Human Pose Estimation: Extension and Application

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science (By Research)
in
Computer Science and Engineering

by

Digvijay Singh
201002052
digvijay.singh@research.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2016

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Human Pose Estimation: Extension and Application" by Digvijay Singh, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. C V Jawahar

To human syndicates arduously satiating the creative desires..

Acknowledgments

Here I will acknowledge the participation of others that led to the development of this thesis.

Foremost, I thank Prof. C V Jawahar for his intellectual guidance on field-related fundamental problems and providing me the most introspective years of my life. Nataraj J has been a great influence on learning things and showing a curious perspective towards matter and ideas. I acknowledge the guidance provided by Dr. Vineeth Balasubramanian on the research work presented here.

I express my gratitude to friends who have interacted and indulged with me through this phase of my life. More than friends, I had the privilege of being accompanied by truer human beings, these are: Nishith, Anubhav, Nishit, Karan, Himanshu, Mohit, Aditya, Gurjot and Aniket. I appreciate the financial support given by my parents, Jay Pandya and Vishal Tiwari.

Abstract

Understanding human appearance in images and videos is one of the most fundamental and explored area in the field. Describing human appearance can be interpreted as a concoction of smaller and more fundamental related aspects like posture, gesture, outlook *etc*. By doing so we try to grasp holistic sense from semantically lower level information. The utility behind understanding human appearance and related aspects is the industrial demand for applications that involve analyzing humans and their interaction with surroundings. This thesis work tackles two of such related aspects: i. more fundamental problem, human pose estimation and ii. deeper understanding from cloth parsing based on pose estimation.

Determining the human body joint locations and configuration is quizzed as human pose estimation problem. In this work we address the problem of human pose estimation for video sequence data type. We exploit the availability of redundant information from redundant data type. The proposed iteratively functioning methodology has the first iteration involving parsing data from a quintessential generic base model. For the following iterations, we run a 3 step pipeline: grabbing confidently positive detections from previous iteration using our novel selection criteria, fine-tuning external-to-base parameters to local distribution by synthesizing exemplars from picked ones, and enforcing the learned information using an updated amalgamation model. The resulting pipeline propagates correctness in temporal neighborhoods of a video sequence. Previous methods that use the same base models have relied more on tracking strategies. From the unbiased experiments conducted, our approach has proven to be much more robust and overall better performing.

In the second half, we indulge in determining a more deeper understanding of human aspects *i.e.* cloth parsing. This involves predicting the cloth types worn by humans and their segmented regions in images. Our work focuses on incorporating robustness to a previously formulated method. Conceivably, determining hot regions for each cloth type is dependent on the underlying pose skeleton of the human, *eg. hat* will be worn on the *head*. Hence, availability of pose information is key to cloth parsing problem, but incorrect body part estimations can also simply lead to false cloth detections and segmentations. The previous method uses pose information from pictorial structure based model, whereas we update the formulation to incorporate information from more robust part detectors. The changed model has shown to be performing better at more wild outdoor settings. However, the performance from available and proposed methods is below par and appears non-viable for application purposes. To answer this, we report a set of experiments in which we take different lookouts and report observations.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the current digital era, almost every device is mounted by a camera unit that captures videos or images for various purposes. Some examples are video cameras for entertainment purposes, closed-circuit television (CCTV) cameras for surveillance purposes, robot mounted cameras for gathering information about surroundings and environments inaccessible to humans *etc*. Even after so much advancements being made in the domain of technology, we fail to synthesize information about the data which is explicit to a human eye. The most relevant subject we frequently analyze and is a point of focus for a lot of current research work is : *human*. A scene including a person can be understood of comprising three elements:

- **Human.** Where is/are the human(s) located? How many people are in the scene? Is that person a male or a female? Where is/are the face(s) located? Where are the limbs located? What gesture is the human making? What clothes the person is wearing?
- **Surroundings.** Is it an office? Is it a party? Are there other relevant subjects present like animals or birds? What kind of objects are present in the surroundings like a football or a car or a balloon?
- **Human Interaction with Surroundings.** Is/Are the human(s) performing some activity? Is the human holding or touching something? How is the human configuration and position relate with the background like is he standing near a basket or sitting on a chair?

To understand a scene we answer such questions explicitly in our mind using our intelligence. Similarly by using artificial intelligence fabricated by humans, machines should be able to automatically make such categorical judgments. However, in reality the technological progress is still confronting difficulties in behaving prudently. Some examples and associated information are displayed in Figure 1.1.

A single program to provide the holistic interpretation of a scene is extremely difficult to architect. However, by fragmenting the perceivable knowledge, the difficult problem reduces to smaller easier puzzles. Taking a bottom-up route has enabled us to design programs that cumulatively perform much better than any holistic methodology. Therefore, understanding a scene is sub-divided into smaller problems and are tackled individually by keeping the bigger picture in mind. Some examples of problems being tackled by the computer vision community include: human detection, face recognition, human

(a) Activity: People toasting.
8 humans: 4 male, 4 female.
Background: Club
Interaction: Holding glasses

(b) Activity: Boys playing football.
8 humans: 8 boys
Background: Field
Interaction: Kicking football

Figure 1.1 Information we can explicitly notice by looking at still images using Human Intelligence. However, producing such noise-free details automatically is difficult using Artificial Intelligence currently.

gesture recognition, human pose configuration estimation, activity recognition, scene segmentation, cloth parsing *etc* in still images. Videos have redundant temporal information which is used to track motion, body-parts *etc*. The additional information present in a temporal sequence can also be used to rectify errors that a single still image might produce.

1.1 Problem Definition

This work focuses on localization of human body parts and its applications, particularly cloth parsing in this work.

Human Pose Estimation (HPE**).** Estimating the human pose configuration is one of the most fundamental problems in computer vision community because of its primal nature of aiding a plethora of other problems. HPE can be defined as estimating human body parts/joints and their relative configuration with respect to each other. Figure 1.2 (left) shows the pose skeleton of a person having 14 body joints *ie* head, neck, left-right shoulder, left-right elbow, left-right wrist, left-right hip, left-right knee and left-right ankle. From these joints, we can create a human body-part model in which each entity represents a bigger semantic part like face, torso *etc* as shown in Figure 1.2 (right). Some of the core applications of HPE are: human activity recognition in videos and images, human-object interaction models, retrieving similar human postures from very large corpora and human cloth parsing, which we explore in this work.

Cloth Parsing. Cloth parsing involves locating, describing and segmenting all the clothes (*e.g.* t-shirt, dress, pants) and accessories (*e.g.* bag, necklace) that the person of interest is wearing. This is a next step to human detection and pose estimation in understanding human appearance and action. From

Figure 1.2 Left: Pose skeleton of a person with 14 human body joints (left joints labeled). Right: Human body-parts model having 10 parts, each composing of more than 1 body-joint.

experiments conducted and conceivable assumptions, human pose estimation is a very important and relevant information required as prior information for probable cloth locations and the shapes they can take in the image plane. There are enormous number of e-commerce applications that can be derived from a reliable cloth parsing system. Some of these systems are similar clothes retrieval from huge online wardrobes, learning and recommending latest fashion trends and enforcing corrections to input human pose estimations because of more abstract context cloth parsing deals with.

1.2 Challenges

The task of human part localization is crippled by a number of challenges that frequently occurs in real-life scenarios like most of the outdoor images.

- **Occlusion: Self and External.** Generally outdoor images have more than one person or the person is interacting with an element of the surroundings. This leads to a person getting occluded by some other person or a non-human entity like a tree. Such partial or full occlusion induces ambiguity in determining the localization of parts that are thus not visible. Above mentioned occlusions have occurred from an entity which is not the human-in-consideration however, the human can take pose configurations that self induces occlusion. Such phenomenon is observed in most of the outdoor scenarios such as sports which is a domain in consideration of this work. Figure 1.3 (a) shows such examples of self and external occlusion.

- **Background clutter.** Outdoor backgrounds can have a variegated characteristic of high complexity which can make foreground-background segmentation a really tough task even for intelligent

(a) Self and External Occlusion (b) Arms and Legs Foreshortening (c) Varying Illumination

(d) Clothing Variations (e) Motion Blur

Figure 1.3 Various real-life challenges impeding clear visibility of human body parts making the task of pose estimation difficult.

human eye. Multi-patterned and colored background is probable to excite the filters being used at various locations leading to a large number of false positives. Figure 1.4 shows the scatter plot of RGB features for body parts and the background demonstrating the clear overlap and high ambiguity.

- **Body part foreshortening.** Another kind of difficulty faced because of complex human pose configuration is foreshortening of body parts especially limbs. In other words, it can be explained as when body part plane is not parallel to or has a higher angle $(approximately + 45°)$ with the image plane. This difficulty makes the child part localization elusive with respect to the parent part. Image provides a 2D projection of the scene captured thus losing relevant depth information. It can although be countered using additional depth information like stereo images or by using depth sensors like Kinect which might not be available in a lot of circumstances. Figure 1.3 (b) shows arms and legs foreshortening when a person is punching or sitting.

- **Illumination variations.** Indoor surroundings and illumination can be manipulated to make the person-of-interest well in-focus with respect to the background. In outdoor environments, the lightning conditions can vary ranging from really high to really low foreground-background contrast and highlights any arbitrary entity in the scene. Figure 1.3 (c) shows varying illumination conditions highlighting the background or the foreground impeding clear visibility of complete human.

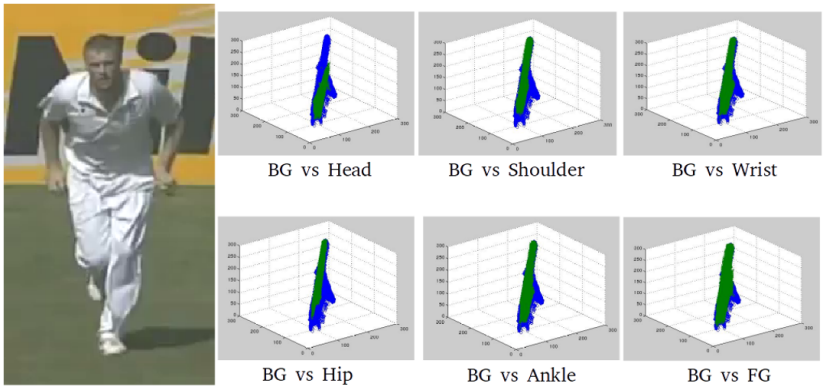


Figure 1.4 RGB scatter plot showing the distribution of various body parts (in green) with respect to background BG (in blue) when obtained from a set of images similar to left image demonstrated.

- **Clothing variations.** The diversity in clothing different people adapt is difficult to encapsulate in a single model. Most of the times the silhouette of a person is the detailing added by the clothing of the person. The silhouette can vary a lot for a girl wearing skirt or a boy wearing pants. Some clothing types cover a number of body parts making it difficult to estimate. Some examples are shown in Figure 1.3 (d).

- **Fast moving parts (In Videos).** In videos where the human is performing fast actions like running behind a ball, the limb parts move rapidly. Even with a high per second frame extraction, the dislocation of corresponding parts is of a high degree. Methods using tracking strategies like Optical Flow or SIFT Flow will not be able to grasp the possibilities unless a high quality part tracker is utilized. Figure 1.3 (e) shows artifacts like motion blur in fast moving parts.

1.3 Contributions

Automatic gain of knowledge regarding human appearance is a fundamental problem in computer vision and the work in this thesis provides extension to Human Pose Estimation problem for robust outdoor data settings as well as one application that leverage information from HPE *i.e.* Cloth Parsing. Listed below are the contributions:

- **Fine-tuning Human Pose Estimations in Videos.** We propose a semi-supervised self-training method for fine-tuning human-pose estimations in videos that uses a dynamic ensemble of exemplars (E-SVM) and a static Pictorial Structure based model as the self-training model that is able to procure accurate estimations and is more robust in complicated settings.

- **Parsing Clothes in Unrestricted Images.** In this work, we propose a method to segment clothes in settings where there is no restriction on number and type of clothes, pose of the person, viewing

angle, occlusion and number of people. Parsing clothing is a direct application of human pose estimation and we show that using better pose representations for real-life images improves the performance over other representations.

1.4 Thesis Outline

The organization of this thesis is as follows. Chapter 2 investigates the recent paradigms in human pose estimation and provides an analysis of failure and success cases. Applications that are directly based on pose estimation outputs are also reviewed. In chapter 3, we present our semi-supervised self-training method for video sequences. The self-trained model is an amalgamation of Pictorial Structure based model and Ensemble of Exemplars. Chapter 4 shows an application of human pose estimation: Cloth parsing. We present our improvement over other methods for real-life unrestricted images followed by further experiments in which we explore various strategies.

Chapter 2

Human Pose Estimation: Recent Paradigms

In this chapter the reader is familiarized with the recent technical advancements related to the problem of Human Pose Estimation and its applications. As already discussed, the task of generating accurate human pose skeletons is impeded by a number of intricacies, some have been handled with the help of meticulous models whereas others like occlusion or great deformation still prevail eventually failing the algorithms in complicated scenarios. We first begin with the introduction to HOG [13] features and their utilization for Human Detection which is followed by some successful paradigms for human pose estimation: Poselets, Pictorial Structures based and DeepNet Architecture based. In section 2.5, we show some qualitative and quantitative analysis from different approaches and section 2.6 mentions some essential applications that directly utilize human pose estimation outputs.

2.1 HOG and Human Detection

The problem of classifying an image as human or not-human was fundamental in the previous decade. Viola *et al.* [60] uses a cascade of proposal-rejecting entities trained using AdaBoost. The problem of human detection was being tackled from different directions [39], [33], [34], [51]. Dalal and Triggs [13] demonstrated that we are underestimating the utility of a dense gradient feature descriptor. They devised Histogram of Oriented Gradients (HOG) feature set which fitted with a very simple architecture to train a Linear SVM improved our stand by a large margin.

In detail, HOG features are obtained by subdividing image regions into cells and synthesizing gradient histograms from each. Obtained set is invariant to color and illumination features that can be different for each person in the images. Figure 2.1 shows HOG features obtained from two images, one having a person and the other not. The gradients enhance the boundaries of the objects with some spatial invariance (depending on the cell size), making it easier to contemplate a human silhouette. Utilization of HOG became fundamental irrespective of models and architecture. The idea of using HOG trained detectors for human detection was extrapolated for the more refined problem of detecting human body parts and pose configuration. Section 2.2, 2.3 and 2.4 details different paradigms for human pose estimation.

Figure 2.1 HOG representation of two images: (a) containing human, (b) missing human. The shape of the gradients suggest an outline of the objects present in the image. Implementation used from VLFeat [59]

2.2 Poselets: Body Part Detectors trained with Clustering in configuration space

Poselets [4] are object detectors that are each trained to locate a human body part with a set viewpoint using 3D annotations from H3D [4] dataset. From a random frame in the training set, keypoint based similarity is computed with rest of all the training set frames and are clustered as one poselet with a cross-validated threshold. All the points in a cluster become examples of one representative poselet. The keypoint based similarity distance of sample b from sample a is computed as :

$$D(a,b) = w_a.||x_a - x_b||_2^2.(1 + v(a,b)) \tag{2.1}$$

where w_a is a Gaussian weight term. x_a are the 3D coordinates of keypoints of sample a and $v(a,b)$ penalizes visual dissimilarity between them. Figure 2.2 shows similar patches obtained using the similarity distance.

Figure 2.2 Left column shows query frame followed by examples retrieved using similarity distance 2.1. Taken from [4].

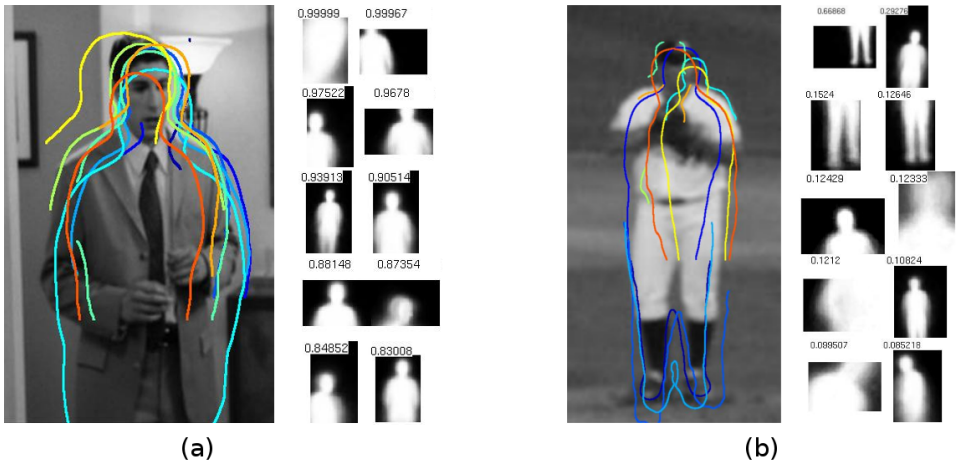


Figure 2.3 Poselet detections (colored contours) with top 10 poselet masks and their scores that each fire one hypothesis for (a) FLIC [47] and (b) LSP [28] dataset.

Poselet classifiers are trained as a linear SVM using poselet examples as positives and other non-person containing images as negatives. Features used are HOG [13]. For testing, probability maps are obtained using all poselet classifiers throughout the image. Next, the probabilities maps are combined optimally using max-margin framework. Being a part detector, each poselet is independent of the presence or absence of other parts to fire. Even with the max-margin framework, this approach fails to exploit a more holistic representation of the complete human pose.

Figure 2.3 shows multiple firing poselet hypothesis and their scores on test images. Figure 2.4 shows final human bounding box and torso bounding box which is achieved by optimally combining poselet probability maps using max-margin framework.

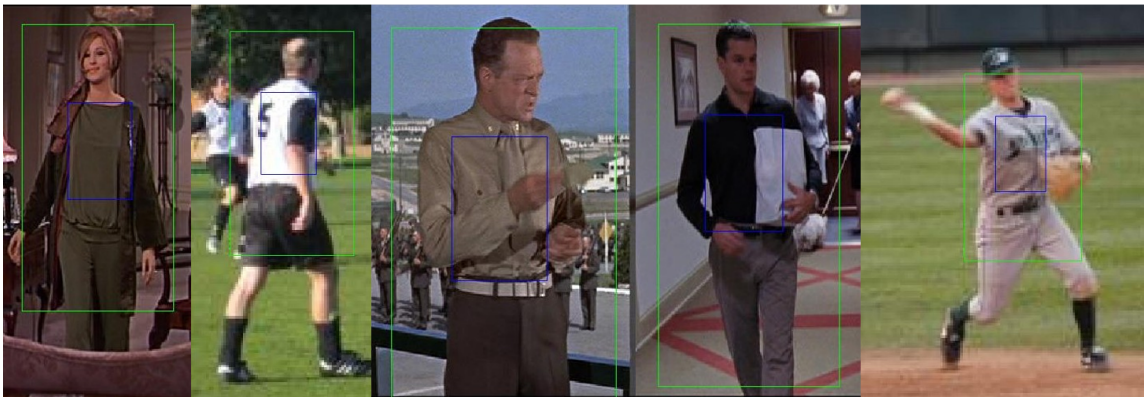


Figure 2.4 Poselet human bounding box (in green) and torso bounding box (in blue) detection for random frames from FLIC [47] and LSP [28] dataset.

2.3 Pictorial Structure based Models

Fischler and Elschlager [20] devised the concept of Pictorial Structures for objects that are an embedding of different prominent sub-parts. From the example given in Figure 2.5 of human face as an object and its sub-parts eyes, nose, mouth *etc* interconnected to maintain a feasible configuration.

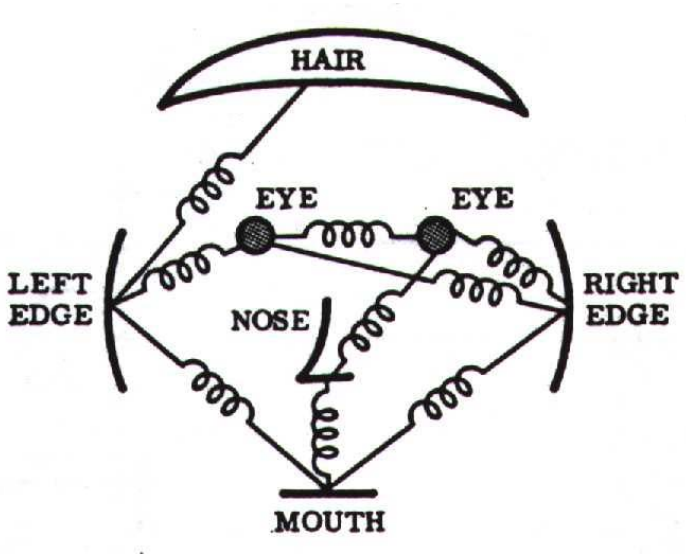


Figure 2.5 Pictorial structure for face as represented by [20].

The observation that a deformable object changes shape and appearance based on the occurrence of different object parts in the space led to the modelling of a framework that allows object parts to occur in a more flexible region space. Some object parts are connected with edges that captures their combination probabilities. The object model can be described as a graph, $G = (V, E)$ where V is the set of vertices $\{v_1, v_2.., v_n\}$ each belonging to an object part and edges E connecting different object parts. Each vertex $v_i \in V$ is associated with a part configuration l_i which contains the observational information related to i^{th} part. l_i can represent the location, orientation and scale for part i and the combination of all part configurations gives us the object configuration $L = \{l_1, l_2.., l_n\}$. The modeling is done using Bayesian framework, aiming to find an optimal object configuration L^*. For image I and modelling graph $G = (V, E)$, L^* is searched as:

$$L^* = \underset{L}{argmax} P(L|I, \theta) \tag{2.2}$$

where θ is the learned model parameters. Using Bayes formula:

$$P(L|I, \theta) = \frac{P(I|L, \theta).P(L|\theta)}{P(I|\theta)} \tag{2.3}$$

$$P(L|I, \theta) \propto \underset{I}{P(I|L, \theta)}.\underset{II}{P(L|\theta)} \tag{2.4}$$

In equation 2.4, first term represents the appearance matching of I with the given configuration and model parameters. Second term is simply a prior probability score evaluating possible configurations.

Similar to face, we can create a pictorial structure framework for the complete human pose in which the different human body parts like head, shoulder, legs become the deformable parts that are connected semantically as shown in Figure 2.7. Some representative approaches utilizing this framework are below.

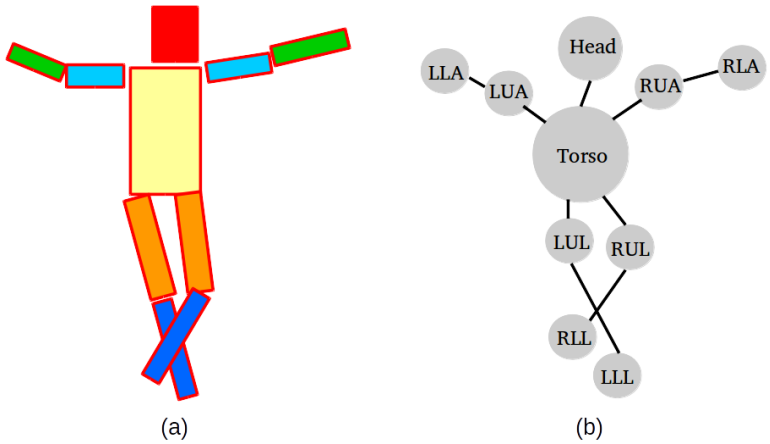


Figure 2.6 (a) Pictorial structure for human pose with different body parts like torso, head, arms and legs forming deformable parts and (b) its tree representation.

Articulated Human Detection with Flexible Mixtures-of-Parts [67].
Instead of having articulated larger parts like torso, a deformable part representation is created with much smaller parts like joints. The joints are dependable on the geometric configurations of their parents. To make larger parts, smaller parts are connected through springs that incorporate flexibility in their connections. The model is a combination of an appearance model and a co-occurrence model. Appearance term parameters are learned optimally from HOG features whereas co-occurence term is a pre-computed value for each combination of pairs. For a graph $G = (V, E)$ given a test image I, the full score equation written as:

$$S(I, z) = \sum_{i \in V} \phi_i(I, z_i) + \sum_{ij \in E} \psi_{ij}(z_i, z_j) \tag{2.5}$$

Parameters ϕ_i and ψ_{ij} are learned using a supervised learning paradigm to maximize the full score in equation 2.5. Inference involves achieving a configuration that maximizes the full score given the learned parameters. This step is efficiently done using dynamic programming since the model matches the criteria for MRF framework.

MODEC: Multimodal Decomposable Models for Human Pose Estimation [47].
Sapp and Taskar [47] commented that traditional approaches are using linear models to learn a trend that is multi-faceted by nature. In this work, they capture multi-modality at a higher granular level of half-bodies. Figure 2.8 shows a MODEC pose model along with some modal average images which are representative of different modes. Modes are obtained from the training set using k-means clustering

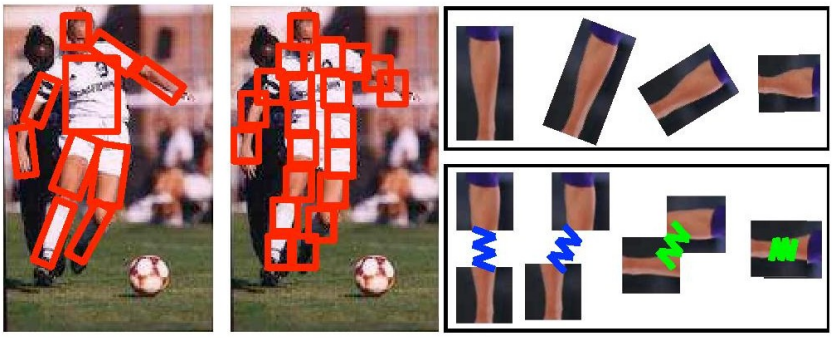


Figure 2.7 Classic deformable parts model (left) consider a body part as a node in the tree whereas, [67] recommends mixture-of-parts (middle) which can handle different warps with more flexibility (right).

using square Euclidean distance measure. After set iterations, each cluster is labelled as a mode and the examples in the cluster become the members for that mode.

The objective of learning is to train parameters that helps in identifying the closest matching mode and locating the body keypoints. To reduce the amount of search space while testing, a number of modes are filtered using a cascaed prediction step in the pipeline. The filtered modes are all used to make predictions on the testing frame and the argmax estimation is picked. This approach differs from a static pictorial structures model such that a mode represents a part of the tree and each of them handles the configuration parameters locally. So depending on the level of abstraction at which the modes are set to be obtained, the size of tree encapsulated by it will change.

Results. Section 2.5 gives a quantitative and qualitative analysis of the above two approaches. We use the code provided by the author of both.

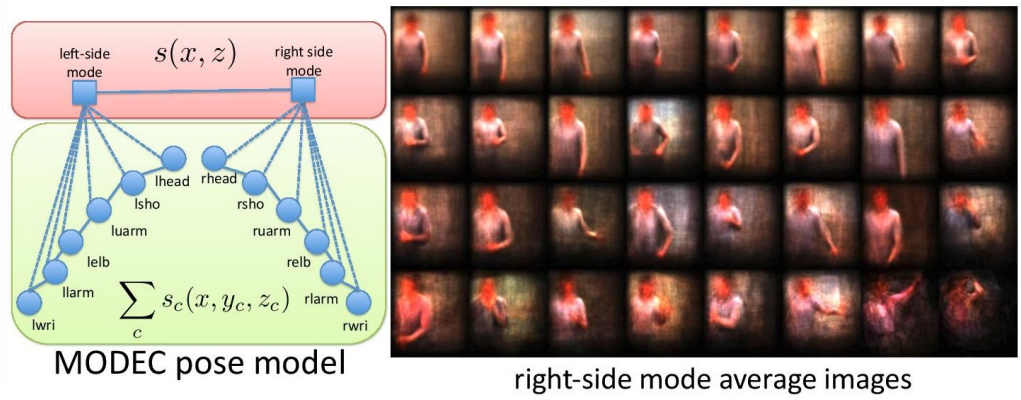


Figure 2.8 Left shows the MODEC pose model proposed by [47] in which the upper body is divided into two left-right modes. Right shows some right-side mode images representing different configurations.

2.4 DeepNet Architecture based Models

Contemporary developments of utilizing Deep Neural Network DNN architecture for fundamental problems led to the formulation of various approaches for Human Pose Estimation using DNN. Following are some of the works that use this architecture:

DeepPose: Human Pose Estimation via Deep Neural Networks [58].
Toshev and Szegedy [58] were the first to adapt the problem of human pose estimation into a DNN framework. The work is however quite simplistic and exploits the meticulously designed architecture from Krizhevsky *et al* [29]. Considering a total of k keypoints to be estimated, the architecture takes a $220X220$ sized input patch and regresses a vector of size $2k$. Figure 2.9 shows the architecture of DNN used and its input-output. For better localization accuracies, a cascade of pose regressors is maintained. This can be visualized as a cascade of identical but parameter-wise independent networks. For training each of the networks in the cascade, loss value is obtained through L_2 distance formula between the prediction and ground-truth as:

$$LossValue(L_2) = \underset{\theta}{argmin} \sum_{i=1}^{k} ||y_i - \phi(x_i, \theta)||_2^2 \tag{2.6}$$

where, θ denotes the network parameters and x is the input frame. Function ϕ represents the non-linear transformation brought upon by the DNN. Further, the loss value is back-propagated to fine-tune the parameters in each of the layers. In the cascade, the networks following the first uses the localization produced by the previous network and tries to improve the estimation at a finer scale.

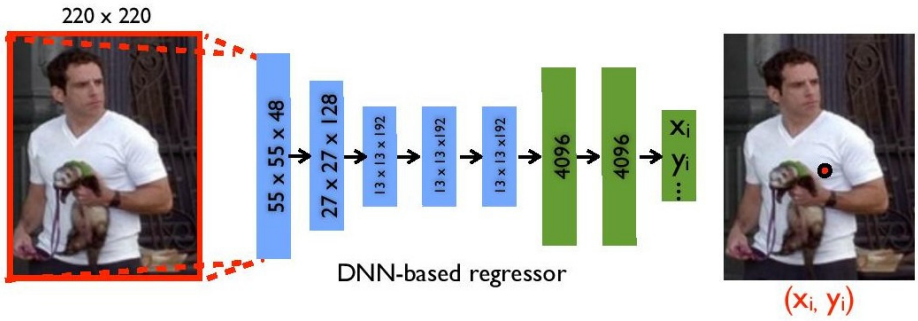


Figure 2.9 DNN architecture used for DeepPose [58] regressors that takes 220x220 input patch and generates a set of coordinates representing body joints.

Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations [10].
This approach takes advantage of the flexibility provided by a graphical model and uses the high performance non-linear projection of feature space produced by the DNN architecture. Note that HOG features are not very useful to be feeded to the network because it has been observed that DNNs generally transforms image features to a much higher non-linear dimension which is more clearly separable than HOG

or any of its variants. They argue that pairwise relations must not only consider the data driven pairwise priors but also learn from the image features. Given an image I, full score for part locations l with pairwise relations t is given as:

$$S(l,t|I) = \sum_{i \in V} U(l_i|I) + \sum_{(i,j) \in E} P(l_i, l_j, t_{ij}|I) \tag{2.7}$$

where the first term is the data appearance term for part i to lie at location l_i. The second term computers the pairwise compatibility between nodes having an edge, and is a combination of spatial deformation between parts and an Image Dependent Pairwise Relational (IDPR) term. Parameters for part appearance detectors and IDPR weights are learned using stochastic gradient applied on a DNN architecture.

Results. Code provided by the authors of [10] is used to obtain experimental results shown in section 2.5.

2.5 Comparison Analysis

In this section, we evaluate and compare different methods for Human Pose Estimation. These are:

- Articulated Human Detection with Flexible Mixture-of-Parts (YR): Deformable Parts Model based.
- MODEC: Multimodal Decomposable Models for Human Pose Estimation (MODEC): Deformable Parts Model based.
- Articulated Pose Estimation by a Graphical Model with Image Dependent Pairwise Relations (IDPR): DNN architecture based.

Datasets Used. We use two datasets for comparison evaluation:

- FLIC. [47] Human upper body dataset having 5000 movie frames with 1000 testing and 4000 training frames.
- LSP. [28] Human full body dataset containing 2000 images from sports domain with 1000 testing and 1000 training images.

Evaluation Metric. We have used the Percentage of Correct Keypoints (PCK) evaluation metric proposed by [67] everywhere in this text. PCK terms only those estimated keypoints correct which lie within a pixel threshold distance of D_T from their counterpart in ground truth. The threshold distance is determined by the formula : $D_T = \beta.max(h, w)$, where h and w are the height and width of the tight bounding box created around the ground truth pose and β is a threshold controlling the relative correctness magnitude. β is set to 0.1 for LSP dataset and 0.2 for FLIC dataset.

Analysis. The analysis done here compares the traditional Deformable Parts based methods (YR and/or MODEC) against one recent DNN architecture based method (IDPR).

FLIC (Upper body-joints detection): The aforementioned three methods are used to predict 6 human upper-body joints (*left-shoulder, left-elbow, left-wrist, right-shoulder, right-elbow, right-wrist*). Table 2.1 shows that DNN architecture based method IDPR outperforms by a huge margin of 21.14% and 28.67% from traditional DPM based methods YR and MODEC respectively. Figure 2.10 shows joint estimations using IDPR (Bottom row) and by using DPM based models YR and MODEC (Top and middle row respectively). DPM based models are approximately equivalent performance-wise but DNN architecture based method improves with a substantial amount and predicts reliably well.

Method	YR [67]	MODEC [47]	IDPR [10]
PCK (%)	70.00	62.47	91.14

Table 2.1 PCK performance of aforementioned three methods on FLIC [47] dataset.

Figure 2.10 Comparison results: Visualizations showing human upper body joints detections by YR (Top row), MODEC (Middle row) and IDPR (Bottom row) on some frames from FLIC [47] dataset.

LSP (Full body-joints detection): Two previously mentioned methods (YR and IDPR) are used to predict 14 human full-body joints (*head, neck, left-shoulder, left-elbow, left-wrist, left-hip, left-knee, left-ankle, right-shoulder, right-elbow, right-wrist, right-hip, right-knee and right-ankle*). Table 2.2 shows that DNN architecture based method IDPR outperforms by a substantial margin of 12.10% from traditional DPM based method YR. Figure 2.10 shows joint estimations using IDPR (Bottom row) and by using DPM based model YR (Top row). Clearly IDPR shows significant improvement over YR even for this really complicated dataset from sports domain.

Method	YR [67]	IDPR [10]
PCK (%)	55.81	67.91

Table 2.2 PCK performance of DPM based method YR and DNN architecture based method IDPR on LSP [28] dataset.

Figure 2.11 Comparison results: Visualizations showing human full body joints detections by YR (Top row) and IDPR (Bottom row) on some images from LSP [28] dataset.

In a nutshell, this analysis states that recent DNN architecture performs considerably better than traditional state-of-the-art methods. This suggests that DNN architecture are better at interpreting the holistic pose configuration using higher dimensional projection when compared with older methods.

2.6 Applications of Human Pose Estimation

In this section, we elaborate on various applications that benefit from Human Pose Estimation. A pose skeleton is a crude way of understanding human appearance *i.e.* the spatial positions occupied by various body joints. It can be used to extract actual appearance of human in interest: Clothing Parsing. Unrelated activity classes can have representative pose skeleton configurations which can be used for Activity Recognition in images and Videos. Pose information is used in Human-Object interaction models that simultaneously detects and corrects body joints locations and object-of-interest localization. Pose estimations are also used to retrieve similar poses in images and videos from large databases.

2.6.1 Clothing Parsing

Human Pose Estimation is key to the problem of Cloth Parsing in providing probable image regions for all cloth labels. Below we mention some methods that exploit this dependence.

(a) Superpixels (b) Pose estimation (c) Predicted Clothing Parse

Figure 2.12 The pipeline adopted by Yamaguchi *et al.* [64] for cloth parsing: (a) Getting superpixels, (b) Getting pose estimation and (c) Cloth label inference using available knowledge.

Parsing Clothing in Fashion Photographs [64]

Yamaguchi *et al.* [64] were the first to formulate the problem of clothing parsing to incorporate pose estimations in a CRF framework. They introduced Fashionista [64] dataset of fashion images having one person in each image wearing clothing items and accessories and the objective is to detect and segment clothing and accessories along-with skin, hair and background. There are three steps in this cloth parsing pipeline as also shown in 2.12:

- Obtaining Superpixels: For a given image I, the labeling is done at superpixel level, which is a region corresponding to similar pixels. The image is divided into N number of superpixels, hence obtaining N number of image regions. Each superpixel region sup_i, where $i \in 1..N$ serves as a node in the CRF framework and constant cloth label l_i is assumed for all pixels belonging to this region.

- Obtaining Human Pose Estimations: To provide pose information to the model, the traditional Pictorial Structures based estimation model by Yang and Ramanan [67] is used. It synthesis a top-scoring pose configuration T for given features in image I.

- Cloth Label Inference: A labeling $L = \{l_i\}$, for $i \in 1..N$ is done by reducing the problem to MAP estimation as follows:

$$P(L|T, I) = \sum_{i \in N} \phi(l_i|T, I) + \sum_{(i,j) \in NR} \lambda_1 \psi_1(l_i, l_j) + \sum_{(i,j) \in NR} \lambda_2 \psi_2(l_i, l_j|T, I) \tag{2.8}$$

  where λ_1 and λ_2 are pairwise model parameters and NR are the neighboring image region pairs. First term on the right hand side is the unary potential for image region i given pose T and image I. Second and third terms are pairwise terms representing co-occurrence prior information of clothing labels in neighboring regions and smoothing term that is a probability estimation of neighboring regions to have the same label.

A High Performance CRF Model for Clothes Parsing [53]

The formulation in [53] is very much similar to the above mentioned [64] method. Simo-Serra *et al.* use a pose-aware CRF framework with the same pairwise potentials as in [64]. However, various unary scores encapsulating information in different forms are used as listed here:

- Clothelets: These are garment likelihood masks that encapsulate the body-joint and garment type dependencies. While training, the region around joint locations are considered and average masks are obtained for each garment type across all the training samples. Further while testing, the masks are laid on the inferred pose to provide garment probable regions in the images.

- Region Shape Features: Region descriptors derived from SIFT are used to encode features in a region of interest and its background. A region corresponds to a superpixel in the image that captures more local shape and appearance of the garment type.

- Object Mask: These masks provide a likelihood of a superpixel belonging to the foreground or background class (foreground representing human occupied region). The masks are obtained per image using CPMC [7] region proposal technique. Top scoring proposal above a learned threshold are the picked regions as human object masks.

- Class Biases: Two types of biases are incorporated in the model. First is a simple entity that encodes a frequently occurring background class bias. Second is a per garment class bias that captures location specific bias for the particular garment class.

- Simple Features: These are RGB, CIEL*a*b, gabor filter and pose coordinate features belonging to a local region inside a superpixel. A logistic regression classifier is trained similar to [64].

2.6.2 Action Recognition

Action recognition in images and videos is a widely researched problem in the community. Current methodologies can be grouped into two categories: using pose estimation features and using coarse/mid-level feature representation. State of the art results are achieved using coarse/mid-level feature representations despite pose seeming to be an evident cue due to the incorrectness in present pose estimation algorithms. Below we mention a couple of approaches that tackle action recognition problem using human pose estimations for video data.

An approach to pose-based action recognition [61]

Work presented by Wang *et al.* [61] deals with the action recognition problem in two steps: 1. Getting improved pose estimations for the complete video sequence, and 2. Action recognition using representative spatial-temporal structures.

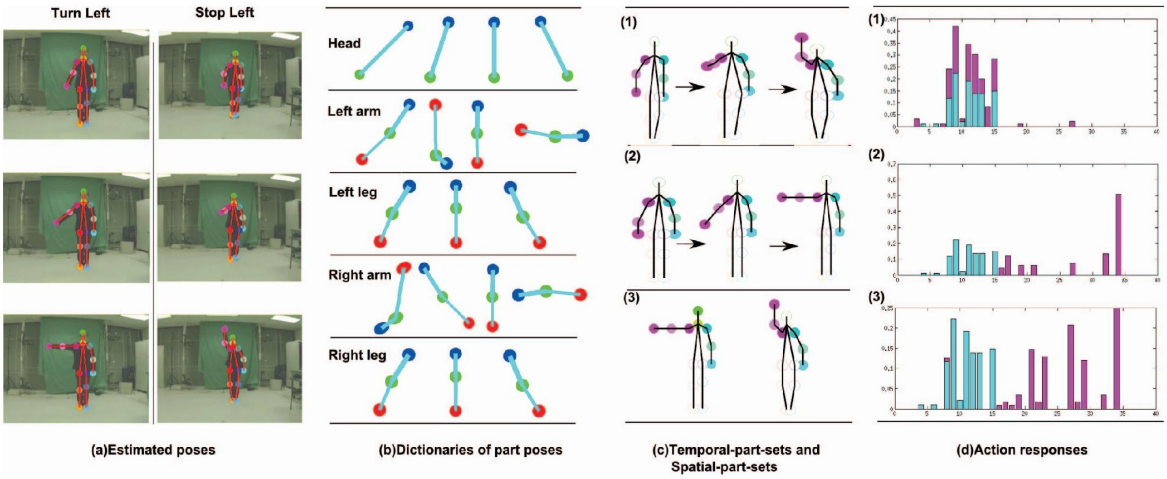


Figure 2.13 Action recognition pipeline postulated by [61]: (a) Pose estimations obtained for two different actions using modified methodology, (b) Dictionaries of different part poses that are obtained by clustering while training, (c) (1-2) shows two temporal part sets and (3) shows two spatial part sets for a sequence, and (d) (1) shows the histogram feature representations for 2 sequences belonging to the same action class and (2-3) shows histogram for disparate action classes. The similarity in histograms from first plot is representative of their similar class and vice-versa for the other two plots.

For initial estimates, the conventional image-based pose estimation method by Yang and Ramanan [67] is used for all frames in the sequence. Instead of picking the top-scored estimation, top K-scored estimations are picked for each frame so as to keep the recall value high. Further, to obtain correct poses and rejecting the false $K-1$ configurations for all the frames in the sequence a new pairwise term is incorporated that ensures temporal consistency across the sequence. The pairwise term accounts for appearance and joint location coherence.

Poses obtained are clustered into part poses, the paper defines five types of such part-poses: Head, Left arm, Left leg, Right arm and Right leg as shown in 2.13 (b). While training, a dictionary of such

part poses is created which is used for mining similar configurations while testing. In the next step, spatial-temporal part sets are obtained. Spatial part sets are spatial configurations of body parts in a sequence frame. The idea of storing such configurations is the distinctive nature of certain configurations for a particular class action. Some spatial part sets are shown in 2.13 (c)(3). To capture joint tracks through time, temporal part sets are obtained. Frequently co-occurring tracks for an action class can be considered as representative tracks for the same class. In the final step, the obtained part steps are quantized into a histogram which becomes the feature vector for the video. SVM is trained using the aforementioned features for each class and label is provided to the class that produces the maximum score.

Joint Action Recognition and Pose Estimation From Video [37]

Nie *et al.* [37] propose that jointly training and inferring action recognition and pose estimation procures improved results over previous methods that deal with these two problems sequentially.

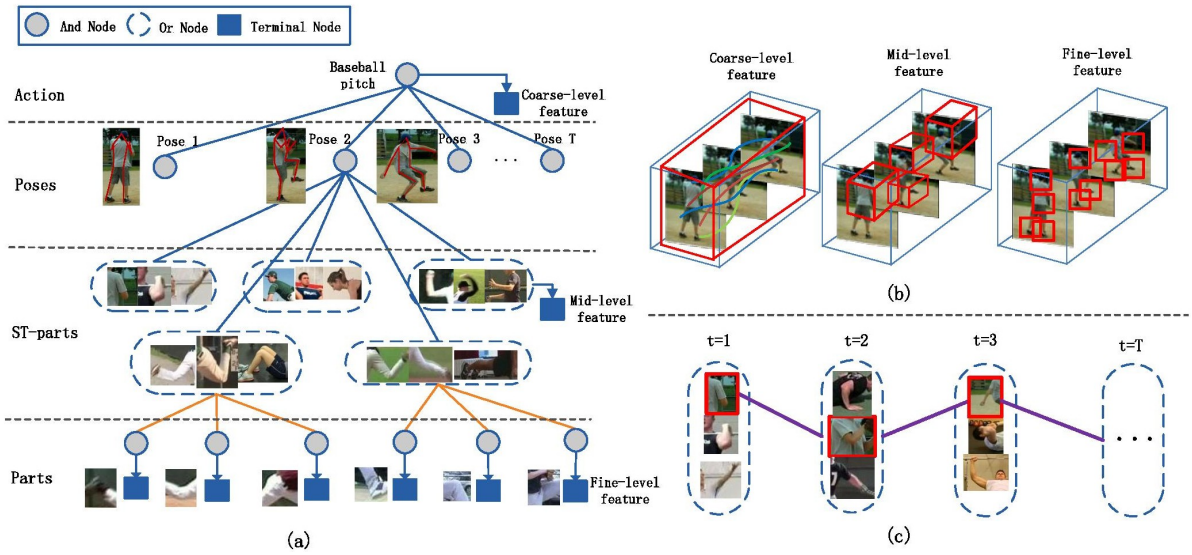


Figure 2.14 Representation by [37]. (a) The spatial-temporal And-Or graph showing hierarchical levels and their decomposition from actions to poses to spatial-temporal parts and parts. (b) Three feature representations utilized at different layers of the graph. (c) Visualization of lateral temporal connections for a particular spatial-temporal part, left arm here. Dynamic programming aided inference determines the best track obtained (shown connected with purple edges).

To represent action, a spatial-temporal And-Or graph model as shown in Figure 2.14 (a) is built that represents both actions and poses. The And-Or graph is composed of three top-down layers and lateral temporal relations for poses and its parts in consecutive frames. At the top layer of the graph, action is represented by coarse-level features. In the top-down hierarchy, action is first decomposed into full poses and each pose is then disintegrated into five spatial-temporal parts. The spatial-temporal parts are picked such that they are independent of each other and present a holistic interpretation when

combined, these are head, right arm, left arm, right leg and left leg. Mid-level features are used to depict these spatial-temporal parts which are also connected laterally to suffice the need of capturing motion information. The spatial-temporal parts are decomposed to obtain single parts from which we also obtain fine-level features at the bottom-most layer of the graph. Figure 2.14 (b) shows the three feature types used at different layers of the graph at action nodes, spatial-temporal part nodes and part nodes. Figure 2.14 (c) gives an example of how spatial-temporal parts are obtained for each frame across the sequence timeline. Further by using MAP formulation and the trained parameters, the best possible track for a spatial-temporal part is obtained using dynamic programming. Finally after obtaining best possible paths for each class by using energy minimization formulation, the action label with maximum score is obtained.

2.6.3 Human-Object Interaction Activities

This problem deals with the interaction of humans and an object of interest. For instance, in an activity like cricket-batting, the *batsman* is the person of interest and the *bat* is the object of interest. Detecting the human parts directly interacting with the object helps in detecting the object of interest. Similarly, detecting the object gives additional cues for possible locations of body-parts that could be interacting with the object. Below mentioned method exploits this mutual correlation in order to improve human part detection and object detection simultaneously.

Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities [68] Yao *et al.* [68] captures the mutual context between object and the human using a random field model as shown in Figure 2.15 (a). The random field shown encodes the action class A, human pose H and object

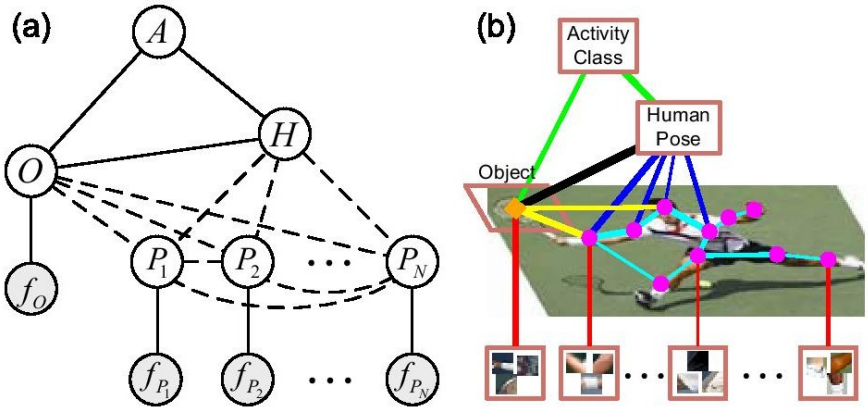


Figure 2.15 Representation by [68]. (a) Hierarchical graph model that shows the connections between action class A, human pose H and object O. Dashed lines show the connections whose weights are decided by the learning process. P is a human body part and f stands for the feature representation of an entity. (b) The graphical model overlaid on an image belonging to $tennisforehand$ action class.

O along-with their mutual connectivity. f stands for the feature representation of different entities and the pose H is decomposed into further pose parts P_n. For an edge e in the field having potential function ϕ_e and edge weight w_e, the model is evaluated as : $\psi = \sum_e w_e \phi_e$. Edge potentials for different kind of connections are evaluated as follows:

- $\phi_e(A, O), \phi_e(A, H), \phi_e(O, H)$: Co-occurrence validity of labels A, O and H from the obtained prior information from training set.
- $\phi_e(O, P_n)$: Spatial relationship between object O and a body part P_n.
- $\phi_e(P_m, P_n)$: Spatial relationship between two body parts.
- $\phi_e(H, P_n)$: Feasibility of spatial layout of part P_n and pose H to co-occur.
- $\phi_e(O, f_O), \phi_e(P_n, f_{P_n})$: The evidence obtained from local image region features for object and all the pose parts.

The model is trained as a structure learning problem to learn the edge connectivity and the model weights to maximize differentiation. For a new testing image I, the best configuration of object and human pose for an action class k is obtained by using the MAP formulation: $max_{O,H}\psi(A_k, O, H, I)$. Finally the action class procuring the maximum confidence score is obtained.

2.6.4 Pose Search in Databases

Pose search is a very quintessential example of the utility automatic pose generation provides for online real-time retrieval of similar pose samples from huge corpora of images and videos.

Video Retrieval by Mimicking Poses [27]

Jammalamadaka *et al.* [27] demonstrate a real-time pose retrieval framework on a database consisting of three million frames. In the pipeline, the frames in the database are processed off-line to generate pose estimates that are later used for matching with the query. To improve the correctness of pose estimations, [27] uses two pictorial structure based methods by Yang and Ramanan [67] and Eichner and Ferrari [15]. Top scored detection are picked from both methods and the pose configuration that appears to be common to both approaches is considered and rest are ignored. Intuition behind the idea is that wrong pose estimates from both algorithms will be very different and the correct configuration of the methods will be the same. In the next step, the pose configuration for all the frames in the database are converted into a representation and in order to achieve this, the angle θ for each part is encoded as an entity pair of $(cos\theta, sin\theta)$. This results in a 12-dimensional vector for the 6 upper-body parts.

Given a query image, a pose Q is obtained using similar strategy of combination of two pose estimation algorithms. The 12-dimensional pose representation V_Q is obtained next. The query representation V_Q is matched against all the representations in the processed database. For a pose representation V_X

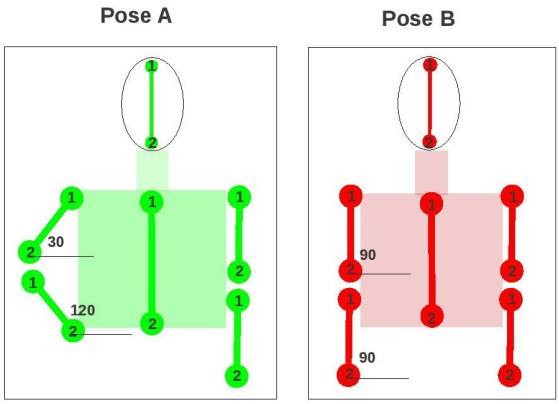


Figure 2.16 Pose representations for two poses A and B which are distinguishable because of the angular feature representation despite bearing similar configurations for most of the parts. Taken from [27].

for instance X in the database and part i, the euclidean distance is computed as:

$$(V_Q - V_X)^2 = \sum_{i \in parts} 2(1 - cos(\theta_Q^i - \theta_X^i)) \tag{2.9}$$

The efficacy of this representation is shown in Figure 2.16 where two poses which differ only at two out of six joints are clearly discriminated based on the angle-based representation. The ranking is then done using the obtained matching distance scores.

2.7 Summary

This chapter familiarizes the reader with popular methods for Human Pose Estimation from different paradigms as well as reviews some essential HPE-based applications . Chapter begins with the introduction to key feature representation: HOG for HPE followed by some methods that utilize these features to generate estimations. First of these is Poselets [4] that works as a part detector for a defined viewpoint in which similar configurations are clustered and trained together to generate part detectors. Next paradigm came with the advent of Pictorial Structure models that represents an object to be composed of articulated smaller parts. Two popular methods from this paradigm [67] and [47] are mentioned and compared. The more recent paradigm involves the utilization of powerful non-linear feature transformation of DNN architecture. We briefly mention two such methods [58] and [10]. We provide quantitative and qualitative analysis of methods from the two paradigms (DPM based and DNN architecture based) that shows the considerable improvement in performance and general applicability DNN architecture based methods provide over older traditional methods. Works demonstrating applications which are directly dependent on human pose knowledge: clothing parsing, action recognition, human-object interaction activities and similar pose retrieval, are described as literature review.

Chapter 3

Human Pose Estimation in Video Sequences

In this chapter, we present our proposed approach that tackles the fundamental problem of human pose estimation for video sequences. The method takes a video sequence of variable size as input and outputs top pose estimations for each frame according to the pipeline. The described method is independent of tracking strategies that gives it an advantage of not getting limited by object smoothness, instead fine-tunes itself with local features from confident frames. To test the method on complicated data, we have collected a dataset of video sequences, all belonging to sports domain from Youtube. Other standard datasets, Poses in the Wild and VideoPose are used to show the general applicability of the procedure. The quantitative and qualitative results demonstrate the robustness and efficiency of our approach.

3.1 Introduction

Over the past few years, we have seen inspiring advancements in different paradigms trying to solve the key problem of human pose estimation from image and video data. Two such major paradigms are based on Pictorial Structures (*e.g.* Yang and Ramanan [67]) and Deep Convolutional Networks (*e.g.* Tompson *et al.* [57]). These models perform well on generic images having considerably comprehensible human pose; however, in videos of complex activities such as sports, such methods turn out to be unreliable. Such models put a higher emphasis on pairwise connections among body-parts and regulate the configuration to follow a generic trend that may be violated in conditions such as playing sports. Videos are generally dealt with tracking strategies like optical flow, SIFT-flow because of possessing redundant temporal information, over a base model that functions at a frame level. However, the generic base models have been observed irregular to be able to produce efficient per-frame estimations that can later be harnessed by tracking strategies. The observed possible complications faced by single image models include occlusions, background cluttering, limb foreshortening, illumination variation *etc*. Moreover, the system settings become restricted when we rely on object tracking methods because of their limitation to track parts that move rapidly, change shape or reappear with a different outlook.

This work focuses on improving individual frame estimations by using temporal information more observantly before handling it to any post-processing smoothness strategy. This results in a formulation that introduces resilience in terms of becoming more independent of part tracks, greater sequence lengths and pose complications. The strategy we are proposing utilizes semi-supervised self-training to fine-tune pose estimations in a video. The self-training model is composed of two components: i) PS-based model that is trained once initially and is used as it is for any arbitrary test video and ii) dynamic ensemble of exemplars model that is progressively augmented with newer examples in a phase-wise manner for each video. The general framework is identical to PS-based framework in which exemplars assist in enforcing part-level appearances to make amends. We will show that this amalgamation satiates the need of strengthening the appearance term which otherwise gets overridden by pairwise score. Other intuition lying behind the idea is that neighboring frames are likely to have similar poses and thus, an exemplar generated from a good estimation can dominantly correct its temporal neighborhood. We use Yang and Ramanan's [67] (YR) model as the base (considering this as a model example of the part-based PS approach to pose estimation) and build up on it iteratively because of its high computation speed and fair reliability. For its efficiency and ease of computation, we use Exemplar-SVM (E-SVM) proposed by [31] to synthesize exemplars from instances. The classifying boundary for an E-SVM is sufficiently taut to disparage far-off indexed instances but remains enough tolerant as to influence a temporal neighborhood in a more direct manner. The eventual self-training framework is sensitive to the correctness of instances picked in each phase. For this purpose, we present a new pose quality ranking criteria that prunes estimations based on their geometric configurations satisfying certain criteria which are presumably valid for any frontal-human pose. Score obtained from this criteria is also used for automatic parameter selection and in post processing as will be shown later.

We present an extensive evaluation of our method on different datasets having arbitrary sequence lengths, as well as varying degrees of part motion and deformation. Poses in the Wild [11] (PIW) and VideoPose [49] (VP) are standard datasets having background clutter and self-occlusion; however, there is minimal camera motion, body deformation or rapid part movements which is present in most of the outdoor videos today. For this purpose, we introduce a new dataset called CVIT-SPORTS having pose configurations that can be called extreme. Quantitative results show that we surpass the state-of-the-art on most of the datasets and lead by a huge margin on CVIT-SPORTS-Videos dataset.

Human pose estimation for two different data types : *images* and *videos* is tackled in different ways because of the nature of the data. In solitary images, the available data is temporally disconnected from past and future events with respect to the moment of picture capturing. Whereas video is a continuous sequence of images in which all the events are connected in the temporal dimension resulting in a certain amount of redundancy. This fact can be exploited in various ways to propagate positive estimations in the neighborhood. Our method works in the same line by synthesizing exemplars from strong positive detections and using them to propagate correctness in the sequence. To determine the strong positive detections we have trained a Pose Quality Ranking SVM that ranks the pose estimations based on their feasibility to occur in the sequence.

3.2 Related Work

There are some past attempts trying to solve human pose estimation using exemplar-based methods. Mori and Malik [36] annotated exemplar 2D views of human pose in different body configurations and camera viewpoints which are used in estimating configuration by shape matching. While testing, these exemplars are used to estimate body joint configuration and pose in three-dimensional space by shape matching. Carlson and Sullivan [6] tackles the problem of action recognition by assuming key frames in a sequence as exemplars which inherit pose information. Such methods fall short because of innumerable combinations of pose configurations and camera viewpoints. Relatively recent models based on deformable parts model give reliable estimations in most of the real-life scenarios. In such models, body is divided into many articulated parts associated with spring-like deformable configurations. First came the attempt by Felzenszwalb and Huttenlocher [18]. Yang and Ramanan [67] considers 'types' for each body part and emphasizes on both human detection and pose estimation by efficient structured learning. Sapp *et al.* [47] present a multimodal approach that captures a wider range of configurations. Dong *et al.* [14] propose a unified framework for human parsing and pose estimation simultaneously. Pons-Moll *et al.* [41] estimate 3D pose by reducing information to boolean geometric relationships among body parts. Ye and Yang [69] estimate pose and shape by embedding deformation model into a gaussian mixture model for single depth images. Initial attempts by Toshev and Szegedy [58] and Ouyang *et al.* [38] have parameterized DeepNet architectures for this problem. Later improvements by Tompson *et al.* [57] and Fan *et al.* [16] have successfully dealt with more complicated configurations.

Most related to our work are methods that estimate pose in a video sequence. Ferrari *et al.* [19] use a spatio-temporal model in between consecutive frames and captures kinematic constraints within a frame resulting in reducing the search space for pose parts. Rohrbach *et al.* [44] try to bring consistency among neighboring estimations based on SIFT-flow information. Ramakrishna *et al.* [42] have presented an occlusion-aware model that uses symmetric parts like shoulders, hands as a supportive information. Another side work by Yu *et al.* [70] infers 3D human poses from frames by spectral embedding and retrieving similar candidates from an exemplar database. Nie *et al.* [37] present a spatial-temporal And-Or model that simultaneously predicts pose estimation and action event. Recent work by Cherian *et al.* [11] mixes body part configurations across the sequence to find best fit detections in each frame with the underlying information provided by optical flow. Pfister *et al.* [40] have used DeepNet architecture by incorporating optical flow information of each part at mid-layer followed by matching neighboring frames predictions across the sequence. Such methods rely heavily on their base model to produce good estimations and fail otherwise, and using object tracking methods put limitations in terms tracking parts facing deformation, high movement as can happen in long videos having considerable human and camera motion. Our approach improves estimations in each frame by transforming confident detections to exemplars and deliberately bringing spatial consistency across the timeline.

Figure 3.1 Frames from 4 different video sequences out of 11 total sequences from CVIT-SPORTS-Videos dataset.

3.3 Dataset, Annotations and Evaluation Metric

We have used a number of datasets for training various models and for testing different set of experiments in this work. Some models are trained initially and are not modified thereafter, whereas dynamic models in the pipeline are trained iteratively and are the focal points for improvements. First the newly introduced dataset with the annotations is described followed by other utilized datasets and the common evaluation metric used for all the experiments.

CVIT-SPORTS. To tackle extreme intricacies involved with human pose estimation, we have picked sports domain. Johnson and Everingham [28] have released Leeds Sports Pose dataset containing 2000 pose annotated images of people playing different sports like badminton, soccer, tennis and volleyball. UCF101 [54], an action recognition dataset is a collection of videos categorized class-wise.The dataset has a certain number of sports classes like cricket bowling, batting, fencing and golf swing with the disadvantage of videos being of low resolution and blurry.

Taking motivation from [28], we present an extremely challenging dataset with high resolution images of human playing different sports like cricket bowling, cricket batting and football. The dataset is

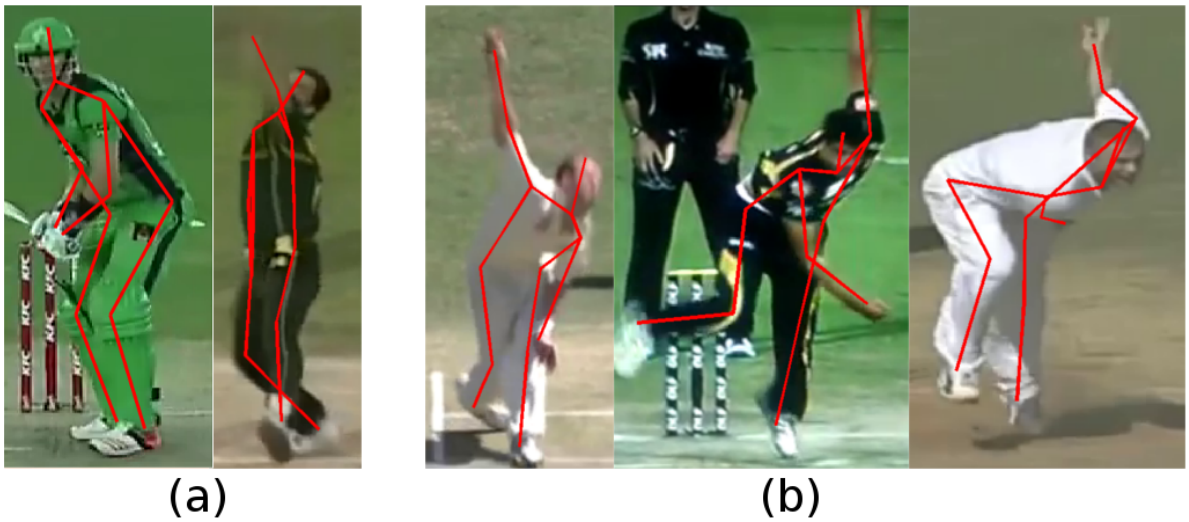


Figure 3.2 Sample frames with ground-truth annotations from CVIT-SPORTS-Videos dataset with complications such as (a) Half body self occlusion and (b) Extreme body deformation.

divided in two parts based on data types *i.e* images and videos. All the frames have complete inclusion of all human body parts of the person-in-consideration irrespective of visible or occluded parts. The complications we face involve high amount of self occlusion, pose variations and extreme body silhouette deformation as shown in Figure 3.2. Most of the recent work revolving around video dataset tracks only upper human body joints, whereas we have incorporated all human body joints in our experiment settings.All the frames in the complete dataset have been annotated with 14 keypoints *i.e* full human pose. From this 14 set, we automatically generate 26 keypoints set that encompasses all the major joints as well as intermediate points useful for tracking like points between shoulder and elbow, elbow and hand, hip and knee, knee and ankle. We use the 26 keypoint set for all of our full body experiments.

1. *CVIT-SPORTS-Videos* This set has a total of 11 videos of a human playing sports retrieved from Youtube. We have included intricate domains like cricket-bowling, cricket-batting and football-kicking. In total, this set has a total of 1446 frames averaging out to 131 frames per video.

2. *CVIT-SPORTS-Images* This set has 698 images, all belonging to cricket-frontal-bowling class. It has 150 random cricket-bowling-action images retrieved from Google. Rest of the images are random frames of cricket-bowling-runup and cricket-bowling-action from different videos. These images are used to get negative examples for training E-SVM in all the experiments.

For both sets, in cases where most of the body parts are not obviously visible, we have annotated the frames to the best of our comprehensibility by looking at previous and future frames in the sequence.

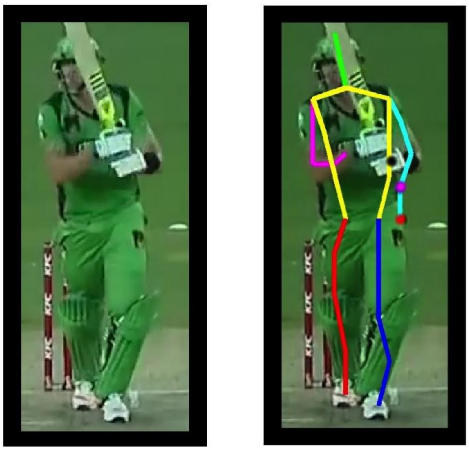
Image-PARSE. [43] This dataset has been used to train and test full-human parsing models. It has a total of 305 images containing people and is annotated with full-body pose. We use the dataset to train the base model YR-PARSE.

FLIC. [47] For comparison experiments, we have used this human upper-body labeled dataset of around 4.5K samples to train the base model, YR-FLIC.

VideoPose [49] This dataset consists of 17 test videos from TV shows with an average sequence length of 30 frames. The dataset is used to assess upper-body part estimations from different part localization algorithms.

Poses in the Wild. [11] This recently introduced dataset consists of 30 video sequences of approximately 30 average frame length trimmed from different Hollywood movies. The dataset has real-life scenes having some amount of camera and human motion and all the frames have been annotated with human upper-body joints. The main purpose this dataset serves is to determine the localization accuracy of upper-body limb parts like shoulders, elbows and wrists.

Evaluation Metric. We have used the Percentage of Correct Keypoints (PCK) evaluation metric proposed by [67] everywhere in this text. PCK terms only those estimated keypoints correct which lie within a pixel threshold distance of D_T from their counterpart in ground truth. The threshold distance is determined by the formula : $D_T = \beta.max(h, w)$,
where h and w are the height and width of the tight bounding box created around the ground truth pose and β is a threshold controlling the relative correctness magnitude. β is set to 0.1 when dealing with full-body poses and to 0.2 when only half-body poses are considered.



(a) Original Image

(b) Skeleton detected by YR method

Figure 3.3 (Best viewed in color) The dominance of the pairwise potential over the unary causes poor performance of [67] on complex unseen poses. The red dot (C) denotes the detected wrist; the black dot denotes the ground-truth wrist (G); and the pink dot denotes the parent of the wrist (P).

3.4 Observations and Assumptions

1. **Pairwise over-rides unary response to make false predictions.** Using generalized models such as [67] for complex poses and long video sequences does not yield acceptable performance, often due to the dominance of the pairwise potential in the random field which is influenced by the configurations seen while training the model. An example of this issue is shown in Figure 3.3. This image illustrates the complex pose when there is a foreshortening of limbs, in this case of that from elbow to wrist. In [67]'s 26-joint human model, there are 3 joint positions from the elbow to the wrist, viz. elbow, mid-joint (between elbow-wrist) and wrist. The parent, P, in the figure corresponds to the mid-joint. Due to the foreshortening of the limb, the mid-joint will not be detected, and the pairwise potential forces the wrist to be placed at location C, instead of G, to ensure inter-part compatibility (observing the unary and pairwise potential values corroborates this claim too). We note that a similar observation of unary potentials being dominated by pairwise potentials at the boundaries was made by Horne *et al.* [25] recently , although their work was in a completely different context.

2. **Slight variation in image features changes the top estimation.** It has been observed that generic models generate very different estimations for consecutive frames in the sequence. Consecutive frames vary very slightly in terms of features but the estimation discrepancy represents the sensitivity of such models.

3. **Tracking is not reliable.** Tracking strategies like TLD, optical flow are restricted by the amount of object movement and image feature changes they can handle. Moreover, tracking body-parts that are ever-changing in outlook exacerbates the situation.

3.5 Fine-tuning Pose Estimation using Semi-Supervised Self-Training

Our studies have shown that failure cases on complex human pose data, even when from the same domain, often occur because: (i) unary potentials may be predicted poorly due to the fact that the appearance terms defining a part may not strongly resemble the appearances of the same part as seen earlier, or (ii) the unary potential is subdued by the pairwise potential (of already seen configurations), when a new configuration of parts is encountered. Hence, in this work, we propose to fine-tune pose estimation for complex poses using a semi-supervised self-training approach that strengthens the unary score of parts with respect to the newer video without additional labeling effort. We achieve this objective using three steps: (i) a pose quality ranking SVM that identifies the quality of the pose detected by a base model on a given image; (ii) a coarse-to-fine strategy that uses the pose quality scores to identify suitable exemplars for self-training; and (iii) an ensemble of exemplar SVMs for semi-supervised self-training to improve the base model from [67]. While the proposed method has been studied with [67], the ideas behind each of the steps are independent of the method, and can easily be integrated with other part-based models for human pose estimation.

Below we will begin with describing the base model we have used in the pipeline which follows the mentioned three steps in our self-training pipeline and the post-processor used.

3.5.1 Base Model

Similar to [67], we denote I for a video frame, $p_i = (x, y)$ for the pixel location of part i and part type t_i for the mixture component of part i, where $i \in \{1, \cdots, K\}$, $p_i \in \{1, \cdots, L\}$ and $t_i \in \{1, \cdots, T\}$. $G = (V, E)$ is a K-node relational graph (in our case, a tree) as in Figure 3.4) whose edges specify the constraints of the pairwise configuration of parts. The full score for a configuration of part types and positions in a video frame is then given by:

$$S(I, p, t) = S(t) + \sum_{i \in V} w_i^{t_i} \cdot \phi(I, p_i) + \sum_{ij \in E} w_{ij}^{t_i, t_j} \cdot \psi(p_i - p_j) \tag{3.1}$$

where $\phi(I, p_i)$ is a feature vector (such as HOG) and $\psi(p_i - p_j) = [dx \quad dx^2 \quad dy \quad dy^2]^T$. $dx = x_i - x_j$ and $dy = y_i - y_j$ are the relative positions on the pixel grid. $S(t) = \sum_{i \in V} b_i^{t_i} + \sum_{ij \in E} b_{ij}^{t_i, t_j}$ where b_is are priors on part types and b_{ij}s are parameters that weight co-occurrences of part types. Evidently, the first sum in Eqn 3.1 is an appearance model that computes the score of a template $w_i^{t_i}$ for part i, tuned for type t_i, at location p_i. The second sum is a spring model that influences the relative placement of p_i and p_j, parameterized by $w_{ij}^{t_i, t_j}$. Inference is carried out by maximizing $S(I, p, t)$ over p and t. Considering the graph G is a tree, this is achieved efficiently using message passing [67].

The factor graph for the above model is illustrated in Figure 3.4, with 26 parts of the human body finally used to model the human pose.

$$p(L|O, \Theta) = \frac{1}{Z(O, \Theta)} \exp\left(-\sum_{i=1}^{26} E^u(I_i, O, \Theta) - \sum_{i\ j} E^p(I_i, I_j, \Theta)\right) \tag{3.2}$$

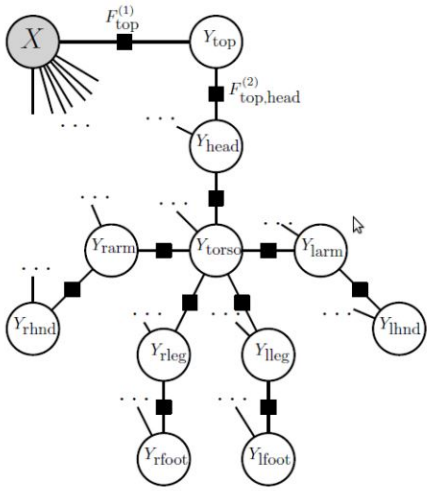


Figure 3.4 Illustration of factor graph (tree structure) for model proposed in [67]

where O is the observed image, Θ are the model parameters, I_k denotes the feature representation of the k^{th} part, $E^u(I_i, O, \Theta)$ are the unary potentials for each of the 26 parts (the appearance models for each part), and $E^p(I_i, I_j, \Theta)$ denotes the pairwise potential between the i^{th} and j^{th} parts.

We use the PS-based model proposed by [67] as our base model and the iterative procedure thrives on it. For full human-body experiments, the base model YR-PARSE is trained on 305 images from Image-PARSE [43] dataset. Number of human body parts considered are 26 and number of types for each part is taken to be 6. Similarly, for human upper-body experiments, YR-FLIC base model is trained on FLIC [47] dataset of around 4.5K labeled images. Types for each part is again 6, whereas total number of body parts are only 18.

Figure 3.5 YR baseline estimations on a set of frames from sequence V1. These detections are obtained in phase 1 of our pipeline.

Method	PCK(%)
YR-PARSE [67]	62.67
FT @ phase 2	65.97
FT-Full	67.83
FT-Full + PP	67.95

Table 3.1 Performance of our approach with different settings on example video V1 having 99 frames. Last row shows the post-processor (PP) improvement over the final phase of our full fine-tuning (FT-Full) model.

We consider video sequence V1, having 99 total frames belonging to cricket-bowling domain, as our primary example to show different experiment settings and their eventual impact. Table 3.1 shows YR baseline gives average PCK score of 62.67% on V1. Figure 3.5 shows YR estimations on frames picked from different time indexes and clearly stipulates its ineffectiveness to track body-parts when faced with occlusion, part foreshortening and other complications.

3.5.2 Pose Quality Ranking SVM

Given a trained base model, self-training refers to the process where the given model is trained iteratively on its own output on newer data. In a typical self-training setting, the model trained on a given dataset is applied on newer data, and the data with the most confident labeling from the new data is chosen with respective labels to retrain the original model. Identifying data with the most confident labels is, however, not straight-forward when dealing with human pose labels. Our initial experiments also showed that using the highest output scores of the base model does not directly translate to pose quality, especially when complex poses are present in the new video sequence (similar to the illustration in Figure 3.3).

Hence, in this work, we use the characteristics of the geometric configuration of the pose detected on a newer video by the base model to obtain a pose quality ranking, which can then be used to select suitable data instances for self-training. Table 3.2 shows the configuration characteristics of the detected pose that are used for this purpose. Items 1-18 are binary variables that indicate the presence of certain configurations that indicate errors in the pose (for instance, the "shoulder swap" binary variable indicates that the detected position of the right shoulder is to the left of the detected position of the left shoulder - which is typically an error, barring exceptional settings). Items 19-20 are real-valued variables that provide the pose scale, and the average half-angle between the parts. Characteristics which are suffixed with N in Table 3.2 indicate that these items are compared with a local temporal neighborhood of the video sequence for outlierness. If the values significantly differ from the local temporal neighborhood, the corresponding binary variable assumes value 1. Figure 3.7 illustrates scenarios of improbable pose configurations, which motivate the use of the aforementioned characteristics to rank the detected pose quality.

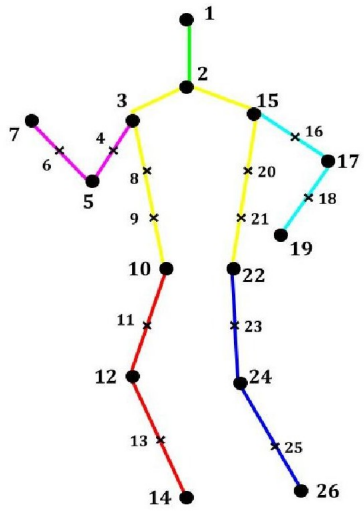


Figure 3.6 Pose configuration $pose_i$ with labeled 26 body-joints.

Consider a full human pose indexed i denoted $pose_i$, having 26 parts as shown Figure 3.6 and each part denoted as $pose_i(j)$, where $j \in [1, 2, ..26]$. Similarly x/y coordinates of part j is $pose_i(j, x/y)$. Neighborhood of $pose_i$, $nbd(pose_i)$ corresponds to mean of pose/pose-parts of pose configurations belonging to index [i-3:i-1, i+1:i+3]. Euclidean distance between part a and b is referred to as $eucl\_dist(a, b)$. Feature vector is constructed as follows:

If following conditions are violated, feature vector is appended with "-1", otherwise "1". Features tagged with "N" are compared with the neighborhood pose configuration entities.

- Left-Right shoulder swap: $pose_i(3, x) <= pose_i(15, x)$
- Left-Right hip swap: $pose_i(10, x) <= pose_i(22, x)$
- Left-Right torso parts intersection: $pose_i(a, x) <= pose_i(b, x), a \in [8, 9, 10], b \in [20, 21, 22]$
- Unlike Left-Right torso lengths: $ratio = left\_torso\_length/right\_torso\_length$
  , and $0.75 >= ratio <= 1.25$.
- Converging torso width from top to bottom:
  $euc - dist(pose_i(3), pose_i(15) <= euc - dist(pose_i(8), pose_i(20)$.
- Converging torso width from top to bottom:
  $euc\_dist(pose_i(8), pose_i(20) <= euc\_dist(pose_i(9), pose_i(21)$.
- Left torso length (N): $ratio = left\_torso\_length/nbd(left\_torso\_length)$
  , and $0.75 >= ratio <= 1.25$.

- Right torso length (N): $ratio = right\_torso\_length/nbd(right\_torso\_length)$
  , and $0.75 >= ratio <= 1.25$.
- Left leg length (N): $ratio = left\_leg\_length/nbd(left\_leg\_length)$
  , and $0.75 >= ratio <= 1.25$.
- Right leg length (N): $ratio = right\_leg\_length/nbd(right\_leg\_length)$
  , and $0.75 >= ratio <= 1.25$.
- Left arm length (N): $ratio = left\_arm\_length/nbd(left\_arm\_length)$
  , and $0.75 >= ratio <= 1.25$.
- Right arm length (N): $ratio = right\_arm\_length/nbd(right\_arm\_length)$
  , and $0.75 >= ratio <= 1.25$.
- Left hip location (N): $euc\_dist(pose_i(10), nbd(pose_i(10))) <= 0.25 * bounding\_box\_height$.
- Right hip location (N): $euc\_dist(pose_i(22), nbd(pose_i(22))) <= 0.25 * bounding\_box\_height$.
- Left shoulder location (N): $euc\_dist(pose_i(3), nbd(pose_i(3))) <= 0.25 * bounding\_box\_height$.
- Right shoulder location (N): $euc\_dist(pose_i(15), nbd(pose_i(15))) <= 0.25 * bounding\_box\_height$.
- Left and Right shoulder distance (N):
  $ratio = euc\_dist(pose_i(3), pose_i(15))/euc\_dist(nbd(pose_i(3), pose_i(15)))$
  , and $0.75 >= ratio <= 2.0$.
- Left-shoulder to neck to Right-shoulder traversal distance (N):
  $travdist\_i = euc\_dist(pose_i(3), pose_i(2)) + euc\_dist(pose_i(2), pose_i(15))$
  $nbd\_travdist\_i = euc\_dist(nbd(pose_i(3), pose_i(2))) + euc\_dist(nbd(pose_i(2), pose_i(15)))$
  $ratio = travdist\_i/nbd\_travdist\_i$, and $0.75 >= ratio <= 1.5$.
- Pose scale (N): $ratio = height(pose_i)/height(nbd(pose_i))$, and $0.75 >= ratio <= 1.25$.
- Half parts angle: Angle of a part j is the shorter angle between lines connecting parent of part to part and part to its child. $angle(pose_i(a)) <= 30\,\text{deg}$, $a \in [4, 6, 11, 13, 16, 18, 23, 25]$.

Given the aforementioned characteristics, a linear Support Vector Regressor is trained to finally provide a pose quality ranking score. This score is used for selecting instances for self-training, as described in subsections below.

The SVM is trained similar to a linear incremental-SVM with the weights initialized from the domain knowledge of frontal-human-pose. Features as mentioned in Table 3.2 are obtained from a sample video sequence and are used to train the desired SVM. For upper-body experiments, a separate SVM is trained by ignoring features that correspond to lower body-parts.

1. L-R shoulder swap	2. L-R hip swap
3. L-R torso parts intersection	4. Unlike L-R torso length
5/6. Converging torso width from top to bottom	7/8. L and R torso lengths (N)
9/10. L and R legs length (N)	11/12. L and R arms lengths (N)
13/14. L and R hip location (N)	15/16. L and R shoulder location (N)
17. L and R shoulder distance (N)	18. L-shoulder to neck to R-shoulder traversal distance(N)
19. Pose scale (N)	20. Half parts angle

Table 3.2 Our Pose Quality Ranking-SVM is trained with features designed using above criteria. Entries with N tag are compared with mean neighborhood entries of the same type. l-r denotes left-right.

3.5.3 Coarse-to-Fine Strategy for Exemplar Selection

Instead of directly using the obtained pose quality scores for selecting exemplars (video frames), we employ a coarse-to-fine strategy over the temporal resolution of the new video sequence to ensure representative data instances are picked from different temporal segments of the video sequence for self-training. This is achieved using a two-step process:

- In the first step, temporal neighborhoods with average pose quality scores that are greater than the mean pose quality scores are chosen. Among these high-ranked neighborhoods, the neighborhoods that have the highest mean output scores from the base method are selected for further processing. This provides a coarse selection of potential exemplars.

- In the second step, in each of the selected neighborhoods, the specific exemplars with pose quality scores greater than the mean pose quality scores of the neighborhood are selected, with a suitable threshold limiting the maximum number of exemplars that can be picked from one temporal neighborhood.

This coarse-to-fine strategy was primarily employed to address scenarios when the highest pose quality scores are all in the same temporal neighborhood of the entire video sequence, which if selected can lead to poor and biased self-training results. This simple multi-resolution approach significantly improved the self-training performance. We also note that where there are ties in the pose quality score, we use the output score of the base model to break the tie.

In our settings for coarse-to-fine exemplar selection strategy, the number of neighborhoods to be processed in each phase is dependent on a combination of pose quality score and method score where each neighborhood is restricted to a maximum size of 10 frames. Threshold value for both scores is taken to be the mean score of neighborhoods in consideration.

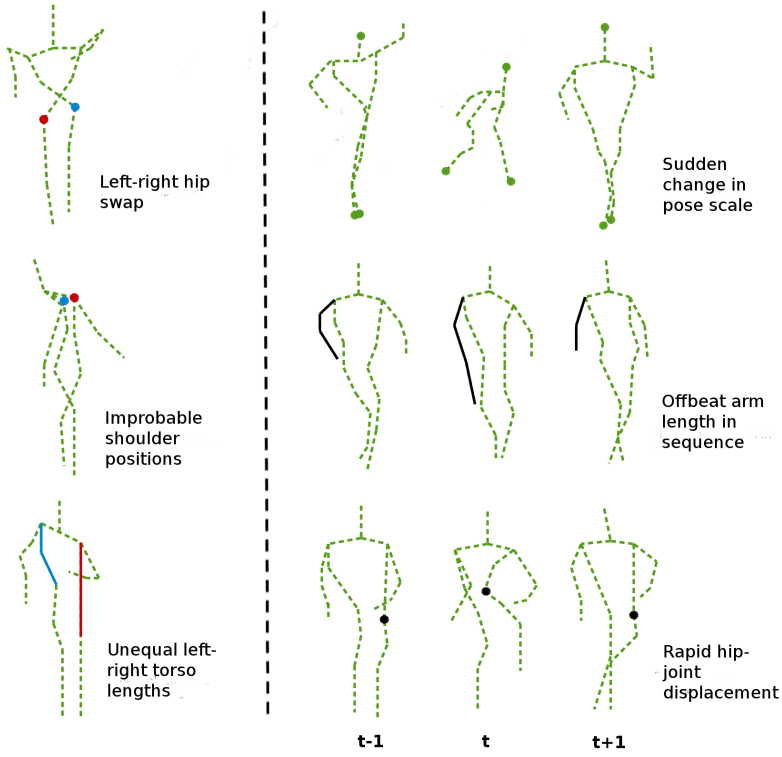


Figure 3.7 Examples of some characteristics of configuration of pose detected by base model used to rank quality of pose estimations in a video sequence. Left column shows improbable estimations at a single image level and right column shows eccentric behavior in a temporal locality of estimations.

From chosen neighborhoods, at max 3 instances are picked using pose quality score and are converted into E-SVM. η scalar defining the unary contribution from YR and E-SVM in equation 3.3 is automatically picked from 11 possible values (ranging from 1 to 0 with 0.1 step size) using summation of pose quality scores for the complete video. From the observation that higher η value in phases ensures long-run consistency, we pick η which gives us the first local maxima. A video is processed iteratively until all the neighborhoods have been exhausted or pose quality score is rendered useless which can happen when pose quality score for reducing η values does not decrease and we get deprived of any local maxima.

3.5.4 Ensemble of Exemplar SVMs for Semi-Supervised Self-Training

Given a set of exemplars that capture the best performance of the base model, we now describe how self-training is performed using this set. Self-training has been used with Conditional Random Fields (CRFs) successfully earlier [9][55] in natural language processing tasks, either by using the most confidently classified data from the new set in a SoftMax SVM formulation, or by simply using the

probabilistic outputs. However, to the best of our knowledge, self-training with CRFs has not been attempted earlier where structural SVMs are used, which poses unique challenges as in the case of human pose estimation. Besides, many existing self-training methods tend to reinforce the knowledge of the base supervised model. Hence, in this work, to overcome these issues and to strengthen the unary scores in the base model, we use the identified exemplars to train an ensemble of exemplar SVMs [31] for each part individually. Hence, each selected exemplar-frame leads to the training of 26 exemplar SVMs corresponding to each of the parts.

As in [31], exemplar SVMs are based on the simple idea to train a classifier for each exemplar in the dataset, and then calibrating the output scores of these exemplar SVMs to obtain a final classifier. Each exemplar SVM, $(\mathbf{w}_E, b_E)$, tries to separate an exemplar $\mathbf{x}_E$ from a set of negative examples, N_E, by the largest possible margin. This is achieved by learning the parameters $(\mathbf{w}_E, b_E)$ that optimize the following convex objective:

$$\min_{(\mathbf{w}_E, b_E)} ||w||^2 + C_1 h(\mathbf{w}^T \mathbf{x}_E + b_E) + C_2 \sum_{x \in N_E} h(-\mathbf{w}^T \mathbf{x} - b)$$

where h stands for a loss function such as the hinge loss $h(x) = \max(0, 1 - x)$. This optimization is solved as in [31], for each of the 26 parts from the selected exemplar-frames in our case. Negative examples are randomly chosen from non-part windows across the video sequence.

Instead of calibrating the resulting ensemble of exemplar SVMs (obtained from the parts of all the selected exemplar-frames) using the procedure in [31], we instead propose a different approach to integrate these exemplar SVMs into the CRF model of the base method [67]. In Equation 3.1 of the base model, we redefine the unary potential term using the exemplar SVMs as follows:

$$\sum_{i \in V} (\eta w_i^{t_i} + (1 - \eta) \hat{w}_i^{t_i}) \cdot \phi(I, p_i) \tag{3.3}$$

where $\hat{w}_i^{t_i}$ are the (normalized) weights learnt from the exemplar SVM for the i^{th} part tuned for type t_i, and η is a parameter that controls the weight given to the exemplars' contribution to the unary score. This unary score is then integrated into the CRF inference in [67] to obtain a new updated model that can be used for detecting poses on the (rest of the) newer video. This self-training approach provides us a seamless way not only to integrate the ensemble of exemplar SVMs into the base model, but also automatically addresses the issue of calibration in the ensemble. We note additionally that unlike other self-training methods which tend to reinforce the knowledge of the base model, the freedom in the choice of negative examples in this ensemble-of-exemplar-SVMs approach allows us to ensure newer knowledge is added to the previously learned model, rather than just reinforce old knowledge.

Each instance picked using our batch selection criteria is converted into a set of E-SVM filters. Since a full body is divided into 26 parts, we procure a set of 26 E-SVMs for every picked frame and for upper body setting, a set of 18 E-SVMs is attained. E-SVM for a part is trained as a linear SVM with the part features as the only positive and 2000 negatives belonging to either background or part classes not belonging to the current part class being trained. Negative 2000 samples for each part are picked randomly out of many more possible candidates from CVIT-SPORTS-Images set.

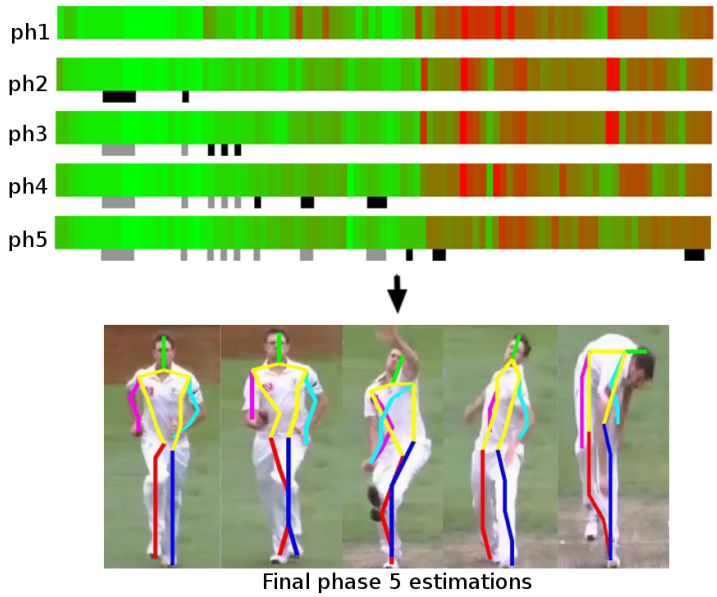
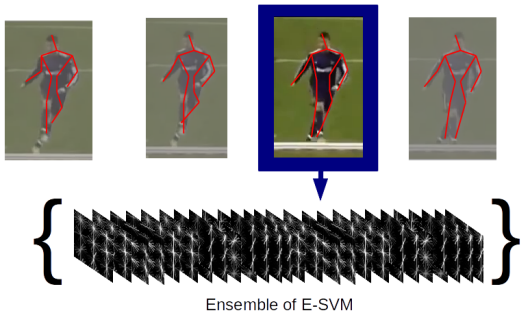


Figure 3.8 (Best viewed in color) Phase-wise PCK performance on sequence V1 showing the improvement trend as we iterate. Transition of color from green to red represents transition from correctness to false estimations. Indexes marked with black are instances converted into E-SVM and used for testing in the same phase, whereas gray-marked indexes are exemplar instances from previous phases.

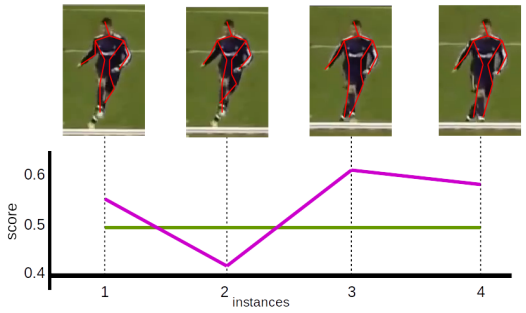
In self-training, a solitary base model is generally assumed whose labeled set is updated with *probable* good detections from the unlabeled set and is re-trained. After running the first phase using base model, we pick instances and synthesize E-SVM from them. Video length is sub-divided into N_t number of neighborhoods and using our coarse-to-fine selection strategy, we pick 3 instances from each neighborhood finally chosen. Generated E-SVMs are augmented in the ensemble and with the updated ensemble and YR base model, we run equation 3.3 across the whole sequence. Same procedure is followed for subsequent phases. The impact of iterations are more explicitly shown in Figure 3.8 and table 3.1 shows our 5.16% gain over the baseline on sequence V1 by our full fine-tuning (FT-Full) model.

3.5.5 Post-processor: Neighborhood Interpolation

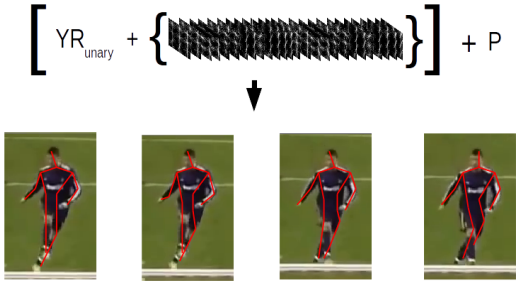
After getting estimations from a phase decided by the stoppage criteria, we run a simple post-processor (PP) that uses information from pose quality scores. For an index, if its pose quality score is less than both of its neighboring indexes we choose the instance for post processing. Estimation of the picked instance is replaced by the mean interpolation of its neighboring estimations *i.e.* $est_i = mean(est_{i-1}, est_{i+1})$. Table 3.1 shows an improvement of 0.12% PCK on our sample video sequence V1 over our FT-Full model and a cumulative 5.28% PCK gain over the baseline.



(a) Using pose quality score for batch selection, instance 3 is picked and is converted into an ensemble of E-SVM of length equal to number of body-joints.



(b) (Best viewed in color). Base model on consecutive frames with pose quality score (green) and base method's output score (red). Where there are ties among the pose quality score, the base method's output score is considered to break the tie.



(c) The procured E-SVMs are incorporated in our method equation 3.3 and used to infer on all frames which leads to correction in previously wrong localization.

Figure 3.9 Illustrative overview of proposed semi-supervised self-training methodology.

Method	PCK(%)
YR-PARSE [67]	72.41
FT @ phase 2	74.31
FT-Full	74.19
FT-Full + PP	**74.74**

Table 3.3 Full Performance on CVIT-SPORTS-Videos dataset with variants. The pose setting considered is full 26 human body-joints.

Method	CVIT-SPORTS-Videos	PIW	VP
YR [67]	64.87	70.74	63.35
[11]	42.12	71.43	**71.95**
YR + FS [44]	31.75	48.04	60.01
FT @ phase 2	**65.20**	72.78	69.11
FT-Full	64.90	72.44	68.65
FT-Full + PP	64.50	**73.26**	68.96

Table 3.4 Comparison results: Performance comparison using PCK measure on CVIT-SPORTS-Videos, PIW [11] and VP [49]. The pose setting used includes 6 upper-body joints.

3.6 Experimental Results

Diagnosing the Model. Table 3.3 shows PCK evaluation of our complete iterative method along-with variants on our CVIT-SPORTS-Videos set. The model being used estimates full 26 human body-joints in each frame. A performance improvement of 1.90% is achieved after running the first phase of combined model: YR and E-SVM. Full model shows net improvement of 1.78% and our post-processor (PP) stretches the lead to 2.33% over the YR baseline.

Comparisons. The method presented in this work is compared with recent well-performing methods tackling the problem of human pose estimation in videos on three datasets: i. CVIT-SPORTS-Videos ii. PIW[11] and iii. VP[49]. Cherian *et al.* [11] uses optical flow to process single frame estimations and Rohrbach *et al.* [44] work on the same line by using SIFT-flow. We use the codes provided by the authors of both. The comparison evaluation is done on the upper-body joints as previous methods use this setting as shown in Figure 3.11 (a) and (b).

For comparisons on CVIT-SPORTS-Vidoes dataset, we have used the upper body-parts detections from our full model trained on PARSE [43] dataset. Table 3.4, first column demonstrates our lead over previous approaches on this extremely complicated dataset using PCK evaluation.

For standard datasets PIW [11] and VP [49], we have trained a new base model, YR-FLIC trained on FLIC [47] dataset similar to [11]. Table 3.4, second and third columns shows 2.02% and 5.76% improvements over the baseline by our method at phase 2 (FT @ phase 2) of the iterative procedure.

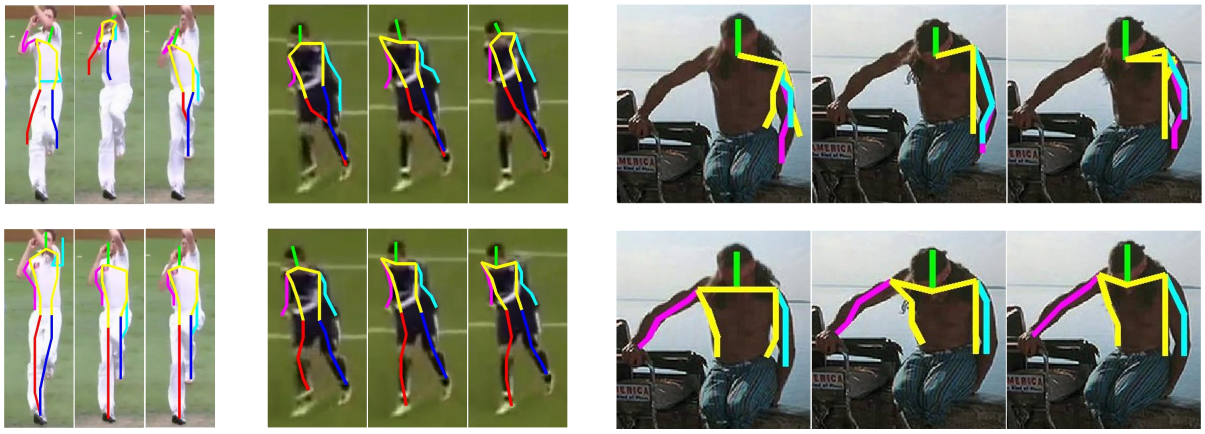


Figure 3.10 Top row shows the estimations with YR [67] and second row shows the corrections being done by our FT-Full model. First two columns are sequence of frames from CVIT-SPORTS-Videos dataset, whereas last column is from PIW [11] dataset.

Performance deteriorates when we run all the phases, although surpasses previous baseline for PIW dataset.

3.7 Discussions

Our self-training method iteratively strengthens positive unary responses across temporal sequence without any external manual intervention, which the baseline loses because of greater false pairwise influence. The whole iterative procedure is sensitive towards the quality of exemplars synthesized at each iteration making the task of determining good quality exemplars of prime significance.

Failure Cases. In shorter videos like in Poses in the Wild [11] dataset, very few exemplars can prominently influence the complete sequence in either good or bad way. Failure in grabbing correct estimations results in influencing neighborhoods inadequately as shown in figure 3.11 last column.

Parameter Selection issues. Choosing optimal parameters in each phase is extremely relevant otherwise, the improvement across the iterations reduces to a minimal amount. We automatically determine the parameters using pose quality score in a manner determined by cross validation.

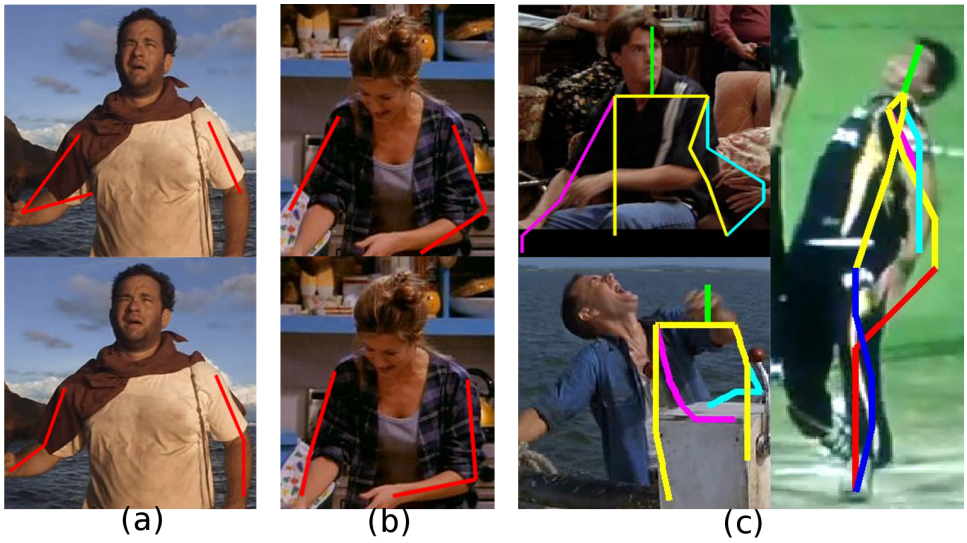


Figure 3.11 Comparison and Failure cases: Top showing estimations by [11] and bottom shows our FT-Full results on (a) PIW and (b) VP dataset. (c) Failure cases from three datasets used.

3.8 Summary

We have presented a self-training approach tackling the problem of Human Pose estimation in videos which instead of directly relying on the baseline's predictions, empowers response enabling us to capture intricate pose configurations. We have also presented a pose quality criteria to pick instances in each iteration which also assists in automatic parameter selection and functions as a low-level pose evaluator. The setting thus obtained surpasses the previous state-of-the-art on standard Poses in the Wild [11] dataset and on our introduced Our-Sports-Videos dataset.

Chapter 4

Cloth Parsing: An application

Parsing for clothes in images and videos is a critical step towards understanding the human appearance. In this work, we propose a method to segment clothes in settings where there is no restriction on number and type of clothes, pose of the person, viewing angle, occlusion and number of people. This is a challenging task as clothes, even of the same category, have large variations in color and texture. The presence of human joints is the best indicator for cloth types as most of the clothes are consistently worn around the joints. We incorporate the human joint prior by estimating the body joint distributions using the detectors and learning the cloth-joint co-occurrences of different cloth types with respect to body joints. The cloth-joint and cloth-cloth co-occurrences are used as a part of the conditional random field framework to segment the image into different clothing. Our results indicate that we have outperformed the recent attempt [64] on H3D [5], a fairly complex dataset.

4.1 Introduction

Cloth parsing involves locating and describing all the clothes (*e.g*, T-shirt, shorts) and accessories (*e.g.* bag) that the person is wearing. As it describes the human appearance, it is a next step to human detection and pose estimation in understanding the images. It plays an important role in human pose estimation [64], action recognition, person search [21, 63], surveillance, cloth retrieval [30] and has applications in fashion industry [64]. Commercially, it can be used in online cloth retail portals where people can try out various clothes. The main challenges in solving this include the large variety of clothing patterns that have been developed across the globe by different cultures. Even within the same cloth type, say T-shirt, there is a significant variation in color, texture and other complicated patterns. Occlusions from other humans or objects, viewing angle and heavy clutter in the background further complicates the problem.

In our work, we aim to segment clothes in unconstrained settings. We handle the the diverse set of challenges by modelling the cloth appearance and its vicinity to a body part. While human pose estimation algorithms [15, 48, 2, 65] provide the body part configuration, they frequently fail and give wrong pose estimates when there are large occlusions and heavy clutter. To handle these challenges, it is

more desirable to have a probabilistic estimate of the body part position than to have single, deterministic but a wrong pose estimate. Poselets [5] offers one such flexibility where they detect a combination of body parts. In this work we adapt poselets to locate human joints and model the cloth-joint co-occurrences by learning a function that assigns a cloth label based on the relative location with respect to joints. Since the neighboring regions which have similar appearance share the same label, we use the conditional random fields to segment different cloth labels in the image. In section 4.6, we explore strategies to improve upon the achieved performance. Our main goal is to obtain estimations that are reliable in terms of practical usage.

4.2 Related Work

In the recent past, there has been considerable attention given to understanding and modelling clothes. Many methods [65, 12, 24, 50, 62] have been proposed to segment clothes in restrictive settings. In [65], cloth recognition is proposed for surveillance videos where the camera angles are fixed and background subtraction can be effectively used for human detection. In [12], a real time upper body cloth segmentation is proposed in images where people are wearing a monochromatic clothing and printed/stitched textures. In [24], the method models a particular cloth combination accounting for color and shape. The method uses the Markov random field (MRF) and tries to incorporate shape priors in inferring location and labels of clothes. While the method shows good invariance to pose changes, it is restricted to a particular set of cloth combinations. In [50], only the regions around the face like skin, hair and background are segmented. Using the Bayesian framework, the class color distribution is learnt and then adapted to a particular image. A spatial prior is also incorporated. In [62], multi-person clothing segmentation algorithm is proposed. Given an image with multiple people occluding each other, the method is restricted to segmenting the upper cloths.

Cloth segmentation and modelling has been used in many interesting applications. Both [21, 63] use cloth segmentation as a strong cue for person identification and retrieval. Liu *et al.* [30] build a cloth retrieval application. Given an image of person on a street, the method aims to retrieve people with similar clothes from the dataset. First the body parts are detected and then features are extracted. Body parts are sparsely reconstructed from the previously available "street dataset" which are in-turn sparsely reconstructed from an online shopping dataset. Chen *et al.* [8] propose a sophisticated And-Or graph based graphical method to model and sketch a cloth. Given an image of the person, the algorithm automatically sketches her outlines which resemble a portrayal by a human. For training, artist drawn sketches as ground truth and edge maps of a person as data are provided. While the algorithm has excellent results on very simple images, their performance in difficult setting is not demonstrated.

Our work is closest to that of Yamaguchi *et al.* [64]. They have proposed a method to parse clothes in fashion photographs. In their work, cloth parsing problem is viewed as an object segmentation using CRFs. While using segmentation algorithms is quite standard and has been applied earlier [56], their innovation comes in defining the unary potential. They use a pose estimation algorithm [66] to model a

Figure 4.1 Unsupervised segmentation algorithms [52] (column 2) cannot differentiate among the body parts (like skin, hair), clothes and background. By appropriately modelling colors and shape priors, the results improve significantly (column 3 and 4)

cloth type. Unlike our work, Yamaguchi *et al.* [64] have limited themselves to restricted settings where a single person is standing upright against a relatively simple background. Real images on the other hand are far more complex containing multiple people present against a complicated background and potentially occluded.

4.3 Robust Pose Information

Several segmentation algorithms have been proposed which differ in determining similarity measures in neighboring image regions and structure of the graph. To effectively capture the variation in clothing patterns both the appearance and human pose have to be modelled. To demonstrate this, we ran two classes of segmentation algorithms on an image. Figure 4.1 displays the segmentations from various algorithms. First an unsupervised algorithm [52] is run on the image. These class of algorithms neither model the appearance of target class nor any inherent structure. They base the segmentation on the similarity in low-level image features like color and texture. As can be seen in Figure 4.1 (column 2) the algorithm completely fails. The generic semantic segmentation algorithms on the other hand use the supervision and model the appearance and a generic structure across the object classes, while for objects like trees and cars the structure is rigid. Next, two supervised algorithms are run which model both the appearance model of different clothes and the human body pose. These of course fair very well (column 3 and 4 in Figure 4.1).

For modelling the clothes in unrestricted settings a robust human joint estimator is needed as clothes are worn around human joints. We illustrate this point by running our cloth segmentation algorithm with a correct and an incorrect pose estimate as input. Figure 4.1 clearly demonstrates that the difference is very stark. Hence modelling the structure should be specialized. Figure 4.1 displays the output from various segmentation algorithms. As illustrated, modelling appearance and structure significantly

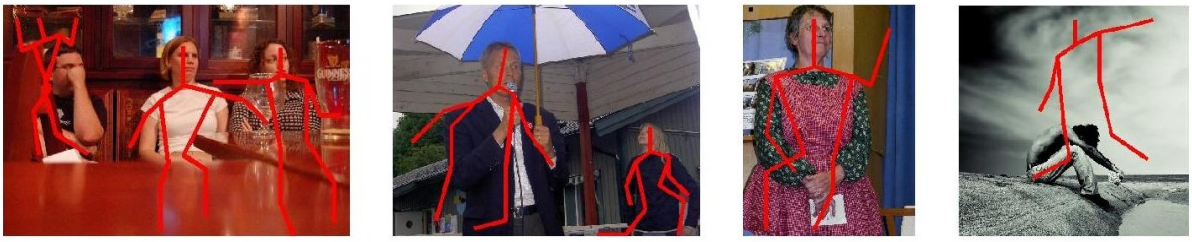


Figure 4.2 Pose Estimates [66] on H3D dataset: In the images containing severe occlusions and clutter, pose estimation fails.

improves the performance. The information, as in RGB intensities of pixels, required to model the appearance is readily available. In case of human body pose though, the information has to be obtained from other strategies that are highly robust, interpret-able and reliable. The popular choice to obtain human body part configuration Z are human pose estimation algorithms [15, 48, 2, 66]. Unfortunately the standard pose estimation algorithms fail to detect occlusions. This is mainly because the top-down model of pose estimation algorithms cannot model occlusions effectively. Figure 4.2 displays the output of pose estimation algorithm [66] on images with large occlusions.

We therefore employ poselets [5] which are human body part detectors. Poselet models a combination of human body joints (*e.g*, face, shoulders and background in row 1 of figure 4.4). A particular combination is chosen based on how frequently it occurs. It is then trained using the support vector machines. Unlike pose estimation algorithms, poselets are immune to occlusion and clutter. Since they are not constrained by top-down geometric constraints, they do not assume the presence of all body parts. Poselets corresponding to a missing body part will simply not fire. In all, 150 poselets which cover different parts of the body are trained [5]. Given an image, each poselet locates the corresponding body part by giving multiple bounding boxes (termed poselets as well). In our implementation, we set the maximum number of poselet clusters to 10. The poselets detected in an image are validated by mutual co-occurrence and all inconsistent poselets are discarded [4]. Using these poselets, the torso and the bounding box of the person are estimated.

Although poselets coarsely locate body parts in the form of bounding boxes, they do not give the exact human joint locations. We solve this problem by annotating the relevant body joint locations in each of the 150 poselets. For example in a poselet modelling the face and shoulder, forehead, chin, neck and the shoulders are manually marked in a normalized space. Given an instance of this poselet in an image, these annotated points are appropriately scaled and translated to get the body joint locations modelled by the poselet. In this paper, we consider 17 points *viz*., background, Head, Neck, and two each of torso, shoulders, elbows, wrists, hips, knee and Ankle. Our algorithm takes poselets, torso, bounding box of the person and the body joint locations as input.

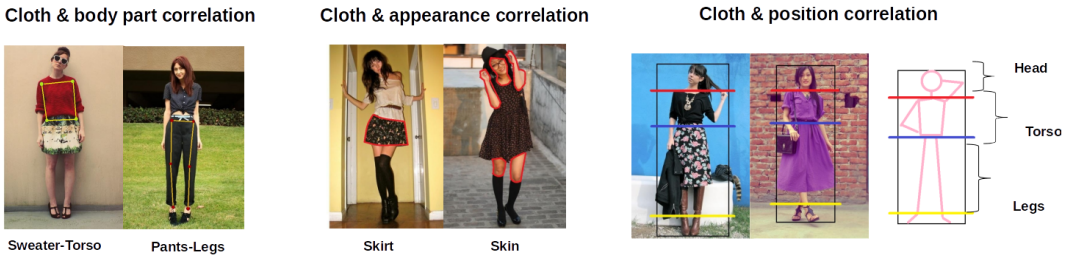


Figure 4.3 Different correlations leading to designing of the pipeline.

4.4 Parsing Clothes in Unrestricted Images

We model cloth parsing as a segmentation problem and assign each image region a label from the set L. Super-pixels are used as the basic image regions on the assumption that pixels which are adjacent and have similar appearance share the same label. Furthermore it can be observed that neighboring image regions have correlation in labels and it is certainly true in case of clothes. Thus Markovian assumption in space is valid and we use conditional random fields (CRF), represented by the undirected graph $G = (V, E)$, to model the cloth parsing problem.

Given an image, first the superpixels and body joint locations are computed. These superpixels form the vertices V of the CRF. Two superpixels which share a border are considered adjacent and are connected by an edge $e \in E$. The segmentation is obtained by taking the configuration with the maximum a posteriori probability, popularly called as MAP configuration. The best labeling using the CRF model is given by the equation,

$$\hat{L} = argmax_L P(L|Z, I), \tag{4.1}$$

where L is the label set, Z is a distribution of the body joint locations and I is the image.

The MAP configuration of CRF probability function given by the equation 4.1 is computationally expensive to compute and is usually a NP-hard problem. We thus make a simplifying assumption that at most two vertices in the graph form a clique thus limiting the order of a potential to two. Thus the CRF factorizes into unary and pair-wise functions and the log probability function is given by,

$$\ln P(L|Z, I) \equiv \sum_{i \in V} \Phi(l_i|Z, I) + \lambda_1 \sum_{(i,j) \in E} \Psi_1(l_i, l_j) + \lambda_2 \sum_{(i,j) \in E} \Psi_2(l_i, l_j|Z, I) - \ln G, \tag{4.2}$$

where V is the set of nodes in the graph, E is the set of neighboring pairs of superpixels, and G is the partition function.

4.4.1 Unary Potential

In CRFs, it is crucial to model the unary potential well for better performance. The unary potential function Φ models the likelihood of a superpixel s_i taking the label l_i. First, using the estimated pose

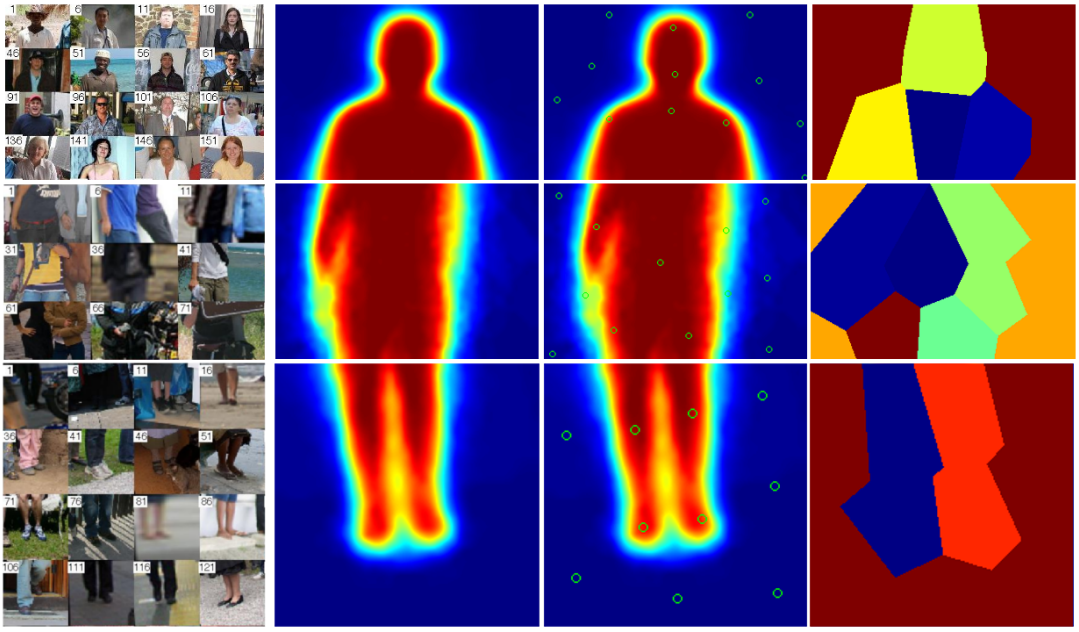


Figure 4.4 First, all the image regions in the training set which have a particular body configuration in common are selected (column one). Poselet [5] classifier is trained using these image regions to obtain a model and a mask (column two). In these poselet masks (column two), blue region represents the background and other color codes represent a human. We annotate the masks with body joints and background points (column three). Each of these masks exert an influence on an area around them (column four).

$Z = (z_1, ..., z_P)$ and the superpixel s_i, a feature vector $\phi(s_i, Z)$ is computed. Using the pre-trained classifier $\Phi(l_i|\phi(s_i, Z)) = \Phi(l_i|Z, I)$ for label l_i, a score is computed.

For the construction of features, the human body joint information, torso, bounding box and poselets are obtained using the procedure described above. The feature vector $\phi(s_i, Z)$ for each superpixel consists of following histograms: (1) RGB color histogram, (2) CIE L*a*b color histogram, (2) histogram of Gabor filter responses, (4) histograms of normalized X and Y coordinates, and (5) histograms of normalized X and Y coordinates relative to each body joint location z_p. The X and Y coordinates are normalized by the width and height respectively of the bounding box containing the person.

The construction of the first four feature types is straightforward. The fifth feature type is constructed using the human body joint information Z as follows. For all the super-pixels which do not intersect with any of the poselet bounding boxes, the relative locations with respect to all the body joints are assigned infinity. For the super-pixel which intersects with a poselet bounding box, relative locations with respect to only the body joints present within the poselet bounding boxes are taken. For all body joints not present in the intersecting poselets, the relative location is assigned infinity. In case a super-pixel intersects with multiple bounding boxes, the relative position of each pixel with respect to a common part z_p is weighed by the poselet score. Intuitively, this procedure can be understood as noting the relative

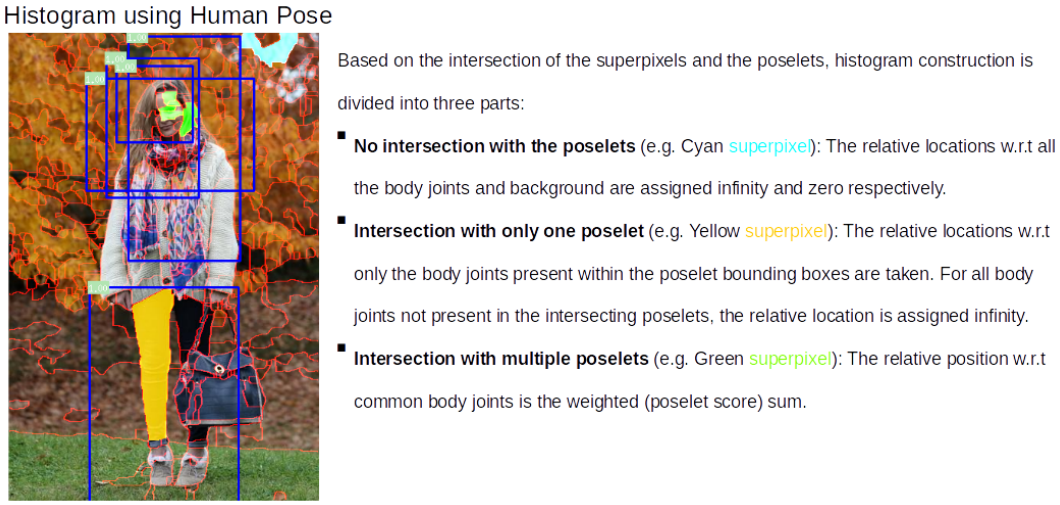


Figure 4.5 Achieving pose histogram using poselet information.

position from a mean body joint location averaged using the poselet scores. Once the relative locations for all the pixels in a super-pixel are noted, a histogram is built.

For the feature ϕ, one of the crucial source of information is the human joint locations. The human pose consists of 14 joints viz., Head, Neck, and two each of shoulders, elbows, wrists, hips, knee and Ankle. Location of each cloth worn by a person is strongly correlated with one or more of the above body joints. Given an image, all the poselets are run on the image. A poselet then decides if a particular image region has the body part that it models. All those regions which are above the pre-determined threshold are said to have belonged to the poselet. The final output is a set of bounding boxes with an associated poselet ID and a score. Unfortunately, the raw poselet output does not contain any body joint locations or a pose. However as mentioned, a probability distribution of the body joints over a pixel locations is desired. To achieve this, we manually annotated all the 150 poselet masks with appropriate body joints. Figure 4.4 illustrates the procedure.

The feature vector from the poselet output is constructed as follows. First, the relative positions of all the pixels with respect to background and body joints are initialized to zero and infinity respectively. Within each poselet bounding box, the relative position of each pixel is noted with respect to the annotated body joints. For all other body joints the relative position would be infinite. Similarly for all the pixels which are outside all the poselet bounding boxes, the relative distance would be infinite. For all those pixels which lie in multiple bounding boxes, the relative positions are weighed by the poselet scores. Intuitively, this procedure can be understood as noting the relative position from a mean body joint location averaged using the poselet scores. Based the powerful evidence from the earlier work [64] and also from our experiments, we use color, texture and absolute position of pixels as additional features. Using these features a model is learnt for each label.

4.4.2 Pair-wise potential

For the pairwise potential, we use the definitions from [64]. Pairwise potential, defined between two neighboring super-pixels, models the interaction between them. The pair-wise potential is defined in equation 4.2 as sum of two functions (called factors) $\Psi(l_i, l_j)$ and $\Psi(l_i, l_j|Z, I)$. The pairwise potential function Ψ_1 models the likelihood of two labels l_i, l_j being adjacent to each other and Ψ_2 models the likelihood of two neighboring sites s_i, s_j taking the same label given by the features $\phi(s_i, Z)$ and $\phi(s_j, Z)$ respectively. The function Ψ_1 is simply a log empirical distribution and Ψ_2 is a model learnt over all the label pairs respectively. The pairwise potential functions are given by,

$$\Psi_1(l_i, l_j), \quad \Psi_2(l_i, l_j|Z, I) \equiv \Psi_2(l_i, l_j|\psi(s_i, s_j, Z)) \tag{4.3}$$

where $\psi(s_i, s_j, Z)$ is defined as,

$$\psi(s_i, s_j, Z) \equiv [(\phi(s_i, Z) + \phi(s_j, Z))/2, |(\phi(s_i, Z) - \phi(s_j, Z))/2|]\,. \tag{4.4}$$

4.4.3 Training and inference

Given a dataset containing images and cloth labels which include background, we wish to learn the cloth parser model. First for each data sample, superpixels and poselets are computed. All the superpixels which share a border are noted as neighbours. For each superpixel which falls within a poselet, the relative distance from the body joints present in the poselets are noted. At each superpixel the feature vector $\phi(s_i, Z)$ consisting of , (1) RGB color histogram, (2) CIE L*a*b color histogram, (2) 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 z_p are noted. The bin size for the histograms is 10. Logistic regression is used to learn $\Phi(l_i|Z, I)$ and $\Psi_2(l_i = l_j|\psi(s_i, s_j, Z))$. Given a new image, the super-pixels, poselets and feature vector ϕ are computed. For each super-pixel, the unary potential and pairwise potential values are computed using the feature vector and the learnt models. The best label is inferred using the belief propagation implemented in libDAI [35] package. The parameters λ_1, λ_2 in the equation 4.2 are found by cross validation.

4.4.4 Clothing pattern Mining

Using the cloth labelling obtained from the algorithm described in the previous section, several interesting clothing patterns can be mined from a collection of images. In this section two such interesting patterns *viz.*, cloth co-occurrences and upper and lower body cloth's color co-occurrences are explored on the Fashionista dataset. Some clothes look pleasant to human perception based on factors such as type of cloth, color of cloth, fitting of cloth etc. People follow the general trend of clothing which is set by the fashion designers and the movie actors/actresses of the current time. On a given dataset of images, we try to determine what kind of trend is followed in it i.e which color has been mostly worn or what type of clothes have been worn by the people.

The objective is to ascertain how many times does $(item_1, ..., item_n)$ item set co-occur. The classic apriori algorithm [1] is an efficient way to determine the co-occurrence frequency. A threshold $mt \in [0, 100]$, called minimum support, has to be specified by the user to prune low frequency item set co-occurrences. The algorithm uses the apriori property which states that any subset of an item set should have a minimum support. Initially frequent item sets of size one are generated that are above the minimum support. In the next step, frequent item sets of size two are generated by taking all the combinations from item sets of size one and then pruned based on minimum support. For the item sets of size greater than two, only those survive whose subsets have a minimum support. Finally the algorithm outputs the item set co-occurrences and their support values.

Cloth co-occurrences are obtained by applying the apriori algorithm on the cloth labels determined by the proposed method. To get interesting co-occurrences, frequent occurring labels like skin, hair *etc*, have been removed. We try to give some information on what types of cloth co-occur in the images. These cloth types are different labels in an image. Hence, cloth co-occurrence is equivalent to label co-occurrence in the dataset.

Color co-occurrences of upper-body and lower-body clothes are mined using the following procedure. First, a representative cloth type for the upper-body and lower-body are determined. This is done by selecting the outermost cloth worn by the person. For example, blazer is worn over t-shirt and hence it represents the upper-body cloth. To decide on what label is to be considered over the other, a precedence list is generated. The order of precedence for upper-body clothes is coat, blazer, jacket, cape, cardigan, sweater, sweatshirt, jumper, shirt, t-shirt, dress, romper, top, vest, blouse and bra. Similarly the order of precedence for lower-body clothes is jeans, pants, skirt, shorts, leggings, tights, stockings and socks. Images that do not have either upper-body cloth or lower-body cloth labels specified above are ignored. The upper-body and lower-body clothes are then assigned a color as follows. The dominant RGB value in the upper-body or lower-body cloth image region is taken as its representation. This RGB value is then vector quantized to one of the 500 cluster centers. Using the map between colors and cluster centers, the image region of upper-body or lower-body is assigned one of blue, brown, orange, pink, red, violet, green and yellow. The apriori algorithm is then applied to obtain the frequent upper-body and lower-body color co-occurrences.

Method	Full-a		Full-m		Unary	
	Pixel acc	mAGR	Pixel acc	mAGR	Pixel acc	mAGR
[64]	61.0 ± 5.0	34.9 ± 3.9	49.9 ± 4.5	39.9 ± 4.8	49.5 ± 4.3	39.6 ± 4.4
Ours	74.5 ± 4.7	49.7 ± 3.8	68.5 ± 5.4	55.2 ± 4.5	68.4 ± 5.4	54.8 ± 4.3
Ours+Noc	77.4 ± 4.0	57.0 ± 3.3	70.2 ± 5.2	63.1 ± 4.5	70.1 ± 5.2	62.4 ± 4.3

Table 4.1 Results on H3D: The baseline for accuracy is 74.7 ± 5.6 and for mAGR is 14.3 ± 0.6. 'Ours+Noc' indicates that the algorithm has been run with out the 'occluding object' label.

Garment	[64]	Ours	Garment	[64]	Ours
background	54.8 ± 4.6	74.0 ± 5.7	shoes	33.3 ± 11.7	51.0 ± 15.2
upperclothes	25.6 ± 6.0	65.6 ± 7.6	bag	12.8 ± 10.9	19.8 ± 12.0
lowerclothes	40.4 ± 9.4	59.9 ± 14.9	occ-object	13.5 ± 8.7	13.5 ± 5.8
skin	71.5 ± 10.5	62.0 ± 8.9	hat	19.2 ± 26.9	40.2 ± 21.6
hair	43.3 ± 11.0	62.2 ± 11.6	socks	0.0 ± 0.0	0.8 ± 2.7
dress	0.6 ± 1.3	0.1 ± 0.2	sunglasses	1.3 ± 4.1	13.6 ± 24.8

Table 4.2 Recall for selected garments on H3D.

4.5 Experimental Results

We evaluate our method on two datasets, a) Fashionista [64] and b) H3d [5]. For each image in the above datasets, we compute the superpixels and poselets, both of which have standard implementations available on the internet. We then compute the cloth labelling using the method described in section 4.4. Using the ground truth segmentation masks, we evaluate our algorithm and also compare it with Yamaguchi *et al.* [64]. Two measures, pixel accuracy and mean average garment recall (mAGR) are used to quantitatively evaluate the algorithms. Pixel accuracy is total number of correct pixel labelling in the image. It is a gross measure and is biased towards labels with large areas (e.g., background). The measure, mAGR, is much more balanced measure and gives equal importance to labels of all the sizes. Two sets of parameters λ_1, λ_2 each are optimized for pixel accuracy and mAGR using cross validation. The outputs from these two parameters are termed Full-a and Full-m respectively. As a baseline, all the pixels are labelled as background and the above two measures are calculated.

4.5.1 Datasets Used

H3D: This dataset has been introduced in [5]. It has a total of 180 images for training, 40 for validation and 107 images for testing. This dataset is derived from flickr images and is very complex with severe occlusions and heavy clutter. The dataset has a total of 22 labels which include, face, hair, upperbody clothes, lowerbody clothes, hair, face, neck, left/right arm, left/right leg, left/right shoe,

Method	Full-a		Full-m		Unary	
	Pixel acc	mAGR	Pixel acc	mAGR	Pixel acc	mAGR
[64]	89.0 ± 0.8	63.4 ± 1.5	88.3 ± 0.8	69.6 ± 1.7	88.2 ± 0.8	69.8 ± 1.8
Ours	87.7 ± 0.8	62.7 ± 1.8	86.0 ± 1.0	70.2 ± 2.0	85.9 ± 1.1	70.4 ± 2.0

Table 4.3 Results on Fashionista: The baseline for accuracy is 77.6 ± 0.6 and for mAGR it is 12.8 ± 0.2.

Garment	[64]	Ours	Garment	[64]	Ours
background	95.3 ± 0.4	92.5 ± 0.9	jacket	51.8 ± 15.2	49.4 ± 15.5
skin	74.6 ± 2.7	74.7 ± 2.1	coat	30.8 ± 10.4	29.5 ± 12.5
hair	76.5 ± 4.0	78.2 ± 4.0	shirt	60.3 ± 18.7	65.5 ± 21.9
dress	65.8 ± 7.7	59.2 ± 10.4	cardigan	39.4 ± 9.5	46.8 ± 11.6
bag	44.9 ± 0.8	44.0 ± 7.6	blazer	51.8 ± 11.2	54.7 ± 9.5
blouse	63.6 ± 9.5	63.1 ± 10.3	t-shirt	63.7 ± 14.0	68.2 ± 10.7
shoes	82.6 ± 7.2	80.3 ± 9.5	socks	67.4 ± 16.1	58.3 ± 23.2
top	62.0 ± 14.7	64.4 ± 12.8	necklace	51.3 ± 22.5	65.6 ± 12.4
skirt	59.4 ± 10.4	55.7 ± 14.0	bracelet	49.5 ± 19.8	50.2 ± 22.8

Table 4.4 Recall for selected garments on Fashionista.

occluding object, bag, hat, dress, left/right glove, left/right sock and sunglasses. Since the main concern is cloth segmentation, the left/right part of the same cloth type is not relevant. All the left/right labels are thus converted into a single label (*e.g.*, left/right shoe become shoes). Furthermore, labelling different body parts like left/right leg, left/right arm, neck and face are not relevant and have been converted to a single label skin in the image. Finally in case of occlusion from a person, the labels of the occluding person is considered and occlusion from an object is labelled as occluding object. In all a total of 13 labels are present *viz.*, upperbody clothes, lowerbody clothes, occluding object, skin, hair, dress, shoes, bag, hat, socks, background and sunglasses.

Fashionista: This dataset has been introduced in [64]. It has a total of 685 images collected from Chictopia.com website made for fashion bloggers. Since these are fashion images, it includes a wide range of garment types and accessories. Each image has one person standing in an upright position with a clean background and is annotated with labels at a pixel level. In all there are about 56 labels. Broadly the labels can be classified as upper-body clothes, lower-body clothes, accessories, skin and hair.

4.5.2 Qualitative and Quantitative Results

H3d: In this dataset, both the algorithms assign a label to each pixel. Since Yamaguchi *et al.* [64] assumes that there is a single person per image, we adapt it to multi-person images. First the pose estimation algorithm [66] available at the author's site is used to predict multiple pose estimates. Then features corresponding to the absolute position and pose-based ones are calculated relative to the bounding box defined by the pose estimate. In our opinion, we have made all the efforts to adapt the

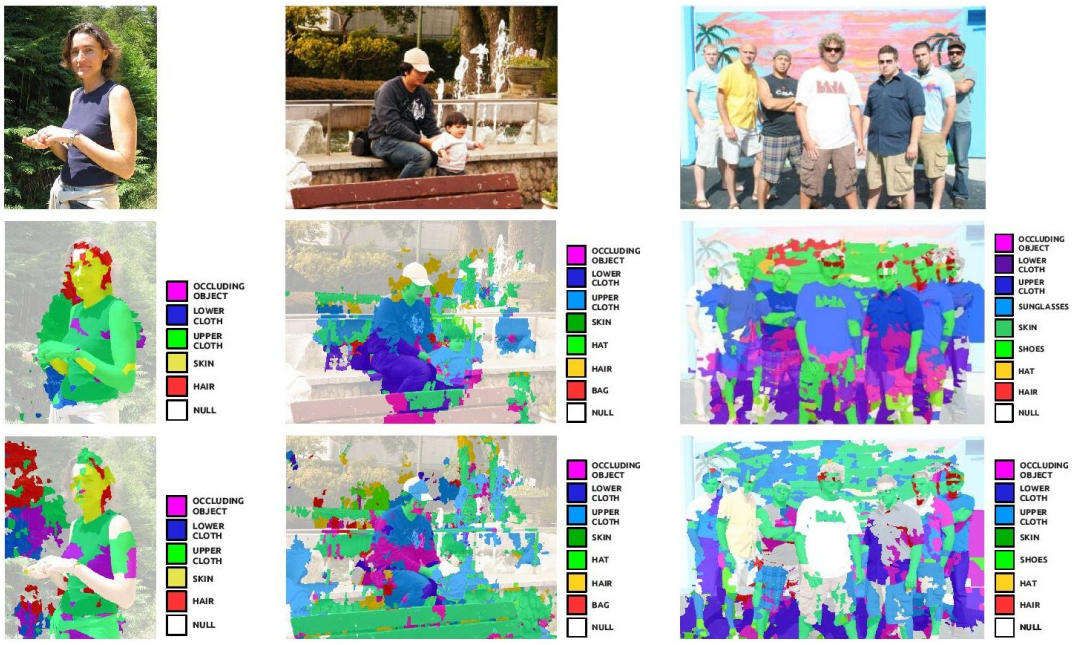


Figure 4.6 H3D results comparison: For the input images from the H3D dataset (row 1), results from our method (row 2) and Yamaguchi *et al.* [64] (row 3) are displayed. Clearly our method has superior segmentation than [64].

algorithm to make a fair comparison. Table 4.1 clearly indicates that our method outperforms [64] by about 13.5% and 15.3% in pixel accuracy and mAGR measures respectively. A similar trend can be seen in recall of individual labels in table 4.2. We also observed that when the 'occluding object' label is removed from the dataset (substituted with 'background'), both accuracy and mAGR values increase significantly (table 4.2) indicating that the algorithm had modest success in modelling the 'occluding object'. Figure 4.7 qualitatively compares our method with [64].

Fashionista: Yamaguchi *et al.* [64] have defined a protocol while evaluating the algorithm. The dataset has 56 labels and typically only a few of them are present in any given image. For each image, the identity of these labels are made available to the algorithm, without of course divulging the location of these labels. The algorithm is then expected to predict the location for each label in the image. As seen in the table 4.3, our method is marginally higher in mAGR and marginally lower in accuracy than the Yamaguchi *et al.* [64]. Table 4.4 shows the recall for several cloth types. Clearly our method is on par with [64] in most of the labels. Figure 4.6 qualitatively compares our method with [64].

Co-occurrences: Using the labelling obtained on Fashionista dataset reported above, cloth and color co-occurrences are computed as described in section as described above. The minimum support threshold is set to 4%. The top 5 cloth co-occurrences for Fashionista dataset are skirt-top (6.3), shorts-top (5.7), blouse-skirt (4.5), tights-dress (4.4) and cardigan-dress (4.1). Similarly the top 5 cloth co-occurrences for Fashionista dataset are upper-blue:lower-blue (17.0), upper-red:lower-red(13.4), upper-

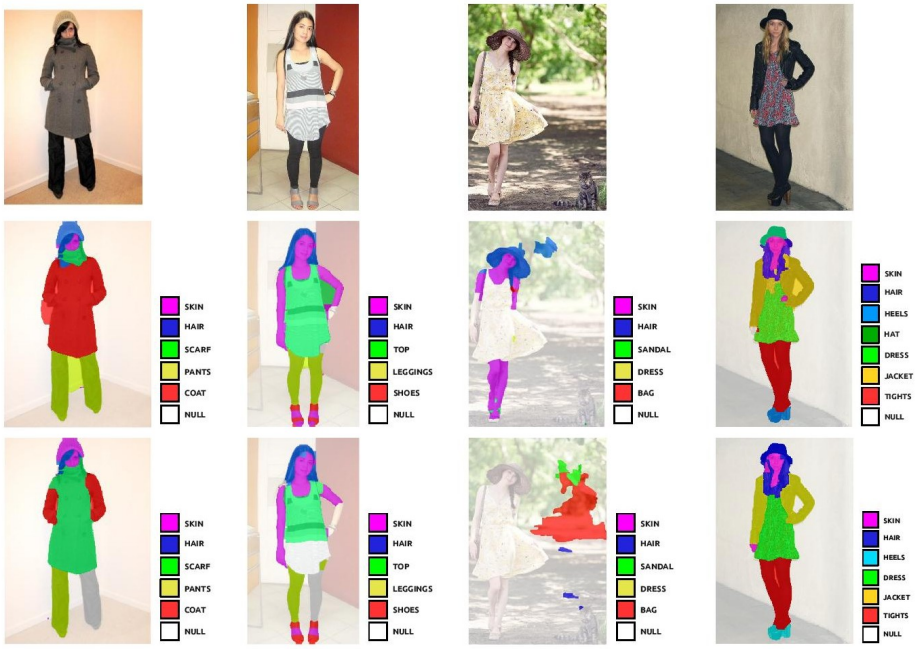


Figure 4.7 For each input (row 1), output from our method (row 2) and Yamaguchi *et al.* [64] (row 3) is displayed.

red:lower-blue (12.0), upper-blue:lower-red (7.0) and upper-white:lower-blue (6.1). Figure 4.8 displays results of both cloth and color co-occurrences.

4.6 Further Experiments

From evidence provided in Table 4.5, we reach to a conclusion that cloth parsing is a lot better when we have domain pre-knowledge for each image. Availability of domain refers to the accessible knowledge of cloth labels present in each image. Without domain knowledge, the algorithm assumes the presence of all possible cloth labels and makes prediction based on the learning. However, methods [64], [26] are trained with supervision and since the labeling is more locality based, this deprives the algorithm from generating more holistic estimations. This results to the method generating a lot of false positive class labellings. As can be seen from Table 4.5, there is a substantial margin in performance when domain knowledge is available (row 1) and when domain knowledge is unavailable (row 2).

In our case, the problem of gaining domain knowledge is equivalent to cloth type classification in an image. Cloth type classification can be modeled with dependence or independence of cloth parsing techniques. Below we demonstrate some experiments conducted in which we attempt improvement for both cloth parsing and cloth type detection problem.

Figure 4.8 (a) Cloth co-occurrence: The first four columns correspond to Shorts-Top, Dress-Cardigan, Skirt-Top and Dress-Tights co-occurrences respectively. (b) Color co-occurrences of "upperbody cloth-lowerbody cloth" combination: The next four columns correspond to white-blue, blue-red, blue-blue and red-blue co-occurrences respectively.

Method	Yamaguchi [64]		Ours [26]	
	Pixel acc	mAGR	Pixel acc	mAGR
With Domain	87.6	72.0	85.9	70.4
Without Domain	76.1	38.0	74.6	35.4

Table 4.5 Unary Results on Fashionista: Comparing the performance with and without domain knowledge per image.

4.6.1 Lexicon Reduction Approach

Lexicon reduction techniques are used when the lexicon size is relatively larger as compared to the actual domain size of a testing image. Because of similarity in system restrictions, we follow the work of Roy *et al.*[46]. This work uses an iterative lexicon reduction technique having both unary and pairwise updations in each iteration. More focus is on getting diverse solutions to maintain a higher recall value. To obtain M diverse solutions the approach involves two steps:

1. Unary updation: CRF inference runs M times for M iterations where we update the unary score in each iteration as follows:
$$U(t+1) = U(t) + \lambda.E(t) \tag{4.5}$$
where $U(t)$ is the unary potential and $E(t)$ is full inference score at iteration t.

2. Pairwise updation: Remove labels from the lexicon which have the highest edit distance from all the diverse solutions obtained. Recompute pairwise potentials based on the new vocabulary obtained. For our case, we generate a new pairwise prior matrix at each iteration so that it corresponds to co-occurrence relations among labels that are currently present in the vocabulary.

For experiments we pick Fashionista [64] dataset. In this dataset, average number of labels per image lie in between 6 to 8 and we keep the domain size finally achieved using lexicon reduction to be equal to 10 ($\text{M} = 10$). Essentially, the domain size for each test image in Fashionista is reduced from 26 (derived set from initial 56 set) to 10. Table 4.6 shows the performance of lexicon reduction technique (last column) in comparison to previously mentioned algorithms for cloth parsing.

Full4a	Yamaguchi [64]	Jammalamadaka [26]	Lexicon reduction
acc	80.9	80.1	79.9
mAR	39.9	38.5	31.0

Table 4.6 Results for lexicon reduction method on Fashionista[64] dataset.

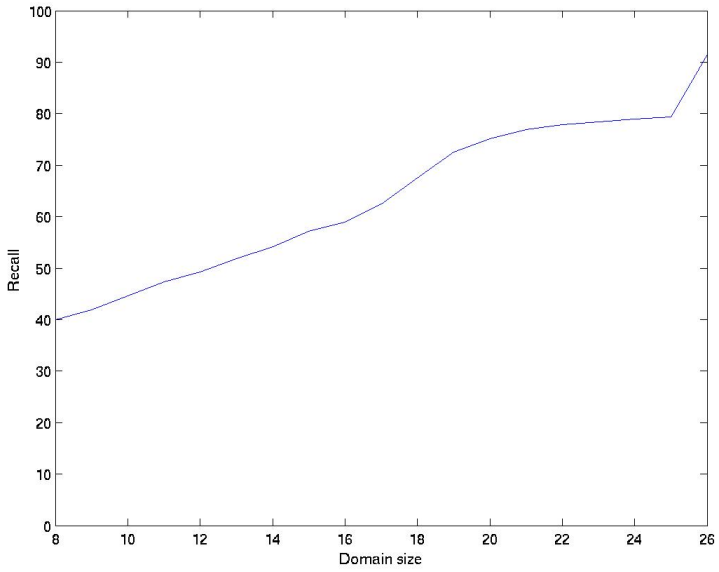


Figure 4.9 Plot showing recall of garment classes as the domain size varies.

Figure 4.9 shows the significant drop in garment recall as the domain size reduces when we apply lexicon reduction technique. The plot implies that we are missing most of the ground truth class labels as our domain size reduces to a lower number (like 10).

Analysis. To understand the efficacy of lexicon reduction technique used, we compare our implementation against its variations and trivially synthesized estimations. Below we elaborate on the strategies compared:

- Method A: **Lexicon Reduction Method**. The above mentioned strategy is used with unary and pairwise updations. To reduce the domain set size from 26 to 10, a label is removed in each iteration which has the least co-occurrence score with all the predicted labels.

- Method B: **Lexicon Reduction + Updated Prior Information**. The label removal strategy is improved such that in each iteration only those labels are considered for removal which are not in the predicted set for current image. This improves the updation of pairwise prior matrix such that it also considers the frequently occurring labels in the dataset.

- Method C: **Using Pre-Defined Cloth Configurations**. Four cloth configuration defining sets are constructed:
  i. InnerUpperCloth : T-shirt, Shirt, Top
  ii. OuterUpperCloth : Coat, Cardigan, Jacket
  iii. LowerCloth : Skirt, Pants, Shorts, Tights
  iv. UpperLowerCloth : Dress
  In each image, we can make a safe assumption that only certain configurations can occur with only one label from each of the set mentioned above. Further, the label removal strategy is modified such that in each iteration we remove one label from any set containing more than one labels. We run iterations until each set has only one label left.

- Method D: **Artificial Label Selection criteria**. To understand the stature of above techniques, we artificially synthesize a set of labels. At each iteration, we simply remove the label which is the least occurring label in the dataset among the labels left in the current domain set. Hence, all the images in the dataset will have the same domain set.

Plot shown in Figure 4.10 shows method A has a very poor performance, whereas method sc b and C are better and comparable. However, method D is giving the best performance despite being a very trivial strategy. We make following observations from the obtained results:

1. The lexicon reduction strategies explored have failed to produce approximately satisfactory results.

2. The current recall and precision values for labels lie around 60%. On an average, an image has 8 labels and 4 labels (skin, background, hair, shows) are omnipresent. Out of the rest 4 labels, we can correctly predict 0.8 label per image.

3. Unary potentials are weak and are unable to classify correctly among present possibilities.

4. The cloth parsing problem is modeled as CRF with superpixels as nodes is a highly unconstrained problem. It does not explicitly model that clothes can occupy many superpixels in a really big region. Superpixel labeling with a very restrained neighborhood information prevents a holistic interpretation.

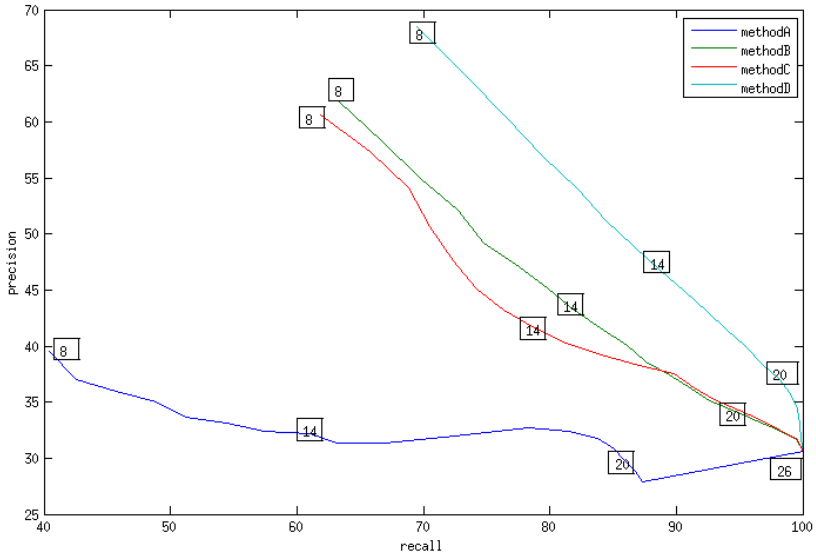


Figure 4.10 PR curve for the 4 lexicon reduction methods compared showing the trivial method D outperforms other designed methods.

4.6.2 Cloth Type Detection using Symmetrical Similarity

Main intuition forming the basis of this experiment is that human clothing is symmetrical along a vertically bi-sectional line. For example, shirt will give similar feature distribution when picked from symmetrical body parts like left-arm and right-arm, trousers will produce similar features for left-leg and right-leg. Using such automatically trained similarity criteria, we can determine clothing type in an image.

To test the viability of the experiment, we assume the knowledge of correct human pose estimations in all testing images. First task for feature generation is to segment image to generate body parts given the available pose skeleton. Human body is divided into 14 body parts, these are: *head, torso, l-uparm, l-lowarm, r-uparm, r-lowarm, l-thigh1, l-thigh2, r-thigh1, r-thigh2, l-leg1, l-leg2, r-leg1 and r-leg2* (l- : left and r- : right). These body parts are shown in red bounding boxes in Figure 4.11 (left). Superpixels covering most of the bounding box region for a body-part become final body-part region in the image. 33-Dimensional RGB features are extracted for N body-parts from the pixels of corresponding part region in the image. For each body-part, its Bhattacharyya distance is computed with the rest N-1 parts. Bhattacharyya distance D_B for two distributions p and q of size s is computed as follows:

$$D_B = \sum_{i=1}^{s} \sqrt{p_i q_i} \tag{4.6}$$

Figure 4.11 (right table) shows top 3 similar parts to each body-part and their respective score for the image shown in left. The classifier should be able to learn a characteristic pattern of similarity for a particular body-part.

Bodypart	Similar part1	Similar part2	Similar part3
r-uparm	l-lowarm:0.98	l-uparm:0.82	head:0.75
r-lowarm	head:0.73	r-thigh1:0.68	r-leg1:0.68
l-uparm	l-lowarm:0.82	r-uparm:0.82	head:0.80
l-lowarm	r-uparm:0.98	l-uparm:0.82	head:0.80
r-thigh1	l-thigh1:0.80	r-lowarm:0.68	r-leg1:0.56
r-thigh2	l-leg2:0.95	r-leg2:0.92	r-leg1:0.91
r-leg1	r-thigh2:0.91	r-leg2:0.91	l-leg1:0.90
r-leg2	l-leg2:0.97	r-thigh2:0.92	r-leg1:0.91
l-thigh1	r-thigh1:0.80	torso:0.74	r-lowarm:0.64
l-thigh2	r-thigh2:0.90	l-leg1:0.88	r-leg1:0.87
l-leg1	l-leg2:0.94	r-thigh2:0.90	r-leg2:0.90
l-leg2	r-leg2:0.97	r-thigh2:0.95	l-leg1:0.94

Figure 4.11 Left: Image describing 14 body parts defined in the experiment. Right: Top 3 similar body-parts when compared with each body-part along with their similarity score.

Results and Observations. We use linear SVM for classification task given the extracted feature set. The average precision (AP) value for some labels over Fashionista [64] dataset is shown in Table 4.7.

Full4a	InnerUpper	Dress	OuterUpper	Bag	Hat
AP (%)	64	32	61	44	16

Table 4.7 AP score of labels for cloth type classification on Fashionista[64] dataset using similarity features.

The values look unsatisfactory even with the availability of correct pose estimations. We pick a particularly poor-performing class *Dress* for analysis and making observations.

- As shown in Figure 4.12, one label can be worn in so many different ways and in practice, the generally assumed trend is not followed many times.
- On looking at SVM weights learned for Dress label, these co-relations were found to be highest weighed: (r-uparm, r-lowarm), (torso, r-thigh1), (l-lowarm, l-leg1), (l-uparm, l-thigh2). Many of the relations are irrelevant as their presence is unnecessary and is valid for other labels as well.
- Similarity features are incapable of handling the real-life possibilities and variations of clothing configuration people wear.

4.6.3 Bottom-Up Approach for Cloth Parsing

The problem of Cloth Parsing is broken down to three fundamental steps which semantically follows a bottom-up criteria of gathering information. We first try to get best human segmentation as this is the region of interest for us. In second step, we segment human region into smaller regions corresponding to

Figure 4.12 Images showing Dress worn in different ways making any kind of configuration assumption invalid.

a body-part. Finally, the part regions are labeled as one of the possible cloth classes. All the experiments are conducted on Fashionista dataset.

1. **Human Segmentation** (HS) : Segment human body from background.
   Following are the various methods we have used to extract human region as foreground from rest of the image.

   - Grabcut [45] (Automated): Generally Grabcut technique assumes manual intervention to obtain small foreground and background regions and utilize energy minimization for segmenting image into foreground and background regions. We have a prior knowledge of the presence of human in the image which is exploited by using human pose estimation algorithms. From the pose skeleton, an expanded rough bounding box is made and the image region outside this box is considered strong background. Given this background mask, Grabcut implementation provides us a foreground mask.
   - Cloth Parsing: Available cloth parsing approaches gives us a possible labeling of the image region with cloth segments. All the non-background labeled pixels are considered as foreground and the map obtained is given as a human segmentation candidate.
   - Simultaneous Detection and Segmentation (SDS) [23]: A more recent approach using DeepNet architecture first generates a huge number of region proposals and then uses classifiers to score and label the regions.

   We use simple evaluation metrics like Intersection-over-Union (IOU), False Positives (FP) and False Negatives (FN). Another metric being used is Body-Parts Missing Rate (partsFN) in which we segment human into body parts and evaluate the missing parts. Table 4.8 shows Grabcut and Cloth Parsing have similar performance on the three metrics whereas background spilling is very low for SDS. However, the foreground miss value is higher for SDS when compared with other methods. Same is the case when we evaluate body-part missing rate partsFN in Table 4.9. We use a combination of SDS and Grabcut, that produces a relatively better performance (Table 4.8

last row) for the next step in the pipeline. Overall, the performance looks grim as there is a clear trade-off between background spilling and missing foreground segments. The body-parts which are most difficult to segment as foreground are limbs: arms and legs which are of high relevance for the task of cloth parsing.

	IOU	FP (bg spilling)	FN (misses)
Grabcut[45]	77.81	14.57	9.68
Cloth Parsing	77.18	15.18	10.51
SDS[23]	73.03	9.14	20.60
SDS+Grabcut	80.09	14.56	6.89

Table 4.8 Table showing the performance of mentioned human segmentation techniques using various evaluation metrics.

	head	torso	r-lowarm	l-lowarm	r-leg	l-leg
Grabcut[45]	10.28	4.86	16.10	16.64	14.73	15.71
Cloth Parsing	4.78	3.64	13.79	14.77	16.58	15.63
SDS[23]	19.38	6.19	33.07	32.13	40.78	39.52

Table 4.9 Table showing the performance of mentioned human segmentation techniques using partsFN evaluation. Higher the value, higher is the percentage of missing body-part in the foreground estimation.

2. **Human Part Segmentation** (HPS) : Divide human region into body-parts.
   Foreground segmentation achieved from previous HS step has to segmented further into smaller regions corresponding to human body parts. Given only the foreground map, we use graph based image segmentation method by Felzenszwalb *et al.* [17]. Several parameters in the implementation have been tuned by experimentation. To evaluate segments obtained, we try to match the boundaries of estimated and ground-truth regions. We use Berkeley Contour Matching (BCM) [32] criteria for evaluation purpose. BCM is f1 score computed from precision and recall of estimated boundaries with respect to ground-truth boundaries. By testing on Fashionista dataset, we get 78.34% precision, 87.08% recall and 81.92% BCM score.

3. **Human Part Classification** (HPC) : Assign label to each segmented part.
   Part segments obtained from the previous step are labeled using following methods to achieve reliable predictions:

   - Fisher Vectors (FV): FV encodes feature vectors into a a high-dimensional representation. Input feature vectors are fitted with a parametric generative model like GMM. Final representation is created by capturing average first and second order difference between features

and GMM centers:

$$\phi_k^{(1)} = \frac{1}{N\sqrt{w_k}} \sum_{p=1}^{N} \alpha_p(k) \left(\frac{x_p - \mu_k}{\sigma_k} \right), \quad \phi_k^{(2)} = \frac{1}{N\sqrt{2w_k}} \sum_{p=1}^{N} \alpha_p(k) \left(\frac{(x_p - \mu_k)^2}{\sigma_k^2} - 1 \right) \tag{4.7}$$

Here, $\alpha_p(k)$ is assignment weight of p-th feature to k-th Gaussian. w_k, μ_k, σ_k are parameters for k-th Gaussian.

We use three different types of input features which have proved to be helpful for fine-grained classification task [22]. These are SIFT, Xcolor and rootSIFT. After getting the encoded representation, we train linear SVM for each of the input feature type. Average Precision (AP) scores for some labels are shown in Table 4.10 which clearly shows rootSIFT derived encoding gives the best performance. However, the mAP score for each of them is quite low to be used for practical purposes.

	hair	blazer	t-shirt	skirt	shorts	**mAP**
SIFT	45.25	10.41	4.56	3.22	5.77	13.84
Xcolor	34.40	13.29	3.43	4.24	4.43	11.95
rootSIFT	52.07	13.42	6.07	3.28	6.02	16.17

Table 4.10 AP score of SVM classification when using different input feature types for FV encoding.

- 2CN-1FC DNN architecture: Part region pixels are labeled using DNN features. An image patch of size $60X60X3$ is obtained with central pixel as the pixel to be labeled. The architecture used for this experiment has 2 convolutional layers, 1 fully-connected layer and a final soft-max regression layer. Rectified Linear Unit function is used for non-linearity and cross entropy loss is utilized to obtain the loss value. Figure 4.13 shows confusion matrix on Fashionista test set which clearly indicates mis-classification for most of the intricate labels like blazer, shirt, pants.
- 4CN-2FC DNN architecture: To make improvements over 2CN-1FC architecture mentioned above, we increase the architecture size and incorporate additional features. The architecture has 4 convolutional layers, 2 fully-connected layers and a final soft-max regression layer. Features used per patch of size $60X60X33$ are:
  - 3 RGB maps
  - 2 absolute position (x and y) maps
  - 14X2 relative position maps to 14 body-joints

Results and Analysis. We compare and analyze the above 3 mentioned approaches for Human Part Classification. Table 4.11 compares the approaches on the most intricate labels using AP evaluation measure. 4CN-2FC DNN architecture with additional position features performs the best with a mean AP score of 28.23%. However, when we look at scores from relevant labels like

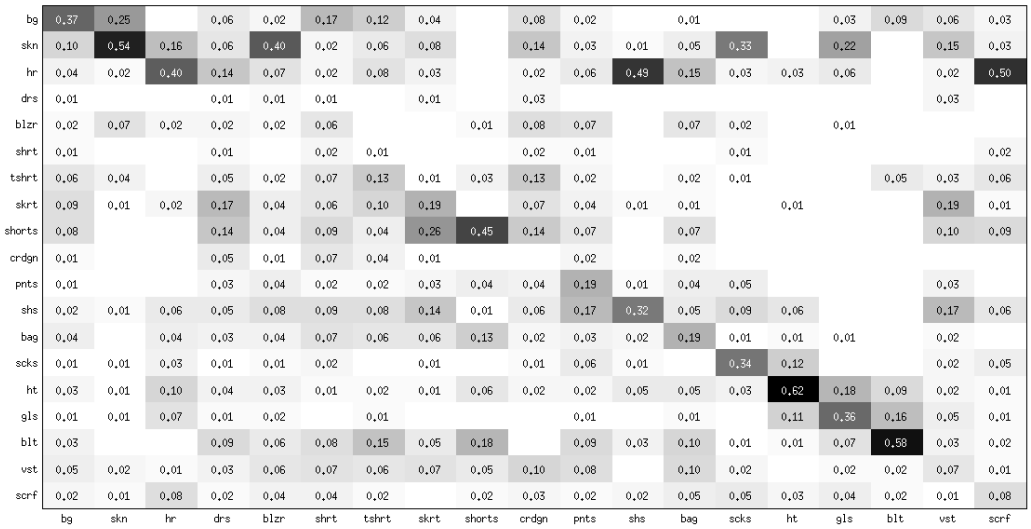


Figure 4.13 Confusion matrix for Fashionista dataset by using 2CN1FC architecture showing misclassification for most of the intricate cloth labels

t-shirt, skirt, shorts *etc.*, the methods are under-performing and can not be used for any practical purposes. More analysis shows that in 4CN-2FC DNN architecture the position features help a lot in focusing on relevant regions for each label but the unary score is still very low to be able to make a confident prediction. Because of the variation of color, texture each label can encompass, it does not seem to be learning much from these.

	hair	blazer	t-shirt	skirt	shorts	**mAP**
FV (rootSIFT)	52.07	13.42	6.07	3.28	6.02	16.17
2CN-1FC DNN	3.54	2.7	0.98	1.60	1.38	10.57
4CN-2FC DNN	69.78	37.70	18.40	10.53	9.02	28.23

Table 4.11 AP score of cloth label classification comparing the best FV performance and two DNN architectures explored.

4.6.4 Cloth type detection using DNN

In the previous experiments using DNN architecture, we faced numerous difficulties for properly training a network. Fashionista dataset has only 685 images whereas a network generally requires many thousands of input patches. So, instead of giving the full image to the network we sub-divided the image into smaller regions ($60X60$) which creates another complication of not being able to capture a holistic interpretation of different cloth labels. We use a new dataset called Apparel Classification with Style (ACS) [3] having 90K images of cropped clothing items.

Pre-trained ImageNet classification models have proved to be useful for estimations on different datasets. We use pre-trained VGG Network and fine-tune the network using training samples from ACS dataset by keeping the standard parameters. Figure 4.14 shows energy and error plot of VGG net on training and testing set of ACS dataset. Using the stopping criteria determined by training error and energy values, we achieve a mean Average Precision of approximately 54% on test set. However, these values are higher than obtained from previous experiments of cloth type classification, this experiment is restricted by the dataset type used. Adapting to a more real-life dataset with people wearing clothes create more intricacies which needs to be diligently taken care of.

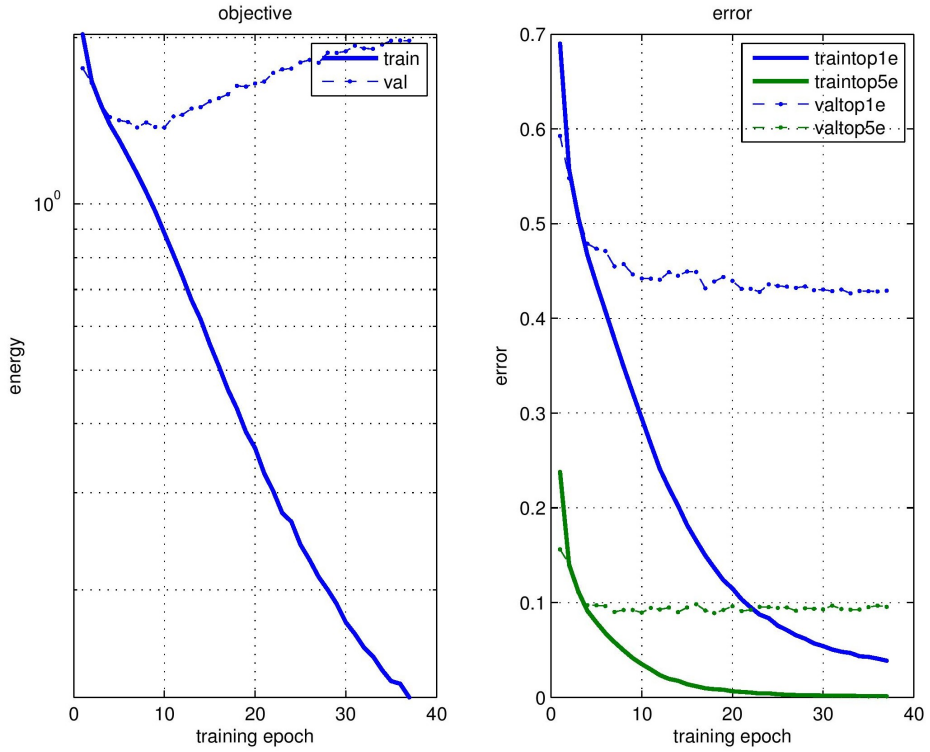


Figure 4.14 Energy and error plot for pre-trained VGG net fine-tuned on ACS dataset. Note that validation set for this dataset is the testing set.

4.7 Discussions

Our initial attempt to adapt the problem of Cloth Parsing to real-life unrestricted images succeeded in improving over previous methods. We use poselets to obtain pose information which is relatively more reliable in unrestricted settings. In further experiments, we explore different set of strategies to improve predictions on standard dataset. However all the experiments conducted have failed to improve upon previous approaches because of unconstrained complications which turned out to be difficult to formulate.

4.8 Summary

Understanding human clothing patterns and appearance has important consequences for human pose estimation, action recognition, surveillance, search and retrieval. Parsing for clothes in unconstrained settings is an important step towards better understanding of images. This paper proposes a method to segment clothes in images with no assumption on pose of the person, viewing angle, occlusion or clutter. An innovative method to model the cloth-joint co-occurrence has been described which is invariant to the above challenges. The efficacy of the algorithm is demonstrated on challenging datasets and is shown to outperform the previous attempt [64]. Unsuccessfully we investigate different techniques to further improve the performance.

Chapter 5

Conclusions and Future Work

Work presented here addresses the problem of understanding human appearance, to be precise namely, human pose estimation in videos and cloth parsing in images. Human pose estimation is a rudimentary problem and forms a prior knowledge for deeper level understanding of human parsing. One such deeper/higher level of parsing is Cloth Parsing, *i.e.* determining cloth types and regions worn by people. Chapter 2 introduces older and newer methods for human pose estimation and ends with an analysis that compares traditional state-of-the-art Pictorial Structure based models against recent Deep Neural Network architecture based methods. According to obtained results, DNN architecture based methods perform substantially better than traditional methods.

Chapter 3 presents our proposed methodology that fine-tunes human pose estimations in video sequences. The devised approach is a semi-supervised self-training pipeline that iteratively improves by grabbing positively correct estimations from base models and propagating attained locally fine-tuned parameters across the temporal sequence of testing video. Instead of a solitary base model (as in general self-training settings), we employ an amalgamation model which is a combination of a static base model and a dynamic ensemble of exemplars. We propose our novel batch selection criteria based on pose quality score that helps in picking correct estimations in each iteration of self-training. The resulting implementation thus obtained performs better than previous methods that use the same level of initial information from base models. Base model used here is Pictorial Structure based because of ease in understanding and modification.

In Chapter 4, we try to understand the intricacies involved with a problem involving a higher level of understanding : Cloth Parsing. Section 4.4 delineates our proposed approach for robust Cloth Parsing in more wild and real-life outdoor settings. We formulate the problem by incorporating more reliable human body part detectors (poselets) in a CRF framework. On observing the predictions qualitatively, the results appear to be confusing and far from being called reliable. In section 4.6, we show various strategies used in order to improve the efficacy of cloth type detection and cloth parsing. The experiments conducted have failed to improve upon previous approaches stating the need for a more deeper level of understanding and improved evidence correctness like better human poses.

Future Directions.

- **Human Pose Estimation:** Analysis done in Chapter 2 indicates that recent DNN based models perform much more reliably than Pictorial Structured based models. One possible future direction can involve using DNN-based models in place of the PS-based models as the base model to re-train which is a conceivably tougher task because of lack of deeper understanding. From the experimental observation showing the importance of an optimal batch selector, a substantial improvement in robustness over proposed pose quality ranking svm criteria can boost the overall performance. This task can also be termed as obtaining a reliable pose evaluator for videos.
- **Cloth Parsing:** In subsection 4.6.4, we conduct an experiment in which we train a Neural Network that does cloth-type classification in images. The network is trained on a bigger dataset for Apparel Classification and shows promising initial results. The work proposed here can benefit from such improved cloth recognition systems and the trained model can be parameterized for different data settings using domain adaptation techniques. Current strategies use superpixels to represent a region in an image which are noisy and impedes the learning of proper garment shapes. Improving the quality of region proposals will de-noise the learning to obtain better weights.

Related Publications

- Digvijay Singh, Vineeth Balasubramanian, C V Jawahar “Fine-Tuning Human Pose Estimations in Videos”, *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV), 2016*
- Nataraj Jammalamadaka, Ayush Minocha, Digvijay Singh and C V Jawahar “Parsing clothes in Unrestricted Images”, *Proceedings of the 24th British Machine Vision Conference (BMVC), 09-13 Sep. 2013, Bristol, UK.*

Bibliography

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. *CVPR*, 2009.

[3] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. J. V. Gool. Apparel classification with style. In *ACCV*, 2012.

[4] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *European Conference on Computer Vision (ECCV)*, 2010.

[5] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *CVPR*, 2009.

[6] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. In *IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision*, 2001.

[7] J. Carreira and C. Sminchisescu. CPMC: automatic object segmentation using constrained parametric min-cuts. *PAMI*, 2012.

[8] H. Chen, Z. Xu, Z. Liu, and S. C. Zhu. Composite templates for cloth modeling and sketching. In *CVPR*, 2006.

[9] M. Chen, J.-T. Sun, X. Ni, and Y. Chen. Improving context-aware query classification via adaptive self-training. In *CIKM*, 2011.

[10] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

[11] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *CVPR*, 2014.

[12] G. A. Cushen and M. S. Nixon. Real-time semantic clothing segmentation. In *ISVC (1)*, 2012.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] J. Dong, Q. Chen, X. Shen, J. Yang, and S. Yan. Towards unified human parsing and pose estimation. In *CVPR*, 2014.

[15] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *BMVC*, 2009.

[16] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *CVPR*, 2015.

[17] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[18] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005.

[19] V. Ferrari, M. J. Marín-Jiménez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[20] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, Jan. 1973.

[21] A. C. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *CVPR*, 2008.

[22] P. H. Gosselin, N. Murray, H. Jégou, and F. Perronnin. Revisiting the fisher vector for fine-grained classification. *Pattern Recognition Letters*, 49:92–98, 2014.

[23] B. Hariharan, P. A. Arbeláez, R. B. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

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

[25] L. Horne, J. Alvarez, M. Salzmann, and N. Barnes. Efficient scene parsing by sampling unary potentials in a fully-connected crf. In *IV*, 2015.

[26] N. Jammalamadaka, A. Minocha, D. Singh, and C. V. Jawahar. Parsing clothes in unrestricted images. In *BMVC*, 2013.

[27] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar. Video retrieval by mimicking poses. In *ICMR*, 2012.

[28] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010. doi:10.5244/C.24.12.

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

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

[31] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011.

[32] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 2004.

[33] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004.

[34] A. Mohan, C. Papageorgiou, and T. A. Poggio. Example-based object detection in images by components. *PAMI*, 2001.

[35] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, Aug. 2010.

[36] G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *ECCV*, 2002.

[37] B. X. Nie, C. Xiong, and S. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[38] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. In *CVPR*, 2014.

[39] C. Papageorgiou and T. A. Poggio. A trainable system for object detection. *IJCV*, 2000.

[40] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. *CoRR*, abs/1506.02897, 2015.

[41] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn. Posebits for monocular human pose estimation. In *CVPR*, 2014.

[42] V. Ramakrishna, T. Kanade, and Y. Sheikh. Tracking human pose by tracking symmetric parts. In *CVPR*, 2013.

[43] D. Ramanan. Learning to parse images of articulated bodies. In *NIPS*, 2006.

[44] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[45] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[46] U. Roy, A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition and retrieval for large lexicons. In *ACCV*, 2014.

[47] B. Sapp and B. Taskar. MODEC: multimodal decomposable models for human pose estimation. In *CVPR*, 2013.

[48] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. *ECCV*, 2010.

[49] B. Sapp, D. J. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *CVPR*, 2011.

[50] C. Scheffler and J.-M. Odobez. Joint adaptive colour modelling and skin, hair and clothes segmentation using coherent probabilistic index maps. In *BMVC*, 2011.

[51] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *IJCV*, 2004.

[52] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[53] E. Simo-Serra, S. Fidler, F. Moreno-Noguer, and R. Urtasun. A high performance CRF model for clothes parsing. In *ACCV*, 2014.

[54] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012.

[55] A. Subramanya, S. Petrov, and F. Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In *EMNLP*, 2010.

[56] J. Tighe and S. Lazebnik. Superparsing - scalable nonparametric image parsing with superpixels. *IJCV*, 101(2):329–349, 2013.

[57] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015.

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

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

[60] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[61] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013.

[62] N. Wang and H. Ai. Who blocks who: Simultaneous clothing segmentation for grouping images. In *ICCV*, 2011.

[63] M. Weber, M. Bäuml, and R. Stiefelhagen. Part-based clothing segmentation for person retrieval. In *AVSS*, 2011.

[64] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In *CVPR*, 2012.

[65] M. Yang and K. Yu. Real-time clothing recognition in surveillance videos. In *ICIP*, pages 2937–2940, 2011.

[66] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[67] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *PAMI*, 2013.

[68] B. Yao and F. Li. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[69] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *CVPR*, 2014.

[70] J. Yu, Y. Guo, D. Tao, and J. Wan. Human pose recovery by supervised spectral embedding. *Neurocomputing*, 2015.