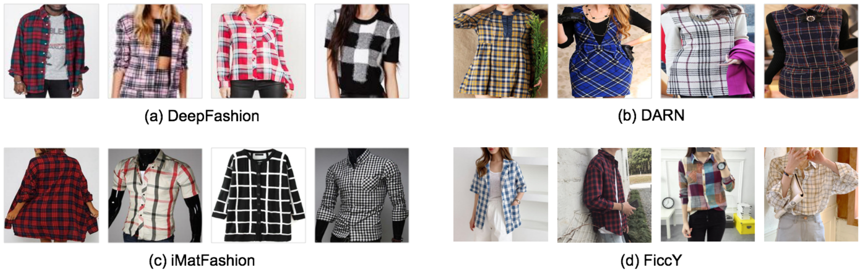
A Dataset overview



(a) DeepFashion

(b) DARN

(c) iMatFashion

(d) FiccY

Fig. 1. Samples of ground-truth ”check” in each benchmark datasets (DeepFashion, DARN, iMatFashion and FiccY). The fact the quality of the image (*e.g. view point, background noise etc.*) significantly differs according to the dataset highlights the importance of domain adaptability.

B Effect on training the student varying the amount of unlabeled data

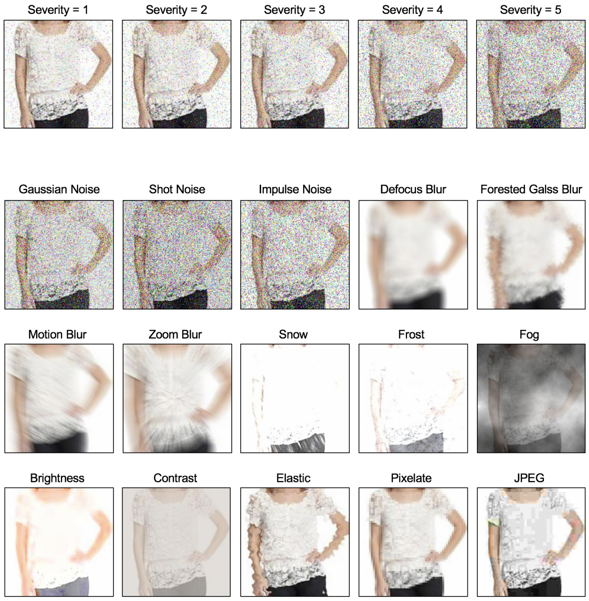
Table 1. The comparison of mean AP (mAP) between the teacher (T) and the student (S) varying the amount of unlabeled images used for training.

mAP	T	S (Multi-task learning with 4 teachers)					
Train Size	-	1K	5K	10K	50K	100K	500K
Category	**85.50**	45.38	64.08	73.76	84.43	**86.04**	**87.18**
Pattern	**72.54**	13.11	41.46	57.80	70.72	**72.9**	**73.72**
Color	**58.58**	14.54	50.60	54.12	57.76	58.48	**59.33**
Texture	**62.30**	40.90	50.61	56.57	**62.97**	**63.75**	**64.20**

C Evaluation Metrics

The proposed model was mainly evaluated using the two most frequently used metrics: recall and F1 score. Recall is also referred to as sensitivity and it measures the probability of a positive detection. Precision, which is frequently used in combination with recall, is the percent of all relevant results among returned predictions. The F1 score is the harmonic mean of prediction and recall. The F1 score can fluctuate depending on which confidence score is used for a class to be considered as a final prediction. To minimize misleading effect by thresholding strategy, predictions were sorted by score and the top-k classes were selected as final predictions, meaning the number of predictions for an image was always k.

D Example of corruptions in DeepFashion



E Implementation details

General. A ResNet50 was used as a backbone for both the teacher and student networks. The number of output dimensions for image embedding was set to 1024. The images for each class were randomly sampled to collect 300K images for each epoch of training. Images were re-sampled for every epoch. A stochastic gradient descent optimizer with default parameters provided by PyTorch was adopted. The classes were sorted by the number of images and only images included in top-40 classes were used for training, while evaluation was performed using all classes. Our implementation is based on PyTorch and the proposed model was converted into the Open Neural Network Exchange for deployment for real production.

Teacher stage. The initial learning rate lr was set to 0.2 with a batch size of 128. lr was set to decay by a factor of 0.5 at every 10 epochs over 40 total epochs. γ was set to 1.0 by default. The images with no ground-truth label for the target attribute type were not used for training the teacher to avoid the unnecessary computational overhead.

Student stage. lr was gradually increased from 0 to 0.4 at the beginning until 1M images were processed. lr was set to decay by a factor of 0.5 at every 10 epochs over 100 total epochs. β was set to 1.0 by default.