the field of computer vision. However, effective solutions (e.g., [5–7]) are typically demonstrated only on datasets of nicely cropped images. In contrast, real-world textures often appear at unknown positions and could be hidden in very large images. This makes it difficult to directly apply such solutions for recognizing textures in real-world contexts. One may expect that, applying these methods to a set of selective patches will improve the performance. However, as we will show in our experiments, existing patch selection methods (e.g., [8, 9]) do not work well. In fact, the classification accuracies when using patches selected by these methods are even worse than those when using dense sampling on whole images. In contrast, our selection method provides patches that work well with traditional texture recognition methods, resulting in significantly higher accuracies.

In this paper, we introduce an efficient discriminative model to identify representative textured regions from product images. The key of our approach is an automatic patch selection process (Figure 2 (A)), which is governed by both macro shape and micro texture. We assess the quality of the model by performing textile categorization on the patches selected by our method. Figure 2 (B) shows an overview of our categorization process. In experiments, categorization performance was significantly better when using our patch selection method, as compared to other methods like manual selection, SIFT detector selection [8], objectness measure [9], and dense sampling. We also introduce a simple texture

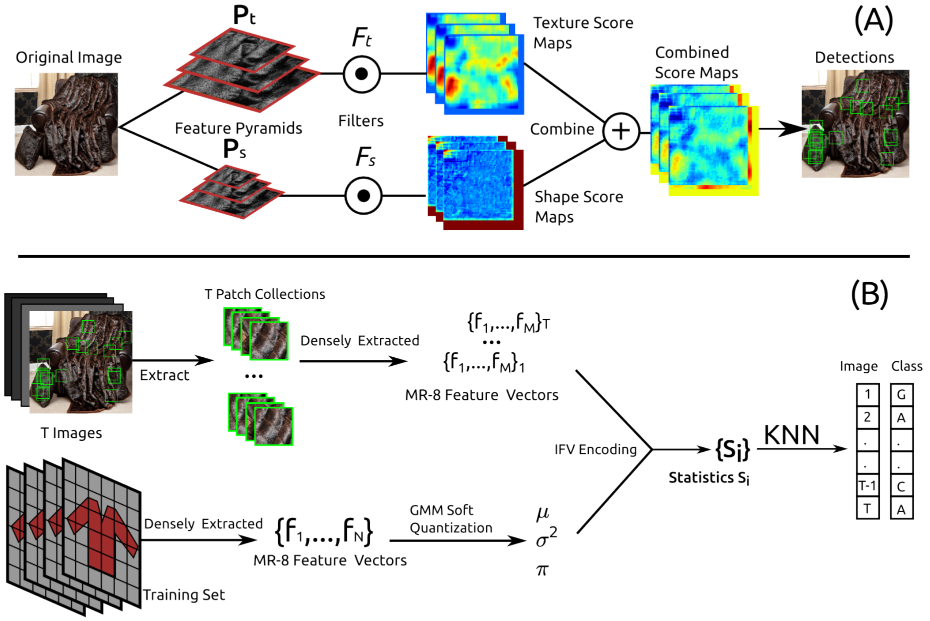

Fig. 2. (A) Patch selection process. First, two feature pyramids $\mathbf{P}_t$ and $\mathbf{P}_s$ are constructed. Next, they are convolved with two filters F_t and F_s to obtain two score maps. Finally, the score maps are combined and highly scored regions are selected. (B) texture categorization process. First, images in the training set are densely sampled and soft quantized to obtain (μ, σ^2 and π). Second, the patch selection algorithm chooses good patches for each image in the whole dataset. Third, feature vectors from those patches are densely extracted and encoded using IFV. Finally, the statistics $\{\mathbf{S}_i\}_{i=0}^{T}$ and the labels of the training data are used to classify novel images.

feature that naturally fits into our framework, and has competitive performance when used with traditional classification models. To summarize, our findings are:

- An efficient discriminative model for texture-aware patch selection.
- A novel feature that is competitive with current state-of-the-art.
- A new textile dataset consisting of 480 samples (1000 × 1000 pixels), grouped into 8 classes and annotated with 9600 labels.

2 Related Work

This paper is connected to three different research fields: patch selection, texture recognition, and garment analysis.

2.1 Patch Selection

There exist different ways to select patches of interest in images. For example, patches can be defined at key points, which can be detected using techniques such as SIFT [8] and SURF [10]. Existing key point detection methods focus mainly on areas with strong edge response but ignore "smooth" regions which may still contain informative texture features. Our patch selection method bears some resemblances to saliency detection, which aims to measure visual importance of individual pixels in an image by evaluating their uniqueness with respect to larger areas [11–14, 9]. Instead of seeking for pixels which are visually "stand-out" from the background, we intend to find regions whose both *micro texture* and *macro shape* are distinguishable from the rest of the image. These properties only coexist in high resolution images, where one can observe both small-scale textures and large-scale deformable surfaces. The advantages of our method over the methods based on key point detection [8] and saliency detection [9] on our dataset will be shown in Section 5.

Our work is also related to object detection (e.g., [15–17]), which aims to detect the existence of a particular object in an image. One common approach is to use a sliding window that scores every region in an image by considering its neighborhoods. We adopt a similar approach to [15] by effectively employing Latent-SVM to train our patch selector. The key differences are: (a) we do not consider patch deformation with respect to a "root" filter, making our selector work on any type of object; (b) we simultaneously consider both texture and shape at different scales, which will be detailed in Section 4.1.

2.2 Texture Recognition

Texture recognition is already a mature field of computer vision. Dozens of techniques and datasets have been developed throughout the last decades. In terms of representation, one of the earliest works employs a bank of wavelets computed at various scales and rotations to capture the characteristics of textures [18, 19, 5]. Our work adopts the MR8 filter banks introduced by Varma and Zisserman [5] to represent textures. We choose this representation for its simplicity and compactness. Varma and Zisserman [20] challenged the role of filter banks by effectively replacing them with simple patches extracted densely on a grid. Approaches based on local binary patterns (LBP) [21, 22] also achieved great performance on standard datasets. Recently, Sifre and Mallat [6] used scattering convolutional network to extract very discriminative texture features and set the new state-of-the-art in texture classification (Rotation, Scaling and Deformation Invariant Scattering – RSDS). We will show the results of RSDS on our dataset in Section 5.

A common framework for texture recognition is the *texton* framework which has been first introduced by [18]. To describe texture, texton-based methods first build a dictionary of textons (visual words), which summarize basic components that make up texture appearance. Next, a histogram of visual words is constructed for every image by assigning a texton label to every pixel in that image. Recently Cimpoi et al. [7] presented a texture recognition method enabled by the Fisher Vector (FV). Instead of assigning hard labels to the pixels, the Fisher Vector uses soft labels and higher order statistics to describe texture. Their method achieves the state-of-the-art performance on very challenging datasets, such as FMD [23] and KTH-TIPS2 [24].

Researchers recently introduced the idea of texture *categorization* instead of *classification* [24, 25]. While classification aims at recognizing instances of a texture, categorization generalizes them to a categorical level. Such generalizations enable the recognition of novel texture instances that do not appear in the training set. KTH-TIPS2 [24] is the first dataset to include different texture instances. We consider our work as texture categorization instead of classification.

Another trend is to recognize textures, since they appear in real-world contexts [25]. A prominent work of Liu et al. [26] is one of the first to address this problem. They introduced a very challenging dataset, namely the FMD dataset, which contains a wide range of texture images collected from Flickr. A significant contribution of their work is a psychological study on how humans recognize texture. They found that global shape is an equally important factor in texture recognition performance as local texture. Their finding forms a basis for our choice of using both shape and local texture for patch selection.

2.3 Garment Recognition

Garment understanding or recognition [27–29] has been an active research topic in recent years, and gives rise to many practical applications like product suggestions [30, 4], genre classification [31], outfit recommendation [32], and clothing retrieval [1]. Our work is related to clothing attribute prediction [3, 2]. Textile recognition is largely ignored in garment recognition, regardless of the fact that textile material plays a central role in customer decision making. For example Chen et al. [2] does not consider material at all. Liu et al. [1] uses material labels for performance measurement only. The work of [3] does include a limited number of materials as clothing attributes. However it does not show any textile prediction results and the reported classification rate is rather low (41%).

3 Textile Dataset

We collected a large number of high-resolution images from online shopping and image sharing websites like Flickr, Polyvore and Amazon. The size of the images was at least 1000 pixels in both dimensions. While a large part of our dataset has relatively clean backgrounds, we chose to also include samples that contain objects as it appear in real-world contexts, e.g., with the human body and background. By doing so, we expect that our method should be able to distinguish between clothing and irrelevant regions such as skin, hair and other unaccounted materials. Although most of the objects are garments, we also included a number of non-garment objects like blankets and pillows. We then cropped and scaled all images to a regular size of 1000×1000 pixels (Fig. 3 (Left)). Images that contain different textile categories are removed to ensure fair testing and training. Unlike other real-world datasets like FMD [23], which provide masks to filter out unrelated regions, we assume there is a certain amount of noise in our samples. Finally, we had a collection of 480 high-resolution images which belong to 8 different textile categories: **Boucle, Fur, Leather, Lace, Knitted, Denim, Corduroy** and **Fleece** (60 samples per each).

Next, the images were contrast normalized and converted to grey-scale. All of our experiments are conducted on grey-scale images and do not consider color. Although color can be a good clue for texture recognition, it is also a source of confusion in the case of clothing since any color can be printed on any material with the current technologies. Fig. 3 (Left) shows the diversity of our dataset. The Fleece category contains a wide range of printed patterns, while the Fur and Leather categories include samples from very different objects like gloves, blankets, shoes.

To supervise the recognition process, we manually labelled 20 patches of size 128 × 128 pixels in each image which contain representative textile instances. Fig. 3 (Right) shows sample patches from the Fur category. It can be seen that different objects have very different appearance at both micro and macro levels even though they all belong to the same category. More challenging, object surfaces are not flat and vary largely with lighting conditions.

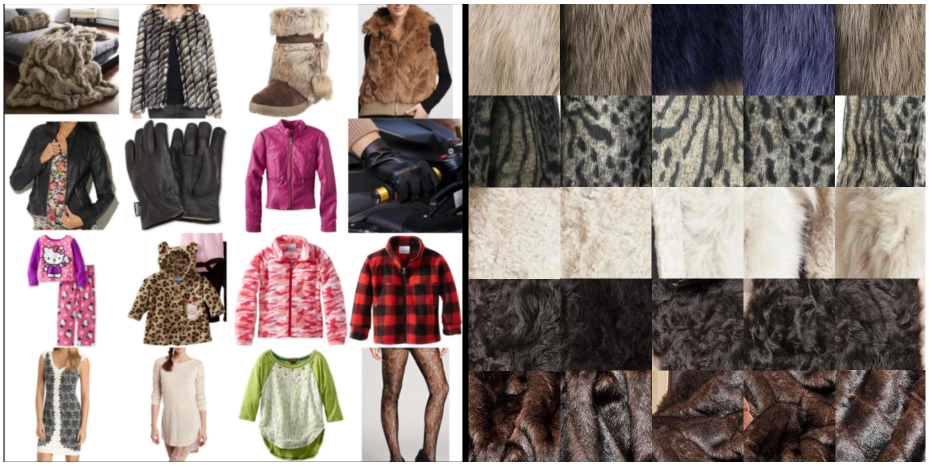

Fig. 3. (Left) Samples from Fur, Leather, Fleece and Lace (from top to bottom). (Right) Varieties of the Fur instances, with patches on the same row belonging to the same objects.

4 Model Learning and Patch Selection

We approach the problem of textile recognition[1] by treating each textile category as a mixture of textile instances. This idea came from the fact that a textile (with textured surface) has different appearances in the real world, depending on its functionality and context. Taking denim as an example, when denim is used to produce a jacket, it can be cut and sewed to make pockets, collars and sleeves which are apparently different from those elements on a pair of denim pants. By modelling each textile instance as a mixture component, we allow the components to compete with each other when fitting onto a novel image. If one of the components matches one part of the image with high confidence, we conclude that the whole item should belong to a specific textile category. This is intuitively natural to the way that human beings recognize textile: when we look at a piece of clothing and are not sure which material it is made of, we

[1] We use the term “recognition” to refer to both patch selection and categorization.

usually focus on the most distinctive part of it, e.g., regions without distracting decorations or details. In our recognition framework, both texture and shape are used for patch selection while only texture is used for categorization. Thus, the textile categorization step is simply *texture categorization* and we will use these two terms interchangeably.

Fig. 2 (A) portrays the patch selection process. Starting from an image without annotations, we construct two feature pyramids $\mathbf{P}_t$ and $\mathbf{P}_s$, which represent texture and shape at multiple scales, respectively. The feature pyramids are then independently convolved with two filters F_t and F_s to obtain the score maps. The score maps are then combined into a final score map, which specifies good and bad regions to sample. Section 4.1 will discuss the score function in details.

Our textile recognition problem can be posed as a multi-instance learning problem, which can be addressed using different machine learning tools such as [33–35]. We employ the Latent-SVM [15] to train the filters F_t and F_s in a one-versus-all fashion.

Once we have selected the patches using the score map, it is straightforward to perform texture categorization. We use K-nearest neighbours in conjunction with bag-of-visual-words encoding to categorize textures. Sections 4.2 and 5 will explain the processes in details.

4.1 Learning Scheme

We use the Latent-SVM to learn the best filter parameters for each textile category. We next discuss the Latent-SVM, the associated score function, and our initialization procedure.

Multi-Instance Learning with Latent-SVM Since our patch selection problem is posed as a multiple-instance learning problem, we use Latent-SVM (L-SVM) [15] to learn the filter parameters F_t and F_s. Similar to a conventional SVM, L-SVM scores an example x with a function of the form

$$f_\beta(x) = \max_{z \in Z(x)} \langle \beta, \Phi(x,z) \rangle \,, \tag{1}$$

where β is a vector of model parameters (filters), z are latent values and Φ is a mapping from image space to feature space, i.e., the feature vector. In our model, β is a vectorized combination of F_t and F_s from all mixture components and will be discussed later. The inner product term in Eq. 1, $\langle \beta, \Phi(x,z) \rangle$, is called the score function. A binary label for x can be obtained by thresholding $f_\beta(x)$. The set $Z(x)$ defines the possible latent values for an example x. In our framework, z is the possible coordinates and scales of an image window in the feature pyramids. Unlike the model in [15], which contains one root and many parts, our method uses one window only. We also do not impose any spatial constraints on the windows and allow them to move freely to any place in the latent value space. Our patch selector thus works on any type of object.

The parameter vector β is learned from labelled examples $D = \{(x_i, y_i)\}_{i=1}^n$, $y_i \in \{-1, 1\}$, by minimizing the following objective function

$$L_D(\beta) = \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i)), \tag{2}$$

where $\max(0, 1 - y_i f_\beta(x_i))$ is the standard hinge loss and the parameter C controls the trade-off between penalizing the loss and maximizing the margins. Rather than optimizing $L_D(\beta)$ directly, L-SVM defines an auxiliary objective function $L_D(\beta, Z_p) = L_{D(Z_p)}(\beta)$, where $D(Z_p)$ is derived from D by restricting the latent values for positive examples which have latent values controlled by Z_p. Because $L_D(\beta) = \min_{Z_p} L_D(\beta, Z_p)$ the auxiliary objective function bounds the L-SVM objective. Please refer to [15] for more details. Subsequently, $L_D(\beta, Z_p)$ is minimized by using a 2-step iterative process:

- Relabel positive examples: Optimize $L_D(\beta, Z_p)$ over Z_p by selecting the highest scoring latent value for each positive example, $z_i = \operatorname{argmax}_{z \in Z(x_i)} \beta \cdot \Phi(x_i, z)$.
- Optimize β: Optimize $L_D(\beta, Z_p)$ over β by solving the convex optimization problem defined by $L_{D(Z_p)}(\beta)$.

In our work, we use 1-vs-all training for each textile category, i.e., the samples from one category form the positive examples and the rest are negative examples.

Score Function The score function is the inner product between the L-SVM parameter vector β and the feature vector Φ (see Eq. 1). The score function is used to score the latent values for positive examples (as in Step-1 of L-SVM optimization), to mine hard-negative examples and to perform patch selection. The patch selection process is discussed previously, and a hard-negative mining algorithm can be found in [15].

We compute the score by applying the filters to two feature pyramids $\mathbf{P}_s$ and $\mathbf{P}_t$, corresponding to the shape and texture of the textile. The texture feature pyramid is calculated at twice the resolution of the shape feature pyramid. The reason for computing texture features at higher resolution is to ensure that the local statistics are sufficient for discriminating different textiles. Whereas, shape feature at higher resolution will likely be less generic, i.e., it will not be sensitive to variations of a textile's macro deformable surfaces. Fig. 4 shows the shape, texture and combined score maps from several images.

As mentioned earlier, we treat each textile category as a mixture model $M = (m_1, ..., m_N)$, where $\{m_i\}_{i=1}^N$ are the components of the mixture and N is the number of components. Each component represents an instance of the textile. A textile instance hypothesis $\mathbf{h} = (i, z)$ for the mixture model specifies a component $i \in \{1, \cdots, N\}$ and a location $z = (u, v, l)$ for the filters of the component m_i, where l is the level in the feature pyramids and (u, v) is the spatial location. The score of hypothesis $\mathbf{h}$ is defined as:

$$Score(\mathbf{h}) = F_s^i \cdot \phi_s(\mathbf{P_s}, z) + F_t^i \cdot \phi_t(\mathbf{P_t}, z) + b_i, \tag{3}$$

where F_s^i and F_t^i are the shape and the texture filters (as vectors) of the model i, respectively. $\phi_s(\mathbf{P_s}, z)$ and $\phi_t(\mathbf{P_t}, z)$ represent the respective shape and texture feature vectors, computed at level l of two feature pyramids $\mathbf{P_s}$, $\mathbf{P_t}$ and at location (u, v). The dot operator is the vector dot product, which is analogous to convolving the filter and feature vectors.

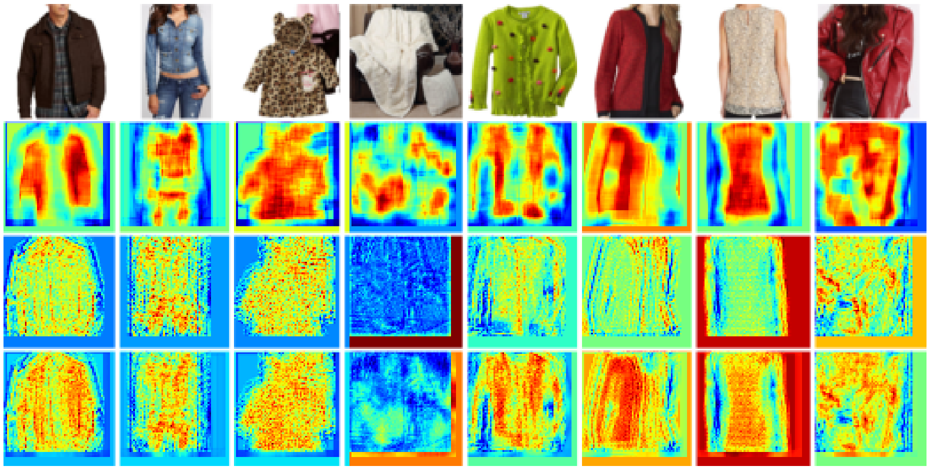

Fig. 4. Score maps from different categories. From top to bottom rows: original images, texture maps, shape maps, and combined score maps.

The score function in Eq. 3 can be written in inner product form, as in Eq. 1. β is formed by concatenating all filter parameters $\{F_s^i, F_t^i, b_i\}$ into a vector,

$$\beta = (F_s^1, F_t^1, b_1, ..., F_s^N, F_t^N, b_N). \tag{4}$$

Similarly, a feature vector for each example (an image patch) is formed by concatenating shape and texture features extracted at location z in the feature pyramids,

$$\Phi_i(\mathbf{P_s}, \mathbf{P_t}, z) = (0, ..., 0, \phi_s(\mathbf{P_s}, z), \phi_t(\mathbf{P_t}, z), 1, ..., 0), \tag{5}$$

where the zeros place the feature vectors in the position corresponding to the filters of the i-th mixture component, i.e., so that they are convolved with the correct filter entries in β.

After having β and Φ_i, we can measure the score of a hypothesis by simply taking a dot product $\langle \beta, \Phi_i(\mathbf{P_s}, \mathbf{P_t}, z)\rangle$.

Intuitively, the texture filter works as a texture detector that is sensitive to micro structures of a textile, whereas, the shape filter's role is to locate areas with similar macro shapes. Fig. 5 shows two cases of textile recognition. In the first case (Left), the two texture histograms are very similar while the shapes (HOG) are obviously different. In contrast in the second case (Right), the two HOG maps are almost the same while the texture histograms are very discriminative.

Initialization Method Initialization is a crucial part in training L-SVM. As noted by [15], β needs to be initialized carefully because the algorithm may select unreasonable latent values for the positive examples in the first iteration, causing bad models. In practice, we extract shape features from all the annotated

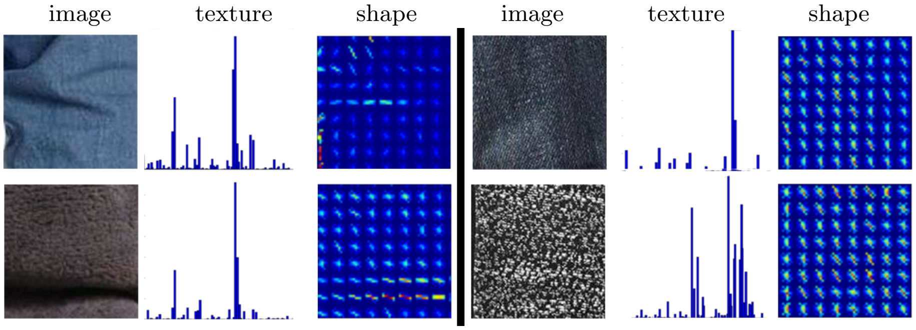

Fig. 5. Two scenarios of textile recognition. (Left) Texture filter fails but shape filter works. (Right) Shape filter fails and texture filter works.

regions in all training images. As introduced in Sec. 3, these annotations are patches identifying the interesting texture regions in each image. Recall that we want to build a mixture model $M = (m_1, ..., m_N)$ that represents N instances of the textile. We start with clustering all the image patches into N clusters by employing the K-means algorithm on the shape features. We do not consider texture as it did not show significant improvement in clustering quality. After the initialization, we have every region associated with a cluster label. The cluster label is used as the initial component assignment for each sample patch.

4.2 Categorization

Having the textile models in hand, we can either use the score assigned by L-SVM to directly classify the images, or use the highly-scored regions to extract the features and then learn another classifier. Experimental results show little difference between the two methods. We, however, choose the latter as this method often yields more stable results.

To proceed, we first perform a simple matching procedure. A mixture M_c is learned for each textile category, resulting in a set of mixtures $\{M_c\}_{c=1}^{C}$, where C is the number of categories. Given an image q we fit all mixtures $\{M_c\}_{c=1}^{C}$ on to the image by convolving the pair of filters from each component with it using Eq. 3. After this step, we obtain $\{\{p_{ic}\}_{i=1}^{N}\}_{c=1}^{C}$ candidate patches. We then calculate the score $\{s_c\}_{c=1}^{C}$ associated with each mixture,

$$s_c = \max_{i \in \{1,..,N\}} Score(p_{ic}),$$

where $Score(p_{ic})$ is computed using Eq. 3. In other words, the score for a given textile is the highest score among its mixture components. Feature vectors of the patches selected by the mixture with the largest s_c are then extracted and aggregated together to form a feature vector $\mathbf{f}$. Finally, textural statistics $\mathbf{S}$ is computed from $\mathbf{f}$ to form a descriptor for each query q:

$$\mathbf{d} = (\mathbf{S}_q, s_1, ..., s_C).$$

The concrete form of the textural statistics will be discussed in Section 5.

Next, a standard K-nearest neighbours classifier with Euclidean distance is used to assign labels to the testing samples. The number of neighbours is the same as the number of textile categories ($K = 8$). Fig. 2 (B) shows the whole categorization process.

5 Experiments

In this section we first give the implementation details and then discuss some experimental results.

5.1 Implementation Details and Experimental Set-up

As mentioned earlier, our model consists of two parts: shape and texture. Our implementation uses histograms of oriented gradients (HOGs) [16] as the shape feature and improved Fisher vector (IFV) [36] with MR-8 Filter Banks [5] as the texture feature. The choice of using IFV fits well in our framework. First, it is shown theoretically in [37] that the distances between IFVs can be accurately measured by simply taking their dot products. Second, as a bag-of-words based method, IFV can effectively summarize textural statistics over multiple patches.

For the GMM soft quantization, we use $K = 200$ Gaussian modes. MR-8 descriptors are extracted densely for every pixel. Prior to extracting the MR-8, the images are normalized to have zero means and unit standard deviation. The parameter C of L-SVM is selected by cross-validation. Although it is possible to have higher accuracy by fine-tuning C for different mixtures, we choose to use the same C for all mixtures.

To train the model, we split the dataset into training and testing datasets with the ratios (35/25), (25/35), (15/45) and performed L-SVM training on each training set. We used $N = 20$ components for each mixture. We treated all images in the training set of a category as positive examples and all training images from other categories as negative examples. Additionally, we also used a set of 50 landscape images as negative examples; this is to make the model more robust against backgrounds. We conducted our experiments on all 8 categories: *Fleece*, *Fur*, *Corduroy*, *Denim*, *Knit*, *Leather*, *Boucle* and *Lace*.

To evaluate the effectiveness we compared our patch selection method with other selection schemes: manual selection, selection based on the SIFT detector [8], selection based on objectness measure (OM) [9] and dense sampling. All training and testing were performed individually on patches selected by each method. All patch selection and texture categorization tasks were repeated on 10 random splits. For SIFT, we computed MR-8 at the location (u, v) and scale l of 20 key-points. For OM, we used the software package provided by the authors with the default settings except for the color contrast cue, which was excluded for fair comparison with other methods. We trained OM on the same set of annotations from our dataset. Furthermore, we also include the results of the current state-of-the-art texture classification framework (RSDS) [6]. While all other methods were tested using the same framework as ours (IFV-MR8 and KNN), RSDS used the feature and the classifier provided by the authors [6].

5.2 Results

Sample Patch Selections Fig. 6 shows some sample images and the patches selected by both human and our method. The bottom-right example shows the difference between the human and machine patches. While the human annotator tends to choose the patches evenly over the object, our method selects patches around some certain areas. It is interesting to see that the pillow which was not noticed at all by the annotator is selected by our method. In contrast our method successfully avoids the rest of the image which contains the curtain and the wall. A similar conclusion is applicable to other examples. In the top-right example (i.e., red jacket), the hair is successfully avoided whereas there is only one misidentified patch in the top-left example.

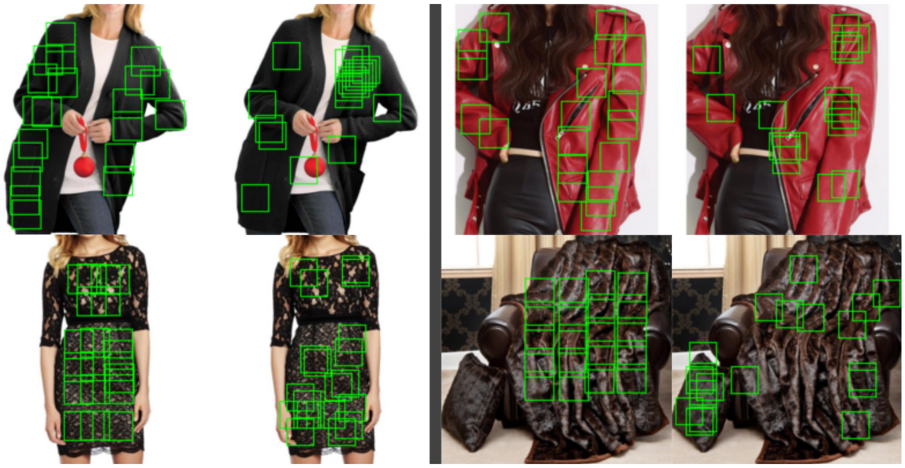

Fig. 6. Patch selection on novel images. In each example, the patches in the left image were human annotated and those in the right image were automatically selected by our method.

Categorization Results Table 1 shows the categorization performance of the compared methods. Despite the diversity of the dataset and the simplicity of the feature our method achieved very promising results. The average accuracy when the model was trained with 35 samples was 64.6%. In particular, the accuracies for the categories of Leather and Boucle were 77.8% and 84.0%, respectively. This is because Boucle and Leather often come with uniform patterns and decorations. The performance on the Fleece category was surprisingly good as we can see how diverse this category is in the dataset. It is worth noting that the differences between using 35, 25 and 15 training samples were not very large. This indicates that out method performs quite well even with a limited number of samples.

The categorization on human annotated patches was inferior to our method in almost all cases (Table 1), mainly because the L-SVM searches for the most discriminative patches. Since we treat the position of a patch as a latent value z, it is allowed to extensively search for better locations around the image. The search is constrained by two criterion: shape and texture. In cases like leather and chiffon that often exhibit little textural information at micro scales, the shape component computed at larger scales will play a major role. Whereas, textiles like fleece and fur have very vague macro shapes that do not contribute much to the discrimination. In these cases, the texture component will be "in charge". More importantly, by selecting patches with similar surfaces from different images, the texture comparison would be more accurate as the lighting condition and deformation are close.

The performance by SIFT and OM (Table 1) was consistently low for all the numbers of training samples. Since the SIFT-based detector and OM tend to select patches with strong edges, they mistakenly skip those "smooth" areas that may contain informative textural patterns. It is a surprise that both densely sampled IFV-MR8 and RSDS performed better than SIFT and OM, even though the features were computed over an entire image. Evidently, only our method selected patches better than dense sampling. When compared on the same feature, our method boosted the accuracy by 11% (the case of 35 training examples, Table 1) and was 5–8% better than the state-of-the-art method, RSDS.

Table 1. Categorization accuracies using patches selected by: Our method (Our), human (Man), SIFT detector, dense sampling (Dens), Objectness Measure (OM) and RSDS.

M	fleece	fur	corduroy	denim	knit1	leather	boucle	lace	average
Number of training examples = 35									
Our	50.2±6.8	59.6±11.2	48.9±7.5	**67.6**±5.5	**63.6**±7.2	**77.8**±4.7	**84.0**±4.2	65.3±8.2	**64.6**±2.3
Man	53.8±5.7	57.3±5.7	40.9±8.2	52.0±7.3	47.6±14.2	74.2±6.0	76.0±8.6	48.0±7.3	56.2±3.5
SIFT	32.0±6.0	46.7±12.2	24.4±12.0	41.3±8.4	28.4±6.9	64.0±7.8	64.9±8.0	55.1±9.8	44.6±3.3
Dens	36.0±5.0	46.2±11.2	28.4±6.7	58.7±8.4	46.7±8.6	67.1±3.7	77.3±6.3	68.9±10.3	53.7±2.4
OM	44.9±5.6	37.8±7.1	26.7±10.3	39.6±8.3	36.0±11.2	52.4±7.4	58.2±7.8	56.9±9.2	44.1±3.2
RSDS	**59.6**±9.3	**60.9**±10.9	**61.3**±10.4	53.8±7.5	62.7±7.2	45.3±13.9	64.9±7.4	**69.3**±9.4	59.7±2.4
Number of training examples = 25									
Our	49.8±9.3	53.3±6.3	40.3±5.5	**58.4**±7.5	**56.2**±5.0	**81.9**±5.9	**86.0**±3.1	**68.6**±4.5	**61.8**±2.4
Man	53.3±6.6	**54.0**±9.4	37.5±6.8	50.5±8.7	47.0±8.5	73.0±4.5	75.6±4.9	45.1±8.6	54.5±3.8
SIFT	23.5±8.1	46.0±8.6	28.9±7.1	38.1±6.0	28.3±5.6	63.5±5.7	64.4±6.5	49.2±10.4	42.7±3.3
Dens	36.5±4.4	44.1±7.1	27.6±8.6	58.7±8.9	46.7±7.9	64.8±4.0	76.8±6.7	62.5±9.1	52.2±2.8
OM	43.8±6.2	38.1±7.1	29.2±8.2	35.9±10.1	36.2±8.4	45.7±6.5	56.2±5.0	55.9±7.1	42.6±1.7
RSDS	**55.6**±8.0	46.0±10.7	**61.9**±13.2	42.5±14.2	47.6±10.6	49.2±9.6	59.4±6.5	67.6±4.3	53.7±2.8
Number of training examples = 15									
Our	32.6±7.3	**52.8**±7.1	37.5±6.9	55.6±5.1	**56.5**±7.1	**73.1**±1.6	**76.8**±4.1	**64.4**±6.7	**56.2**±1.8
Man	41.7±7.4	42.2±7.9	38.3±8.2	44.0±3.4	50.1±7.5	72.1±6.2	72.1±7.0	34.6±7.8	49.4±1.7
SIFT	24.0±7.5	35.3±8.3	27.4±8.2	31.9±7.6	28.9±9.9	62.2±8.2	60.5±5.4	35.6±9.2	38.2±2.5
Dens	32.1±6.5	38.5±7.6	27.9±7.8	**57.8**±12.7	46.7±4.4	62.2±6.2	75.6±3.8	49.9±8.6	48.8±1.7
OM	39.0±9.2	35.3±7.1	26.7±12.1	31.4±9.0	34.1±8.1	36.5±5.7	53.8±4.2	46.7±8.2	37.9±3.2
RSDS	**53.1**±14.5	41.7±7.2	**50.9**±10.3	37.0±11.2	51.1±12.0	52.1±10.4	42.7±14.5	58.0±12.5	48.3±1.4

Impact of Shape and Texture Components To support our decision on choosing the components for patch detector, we show in Table 2 the categorization results when using shape-only and texture-only features for patch selection.

Table 2. Comparison of textile categorization using shape-only (Shape), texture-only (Tex) and shape-texture (Both) features for patch selection.

Method	fleece	fur	corduroy	denim	knit1	leather	boucle	lace	average
Number of training examples = 35									
Both	**50.2**±6.8	**59.6**±11.2	**48.9**±7.5	**67.6**±5.5	**63.6**±7.2	**77.8**±4.7	**84.0**±4.2	**65.3**±8.2	**64.6**±2.3
Tex	28.9±7.5	50.7±7.8	38.7±4.2	48.4±6.7	33.3±7.8	59.6±7.4	72.9±5.9	61.3±7.8	49.2±2.2
Shape	33.3±7.1	28.9±7.5	24.9±7.7	44.9±9.9	41.3±8.4	44.9±7.7	66.7±4.2	44.0±8.0	41.1±3.0
Number of training examples = 25									
Both	**49.8**±9.3	**53.3**±6.3	**40.3**±5.5	**58.4**±7.5	**56.2**±5.0	**81.9**±5.9	**86.0**±3.1	**68.6**±4.5	**61.8**±2.4
Tex	33.7±5.5	45.7±9.3	34.6±9.8	43.8±5.4	31.7±7.8	59.7±10.6	68.6±6.7	54.9±8.3	46.6±1.8
Shape	38.4±4.3	47.0±12.8	40.3±6.4	45.1±10.0	47.9±9.0	58.4±8.4	71.4±3.3	59.7±6.9	51.0±2.0
Number of training examples = 15									
Both	**32.6**±7.3	**52.8**±7.1	**37.5**±6.9	**55.6**±5.1	**56.5**±7.1	**73.1**±1.6	**76.8**±4.1	**64.4**±6.7	**56.2**±1.8
Tex	23.0±5.5	38.5±6.9	30.1±6.4	35.6±10.2	26.2±6.7	54.1±9.5	69.1±6.9	53.3±8.0	41.2±2.4
Shape	26.9±5.7	37.8±6.5	29.4±7.5	53.6±6.3	49.9±5.7	50.9±10.9	61.7±4.3	48.4±7.3	44.8±1.3

Overall, the performance of the shape-only or texture-only method was clearly inferior to their combination. When using 35 training examples, patches selected with shape-texture features yielded 64.6% accuracy, while the accuracies for shape-only and texture-only were only 41.1% and 49.2%, respectively. In addition, the performance of patches selected with texture features was generally lower than that of shape feature. However, texture feature is more robust in the cases of Fur, Lace and Boucle.

6 Conclusions

We have presented a novel model for textile recognition, consisting of patch selection and textile categorization. The patch selection process is carried out with Latent-SVM, an efficient learning method recently introduced into the world of object recognition, and uses both micro level texture and macro deformable shape to select representative patches. Our model is capable of detecting the most representative textural regions in an image, leading to significantly better textile categorization performance. An important property of our method is the ability of the machine to learn from human annotations patches and then refine them to produce more discriminative patches. We believe that by replacing the basic texture feature with more advanced ones our model could achieve even better performance.

References

1. 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. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2012) 3330–3337
2. Chen, H., Gallagher, A., Girod, B.: Describing clothing by semantic attributes. In: Computer Vision (ECCV), Eropean Conference on. Springer (2012) 609–623
3. Bossard, L., Dantone, M., Leistner, C., Wengert, C., Quack, T., Van Gool, L.: Apparel classification with style. In: Computer Vision (ACCV), Asian Conference on. Springer (2013) 321–335

4. Wang, X., Zhang, T.: Clothes search in consumer photos via color matching and attribute learning. In: Proceedings of the 19th ACM International Conference on Multimedia. (2011) 1353–1356
5. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. International Journal of Computer Vision (IJCV) **62** (2005) 61–81
6. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2013) 1233–1240
7. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2014)
8. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (IJCV) (2004) 91–110
9. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. Pattern Analysis and Machine Intelligence, IEEE Transactions on (2012)
10. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer Vision (ECCV), European Conference on. (2006)
11. Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2007) 1–8
12. Harel, J., Koch, C., Perona, P.: Graph-based visual saliency. In: Advances in Neural Information Processing Systems (NIPS). (2006)
13. Gao, D., Vasconcelos, N.: Bottom-up saliency is a discriminant process. In: Computer Vision (ICCV), IEEE International Conference on. (2007)
14. Bruce, N., Tsotsos, J.: Saliency based on information maximization. In: Advances in Neural Information Processing Systems (NIPS). (2005)
15. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010) 1627–1645
16. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2005)
17. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval. (2007) 401–408
18. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision (IJCV) **43** (2001) 29–44
19. Schmid, C.: Constructing models for content-based image retrieval. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. Volume 2. (2001) II–39
20. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31** (2009) 2032–2047
21. Guo, Z., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. Image Processing, IEEE Transactions on (2010)
22. Liao, S., Law, M.W., Chung, A.C.: Dominant local binary patterns for texture classification. Image Processing, IEEE Transactions on (2009)
23. Sharan, L., Liu, C., Rosenholtz, R., Adelson, E.H.: Recognizing materials using perceptually inspired features. International Journal of Computer Vision (IJCV) **103** (2013) 348–371

24. Caputo, B., Hayman, E., Mallikarjuna, P.: Class-specific material categorisation. In: Computer Vision (ICCV), IEEE International Conference on. Volume 2. (2005) 1597–1604
25. Caputo, B., Hayman, E., Fritz, M., Eklundh, J.O.: Classifying materials in the real world. Image and Vision Computing **28** (2010) 150–163
26. Liu, C., Sharan, L., Adelson, E.H., Rosenholtz, R.: Exploring features in a bayesian framework for material recognition. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2010) 239–246
27. Yamaguchi, K., Kiapour, M.H., Ortiz, L.E., Berg, T.L.: Parsing clothing in fashion photographs. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2012) 3570–3577
28. Yamaguchi, K., Kiapour, M.H., Berg, T.L.: Paper doll parsing: Retrieving similar styles to parse clothing items. In: Computer Vision (ICCV), IEEE International Conference on. (2012) 3519–3526
29. Dong, J., Chen, Q., Xia, W., Huang, Z., Yan, S.: A deformable mixture parsing model with parselets. In: Computer Vision (ICCV), IEEE International Conference on. (2013) 3408–3415
30. Kalantidis, Y., Kennedy, L., Li, L.J.: Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval. (2013) 105–112
31. Hidayati, S.C., Cheng, W.H., Hua, K.L.: Clothing genre classification by exploiting the style elements. In: Proceedings of the 20th ACM International Conference on Multimedia, ACM (2012) 1137–1140
32. Liu, S., Feng, J., Song, Z., Zhang, T., Lu, H., Xu, C., Yan, S.: Hi, magic closet, tell me what to wear! In: Proceedings of the 20th ACM International Conference on Multimedia. (2012) 619–628
33. Blaschko, M.B., Hofmann, T.: Conformal multi-instance kernels. In: Learning to Compare Examples (NIPS), Workshop on. (2006) 1–6
34. Gehler, P.V., Chapelle, O.: Deterministic annealing for multiple-instance learning. In: Artificial Intelligence and Statistics, International Conference on. (2007) 123–130
35. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems (NIPS). (2002) 561–568
36. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Computer Vision (ECCV), European Conference on, Springer (2010) 143–156
37. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on. (2010) 3384–3391