Cloth manipulation based on category classification and landmark detection

Oscar Gustavsson1,*, Thomas Ziegler2,*, Michael C Welle1, Judith Bütepage1, Anastasiia Varava1 and Danica Kragic1

Abstract
Cloth manipulation remains a challenging problem for the robotic community. Recently, there has been an increased interest in applying deep learning techniques to problems in the fashion industry. As a result, large annotated data sets for cloth category classification and landmark detection were created. In this work, we leverage these advances in deep learning to perform cloth manipulation. We propose a full cloth manipulation framework that, performs category classification and landmark detection based on an image of a garment, followed by a manipulation strategy. The process is performed iteratively to achieve a stretching task where the goal is to bring a crumbled cloth into a stretched out position. We extensively evaluate our learning pipeline and show a detailed evaluation of our framework on different types of garments in a total of 140 recorded and available experiments. Finally, we demonstrate the benefits of training a network on augmented fashion data over using a small robotic-specific data set.

Keywords
Cloth/garment manipulation, classification, vision for robotics, data augmentation

Introduction
Grasping and manipulation of rigid objects have been studied extensively.1–4 In contrast, deformable object manipulation received relatively little attention due to the challenges related to the complexity in modeling, tracking and control.5 Manipulating clothing items is particularly difficult, as classical control approaches that require modeling the objects’ dynamics are only applicable in restrictive settings.6 Learning-based and data-driven approaches that do not rely on specific models are a viable approach for tasks that involve highly deformable objects.7

Clothing items are one example of highly deformable objects and have been used in applications such as grasp point detection,8 folding,9 sorting,10–12 unfolding,13 dressing,12 and classification.13–17 There is also an increased interest in using deep learning techniques for online shopping and e-commerce in the fashion industry addressing problems such as clothing category classification, fashion landmark detection, image retrieval and similarity-based recommendations. Following the creation of large-scale fashion data sets,18–20 significant progress has been made in fashion image analysis. Deep learning-based models have achieved significant performance gain in clothing
category classification,\textsuperscript{19,21–23} item recommendation\textsuperscript{24,25} and retrieval.\textsuperscript{19,26}

We present an extension of our earlier work on clothing category classification and fashion landmark detection.\textsuperscript{27} While the fashion industry often considers structured data, such as a human wearing clothes facing the camera, the data in robotic applications is less structured and can contain images of upside-down, crumpled clothing items. We built upon the progress made in fashion image analysis and proposed a network architecture and training procedure on a large-scale fashion data set DeepFashion.\textsuperscript{19} Our model was capable to generalize well to the noisy, poorly controlled conditions encountered in robotic clothing manipulation. We introduced \textit{elastic warping}, a novel image augmentation method that uses random displacement fields to create authentic looking clothing configurations to resemble the more challenging clothing configurations encountered in robotic manipulation. Furthermore, we incorporated rotation invariance and attention mechanisms in order to handle difficult configurations faced in robotic manipulation.

In the work presented here, we extend our earlier work\textsuperscript{27} and present a full robotic manipulation framework able to classify different clothing items in a robotic manipulation context, and that makes use of the detected landmarks to manipulate the garments. The contributions are: (i) A robotic cloth manipulation framework based on category classification and landmark detection. (ii) An extensive experimental evaluation on a real robot (140 recorded experiments (https://cloth-manipulation-landmarks.github.io/cml-web/)). (iii) An extended analysis on the effect of the elastic warping method parameters. (iv) A comprehensive description of all parts of the framework, including a more detailed description of the underlying network architecture.

\textbf{Related work}

The release of large-scale fashion data sets have sparked increased interest in the computer vision community on the analysis on fashion images addressing clothing recognition,\textsuperscript{19,21–23} recommendation,\textsuperscript{25} retrieval\textsuperscript{26} and fashion landmark localization.\textsuperscript{19,23,28,29} Liu et al.\textsuperscript{19} propose a multi-branch network for simultaneous classification, retrieval and landmark localization and in Liu et al.,\textsuperscript{20} they demonstrate refinement of landmark localization. Works of Wang et al.\textsuperscript{29} and Liu and Lu\textsuperscript{30} are examples of deep fashion grammar network for combined clothing category classification and landmark localization.

Image data used by the robotics community differs significantly from that commonly used in retail applications. The items are either spread out or crumpled on a flat surface,\textsuperscript{13,15} or they are in a hanging state when grasped by a robotic gripper.\textsuperscript{9,31–33} The robotics community has mostly focused on task specific, handcrafted feature extraction, such as edges and corners\textsuperscript{34} and wrinkles.\textsuperscript{35–37} Due to the 3D nature of the manipulation task, the use of physics and volumental simulators is more common in robotics.\textsuperscript{16,38} Recent methods\textsuperscript{9,32,33} use convolutional neural networks (CNN) instead of handcrafted features for classification.

Our previous work, focused on category classification and landmark localization, which is now extended here in a complete robotic cloth manipulation pipeline. The network has a similar architecture to,\textsuperscript{29,30} but has been extended to more challenging clothing configurations present in robotic applications. Our method does not require generation of a specific labeled data set with predefined grasp points. Instead, it leverages the existing labeled landmarks present in the recent fashion data sets and generalizes to images taken in a robotic lab.

\textbf{Method}

We first formulate the problem of category classification and landmark prediction to be used in a cloth manipulation pipeline. We introduce two image augmentation methods to perturb clothing configuration in such a way that they are more representative of clothing configurations encountered during a robotic cloth manipulation task. We then give a detailed description of the proposed network and describe how the gained knowledge is used in the downstream cloth manipulation task.

\textbf{Problem formulation}

Our goal is to simultaneously predict the landmark locations $\mathbf{L}$ and category classification $\mathbf{C}$, on a given image $\mathbf{I}$. The landmarks are defined as $\mathbf{L} = \{(x_k, y_k)\}_{k=1}^{n_L}$, where $(x_k, y_k)$ is the $k$th pixel coordinate position in $\mathbf{I}$ and $n_L$ the total number of landmarks per image. The category classification $\mathbf{C} \in [0, 1]^{n_c}$ satisfies $\sum_{j=1}^{n_c} C_j = 1$, where $n_c$ is the number of categories depending on the used data set. Using $\mathbf{L}$ and $\mathbf{C}$, the goal is to further manipulate the garment in a way that minimizes the landmark position error: $\mathbf{I}_{err} = \mathbf{L} - \mathbf{L}_{\text{template}}$, where $\mathbf{L}_{\text{template}}$ is the desired landmark position.

\textbf{Image augmentation}

The two proposed image augmentations are image rotation and \textit{elastic warping}. To augment an image together with its landmarks we define the image before transformation as input image $\mathbf{I}$ and the image after the transformation as transformed image $\tilde{\mathbf{I}}$. In both cases, $w, h$ stand for the width and height of the image, respectively.

The transformation can be represented as a mapping of the pixels, $\psi(x, y) \in [1, w] \times [1, h]$:

$$\tilde{\mathbf{I}}(\tilde{x}, \tilde{y}) = \mathbf{I}(x(\tilde{x}, \tilde{y}), y(\tilde{x}, \tilde{y})), \quad (1)$$

where $x, y$ are the pixel location in the input image $\mathbf{I}$ and $\tilde{x}, \tilde{y}$ the pixel location in the transformed image $\tilde{\mathbf{I}}$. The
clothing landmark locations $\mathbf{L} = \{(x_k, y_k)\}_{k=1}^{n_L}$ are a set of $n_L$ specific pixel coordinates in the input image $\mathbf{I}$.

When $x(\hat{x}, \hat{y})$ and/or $y(\hat{x}, \hat{y})$ are noninteger, interpolation is needed. We apply the commonly used bilinear interpolation\(^\text{39}\) in such a case.

**Rotation.** Rotating images is often used to increase the performance in classification and/or detection tasks.\(^\text{40}\) When clothing items lie on a flat surface, they can be in any orientation. We hence randomly sample an angle $\theta$ in the range $[0, 2\pi]$ for each rotation.

**Elastic warping.** Our proposed elastic warping method is similar to the elastic deformation proposed in Simard et al.\(^\text{39}\) but is further extended to produce realistic, task-specific images and to allow for landmark detection.

The deformation is created by generating two random displacement fields $\Delta x(\hat{x}, \hat{y})$ and $\Delta y(\hat{x}, \hat{y})$. The whole augmentation is performed in four steps:

First: Sample $n_S$ pixel positions uniformly in the transformed image: $\mathcal{S} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{n_S}$.

Second: For each pixel location in $\forall(\hat{x}_i, \hat{y}_i) \in \mathcal{S}$ sample a random displacement from a uniform distribution $\mathcal{U}(-\alpha, \alpha)$

$$\Delta x(\hat{x}_i, \hat{y}_i) \sim \mathcal{U}(-\alpha, \alpha), \Delta y(\hat{x}_i, \hat{y}_i) \sim \mathcal{U}(-\alpha, \alpha).$$  

(2)

All other entries in the displacement fields are set to 0.

Third: Convolve the two displacement fields with a Gaussian filter $\mathbf{G}$, $\forall(\hat{x}, \hat{y}) \in [1, w] \times [1, h]$

$$\Delta x(\hat{x}, \hat{y}) = \Delta x(\hat{x}, \hat{y}) \ast \mathbf{G}(\hat{x}, \hat{y})$$  

(3)

$$\Delta y(\hat{x}, \hat{y}) = \Delta y(\hat{x}, \hat{y}) \ast \mathbf{G}(\hat{x}, \hat{y})$$  

(4)

where $\ast$ denotes the convolution operator and $\mathbf{G}(\hat{x}, \hat{y})$ is a Gaussian filter with variance parameter $\sigma$.

Fourth: Use the smoothed displacement field to create the transformed image, $\forall(\hat{x}, \hat{y}) \in [1, w] \times [1, h]$

$$\tilde{\mathbf{I}}(\hat{x}, \hat{y}) = \mathbf{I}(\hat{x} + \Delta x(\hat{x}, \hat{y}), \hat{y} + \Delta y(\hat{x}, \hat{y}))$$  

(5)

The strength of the distortion can be adjusted by the number of initially displaced pixels $n_S$, the scaling of the uniform distribution $\alpha$ and the smoothness of the Gaussian filter $\sigma$. We use $n_S = 3$, $\alpha = 500$ and $\sigma = 40$ in our experiments. Figure 1 shows some examples when using this configuration. While the presented elastic warping method can for example not emulate folded configuration, the possibility to adjust the distortion with three hyperparameters gives a wide range of different data-augmentation possibilities. Note that a too high $n_S$ can lead to undesirable image artifacts.

**Landmark warping.** The displacement fields indicate where a pixel in the transformed image was located in the input image. Due to the random nature of these fields no
inverse exists. Therefore it is not trivial to know if/where the pixels of the input image are found in the transformed image. As our goal is to preserve the correct position of the landmarks defined in the input we describe an efficient method for retrieving the landmark position in the transformed image.

For every landmark position \( L_k = (x_k, y_k) \), we find \( n \) possible pixels in the transformed image \( \tilde{I} \) which originated at or near the position of the landmark in the input image \( I \):

\[
X = \arg\min_n - n \cdot \text{sort}|\tilde{x} + \Delta x(\tilde{x}, \tilde{y}) - x_k|,
\]

\[
Y = \arg\min_n - n \cdot \text{sort}|\tilde{y} + \Delta y(\tilde{x}, \tilde{y}) - y_k|,
\]

where \( \arg\min - n \) returns the \( n \) smallest values from a sorted set. Note that both \( X \) and \( Y \) contain coordinate pairs \((\tilde{x}, \tilde{y})\). The value of \( n \) depends on the image size and the chosen parameters \( n_S, \alpha, \) and \( \sigma \) in the elastic warping. We use \( n = 200 \) in our experiments. To get the transformed landmark \( \tilde{L}_k \) we need to find the coordinate pair \((\tilde{x}^*, \tilde{y}^*)\) that is either present in both \( X \) and \( Y \) or the coordinate pair in \( X \) with closest neighbor in \( Y \).

We use the fact that the pixel coordinates are unique integer values and create a hash table for all coordinate pairs in one set. In the following, one can search for each pair in the other set if a key exist in the hash table, which reduces time complexity for existing coordinated pairs to \( O(n) \).

If the hash table does not return a valid value, no exact match exists in \( X \) and \( Y \). In this case, one can create a kd-tree (\( O(n \log n) \) for all coordinate pairs in \( Y \)) and use kd-tree search\(^{41}\) to find the nearest neighbor for the coordinate pairs in \( X \).

**Network architecture**

The main network architecture is loosely based on the VGG-16\(^{42}\) network structure similar to the networks proposed in Wang et al.\(^{29}\) and Liu and Lu.\(^{30}\) The structure can be seen in Figure 2(a). Compared to the base VGG-16 network, several structural changes are included: rotation invariance layers, a landmark localization branch and attention branches for classification.

**Rotation invariance.** As mentioned before, variation in orientation occurs more often in a robotic cloth manipulation task. In order to account for this, we replace the 2D convolution in the \( \text{conv1} \) to \( \text{conv4} \) layers with Averaged Oriented Response Convolutions (A-ORConvs). They produce enriched feature maps with the orientation information explicitly encoded.\(^{43}\)

A-ORConvs are an improvement of the Oriented Response Convolutions (ORConvs) initially proposed in Zhou et al.\(^{44}\) These convolution blocks use Averaged Active Rotating Filters (A-ARFs) and Active Rotating Filters (ARFs), respectively. Both are a 5D tensors of size
\( n_\Omega \times n_1 \times w_f \times h_f \times N \), where \( n_\Omega \) is the number of output channels, \( n_1 \) the number of input channels, \( w_f \) and \( h_f \) are the width and height of the filter and \( N \) is the number of filter orientations. This means that in ARFs for each materialized filter, \( N - 1 \) immaterialized rotated copies of the same filter are present. Therefore, during forward propagation one ARF produces a feature map of \( N \) channels with orientation information encoded. Depending on the orientation of the input image a different copy of the filter has the highest response. A-ORConvs improve over ORConvs by reducing the risk of gradient explosion during training by updating the feature map with the mean value of the gradients from all its rotated copies instead of the sum of all gradients.

In our network (Figure 2(b)), we use the A-ORConvs with four orientation channels (i.e. \( N = 4 \)). We use the same filter size and the same number of total channels when replacing the standard 2D convolution in the \( conv1 \) to \( conv4 \) layers. This means that the effective number of parameters of the A-ORConvs is only a quarter of the normal convolution blocks. In order to create rotation invariant features, a Squeeze-ORAlign (S-ORAlign) layer\(^43\) is used to find the main response channel. The S-ORAlign is inspired by the Squeeze-and-Excitation (SE) block,\(^45\) first a \emph{squeeze} operation is performed by global average pooling. Then the main orientation channel is found via a maximum function and finally all channels are spun such that the main response channel is in the first position.

**Landmark localization branch.** The landmark localization branch is the same as proposed in Liu and Lu.\(^30\) The branch structure is depicted in Figure 2(c). It uses transposed convolutions\(^46\) to produce heatmaps for all landmarks. The transposed convolutions allow for an upsampling of the S-ORAlign features \( \mathbf{F} \) of dimension \( w_f \times h_f \times n_o \), where \( w_f \) and \( h_f \) are width and height of the feature map and \( n_o \) is the number of output channels, back to the original input image size. Given the features \( \mathbf{F} \) a \( 1 \times 1 \) convolution is applied to reduce the number of channels in the feature map to \( \mathbf{F}^{(1)} \). Then three blocks of two \( 3 \times 3 \) convolutions followed by a \( 4 \times 4 \) transposed convolution are utilized. The padding and stride of the transposed convolution are 1 and 2, respectively. Hence, such a block upsamples the feature maps by a factor of two, at the same time the number of channels is reduced by a factor of two. Finally, a \( 1 \times 1 \) convolution with a sigmoid activation is used to convert the \( \mathbf{F}^{(4)} \) feature map into the predicted heatmaps \( \mathbf{M} \) of dimension \( w_f \times h_f \times n_L \), where \( n_L \) is the number of landmarks (which corresponds to the maximum number of landmarks in any category considered).

The landmark localization branch can be trained separately from the classification. Let \( \mathbf{M}_k \in [0, 1]^{w_f \times h_f} \) and \( \mathbf{M}_k \in [0, 1]^{w_f \times h_f} \) denote the groundtruth heatmap and the predicted heatmap for the \( k \)th landmark, respectively. The landmark localization branch is trained using pixel-wise mean square differences

\[
\mathcal{L}_{LM} = \sum_{i=1}^{n_B} \sum_{k=1}^{n_L} \sum_{x=1}^{w_f} \sum_{y=1}^{h_f} \| \mathbf{M}_k(x, y) - \mathbf{M}'_k(x, y) \|^2_2, \tag{8}
\]

where \( n_B \) is the total number of training samples. The groundtruth heatmap \( \mathbf{M}_k \) is generated by adding a 2D Gaussian filter at the corresponding location \( \mathbf{L}^i_k \). Given a sample \( i \) the predicted coordinates for the \( k \)th landmark \( \mathbf{L}^i_k \) corresponds to the maximal value in the predicted heatmap

\[
\hat{\mathbf{L}}^i_k \in \arg\max_{(x, y) \in [1, w_f] \times [1, h_f]} \mathbf{M}'_k(x, y). \tag{9}
\]

If there is more than one maximum per landmark one of them is chosen at random.

**Attention branch.** The attention branch can be seen as a union of spatial attention\(^47\) and channel attention.\(^45\) The attention branch learns a saliency weight map \( \mathbf{A} \) of the same size as the S-ORAlign features \( \mathbf{F} \). Inspired by the proposed attention modules in Wang et al.,\(^29\) the spatial attention itself contains two types of attention, a landmark attention \( \mathbf{A}^{L}_{\text{spatial}} \) and a category attention \( \mathbf{A}^{C}_{\text{spatial}} \). Thus, the attention branch is designed as a three branch unit; two branches for the spatial attention \( \mathbf{A}^{L}_{\text{spatial}} \) and \( \mathbf{A}^{C}_{\text{spatial}} \) (Figure 2(d)) and one for the channel attention \( \mathbf{A}^{\text{channel}} \) (Figure 2(f)). These are combined in a factorized manner as \( \mathbf{A} = (\mathbf{A}^{L}_{\text{spatial}} + \mathbf{A}^{C}_{\text{spatial}}) \times \mathbf{A}^{\text{channel}} \).

**Spatial attention – Landmark.** Clothing landmarks represent functional regions of clothing and provide useful information about an item. The predicted heatmaps \( \{\mathbf{M}_k\}_{k=1}^{n_L} \) are used to guide attention to the functional clothing regions. The weight map is created by downsampling the predicted heatmaps by a factor \( d \), followed by a max-pooling operation

\[
\mathbf{M}' = \{ \text{downsample}_d \mathbf{M}_k \}_{k=1}^{n_L}, \tag{10}
\]

\[
\mathbf{A}^{L}_{\text{spatial}}(x, y) = \max_k \mathbf{M}'_k(x, y) \tag{11}
\]

This attention is learned in a supervised manner since it is directly derived from the predicted heatmaps.

**Spatial attention – Category.** Since the landmark attention only covers corner points of a clothing item, an additional spatial attention is used that focuses more on the clothing center. The category attention (Figure 2(e)) is modeled using an U-Net structure.\(^48\)

Given the S-ORAlign features \( \mathbf{F} \) a \( 1 \times 1 \) convolution is applied to convert the features into \( \mathbf{F}^{(1)} \). The U-Net consists of a contracting path that consists of two \( 4 \times 4 \) convolutions with stride 2, which squeeze the features down to \( \mathbf{F}^{(3)} \).
The number of feature channels doubles at every contracting step. Then a $1 \times 1$ convolution and $4 \times 4$ transposed convolution are applied generating the features $F_d^{(4)}$. Followed by the U-Net expanding path, which consists of two $4 \times 4$ transposed convolution. The input of the transposed convolution is a concatenation of the output from the previous transposed convolution and the corresponding feature map from the contraction path. The number of feature channels halves at every expanding step. At the end a $1 \times 1$ convolution is used to convert the channels to the same number as in the S-ORAlign features. The downpooling to a low resolution of $7 \times 7$ gives the spatial attention a large receptive field in the feature map of $F$. Upsampling is then used to have a weight map of the same size as $F$. The model learns the important regions of an image by itself. In contrast, our landmark attention receives the groundtruth heatmaps $M$, which resemble the landmark attention, during training.

**Channel attention.** The channel attention (Figure 2(f)) is implemented via a Squeeze-and-Excitation block. A squeeze operation creates $S$, an embedding of the global distribution of the channel-wise feature responses in $F$. This channel descriptor is created using average pooling

$$S(c) = \frac{1}{w_f \times h_f} \sum_{u=1}^{w_f} \sum_{v=1}^{h_f} F(u,v,c) \quad \forall c \in \{1, \ldots, n_O\}$$

where $F(\cdot, \cdot, c)$ is the feature map of the $c$th channel band $n_O$ the number of output channels. Then an excitation operation is performed on the channel wise aggregated feature map to create the channel attention. Following the proposal in Hu et al., a bottleneck is created using two fully connected layers, with a reduction rate $r$.

**Factorization.** The factorization (Figure 2(d)) is performed by multiplying the channel-wise feature responses in the spatial attention with the corresponding channel weights

$$\hat{A}(x, y, c) = \left( A_{\text{spatial}}^L(x, y, c) + A_{\text{spatial}}^C(x, y, c) \right) A_{\text{channel}}(c)$$

$$\forall (x, y) \in \{1, \ldots, w_f\} \times \{1, \ldots, h_f\} \quad \forall c \in \{1, \ldots, n_O\}.$$  

(13)

To refine the attention, an additional $1 \times 1$ convolution layer is added afterwards. This is motivated by the fact that the spatial and channel attention are not mutually exclusive but with co-occurring complementary relationship. Afterwards, a tan $h$ function is used to shrink the attention values into a range of $A \in [-1, 1]^{w_f \times h_f \times n_O}$.

**Output architecture.** Given $A$, we weight the S-ORAlign features $F_U = (1 + A) \circ F$, where $\circ$ denotes the Hadamard product and $1$ is a tensor. Hence, features where $A(\cdot, \cdot, \cdot) \in [-1, 0)$ are reduced and features where $A(\cdot, \cdot, \cdot) \in (0, 1]$ are increased. Our attention incorporates semantic information and global information into the network helping to focus on important regions in the images. The features $U$ are then fed in to the conv5-1 layer. The rest of the network follows the VGG-16 structure.

### Manipulation framework

Our manipulation framework, shown in Figure 3, consists of the deep neural network described in the previous section. It takes an image of the current scene containing a garment, and outputs the estimated landmarks $\hat{L}$ as well as the predicted class $\hat{C}$. This is taken in by the manipulation
algorithm described in the following section. The manipulation strategy is then executed and the new state of the garment is passed to the network again. This process is repeated until the desired template configuration is achieved or the process terminates.

**Manipulation strategy**

The implemented algorithm consists of two parts: an analysis step and a manipulation step. The analysis step detects the landmarks and clothing category. Based on the certainty of the landmarks a mode of operation for the manipulation step is selected, and based on the category a template is selected. The manipulation step has two modes of operation: landmark placement, where a landmark is picked and placed at its position in the template, or stochastic stretching, where a random point on the edge of the clothing is picked and placed a distance outward. The intuitive reason for the two modes for the manipulation strategy is that if the method is not confident to have identified the right category and landmarks, the clothing item needs to be further spread out to make an identification of the category/landmarks feasible. After each execution of the manipulation step, the analysis step is repeated followed by another manipulation step until the clothing has reached the final state described by the template.

A template $\mathcal{T}$ consists of the indices of the relevant landmarks, their position in world coordinate frame, and their corresponding weight. We use the following notation: $i \in \text{ind} \mathcal{T}$ to indicate that $i$ is a relevant landmark index in $\mathcal{T}$, $p_i^\mathcal{T}$ to indicate the position of the landmark with index $i$ and $w_i^\mathcal{T}$ to indicate the weight of landmark $i \in \text{ind} \mathcal{T}$. A landmark with index $i$ matches template $\mathcal{T}$ with tolerance $\varepsilon$, written $i \in \mathcal{T}$, if $i \in \text{ind} \mathcal{T}$ and $\| p_i^\mathcal{T} - p_i \| \leq \varepsilon$ where $p_i$ (obtained from the landmark localization branch) is the current location of the landmark with index $i$.

**Analysis step**

An image is taken and transformed to match an image taken from a virtual camera located right above the clothing to remove perspective distortion and rotation, improving the accuracy of the proposed network. The virtual camera is placed such that the bottom of the clothing in the template is parallel to the bottom of the image. When initializing the algorithm, the homography $\mathcal{H}$ between the two camera frames is computed, and subsequently used in the analysis step to transform every point $(x_i, y_i)$ in the captured image to a point $(x'_i, y'_i)$ in the transformed image as

\[
k \begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} = \mathcal{H} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}
\]

where $k$ is a scalar.

The contour of the garment is found using the OpenCV implementation of Suzuki et al., and a bounding box containing the region of interest is determined. The region of interest is passed through the network yielding the class $\mathcal{C}$, the location $(x_{i,lm}, y_{i,lm})$ of each landmark in the transformed image, and a distribution $p_i(j, k)$ for each landmark. All landmarks that lie outside of the contour are brought to the nearest contour vertex. If the certainty reported by the network for the category is above a certain threshold and there exists a template for class $\mathcal{C}$, the template $\mathcal{T}$ for class $\mathcal{C}$ is selected. If no template has been selected yet, and the certainty is not high enough or there is no template for class $\mathcal{C}$, the manipulation step is configured to do stochastic stretching and the analysis step terminates. The covariance matrix

\[
\Sigma_i = \sum_j \sum_k p_i(j, k) \begin{bmatrix} j - x_{i,lm} \\ k - y_{i,lm} \end{bmatrix} \begin{bmatrix} j - x_{i,lm} & k - y_{i,lm} \end{bmatrix}
\]

is computed for each landmark, the position is transformed to the original image using the inverse homography

\[
k \begin{bmatrix} x_{i,lm} \\ y_{i,lm} \\ 1 \end{bmatrix} = \mathcal{H}^{-1} \begin{bmatrix} x'_{i,lm} \\ y'_{i,lm} \\ 1 \end{bmatrix}
\]

where $k$ is a scalar. The landmarks are finally transformed to a three dimensional point in the world coordinate frame by assuming that all landmarks lie in the plane coinciding with the table. They are then compared with the template to form a set $\mathcal{L}_{err} = \{ i \mid i \notin \mathcal{T} \}$ of all landmarks that do not match their location in the template. If $|\mathcal{L}_{err}| = 0$ the algorithm terminates.

The final part of the analysis step determines the certainty of the landmarks to select between the two modes of operation. As a measure of uncertainty $U_i$ for landmark $i$ the weighted maximum eigenvalue of the covariance matrix $U_i = w_i^\mathcal{T} \lambda_{max}(\Sigma_i)$ is used. If $\min_{i \in \mathcal{L}_{err}} U_i$ is below a certain threshold, landmark argmin$_{i \in \mathcal{L}_{err}} U_i$ is used for landmark placement in the manipulation step, otherwise the manipulation step is configured to do stochastic stretching.

**Manipulation step**

Dependent on the result of the analysis step, the manipulation step does either landmark placement or stochastic stretching.

Landmark placement: The landmark $i$ with lowest uncertainty as selected by the analysis step is picked and placed at its location $p_i^\mathcal{T}$ in the template.

Stochastic stretching: If the analysis step could not determine any certain landmarks, the manipulation step does stochastic stretching. A contour around the clothing

\[
\Sigma_i = \sum_j \sum_k p_i(j, k) \begin{bmatrix} j - x_{i,lm} \\ k - y_{i,lm} \end{bmatrix} \begin{bmatrix} j - x_{i,lm} & k - y_{i,lm} \end{bmatrix}
\]

is computed for each landmark, the position is transformed to the original image using the inverse homography

\[
k \begin{bmatrix} x_{i,lm} \\ y_{i,lm} \\ 1 \end{bmatrix} = \mathcal{H}^{-1} \begin{bmatrix} x'_{i,lm} \\ y'_{i,lm} \\ 1 \end{bmatrix}
\]

where $k$ is a scalar. The landmarks are finally transformed to a three dimensional point in the world coordinate frame by assuming that all landmarks lie in the plane coinciding with the table. They are then compared with the template to form a set $\mathcal{L}_{err} = \{ i \mid i \notin \mathcal{T} \}$ of all landmarks that do not match their location in the template. If $|\mathcal{L}_{err}| = 0$ the algorithm terminates.

The final part of the analysis step determines the certainty of the landmarks to select between the two modes of operation. As a measure of uncertainty $U_i$ for landmark $i$ the weighted maximum eigenvalue of the covariance matrix $U_i = w_i^\mathcal{T} \lambda_{max}(\Sigma_i)$ is used. If $\min_{i \in \mathcal{L}_{err}} U_i$ is below a certain threshold, landmark argmin$_{i \in \mathcal{L}_{err}} U_i$ is used for landmark placement in the manipulation step, otherwise the manipulation step is configured to do stochastic stretching.

**Manipulation step**

Dependent on the result of the analysis step, the manipulation step does either landmark placement or stochastic stretching.

Landmark placement: The landmark $i$ with lowest uncertainty as selected by the analysis step is picked and placed at its location $p_i^\mathcal{T}$ in the template.

Stochastic stretching: If the analysis step could not determine any certain landmarks, the manipulation step does stochastic stretching. A contour around the clothing

\[
\Sigma_i = \sum_j \sum_k p_i(j, k) \begin{bmatrix} j - x_{i,lm} \\ k - y_{i,lm} \end{bmatrix} \begin{bmatrix} j - x_{i,lm} & k - y_{i,lm} \end{bmatrix}
\]

is computed for each landmark, the position is transformed to the original image using the inverse homography

\[
k \begin{bmatrix} x_{i,lm} \\ y_{i,lm} \\ 1 \end{bmatrix} = \mathcal{H}^{-1} \begin{bmatrix} x'_{i,lm} \\ y'_{i,lm} \\ 1 \end{bmatrix}
\]

where $k$ is a scalar. The landmarks are finally transformed to a three dimensional point in the world coordinate frame by assuming that all landmarks lie in the plane coinciding with the table. They are then compared with the template to form a set $\mathcal{L}_{err} = \{ i \mid i \notin \mathcal{T} \}$ of all landmarks that do not match their location in the template. If $|\mathcal{L}_{err}| = 0$ the algorithm terminates.

The final part of the analysis step determines the certainty of the landmarks to select between the two modes of operation. As a measure of uncertainty $U_i$ for landmark $i$ the weighted maximum eigenvalue of the covariance matrix $U_i = w_i^\mathcal{T} \lambda_{max}(\Sigma_i)$ is used. If $\min_{i \in \mathcal{L}_{err}} U_i$ is below a certain threshold, landmark argmin$_{i \in \mathcal{L}_{err}} U_i$ is used for landmark placement in the manipulation step, otherwise the manipulation step is configured to do stochastic stretching.
in the transformed image is found by using the OpenCV implementation of Suzuki et al., and a random vertex \( v' = (v'_x, v'_y) \) on the contour is chosen. The contour’s centroid \( c \) is computed to determine a destination point \( p' = (p'_x, p'_y) \) outside of the contour as \( p' = v' + \alpha \frac{v' - c}{\|v' - c\|} \) where \( \alpha \) is the distance to displace the vertex outward, and a source point \( s' = v' + \beta \frac{c - v'}{\|c - v'\|} \) where \( \beta \) is the distance to displace the source point inward. The points \( s' \) and \( p' \) are transformed into two 3D points \( s_a \) and \( p_a \) in the world coordinate frame. The robot then picks the point \( s_a \) and places it at \( p_a \). An advantage of the stochastic stretching approach is that it does not depend on a good state representation in the first place and is therefore applicable to nearly any configuration.

**Data sets**

In this section we introduce all data sets used for the training and evaluation.

**DeepFashion data set**

The DeepFashion: Category and Attribute Prediction Benchmark (DeepFashion data set [http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion/AttributePrediction.html]) [19] is a large collection of fashion images. It consists of 289,222 annotated images that were collected from shopping websites and Google image search. The images cover 46 clothing items with up to 8 different landmarks (i.e. left/right collar, left/right sleeve, left/right waistline and left/right hem). Furthermore for each clothing item a bounding box is provided as well as a total of 1000 clothing attributes.

This data set is used to train the networks with our proposed augmentation methods.

**CTU Color and Depth Image Dataset**

The CTU Color and Depth Image Dataset of Spread Garments (CTU data set [https://github.com/CloPeMa/garment_dataset]) [52] is designed for testing and benchmarking garment segmentation and recognition. This data set exemplifies the unstructured clothing configuration often found in robotic cloth manipulation, meaning the garments are not only spread out flat but also exhibit wrinkles and a huge number of different orientations. The data set contains 1372 images that are taken from a top view of 17 different clothing items divided into 9 categories. We manually labeled landmark positions in each image. We use this data set to train our network and evaluate its performance on more challenging clothing configurations typical in robotics and to evaluate the effect of our augmentation methods when purely trained on the DeepFashion data set.

**In-Lab data set**

While the CTU data set is much closer to a real robotics cloth manipulation task than the DeepFashion data set, we created a small In-Lab data set that is even more typical for robotic tasks. It contains 117 images from 6 different clothing categories (i.e. Tank, Tee, Sweater, Hoody, Jacket, Jeans). To highlight the robotic component, each item is held by two robotic arms at predefined grasping points (i.e. shoulders and waist). The robotic arms are then moved to nine different configuration for each item. This results in the robotic arms occluding part of the garments, making the data set more challenging. Furthermore, the background is not uniform and is partially cluttered. We annotated the images with the same landmarks as in the DeepFashion data set and extracted a similar bounding box around each item. We use this data set to evaluate the performance of our network on previously unseen items in a realistic lab environment.

**Garments used for manipulation experiments**

To evaluate the manipulation algorithm, seven garments from different categories and with different visual features were used. Three of them in the category ‘t-shirt’: one grey with a black star-shaped pattern, one with colorful stripes and one with a wide dark region. Furthermore, we used one grey sweater with colorful thin stripes, a dark and a white blouse, a pair of orange shorts and a pair of blue jeans.

**Learning experiments**

This section describes different experiments designed to evaluate the performance of our network and learning procedure. In this section the individual experiments and results are described in detail.

**Pretraining on the DeepFashion data set**

We use the same settings as in the literature [19,29,30] for training and evaluation. The training set contained 20,922 images while the validation set holds an additional 40,000 images. The test set (used for the final evaluation) is composed of the remaining 40,000 images.

We use the normalized error (NE) [20] as the landmark localization error measure. This is the \( l_2 \) distance between the predicted and ground truth landmark in normalized coordinates. For the category and attribute classification \( top-k \) classification accuracy is used.

Before training, the images are cropped to their bounding boxes. We train our model with and without our proposed data augmentation steps whereas the evaluation is always performed without augmentation. All implementation details can be found in the code base fn: website
Experiments on CTU data set

We perform two types of experiments on the CTU data set. In the first experiment, we analyze the inference performance of our network, solely trained on the entire DeepFashion data set. This is done in order to be able to evaluate the usefulness of the proposed data augmentation methods. In the second experiment, we evaluate the performance of our network when trained and evaluated on the CTU data set.

Experimental setup. In order to use both, the \( \approx 5 \) times larger DeepFashion data set and the CTU data set, we need to resolve the difference in category annotation as they do not exactly overlap. If an item has a collar it is categorized as polo in the CTU data set even though it might look more like a jacket than a polo shirt to a human. Furthermore, the CTU data set distinguishes between long and short sleeve items whereas DeepFashion does not (e.g. t-shirt and t-shirt-long can both be in the Tee category). We combine the categories as follows: bluse = (Blouse), hoody = (Hoodie, Sweater), pants = (Jeans, Jeggings, Joggers, Leggings), polo = (Tee, Button-Down), polo-long = (Button-Down, Henley, Jacket), skirt = (Skirt), t-shirt = (Tee), t-shirt-long = (Cardigan, Sweater, Tee). Note that since the DeepFashion data set does not contain any towels, we ignore them in these experiments.

For the second experiment, we split the CTU images randomly into a train, validate and test set (i.e. 787, 240, 270 images). Both experiments are compared to the publicly available implementation of Liu and Lu.\(^{30}\) We train both models with the same augmentation methods (i.e. no augmentation, elastic warping (EW), rotation (R) and rotation & elastic warping (R & EW)) to make the comparison as fair as possible.

Performance evaluation. The results of landmark prediction and category classification on the CTU data set with pretrained models are shown in Tables 1 (top) and 2, respectively. First we note that the benefit of training with rotated images becomes apparent. That rotations are boosting the performance is not surprising considering the composition of the CTU data set that contains images taken in a high variety of orientations, where in the DeepFashion data set all items of clothing are upright. Adding elastic warping increases the performance further for the landmark prediction for all cases except the one where training was performed on DeepFashion with no rotation. The overall classification accuracy of 85\% shows that our model is able to generalize well even when trained on a data set with significantly different configurations (e.g. items of clothing worn by persons) compared to 56\% reached by Liu and Lu.\(^{30}\)

The results of the second experiment, trained and evaluated on CTU data set, are shown in Table 1 (bottom). Note that landmark predictions are obviously significantly better when learned on the original data set. In this case the elastic warping seems to especially boost the performance in the case of no rotations. We hypothesize that this is probably connected to the data set composition and size as the EW augmented images boost the performance. We omit the category classification results on the CTU data set since all the tested models achieve 100\% accuracy.

We conclude that, adding elastic warping as a data augmentation method improves the performance in most of the evaluated cases. Our network outperforms the one proposed by Liu and Lu\(^{30}\) when trained with the same augmentation methods in both experiments. This indicates that state-of-the-art methods are likely to not generalize well to more challenging robotic focused data sets.

Experiments on In-Lab data set

We leverage our In-Lab data set to investigate the performance of the network solely trained on the DeepFashion data set, and then subsequently used to classify images taken in a robotic lab environment.

The results for landmark prediction and category classification are shown in Tables 3 and 4, respectively. Some landmark predictions are exemplified in Figure 4. Interestingly, the hoody item is almost always misclassified with the exception of the model employing the elastic warping method. Furthermore, the long sleeve t-shirt (Figure 4 top row in the middle) is often classified as a sweater. With these two challenging items the best accuracy we achieve is 78.63\%. Without these two items the accuracy increases to 93.33\%. Due to the limited size of our data set these two items have a significant impact. As the data set is very limited in size, elastic warping can have a negative effect as well, as can be seen, for instance, in the drop in classification accuracy for the class Jacket. Nevertheless, the combination of rotation and elastic warping leads to the best overall performance. The results of the landmark localization also show that our network is able to perform well even when an image contains parts of the robot. We exemplify this useful behaviour in a short video on our project website\(^{fn: website}\), where the landmarks are being continuously detected while the garment is being folded despite the robotic arms occluding large parts of the garment itself.

Elastic warping parameters

We investigated the effect of the elastic warping parameters \( \alpha \) and \( \sigma \) via the performance on a wider range of values. Table 5 shows the results for our model using different \( \alpha \) and \( \sigma \) parameters for the elastic warping for the methods trained on the DeepFashion data set (top) and the CTU data set (bottom). We compare three additional parameter combinations (\( \alpha = [100, 150, 200] \) and \( \sigma = 10 \)) with the previous model (\( \alpha = 500 \) and \( \sigma = 40 \)). We observe that the model with \( \alpha = 100 \) and \( \sigma = 10 \) performs best but landmarks are effected differently from the variation of the elastic warping parameters. This shows that it is possible
| Methods (Trained on DF) | L.Collar | R.Collar | L.Sleeve | R.Sleeve | L.Waistline | R.Waistline | L.Hem | R.Hem | Avg. |
|-------------------------|---------|---------|---------|---------|-------------|-------------|-------|-------|------|
| Liu and Lu<sup>30</sup> | 0.5056  | 0.4810  | 0.3288  | 0.2623  | 0.4908      | 0.4665      | 0.4047| 0.4774| 0.4272|
| Ours                   | 0.4972  | 0.4835  | 0.2846  | 0.2055  | 0.4870      | 0.4677      | 0.4069| 0.4727| 0.4131|
| Liu and Lu<sup>30</sup> EW | 0.5096  | 0.4995  | 0.3314  | 0.2626  | 0.4992      | 0.4730      | 0.4063| 0.4698| 0.4314|
| Ours EW                | 0.5194  | 0.5204  | 0.3538  | 0.2601  | 0.4935      | 0.5251      | 0.4185| 0.4805| 0.4464|
| Liu and Lu<sup>30</sup> R | 0.0947  | 0.1004  | 0.0814  | 0.0670  | 0.1215      | 0.1018      | 0.2196| 0.2177| 0.1255|
| Ours R                 | 0.1056  | 0.1075  | 0.0763  | 0.0708  | 0.1133      | 0.1206      | 0.1756| 0.1526| 0.1153|
| Liu and Lu<sup>30</sup> R & EW | 0.0863  | 0.0880  | 0.0775  | 0.0717  | 0.1030      | 0.1265      | 0.2039| 0.1860| 0.1179|
| Ours R & EW            | 0.0999  | 0.0949  | 0.0639  | 0.0581  | 0.1039      | 0.1151      | 0.1557| 0.1474| 0.1047|

| Methods (Trained on CTU) | L.Collar | R.Collar | L.Sleeve | R.Sleeve | L.Waistline | R.Waistline | L.Hem | R.Hem | Avg. |
|--------------------------|---------|---------|---------|---------|-------------|-------------|-------|-------|------|
| Liu and Lu<sup>30</sup> | 0.0560  | 0.0484  | 0.0473  | 0.0572  | 0.0473      | 0.0560      | 0.1010| 0.0929| 0.0632|
| Ours                    | 0.0500  | 0.0801  | 0.0790  | 0.0745  | 0.0590      | 0.0713      | 0.0749| 0.0853| 0.0719|
| Liu and Lu<sup>30</sup> EW | 0.0447  | 0.0442  | 0.0447  | 0.0481  | 0.0612      | 0.0826      | 0.0860| 0.0780| 0.0612|
| Ours EW                 | 0.0260  | 0.0267  | 0.0319  | 0.0262  | 0.0311      | 0.0359      | 0.0620| 0.0548| 0.0368|
| Liu and Lu<sup>30</sup> R | 0.0299  | 0.0314  | 0.0289  | 0.0335  | 0.0560      | 0.0402      | 0.0539| 0.0460| 0.0400|
| Ours R                  | 0.0181  | 0.0194  | 0.0253  | 0.0192  | 0.0374      | 0.0382      | 0.0314| 0.0383| 0.0284|
| Liu and Lu<sup>30</sup> R & EW | 0.0295  | 0.0277  | 0.0370  | 0.0403  | 0.0350      | 0.0561      | 0.0483| 0.0509| 0.0406|
| Ours R & EW             | 0.0199  | 0.0248  | 0.0348  | 0.0244  | 0.0274      | 0.0204      | 0.0334| 0.0276| 0.0266|

R & EW: rotation & elastic warping; DF: DeepFashion.

The values represent the normalized error (NE). Best results are marked in bold.
with enough tuning to find a more suitable elastic warping parameter that improve the generalization towards the target distribution. We are therefore confident that the elastic warping data augmentation method can be used to improve generalization towards a real world scenario as well.

**Robotic experiments**

The algorithm described in the ‘Manipulation strategy’ section was implemented on a Baxter robot. The proposed network is used to perform cloth manipulation with the aim to stretch garments. The robot is presented with garments in different predefined starting states and the evaluation criteria is based on whether it can bring the garment into the state described by its template. The experiments is ended either when the state was within the tolerance from the template or when manually terminated.

**Experimental setup**

The manipulation clothing was tested with the initial states: folded hem, folded collar, folded sleeves, folded waist, folded legs and crumbled, see Table 6. All experiments were repeated five times with a model trained on the DeepFashion data set with rotation and with elastic warping parameters $\alpha = 150$ and $\sigma = 10$. To extensively evaluate the stretching with landmarks performance, the template are selected manually for the first set of experiments. To compare the performance of the full algorithm as described in the ‘Manipulation strategy’ section, the experiments folded hem and folded collar for garments Tee (stars) and Tee (stripes) were repeated with classification activated (five trials each). As a final set of experiments, the starting configuration folded hem for Tee (stars) was performed using a model trained on the CTU data set with $\alpha = 150$ and $\sigma = 10$, and with five trials (template set manually).

A ‘–’ is placed where the experiment is not applicable.

Each experiment was repeated five times. The limits for class and landmark certainty and the weights were chosen empirically, see Table 7. A tolerance of $\varepsilon = 0.06$, a lower limit on class certainty of 0.4, and an upper limit on landmark uncertainty of 3500 was used for all experiments. Templates for the classes Blouse, Tee, Sweater, Jeans and Shorts were available at all times.

**Manipulation results**

The success rates for the experiments are shown in Table 6. The results are summarized in Table 8 where ‘manually terminated’ are the proportion of failures that were terminated manually because the solution was not advancing, ‘closeness’ indicates how close the manually terminated experiments were from being solved, ‘closeness stdev’ is the standard deviation of the closeness number, ‘bad move’ is the proportion of failures that were terminated manually because of the robot making an irrecoverable bad move, and

Table 2. Results on CTU data set category classification with different augmentation methods, when trained on the DF data set.

| Methods (Trained on DF) | Blouse | Hoody | Pants | Shirt | Polo-long | Top-1 | Top-3 | Top-1 | Top-3 | Top-1 | Top-3 |
|-------------------------|--------|-------|-------|-------|-----------|-------|-------|-------|-------|-------|-------|
| Liu and Lu30            | 35.00  | 31.58 | 37.63 | 33.83 | 37.63     | 35.00 | 31.58 | 37.63 | 33.83 | 37.63 | 35.00 |
| Ours                    | 20.00  | 15.05 | 21.05 | 19.05 | 21.05     | 20.00 | 15.05 | 21.05 | 19.05 | 21.05 | 20.00 |
| Liu and Lu30 R R & EW   | 55.00  | 51.43 | 57.14 | 53.85 | 57.14     | 55.00 | 51.43 | 57.14 | 53.85 | 57.14 | 55.00 |
| Ours R & EW             | 80.00  | 76.69 | 82.86 | 79.47 | 82.86     | 80.00 | 76.69 | 82.86 | 79.47 | 82.86 | 80.00 |

R & EW: rotation & elastic warping, DF: DeepFashion. *Best results marked in bold.
Table 3. Results on In-Lab data set for landmark localization on unknown items of clothing.a

| Methods (Trained on DF) | L.Collar | R.Collar | L.Sleeve | R.Sleeve | L.Waistline | R.Waistline | L.Hem | R.Hem | Avg.   |
|-------------------------|----------|----------|----------|----------|-------------|-------------|-------|-------|--------|
| Liu and Lu30            | 0.0819   | 0.1061   | 0.0910   | 0.0975   | 0.0185      | 0.0175      | 0.0437| 0.0788| 0.0669 |
| Ours                    | **0.0557**| **0.0682**| 0.0947   | 0.1234   | **0.0177**   | **0.0135**   | 0.0497| 0.0908| 0.0642 |
| Liu and Lu30 EW         | 0.0910   | 0.1059   | 0.0915   | 0.0470   | 0.0341      | 0.0196      | **0.0405**| 0.0690| **0.0623**|
| Ours EW                 | 0.0698   | 0.0923   | 0.1193   | 0.0843   | 0.0380      | 0.0315      | 0.0458| **0.0525**| 0.0667 |
| Liu and Lu30 R          | 0.0620   | 0.0930   | 0.0924   | 0.0663   | **0.0139**   | 0.0171      | 0.0478| 0.1035| 0.0620 |
| Ours R                  | 0.0621   | **0.0767**| 0.0949   | 0.0576   | 0.0527      | **0.0134**   | 0.0926| 0.0998| 0.0687 |
| Liu and Lu30 R & EW     | 0.0657   | 0.1135   | 0.0892   | 0.0523   | 0.0163      | 0.0206      | 0.0586| **0.0662**| 0.0603 |
| Ours R & EW             | **0.0532**| 0.1129   | **0.0827**| 0.0535   | 0.0155      | 0.0202      | 0.0524| 0.0817| **0.0590**|

R & EW: rotation & elastic warping; DF: DeepFashion.

The values represent the normalized error (NE). Best result marked in bold.

Table 4. Classification accuracy on In-Lab data set for unknown items of clothing.a

| Methods                      | Hoodie | Jacket | Sweater | Tank | Tee | Jeans | Overall |
|------------------------------|--------|--------|---------|------|-----|-------|---------|
| Liu and Lu30                 | 00.00  | 100.0  | 84.21   | 100.0| 55.56| 100.0 | 71.65   |
| Ours                         | 05.88  | 88.89  | 100.0   | 100.0| 62.96| 96.30 | 76.07   |
| Liu and Lu30 R               | 00.00  | 11.11  | 100.0   | 77.78| 62.96| 100.0 | 66.67   |
| Ours R                       | 00.00  | 100.0  | 100.0   | 100.0| 62.96| 100.0 | 76.92   |
| Liu and Lu30 R & EW          | 00.00  | 77.78  | 100.0   | 100.0| **66.67**| 100.0 | 76.07   |
| Ours R & EW                  | **47.06**| 88.89  | 94.74   | 100.0| 51.05| 96.30 | **78.63**|

R & EW: rotation & elastic warping.

*Best result marked in bold.

Figure 4. Example images of the landmark localization on our In-Lab data set. The categories are from top left to bottom right: Tank, Tee, Sweater, Hoody, Jacket, Jeans. Robot arms are visible in the images. The predicted heatmaps are shown in red and the blue crosses denote the selected maximum values.
| Methods (Trained on DF) | L.Collar | R.Collar | L.Sleeve | R.Sleeve | L.Waistline | R.Waistline | L.Hem | R.Hem | Avg. |
|------------------------|----------|----------|----------|----------|-------------|-------------|-------|-------|------|
| Ours EW (α = 150, σ = 10) | 0.5048 | 0.4982 | 0.3116 | 0.2893 | 0.4796 | 0.4386 | 0.4190 | 0.4708 | 0.4265 |
| Ours EW (α = 100, σ = 10) | 0.5003 | 0.4829 | 0.2915 | 0.2334 | 0.4689 | 0.4629 | 0.4146 | 0.4638 | 0.4148 |
| Ours EW (α = 200, σ = 10) | 0.5174 | 0.4926 | 0.2813 | 0.2372 | 0.4918 | 0.4393 | 0.4666 | 0.4766 | 0.4216 |
| Ours EW (α = 500, σ = 40) | 0.5194 | 0.5204 | 0.3538 | 0.2601 | 0.4935 | 0.5251 | 0.4185 | 0.5805 | 0.4644 |
| Ours R & EW (α = 150, σ = 10) | 0.1075 | 0.0970 | 0.0718 | 0.0715 | 0.0976 | 0.1083 | 0.1505 | 0.1569 | 0.1076 |
| Ours R & EW (α = 100, σ = 10) | 0.0981 | 0.0904 | 0.0689 | 0.0618 | 0.0838 | 0.0963 | 0.1530 | 0.1643 | 0.1021 |
| Ours R & EW (α = 200, σ = 10) | 0.0977 | 0.1058 | 0.0801 | 0.0643 | 0.0920 | 0.1192 | 0.1683 | 0.1747 | 0.1128 |
| Ours R & EW (α = 500, σ = 40) | 0.0999 | 0.0949 | 0.0639 | 0.0581 | 0.1039 | 0.1151 | 0.1557 | 0.1474 | 0.1047 |
| Methods (Trained on CTU) | L.Collar | R.Collar | L.Sleeve | R.Sleeve | L.Waistline | R.Waistline | L.Hem | R.Hem | Avg. |
|------------------------|----------|----------|----------|----------|-------------|-------------|-------|-------|------|
| Ours EW (α = 150, σ = 10) | 0.0263 | 0.0336 | 0.0273 | 0.0361 | 0.0431 | 0.0407 | 0.0483 | 0.0512 | 0.0383 |
| Ours EW (α = 100, σ = 10) | 0.0261 | 0.0252 | 0.0264 | 0.0268 | 0.0303 | 0.0444 | 0.0536 | 0.0480 | 0.0354 |
| Ours EW (α = 200, σ = 10) | 0.0324 | 0.0323 | 0.0304 | 0.0472 | 0.0210 | 0.0179 | 0.0555 | 0.0515 | 0.0360 |
| Ours EW (α = 500, σ = 40) | 0.0260 | 0.0267 | 0.0319 | 0.0262 | 0.0311 | 0.0359 | 0.0620 | 0.0548 | 0.0368 |
| Ours R & EW (α = 150, σ = 10) | 0.0282 | 0.0251 | 0.0230 | 0.0291 | 0.0179 | 0.0256 | 0.0293 | 0.0285 | 0.0258 |
| Ours R & EW (α = 100, σ = 10) | 0.0216 | 0.0186 | 0.0275 | 0.0237 | 0.0252 | 0.0314 | 0.0239 | 0.0275 | 0.0249 |
| Ours R & EW (α = 200, σ = 10) | 0.0222 | 0.0239 | 0.0157 | 0.0180 | 0.0356 | 0.0302 | 0.0212 | 0.0348 | 0.0252 |
| Ours R & EW (α = 500, σ = 40) | 0.0199 | 0.0248 | 0.0348 | 0.0244 | 0.0274 | 0.0204 | 0.0334 | 0.0276 | 0.0266 |

R & EW: rotation & elastic warping; DF: DeepFashion.

*The values represent the normalized error (NE).
"false success" is the proportion of failures that the algorithm reported as a success. The results of running the complete algorithm with classification are shown in Table 9, where 'no class' means that the algorithm was manually terminated because no class was ever determined. The experiments on the CTU data set resulted in 0 successes. Recordings of all experiments are available at the website.

The closeness is computed as the mean error of the landmarks compared to the template. Each time the landmarks are measured in the analysis step, the mean distance to their position in the template is computed based on those measurements. The minimum value during one execution of an experiment, or the minimum mean error, is taken as a measure of closeness to the solution for that execution. The mean of all minimum mean errors, taken across all executions of all experiments for a class that resulted in manual termination, is taken as mean minimum mean error and is reported as 'closeness' in Table 8.

The manipulation strategy has no notion of in what order the landmarks should be placed, it merely picks the one with least uncertainty. This leads to the common failure scenario of placing a landmark in a way that moves the clothing into a state that is much harder to solve. Another failure reason is incorrect output of the network. For some configurations of the clothing the network is over confident in the position of the landmarks, making the manipulation strategy perform a bad move. The extent of this depends on

| Table 6. Success rate of manipulation with model trained on the DeepFashion data set with elastic warping parameters $\alpha = 150$ and $\sigma = 10$.a |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Experiment                      | Tee (stars)     | Tee (stripes)   | Tee (dark)      | Sweater         | Blouse          | Shorts          | Jeans           |
| Folded hem                      | 1               | 0.8             | 0               | 0.4             | 0.2             | -              | -              |
| Folded collar                   | 0.8             | 1               | 0               | 0.4             | 0.6             | -              | -              |
| Folded sleeves                  | -               | -               | -               | 0.6             | 1               | -              | -              |
| Folded legs                     | -               | -               | -               | -               | -               | 0              | 0              |
| Folded waist                    | -               | -               | -               | -               | -               | 0.8            | 1              |
| Crumbled                        | 0.2             | 0               | 0               | 0               | 0               | 0              | 0              |

| Table 7. Weights and relevant landmarks for the robotic experiments.a Omitted weights have a value of 1. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Class                           | Landmarks       | Weights         |
| Blouse                          | 0–3, 6, 7       | $w^1_0 = w^1_1 = 1.6, w^1_2 = w^1_3 = 0.7$ |
| Tee                             | 0–3, 6, 7       | $w^2_0 = w^2_1 = 1.2, w^2_2 = w^2_3 = 0.7$ |
| Sweater                         | 0–3, 6, 7       | $w^3_2 = w^3_3 = 0.7$ |
| Jeans                           | 4–7             | $w^4_2 = w^4_3 = 0.6, w^4_4 = w^4_5 = 0.8$ |
| Shorts                          | 4–7             | $w^5_4 = 0.5, w^5_5 = 0.4, w^5_6 = w^5_7 = 0.8$ |

| Table 8. Summary of rates of success and failures for each class with model trained on the DeepFashion data set with elastic warping parameters $\alpha = 150$ and $\sigma = 10$.a |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Class                           | Success rate    | Manually terminated | Closeness | Coloseness stdev | Bad move | False success |
| Tee (stars)                     | 0.67            | 0.6             | 0.052         | 0.014           | 0.4       | 0              |
| Tee (stripes)                   | 0.6             | 0.5             | 0.067         | 0.035           | 0.5       | 0              |
| Tee (dark)                      | 0               | 0.8             | 0.15          | 0.068           | 0.2       | 0              |
| Sweater                         | 0.35            | 0.62            | 0.052         | 0.018           | 0.31      | 0.08           |
| Blouse                          | 0.45            | 0.45            | 0.048         | 0.0057          | 0.55      | 0              |
| Shorts                          | 0.27            | 0.36            | 0.082         | 0.040           | 0.64      | 0              |
| Jeans                           | 0.33            | 0.6             | 0.080         | 0.044           | 0.4       | 0              |

* a: Closeness is the mean minimum mean error of the manually terminated experiments.

| Table 9. Success rates [0–1] for experiments folded hem and folded collar for garments tee (stars) and tee (stripes) with classification. |
|---------------------------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                              | Tee (stars)     | Tee (stripes)   |
|                              | Folded hem      | Folded collar   | Folded hem      | Folded collar   |
| Success                       | 0.6             | 0.4             | 0               | 0.2             |
| Manually                      | 0.2             | 0.4             | 0.4             | 0.2             |
| Bad move                      | 0.2             | 0.2             | 0.2             | 0               |
| No class                      | 0               | 0               | 0.4             | 0.6             |
the garment being used, and the effect can also appear when the manipulation strategy tries to place the landmarks in a bad order, resulting in a challenging state as discussed previously.

**Discussion and limitations**

The method was sensitive to lighting conditions and to the color of the garment. A garment with similar color to the background showed to be problematic for the method as can be seen for Tee (dark) in Table 8. Garments that had smaller parts with a similar color to the background or garments with an overall slightly similar color could cause problems on the contour detection, as the detection of the contour would miss part of the garments and present an incomplete image to the network (see Figure 5). The dependence on lighting can be observed for the garments Sweater and Jeans in Table 8, where the success rate is low and the rate of manual termination is high. Wrinkles and small displacements could influence the detected class with the effects largely affected by the lighting conditions.

The model trained on the CTU data set had poor performance, with no successes at all. Even though the CTU data set is more similar to the application than the DeepFashion data set.

This indicates that the smaller size of the CTU data set has lead to over fitting and showcases the importance of using large-scale data sets and data augmentation methods like elastic warping combined with more robust manipulation strategies.

The experiments show the potential in using landmark placement for robotic cloth manipulation. As can be seen in Table 6 some simple cases had a high success rate and there is a possibility of solving the hard initially crumbled state. Furthermore, the method has a low rate of false success, it can accurately determine whether the garment is stretched.

**Conclusion and future work**

We presented a complete cloth manipulation framework based on category classification and landmark detection. We use a large publicly available fashion image data set with a data augmentation method called elastic warping to train a network for garment classification and landmark detection for robotic manipulation application. We evaluate the performance of the network and the effects of the elastic warping thoroughly. We show that the parameter of the method can be tuned to fit a desired target distribution. Furthermore, we perform a wide set of real world robotic experiments where the goal is to stretch the garment from different starting configurations and provide all experimental videos on our supplementary website: website. This extensive evaluation highlights the importance of more robust preprocessing methods as the used contour detection, as it is susceptible to different lighting conditions as well as erroneous if the garment color is similar to the background color. Finally, we show the inadequacy of using smaller data set for robotic purposes when dealing with novel clothing items, by comparing the performance of our method trained on a large-scale fashion data set with the performance trained on a robotic specific data set. Furthermore we plan to incorporate the learning component also in the manipulation step to formulate more robust manipulation strategies and combine the stretching step with the manipulation step. Performing an investigation towards the effect of occlusions.

**Authors’ note**

Oscar Gustavsson and Thomas Ziegler contributed equally to this article.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed the receipt of the following financial support for the research, authorship, and/or publication of this article: This research has been financed by the Swedish Research Council, Knut and Alice Wallenberg Foundation and European Research Council (ERC) (grant Agreement No. 884807).

**ORCID iD**

Michael C Welle https://orcid.org/0000-0003-3827-3824

**Supplemental material**

Supplemental material for this article is available online.

**References**

1. Bohg J, Morales A, Asfour T, et al. Data-driven grasp synthesis – a survey. *IEEE Trans Robot* 2014; 30(2): 289–309.
2. Bohg J, Hausman K, Sankaran B, et al. Interactive perception: leveraging action in perception and perception in action. *IEEE Trans Robot* 2017; 33(6): 1273–1291.

3. Bütepage J, Cruciani S, Kokie M, et al. From visual understanding to complex object manipulation. *Ann Rev Contr Robot Auto Syst* 2019; 2: 161–179.

4. Billard A and Kragic D. Trends and challenges in robot manipulation. *Science* 2019; 364(6446). DOI: 10.1126/science.aat8414.

5. Arriola-Rios VE, Güler P, Ficuciello F, et al. Modeling of deformable objects for robotic manipulation: a tutorial and review. *Front Robot AI* 2020; 7: 82.

6. Jiménez P. Survey on model-based manipulation planning of deformable objects. *Robot Comput-Integr Manuf* 2012; 28(2): 154–163.

7. Yin H, Varava A and Kragic D. Modeling, learning, perception, and control methods for deformable object manipulation. *Sci Robot* 2021; 6(54): eab8803.

8. Ramisa A, Alenyà G, Moreno-Noguer F, et al. A 3D descriptor to detect task-oriented grasping points in clothing. *Pattern Recog* 2016; 60: 936–948.

9. Corona E, Alenyà G, Gabas A, et al. Active garment recognition and target grasping point detection using deep learning. *Pattern Recog* 2018; 74: 629–641.

10. Doumanoglou A, Stria J, Peleka G, et al. Folding clothes autonomously: a complete pipeline. *IEEE Trans Robot* 2016; 32(6): 1461–1478.

11. Lippi M, Poklukar P, Welle MC, et al. Latent space roadmap for visual action planning of deformable and rigid object manipulation. In 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021, pp. 5619–5626. IEEE.

12. García-Camacho I, Lippi M, Welle MC, et al. Benchmarking bimanual cloth manipulation. *IEEE Robot Auto Lett* 2020; 5(2): 1111–1118.

13. Sun L, Aragon-Camarasa G, Rogers S, et al. Single-shot clothing category recognition in free-configurations with application to autonomous clothes sorting. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), 2017, pp. 6699–6706.

14. Doumanoglou A, Kargakos A, Kim T, et al. Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In: 2014 IEEE international conference on robotics and automation (ICRA), Hong Kong, China, 31 May–07 June 2014, pp. 987–993. DOI: 10.1109/ICRA.2014.6906974.

15. William B, Walker I and Birchfield S. A new approach to clothing classification using mid-level layers. In: 2013 IEEE international conference on robotics and automation, Karlsruhe, Germany, 06–10 May 2013, pp. 4271–4278. DOI: 10.1109/ICRA.2013.6631181.

16. Li Y, Wang Y, Case M, et al. Real-time pose estimation of deformable objects using a volumetric approach. In: 2014 IEEE/RSJ international conference on intelligent robots and systems, Chicago, IL, USA, 14–18 September 2014, pp. 1046–1052. DOI: 10.1109/IROS.2014.6942687.

17. Stria J and Hlaváč V. Classification of hanging garments using learned features extracted from 3D point clouds. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain, 01–05 October 2018, pp. 5307–5312. DOI: 10.1109/IROS.2018.8593741.

18. Liang X, Liu S, Shen X, et al. Deep human parsing with active template regression. *IEEE Trans Patt Anal Mach Intell* 2015; 37(12): 2402–2414.

19. Liu Z, Luo P, Qiu S, et al. DeepFashion: powering robust clothes recognition and retrieval with rich annotations. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016, pp. 1096–1104. DOI: 10.1109/cvpr.2016.124.

20. Liu Z, Yan S, Luo P, et al. Fashion landmark detection in the wild. In: Leibe B, Matas J, Sebe N, et al. (eds) *Computer Vision – ECCV 2016*, Cham: Springer International Publishing. ISBN 978-3-319-46475-6, pp. 229–245.

21. Huang J, Feris R, Chen Q, et al. Cross-domain image retrieval with a dual attribute-aware ranking network. In: 2015 IEEE international conference on computer vision (ICCV), Santiago, Chile, 07–13 December 2015, pp. 1062–1070. DOI: 10.1109/ICCV.2015.127.

22. Lu Y, Kumar A, Zhai S, et al. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017, pp. 1131–1140. DOI: 10.1109/CVPR.2017.126.

23. Ge Y, Zhang R, Wang X, et al. Deepfashion2: a versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5337–5345.

24. Ma Y, Jia J, Zhou S, et al. Towards better understanding the clothing fashion styles: a multimodal deep learning approach. In: Proceedings of the thirty-first AAAI Conference on Artificial Intelligence (AAAI-17), pp. 38–44.

25. Han X, Wu Z, Jiang YG, et al. Learning fashion compatibility with bidirectional LSTMs. In: Proceedings of the 25th ACM international conference on multimedia, MM ’17, New York, NY, USA, ACM. ISBN 978-1-4503-4906-2, pp. 1078–1086. DOI: 10.1145/3123266.3123394.

26. Kiapour MH, Han X, Lazebnik S, et al. Where to buy it: matching street clothing photos in online shops. In: 2015 IEEE international conference on computer vision (ICCV), Araucano Park, 11–18 December 2015, pp. 3343–3351. DOI: 10.1109/ICCV.2015.382.

27. Ziegler T, Bütepage J, Welle MC, et al. Fashion landmark detection and category classification for robotics. In: 2020 IEEE international conference on autonomous robot systems and competitions (iCARSC), Piscataway, NJ, 2020, pp. 81–88. IEEE.

28. Yan S, Liu Z, Luo P, et al. Unconstrained fashion landmark detection via hierarchical recurrent transformer networks. In: Proceedings of the 25th ACM international conference on
