RobotFashion: A dataset on classifying and localizing clothing in deformed state

Deepika Anantha Padmanaban
deap@kth.se

Nik Vaessen
vaessen@kth.se

Sayyed Ali Kiaian Mousavy
sakm2@kth.se

Sri Datta Budaraju
budaraju@kth.se

Abstract

Fashion datasets have been extensively used in the clothing industry for a variety of tasks. The focus of this project has been to create a dataset that fulfils these criteria for robotic manipulation, where the clothing item is being held by a robot in various other configurations than the default full-view paired with the corresponding depth images of the clothing item. The dataset created closely follows the classes in DeepFashion2[4] and is meant to support transferring knowledge from DeepFashion2[4] to fit the requirement for robotic manipulation. The learnability of the dataset has been measured by using a Faster R-CNN[11] for classification and localization specific tasks. The code can be found at `https://github.com/nikvaessen/robotfashion`.

1 Introduction

Cloth Perception and Manipulation has become an active research topic during recent years because of its potential in the industry. Fashion datasets, such as Deepfashion2[4],Deep Fashion[9], ModaNet[14], are proposed to address variations of clothes styles, textures, and form in real-world usages. Though these datasets have already created their own niche of applications, the clothing items in these datasets include images in real-world scenario with clothes on a human or perfectly set on a table.

In a robotic environment, such datasets particularly seem to fail as they do not include robots in the scene. Also, the fact that clothes are deformable and have high chances of being handled in an orientation that need not necessarily give a full visibility of a clothing image to be detected by a machine in a robotic manipulation task, is not considered by most of the currently available datasets. The closest to the required task in hand is the Glasgow's Stereo Image Database of Garments[2], which is a database of only 80 stereo-pair color images collected from a set of 16 clothing items.

To address this challenge, we have gathered different clothing items and collected images of them being held by a robot in various orientations and a couple of viewpoints and annotated them with labels and bounding boxes such that the dataset can be used for standard classification and localization tasks. We defined sampling protocols to be able to capture the images from the Baxter robot in predictable positions.

The resulting samples were region annotated deformed clothes photos in robotic hands. Due to the comparably small size of collected samples for labels, we used labels from Deepfashion2 proposed categories to be able to facilitate transfer learning from existing networks pre-trained on that dataset.

Preprint. Under review.

To evaluate the newly created dataset, we have benchmarked our data on the PyTorch [10] implementation of faster-R-CNN [11] pre-trained on COCO [8] 2017 and then retrained on Deepfashion2 to aggregate the learned features.

1.1 Related work

1.1.1 On-Person Clothing Datasets

There are several clothing datasets including [4, 9, 14] which present a large number of annotated images with people wearing a broad number of fashion items. For example, DeepFashion [9] has 800K images from 50 categories and DeepFashion2 [4] has 491K images from 13 categories. The annotations include class labels, bounding boxes, landmarks, key points and semantic masks. ModaNet [14] also includes fashion objects like belts, bags, boots, scarfs and ties and provide polygon annotations. These datasets can be leveraged to train models for the tasks of object detection, key point regression, semantic segmentation and instance segmentation.

1.1.2 Deformed Clothing Datasets

There are a few deformed clothing datasets from CloPeMa (Clothes Perception and Manipulation) project [3] such as [2, 13, 1] which focus on learning to detect clothes and their relevant features. The aim is to teach robots perceive and manipulate clothes.

Glasgow's stereo image dataset [2] contains 80 stereo pair images from 16 different clothing items with mask annotation obtained from the stereo depth. The clothes were photographed in various positions, i.e., open on a table, folded, wrinkled, and held by robotic arms using a pair of DSLRs as the robot's head. Where CTU dataset [13] also has 17 unique clothing items placed on a flat table folded to some extent. This dataset does not include robot or robots arm interacting with the clothes but includes key point annotations and depth information. These datasets being confined to a very small number of clothing items, makes it impossible to train models and to generalise for any real-world scenarios.

The Autonomous active recognition dataset [1] on the other hand also uses 24 clothes but generates 28,000 images with depth and landmark annotations. The authors also propose random forest based approach to detect landmark in depth images. The data collection was scaled by suspending the clothes from robotic arms and rotating 360°. Thus over-fitting the comparatively large dataset to only 24 clothes making it harder to justify that the algorithms proposed could be generalised. Moreover, this dataset could not be accessed publicly.

2 Methodology

2.1 Data Collection

In this research, clothing items from recycle collection points were donated to the research team. For the first part of the work, a framework for creating a dataset was proposed. Since the primary purpose of the project was to create a fashion dataset that can be useful for robotic manipulation tasks, we decided to make use of Baxter, the robot available in the RPL lab at KTH. The idea was to create a dataset with images that had clothing items being held by the robot in multiple orientations such that robotic arms where included in the scene. The orientations of the clothing items were narrowed down to five states with varying levels of occlusion - from fully-visible to more than half the clothing item being occluded. Each clothing item was captured in two different views. We used Intel RealSense depth and tracking cameras for the purpose. The orientations that were considered are;

1. Clothing item held perpendicular to the view of the camera - no occlusion of the clothing item.
2. Clothing item folded in half, leading to partial occlusion with only a few landmarks of the cloth being visible depending on the view.
3. Clothing item held by the robot with one arm held high and the other held down, leading to skewed view of the orientation 1 with slight occlusion in the ends.
4. Clothing item held parallel to the view of the camera, leading to heavy occlusion with most of the landmarks being occluded.

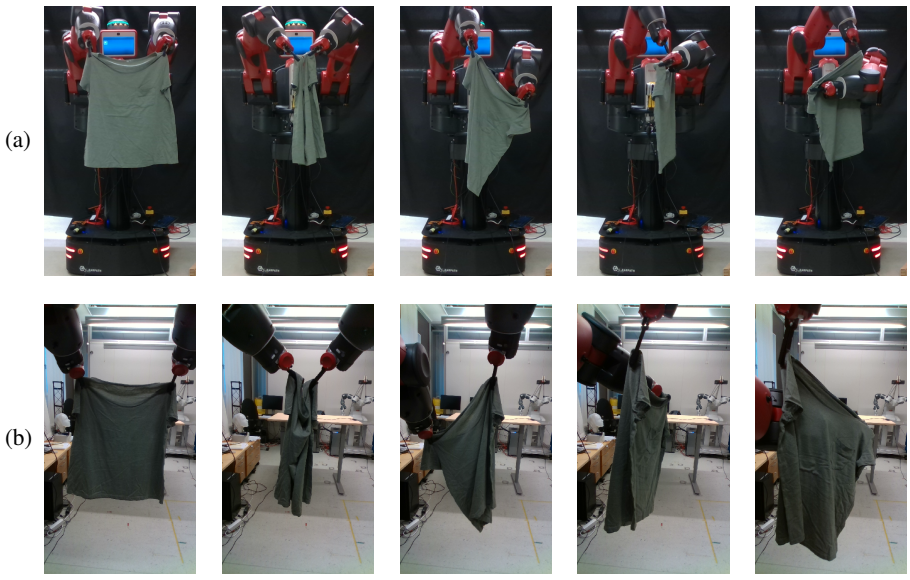


Figure 1: The orientations of the clothing items as seen from the (a) front view camera (b) rear view camera on the robot

5. Clothing item held in such a way that the occlusion of the image is due to the robotic arm overlapping over the cloth's view.

The above orientations are shown in the figure 1 for both the front and rear view camera. The orientations were recorded as a movement of the robot arms to use the same pattern for all the clothing items. The video was captured by the realsense camera at 6fps with a resolution of 1280 x 720. The frames from the video was extracted in the npz format using the realsense SDK. These npz files were then processed to obtain the corresponding depth and color images. Each of the clothing item was with 5 orientations were recorded as a 45secs video, summing upto a total of 270 frames, for each camera. The best frames for each of the five orientations were handpicked for each camera. Thus, for every clothing item, we have 10 pictures - 5 orientations each from the two cameras.

2.2 Data Labelling

The frames that were handpicked from the raw data were cropped to maintain the same size and aspect ratio. The processed images were annotated using the labelImg software[12]. Each image was labelled with their corresponding class name for the classification task and a bounding box was tightly fit around the clothing item for the localization task. All our images have only one instance of the object and hence, only one bounding box per image. The annotations were stored in the PascalVOC format. An example of the annotated image can be found in figure 2.

2.3 Dataset Details

There were a total of 362 clothing items that we had access to and used for collecting data. Each of the clothing item has been tagged with a unique ID. The class labels of the clothing items are closely inspired by those in DeepFashion2 inorder to facilitate Transfer Learning as the number of samples of data we have is too less and imbalanced for deep learning purposes. The classes of items are short sleeve top, long sleeve top, long sleeve outerwear, vest, sling, shorts, trousers, skirt, short sleeve dress, long sleeve dress, vest dress and sling dress. Unfortunately we did not have any sample for short sleeve outerwear as in DeepFashion2. Thus, we had samples for 12 out of the 13 classes available in DeepFashion2, but with a very few samples for a few of the classes, which were not considered while training.The distribution of the number of items available in each of the class can be found in table 1.

For illustrative purposes we have a pair of color-depth image representing each class shown in the figure 3 and 4.

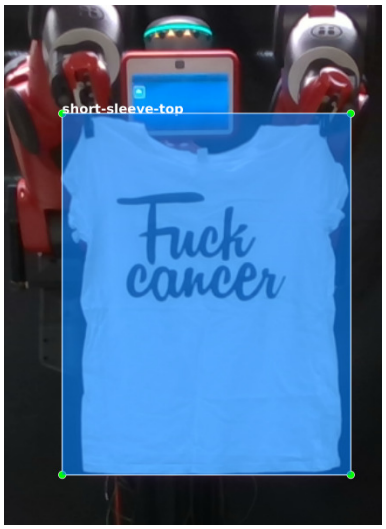


Figure 2: An annotated image showing the class label and bounding box

Table 1: Distribution of the clothing in the collected data. The class labels are equivalent to the class labels used in DeepFashion 2 [4]. The amount of items for the classes short sleeve outwear, sling and sling dress are so low that they did not end up in the final dataset.

Class label	Number of unique items	Number of images
short sleeve top	98	979
long sleeve top	67	666
short sleeve outwear	0	0
long sleeve outwear	44	433
vest	28	280
sling	2	20
shorts	7	65
trousers	84	834
skirt	14	140
short sleeve dress	9	90
long sleeve dress	3	30
vest dress	5	50
sling dress	1	10

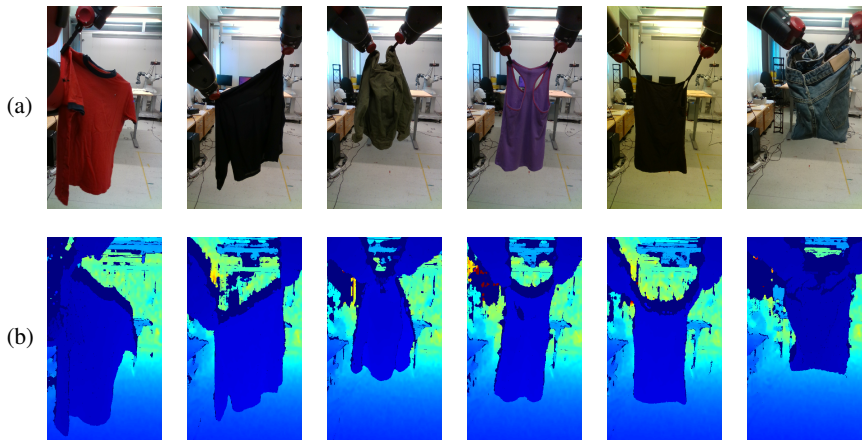


Figure 3: The data samples of (a) RGB image (b) depth image of Short Sleeve Top, Long Sleeve Top, Long Sleeve Outerwear, Vest, Sling and Shorts respectively.

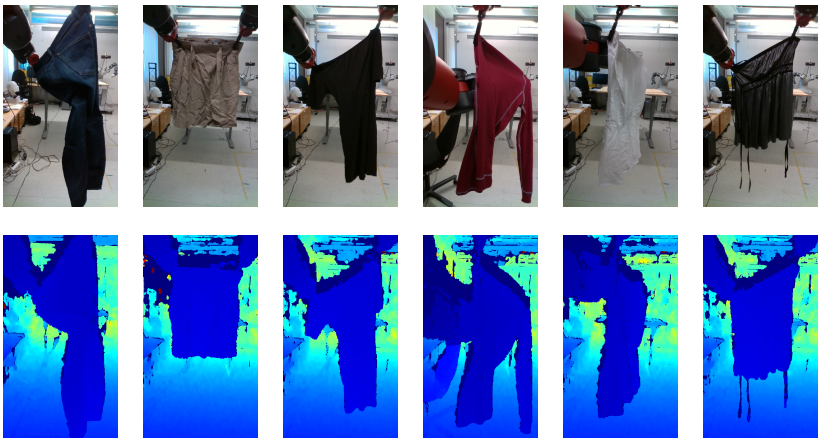
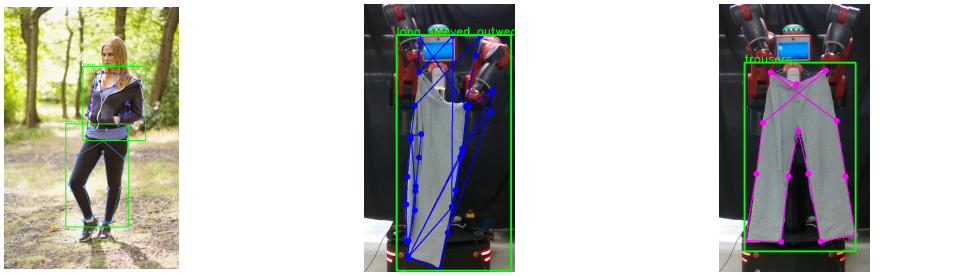


Figure 4: The data samples of (a) RGB image (b) depth image of Trousers, Skirt, Short Sleeve Dress, Long Sleeve Dress, Vest Dress and Sling Dress respectively.



(a) Google image (b) RobotFashion stretched item (c) RobotFashion deformed item

Figure 5: Resulting soft annotation on different test cases

3 Soft labels

Initial pre-experiments were conducted with the DeepFashion 2 dataset regarding soft annotation. In the pre-experiment, the Keypoint R-CNN model implementation in PyTorch is pre-trained on the entire DeepFashion 2 training dataset in order to learn the landmarks and bounding boxes of different classes of items. For this training, optimizer Adam [6] was used with *learning rate* $= 0.0005$, $\beta_1 = 0.9$ and was trained for 6 epochs. Results in validation steps shown that the 4th epoch achieved the best performance and was used in the evaluation.

The pre-experiment trained Keypoint-RCNN model on the DeepFasion 2 dataset was then evaluated on both google images and on RobotFashion images, and the resulting annotation was salvaged. Results shown that the trained model had acceptable performance in soft annotating google images data, especially those data that contains people wearing the items. The results also shown that the resulting soft annotation was usable on the cases of RobotFashion where the clothes were not deformed. In cases of RobotFashion data where the clothes were deformed, the results were too chaotic to be corrected and reused for training purposes.

4 Experiments

We validate the quality of the dataset by training and evaluating a PyTorch [10] implementation of Faster R-CNN [11] for the object detection task. The model uses ResNet-50 [5] with Feature Pyramid Network [7] as backbone, and is pre-trained on the COCO [8] dataset. For all the experiments the backbone layers were frozen.

The experiments were conducted with the use of 2 datasets. The first datset, termed by us as RobotFashion, is the dataset we collected ourselves. RobotFashion has 2,586 images for training,

Table 2: Evaluation of the trained models on the test set of RobotFashion. Both models were evaluated on both the classification accuracy as well as the mean Intersection over Union. The predictions column (indicated with #) displays the amount of times the model predicted an image to be from that class.

Classes		Trained on RobotFashion			Trained on DeepFashion 2		
description	samples	#	accuracy	mIoU	#	accuracy	mIoU
all	618	-	0.196	0.724	-	0.262	0.505
short sleeve top	149	5	0.000	0.724	23	0.087	0.449
long sleeve top	111	138	0.099	0.758	21	0.018	0.509
long sleeve outwear	79	241	0.671	0.756	73	0.228	0.509
vest	50	172	0.560	0.696	1	0.000	0.472
shorts	20	16	0.350	0.688	1	0.200	0.532
trousers	129	24	0.155	0.692	484	0.946	0.547
skirt	40	19	0.050	0.726	4	0.075	0.581
short sleeve dress	20	3	0.000	0.729	1	0.000	0.58
long sleeve dress	10	0	0.000	0.756	0	0.000	0.38
vest dress	10	0	0.000	0.626	0	0.000	0.489

363 images for validation, and 618 images for the test set. The train, validation, and test datasets contain roughly ~75%, ~10%, ~15% of the images from each class, respectively. Only 10 of the 13 classes as described in Table 1 had enough unique items ($n >= 3$) to create a train/val/test split. We also made use of the DeepFashion 2 [4] dataset. As training on the full-sized DeepFashion 2 dataset was not feasible with the computation resources available to us only the first 10% of the training and validation set was used. This subset of DeepFashion2 was verified to have a minimum of 66 samples for each class. Similarly to RobotFashion the distribution of classes is skewed with an over representation of trousers and short-sleeve-tops.

Training solely on RobotFashion In the first experiment the Faster R-CNN model is used to train on RobotFashion. We use all the default values of the PyTorch implementation of the model. As optimizer Adam [6] was used with default with default *learning rate* $= 0.001$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. No learning rate decay was used. We trained for 40 epochs (~11 hours on a single NVIDIA K80 GPU). We evaluated the weights at the last epoch[1].

Training solely on DeepFashion2 In the second experiment the Faster R-CNN model is used is trained on the subset of DeepFashion2. Simarly to training on RobotFashion, all default values for the model as well as Adam were used. We also did not use learning rate decay. We trained for 15 epochs (~18 hours on a single NVIDIA T4 GPU). We evaluated the weights with the lowest validation loss.

5 Results

We evaluated both model on the test set of RobotFashion. The TensorBoard graphs of training can be found at `https://tensorboard.dev/experiment/UvOJo3TOSY6X2JIzOVgDKQ/`. Table 2 shows the classification accuracy and mean intersection over union of the bounding boxes. Over all the classes the model trained on RobotFashion has a worse accuracy but a better mIoU (mean intersection over union). The model trained on DeepFashion2 has a very high accuracy on trousers, while the model trained on RobotFashion has a high accuracy on long sleeve outwear. In the prediction columns the distribution of predictions over all the classes is shown. We can see that the DeepFashion model overwhelmingly predicts the trouser class. The RobotFashion model tends to predict long sleeve outwear, but seems to pick other classess more equally than the DeepFashion2 model. Figure 6 shows the predicted bounding boxes of both the trouser and the long sleeve outwear class.

[1] One of the annotations had a second, 0 area bounding box ($x_{min} = x_{max}$). Therefore all our validation losses were NaN and couldn't be used to select the model with lowest validation loss.

(a)

(b)

Figure 6: The bounding boxes generated by (a) the model trained on RobotFashion and (b) the model trained on DeepFashion. Left is a picture of the trousers class and right a picture on the long-sleeve-outwear class. The red bounding box is the ground truth, while the green bounding box is predicted by the models.

Figure 7: The test set predictions of the model (a) trained on DeepFashion2 (b) re-trained on RobotFashion

Table 3: Evaluation of the trained models on the test set of RobotFashion. Instead of grouping by class as in Table 2 the accuracies are grouped by the states of clothing as described in Section 2.1.

Orientation		Trained on RobotFashion		Trained on DeepFashion2	
description	samples	accuracy	mIoU	accuracy	mIoU
1	110	0.191	0.691	0.574	0.484
2	110	0.200	0.718	0.218	0.459
3	110	0.200	0.758	0.309	0.548
4	110	0.182	0.722	0.227	0.495
5	110	0.200	0.741	0.173	0.466

6 Discussions

The noisy nature of data and the robot presence and movements in this dataset, as well as the annotations, could facilitate developments of such datasets in future works. By works that are done, as well as the lessons learned, it would be easier to improve such works in the future.

There are two categories that this project could be potentially improved for future works. One of them is to make use of the third dimension that was available in the captured samples but was unused in the training algorithm. Second, we could try to denoise the background of the captured images by trying to apply image filters in the initial steps of training, and gradually lift them as the training progresses, to see if the deep learning algorithm has learned the desired features better comparably

7 Conclusions

In this project, we have proposed a new dataset for clothes being held and deformed by robots. This dataset consists of 3567 images, taken from frame samples of videos captured of a robot moving and deforming clothes. Each image is labeled with category and bounding box by a human annotator. We then established different benchmarks in order to measure improvements achieved by retraining on this dataset using either COCO 2017 or Deepfashion2 as a base trained model.

Finally, we were able to achieve marginal improvements after the retraining when we applied tests using a separate dataset. We believe this project has the potential for future works like using the third dimension of captured images and using mask filters.

Acknowledgments

The authors would like to thank *Google* for providing a Google Cloud Platform educational grant to the KTH course *DD2430 Project Course in Data Science* under which the authors conducted the research described above. They would also like to thank *H&M* for providing the clothes which were used to create the dataset.

Reflection on discussion with group 18

We discussed our project with group 18. Their project involved getting insights into NLP models. Similarly to our project, this required them to collect their own dataset. Their project required them to create sets of texts embedded with different context and knowledge by scraping Wikipedia. We drew a parallel with our project as automating the dataset creation as much as possible is required to get enough data to learn. Therefore we suggested spending time on verifying their dataset creation process and even do experiments to see what the best approach would be. Their suggestion to us was to finish our pipeline from data to prediction for at least one class in order to get results more quickly. We ended up not using this advice as a few days later we had recorded all the clothing pieces in our possession and we thought it was reasonable for everyone in our project group to focus on annotating so that is was finished. It, however, took longer than expected to annotate the data as we didn't consider the amount of time it would take to clean the data (deal with small inconsistencies, differences in annotations, etc).

References

[1] Tae-Kyun Kim Andreas Doumanoglou Andreas Kargakos and Sotiris Malassiotis. "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning". In: *IEEE International Conference on Robotics & Automation*. 2014.

[2] Gerardo Aragon-Camarasa et al. "Glasgow's stereo image database of garments". In: *arXiv preprint arXiv:1311.7295* (2013).

[3] *CloPeMa - Clothes Perception and Manipulation*. URL: `http://www.clopema.eu/` (visited on 12/06/2019).

[4] Yuying Ge et al. "A Versatile Benchmark for Detection, Pose Estimation, Segmentation and Re-Identification of Clothing Images". In: *CVPR* (2019).

[5] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: `1512.03385 [cs.CV]`.

[6] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: `1412.6980 [cs.LG]`.

[7] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2016. arXiv: `1612.03144 [cs.CV]`.

[8] Tsung-Yi Lin et al. *Microsoft COCO: Common Objects in Context*. 2014. arXiv: `1405.0312 [cs.CV]`.

[9] Ziwei Liu et al. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1096–1104.

[10] Adam Paszke et al. "Automatic Differentiation in PyTorch". In: *NeurIPS Autodiff Workshop*. 2017.

[11] Shaoqing Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. arXiv: `1506.01497 [cs.CV]`.

[12] *Tzutalin. LabelImg. Git code (2015).* URL: `https://github.com/tzutalin/labelImg`.

[13] L. Wagner and D. Krejčová. "CTU color and depth image dataset of spread garments." In: *Research Report CTU-CMP-2013-25, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic*. 2013.

[14] Shuai Zheng et al. "ModaNet: A Large-Scale Street Fashion Dataset with Polygon Annotations". In: *ACM Multimedia*. 2018.