Paper Doll Parsing: Retrieving Similar Styles to Parse Clothing Items

Kota Yamaguchi
Stony Brook University
Stony Brook, NY, USA
kyamagu@cs.stonybrook.edu

M. Hadi Kiapour
UNC at Chapel Hill
Chapel Hill, NC, USA
hadi.kiapour@gmail.com

Tamara L. Berg
UNC at Chapel Hill
Chapel Hill, NC, USA
berg.tamara@gmail.com

Abstract

Clothing recognition is an extremely challenging problem due to wide variation in clothing item appearance, layering, and style. In this paper, we tackle the clothing parsing problem using a retrieval based approach. For a query image, we find similar styles from a large database of tagged fashion images and use these examples to parse the query. Our approach combines parsing from: pre-trained global clothing models, local clothing models learned on the fly from retrieved examples, and transferred parse masks (paper doll item transfer) from retrieved examples. Experimental evaluation shows that our approach significantly outperforms state of the art in parsing accuracy.

1. Introduction

Clothing choices vary widely across the global population. For example, one person's style may lean toward preppy while another's trends toward goth. However, there are commonalities. For instance, walking through a college campus you might notice student after student consistently wearing combinations of jeans, t-shirts, sweatshirts, and sneakers. Or, you might observe those who have just stumbled out of bed and are wandering to class looking disheveled in their pajamas. Even hipsters who purport to be independent in their thinking and dress, tend to wear similar outfits consisting of variations on tight-fitting jeans, button down shirts, and thick plastic glasses. In some cases, style choices can be a strong cue for visual recognition.

In addition to style variation, individual clothing items also display many different appearance characteristics. As a concrete example, shirts have an incredibly wide range of appearances based on cut, color, material, and pattern. This can make identifying part of an outfit as a shirt very challenging. Luckily, for any particular choice of these parameters, *e.g.*, blue and white checked button down, there are many shirts with similar appearance. It is this visual similarity and the existence of some consistency in style choices discussed above that we exploit in our system.

In this paper, we take a data driven approach to clothing parsing. We first collect a large, complex, real world collection of outfit pictures from a social network focused on fashion, `chictopia.com`. Using a very small set of hand parsed images in combination with the text tags associated with each image in the collection, we can parse our large database accurately. Now, given a query image without any associated text, we can predict an accurate parse by retrieving similar outfits from our parsed collection, building local models from retrieved clothing items, and transferring inferred clothing items from the retrieved samples to the query image. Final iterative smoothing produces our end result. In each of these steps we take advantage of the relationship between clothing and body pose to constrain prediction and produce a more accurate parse. We call this *paper doll parsing* because it essentially transfers predictions from retrieved samples to the query, like laying paper cutouts of clothing items onto a paper doll. Consistencies in dressing make this retrieval based effort possible.

In particular, we propose a retrieval based approach to clothing parsing that combines:

- Pre-trained global models of clothing items.
- Local models of clothing items learned on the fly from retrieved examples.
- Parse mask predictions transferred from retrieved examples to the query image.
- Iterative label smoothing.

Clothing recognition is a challenging and societally important problem – global sales for clothing total over a hundred billion dollars, much of which is conducted online. This is reflected in the growing interest in clothing related recognition papers [11, 10, 25, 16, 26, 7, 2, 4], perhaps boosted by recent advances in pose estimation [27, 3]. Many of these papers have focused on specific aspects of clothing recognition such as predicting attributes of clothing [7, 2, 4], outfit recommendation [15], or identifying aspects of socio-identity through clothing [18, 20].

We attack the problem of clothing parsing, assigning a semantic label to each pixel in the image where labels can

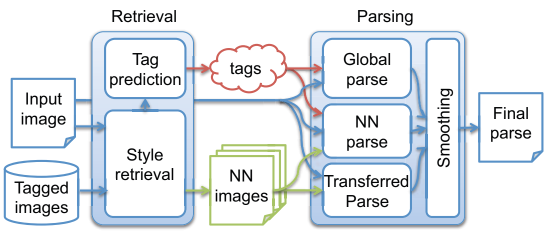

Figure 1: Parsing pipeline.

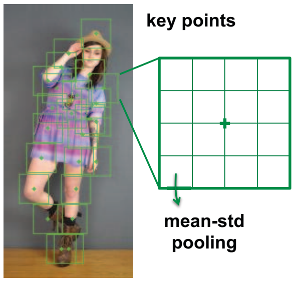

Figure 2: Spatial descriptors for style representation.

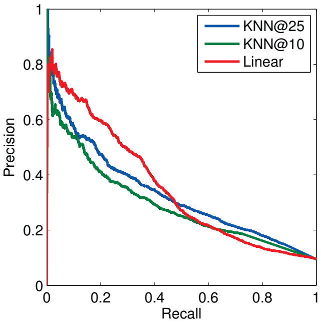

Figure 3: Tag prediction PR-plot.

be selected from background, skin, hair, or from a large set of clothing items (e.g. boots, tights, sweater). Effective solutions to clothing parsing could enable useful end-user applications such as pose independent clothing retrieval [26] or street to shop applications [16]. This problem is closely related to the general image parsing problem which has been approached successfully using related non-parametric methods [21, 14, 22]. However, we view the clothing parsing problem as suitable for specialized exploration because it deals with people, a category that has obvious significance. The clothing parsing problem is also special in that one can take advantage of body pose estimates during parsing, and we do so in all parts of our method.

Previous state of the art on clothing parsing [26] performed quite well on the constrained parsing problem, where test images are parsed given user provided tags indicating depicted clothing items. However, they were less effective at unconstrained clothing parsing, where test images are parsed in the absense of any textual information. We provide an approach to unconstrained clothing parsing that performs much better than previous state of the art, boosting overall image labeling performance from *77%* to *84%* and performance of labeling foreground pixels (those actually on the body) from *23%* to *40%*, an increase of 74% of the previous accuracy.

2. Dataset

This paper uses the Fashionista dataset provided in [26] and an expansion called the Paper Doll dataset which we collected for this paper. The Fashionista dataset provides 685 fully parsed images that we use for supervised training and performance evaluation, 456 for training and 229 for testing. The training samples are used for learning feature transforms, building global clothing models, and adjusting parameters. The testing samples are reserved for evaluation.

The Paper Doll dataset is a large collection of tagged fashion pictures. We collected over 1 million pictures from `chictopia.com` with associated metadata tags denoting characteristics such as color, clothing item, or occasion. Since the Fashionista dataset also uses Chictopia, we automatically exclude any duplicate pictures from the Paper Doll dataset. From the remaining, we select pictures tagged with at least one clothing item and run a full-body pose detector [27], keeping those that have a person detection. This results in 339,797 pictures weakly annotated with clothing items and estimated pose. Though the annotations are not always complete – users often do not label all depicted items, especially small items or accessories – it is rare to find images where an annotated tag is not present. We use the Paper Doll dataset for style retrieval.

3. Approach overview

For a query image, our approach consists of two steps:

1. Retrieve similar images from the parsed database.
2. Use retrieved images and tags to parse the query.

Figure 1 depicts the overall parsing pipeline.

3.1. Low-level features

We first run a pose estimator [27] and normalize the full-body bounding box to fixed size. The pose estimator is trained using the Fashionista training split and negative samples from the INRIA dataset. During parsing, we compute the parse in this fixed frame size then warp it back to the original image, assuming regions outside the bounding box are background.

Our methods draw from a number of dense feature types (each parsing method uses some subset):

RGB RGB color of the pixel.

Lab L*a*b* color of the pixel.

MR8 Maximum Response Filters [23].

Gradients Image gradients at the pixel.

HOG HOG descriptor at the pixel.

Boundary Distance Negative log-distance from the boundary of an image.

Pose Distance Negative log-distance from 14 body joints and any body limbs.

Whenever we use a statistical model built upon these features, we first normalize features by subtracting their mean and dividing by 3 standard deviations for each dimension. Also, when we use logistic regression [8], we use these normalized features and their squares, along with a constant bias. So, for an N-dimensional feature vector, we always learn $2N + 1$ parameters.

4. Style retrieval

Our goal for retrieving similar pictures is two-fold: a) to predict depicted clothing items, and b) to obtain information helpful for parsing clothing items.

4.1. Style descriptor

We design a descriptor for style retrieval that is useful for finding styles with similar appearance. For an image, we obtain a set of 24 key points interpolated from the 27 pose estimated body joints. These key points are used to extract part-specific spatial descriptors - a mean-std pooling of normalized dense features in 4-by-4 cells in a 32-by-32 patch around the key point. That is, for each cell in the patch, we compute mean and standard deviation of the normalized features (Figure 2 illustrates). The features included in this descriptor are RGB, Lab, MR8, HOG, Boundary Distance, and Skin-hair Detection.

Skin-hair Detection is computed using logistic regression for *skin*, *hair*, *background*, and *clothing* at each pixel. For its input, we use RGB, Lab, MR8, HOG, Boundary Distance, and Pose Distance. Note that we do not include Pose Distance as a feature in the style descriptor, but instead use Skin-hair detection to indirectly include pose-dependent information in the representation since the purpose of the style descriptor is to find similar styles independent of pose.

For each key point, we compute the above spatial descriptors and concatenate to describe the overall style, resulting in a 39,168 dimensional vector for an image. For efficiency of retrieval, we use PCA for dimensionality reduction to a 441 dimensional representation. We use the Fashionista training split to build the Skin-hair detector and also to train the PCA model.

4.2. Retrieval

We use L2-distance over the style descriptors to find the K nearest neighbors (KNN) in the Paper Doll dataset. For efficiency, we build a KD-tree [24] to index samples. In this paper, we fix $K = 25$ for all the experiments. Figure 4 shows two examples of nearest neighbor retrievals.

4.3. Tag prediction

The retrieved samples are first used to predict clothing items potentially present in a query image. The purpose of tag prediction is to obtain a set of tags that might be relevant to the query, while eliminating definitely irrelevant items for consideration. Later stages can remove spuriously predicted tags, but tags removed at this stage can never be predicted. Therefore, we wish to obtain the best possible predictive performance in the high recall regime.

Tag prediction is based on a simple voting approach from KNN. Each tag in the retrieved samples provides a vote weighted by the inverse of its distance from the query, which forms a confidence for presence of that item. We threshold this confidence to predict the presence of an item.

We experimentally selected this simple KNN prediction instead of other models, because it turns out KNN works well for the high-recall prediction task. Figure 3 shows performance of linear vs KNN at 10 and 25. While linear classification (clothing item classifiers trained on subsets of body parts, e.g. *pants* on lower body keypoints), works well in the low-recall high precision regime, KNN outperforms in the high-recall range. KNN at 25 also outperforms 10.

Since the goal here is only to eliminate obviously irrelevant items while keeping most potentially relevant items, we tune the threshold to give 0.5 recall in the Fashionista training split. Due to the skewed item distribution in the Fashionista dataset, we use the same threshold for all items to avoid over-fitting the predictive model. In the parsing stage, we always include *background*, *skin*, and *hair* in addition to the predicted clothing tags.

5. Clothing parsing

Following tag prediction, we start to parse the image in a per-pixel fashion. Parsing has two major phases:

1. Compute pixel-level confidence from three methods: global parse, nearest neighbor parse, and transferred parse.
2. Apply iterative label smoothing to get a final parse.

Figure 5 illustrates outputs from each parsing stage.

5.1. Pixel confidence

Let us denote y_i as the clothing item label at pixel i. The first step in parsing is to compute a confidence score of assigning clothing item l to y_i. We model this scoring function S as the mixture of three confidence functions.

$$\begin{aligned} S(y_i|\mathbf{x}_i, D) \quad \equiv \quad & S_{\text{global}}(y_i|\mathbf{x}_i, D)^{\lambda_1} \cdot \\ & S_{\text{nearest}}(y_i|\mathbf{x}_i, D)^{\lambda_2} \cdot \\ & S_{\text{transfer}}(y_i|\mathbf{x}_i, D)^{\lambda_3}, \end{aligned} \tag{1}$$

where $\mathbf{x}_i$ denotes pixel features, $\Lambda \equiv [\lambda_1, \lambda_2, \lambda_3]$ are mixing parameters, and D is a set of nearest neighbor samples.

5.1.1 Global parse

The first term in our model is a global clothing likelihood, trained for each clothing item on the hand parsed Fashionista training split. This is modeled as a logistic regression

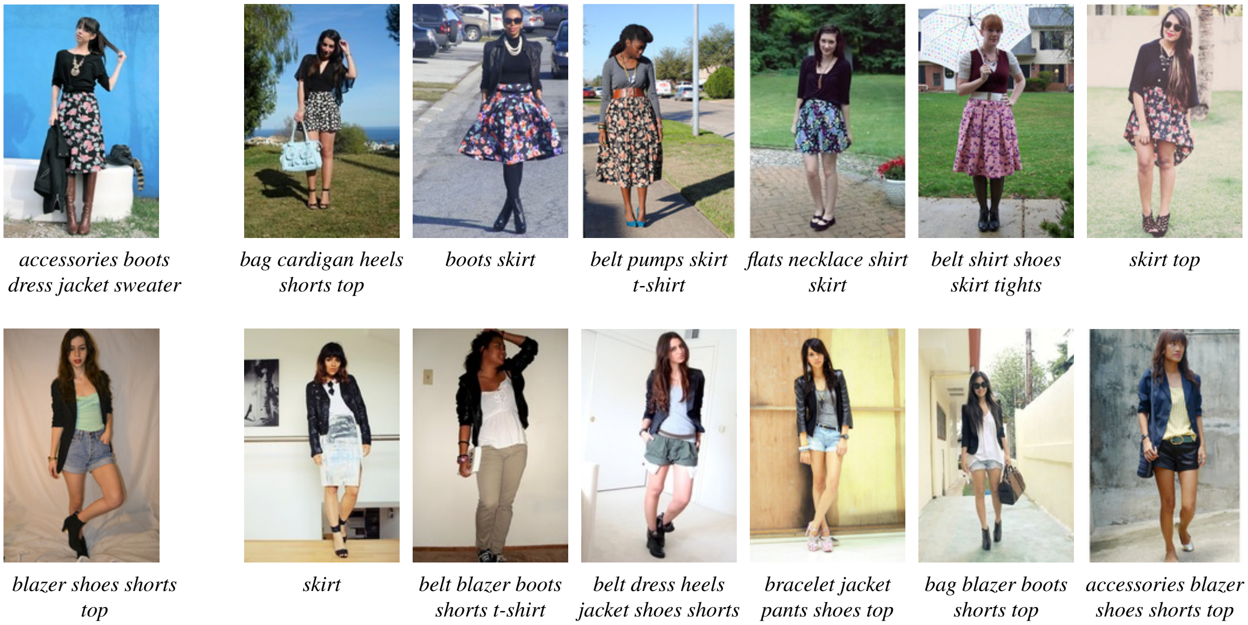
accessories boots dress jacket sweater
bag cardigan heels shorts top
boots skirt
belt pumps skirt t-shirt
flats necklace shirt skirt
belt shirt shoes skirt tights
skirt top
blazer shoes shorts top
skirt
belt blazer boots shorts t-shirt
belt dress heels jacket shoes shorts
bracelet jacket pants shoes top
bag blazer boots shorts top
accessories blazer shoes shorts top

Figure 4: Retrieval examples. The leftmost column shows query images with ground truth item annotation. The rest are retrieved images with associated tags in the top 25. Notice retrieved samples sometimes have missing item tags.

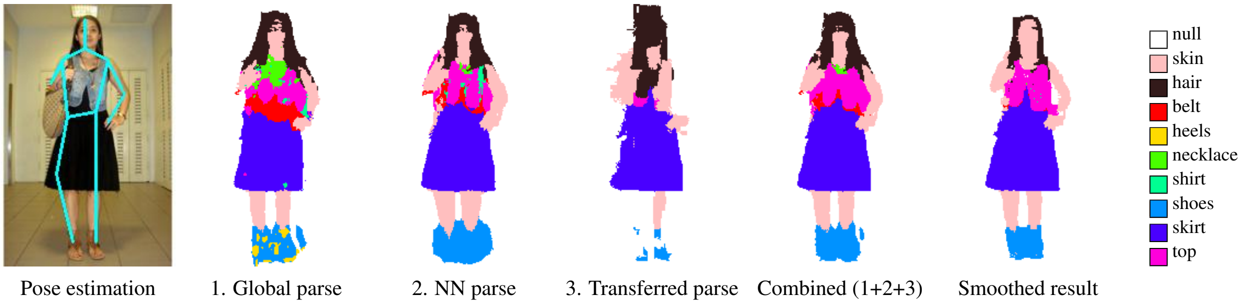

Figure 5: Parsing outputs at each step. Labels are MAP assignments of the scoring functions.

that computes the likelihood of a label assignment to each pixel for a given set of possible clothing items:

$$S_{\text{global}}(y_i|\mathbf{x}_i, D) \equiv P(y_i = l|\mathbf{x}_i, \theta_l^g) \cdot \mathbf{1}[l \in \tau(D)], \quad (2)$$

where P is logistic regression given feature $\mathbf{x}_i$ and model parameter θ_l^g, $\mathbf{1}[\cdot]$ is an indicator function, and $\tau(D)$ is a set of predicted tags from nearest neighbor retrieval. We use RGB, Lab, MR8, HOG, and Pose Distances as features. Any unpredicted items receive zero probability.

The model parameter θ_l^g is trained on the Fashionista training split. For training each θ_l^g, we select negative pixel samples only from those images having at least one positive pixel. That is, the model gives localization probability given that a label l is present in the picture. This could potentially increase confusion between similar item types, such as *blazer* and *jacket* since they usually do not appear together, in favor of better localization accuracy. We chose to rely on the tag prediction τ to resolve such confusion.

Because of the tremendous number of pixels in the dataset, we subsample pixels to train each of the logistic regression models. During subsampling, we try to sample pixels so that the resulting label distribution is close to uniform in each image, preventing learned models from only predicting large items.

5.1.2 Nearest neighbor parse

The second term in our model is also a logistic regression, but trained only on the retrieved nearest neighbor (NN) images. Here we learn a local appearance model for each clothing item based on examples that are similar to the query, e.g. *blazers* that look similar to the query blazer because they were retrieved via style similarity. These local models are much better models for the query image than those trained globally (because *blazers* in general can take on a huge range of appearances).

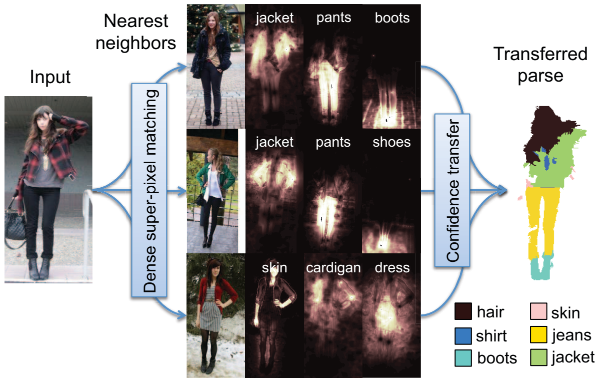

Figure 6: Transferred parse. Likelihoods in nearest neighbors are transferred to the input via dense matching.

$$S_{\text{nearest}}(y_i|\mathbf{x}_i, D) \equiv P(y_i = l|\mathbf{x}_i, \theta_l^n) \cdot \mathbf{1}[l \in \tau(D)]. \quad (3)$$

The model parameter θ_l^n is locally learned from the retrieved samples D, using RGB, Lab, Gradient, MR8, Boundary Distance, and Pose Distance.

In this step, predicted pixel-level annotations from the retrieved samples are used (computed during pre-processing detailed in Section 5.3) to learn local appearance models. NN models are trained using any pixel (with subsampling) in the retrieved samples in a one-vs-all fashion.

5.1.3 Transferred parse

The third term in our model is obtained by transferring the parse mask likelihoods estimated by the global parse S_{global} from the retrieved images to the query image (Figure 6 visualizes an example). This approach is similar in spirit to approaches for general segmentation that transfer likelihoods using over segmentation and matching [1, 13, 17], but here because we are performing segmentation on people we can take advantage of pose estimates during transfer.

In our approach, we find dense correspondence based on super-pixels instead of pixels (e.g., [21]) to overcome the difficulty in naively transferring deformable, often occluded clothing items pixel-wise. Our approach first computes an over-segmentation of both query and retrieved images using a fast and simple segmentation algorithm [9], then finds corresponding pairs of super-pixels between the query and each retrieved image based on pose and appearance:

1. For each super-pixel in the query, find the 5 nearest super-pixels in each retrieved image using L2 Pose Distance.
2. Compute a concatenation of bag-of-words from RGB, Lab, MR8, and Gradient for each of those super-pixels.
3. Pick the closest super-pixel from each retrieved image using L2 distance on the bag-of-words feature.

Let us denote the super-pixel of pixel i with s_i, the selected corresponding super-pixel from image r with $s_{i,r}$, and the bag-of-words features of super-pixel s with $h(s)$. Then, our transferred parse is computed as:

$$S_{\text{transfer}}(y_i|\mathbf{x}_i, D) \equiv \frac{1}{Z} \sum_{r \in D} \frac{M(y_i, s_{i,r})}{1 + \|h(s_i) - h(s_{i,r})\|}, \quad (4)$$

where we define:

$$M(y_i, s_{i,r}) \equiv \frac{1}{|s_{i,r}|} \sum_{j \in s_{i,r}} P(y_i = l|\mathbf{x}_i, \theta_l^g) \cdot \mathbf{1}[l \in \tau(r)], \quad (5)$$

which is a mean of the global parse over the super-pixel in a retrieved image. Here we denote a set of tags of image r with $\tau(r)$, and normalization constant Z.

5.1.4 Combined confidence

After computing our three confidence scores, we combine them with parameter Λ to get the final pixel confidence S as described in Equation 1. We choose the best mixing parameter such that MAP assignment of pixel labels gives the best foreground accuracy in the Fashionista training split by solving the following optimization (on foreground pixels F):

$$\max_{\Lambda} \sum_{i \in F} \mathbf{1}\left[\tilde{y}_i = \arg\max_{y_i} S_\Lambda(y_i|\mathbf{x}_i)\right], \quad (6)$$

where $\tilde{y}_i$ is the ground truth annotation of the pixel i. The nearest neighbors D in S are dropped in the notation for simplicity. We use a simplex search algorithm to solve for the optimum parameter starting from uniform values. In our experiment, we obtained $(0.41, 0.18, 0.39)$.

We exclude background pixels from this optimization because of the skew in the label distribution – background pixels in Fashionista dataset represent 77% of total pixels, which tends to direct the optimizer to find meaningless local optima; i.e., predicting everything as *background*.

5.2. Iterative label smoothing

The combined confidence gives a rough estimate of item localization. However, it does not respect boundaries of actual clothing items since it is computed per-pixel. Therefore, we introduce an iterative smoothing stage that considers all pixels together to provide a smooth parse of an image. Following the approach of [19], we formulate this smoothing problem by considering the joint labeling of pixels $Y \equiv \{y_i\}$ and item appearance models $\Theta \equiv \{\theta_l^s\}$, where θ_l^s is a model for a label l. The goal is to find the optimal joint assignment Y^* and item models Θ^* for a given image.

We start this problem by initializing the current predicted parsing $\hat{Y}_0$ with the MAP assignment under the combined confidence S. Then, we treat $\hat{Y}_0$ as training data to build initial image-specific item models $\hat{\Theta}_0$ (logistic regressions). For these models, we only use RGB, Lab, and Boundary Distance since otherwise models easily over-fit. Also, we use a higher regularization parameter for training instead of finding the best cross-validation parameter, assuming the initial training labels $\hat{Y}_0$ are noisy.

After obtaining $\hat{Y}_0$ and $\hat{\Theta}_0$, we solve for the optimal assignment $\hat{Y}_t$ at the current step t with the following optimization:

$$\hat{Y}_t \in \arg\max_{Y} \prod_{i} \Phi(y_i|\mathbf{x}_i, S, \hat{\Theta}_t) \prod_{i,j \in V} \Psi(y_i, y_j|\mathbf{x}_i, \mathbf{x}_j), \quad (7)$$

where we define:

$$\begin{aligned} \Phi(y_i|\mathbf{x}_i, S, \hat{\Theta}_t) &\equiv \quad S(y_i|\mathbf{x}_i)^{\lambda} \cdot P(y_i|\mathbf{x}_i, \theta_l^s)^{1-\lambda}, & (8) \\ \Psi(y_i, y_j|\mathbf{x}_i, \mathbf{x}_j) &\equiv \quad \exp\{\gamma e^{-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2} \cdot \mathbf{1}\left[y_i \neq y_j\right]\}. & (9) \end{aligned}$$

Here, V is a set of neighboring pixel pairs, λ, β, γ are the parameters of the model, which we experimentally determined in this paper. We use the graph-cut algorithm [6, 5, 12] to find the optimal solution.

With the updated estimate of the labels $\hat{Y}_t$, we train the logistic regressions $\hat{\Theta}_t$ and repeat until the algorithm converges. Note that this iterative approach is not guaranteed to converge. We terminate the iteration when 10 iterations pass, when the number of changes in label assignment is less than 100, or the ratio of the change is smaller than 5%.

5.3. Offline processing

Our retrieval techniques require the large Paper Doll Dataset to be pre-processed (parsed), for building nearest neighbor models on the fly from retrieved samples and for transferring parse masks. Therefore, we estimate a clothing parse for each sample in the 339K image dataset, making use of pose estimates and the tags associated with the image by the photo owner. This parse makes use of the global clothing models (constrained to the tags associated with the image by the photo owner) and iterative smoothing parts of our approach.

Although these training images are tagged, there are often clothing items missing in the annotation. This will lead iterative smoothing to mark foreground regions as *background*. To prevent this, we add an *unknown* item label with uniform probability and initialize $\hat{Y}_0$ together with the global clothing model at all samples. This effectively prevents the final estimated labeling $\hat{Y}$ to mark missing items with incorrect labels.

Offline processing of the Paper Doll Dataset took a few of days with our Matlab implementation in a distributed environment. For an unseen query image, our full parsing pipeline takes 20 to 40 seconds, including pose estimation. The major computational bottlenecks are in pose estimation and iterative smoothing.

6. Experimental results

We evaluate parsing performance on the 229 testing samples from the Fashionista dataset. The task is to predict a label for every pixel where labels represent a set of 56 different categories – a very large and challenging variety of clothing items.

Performance is measured in terms of standard metrics: accuracy, average precision, average recall, and average F-1 over pixels. In addition, we also include foreground accuracy (See eqn 6) as a measure of how accurately each method is at parsing foreground regions (those pixels on the body, not on the background). Note that the average measures are over non-empty labels after calculating pixel-based performance for each since some labels are not present in the test set. Since there are some empty predictions, F-1 does not necessarily match the geometric mean of average precision and recall.

Table 1 summarizes predictive performance of our parsing method, including a breakdown of how well the intermediate parsing steps perform. For comparison, we include the performance of previous state of the art on clothing parsing [26]. Our approach outperforms the previous method in overall accuracy (**84.68**% vs **77.45**%). It also provides a huge boost in foreground accuracy. The previous approach provides **23.11**% foreground accuracy, while we obtain **40.20**%. We also obtain much higher precision (**10.53**% vs **33.34**%) without much decrease in recall (**17.2**% vs **15.35**%).

Figure 7 shows examples from our parsing method, with ground truth annotation and the method of [26]. We observe that our method produces a parse that respects the actual item boundary, even if some items are incorrectly labeled; e.g., predicting *pants* as *jeans*, or *jacket* as *blazer*. However, often these confusions are due to high similarity in appearance between items and sometimes due to non-exclusivity in item types, i.e., *jeans* are a type of *pants*.

Figure 8 plots F-1 scores for non-empty items (items predicted on the test set) comparing the method of [26] with our method. Our model outperforms the prior work on many items, especially major foreground items such as *dress*, *jeans*, *coat*, *shorts*, or *skirt*. This results in a significant boost in foreground accuracy and perceptually better parsing results.

Though our method is successful at foreground prediction overall, there are a few drawbacks to our approach. By design, our style descriptor is aimed at representing whole outfit style rather than specific details of the outfit. Consequently, small items like accessories tend to be less weighted during retrieval and are therefore poorly predicted during parsing. However, prediction of small items is inherently extremely challenging because they provide limited appearance information.

Another issue for future work is the prevention of con-

Method	Accuracy	F.g. accuracy	Avg. precision	Avg. recall	Avg. F-1
CRF [26]	77.45	23.11	10.53	**17.20**	10.35
1. Global parse	79.63	35.88	18.59	15.18	12.98
2. NN parse	80.73	38.18	21.45	14.73	12.84
3. Transferred parse	83.06	33.20	31.47	12.24	11.85
4. Combined (1+2+3)	83.01	39.55	25.84	15.53	14.22
5. Our final parse	**84.68**	**40.20**	**33.34**	15.35	**14.87**

Table 1: Parsing performance for final and intermediate results (MAP assignments at each step).

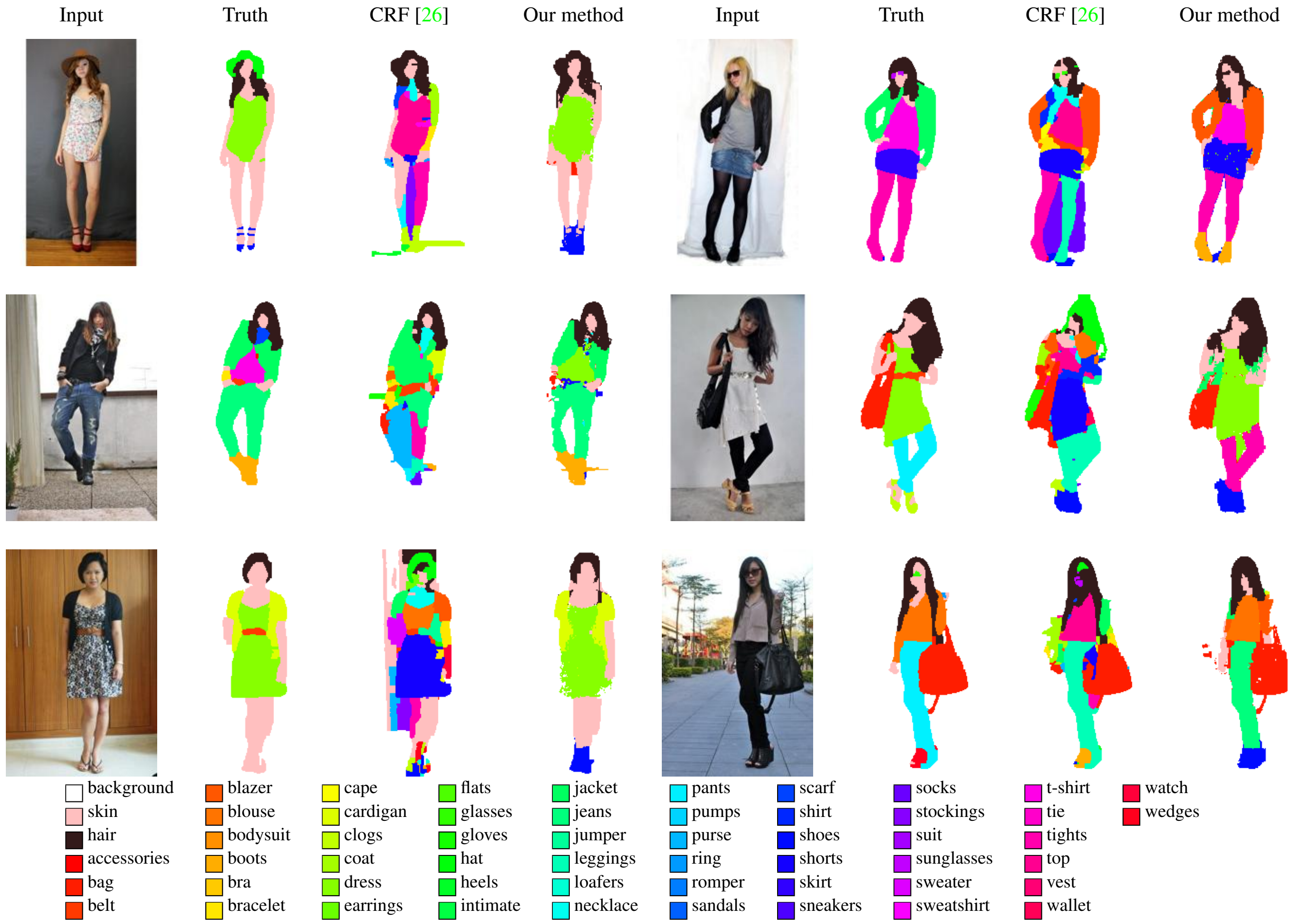

Figure 7: Parsing examples. Our method sometimes confuses similar items, but gives overall perceptually better results.

flicting items from being predicted for the same image, such as *dress* and *skirt*, or *boots* and *shoes* which tend not to be worn together. Our iterative smoothing is effectively reducing such confusion, but the parsing result sometimes contains one item split into two conflicting items. One way to resolve this would be to enforce constraints on the overall combination of predicted items, but this leads to a difficult optimization problem and we leave it as future work.

Lastly, we find it difficult to predict items with skin-like color or coarsely textured items (similar to issues reported in [26]). Because of the variation in lighting condition in pictures, it is very hard to distinguish between actual skin and clothing items that look like skin, e.g. slim khaki pants. Also, it is very challenging to differentiate for example between bold stripes and a belt using low-level image features. These cases will require higher-level knowledge about outfits to correctly parse.

7. Conclusion

We describe a clothing parsing method based on nearest neighbor style retrieval. Our system combines: global parse models, nearest neighbor parse models, and transferred parse predictions. Experimental evaluation shows successful results, demonstrating a significant boost of over-

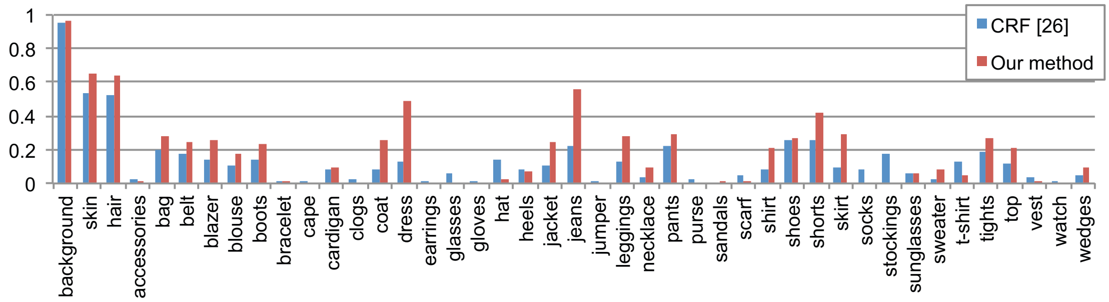

Figure 8: F-1 score of non-empty items. We observe significant performance gains, especially for large items.

all accuracy and especially foreground parsing accuracy over previous work. It is our future work to resolve the confusion between very similar items and to incorporate higher level knowledge about outfits.

Acknowledgements: Research supported by Google Faculty Award, "Seeing Social: Exploiting Computer Vision in Online Communities" & NSF Career Award #1054133.

References

[1] E. Borenstein and J. Malik. Shape guided object segmentation. In *CVPR*, volume 1, pages 969–976, 2006. 5

[2] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool. Apparel classification with style. *ACCV*, pages 1–14, 2012. 1

[3] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 1

[4] L. Bourdev, S. Maji, and J. Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, pages 1543–1550, 2011. 1

[5] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004. 6

[6] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001. 6

[7] H. Chen, A. Gallagher, and B. Girod. Describing clothing by semantic attributes. In *ECCV*, pages 609–623. 2012. 1

[8] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal. Machine Learning Research*, 9:1871–1874, 2008. 3

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004. 5

[10] A. C. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *CVPR*, pages 1–8, 2008. 1

[11] B. Hasan and D. Hogg. Segmentation using deformable spatial priors with application to clothing. In *BMVC*, 2010. 1

[12] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004. 6

[13] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008. 5

[14] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):978–994, 2011. 2

[15] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In *ACM international conference on Multimedia*, pages 619–628. ACM, 2012. 1

[16] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, pages 3330–3337, 2012. 1, 2

[17] M. Marszałek and C. Schmid. Accurate object recognition with shape masks. *IJCV*, 97(2):191–209, 2012. 5

[18] A. C. Murillo, I. S. Kwak, L. Bourdev, D. Kriegman, and S. Belongie. Urban tribes: Analyzing group photos from a social perspective. In *CVPR Workshops*, pages 28–35, 2012. 1

[19] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, pages 1–15, 2006. 5

[20] Z. Song, M. Wang, X.-s. Hua, and S. Yan. Predicting occupation via human clothing and contexts. In *ICCV*, pages 1084–1091, 2011. 1

[21] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. *ECCV*, pages 352–365, 2010. 2, 5

[22] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. *CVPR*, 2013. 2

[23] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1-2):61–81, 2005. 2

[24] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 3

[25] N. Wang and H. Ai. Who blocks who: Simultaneous clothing segmentation for grouping images. In *ICCV*, pages 1535–1542, 2011. 1

[26] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, pages 3570–3577, 2012. 1, 2, 6, 7

[27] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, 2011. 1, 2