LARGE SCALE VISUAL RECOGNITION OF CLOTHING, PEOPLE AND STYLES

M. Hadi Kiapour

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2015

Approved by:

Tamara L. Berg

Alexander C. Berg

Svetlana Lazebnik

Jan-Michael Frahm

Robinson Piramuthu

ABSTRACT

M. HADI KIAPOUR: LARGE SCALE VISUAL RECOGNITION OF CLOTHING, PEOPLE AND STYLES.
(Under the direction of Tamara L. Berg.)

Clothing recognition is a societally and commercially important yet extremely challenging problem due to large variations in clothing appearance, layering, style, body shape and pose. In this dissertation, we propose new computational vision approaches that learn to represent and recognize clothing items in images.

First, we present an effective method for parsing clothing in fashion photographs, where we label the regions of an image with their clothing categories. We then extend our approach to tackle the clothing parsing problem using a data-driven methodology: for a query image, we find similar styles from a large database of tagged fashion images and use these examples to recognize clothing items in the query. Along with our novel large fashion dataset, we also present intriguing initial results on using clothing estimates to improve human pose identification.

Second, we examine questions related to fashion styles and identifying the clothing elements associated with each style. We first design an online competitive style rating game called *Hipster Wars* to crowd source reliable human judgments of clothing styles. We use this game to collect a new dataset of clothing outfits with associated style ratings for different clothing styles. Next, we build visual style descriptors and train models that are

able to classify clothing styles and identify the clothing elements are most discriminative in every style.

Finally, we define a new task, Exact Street to Shop, where our goal is to match a real-world example of a garment item to the same exact garment in an online shop. This is an extremely challenging task due to visual differences between street photos that are taken of people wearing clothing in everyday uncontrolled settings, and online shop photos, which are captured by professionals in highly controlled settings. We introduce a novel large dataset for this application, collected from the web, and present a deep learning based similarity network that can compare clothing items across visual domains.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepst gratitude to my advisor, Professor Tamara Berg. Her knowledge, enthusiasm, guidance and brilliant ideas have enlightened my research through these years. Tamara taught me how to be a researcher, identify promising scientific objectives and explore novel directions for research. I am so grateful for the countless hours she spent with me for brainstorming and helping me present my findings. I was also extremely fortunate to collaborate closely with Professor Alex Berg. I feel very grateful to him for his incredible insights, comments and his continuous encouragement from the beginning of my graduate career. Special thanks are also extended to my wonderful thesis committee members: Professor Svetlana Lazebnik, Professor Jan-Michael Frahm, and Dr. Robinson Piramuthu.

I will forever be thankful to all the professors and teachers who have helped me succeed in my academic life. Special thanks to Professor Luis Ortiz whose knowledge and expertise has significantly contributed to the success of this thesis. Thanks to Professor Dimitris Samaras for all his teaching, friendliness, advices and insightful discussions. I would like to especially thank Professor Svetlana Lazebnik, who further contributed to shape the researcher in me by her invaluable collaboration and feedbacks. Thanks to Professor Jan-Michael Frahm for creating an environment of enthusiasm for learning, his support of my research, and his service on my thesis committee. Gratitude is extended

to Professor Fred Brooks and Professor Ron Alterovitz for sharing their enthusiasm on teaching in technical communication class.

I am also grateful to all of my industrial collaborators. I spent two summers at eBay Research where I had the chance to collaborate with world-class researchers and engineers. In particular, I would like to thank Dr. Robinson Piramuthu for his leadership and supervision which led me to develop a passion for tackling challenging, open-ended, real-world problems. I am also thankful to Dr. Hassan Sawaf for his extensive support and confidence in my work. Also thanks to all my collaborators during my internships, especially Dr. Wei Di and Dr. Vignesh Jagadeesh.

I want to thank Professor Kota Yamaguchi for being an excellent friend, mentor and a great inspiration for me. I truly appreciate and value everything I have learned from you. I would also like to thank Dr. Kevin Yager for his impressive work in our collaborative research with the Brookhaven National Labs.

Special thanks to Professor Mehrdad Shahshahani for sparking my passion for computer vision, his exceptional leadership, support and providing me with the opportunity to conduct research during my undergraduate years. I gratefully thank Professor Ali Farhadi for his supervision and guidance from the very early stages of my research in the area. Thank you to many wonderful professors and colleagues at KTH Royal Institute of Technology in Stockholm, especially Professor Stefan Carlsson and Professor Jan-Olof Eklundh for providing me the opportunity to become a better scientist.

This journey would not have been possible without the support of my family. To my parents, thank you for your endless love and encouraging me in all of my pursuits and

inspiring me to chase my dreams. Thank you uncle Michael and aunt Adele, for your wisdom, kindness and for believing in me. I am so blessed to have you in my life.

Thanks to the State University of New York at Stony Brook and the University of North Carolina at Chapel Hill, the faculty, and administrative staff of Computer Science Departments, for all their support throughout my PhD studies. Thanks to all my former professors, colleagues, and friends back in Sharif University of Technology. Finally, thank you to all my fellow students for making my life in graduate school a truly pleasurable experience. I will always appreciate our time together and I hope that our friendship lasts for a lifetime.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Motivation

Imagine waking up one day to a world where everyone wears the same outfit. What a strange experience would it be? The world would appear much less colorful and interesting. Clothing is an integral part of our daily lives, both at the individual and community levels. Choice of clothing communicates a great deal of non-verbal signals that can be interpreted consciously or unconsciously by the observer. In other words, we are what we wear. Our clothing often reveals hints of our wealth, occupation, religion, location and social identity. People purposely select different styles of clothing to wear in different types of social contexts. Fashion is a form of self expression, both to who create it and to the ones who wear it. Understanding clothing is essential to how we perceive the world and form impressions of the ones with whom we engage and interact.

The next natural question that comes to mind is how do we perceive clothing? Among our five senses, vision has an overriding importance in every aspect of our day-to-day lives. We gather a lot of knowledge about ourselves and the world around us by visual perception. While it seems like a trivial task, human vision is a product of an extraordinary developed and complex system. In brain itself, neurons devoted to visual processing take up to about 30 percent of the cortex, as compared to 8 percent for touch and just 3 percent for hearing. We heavily rely on recognition of large variations of visual objects

in order to navigate and act in our daily lives. We rapidly and effortlessly recognize objects in various contexts and estimate their geometric relationships even when they are encountered in unusual orientations, under different illumination conditions or partially occluded by other objects in a visually complicated environment. In particular, visual perception plays a fundamental role in the way people form impressions of and make inference about their clothing. People make snap judgements about the aesthetic value of attires. It takes only a glimpse for a person to judge the visual appeal of an outfit or a fashion style. We have an extraordinary cognitive ability to analyze what we see, both at a high level and in finding the most salient constructing elements. Since clothing is generally composed of visual elements, computational vision techniques are the best avenue to automate the exploration of clothing at a large scale. While we highly rely on computers for analyzing large amounts of data, when it comes to visual recognition, human brain far surpasses even the most advanced artificial vision systems. The difference becomes even more evident as one explores complex visual data such as clothing. Clothing produces extremely complex visual patterns, due to large number of possible garment items, large variations in configurations, deformation, appearance, layering, occlusion and body poses. While artificial intelligence researchers strive to bridge the gap between human and machine intelligence, a little previous work is devoted to the particular problem of clothing recognition. Our main goal in this thesis is to study and develop scalable computer vision and machine learning systems that learn to represent and identify visual data in order to recognize clothing at a large scale.

To grasp the potential impacts of clothing recognition systems, consider e-commerce.

In 2014, retail e-commerce revenues from apparel and accessories sales amounted to 52.2 billion U.S. dollars and is projected to exceed 80 billion dollars in 2018[1]. Search is an incredibly important part of e-commerce. People go to online marketplaces looking for a specific product or type of clothing. While e-commerce has historically relied heavily on text-based search engines, visual search technology is currently one of the fastest growing and exciting trends in e-commerce. The results of a text-based search engine are only as good as the user's ability to describe an item and also depends on how well the given keywords match to the product description on the web. In contrast, visual search can provide a significantly more intuitive way to connect with information which leads to more accurate results. Today's mobile developers strive to build applications that allow customers to snap a photo of a clothing product on the street in the real world and directly search through massive number of products in online shops. This is a very challenging task due to extreme visual differences between real-world photos taken of people wearing clothing in everyday uncontrolled settings, and online shop photos that are taken of clothing items on models, mannequins or in isolation, captured by professionals in highly controlled settings. In addition, highly costumed garment items and unbranded apparels make visual retrieval of clothing even more challenging compared to many other object categories such as electronics.

While search plays a central role in today's world of e-commerce, the future of e-commerce relies on adding a sense of discovery to the utilitarian nature of search. Recommendation engines and product discovery sites are nowadays among the top fast growing

[1]http://www.statista.com/statistics/278890/us-apparel-and-accessories-retail-e-commerce-revenue/

trends in e-commerce. Targeted discovery, which means guiding consumers to specific products based on their history and personal preferences, creates a custom experience for digital shoppers according to their passions. This is even more important in the clothing and fashion industry, since many desirables are less about search and more about discovery. Some examples are "what should I wear with my cowgirl boots?", "what should I wear to look a bit cooler?" or "what's the best outfit for the weather today?". Building artificial intelligence systems that construct complex semantics from the enormous amount of clothing data available online, can be of impressive commoditization value.

With the rise of social networking in the past years, many online communities are formed around connecting people who share the same fashion taste or are passionate about sparing and taking inspirations. Fashion is a fast pace, exciting, transcending field full of creativity. People deeply care about sharing and communicating inspirations visually. Enabling computers to have a sense of current fashion, predict future trends and to understand personal styles can have an exceptional value in numerous applications including categorizing personal or public photo galleries, personalized advertisements, personalized outfit composition, recommendation systems, and matchmaking in online dating platforms.

In this dissertation, we study three main problems, all essential to a comprehensive automated clothing recognition system: clothing parsing (Chapter 3), fashion style recognition (Chapter 4) and visual matching of clothing items (Chapter 5).

1.2 Thesis Statement

This thesis addresses the problem of clothing recognition using computational visual representations empowered by machine learning. We introduce effective techniques for representation and identification of clothing in visual data with applications in clothing parsing, recognizing people's fashion styles and clothing retrieval in large scale.

1.3 Outline of Contributions

In Chapter 3, we propose novel approaches for parsing clothing in fashion photographs, an extremely challenging problem due to the large number of possible garment items, variations in configuration, garment appearance, layering, and occlusion. We first present a probabilistic approach for labeling super pixels in an image with their clothing labels using conditional random fields. Next, we extend our clothing parsing system by using a retrieval-based approach: For every query image, we find similar styles from a large database of tagged fashion images and use these examples to recognize clothing items in 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 from retrieved examples. We demonstrate a prototype application for pose-independent visual garment retrieval and present intriguing initial results on using clothing estimates to improve pose identification.

Chapter 4, studies what our clothing reveals about our personal style. We first design an online competitive style rating game called *Hipster Wars* to crowd source reliable

human judgments of style. We use this game to collect a new dataset of clothing outfits with associated style ratings for 5 style categories: hipster, bohemian, pinup, preppy, and goth. Next, we train models for between-class and within-class classification of styles. Finally, we explore methods to identify clothing elements that are generally discriminative for a style, and methods for identifying items in a particular outfit that may indicate a style.

In Chapter 5, we define a new task, Exact Street to Shop, where our goal is to match a real-world example of a garment item to the same garment in an online shop. This is an extremely challenging task due to visual differences between street photos (pictures of people wearing clothing, captured in everyday, uncontrolled settings) and online shop photos (pictures of clothing items on people, mannequins, or in isolation, captured by professionals in highly controlled settings). We collect a novel large dataset for this application containing photos from online shops and daily outfit photos. We present our deep learning based similarity network for measuring the similarity between pairs of clothing items across different visual domains.

In summary the novel contributions presented in this thesis are as follows:

- Novel clothing parsing approaches for precise prediction of clothing items and their location in images.
- Large novel dataset for studying clothing parsing, consisting of fashion photos
- Initial experiments on how clothing prediction can improve pose estimation.
- An online competitive rating game to collectively compute style ratings based on

human judgments.

- A new style dataset depicting different fashion styles with associated crowd sourced style ratings.
- Between-class and within-class classification of styles.
- Experiments to identify the outfit elements that are most predictive for a fashion style or within an image.
- Introduction of the Exact Street to Shop task
- A novel large dataset, the Exact Street to Shop Dataset, for street-to-shop clothing retrieval.
- A deep learning based similarity network for the Exact Street to Shop retrieval task.
- Human evaluations of the Exact Street to Shop task.

CHAPTER 2: RELATED WORK

2.1 Clothing Parsing

Image parsing has been studied as a step toward general image understanding, where the goal is to assign a semantic label to every pixel or segmentation region in an image (Shotton et al., 2006; Gould et al., 2009; Tighe and Lazebnik, 2010; Farabet et al., 2012; Guo and Hoiem, 2012; Ladicky et al., 2010; Liu et al., 2011; Long et al., 2015). Many of the existing image parsing methods use Markov random fields (MRF) or conditional random fields (CRF), with higher order potentials, long-range dependencies and fully connected graphs to achieve semantic segmentation (Krahenbuhl and Koltun, 2011, 2013; Vineet et al., 2012). A growing number of researchers combine state-of-the-art methods with objects detection and scene recognition (Yao et al., 2012; Kim et al., 2012; Tighe and Lazebnik, 2013; Tighe et al., 2014). More recently, supervised deep learning approaches have proved immensely successful in semantic image segmentation (Luo et al., 2012, 2013; Long et al., 2015; Chen et al., 2015b; Zheng et al., 2015). In this thesis we study clothing parsing, which is similar in spirit to general image parsing, but focuses on estimating labelings for a particularly interesting type of object, people wearing clothes. We build models to estimate an intricate parse of a person's outfit into its constituent garments. There has been growing interest in clothing parsing in the computer vision and multimedia communities (Shotton et al., 2006; Wang and Ai, 2011; Dong et al.,

2013; Liu et al., 2014; Hasan and Hogg, 2010; Scheffler and Odobez, 2011; Yang and Yu, 2011; Gallagher and Chen, 2008; Jammalamadaka et al., 2013; Simo-Serra et al., 2014). In Yang and Yu (2011), clothing recognition is used in surveillance videos. In Hasan and Hogg (2010), they improve the MRF formulation by adding prior models on shape and color of clothing items. In Scheffler and Odobez (2011), the regions around faces are labeled as skin, hair, clothing and background. The work of Wang and Ai (2011) attacks multi-person clothing segmentation in highly occluded images. In Simo-Serra et al. (2014), they incorporate appearance and location priors for each garment, as well as symmetry in their parsing. Dong et al. (2013) introduce *Parselets*, mid-level segments that carry strong semantic information, into parsing. In Chapter 3, we tackle clothing parsing problem as an object segmentation using CRFs. Our main contribution lies in defining the unary potential, where we use a pose estimation algorithm (Yang and Ramanan, 2011) to model a clothing type. Great performance was obtained when the system was given information about which garment classes, but not their location, are present for each test image. This issue is partially addressed in 3.3, where we utilize over 300 thousand weakly labeled images, where the weak annotations are in the form of image-level tags.

2.2 Pose Estimation

Pose estimation is a popular and well-studied problem. Some previous approaches have considered pose estimation as a labeling problem, assigning most likely body parts to super pixels (Mori et al., 2004), or triangulated regions (Ren et al., 2005). Other

earlier attempts were based on detecting body part (Ramanan, 2006; Andriluka et al., 2009; Marcin and Ferrari, 2009). The work of Yang and Ramanan (2011) uses a mixture model of parts, which jointly captures the spatial relations between part locations and co-occurrence between parts. Mixture models are improved in Johnson and Everingham (2011) to handle much larger quantities of training data. Higher-order spatial correspondences were modeled in hierarchical models (Tian et al., 2012). More recently, researchers have successfully deployed deep learning methods for human pose estimation (Toshev and Szegedy, 2014; Pfister et al., 2014; Tompson et al., 2014; Xianjie and Yuille, 2014). Our pose estimation subgoal builds on the method of Yang and Ramanan (2011), where we extend our approach to incorporate clothing parsing in mixture models for improving pose identification.

2.3 Semantic Clothing Recognition

There has been a growing interest in applications of clothing recognition such as learning semantic clothing attributes (Chen et al., 2012; Bossard et al., 2012), identifying people based on their outfits, predicting occupation (Song et al., 2011; Shao et al., 2013), urban tribes (Murillo et al., 2012; Kwak et al., 2013), fashion styles (Kiapour et al., 2014), outfit similarity (Vittayakorn et al., 2015) and outfit recommendations (Liu et al., 2012a). Some recent attempts also aimed to automatically reason about aesthetics and fashionability of clothing in a photograph(Yamaguchi et al., 2014; Simo-Serra et al., 2015). Chen et al. (2015a) investigate the possible effects of New York fashion shows on street-chic images of New Yorkers. Veit et al. (2015) train convolutional neural networks

on large scale co-purchase datasets obtained from online shops to predict what items may go well together. In Jing et al. (2015), they deploy distributed computational platforms to build a large-scale clothing search system for commercial applications.

2.4 Clothing Retrieval

Image retrieval is a fundamental problem for computer vision with wide applicability to commercial systems. Many recent retrieval methods at a high-level consist of three main steps: pooling local image descriptors, such as Fisher Vectors (Perronnin and Dance, 2006; Perronnin et al., 2010b,a) or VLAD (Jegou et al., 2010), dimensionality reduction, and indexing. Lim et al. (2013) used keypoint detectors to identify furniture items by aligning 3D models to 2D image regions. Recently, Gong et al. (2014) proposed a multi-scale orderless pooling scheme on deep CNN activations (Krizhevsky et al., 2012) for indexing that significantly improved the geometric invariance of the final representation over global CNN activations. Generally, these methods work quite well for instance retrieval of rigid objects, but may be less applicable for retrieving the soft, deformable clothing items that are our focus.

Despite recent advances in generic image retrieval, there have been relatively few studies focused specifically on clothing retrieval. Some related works have performed garment retrieval using parsing (Yamaguchi et al., 2012), using global or fine-grained attribute prediction (Di et al., 2013) or hashing representations that are able to match high-level category and attributes (Lin et al., 2015). There have also been some efforts on cross-scenario retrieval (Liu et al., 2012b,a; Fu et al., 2012; Kalantidis et al., 2013).

Our work in Chapter 5 is inspired by the street-to-shop (Liu et al., 2012b) approach, which tackles the domain discrepancy between street photos and shop photos using sparse representations. However, their approach depends on upper/lower body detectors to align local body parts in street and shop images, which may not be feasible in all types of shop images. They also evaluate retrieval performance in terms of a fixed set of hand-labeled attributes. For example, evaluating whether both the query and shop images depict a "blue, long-sleeved, shirt". While this type of evaluation may suit some shoppers' needs, our work aims to find *exactly* the same item depicted in a street photo in an online shop.

2.5 Domain Adaptation and Deep Similarity Learning

The concept of adapting models between different dataset domains has been well explored. Many works in this area tackle the domain adaptation problem by learning a transformation that aligns the source and target domain representations into a common feature space (Bell and Kavita, 2015; Fernando et al., 2013; Gopalan et al., 2011; Gong et al., 2012). Other approaches have examined domain adaptation methods for situations where only a limited amount of labeled data is available in the target domain. These methods train classifiers on the source domain and regularize them against the target domain (Bergamo and Torresani, 2010; Saenko et al., 2010). Recently, as deep convolutional neural networks are becoming ubiquitous for feature representations, supervised deep CNNs have proved to be extremely successful for the domain adaptation task (Donahue et al., 2014; Hoffman et al., 2014; Yosinkski et al., 2014). Our data can be seen as consisting of two visual domains, shop images and street images. Other examples include

methods for fine-grained object retrieval (Wang et al., 2014; Lai et al., 2015), face verification (Schroff et al., 2015; Taigman et al., 2014), or image patch-matching (Zagoruyko and Komodakis, 2015; Han et al., 2015; Zbontar and LeCun, 2015). These techniques learn representations coupled with either predefined distance functions, or with more generic learned multi-layer network similarity measures. In our similarity learning method, presented in Chapter 5, we learn a multi-layer network similarity measure on top of existing pre-trained deep features that is capable of predicting a similarity score given images of two different visual domains.

CHAPTER 3: CLOTHING PARSING

3.1 Introduction

In this chapter we tackle the problem of clothing parsing in fashion photographs, an extremely challenging problem due to the large number of possible garment items, variations in configuration, garment appearance, layering, and occlusion. We study clothing estimation at a much more general scale than previous works for real-world pictures. We consider a large number (53) of different garment types, e.g. shoes, socks, belts, rompers, vests, blazers, hats, etc., and explore techniques to accurately parse pictures of people wearing clothing into their constituent garment pieces. We also demonstrate a prototype application for pose-independent visual garment retrieval. Furthermore, we also exploit the relationship between clothing and the underlying body pose in two directions: to estimate clothing given estimates of pose, and to estimate pose given estimates of clothing.

3.2 CRF Parsing

In this approach, we consider the problem of predicting a clothing parse given estimates for human body pose. Clothing parsing is formulated as a labeling problem, where images are segmented into superpixels and then clothing labels for every segment are predicted in a CRF model. Unary potentials account for clothing appearance and

clothing item location with respect to body parts. Pairwise potentials incorporate label smoothing, and clothing item co-occurrence.

3.2.1 Dataset

We use Fashionista dataset described in detail in Yamaguchi et al. (2012), useful for training and testing clothing estimation techniques. In this dataset, there are 685 selected photos with good visibility of the full body and covering a variety of clothing items, fully annotated with ground truth clothing labels and pose annotations for 14 body parts. In the ground truth data set, there are 53 different clothing items, of which 43 items have at least 50 image regions. Adding additional labels for *hair*, *skin*, and *null* (background), gives a total of 56 different possible clothing labels.

3.2.2 Problem Formulation

We formulate the clothing parsing problem as a labeling of image regions. Let I denote an image showing a person. The goal is to assign a label of a clothing or null (background) item to each pixel, analogous to the general image parsing problem. However, in this work we simplify the clothing parsing problem by assuming that uniform appearance regions belong to the same item, as reported in (Gallagher and Chen, 2008), and reduce the problem to the prediction of a labeling over a set of superpixels. We denote the set of clothing labels by $L \equiv \{l_i\}$, where $i \in U$ denotes a region index within a set of superpixels U in I, and l_i denotes a clothing label for region indexed by i (e.g., $l_i = t\text{-}shirt$ or $pants$). Also let $\boldsymbol{s}_i$ denote the set of pixels in the i-th region.

We take a probabilistic approach to the clothing parsing problem. Within our framework, we reduce the general problem to one of *maximum a posteriori* (MAP) assignments; we would like to assign clothing labels based on the most likely joint clothing label assignments under a probability distribution $P(L|I)$ given by the model. However, it is extremely difficult to directly define such a distribution due to the varied visual appearance of clothing items. Therefore, we introduce another variable, human pose configuration, and consider the distribution in terms of interactions between clothing items, human pose, and image appearance. We denote a human pose configuration by $X \equiv \{\boldsymbol{x}_p\}$, which is a set of image coordinates $\boldsymbol{x}_p$ for body joints p, e.g., *head* or *right elbow*.

Ideally, one would then like to find the joint MAP assignment over both clothing and pose labels with respect to the joint probability distribution $P(X, L|I)$ simultaneously. However, such MAP assignment problems are often computationally intractable because of the large search space and the complex structure of the probabilistic model. Instead, we split the problem into parts, solving the MAP assignment of $P(L|X, I)$ and $P(X|I)$ separately. Our clothing parsing pipeline proceeds as follows:

1. Obtain superpixels $\{\boldsymbol{s}_i\}$ from an image I
2. Estimate pose configuration X using $P(X|I)$
3. Predict clothes L using $P(L|X, I)$
4. Optionally, re-estimate pose configuration X using model $P(X|L, I)$

Figure 3.1 shows an example of this pipeline. We now briefly describe each step and

(a) Superpixels (b) Pose estimation (c) Predicted Clothing Parse (d) Pose re-estimation

Figure 3.1: Clothing parsing pipeline: **(a)** Parsing the image into Superpixels (Arbelaez et al., 2011), **(b)** Original pose estimation using state of the art flexible mixtures of parts model (Yang and Ramanan, 2011). **(c)** Precise clothing parse output by our proposed clothing estimation model (note the accurate labeling of items as small as the wearer's necklace, or as intricate as her open toed shoes). **(d)** Optional re-estimate of pose using clothing estimates (note the improvement in her left arm prediction, compared to the original incorrect estimate down along the side of her body).

formally define our probabilistic model.

3.2.3 Superpixels

We use an image segmentation algorithm (Arbelaez et al., 2011) to obtain superpixels. The algorithm provides a hierarchical segmentation, but we set the threshold value to 0.05 to obtain a single over-segmentation for each image. This process typically yields between a few hundred to a thousand regions per image, depending on the complexity of the person and background appearance (Fig 3.1(a) shows an example).

3.2.4 Pose Estimation

We begin our pipeline by estimating pose $\hat{X}$ using $P(X|I)$:

$$\hat{X} \in \arg\max_X P(X|I) \; . \tag{3.1}$$

For our initial pose estimate, we make use of (Yang and Ramanan, 2011). In addition to the above terms, this model includes an additional hidden variable representing a type label for pose mixture components, $T \equiv \{t_p\}$ for each body joint p, containing information about the types of arrangements possible for a joint. Therefore, the estimation problem is written as $(\hat{X}, \hat{T}) \in \arg\max_{X,T} P(X, T|I)$. The scoring function used to evaluate pose (Yang and Ramanan, 2011) is:

$$\begin{aligned} \ln P(X, T|I) \equiv \textstyle\sum_p \boldsymbol{w}_p(t_p)^{\mathrm{T}} \boldsymbol{\phi}(x_p|I) + \\ \textstyle\sum_{p,q} \boldsymbol{w}_{p,q}(t_p, t_q)^{\mathrm{T}} \boldsymbol{\psi}(x_p - x_q) - \ln Z, \end{aligned} \tag{3.2}$$

where, $\boldsymbol{w}$ are the model parameters, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are feature functions, and Z is a partition function.

3.2.5 Clothing Labeling

Once we obtain the initial pose estimate $\hat{X}$, we can proceed to estimating the clothing labeling:

$$\hat{L} \in \arg\max_L P(L|\hat{X}, I) \ . \tag{3.3}$$

We model the probability distribution $P(L|X, I)$ with a second order conditional random field (CRF):

$$\begin{aligned} \ln P(L|X, I) \equiv & \sum_{i \in U} \Phi(l_i|X, I) + \sum_{(i,j) \in V} \lambda_1 \Psi_1(l_i, l_j) \\ & + \sum_{(i,j) \in V} \lambda_2 \Psi_2(l_i, l_j|X, I) - \ln Z, \end{aligned} \tag{3.4}$$

where V is a set of neighboring pairs of image regions, λ_1 and λ_2 are model parameters, and Z is a partition function.

We model the unary potential function Φ using the probability of a label assignment, given the feature representation of the image region $\boldsymbol{s}_i$:

$$\Phi(l_i|X, I) \equiv \ln P(l_i|\phi(\boldsymbol{s}_i, X)). \tag{3.5}$$

We define the feature vector ϕ as the concatenation of (1) normalized histograms of RGB color, and (2) normalized histogram of CIE L*a*b* color, (3) histogram of Gabor filter responses, (4) normalized 2D coordinates within the image frame, and (5) normalized 2D coordinates with respect to each body joint location $\boldsymbol{x}_p$. In our experiments, we use 10 bins for each feature type. Using a 14-joint pose estimator, this results in a 360 dimensional sparse representation for each image region. For the specific marginal probability model $P(l_i|\phi(\boldsymbol{s}, X))$, we experimentally evaluated a few distributions and found that logistic regression works well for our setting.

The binary potential function Ψ_1 is a log empirical distribution over pairs of clothing region labels in a single image:

$$\Psi_1(l_i, l_j) \equiv \ln \tilde{P}(l_i, l_j). \tag{3.6}$$

This term serves as a prior distribution over the pairwise co-occurrence of clothing labels (e.g. shirts are near blazers, but not shoes) in neighboring regions within an image. We compute the function by normalizing average frequency of neighboring label pairs in training samples.

The last binary potential in (3.4) estimates the probability of neighboring pairs having the same label (i.e. label smoothing), given their features, ψ:

$$\Psi_2(l_i, l_j|X, I) \equiv \ln P(l_i = l_j|\psi(\boldsymbol{s}_i, \boldsymbol{s}_j, X)). \tag{3.7}$$

We define the feature transformation to be

$$\psi(\boldsymbol{s}_i, \boldsymbol{s}_j) \equiv [(\phi(\boldsymbol{s}_i) + \phi(\boldsymbol{s}_j))/2, |\phi(\boldsymbol{s}_i) - \phi(\boldsymbol{s}_j)|] \tag{3.8}$$

As with the unary potential, we use logistic regression for this probability distribution.

Because of the loopy structure of our graphical model, it is computationally intractable to solve (3.3) exactly. Therefore, we use belief propagation to obtain an approximate MAP assignment, using the libDAI (Mooij, 2010) implementation.

In practice, regions outside of the bounding box around pose estimation are always

background. Therefore, in our experiment, we fix these outside regions to *null* and run inference only within the foreground regions.

3.2.6 Training

Training of our parser includes parameter learning of the pose estimator $P(X|I)$ and $P(X|L, I)$, learning of potential functions in $P(L|X, I)$, and learning of CRF parameters in (3.4).

Pose estimator: The training procedure of (Yang and Ramanan, 2011) uses separate negative examples, sampled from scene images to use the pose estimator as a detector. Since our problem assumes a person is shown, we do not use a scene based negative set, but rather mine hard negative examples using false detections in our images. We treat a detection as negative if less than 30% of the body parts overlap with their true locations with ratio more than 60%.

Potential functions: We learn the probability distributions $P(l_i|\phi)$ and $P(l_i = l_j|\psi)$ in (3.5) and (3.7) using logistic regression with L2 regularization (liblinear implementation (Fan et al., 2008a)). For each possible clothing item, e.g. *shirt* or *boots* we learn the distribution its regional features, $P(l_i|\phi)$. We learn this model using a one-versus-all approach for each item. This usually introduces an imbalance in the number of positive vs negative examples, so the cost parameter is weighted by the ratio of positive to negative samples.

CRF parameters: Our model (3.4) has two parameters λ_1 and λ_2. We find the best parameters by maximizing cross validation accuracy over pixels in our training data

using line search and a variant of the simplex method (`fminsearch` in Matlab). In our experiment, typically both λ_1 and λ_2 preferred small values (e.g., 0.01-0.1).

3.2.7 Experimental Results

We evaluate the performance of our approach using 685 annotated samples from the Fashionista Dataset (described in Sec 3.2.1). All measurements use 10-fold cross validation (9 folds used for training, and the remaining for testing). Since the pose estimator contains some random components, we repeat this cross validation protocol 10 times.

Clothing Parsing Accuracy

We measure performance of clothing labeling in two ways, using average pixel accuracy, and using mean Average Garment Recall (mAGR). mAGR is measured by computing the average labeling performance (recall) of the garment items present in an image, and then the mean is computed across all images. Table 3.1 shows a comparison for 8 versions of our approach. Full-a and Full-m are our models with CRF parameters learned to optimize pixel accuracy and mAGR respectively (note that the choice of which measure to optimize for is application dependent). The most frequent label present in our images is *background*. Naively predicting all regions to be *background* results in a reasonably good **77%** accuracy. Therefore, we use this as our baseline method for comparison. Our model (Full-a) achieves a much improved **89%** pixel accuracy, close to the result we would obtain if we were to use ground truth estimates of pose (**89.3%**). If no pose infor-

Method	Pixel acc	mAGR
Full-a	**89.0** ± 0.8	63.4 ± 1.5
with truth	89.3 ± 0.8	64.3 ± 1.3
without pose	86.0 ± 1.0	58.8 ± 2.1
Full-m	88.3 ± 0.8	**69.6** ± 1.7
with truth	88.9 ± 0.7	71.2 ± 1.5
without pose	84.7 ± 1.0	64.6 ± 1.6
Unary	88.2 ± 0.8	**69.8** ± 1.8
Baseline	77.6 ± 0.6	12.8 ± 0.1

Table 3.1: Clothing Parsing performance. Results are shown for our model optimized for accuracy (**top**), our full model optimized for mAGR (**2nd**), our model using unary term only (**3rd**), and a baseline labeling (**bottom**).

mation is used, clothing parsing performance drops significantly (**86%**). For mAGR, the Unary model achieves slightly better performance (**69.8%**) over the full model because smoothing in the full model tends to suppress infrequent (small) labels.

Qualitative evaluation

We also test our clothing parser on all 158k un-annotated samples in the Fashionista dataset. Since we don't have ground truth labels for these photos, we just report qualitative observations. From these results, we confirm that our parser predicts good clothing labels on this large and varied dataset. Figure 3.2 shows some good parsing results, even handling relatively challenging clothing (e.g. small hats, and partially occluded shoes). Generally the parsing problem becomes easier in highly distinguishable appearance situations, such as on clean backgrounds, or displaying distinctive clothing regions. Failure cases (Fig 3.3) are observed due to ambiguous boundaries between foreground and background, when initial pose estimates are quite incorrect, or in the presence of very coarse

Garment	Full-m	with truth	without pose
background	95.3± 0.4	95.6± 0.4	92.5± 0.7
skin	74.6± 2.7	76.3± 2.9	78.4± 2.9
hair	76.5± 4.0	76.7± 3.9	69.8± 5.3
dress	65.8± 7.7	67.7± 9.4	50.4±10.2
bag	44.9± 8.0	47.6± 8.3	33.9± 4.7
blouse	63.6± 9.5	66.2± 9.1	52.1± 8.9
shoes	82.6± 7.2	85.0± 8.8	77.9± 6.6
top	62.0±14.7	64.6±13.1	52.0±13.8
skirt	59.4±10.4	60.6±13.2	42.8±14.5
jacket	51.8±15.2	53.3±13.5	45.8±18.6
coat	30.8±10.4	31.1± 5.1	22.5± 8.8
shirt	60.3±18.7	60.3±17.3	49.7±19.4
cardigan	39.4± 9.5	39.0±12.8	27.9± 8.7
blazer	51.8±11.2	51.7±10.8	38.4±14.2
t-shirt	63.7±14.0	64.1±12.0	55.3±12.5
socks	67.4±16.1	67.8±19.0	74.2±15.0
necklace	51.3±22.5	46.5±20.1	16.2±10.7
bracelet	49.5±19.8	56.1±17.6	45.2±17.0

Table 3.2: Recall for selected garments

patterns. Other challenges include pictures with out of frame body joints, close ups of individual garment items, or no relevant entity at all.

3.2.8 Retrieving Visually Similar Garments

We build a prototype system to retrieve garment items via visual similarity in the Fashionista dataset. For each parsed garment item, we compute normalized histograms of RGB and L*a*b* color within the predicted labeled region, and measure similarity between items by Euclidean distance. For retrieval, we prepare a query image and obtain a list of images ordered by visual similarity. Figure 3.4 shows a few of top retrieved results for images displaying *shorts*, *blazer*, and *t-shirt* (query in leftmost col, retrieval results in right 4 cols). These results are fairly representative for the more frequent garment items in the dataset.

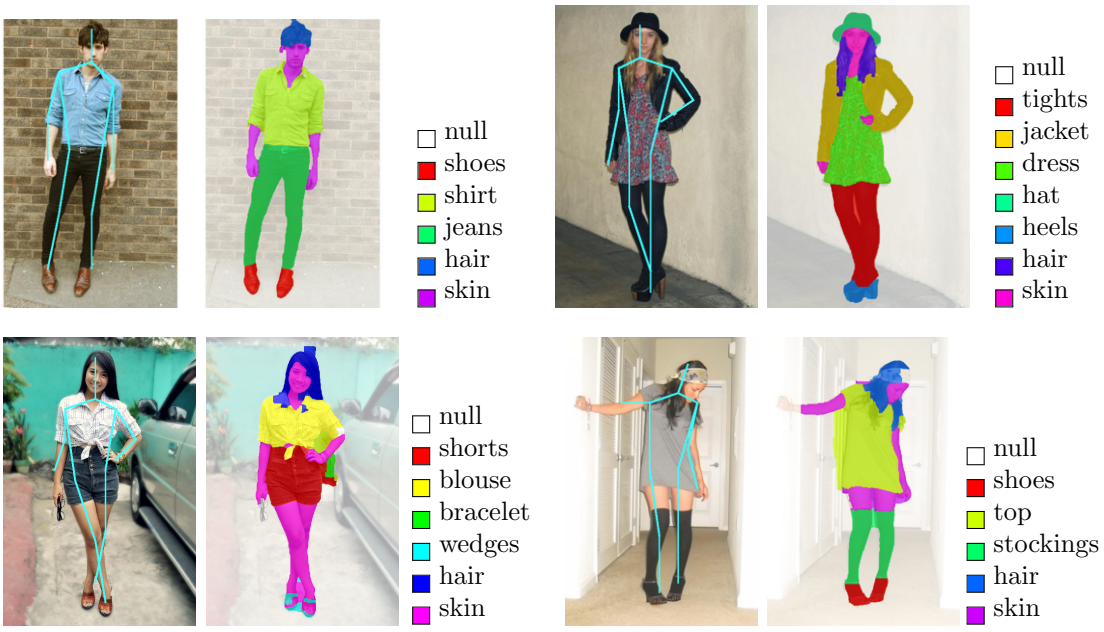
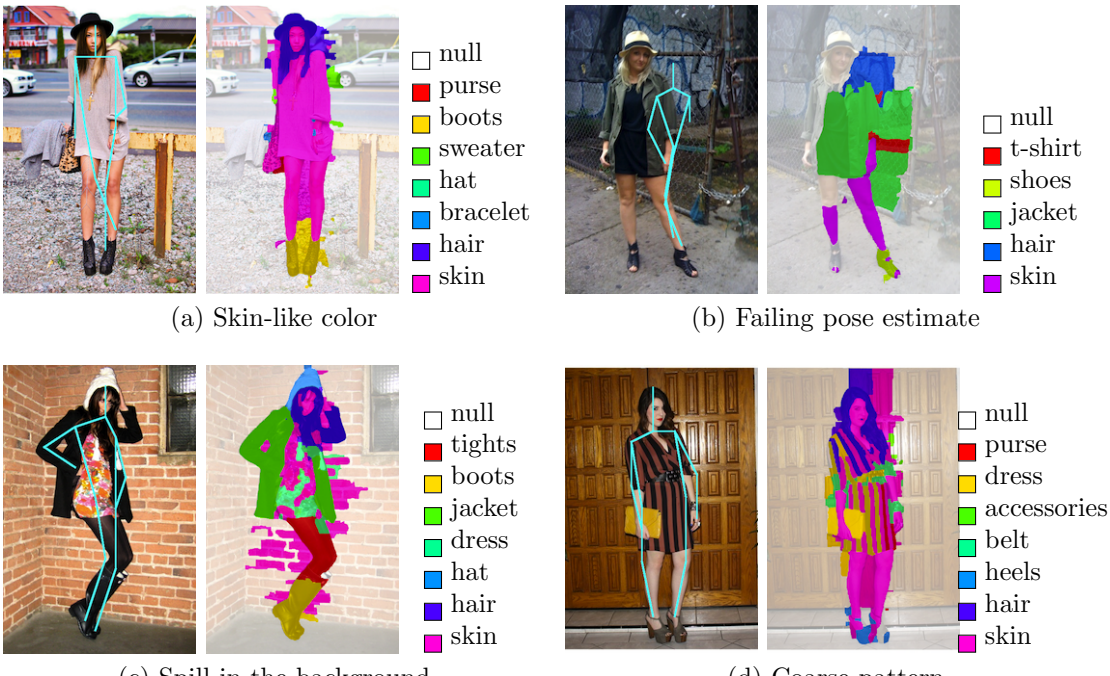


Figure 3.2: Example successful results on the Fahionista dataset



(a) Skin-like color

(b) Failing pose estimate

(c) Spill in the background

(d) Coarse pattern

Figure 3.3: Example failure cases

Figure 3.4: Prototype garment search application results. Query photo (left column) retrieves similar clothing items (right columns) *independent of pose and with high visual similarity.*

3.3 PAPER DOLL PARSING

The parsing approach in section 3.2 performed quite well in localization scenarios, where test images are parsed given user provided tags indicating depicted clothing items. However, this approach was less effective at unconstrained clothing parsing, where test images are parsed in the absence of any textual information (detection problem). In this section, we use a large-scale dataset to solve the clothing parsing problem in this challenging detection scenario. 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 recognize clothing items in 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 from retrieved examples.

Furthermore, we provide new experiments evaluating how the resulting clothing parse can benefit the general pose estimation problem.

3.3.1 Paper Doll Dataset

In this approach, we use the Fashionista dataset presented in 3.2.1 and a newly collected expansion called the Paper Doll dataset. The Fashionista dataset provides 685 images with clothing and pose annotation that we use for supervised training and performance evaluation, 456 for training and 229 for testing. The training samples are used to train a pose estimator, learn feature transformations, build global clothing models, and adjust parameters.

The Paper Doll dataset is a large collection of tagged fashion pictures with no manual annotation. 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 was also collected from chictopia.com, we exclude any duplicate pictures from the Paper Doll dataset. From the remaining, we select pictures tagged with at least one item and run a full-body pose detector (Yang and Ramanan, 2011) that we learned from the Fashionista dataset, keeping those having 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 of the items they are wearing, 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.

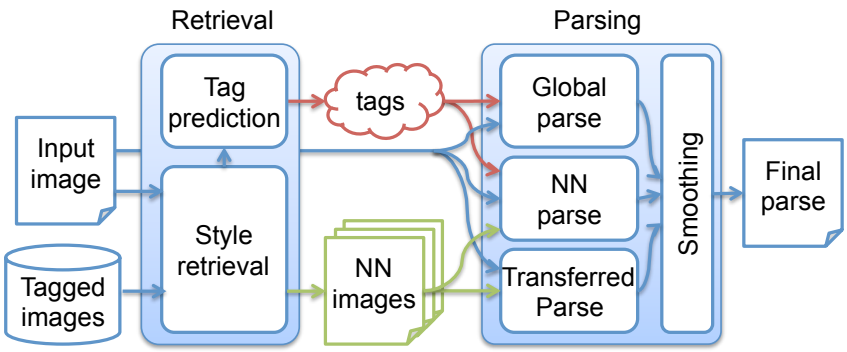


Figure 3.5: Parsing pipeline. Retrieved images and predicted tags augment clothing parsing.

3.3.2 Paper Doll Parsing Overview

Our parsing approach consists of two major steps:

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

Figure 3.5 depicts the overall parsing pipeline. Section 3.3.3 describes our tag prediction, and Section 3.3.4 details our parsing approach that combines three methods from the retrieval result.

Low-level features

We first run a pose estimator (Yang and Ramanan, 2011) and normalize the full-body bounding box to a fixed size, 302×142 pixels. The pose estimator is trained using the Fashionista training split and negative samples from the INRIA dataset (Dalal and Triggs, 2005). 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.

Table 3.3: Low-level features for parsing.

Name	Description
RGB	RGB color of the pixel.
Lab	L*a*b* color of the pixel.
MR8	Maximum Response Filters (Varma and Zisserman, 2005).
Gradients	Image gradients at the pixel.
HOG	HOG descriptor at the pixel (Dalal and Triggs, 2005).
Boundary Distance	Negative log-distance transform from the boundary of an image.
Pose Distance	Negative log-distance transform from 14 body joints and any body limbs.

Our methods draw from a number of dense feature types (each parsing method uses some subset). Table 3.3 summarizes them.

We compute Pose Distance by first interpolating 27 body joints estimated by a pose estimator (Yang and Ramanan, 2011) to obtain 14 points over body. Then, we compute a log-distance transform for each point. Also we compute log-distance transform of skeletal drawing of limbs (lines connecting 14 points). In total, we get a 15 dimensional vector for each pixel.

Whenever we use logistic regression (Fan et al., 2008b) built upon these features in parsing, we first normalize features by subtracting their mean and dividing by 3 standard deviations for each dimension. Also, when we use logistic regression, 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. We find parameters of logistic regressions by 3-fold cross validation within training data.

Figure 3.6: 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.

3.3.3 Tag prediction

We use the style descriptor introduced in Yamaguchi et al. (2013), useful for finding images with similar outfits. 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 performance in the high-recall regime. Figure 3.6 shows two examples of nearest neighbor retrievals.

Our tag prediction is based on a simple voting approach from KNN. While simple, a

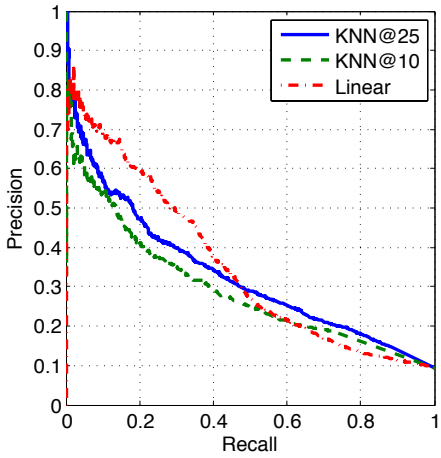


Figure 3.7: Tag prediction PR-plot. KNN performs better in the high-recall regime.

data-driven approach is shown to be effective in tag prediction (Guillaumin et al., 2009). In our approach, 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 3.7 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 prediction model. In the parsing

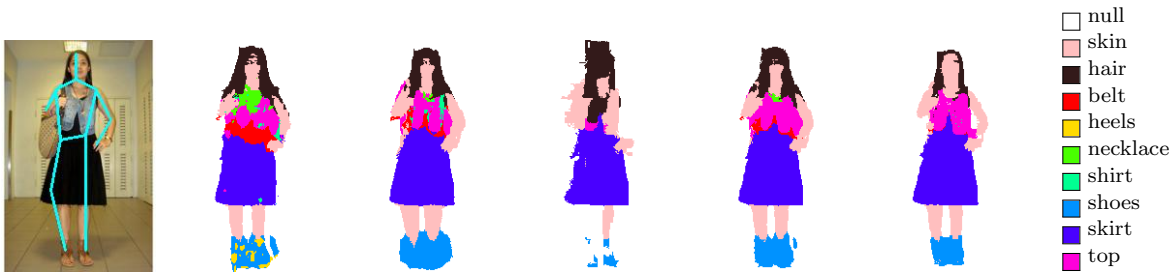


Figure 3.8: Parsing outputs at each step. Labels are MAP assignments of the scoring functions.

stage, we always include *background*, *skin*, and *hair* in addition to the predicted tags.

3.3.4 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 3.8 illustrates outputs from each parsing stage.

Pixel confidence

We denote the clothing item label at pixel i by y_i. The first step is to compute a confidence score of assigning clothing item l to y_i. We model this scoring function S_Λ as

the product mixture of three confidence functions.

$$\begin{aligned} S_\Lambda(y_i|\mathbf{x}_i, D) \equiv\ & S_{\text{global}}(y_i|\mathbf{x}_i, D)^{\lambda_1} \cdot \\ & S_{\text{nearest}}(y_i|\mathbf{x}_i, D)^{\lambda_2} \cdot \\ & S_{\text{transfer}}(y_i|\mathbf{x}_i, D)^{\lambda_3}, \end{aligned} \tag{3.9}$$

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

Global parse

The first term in our model is a global clothing likelihood, trained for each clothing item on the Fashionista training split. This is modeled as a logistic regression that computes the likelihood of a label assignment to each pixel for a given set of possible clothing items:

$$S_{\text{global}}(y_i|\mathbf{x}_i, D) \equiv P(y_i = l|\mathbf{x}_i, \theta_l^g) \cdot \mathbf{1}[l \in \tau(D)], \tag{3.10}$$

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.

We trained the model parameter θ_l^g 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.

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).

$$S_{\text{nearest}}(y_i|\mathbf{x}_i, D) \equiv P(y_i = l|\mathbf{x}_i, \theta_l^n) \cdot \mathbf{1}[l \in \tau(D)]. \tag{3.11}$$

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

In this step, we learn local appearance models using predicted pixel-level annotations from the retrieved samples computed during pre-processing detailed in Section 3.3.4. We train NN models using any pixel (with subsampling) in the retrieved samples in an one-vs-all fashion.

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 3.9 depicts an example). This approach is similar in spirit to approaches for general segmentation that transfer likelihoods using over-segmentation and matching (Borenstein and Malik, 2006; Leibe et al., 2008; Marszałek and Schmid, 2012); 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 (Tighe and Lazebnik, 2010) 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 (Felzenszwalb and Huttenlocher, 2004), 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

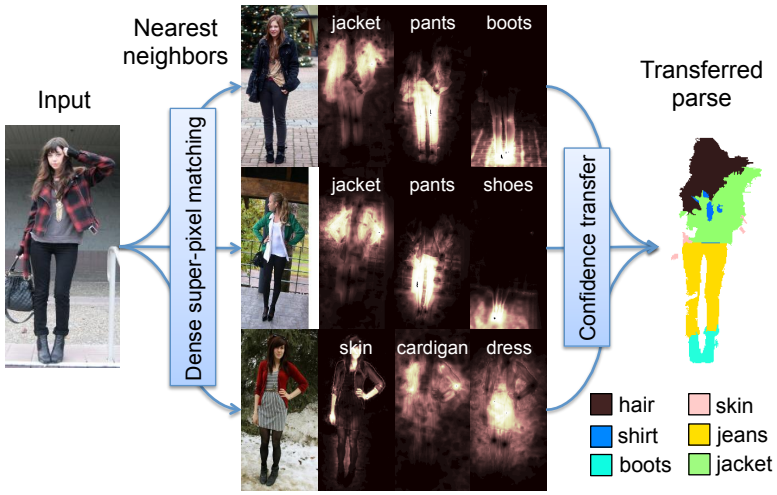


Figure 3.9: Transferred parse. We transfer likelihoods in nearest neighbors to the input via dense matching.

image using L2 Pose Distance.

2. At each super-pixel, compute a bag-of-words representation (Sivic and Zisserman, 2003) for each of RGB, Lab, MR8, and Gradient, and concatenate all.

3. Pick the closest super-pixel from each retrieved image using L2 distance on the concatenated bag-of-words feature.

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

$$S_{\text{transfer}}(y_i|\mathbf{x}_i, D) \equiv \frac{1}{Z}\sum_{r\in D}\frac{M(y_i, s_{i,r})}{1+\|h(s_i)-h(s_{i,r})\|}, \tag{3.12}$$

where we define

$$M(y_i, s_{i,r}) \equiv \frac{1}{|s_{i,r}|} \sum_{j \in s_{i,r}} P(y_i = l|\mathbf{x}_i, \theta_l^g) \cdot \mathbf{1}[l \in \tau(r)], \tag{3.13}$$

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 by $\tau(r)$, and the normalization constant by Z.

Combined confidence

After computing our three confidence scores, we combine them with parameter Λ to get the final pixel confidence S_Λ as described in Equation 3.9. 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):

$$\max_{\Lambda} \sum_{i \in F} \mathbf{1} \left[\tilde{y}_i = \arg\max_{y_i} S_\Lambda(y_i|\mathbf{x}_i) \right], \tag{3.14}$$

where $\tilde{y}_i$ is the ground truth annotation of the pixel i. For simplicity, we drop the nearest neighbors D in S_Λ from the notation. We use a simplex search algorithm over the simplex induced by the domain of Λ to solve for the optimum parameter starting from uniform values. In our experiment, we obtained $(0.41, 0.18, 0.39)$ using the training split of the Fashionista dataset.

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*.

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 Shotton et al. (Shotton et al., 2006), 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 smoothing 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 models $\hat{\Theta}_0$ (logistic regressions). 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:

$$\hat{Y}_t \in \arg\max_Y \prod_i \Phi(y_i|\mathbf{x}_i, S, \hat{\Theta}_t) \prod_{i,j \in V} \Psi(y_i, y_j|\mathbf{x}_i, \mathbf{x}_j), \tag{3.15}$$

where we define:

$$\Phi(y_i|\mathbf{x}_i, S, \hat{\Theta}_t) \equiv \quad S(y_i|\mathbf{x}_i)^{\lambda} \cdot P(y_i|\mathbf{x}_i, \theta_l^s)^{1-\lambda}, \tag{3.16}$$

$$\Psi(y_i, y_j|\mathbf{x}_i, \mathbf{x}_j) \equiv \quad \exp\{\gamma e^{-\beta\|\mathbf{x}_i - \mathbf{x}_j\|^2} \cdot \mathbf{1}\left[y_i \neq y_j\right]\}. \tag{3.17}$$

Here, V is a set of neighboring pixel pairs, λ, β, γ are the parameters of the model, which we set to $\beta = -0.75, \lambda = 0.5, \gamma = 1.0$ according to perceptual quality in the training images[1]. We use the graph-cut algorithm (Boykov et al., 2001; Boykov and Kolmogorov, 2004; Kolmogorov and Zabin, 2004) to find the optimal solution.

With the updated estimate of the labels $\hat{Y}_t$, we learn 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%.

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

[1]It is computationally prohibitive to optimize the parameters.

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 entire Paper Doll dataset took a few days using our Matlab implementation in a distributed environment. For a novel query image, our full parsing pipeline takes 20 to 40 seconds, including pose estimation. The major computational bottlenecks are the nearest neighbor parse and iterative smoothing.

3.3.5 Experimental results

Parsing performance

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 score over pixels. In addition, we also include foreground accuracy (See Eq. 3.14) 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,

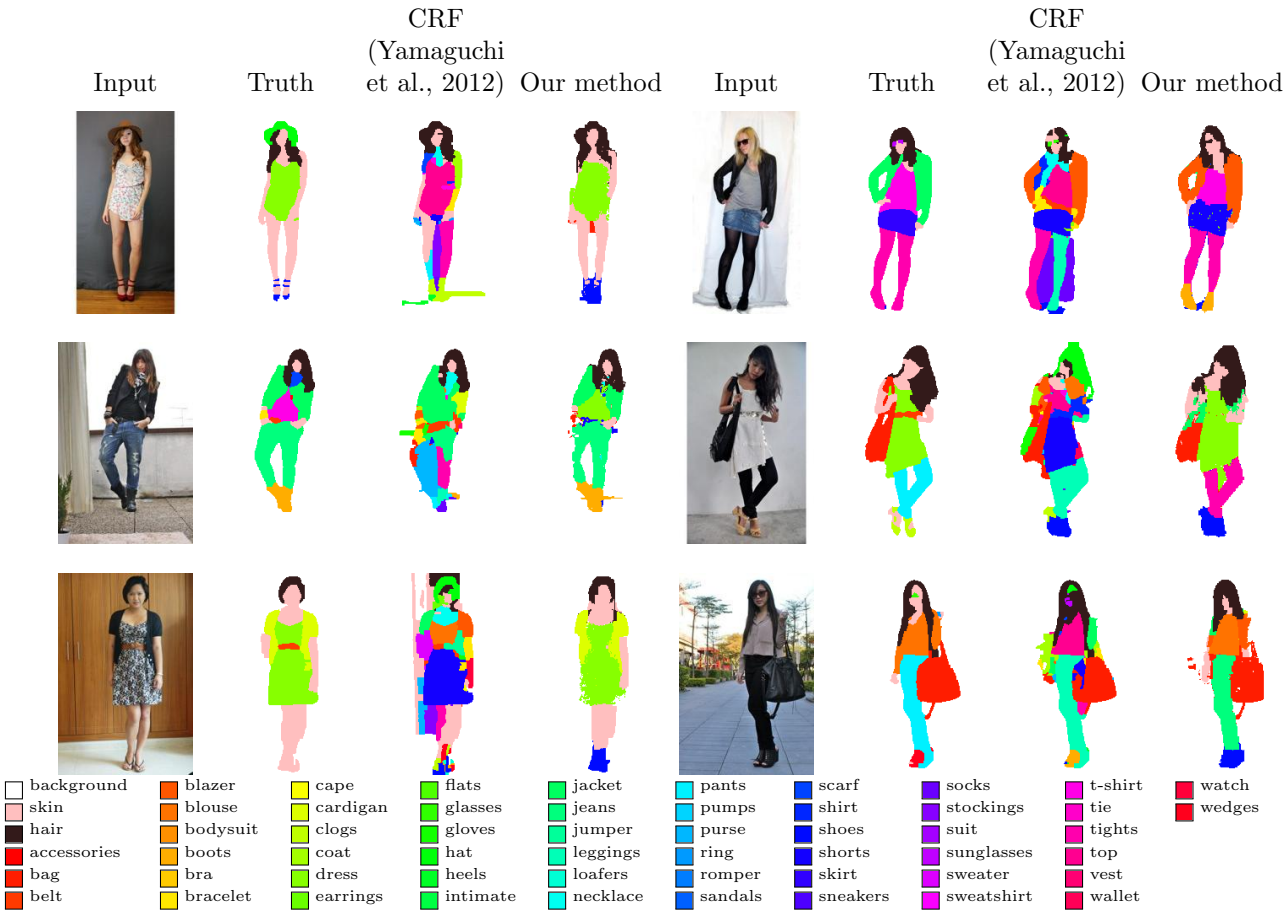


Figure 3.10: Parsing examples (best seen in color). Our method sometimes confuses similar items, but gives overall perceptually better results.

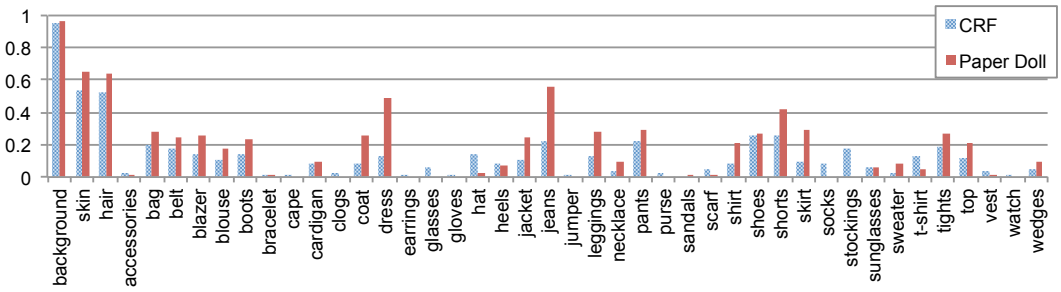


Figure 3.11: F-1 score of non-empty items. We observe significant performance gains, especially for large items.

Method	Accuracy	F.g. Accuracy	Avg. Precision	Avg. Recall	Avg. F-1
CRF (Yamaguchi et al., 2012)	77.45	23.11	10.53	**17.20**	10.35
Final	**84.68**	**40.20**	**33.34**	15.35	**14.87**
Global	79.63	35.88	18.59	15.18	12.98
Nearest	80.73	38.18	21.45	14.73	12.84
Transferred	83.06	33.20	31.47	12.24	11.85
Combined	83.01	39.55	25.84	15.53	14.22

Table 3.4: Parsing performance for final and intermediate results (MAP assignments at each step) in percentage.

F-1 does not necessarily match the geometric mean of average precision and recall.

Table 3.4 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 (Yamaguchi et al., 2012). 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**%).

In Table 3.4, we can observe that different parsing methods have different strength. For example, the global parse achieves higher recall than others, but the nearest-neighbor parse is better in foreground accuracy. Ultimately, we find that the combination of all three methods produces the best result. We provide further discussion in Section 3.3.5.

Figure 3.10 shows examples from our parsing method, compared to the ground truth annotation and the CRF-based method.We observe that our method usually produces a parse that is qualitatively superior to the previous approach in that it usually respects

the item boundary. In addition, many confusions are between similar item categories, e.g., predicting *pants* as *jeans*, or *jacket* as *blazer*. These confusions are reasonable 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 3.11 plots F-1 scores for non-empty items (items predicted on the test set) comparing the CRF-based method with our new method. Our model outperforms the previous 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.

Discussion

Though our method is successful at foreground prediction overall, there are a few drawbacks to our approach. By design, our style descriptor aims to represent whole outfit style rather than specific details of the outfit. Consequently, small items like accessories tend to be weighted less during retrieval and are therefore poorly predicted during parsing. This is also reflected in Table 3.4; the global parse is better than the nearest parse or the transferred parse in recall, because only the global parse could retain a stable appearance model of small items. However, in general, prediction of small items is inherently extremely challenging because of limited appearance information.

Another problem is the prevention of conflicting 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 helps reduce such confusions, but the parsing result

sometimes contains one item split into two conflicting items.

These two problems are the root of the error in tag prediction – either an item is missing or incorrectly predicted – and result in the performance gap between detection and localization. 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. 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 be correctly parsed.

3.4 Parsing for Pose Estimation

In this section, we examine the effect of using clothing parsing to improve pose estimation. We take advantage of pose estimation in parsing, because clothing items are closely related to body parts. Similarly, we can benefit from clothing parsing in pose estimation, by using parsing as a contextual input in estimation.

We compare the performance of the pose estimator (Yang and Ramanan, 2011), using three different contextual input.

- **Baseline**: using only HOG feature at each part.

Table 3.5: Pose estimation performance with or without conditional parsing input.

Average precision of keypoints (APK)

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
Baseline	0.9956	0.9879	0.8882	0.5702	0.7908	0.8609	0.8149	0.8440
Clothing	1.0000	0.9927	0.8770	0.5601	0.8937	0.8868	0.8367	0.8639
- Ground truth	1.0000	0.9966	0.9119	0.6411	0.8658	0.9063	0.8586	0.8829
Foreground	1.0000	0.9926	0.8873	0.5441	0.8704	0.8522	0.7760	0.8461
- Ground truth	0.9976	0.9949	0.9244	0.5819	0.8527	0.8736	0.8118	0.8624

Percentage of correct keypoints (PCK)

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Mean
Baseline	0.9956	0.9891	0.9148	0.7031	0.8690	0.9017	0.8646	0.8911
Clothing	1.0000	0.9934	0.9127	0.6965	0.9345	0.9148	0.8843	0.9052
- Ground truth	1.0000	0.9978	0.9323	0.7467	0.9192	0.9367	0.9017	0.9192
Foreground	1.0000	0.9934	0.9148	0.6878	0.9127	0.8996	0.8450	0.8933
- Ground truth	0.9978	0.9956	0.9389	0.7183	0.9105	0.9214	0.8734	0.9080

- **Clothing**: using a histogram of clothing in addition to HOG feature.
- **Foreground**: using a histogram of figure-ground segmentation in addition to HOG feature.

Here we concatenate all features into a single descriptor and learn a max-margin linear model (Yang and Ramanan, 2011). All models use 5 mixture components in this experiment. We compute the foreground model simply by treating non-background regions in clothing parsing as foreground. Comparing the clothing model and the foreground model reveals how *semantic* information helps pose estimation compared to non-semantic segmentation. We use the Fashionista dataset in this comparison, with the same train-test split described in Section 3.3.1.

Table 3.5 summarizes average precision of keypoints (APK) and percentage of correct keypoints (PCK) using the Fashionista dataset. For clothing and foreground cases, we

also compare the performance when we use the ground-truth pixel annotation, which serves as an upper bound on performance for each model given a perfect segmentation. Clearly, introducing clothing parsing improves the quality of pose estimation. Furthermore, the improvement of the clothing model over the foreground model indicates that the contribution is coming from the inclusion of *semantic* parsing rather than from a simple figure-ground segmentation.

From these performance numbers we can see that clothing parsing is particularly effective for improving localization of end parts of the body, such as the wrist. Perhaps this is due to items specific to certain body parts, such as *skin* for wrist and *shoes* for ankle. Note that a figure-ground segmentation cannot provide such semantic context. This result gives an important insight into the pose estimation problem, since improving estimation quality for such end parts is the key challenge in pose estimation, while state-of-the-art methods can already accurately locate major parts such as head or torso. We believe that semantic parsing provides a strong context to improve localization of minor parts that often suffer from part articulation.

3.5 Summary and Discussion

In this chapter, we described novel approaches for clothing parsing. Our experimental results indicate that our data-driven approach is particularly beneficial in the detection scenario, where we need to both identify and localize clothing items without any prior knowledge about depicted clothing items. We also empirically show that pose estimation can benefit from the semantic information provided by clothing parsing.

CHAPTER 4: CLOTHING STYLE RECOGNITION

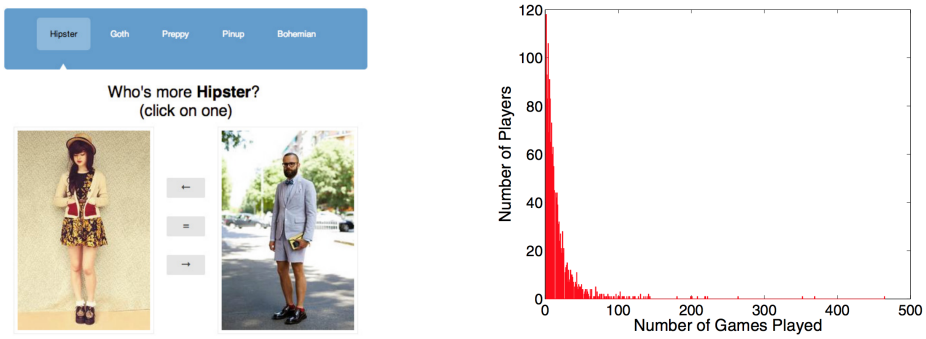
The clothing we wear and our identities are closely tied, revealing to the world clues about our wealth, occupation, and socio-identity. In this chapter, we examine questions related to what our clothing reveals about our personal style. We first design an online competitive Style Rating Game called *Hipster Wars* to crowd source reliable human judgments of style. We use this game to collect a new dataset of clothing outfits with associated style ratings for 5 style categories: hipster, bohemian, pinup, preppy, and goth. Next, we train models for between-class and within-class classification of styles. Finally, we explore methods to identify clothing elements that are generally discriminative for a style, and methods for identifying items in a particular outfit that may indicate a style.

4.1 Hipster Wars: Style Dataset and Rating Game

To study style prediction we first collect a new dataset depicting different fashion styles (Section 4.1.1). We then design a crowd-sourcing game called *Hipster Wars* to elicit style ratings for the images in our dataset (Section 4.1.2).

4.1.1 Data Collection

We collect a new dataset of images depicting five fashion styles, *Bohemian, Goth, Hipster, Pinup,* and *Preppy.* To construct our initial seed corpus, we query Google Image



(a) Snapshot of the game

(b) Distribution of games played

Figure 4.1: Left shows an example game for the hipster category on Hipster Wars. Users click on whichever image is more hipster or click "=" for a tie. Right shows the number of games played per player.

Search using each style name and download top ranked images. We then use Google's "Find Visually Similar Images" feature to retrieve thousands of additional visually similar images to our seed set and manually select images with good quality, full body outfit shots. We repeat this process with expanded search terms, e.g. "pinup clothing" or "pinup dress", to collect 1893 images in total. The images exhibit the styles to varying degrees. Average images for each style category is shown in Figure 4.2.

4.1.2 Rating Game

We want to rate the images in each style category according to how strongly they depict the associated style. As we show in section 4.1.5, simply asking people to rate individual images directly can produce unstable results because each person may have a different internal scale for ratings. Therefore, we develop an online game to collectively crowd-source ratings for all images within each style category. A snapshot of the game is shown in Figure 4.1. Our game was released to great success, attracting over

Figure 4.2: Average image for each style category

1700 users who provided over 30,000 votes at the time of analysis and the number of votes is growing every day. Our players are also scattered around the globe. Figure 4.3 presents the percentage of players from each continent. While the majority of players are from Americas (74.57%), players from Europe contribute significantly to the game as well (17.53%). Asia, Oceania and Africa constitute a smaller fraction (7.9%) of players together.

Our game is designed as a tournament where a user is presented with a pair of images from one of the style categories and asked to click on whichever image more strongly depicts the solicited style, or to select "Tie" if the images equally depict the style. For example, for images in the hipster category the user would be asked "Who's more hipster?" After each pair of images, the user is provided with feedback related to the winning and losing statistics of the pair from previous rounds of the tournament.

Because we cannot afford to gather comparisons for all pairs of images in our dataset, we make use of the TrueSkill algorithm (Herbrich et al., 2007). This algorithm iteratively

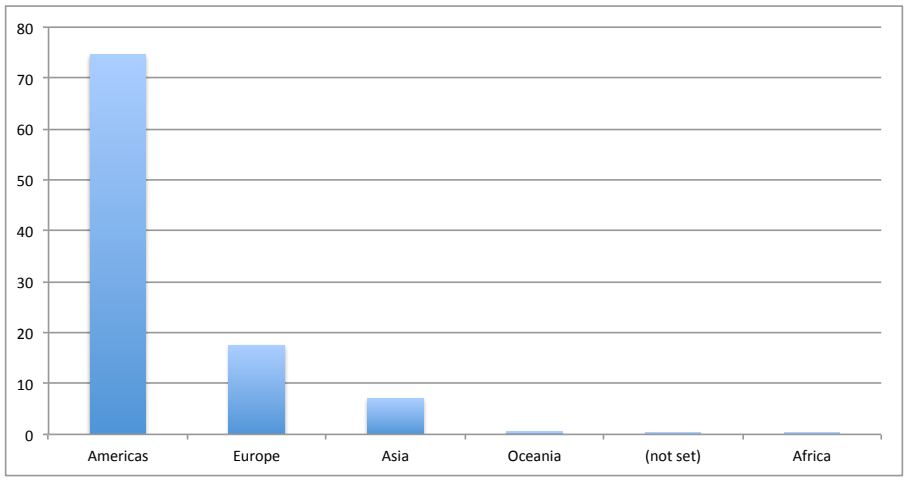


Figure 4.3: Distribution of players across the continents. While the majority of players are from Americas (74.57%), players from Europe contribute significantly to the game as well (17.53%). Asia, Oceania and Africa constitute a smaller fraction (7.9%) of players together.

determines which pair of images to compare in each tournament, and based on user input successively updates the ratings of images in our dataset. TrueSkill is a popular ranking system, originally developed to pair users in XBox Live. Though it was originally developed to pair players and determine their gaming skill levels, it is a general model that can be applied in any competitive game. Here we apply it to images.

There are several reasons we choose the TrueSkill algorithm. For each tournament the algorithm pairs up images with similar estimated ratings. Therefore, over time we are able to focus on finer-grained distinctions between images and minimize the number of comparisons we need to make to estimate the true image ratings. Additionally, the algorithm is online (as opposed to batch). Users can upload their own photos and merge them seamlessly into the tournaments even after the game has started. The algorithm is also efficient, allowing us to update rankings in real-time after each tournament even when many users are playing at once. It also explicitly allows for ties and models uncertainty

in ratings. Finally, TrueSkill converges quickly, reducing the number of games necessary to compute ratings for images.

4.1.3 Game Details

Each image is associated with a *skill* variable, s, representing how strongly the image represents the associated style. Our goal is to determine this skill level for each image in the dataset. An image's skill is modeled by a Gaussian distribution with mean, μ, and variance, σ^2, where $s \sim \mathcal{N}(s; \mu, \sigma^2)$. As different users play the tournaments there may be variations in how the styles are perceived, this is modeled with another variable, p, a Gaussian distribution around skill level, $p \sim \mathcal{N}(p; s, \beta^2)$.

Updating after Win/Loss: After each tournament is played, if the tournament does not result in a tie, we update the skill estimates for the winning player as:

$$\mu_{\text{winner}} \leftarrow \mu_{\text{winner}} + \frac{\sigma^2_{\text{winner}}}{c}.\mathbb{V}\left(\frac{(\mu_{\text{winner}} - \mu_{\text{loser}})}{c}, \frac{\epsilon}{c}\right) \tag{4.1}$$

$$\sigma^2_{\text{winner}} \leftarrow \sigma^2_{\text{winner}}.\left(1 - \frac{\sigma^2_{\text{winner}}}{c^2}.\mathbb{W}\left(\frac{(\mu_{\text{winner}} - \mu_{\text{loser}})}{c}, \frac{\epsilon}{c}\right)\right) \tag{4.2}$$

$$\tag{4.3}$$

Where:

$$\mathbb{V}(a, b) = \frac{\mathcal{G}_{0,1}(a-b)}{\Phi_{0,1}(a-b)} \tag{4.4}$$

$$\mathbb{W}(a, b) = \mathbb{V}(a, b) \cdot (\mathbb{V}(a, b) + a - b) \tag{4.5}$$

$$c^2 = 2\beta^2 + \sigma^2_{\text{winner}} + \sigma^2_{\text{loser}} \tag{4.6}$$

Where $\mathcal{G}_{0,1}$ and $\Phi_{0,1}$ are the PDF and CDF of normal distributions with zero mean and unit variance. The intuition behind these updates is that if the win was expected, i.e. the difference between skills of the winner image and the losing image was large relative to the total uncertainty, c, then the update on image skill estimates will be small. However, if the outcome of the tournament was surprising, the updates will be larger. Similar update rules are applied for the loser of the tournament.

Updating after Tie: If a tournament is tied, $\mathbb{V}$ and $\mathbb{W}$ are computed as:

$$\mathbb{V}(a,b) = \frac{\mathcal{G}_{0,1}(-b-a) - \mathcal{G}_{0,1}(b-a)}{\Phi_{0,1}(b-a) - \Phi_{0,1}(-b-a)} \tag{4.7}$$

$$\mathbb{W}(a,b) = \mathbb{V}^2(a,b) + \frac{(b-a)\cdot\mathcal{G}_{0,1}(b-a) + (a+b)\cdot\mathcal{G}_{0,1}(a+b)}{\Phi_{0,1}(b-a) - \Phi_{0,1}(-b-a)} \tag{4.8}$$

Similar intuition applies here. If both images already had similar skill levels there are not significant updates on beliefs for either image. If the result was more surprising, updates are more significant.

Selecting Pairs: For each tournament we must select a pair of images to play against each other. We would like to optimize two things: every image should be played enough times to reliably determine its rating, and we would like to pair up images with similar ratings estimates in order to produce fine-grained estimates of their ratings. Therefore, to select pairs we first choose the least played image from the dataset and then we choose as its pair, the image with highest probability of creating a draw with that image (to maximize the informativeness of each tournament) which following (Herbrich et al., 2007)

is computed as:

$$q_{\text{draw}}(\beta^2, \mu_i, \mu_j, \sigma_i, \sigma_j) \equiv \sqrt{\frac{2\beta^2}{2\beta^2 + \sigma_i^2 + \sigma_j^2}} \cdot \exp\left(-\frac{(\mu_i - \mu_j)^2}{2(2\beta^2 + \sigma_i^2 + \sigma_j^2)}\right) \tag{4.9}$$

Implementation details: We design our game such that image scores fall into the range $[0, 50]$. Our ranking system initializes ratings for all images with $\mu = 25$ and uncertainty $\sigma = \frac{25}{3}$. Value for ϵ, the draw margin, is calculated based on a 10% chance of draw assumed in every game and default value for β is set to $\frac{25}{6}$. Finally the 'true skill' of each image is given by $\mu - 3\sigma$, a conservative estimate which ensures images with high skill means and least uncertainties will be placed on the top.

4.1.4 Game Results

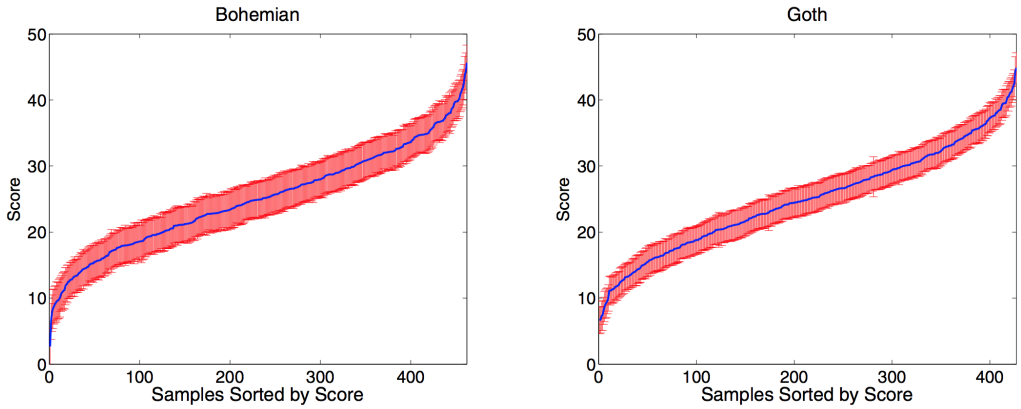
Our style rating game was played by 1702 users for over 30,000 tournaments. On average users played about 18.5 tournaments, indicating reasonable engagement. Some users played hundreds of tournaments, with a max of 465. The distribution of number of games played per user is shown in Figure 4.1. Scores sorted by their mean along with their uncertainty for two sample categories are shown in Figure 4.5.

This produces very reasonable ratings for each style category. Top and bottom rated images for each style are shown in Figure 4.4. Top rated images tend to depict very strong indications of the associated style while images rated toward the bottom of the set depict the style with much less strength.

Figure 4.4: Example results from our style rating game, *Hipster Wars*. Top and bottom rated for each style category.

4.1.5 Pairwise vs. Individual Ratings

In order to evaluate the effectiveness of pairwise comparisons on Hipster Wars over a standard approach of rating style independently for each image, we used Amazon Mechanical Turk to conduct the following experiment. For each of the style categories, we divided the range of skills obtained from Hipster Wars into 10 equal size intervals which we call skill levels (1 :lowest, 10 :highest) and picked a subset of 100 images



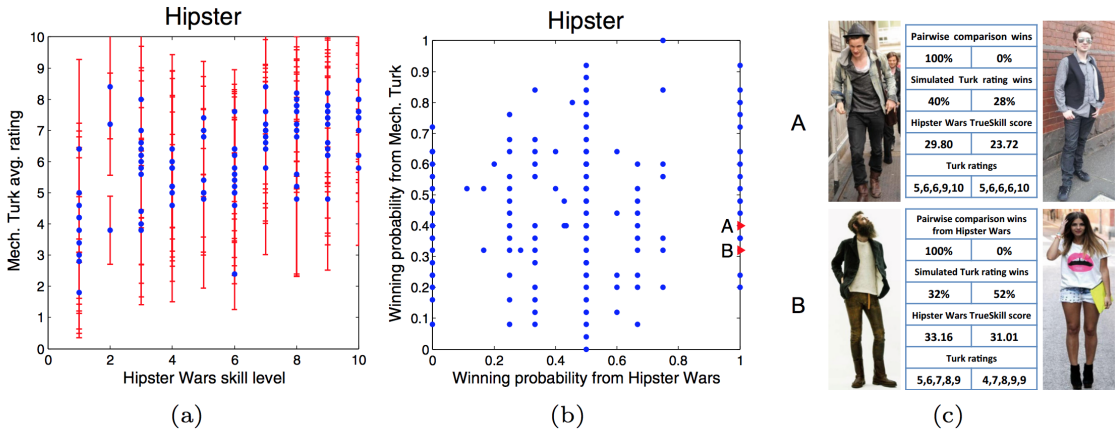
(a) Bohemian (b) Goth

Figure 4.5: Style scores computed by our Style Rating Game, showing means and uncertainties for images sorted from smallest to largest mean.

distributed uniformly over the intervals. For each of the images, we asked 5 individuals (on Mechanical Turk) to rate the degree of a particular style category. Example ratings from all skill levels were provided. Figure 4.6 shows a scatter plot of average ratings from Mechanical Turk vs the skill level estimated by Hipster Wars. It also shows the average ratings vs the actual win percentage of games on Hipster Wars. In general the ratings are much noisier than either the direct pairwise comparisons or the skill level estimated by Hipster Wars. Figure 4.6 shows example pairs where this discrepancy is very large. These results indicate that the pairwise comparison approach can provide more stable and useful ratings for subtle cues like style.

4.2 Style Representation

We represent the style of outfits using a number of visual descriptors found to be useful for clothing recognition tasks (Yamaguchi et al., 2013), including descriptors related to color, texture, and shape. In particular, we calculate a vector of the following



(a) (b) (c)

Figure 4.6: Hipster Wars Pairwise ratings vs. individual ratings from Amazon Mech. Turk.

features at each pixel within a patch centered around the pixel: a) RGB color value, b) Lab color value, c) MR8 texture response (Varma and Zisserman, 2005) (to encode local patterns) d) HOG descriptor (Dalal and Triggs, 2005) (to measure local object shape), e) Distance from image border, f) Probability of pixels belonging to skin and hair categories (Yamaguchi et al., 2013). We form the Style Descriptor, illustrated in Figure 4.7, by accumulating these features following (Yamaguchi et al., 2013), but without dimensionality reduction to capture the details of clothing appearance. The exact procedure is as following: 1) We first estimate body pose (Yang and Ramanan, 2011). 2) For each of the 24 estimated body part keypoints, we extract an image patch of size 32×32 pixels surrounding the keypoint. 3) We split each image patch into 4×4 cells and mean-std pooling of the features described above are computed. 4) We concatenate all pooled features over all 24 patches, for a total of $39,168$ dimensions.

We compared the classification performance of Style Descriptor against two other global visual descriptors computed on the detected bounding box by pose estimator:

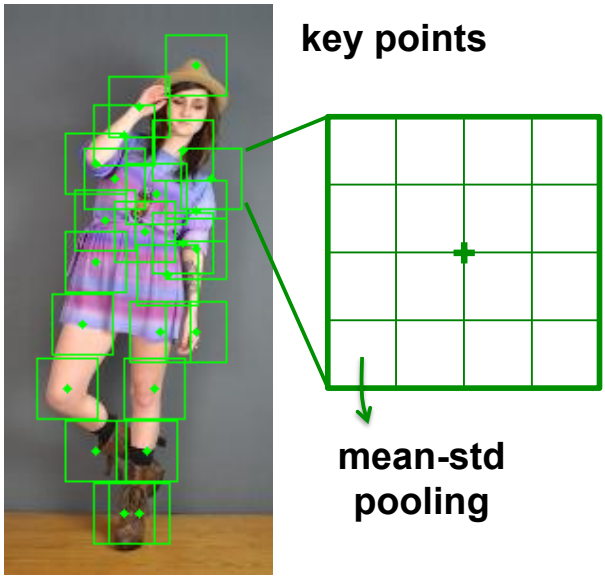
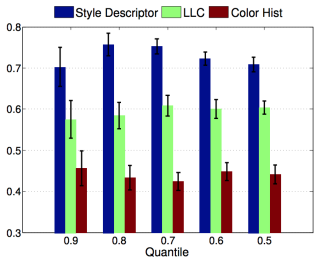


Figure 4.7: Style descriptor

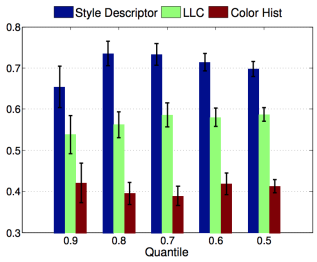
LLC encoding (Wang et al., 2010) of local SIFT (Lowe, 1999) descriptors and color histogram. For LLC we extract SIFT features on a dense grid over the image and use LLC coding to transform each local descriptor into a sparse code and apply a multi-scale spatial pyramid (1×1, 2×2, 4×4) (Lazebnik et al., 2006) max-pooling to obtain the final 43008-dimensional representation. Color histogram features were constructed by quantizing the R,G,B channels into 16 bins each, giving a final 4096-dimensional histogram for each image.

4.3 Predicting Clothing Styles

We consider two different style recognition tasks: Between-class classification - Classifying outfits into one of the five fashion styles (Sec 4.3.1). Within-class classification - differentiating between high and low rated images for each style (Sec 4.3.2). For each of these tasks, we compare Style Descriptor versus the other global descriptors which we



(a) Accuracy



(b) Average F-1

	Bohemian	Goth	Hipster	Pinup	Preppy
Bohemian	1904	99	135	73	99
Goth	83	1763	168	21	75
Hipster	188	303	1012	39	338
Pinup	137	59	27	668	47
Preppy	228	233	281	107	1313

(c) Confusion matrix of 5 way clothing style classification at $\delta = 0.5$

Figure 4.8: Between-class classification results showing accuracy and average f-1 scores for each style are computed over random 100 folds for the classification of the top δ% rated images. Error bars are 95% confidence intervals from statistical bootstrapping. Confusion matrix of shows the results for 5 way clothing style classification at $\delta = 0.5$

considered as baseline. In all classification experiments we use a linear kernel SVM using the liblinear package (Fan et al., 2008c).

4.3.1 Between-class Classification

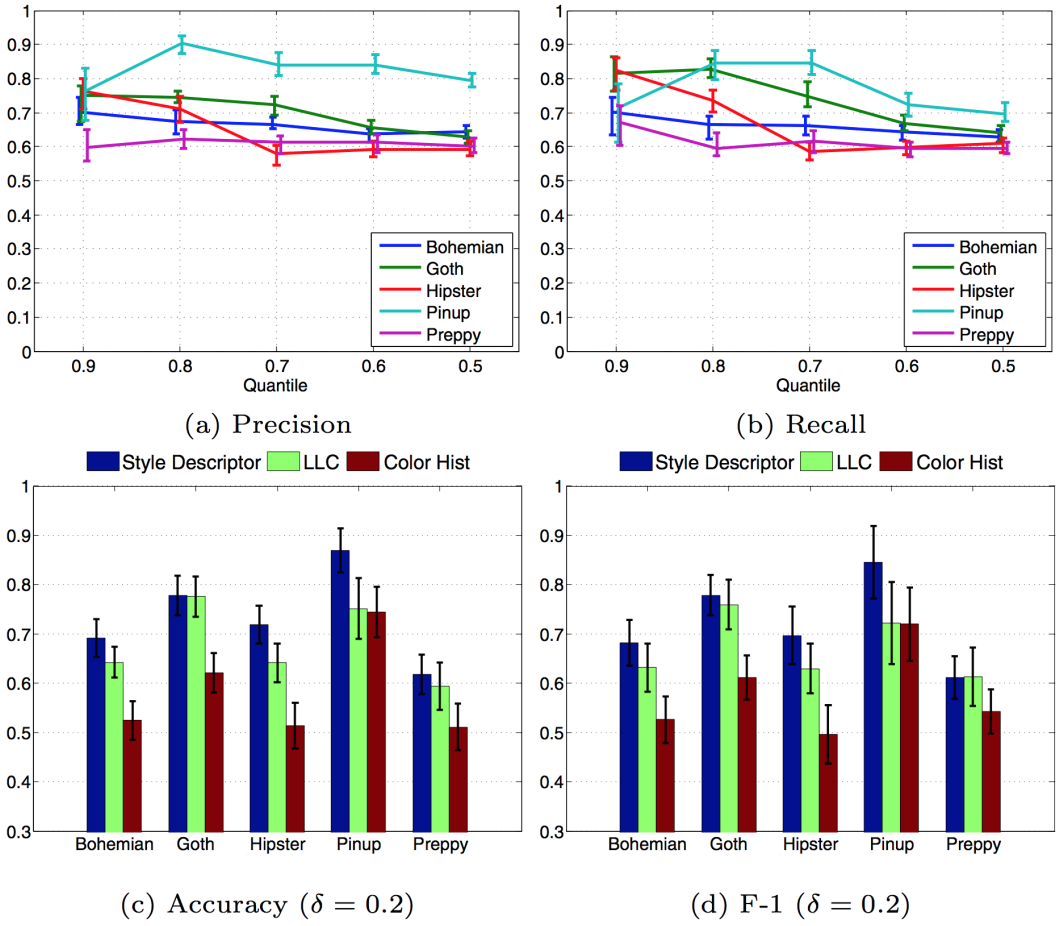
We consider classifying images as one of five styles. Results examine how performance varies for different splits of the data, defining a parameter δ which determines what percentage of the data is used in classification. We vary values of δ from 0.1 to 0.5 where $\delta = 0.1$ represents a classification task between the top rated 10% of images from each style (using the ratings computed in Sec 4.1.2). We use a $9:1$ train to test ratio, and repeat the train-test process 100 times. The results of our between-class classification are shown in Figure 4.8. Performance is good, varying slowly with δ, and the pattern of confusions is reasonable.

Figure 4.9: Example results of within-classification task with $\delta = 0.5$. Top and bottom predictions for each style category are shown.

4.3.2 Within-class Classification

Our next style recognition tasks considers classification between top rated and bottom rated examples for each style independently. Here we learn one linear SVM model for each style. The variable $\delta = 10\% \ldots 50\%$ determines the percentage of top and bottom ranked images considered. For example, $\delta = 0.1$ means the top rated 10% of images are used as positives and the bottom rated 10% of samples as negatives. We repeat the

experiments for 100 random folds with a $9:1$ train to test ratio. In each experiment, C, is determined using 5 fold cross-validation.



(a) Precision (b) Recall

(c) Accuracy ($\delta = 0.2$) (d) F-1 ($\delta = 0.2$)

Figure 4.10: Within-Class classification results averaged for each style computed over random 100 folds balanced classification of the top and bottom δ% quartiles. Error bars are 95% confidence intervals from statistical bootstrapping.

Results are reported in Figure 4.10. We observe that when δ is small we generally have better performance than for larger δ, probably because the classification task becomes more challenging as we add less extreme examples of each style. Additionally, we find best performance on the pinup category. Performance on the goth category comes in second. For the hipster category, we do quite well at differentiating between extremely

strong or weak examples, but performance drops off quickly as δ increases. Example predictions for each style are shown in Figure 4.9.

4.4 Discovering the Elements of Styles

In this section, we are interested in two different questions: 1) what elements of style contribute to people in general being a hipster (or goth or preppy, etc), and 2) for a particular photo of a person, what elements of their outfit indicate that they are a hipster (or goth or preppy, etc)?

4.4.1 General Style Indicators

We would like to determine which garment items are most indicative of each style in general. For this, we compute clothing segmentation on all images of each style, and obtain the percentage of each predicted garment item present. Figure 4.11 shows the percentage of pixels occupied by each garment item across images of each style. Based on this automatic analysis, we can make some interesting observations using our clothing recognition predictions. For example, we find that pinups and bohemians tend to wear dresses whereas hipsters and preppies do not. Goths fall somewhere in between. Pinups also tend to display a lot of skin while this is less true for goths. Hipsters and preppies wear the most jeans and pants. Preppies tend to wear more blazers while goths and hipsters wear the most boots.

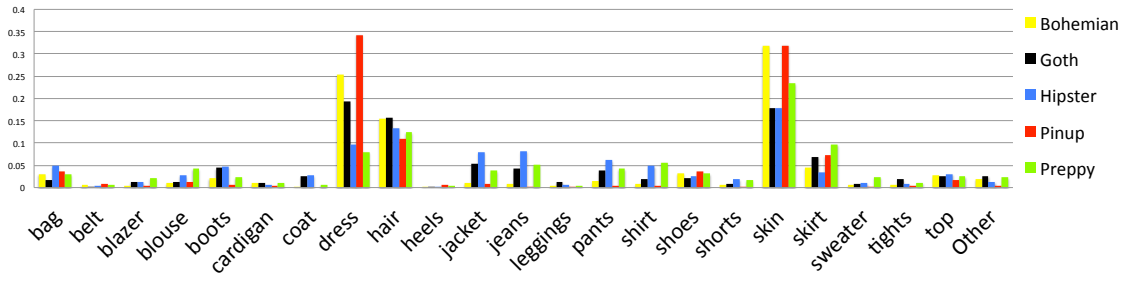


Figure 4.11: Clothing items across styles.

4.4.2 Style Indicators for Individuals

Our second approach is a bit more complex. In this model we make use of our models trained on Style Descriptors. We essentially transfer predictions from the Style Descriptor to the underlying parse while making use of computed priors on which garment items are most likely for each style.

Discriminative part discovery: Suppose we have a set of image features $\mathbf{x}_i$ from each part i that we locate from a pose estimator. Then our style prediction model can be described by a linear model:

$$y = \sum_{i \in \text{parts}} \mathbf{w}_i^{\mathrm{T}} \mathbf{x}_i + b, \tag{4.10}$$

where y is a decision value of the prediction, $\mathbf{w}_i$ is model parameters corresponding to part i, and b is a bias parameter.

In this chapter, we specifically view the individual term $\mathbf{w}_i \mathbf{x}_i$ as a distance from the decision boundary for part i in the classification, and utilize the weights to *localize* where discriminative parts are located in the input image. This interpretation is possible when

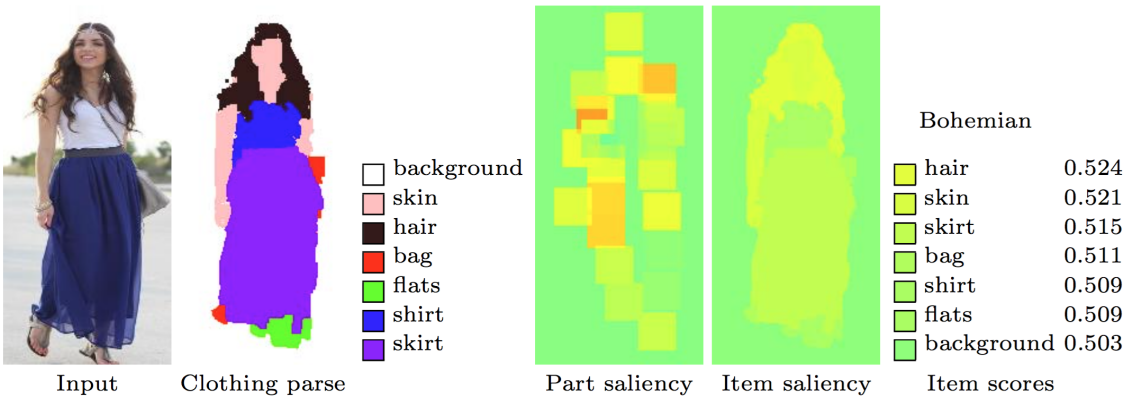


Figure 4.12: From parts to items. We compute contributions of each part, and project them in image coordinates (Part saliency). Then, using clothing parse, we compute the scores of items. When the score is above 0.5, the associated item indicates a positive influence on the queried style. Note that the scores only show the relative strength of style-indication among items in the picture.

the input to the linear model is uniformly interpretable, i.e., same feature from different locations. Also to guarantee the equivalence of parts interpretation, we normalize the part features $\mathbf{x}_i$ to have zero-mean and uniform standard deviation in training data.

To calculate the score of the part i, we apply a sigmoid function on the decision value and get probabilities of a style given a single part:

$$p_i \equiv \frac{1}{1 + \exp\left(-\mathbf{w}_i^{\mathrm{T}} \mathbf{x}_i\right)}. \tag{4.11}$$

Learning is done in the same manner as within-class style classification, using L2-regularized logistic regression.

From parts to items: Part scores tell us which locations in the outfit are affecting style prediction. However, to convert these to an interpretable prediction, we map predicted garments back to garments predicted in the original parse. This produces a more semantic output, e.g. "She looks like a hipster because of her hat." To map parts

to garments in the parse, we first compute a *saliency map* of parts; At each keypoint, we project the part score $p(\mathbf{x}_i)$ to all pixels in the patch location. Articulated parts get the average score from all parts. Areas outside of any patch are set to 1/2 (i.e., decision boundary). Using the computed clothing segmentation (Yamaguchi et al., 2013), we compute the average score of each garment item from the *saliency map*. This produces, for each item k in the clothing parse of an image, a score p_k that we can use to predict items that strongly indicate a style. Figure 4.12 depicts this process.

Prior filtering: The result of part-based factorization can still look noisy due to errors in pose estimation and clothing parsing. Therefore, we smooth our predictions with a prior on which garment items we expect to be associated with each style.

Our prior is constructed by building a linear classifier based on the area of each clothing item that we obtain from the clothing segmentation (Yamaguchi et al., 2013). Denoting the log pixel-count of item k by x_k, we express the prior model by a linear function: $y = \sum_k w_k x_k + b$, where y is the decision value of style classification, and w_k and b are model parameters. Using the same idea from the previous subsections, we compute the score of each item by: $q_k \equiv \frac{1}{1+\exp(-w_k x_k)}$.

Once we compute the part-based score p_k and the prior score q_k, we merge them into the final indicator score r_k for garment-item k:

$$r_k \equiv \lambda_1 p_k + \lambda_2 \left[\frac{\sigma_p}{\sigma_q} \left(q_k - \frac{1}{2} \right) + \frac{1}{2} \right], \tag{4.12}$$

where λ_1 and λ_2, are weights given to each score, σ_p and σ_q are standard deviations of

Figure 4.13: Example predicted style indicators for individuals.

p_k and q_k at each image. The intuition here is that we assume both p_k and q_k follow a normal distribution with mean at 0.5. We adjust the shape of q_k distribution to that of p_k in the second term. Then, we use λ's to mix two scores and produce the final result. We set λ's to cross-validation accuracies of classification during training normalized to sum to a unit, so that the resulting score reflects the accuracy of style prediction.

4.4.3 Analysis of Style Indicators for Individuals

Figure 4.13 shows examples of discovered style indicators for individuals. Predicted elements for each outfit are ordered by indicator scores. We find that our method captures the most important garment-items well such as shirt for preppy styles, graphic t-shirts for hipsters, or dresses for pinups.

We also attempted to quantitatively verify the results using crowdsourcing. We obtained the "ground truth" by asking workers to vote on which element they think is making a certain style. However, the naive application of this approach resulted in a number of problems; 1) workers tend to just vote on all visible items in the picture, 2)

Method	Bohemian	Goth	Hipster	Pinup	Preppy
Random	0.357	0.258	0.171	0.427	0.232
Our method	0.379	0.282	0.154	0.454	0.241

Table 4.1: Ratio of images that include the top choice from crowds in the first 5 elements of our discovery method.

small items are ignored, 3) workers mark different items with a different name (e.g., shoes vs. flats) and 4) different workers are not consistent due to the great subjectivity in the question. We show in Table 4.1 the ratio of images from our discovery that included the worker's top choice. Our method achieved slightly better result than the random ordering. However, we note that the "ground truth" in this evaluation does not necessarily constitute a good measurement for benchmarking, leaving open the question of how to "ground truth" annotation for such subtle socially-defined signals.

4.5 Summary and Discussion

We have designed a new game for gathering human judgments of style ratings and have used this game to collect a new dataset of rated style images. We have explored recognizing and estimating the degree of fashion styles. We have also begun efforts to recognize which elements of outfits indicate styles generally and which items in a particular outfit indicate a style. Results indicate that it is possible to determine whether you are a hipster and that it may even be possible to determine why you are a hipster!

CHAPTER 5: CLOTHING RETRIEVAL

In this chapter, we look at a new task related to online shopping, the exact street to shop problem. Given a real world photo of a clothing item, e.g. taken on the street, the goal of this task is to find that clothing item in an online shop. This is extremely challenging due to differences between depictions of clothing in real world settings versus the clean simplicity of online shopping images. For example, clothing will be worn on a person in street photos, whereas in online shops clothing items may also be portrayed in isolation or on mannequins. Shop images are professionally photographed, with cleaner backgrounds, better lighting, and more distinctive poses than may be found in real world consumer captured photos of garments. To deal with these challenges we introduce a deep learning based methodology to learn a similarity measure between street and shop photos.

The street to shop problem has been explored in some previous work (Liu et al., 2012b). There the goal was to find *similar* clothing items in online shops, where performance is measured according to how well retrieved images match a fixed set of attributes, e.g. color, length, material, that have been hand-labeled on the query clothing items. However, finding a similar garment item may not always correspond to what a shopper desires. Often when a shopper wants to find an item online, they want to find *exactly* that item to purchase.

Therefore, we define a new task, *Exact Street to Shop*, where our goal is for a query

Figure 5.1: Our task is to find the exact clothing item, here a dress, shown in the query. Only the first dress, in the green rectangle would be considered correct. This is different from previous work e.g. (Liu et al., 2012b) that considers whether retrieved items have similar high-level features. Under that, more relaxed, evaluation all of the dresses shown are correct. (For this query, our similarity learning ranked the correct match first.)

street garment item, to find exactly the same garment in online shopping images. Perhaps surprisingly, with our learned similarity metric, we can actually do this sometimes. Although clothing items such as belts and shoes are more difficult, we can find an exactly matching dress or a top (out of tens of thousands of possibilities) in a shortlist of 20 items in about a third of trials!

To study exact street to shop at large scale, we collected and labeled a dataset of 20,357 images of clothing worn by people in the real world, and 404,683 images of clothing from shopping websites. The dataset contains 39,479 pairs of exactly matching items worn in street photos and shown in shop images.

Our approach attacks the exact street to shop problem using multiple methods. We first look at how well standard deep feature representations on whole images or on object proposals can perform on this retrieval task. Then, we explore methods to learn similarity metrics between street and shop item photos. These similarities are learned between existing deep feature representations extracted from images. To examine the difficulty

of the exact street to shop task, we also provide several human experiments, evaluating when and where exact item retrieval is feasible, and to evaluate our retrieval results.

In summary, our contributions are:

- Introduction of the exact street to shop task and collection of a new dataset, the Exact Street2Shop Dataset, for evaluating performance on this task.
- Evaluation of deep learning and CNN-based similarity learning methods for the exact street to shop retrieval task.
- Human evaluations of the exact street to shop task and of our results.

The rest of this chapter is organized as follows. First we describe our new dataset (Sec 5.1) and approaches (Sec 5.2) for the Exact Street to Shop task. Finally we provide experimental results and discuss them in (Sec 5.3).

5.1 Dataset

We collect a new dataset, the *Exact Street2Shop Dataset*, to enable retrieval applications between real world photos and online shopping images of clothing items. This dataset contains two types of images: *a) street photos*, which are real-world photographs of people wearing clothing items, captured in everyday uncontrolled settings, and *b) shop photos*, which are photographs of clothing items from online clothing stores, worn by people, mannequins, or in isolation, and captured by professionals in more controlled settings. Particular clothing items that occur in both the street and shop photo col-

Figure 5.2: Example *street outfit photos*, including large variations in pose, camera angle, composition and quality.

lections form our *exact street-to-shop pairs* for evaluating retrieval algorithms. In the following sections we describe our dataset and annotation process in detail.

5.1.1 Image Collection

In this section, we describe our data collection of street photos (Sec. 5.1.1), shop photos (Sec. 5.1.1), and correspondence between street and shop items (Sec. 5.1.1).

Figure 5.3: Example *shop photos*, displaying a wide range of apparel photography techniques.

Street Photos

To create a useful dataset for evaluating clothing retrieval algorithms, we would like to collect street photographs of clothing items for which we know the correspondence to the same clothing items in online shops. There are a number of social communities focused on fashion, such as Chictopia and various fashion blogs, where people post photographs of themselves wearing clothing along with links to purchase the items they are wearing. However, these links are often outdated or pointing to items that are similar but not identical to the items being worn.

Instead, to gather corresponding street-shop item pairs for a wide range of different

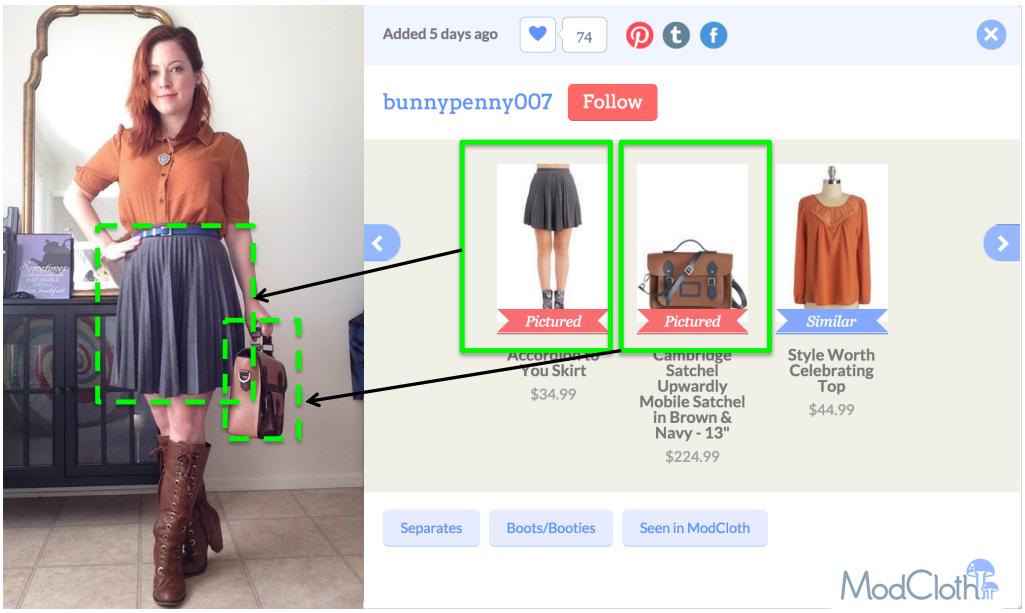


Figure 5.4: An example snapshot of ModCloth data. We collect the bounding box location of pictured items on street photos using Amazon Mechanical Turk.

people and environments, we make use of style galleries from ModCloth[1]. ModCloth is a large online retail store specializing in vintage fashions that sells clothes from a wide variety of brands. Style galleries contains user-contributed outfit posts, in which people upload photos of themselves wearing ModCloth clothing items and provide shopping links to the exact items they are wearing. Figure 5.4 shows an example snapshot from ModCloth website.

We collect 20,357 style gallery outfit posts, spanning user-contributed photos (example outfit photos are shown in Figure 5.2). Each outfit post consists of a *street photo* that depicts at least one of the clothing items offered on the ModCloth website. These photographs aim to showcase how one would style an outfit or to help others decide whether they want to purchase an item. There are large variations in the quality of

[1]http://www.ModCloth.com

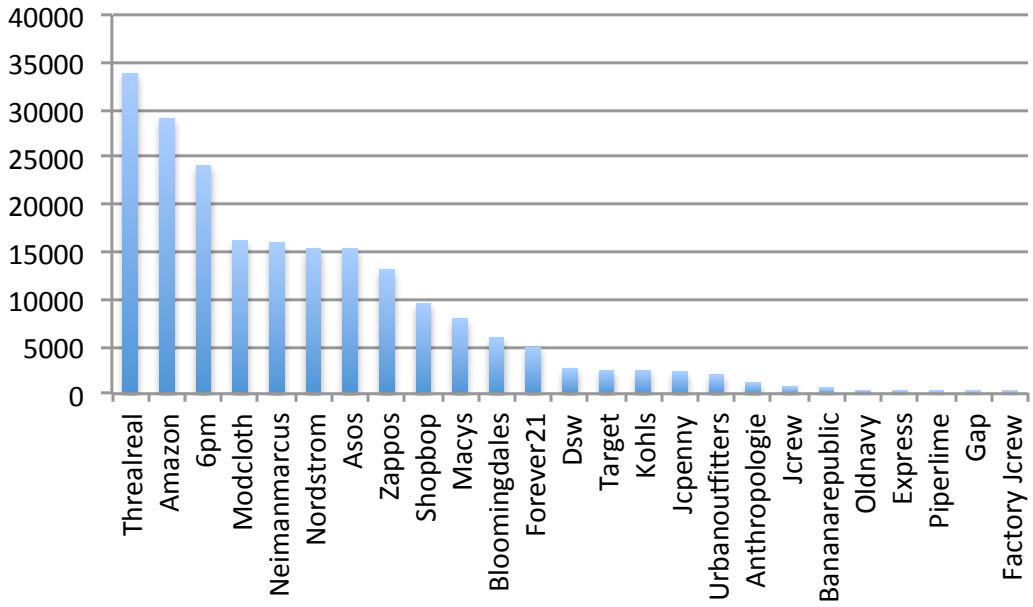
the contributed photographs, lighting, indoor vs outdoor environments, body shapes and sizes of the people wearing the clothing, depicted pose, camera viewing angle, and a huge amount of occlusion due to layering of items in outfits. In addition, a photo may depict a head-to-toe shot, or several partial-body shots. These characteristics reflect the extreme challenges and variations that we expect to find for clothing retrieval in real-world applications.

Shop Photos

We have collected 404,683 shop photos from 25 different online clothing retailers. These photos depict 204,795 distinct clothing items (each clothing item may be associated with multiple photographs showing different views of the item). Moreover, when available, the title and a detailed description of the item is extracted from the product's webpage. We collect 11 different broad categories of clothing items, ranging from small items such as belts and eyewear, to medium size items such as hats or footwear, to larger items such as dresses, skirts, or pants.

Shop photos differ drastically from street photos in that they are professionally produced and tend to be high resolution with clean backgrounds, captured under nice lighting with highly controlled conditions. Different brands have different styles of fashion photography, ranging from more basic depictions to professional models. In addition, while some shop photos display a clothing product on a full or partial mannequin, or on a live model, others depict clothing items folded or lying flat on a surface. Shop images also often include close-up shots that display clothing details such as fabric texture or pattern.

Figure 5.5: Distribution of collected items across shopping sites.



Altogether, these qualities make our shop dataset highly diverse. Example shop photos are shown in Figure 5.3 and the distribution of collected items across shopping websites is displayed in Figure 5.5.

Street-to-Shop Pairs

Each street photo in our dataset is associated with two types of links to shop clothing items: the first set contains links to products that exactly match one of the pictured items in a street photo, while links in the second set indicate items that are only similar to a street item. These links are user provided, but we have manually verified that the links are highly accurate. We make use of only the exact matching items to create our street-to-shop pairs, but we also release the similar matching pairs in our public dataset for evaluation of other types of image retrieval algorithms. In total, there are 39,479

exact matching street-to-shop item pairs.

5.1.2 Image Annotation

For the retrieval task we assume that we know two things about a query street image: what category of item we are looking for, and where in the image the item is located. In a real-world retrieval application, this information could easily be provided by a motivated user through input of a bounding box around the item of interest and selection of a high level category, e.g. skirt. Therefore, we pre-annotate our dataset in two ways. First, we automatically associate a high level garment category with each item in the dataset (Sec 5.1.2). Then, we collect bounding boxes for each street query item (Sec 5.1.2). The latter task is performed using Amazon's Mechanical Turk service.

Category Labeling

Our category labeling relies on the meta data associated with collected items. For every item, its product category on the website, web url, title and description are collected if available. We then create a mapping between product keywords and our final list of 11 garment categories: bags, belts, dresses, eyewear, footwear, hats, leggings, outerwear, pants, skirts and tops. Finally, we label every item with the category associated with the keywords found in its meta data. A sample of the mappings from keywords to garment categories are shown in Table 5.1.

Instance Annotation

For every street-to-shop pair, we collect bounding boxes for the item of interest in the street photograph. Note, street photos depict entire outfits, making it necessary to present both the street photo and the corresponding shop photos to the turker during this annotation process. In particular, we show the workers a street photo and the corresponding shop item photos and ask the turker to annotate instances of the shop item in the street photograph. Annotators draw a tight bounding box around each instance of the shop item in the provided street photo. To aid this process turkers are provided with example annotations for each item type, including items with multiple objects, e.g. shoes.

5.2 Approaches

We implement several different retrieval methods for the street to shop matching problem. Input to our methods are a street query image, the category of item of interest, and a bounding box around the item in the query image. On the shop side, since there are a large number of images, we do not assume any hand-labeled localization of items, instead letting the algorithm rely on features computed on the entire image or on object proposal regions.

We first describe two baseline retrieval methods for the street to shop task using deep learning features as descriptors for matching to an entire shop image (Sec 5.2.1) or to object proposal regions within the shop images (Sec 5.2.2). Next we describe our approach to learn a similarity metric between street and shop items using deep networks

Category	Keywords
bags	backpack, backpacks, bag, bags, clutch, clutches, evening-handbags, hobo-bag, hobo-bags, satchel, satchels, shoulder-bags, tote-bags, wallet, wallets
dresses	bridal-dresses, bridal-mother, bride, bridesmaid, casual-dresses, cocktail-dresses, day-dresses, dress, dress-pants, evening-dresses, fit-flare-dresses, gown, gowns, longer-length-dresses, maternity-dresses, maxi-dresses, party-dresses, petite-dresses, plus-size-dresses, special-occasion-dresses, teen-girls-dresses, work-dresses
footwear	boot, boots, evening-shoes, flats, heel, mules-and-clogs, platforms, pump, pumps, sandal, sandals, shoe, shoes-athletic, shoes-boots, shoes-flats, shoes-heels, shoes-sandals, shoes-wedges, slipper, slippers, wedges, womens-sneakers
hats	hat, hats, hats-hair-accessories,
leggings	legging, leggings, plus-size-tights, socks, tights, tights-socks,
pants	athletic-pants, bootcut-jeans, casual-pants, classic-jeans, cropped-jeans, cropped-pants, distressed-jeans, flare-jeans, maternity-jeans, maternity-pants, pants, petite-jeans, petite-pants, plus-size-jeans, plus-size-pants, relaxed-jeans, skinny-jeans, skinny-pants, straight-leg-jeans, stretch-jeans, teen-girls-jeans, teen-girls-pants, wide-leg-pants,
tops	athletic-tops, blouse, blouses, button-front-tops, camisole, camisole-tops, camisoles, cashmere-tops, graphic-tees, halter-tops, knit-tops, longsleeve-tops, maternity-tops, petite-tops, plus-size-tops, polo-tops, shirt, shortsleeve-tops, sleeveless-tops, t-shirt, tank, tank-tops, tee, teen-girls-tops, tees-and-tshirts, top, tshirt, tunic, tunic-tops,

Table 5.1: Example mappings between keywords and high-level item categories.

(Sec 5.2.3).

5.2.1 Whole Image Retrieval

In this approach, we apply the widely used CNN model of Krizhevsky *et al.* (Krizhevsky et al., 2012), pre-trained for image classification of 1000 object categories on ImageNet. As our feature representation we use the activations of the fully-connected layer FC6 (4096-dimension). For query street photos, since we have item bounding boxes available, we compute features only on the cropped item region. For shop images we compute CNN

features on the entire image. We then compare the cosine similarity between the query features and all shop image features and rank shop retrievals based on this similarity.

5.2.2 Object Proposal Retrieval

In this approach, we use the selective search method (van de Sande et al., 2011) to extract a set of object proposals from shop photos. Ideally, the proposed windows will encapsulate visual signals from the clothing item, limiting the effects of background regions and lead to more accurate retrievals. In addition this step should serve to reduce some of the variability observed across different online shops and item depictions.

Specifically we use the selective search algorithm and filter out any proposals with a width smaller than $\frac{1}{5}$ of the image width since these usually correspond to false positive proposals. Finally, the 100 most confident object proposals are kept for efficiency. The remaining set of objects proposals have an average recall of 97.76%, evaluated on an annotated subset of 13,004 shop item photos. Similar to the whole image retrieval method, we compute FC6 features on the street item bounding box and on the 100 most confident object proposals for each shop image. We then rank shop item retrievals using cosine similarity.

5.2.3 Similarity Learning

In this approach our goal is to learn a similarity measure between query and shop items. Our hypothesis is that the cosine similarity on existing CNN features may be too general to capture the underlying differences between the street and shop domains.

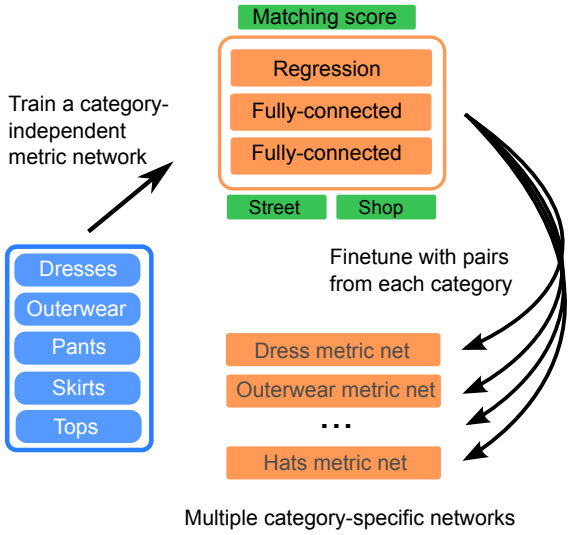


Figure 5.6: Illustration of the training-followed-by-fine-tuning procedure for training category-specific similarity for each category. To deal with limited data, we first train a generic similarity using five large categories, and then fine-tune it with each category individually. See Section 5.2.3 for more descriptions.

Therefore, we explore methods to learn the similarity measure between CNN features in the street and shop domains.

Inspired by recent work on deep similarity learning for matching image patches between images of the same scene (Han et al., 2015; Zbontar and LeCun, 2015), we model the similarity between a query feature descriptor and a shop feature descriptor with a three-layer fully-connected network, and learn the similarity parameters. Here labeled data for training consists of positive samples, selected from exact street-to-shop pairs, and negative samples selected from non-matching street-to-shop items.

Specifically, the first two fully-connected layers of our similarity network have 512 outputs and use Rectified Linear Unit (ReLU) as their non-linear activation function. The third layer of our network has two output nodes and uses the soft-max function as its activation function. The two outputs from this final layer can be interpreted as esti-

Figure 5.7: Example retrievals. Top and bottom three rows show example successful and failure cases respectively.

mates of the probability that a street and shop item "match" or "do not match", which is consistent with the use of cross-entropy loss during training. Once we have trained our network, during the testing query phase we use the "match" output prediction as our similarity score. Previous work has shown that this type of metric network has the capacity for approximating the underlying non-linear similarity between features. For example, Han et al. (Han et al., 2015) showed that the learned similarity for SIFT features, modeled by such a network, is more effective than L2-distance or cosine-similarity for matching patches across images of a scene.

More concretely, we formulate the similarity learning task as a binary classification problem, in which positive/negative examples are pairs of CNN features from a query bounding-box and a shop image selective search based item proposal, for the same item/different items. We minimize the cross-entropy error:

$$E = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})] \tag{5.1}$$

over a training set of n bounding-box pairs using mini-batch stochastic gradient descent. Here, $y_i = 1$ for positive examples; $y_i = 0$ for negative examples; $\hat{y_i}$ and $1 - \hat{y_i}$ are the two outputs of the metric network. One complication is that we do not have hand-labeled bounding boxes for shop images. We could use all object proposals for a shop image in a matching street-to-shop pair as positive training data, but because many boxes returned by the selective-search procedure will have low intersection-over-union (IoU) with the shop item of interest, it would introduce too many noisy training examples. Another source of noisy examples for similarity training is that, due to large pose differences in images for an item, some images on the shop side will bear little similarity in appearance to a particular query item view. Labeling such visually distinct pairs as positives would likely confuse the classifier during training.

We handle these challenges by training our metric network on a short list of top retrieved shop bounding boxes using the object proposal retrieval approach described in Sec. 5.2.2. At test time, we similarly use the object proposal approach to provide a short list of candidate retrievals and then re-rank this list using our learned similarity. This has

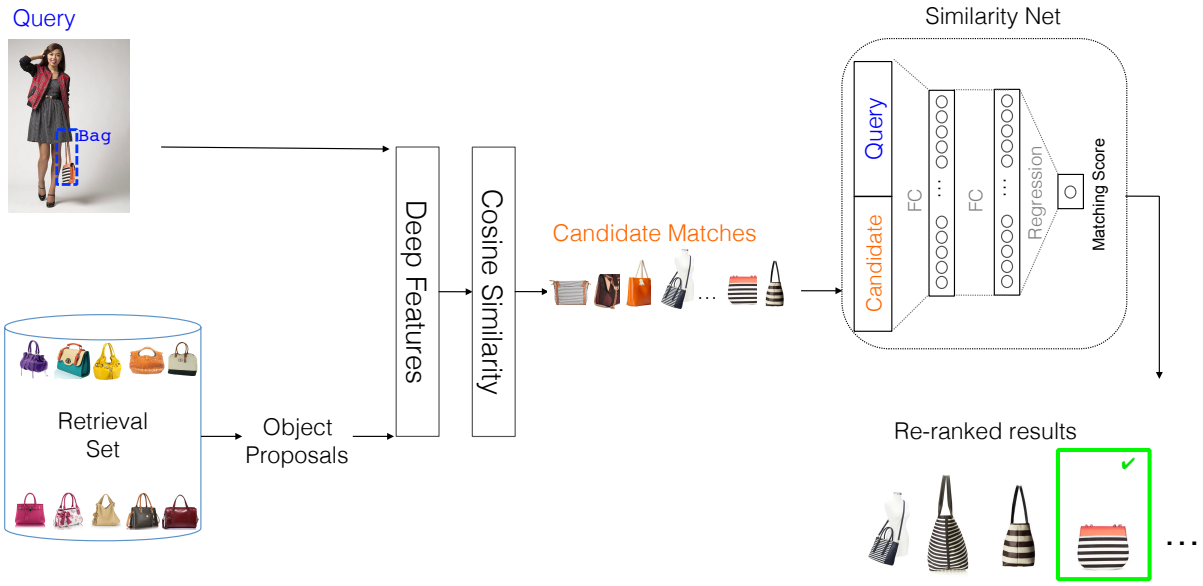


Figure 5.8: Our Exact Street to Shop pipeline

an added benefit of improving the efficiency of our retrieval approach since the original cosine similarity measure is faster to compute than the learned similarity. Figure 5.8 depicts our Exact Street to Shop pipeline.

More specifically, to construct training and validation sets for similarity learning, for each training query item, q, we retrieve the top-1000 selective search boxes from shop images using cosine similarity. From this set, for each bounding box, b, from shop photo, s if the shop image s is a street-to-shop pair with q then (q, b) is used as a positive sample. Otherwise, (q, b) is used as a negative sample[2].

Intuitively, we might want to train a different similarity measure for each garment categories – since for examples objects such as hats might undergo different deformations and transformations than objects like dresses. However, we are limited in the number of

[2]Note, here we use only shop bounding boxes for training belonging to the top-K ($K = 75$) items in the retrieval set

Category	Avg. boxes	Train+	Train−	Val+	Val−
Bags	326	5.6	65.5	1.9	20.7
Belts	384	1.2	28.0	0.4	12.1
Dresses	426	99.7	1456.2	46.1	670.1
Eyewear	276	1.7	69.4	0.1	3.9
Footwear	255	4.3	92.3	5.1	113.2
Hats	356	3.0	82.0	0.4	12.4
Leggings	344	6.5	145.1	2.0	52.1
Outerwear	274	6.5	118.7	2.8	52.4
Pants	443	5.4	75.2	2.3	27.2
Skirts	559	59.2	901.2	17.3	276.1
Tops	349	15.4	186.2	6.9	92.1

Table 5.2: Size statistics of the training and validation sets for similarity learning. Numbers in the last four columns are in units of 1,000.

positive training examples for each category and by the large negative to positive ratio. Therefore we employ negative sampling to balance the positive and negative examples in each mini-batch, and train a general street to shop similarity measure, followed by fine-tuning for each garment category to achieve *category-specific* similarity (See Figure 5.6).

In the first stage of training, we select five large garment categories from our garment categories: Dresses, Outerwear, Pants, Skirts, and Tops and combine their training examples. Using these examples, we train an initial *category-independent* metric network. We first set learning rate to be 0.001, momentum 0.9, and train for 24,000 iterations, then lower the learning rate to 0.0001 and train for another 18,000 iterations. In the second stage of learning, we fine-tune the learned metric network on each category independently (with learning rate 0.0001), to produce category-dependent similarity measures. In both stages of learning, the corresponding validation sets are used for monitoring purposes to determine when to stop training.

Category	Queries	Query Items	Shop Images	Shop Items	Whole Im.	Sel. Search	Similarity.	F.T. Similarity.
Bags	174	87	16,308	10,963	23.6	32.2	31.6	**37.4**
Belts	89	16	1,252	965	6.7	6.7	11.2	**13.5**
Dresses	3,292	1,112	169,733	67,606	22.2	25.5	36.7	**37.1**
Eyewear	138	15	1,595	1,284	10.1	**42.0**	27.5	35.5
Footwear	2,178	516	75,836	47,127	5.9	6.9	7.7	**9.6**
Hats	86	31	2,551	1,785	11.6	36.0	24.4	**38.4**
Leggings	517	94	8,219	4,160	14.5	17.2	15.9	**22.1**
Outerwear	666	168	34,695	17,878	9.3	13.8	18.9	**21.0**
Pants	130	42	7,640	5,669	14.6	21.5	28.5	**29.2**
Skirts	604	142	18,281	8,412	11.6	45.9	**54.6**	**54.6**
Tops	763	364	68,418	38,946	14.4	27.4	36.6	**38.1**

Table 5.3: Dataset statistics and top-20 item retrieval accuracy for the Exact-Street-to-Shop task. Last four columns report performance using whole-image features, selective search bounding boxes, and re-ranking with learned generic similarity or fine-tuned similarity.

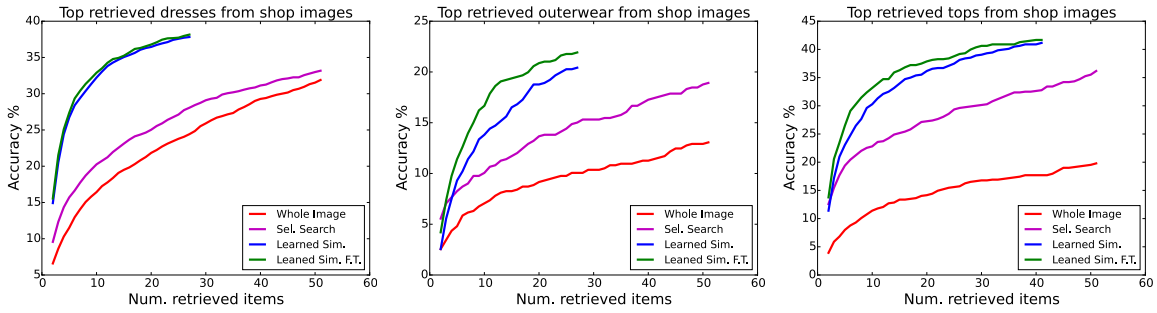


Figure 5.9: Top-k item retrieval accuracy for different numbers of retrieved items.

5.2.4 Train/val Sets for Similarity Learning

Table 5.2 reports the size of `train` and `val` splits of each category for training the similarity learning network. The column for average number of boxes shows the number of selective search boxes used, averaged across query images with at least one successful retrieval within the short list.

5.3 Experimental Results

The proposed retrieval approaches are evaluated with a series of retrieval experiments. For these experiments, we split the exact matching pairs into two disjoint sets such that there is no overlap of items in street and shop photos between train and test. In particular,

for each category, the street-to-shop pairs are distributed into train and test splits with a ratio of approximately 4:1. For our retrieval experiments, a query consists of two parts: 1) a street photo with an annotated bounding box indicating the target item, and 2) the category label of the target item. We view these as simple annotations that a motivated user could easily provide, but this could be generalized to use automatic detection methods. Since the category is assumed to be known, retrieval experiments are performed within-category. Street images may contain multiple garment items for retrieval. We consider each instance as a separate query for evaluation. Table 5.3 (left) shows the number of: query images, query items, shop images, and shop items.

Performance is measured in terms of *top-k accuracy*, the percentage of queries with at least one matching item retrieved within the first k results. Table 5.3 (right) presents the exact matching performance of our baselines and learned similarity approaches (before and after fine-tuning) for k=20. Whole image retrieval performs the worst on all categories. The object proposal method improves over whole image retrieval on all categories, especially on categories like eyewear, hats, and skirts where localization in the shop images is quite useful. Skirts, for example, are often depicted on models or mannequins, making localization necessary for accurate item matching. We also trained category-specific detectors (Girshick et al., 2014) to remove the noisy object proposals from shop images. Keeping the top 20 confident detections per image, we observe a small drop of 2.16% in top-20 item accuracy, while we are able to make the retrieval runtime up to almost an order of magnitude more efficient (e.g.7.6x faster for a single skirt query on one core).

Our final learned similarity after category specific fine-tuning achieves the best performance on almost all categories. The one exception is eyewear for which the object proposal method achieves the best top-20 accuracy. The initial learned similarity measure before fine-tuning achieves improved performance on categories that it was trained on, but less improvements on the other categories.

Example retrieval results are shown in Figure 5.7. The top three rows show success, where the exact matches are among the top results. Failure examples are in the bottom rows. These can happen for several reasons, e.g., visual distraction from textured backgrounds (e.g., 4^{th} row). A better localization of the query item, might be helpful in these cases but perhaps costly. Sometimes items are too visually generic to find the exact item in shop images (e.g., blue jeans in 5^{th} row). Finally, current deep representations may fail to capture some subtle visual differences between items (last row). We also observe errors due to challenging street item viewpoints.

Additionally, in Figure 5.9 we plot the top-k retrieval accuracy over values of k for three example categories (dresses, outerwear and tops). For similarity learning, we vary k from 1 to the number of available items in the retrieved short list. For the baseline methods, we plot accuracy for k=1 to 50. We observe that performance of similarity network grows significantly faster than the baseline methods. This is particularly useful for real-world search applications, where users rarely look beyond the first few highly ranked results.

	Source of distractors	
Category	Similar-to-query (%)	Similar-to-item (%)
Bags	77.3	81.6
Belts	65.5	53.9
Dresses	87.9	69.8
Eyewear	29.6	33.3
Footwear	58.9	44.1
Hats	69.8	57.0
Leggings	45.1	29.4
Outerwear	66.9	57.5
Pants	44.4	37.7
Skirts	69.4	66.6
Tops	78.1	66.1

Table 5.4: Human accuracy at choosing the correct item from different short-lists. Figure 5.10 shows examples of the tasks.

5.3.1 Human Evaluation

After developing automated techniques for finding clothing items from street photos we then performed experiments to evaluate how difficult these tasks were for humans, and to obtain a measure of how close the algorithms came to human ability. In these evaluations a human labeler was presented with the same query that would be given to an algorithm, and a set of possibly matching images. The task for the person was to select the correct item from the options. We use two criteria for determining what set of possibly matching images to show people. Figure 5.10 shows a query and two sets of possible shop photos.

As an initial measure of the difficulty of the task, we have people select a matching item for the query from the items in the dataset that are most similar to the *correct item*. We use whole image similarity to find those most similar items to the correct one. This set of possible choices is illustrated in the bottom part of Figure 5.10. The human labelers select the correct item out of the 10 choices just over half of the time (54.2%

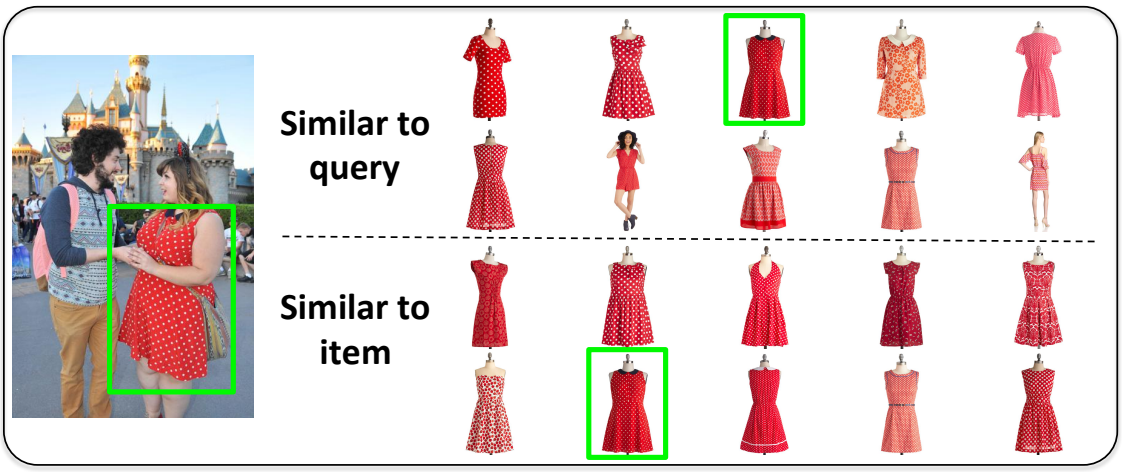


Figure 5.10: An example of our human evaluation tasks

averaged across all item types). This means that just under half the time, people do not pick the correct matching item, even out of a subset of only 10, albeit very similar, choices! This is one indication of the difficulty of the task. Table 5.4 shows results in the "Similar-to-item" column. Figure 5.11 shows two example results of this task. The top row depicts a case where the human labeler failed to pick the exact match. The shop photo selected incorrectly by the human labeler is marked in red, whereas the correct match is marked by green. We observe that humans can make mistakes due to the different lighting conditions between the street photo and the shop photo. Bottom row shows an example where the human labeler successfully picked the correct match.

The second human experiment is designed to temper our optimism about the success of the method. Here we construct the 10 options to include the correct item as well as the 9 items most similar to the *query* according to our learned similarity. (If the correct item was in the top 9 then we add the 10th.) This is illustrated in the top part of Figure 5.10. Here we let the learned similarity choose which image or view of an item

Figure 5.11: Examples of similar-to-item task. Top and bottom rows show cases where the human labeler did not and did pick the exact match respectively.

should be used. Ideally the images picked by our algorithm as good matches for the query will be confusing to the human labelers and they will often pick on of these instead of the correct item. Alas, there is some room for improvement in algorithms. Consider dresses, where our algorithm does relatively well, picking the correct item in the top 10 in 33.5% of trials and getting the first item correct in only 15.6%. In our human experiments, people pick the correct item out of 10 choices 87% of the time for dresses, significantly better. Table 5.4 shows results in the "Similar-to-query" column. Figure 5.12 shows two example results of this task. The top row depicts a case where the human labeler failed to pick the exact match. The shop photo selected incorrectly by the human labeler is marked in red whereas the correct match is marked by green. We observe that difficult poses and viewpoints on the street photo can confuse humans in picking the exact match. Bottom row shows an example where the human labeler successfully picked the correct match despite the fact that most of the candidates are very similar to the query.

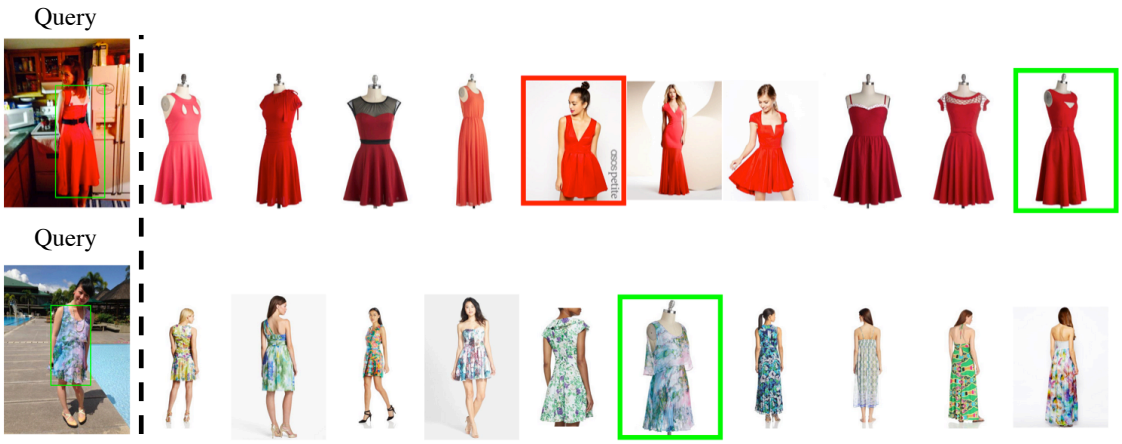


Figure 5.12: Examples of similar-to-item task. Top and bottom rows show cases where the human labeler did not and did pick the exact match respectively.

5.4 Summary and Discussion

We presented a novel task, Exact Street to Shop, and introduced a new dataset. Using this dataset, we have evaluated three methods for street-to-shop retrieval, including our approach to learn similarity measures between the street and shop domains. Finally, we have performed quantitative and human evaluations of our results, showing good accuracy for this challenging retrieval task. These methods provide an initial step toward enabling accurate retrieval of clothing items from online retailers.

CHAPTER 6: SUMMARY AND DISCUSSION

In this thesis we have studied, proposed, implemented and evaluated systems that can automatically learn to represent and identify clothing in images. We focused the scope of some existing computer vision problems on clothing recognition and introduced new problems to the research community that are particularly challenging due to the complex appearance of clothing and their large scale nature.

In Chapter 3 we introduce novel probabilistic and retrieval-based clothing parsing approaches. Our work was the first attempt that tackled clothing parsing in a large scale with 56 different semantic labels. We present new large datasets of fashion images associated with rich meta data that can be potentially useful for research in other interdisciplinary directions. One drawback of the proposed CRF approach was that it only performed well on images for which the labels for depicted clothing items are known in advance. We tackled this problem by using an aggregation of associated tags with the retrieved nearest neighbors in our second approach. Localization and parsing go hand-in-hand. Given the recent advances in deep learning based object detectors, more exploration can be done in training detectors separately for every clothing item to obtain spatial priors for garment items before semantic parsing.

Chapter 4 brings a modern perspective to clothing recognition, which is to examine what our clothing style reveals about us and what are the most discriminative elements that constitute a style. We collected an image dataset depicting five different fashion

styles and collected crowd sourced style rating using our online game. Since our dataset was quite small, we mainly relied on simple classifiers to make predictions about clothing styles in images. Future work includes collecting larger image datasets that contain information about other contributing factors to fashion styles such as gender, country, etc. Larger datasets combined with modern approaches in deep learning would also allow to learn rich mid-level image representations as opposed to hand-designed low-level features used in our image classification methods.

Chapter 5 presents a novel task to the research community, Exact Street to Shop, and introduces a new large dataset of clothing images both from real-world settings and online shops. Using this dataset, we have evaluated three methods for street-to-shop retrieval, including our deep learning based similarity learning. We also performed quantitative and human evaluations of our results, showing good accuracy for this challenging retrieval task. These methods provide an initial step toward enabling accurate retrieval of clothing items from online retailers. Future work includes developing methods for more precise alignment between street and shop items for improving retrieval performance.

BIBLOGRAPHY

Andriluka, M., Roth, S., and Schiele, B. (2009). Pictorial structures for object recognitionrevisited: People detection and articulated pose estimationz. In *CVPR*.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898 –916.

Bell, S. and Kavita, B. (2015). Learning visual similarity for product design with convolutional neural networks. In *ACM Transaction on Graphics (SIGGRAPH)*.

Bergamo, A. and Torresani, L. (2010). Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach. *NIPS*.

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

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

Boykov, Y. and Kolmogorov, V. (2004). 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.

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

Chen, H., Gallagher, A., and Girod, B. (2012). Describing clothing by semantic attributes. In *ECCV*, pages 609–623.

Chen, K.-T., Chen, K., Cong, P., Hsu, W. H., and Luo, J. (2015a). Who are the devils wearing prada in new york city. *ACM Multimedia*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2015b). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1.

Di, W., Wah, C., Bhardwaj, A., Piramuthu, R., , and Sundaresan, N. (2013). Style finder: Fine-grained clothing style recognition and retrieval. In *IWMV of CVPR*.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*.

Dong, J., Chen, Q., Xia, W., Huang, Z., and Yan, S. (2013). A deformable mixture parsing model with parselets. *ICCV*.

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

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

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008c). Liblinear: A library for large linear classification. *Journal of Machine Learning Research*.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. (2012). Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *ICML*.

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

Fernando, B., Habrard, M., Sebban, M., and Tuytelaars, T. (2013). Unsupervised visual domain adaptation using subspace alignment. *ICCV*.

Fu, J., Wang, J., Li, Z., Xu, M., and Lu, H. (2012). Efficient clothing retrieval with semantic-preserving visual phrases. In *ACCV*.

Gallagher, A. and Chen, T. (2008). Clothing cosegmentation for recognizing people. In *CVPR*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*.

Gong, B., Yuan, S., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. *CVPR*.

Gong, Y., Wang, L., Guo, R., , and Lazebnik, S. (2014). Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*.

Gopalan, R., Li, R., and Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. *ICCV*.

Gould, S., Gao, T., and Koller, D. (2009). Region-based segmentation and object detection. In *NIPS*.

Guillaumin, M., Mensink, T., Verbeek, J., and Schmid, C. (2009). Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 309–316. IEEE.

Guo, R. and Hoiem, D. (2012). Beyond the line of sight: labeling the underlying surfaces. In *ECCV*.

Han, X., Leung, T., Jia, Y., Sukthankar, R., and Berg, A. C. (2015). Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*.

Hasan, B. and Hogg, D. (2010). Segmentation using deformable spatial priors with application to clothing. In *BMVC*.

Herbrich, R., Minka, T., and Graepel, T. (2007). Trueskill(tm): A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576.

Hoffman, J., Tzeng, E., Donahue, J., Jia, Y., Saenko, K., and Darrell, T. (2014). One-shot adaptation of supervised deep convolutional models. *ICLR*.

Jammalamadaka, N., Minocha, A., Singh, D., and Jawahar, C. (2013). Parsing clothes in unrestricted images. In *BMVC*.

Jegou, H., Douze, M., Schmid, C., and Perez, P. (2010). Aggregating local descriptors into a compact image representation. *CVPR*.

Jing, Y., Liu, D., Kislyuk, D., Zhai, A., Xu, J., and Donahue, J. (2015). Visual search at pinterest. *KDD*.

Johnson, S. and Everingham, M. (2011). Learning effective human pose estimation from inaccurate annotation. In *CVPR*.

Kalantidis, Y., Kennedy, L., and Li, L.-J. (2013). Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 105–112. ACM.

Kiapour, M. H., Yamaguchi, K., Berg, A. C., and Berg, T. L. (2014). Hipster wars: Discovering elements of fashion styles. *ECCV*.

Kim, B., Sun, M., Kohli, P., and Savarese, S. (2012). Relating things and stuff by high-order potential modeling. *ECCV Workshop on Higher Order Models and Global Constraints in Computer Vision*.

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

Krahenbuhl, P. and Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*.

Krahenbuhl, P. and Koltun, V. (2013). Parameter learning and convergent inference for dense random fields. In *ICML*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.

Kwak, I. S., Murillo, A. C., Belhumeur, P., Belongie, S., and Kriegman, D. (2013). From bikers to surfers: Visual recognition of urban tribes. In *British Machine Vision Conference (BMVC)*, Bristol.

Ladicky, L., Sturgess, P., Alahari, K., Russell, C., and Torr, P. H. (2010). What, where and how many? combining object detectors and crfs. In *ECCV*.

Lai, H., Pan, Y., Liu, Y., and Yan, S. (2015). Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*. IEEE.

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

Lim, J. J., Pirsiavash, H., and Torralba, A. (2013). Parsing ikea objects: Fine pose estimation. *ICCV*.

Lin, K., Yang, H., Liu, K., Hsiao, J., and Chen, C. (2015). Rapid clothing retrieval via deep learning of binary codes and hierarchical search. In *ACM on International Conference on Multimedia Retrieval*.

Liu, C., Yuen, J., and Torralba, A. (2011). Nonparametric scene parsing via label transfer. In *PAMI*.

Liu, S., Feng, J., Domokos, C., Xu, H., Huang, J., Hu, Z., and Yan, S. (2014). Fashion parsing with weak color-category labels. *IEEE Transactions on Multimedia*, 16(1).

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

Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., and Yan, S. (2012b). Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *CVPR*.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157.

Luo, P., Wang, X., and Tang, X. (2012). Hierarchical face parsing via deep learning. In *CVPR*.

Luo, P., Wang, X., and Tang, X. (2013). Pedestrian parsing via deep decompositional network. In *ICCV*.

Marcin, E. and Ferrari, V. (2009). Better appearance models for pictorial structures. In *BMVC*.

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

Mooij, J. M. (2010). libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173.

Mori, G., Ren, X., Efros, A., and Malik, J. (2004). Recovering human body configurations: combining segmentation and recognition. In *CVPR*.

Murillo, A. C., Kwak, I. S., Bourdev, L., Kriegman, D., and Belongie, S. (2012). Urban tribes: Analyzing group photos from a social perspective. In *CVPR Workshop on Socially Intelligent Surveillance and Monitoring (SISM)*, Providence, RI.

Perronnin, F. and Dance, C. (2006). Fisher kenrels on visual vocabularies for image categorizaton. *CVPR*.

Perronnin, F., Liu, Y., Sanchez, J., and Poirier, H. (2010a). Large-scale image retrieval with compressed fisher vectors. *CVPR*.

Perronnin, F., Sanchez, J., and Mensink, T. (2010b). Improving the fisher kernel for large-scale image classification. *ECCV*.

Pfister, T., Simonyan, K., Charles, J., and Zisserman, A. (2014). Deep convolutional neural networks for efficient pose estimation in gesture videos. In *ACCV*.

Ramanan, D. (2006). Learning to parse images of articulated bodies. In *NIPS*.

Ren, X., Berg, A., and Malik, J. (2005). Recovering human body configurations using pairwise constraints between parts. In *ICCV*.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. *ECCV*.

Scheffler, C. and Odobez, J. (2011). Joint adaptive colour modelling and skin, hair and clothes segmentation using coherent probabilistic index maps. In *BMVC*.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering facenet. In *CVPR*.

Shao, M., Li, L., and Fu, Y. (2013). What do you do? occupation recognition in a photo via social context. *ICCV*.

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

Simo-Serra, E., Fidler, S., Moreno-Noguer, F., and Urtasun, R. (2014). A high performance crf model for clothes parsing. In *ACCV*.

Simo-Serra, E., Fidler, S., Moreno-Noguer, F., and Urtasun, R. (2015). Neuroaesthetics in fashion: Modeling the perception of beauty. *CVPR*.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE.

Song, Z., Wang, M., Hua, X.-S., and Yan, S. (2011). Predicting occupation via human clothing and contexts. In *ICCV*.

Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *CVPR*.

Tian, Y., Zitnick, C. L., and Narasimhan, S. G. (2012). Exploring the spatial hierarchy of mixture models for human pose estimation. In *ECCV*.

Tighe, J. and Lazebnik, S. (2010). Superparsing: scalable nonparametric image parsing with superpixels. *ECCV*, pages 352–365.

Tighe, J. and Lazebnik, S. (2013). Finding things: Image parsing with regions and per-exemplar detectors. *CVPR*.

Tighe, J., Niethammer, M., and Lazebnik, S. (2014). Scene parsing with object instances and occlusion ordering. *CVPR*.

Tompson, J., Jain, A., LeCun, Y., and Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*.

Toshev, A. and Szegedy, C. (2014). Deeppose: Human pose estimation via deep neural networks. In *CVPR*.

van de Sande, K. E. A., Uijlings, J. R. R., Gevers, T., and Smeulders, A. W. M. (2011). Segmentation as selective search for object recognition. In *ICCV*.

Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *Int. J. Comput. Vision*, 62(1-2):61–81.

Veit, A., Kovacs, B., Bell, S., McAuley, S., Bala, K., and Belongie, S. (2015). Learning visual clothing style with heterogeneous dyadic co-occurrences. *ICCV*.

Vineet, V., Warrell, J., and Torr, P. H. (2012). Filter-based mean- field inference for random fields with higher-order terms and product label-spaces. In *ECCV*.

Vittayakorn, S., Yamaguchi, K., Berg, A. C., and L., B. T. (2015). Runway to realway: Visual analysis of fashion. In *WACV*.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *CVPR*.

Wang, J., Yang, J., Yu, K., T. Huang, F. L., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE.

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

Xianjie, C. and Yuille, A. L. (2014). Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*.

Yamaguchi, K., Berg, T. L., and Ortiz, L. E. (2014). Chic or social: Visual popularity analysis in online fashion networks. *ACM MM*.

Yamaguchi, K., Kiapour, M. H., and Berg, T. L. (2012). Parsing clothing in fashion photographs. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 3570–3577, Washington, DC, USA. IEEE Computer Society.

Yamaguchi, K., Kiapour, M. H., and Berg, T. L. (2013). Paper doll parsing: Retrieving similar styles to parse clothing items. In *Computer Vision (ICCV), 2013 IEEE International Conference on*.

Yang, M. and Yu, K. (2011). Real-time clothing recognition in surveillance videos. In *ICIP*.

Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392.

Yao, J., Fidler, S., and Urtasun, R. (2012). Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*.

Yosinkski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferrable are features in deep neural networks? *NIPS*.

Zagoruyko, S. and Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *CVPR*.

Zbontar, J. and LeCun, Y. (2015). Computing the stereo matching cost with a convolutional neural network. In *CVPR*.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *ICCV*.