

Human Layout Estimation using Structured Output Learning

D.Phil. Thesis

Robotics Research Group
Department of Engineering Science
University of Oxford



Supervisors:
Professor Andrew Zisserman
Professor Philip Torr

Arpit Mittal
Brasenose College

Hilary Term, 2012

Arpit Mittal
Brasenose College

Doctor of Philosophy
Hilary Term, 2012

Abstract

In this thesis, we investigate the problem of human layout estimation in unconstrained still images. This involves predicting the spatial configuration of body parts.

We start our investigation with pictorial structure models and propose an efficient method of model fitting using skin regions. To detect the skin, we learn a colour model locally from the image by detecting the facial region. The resulting skin detections are also used for hand localisation.

Our next contribution is a comprehensive dataset of 2D hand images. We collected this dataset from publicly available image sources, and annotated images with hand bounding boxes. The bounding boxes are not axis aligned, but are rather oriented with respect to the wrist. Our dataset is quite exhaustive as it includes images of different hand shapes and layout configurations.

Using our dataset, we train a hand detector that is robust to background clutter and lighting variations. Our hand detector is implemented as a two-stage system. The first stage involves proposing hand hypotheses using complementary image features, which are then evaluated by the second stage classifier. This improves both precision and recall and results in a state-of-the-art hand detection method. In addition we develop a new method of non-maximum suppression based on super-pixels.

We also contribute an efficient training algorithm for structured output ranking. In our algorithm, we reduce the time complexity of an expensive training component from quadratic to linear. This algorithm has a broad applicability and we use it for solving human layout estimation and taxonomic multiclass classification problems.

For human layout, we use different body part detectors to propose part candidates. These candidates are then combined and scored using our ranking algorithm. By applying this bottom-up approach, we achieve accurate human layout estimation despite variations in viewpoint and layout configuration. In the multiclass classification problem, we define the misclassification error using a class taxonomy. The problem then reduces to a structured output ranking problem and we use our ranking method to optimise it. This allows inclusion of semantic knowledge about the classes and results in a more meaningful classification system.

Lastly, we substantiate our ranking algorithm with theoretical proofs and derive the generalisation bounds for it. These bounds prove that the training error reduces to the lowest possible error asymptotically.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Arpit Mittal, Brasenose College

Copyright ©2012

Arpit Mittal

All rights reserved.

Acknowledgements

I am very grateful to my supervisors Professor Andrew Zisserman and Professor Phil Torr for introducing me to the amazing field of computer vision. Their guidance enabled me to think through the problems and make this contribution to the field. I learned a lot professionally and personally from both of them which will be invaluable to me for rest of my career.

I would like to thank my collaborators (and mentors): Matthew Blaschko, Alonso Patron and Esa Rahtu. We had both tough and happy times, but I always had a ball working with you.

It was an honour for me to be a part of the Visual Geometry Group where my lab-mates (and friends) provided me with a lovely work environment. I want to thank (not exclusively): Relja Arandjelović, Yusuf Aytar, Patrick Buehler, Yuning Chai, Ken Chatfield, Mircea Cimpoi, Amr El-labban, Siddharth Eswaran, Varun Gulshan, Minh Hoai, Nataraj Jammalamadaka, Mayank Juneja, Lubor Ladický, Victor Lempitsky, Meelis Lootus, Marcin Marszalek, Maria-Elena Nilsback, Omkar Parkhi, Rohan Paul, Tomas Pfister, James Philbin, Karen Simonyan and Andrea Vedaldi.

My time in Oxford has been enriched by the company of Siddharth Arora, Sabyasachi Baboo, Chetak Gandhi, and Pinky Raychaudhuri. I do not think I would have finished my DPhil without their encouragement and motivation. Thank you very much for always being there for me. A special thank you goes to Sophie Richter for being so caring.

Most importantly, I am grateful to my parents for believing in me and giving me freedom to follow my dreams. I also want to express my sincere regards to my uncle (Mr. OmPrakash Garg) and my aunt (Mrs. Vimla Goyal) for supporting me.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Challenges	5
1.3	Contributions	7
1.4	Thesis outline	10
2	Literature Survey	12
2.1	Object category detection	13
2.1.1	Image features	13
2.1.2	ROI selection	15
2.1.3	ROI representation	16
2.1.4	Verification	17
2.2	Hand detection	21
2.3	Pictorial structure	24
2.4	Human layout estimation	28
2.4.1	Example-based methods	29
2.4.2	Probabilistic assembly of parts	32
2.5	Taxonomic multiclass prediction	43

2.5.1	Building taxonomies	44
2.5.2	Learning using taxonomies	46
2.6	Large margin learning	46
2.6.1	Large-scale training of SVM	48
2.6.2	Structured output SVM	50
2.6.3	Ordinal regression SVM	52
2.6.4	1-slack formulation of SVM	54
2.6.5	Structured output ranking SVM	57
3	Datasets and Evaluation	59
3.1	Skin dataset	60
3.2	Buffy and PASCAL stickmen dataset	61
3.3	Hand dataset	62
3.4	Signer dataset	65
3.5	Indoor scene database	67
3.6	PASCAL VOC dataset	68
3.6.1	VOC 2007 classification dataset	68
3.6.2	VOC person layout dataset	70
4	Human body parts localisation using skin detection	72
4.1	Skin detection	73
4.1.1	Implementation details	74
4.1.2	Experimental results	77
4.2	Efficient human upper body layout	77
4.2.1	Overview	79
4.2.2	Hypothesis generation	82

4.2.3	Discriminatively trained parts-based model	83
4.2.3.1	Appearance features and pairwise potentials	84
4.2.3.2	Learning the model	87
4.2.4	Body parts localisation results	88
4.2.4.1	Search space reduction	89
4.2.4.2	Layout detection results	89
4.3	Hand detection using skin regions	93
4.3.1	PASCAL VOC 2010 results	95
4.4	Conclusions	95
5	Hand detection using multiple proposals	98
5.1	Proposal methods	99
5.1.1	Hand shape detector	100
5.1.2	Context detector	100
5.1.3	Skin-based detector	102
5.2	Classification of hand proposals	102
5.3	Super-pixel based NMS	105
5.4	Results	106
5.4.1	Hand dataset	106
5.4.2	Signer dataset	108
5.4.3	PASCAL VOC 2010 person layout test dataset	109
5.5	Rescoring using position & scale attributes	111
5.5.1	Results	114
5.6	Conclusions	115

6	Efficient structured output ranking SVM	116
6.1	Linear time constraint generation	117
6.2	Slack rescaling formulation	120
6.3	Margin rescaling formulation	122
6.4	Conclusions	124
7	Person layout and Taxonomic multiclass prediction	126
7.1	Person layout	128
7.1.1	Part candidates generation	129
7.1.2	Joint learning using structured ranking	130
7.1.2.1	Implementation details	130
7.1.2.2	Results	131
7.1.3	PASCAL VOC 2011 results	134
7.2	Taxonomic multiclass prediction	137
7.2.1	Experiments	140
7.2.1.1	Implementation details	140
7.2.1.2	Results	141
7.3	Conclusions	145
8	Generalisation bounds for structured output ranking	147
8.1	Slack rescaling vs. Margin rescaling	148
8.2	Generalisation bounds	151
8.3	Conclusions	157
9	Conclusions	158
9.1	Future work	160

9.1.1	Hand tracking	160
9.1.2	Markerless motion capture	161
9.1.3	Complete human body layout estimation	161
9.1.4	Consistency proof for structured output ranking	161
9.1.5	Taxonomy learning	162
Appendix A Line fitting algorithm		163
Appendix B More qualitative results for hand detection		165

Chapter 1

Introduction

The goal of this thesis is human layout estimation in images. Human layout estimation involves prediction of human body pose, i.e., spatial configuration of body parts (Figure 1.1). It is one of the major areas of research in the field of computer vision with numerous applications.

In this particular work, we are interested in performing accurate human pose estimation in unconstrained 2D images taken from any viewpoint. Existing methods fail to perform well when the human body is truncated or some of the body parts

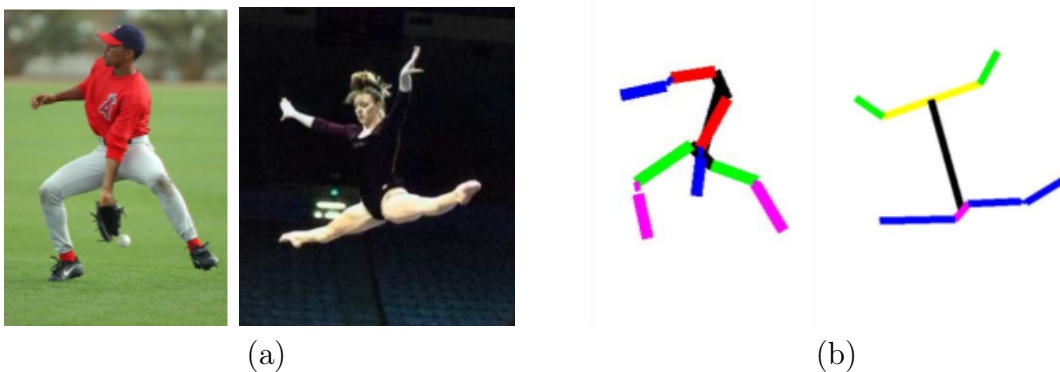


Figure 1.1: (a) Input image. (b) Pose estimation is characterised by the accurate localisation of the body parts. Images are taken from [Mori et al. \(2004\)](#).

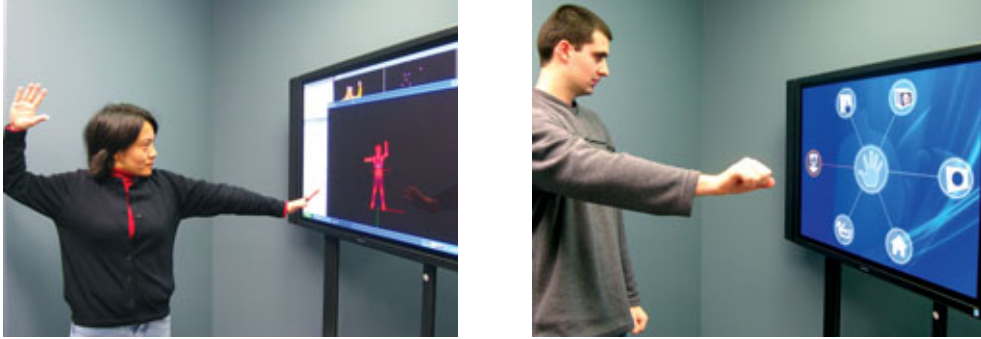


Figure 1.2: Two different examples of human computer interaction via body pose or gestures.

are occluded. Most existing systems can only deal with near frontal poses. However, we aim to build a system that works even when body parts are occluded and also for different viewpoints (front, side and back). The system should be able to estimate the layout despite high variation in the pose configurations and unusual lighting conditions.

In this thesis, we restrict our investigation to layout estimation of the upper body and localise the body parts up-to the extent of bounding boxes.

1.1 Motivation

An accurate layout estimation can enable machines to understand human pose and gestures, which could simplify many tasks, such as:

Human computer interaction. Human gesture could act as a new method of interacting with computers (Figure 1.2). Human gesture can be estimated by performing layout estimation on individual video frames. There are many commercial products like [Microsoft XBox Kinect](#), [Sony Eye Toy](#) etc. which are harnessing this idea for recreational purposes. In these devices, human gestures lead to specific

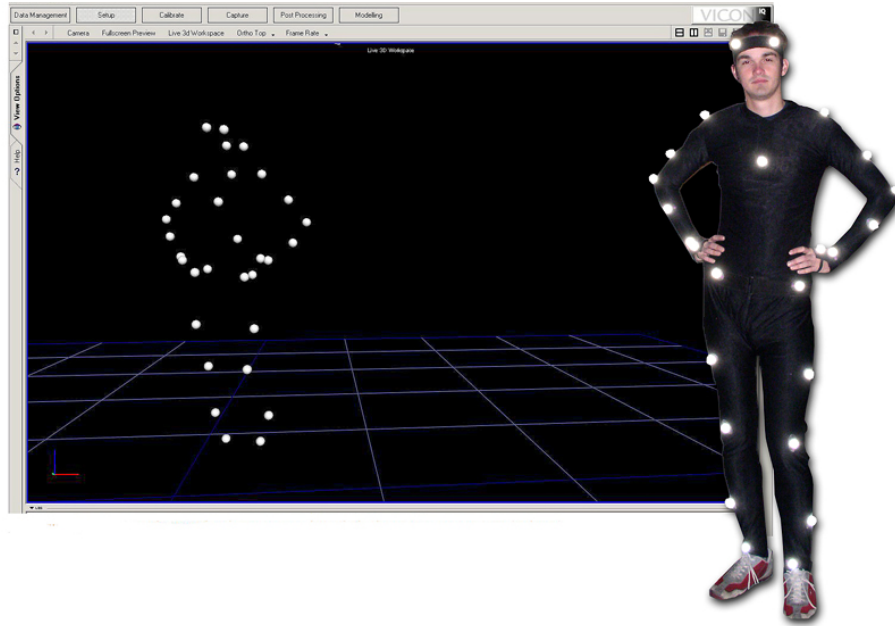


Figure 1.3: **Vicon Motion Capture** system. Optical markers are attached to different body parts, and are tracked in real time to get the pose estimate.

actions, which results in a more involved gaming experience. Similar technology could be extended to other devices like mobile phones and televisions where specific gestures may trigger different functionality (e.g., changing channels, switching the device off etc.).

Motion capture. Motion capture refers to capturing the motion of complete human body by tracking various body parts. Currently motion capture is performed by using numerous optical markers, which are attached to the body parts, and are tracked throughout the video to get the pose estimate of the person. Figure 1.3 shows the results from **Vicon Motion Capture** system.

Motion capture is done in a controlled lab environment and it requires a lengthy set-up procedure. An accurate human layout estimation system will make this procedure more efficient and this could be done without any use of markers. Motion



Figure 1.4: Top row shows the sign action for ‘golf’ and the bottom row shows the sign action for ‘tree’.

capture has already found applications in animation (for instance, it is used for making the movie ‘Avatar’), biomechanics and many other engineering related studies. All of these fields will benefit from a reliable human layout estimation system.

Sign language recognition. Sign language is the visual communication language used by people with hearing disability. However, its use restricts the communication between hearing and non-hearing people due to the need for a human translator for those who do not understand it. Figure 1.4 shows examples of two different signs from signing video sequences. A reliable human layout estimation can help in detecting hands, which can then be used for understanding the signs (Buehler et al. (2008, 2009)). Human layout estimation could be the building block for sign language recognition systems, which could be trained to automatically detect signs in videos.

Scene understanding. Once human pose is estimated, it can be used to retrieve video frames where similar poses are encountered. Thus, a novel retrieval system could be designed where videos are searched not by their title, but by the action or

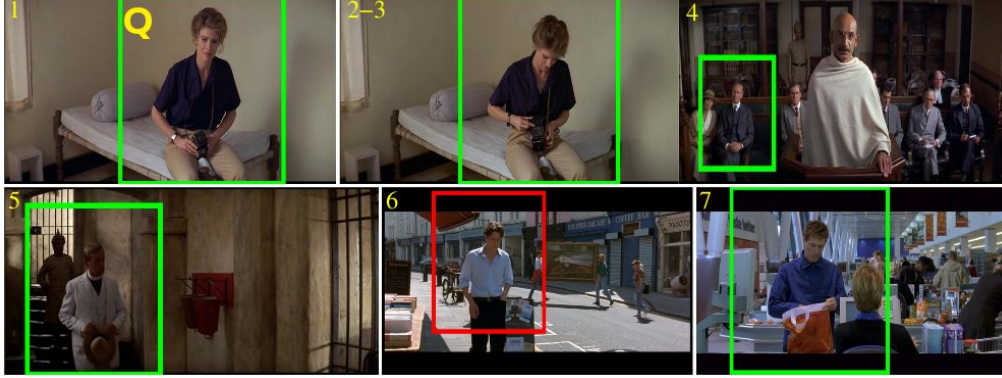


Figure 1.5: The frame marked with ‘Q’ is the query frame and the remaining frames are the retrieved frames for pose search system of [Ferrari et al. \(2009\)](#). The retrieved frames are having people in the same pose as in the query frame.

pose of people within it. Figure 1.5 shows the results of pose search system of [Ferrari et al. \(2009\)](#). This technique also finds wide application in action recognition ([Laptev et al. \(2008\)](#); [Patron-Perez et al. \(2010\)](#)), which leads to better scene understanding in images and videos.

Video surveillance. Video surveillance will benefit greatly from the inclusion of human pose estimation. A system could be designed to trigger an alarm when certain poses, such as ‘fell over’, are detected. This would build an intelligent surveillance system that could perform monitoring autonomously. This could result in a reduction of manual human effort to monitor surveillance videos.

1.2 Challenges

Layout estimation involves dealing with the wide variety of appearance of people due to clothing and lighting. In the majority of images, it is not possible to separate human body parts from the background using existing techniques. This is because

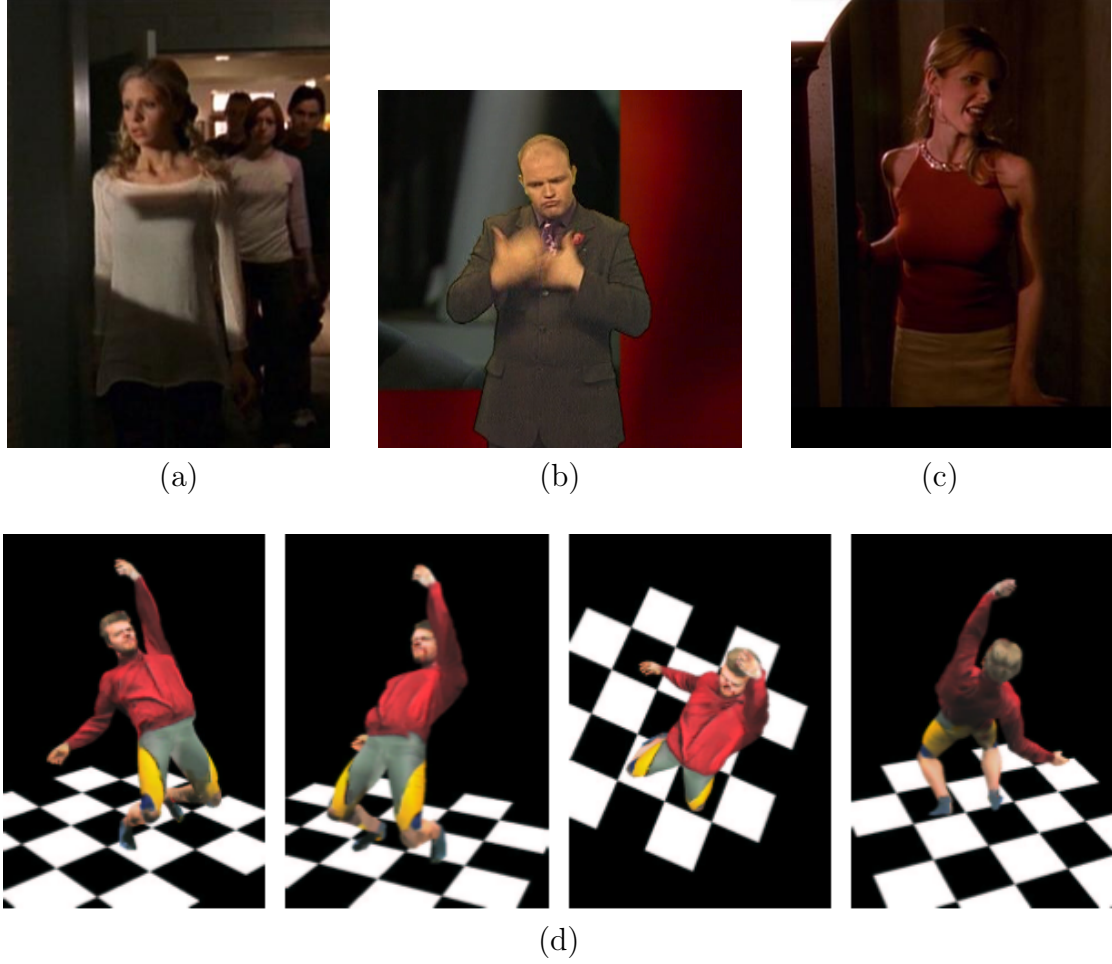


Figure 1.6: **Challenges in human layout estimation.** There are many reasons that make human layout estimation difficult: (a) Bad lighting significantly changes the appearance of the body parts and cluttered background makes background-foreground distinction difficult; (b) Motion blur makes the part boundaries fuzzy; (c) Occlusion of a body part may affect localisation of other parts, since the body parts are inter-connected ; (d) Change of viewpoint causes variation in 2D pose configuration and parts appearance. It also makes some of the body parts occluded.

of cluttered backgrounds or poor background-foreground distinction (Figure 1.6(a)). Often pictures are captured with insufficient lighting, and as a result human figures are not properly visible, which again leads to poor localisation.

Images also get spoilt by motion blur, which occurs when a picture of a moving

object is taken (Figure 1.6(b)). In the case of human body, due to the motion blur, part boundaries get fuzzy, which makes the body parts difficult to detect. Occlusion is another reason for poor detection (Figure 1.6(c)). If a part is occluded then it affects the localisation of other connecting parts as well (since they have to maintain a pose configuration). Another problem that has to be addressed, is changes in viewpoint (Figure 1.6(d)). When the viewpoint of the human body changes, then also its orientation within the image plane, as well as its appearance and 2D pose configuration changes. Furthermore, it may also cause some parts to become occluded.

1.3 Contributions

We can consolidate our major contributions as follows:

Hand dataset. We collected a comprehensive dataset of hand images from various public image sources, and annotated images with bounding boxes around hands, which do not have to be axis aligned, but are oriented with respect to the wrist. While collecting the data, we imposed no restriction on the pose or visibility of people, nor did we put any constraints on the surroundings. This dataset is quite exhaustive in terms of hand shapes and layout configurations.

A novel hand detector. We propose a novel hand detection system, which provides candidate hand bounding boxes in images. Our hand detector is capable of performing hand detection in all kinds of images, despite high variation in hand shapes and unusual lighting conditions. We build a two-stage hypothesis and classify framework for detecting hands and their orientation in unconstrained images.

The first stage uses three complementary detectors, based on colour, shape and context information, to propose hand bounding boxes. The second stage classifier computes a final confidence score for the proposals. The first stage detectors ensure good recall and the second stage classifier ensures good precision of our system. We show that our hand detector exceeds the state-of-the-art on various public datasets, including the PASCAL VOC 2010 person layout dataset (Everingham et al. (2010b)).

Super-pixel based non-maximum suppression. Non-maximum suppression (NMS) is a standard post-processing step in any typical object detection system. Under this method, the overlapping detection bounding boxes of object instances are removed. This method works well for bigger objects, such as cars, aeroplanes etc.. However, hands are often found either overlapping or very close to each-other in images and during non-maximum suppression, correct detections get removed. To avoid this, we propose a new method of non-maximum suppression based on super-pixels. In our method, we perform non-maximum suppression only on the bounding boxes overlapping a common super-pixel. We show that our method gives significantly better results compared to the traditional NMS technique.

Efficient structured output ranking training algorithm. We propose a new training algorithm for efficient learning of the structured multiclass problem for the ranking task. We improve upon existing methods and reduce the time complexity of an otherwise expensive component from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$, where n denotes the number of training samples. The algorithm that we have developed is of broad interest and can be used for solving many different tasks in computer vision. We use this algorithm to solve human layout estimation and taxonomic multiclass classification

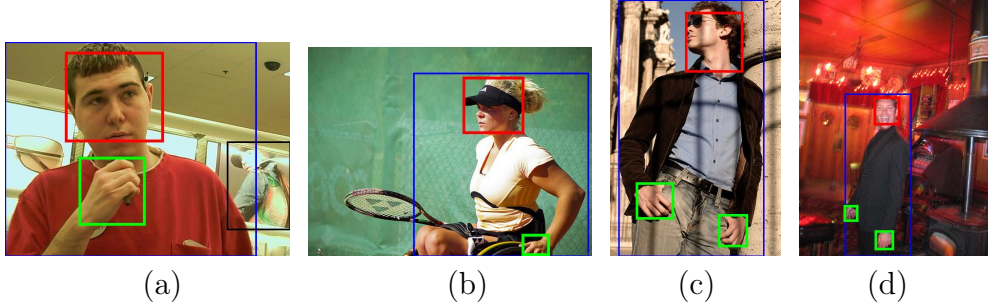


Figure 1.7: **Sample results of our method.** Green rectangle represents hand bounding box and red rectangle represents head bounding box. In (a), the image is truncated, also there is occlusion of the body parts. (b) and (d) capture side view of the person, which further results in occlusion. In (c), the upper arm is occluded. Despite of all these difficulties, our method gives accurate layout estimation, which is not possible using any other existing method.

problems.

Human layout estimation. We propose a system which performs very accurate human layout estimation despite variations in viewpoint, layout configurations and lighting. We treat the problem of human layout estimation as a generalisation of object class detection to multiple classes, where different classes refer to different body parts. Any layout induces structure in the part configuration space, which governs the relative positioning of body parts. Therefore, we formulate this problem as learning a multiclass structured space defined by the parts configurations. Appearance of body parts is used to propose human body part candidates that are later verified using a structured output ranking system, which encapsulates both the appearance and the relative positioning of the body parts. Figure 1.7 shows some sample results of our method. Using this method we also won the PASCAL VOC 2011 person layout competition (Everingham et al. (2011)).

Taxonomic multiclass classification. We use our structured ranking learning algorithm for multiclass classification, where classification error is calculated using taxonomies (class hierarchies). This facilitates inclusion of semantic knowledge regarding classes and gives more meaningful classification results. Using our method we report better results than all other related methods.

Generalisation bounds for structured output ranking. We are the first to derive the generalisation bounds for both the formulations of structured output ranking algorithm. The generalisation bounds show that the training error reduces asymptotically to the lowest possible error. This guarantees that our method converges to the same solution for unlimited amount of training data.

1.4 Thesis outline

Chapter 2 presents a review of existing methods for object category detection, specifically hand detection and human layout estimation. It also explains taxonomic multiclass classification and techniques for doing it. A separate section is devoted to explaining large margin classifiers and their large scale training. This section also presents different formulations of large margin classifiers of structured spaces for classification and ranking tasks.

Chapter 3 illustrates different datasets used for various experiments in the thesis. The evaluation criteria for each of the datasets is also given. In this chapter, we also present our dataset of hand instances.

Chapter 4 describes our method for skin detection. We use skin detection for efficient pictorial structure fitting and for hand detection. Experimental results are given to demonstrate the usefulness of our method for both the tasks. We also

present our results over the PASCAL VOC 2010 person layout challenge dataset (Everingham et al. (2010b)) for the hand detection task.

Chapter 5 takes a step towards more accurate hand detection system. Here we explain our method of hand detection using multiple proposals from different information sources. Skin detection, described in Chapter 4, becomes one of the components of this system. We further add shape and context information for hands and propose a very accurate hand detection method. We demonstrate the performance of our method both qualitatively and quantitatively. In the last section, we explain how we specialise hand detection results when the bounding box for the person is available, like in the PASCAL VOC person layout challenge (Everingham et al. (2010a)).

Chapter 6 presents our algorithm for large scale training of structured output ranking. We show that the run-time complexity of our algorithm is much lower compared to other existing algorithms.

Chapter 7 describes our method for human layout estimation and taxonomic multiclass prediction using structured output ranking. We show that our method gives better results compared to other related techniques. In this chapter we also present our method for PASCAL VOC 2011 submission (Everingham et al. (2011)).

Chapter 8 gives generalisation bounds for structured output ranking algorithms. We also discuss the significance of these bounds.

Chapter 9 summarises our contributions and lists possible directions for future research.

Chapter 2

Literature Survey

In this chapter, we survey the background to the object detection and learning methods that are employed in the thesis. In the first part of the chapter (Section 2.1), we will discuss various methods for object category detection used in computer vision. A more in-depth survey of hand detection methods is given in Section 2.2 as later we propose our own hand detection system. The pictorial structure model is introduced in Section 2.3, which is the building block for many of the human layout estimation systems discussed in Section 2.4. Taxonomic multiclass prediction and other methods of multiclass classification are described in Section 2.5. In Section 2.6, we introduce the Support Vector Machine (SVM) and explain various related methods of large margin classification, such as structured output SVM, ordinal regression and structured output ranking. The 1-slack formulation of SVM proposed by [Joachims \(2006\)](#) is also explained in this section, which we later extend to propose a fast method for training large scale structured output ranking problems.

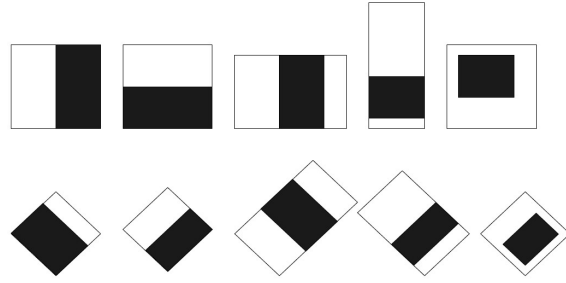


Figure 2.1: Sample Haar-like features. Adjacent rectangular regions are shown in different colours. The feature value is computed as the difference of the cumulative pixel intensity values in the adjacent rectangular regions.

2.1 Object category detection

Object category detection refers to the task of finding object category instances in images. This is a challenging task since it involves detecting instances of a given category up-to the object boundaries.

A typical object detection system involves two steps: (i) it hypothesises object ROIs, (ii) the hypothesised ROIs are verified using some learned or pre-defined model and depending on the matching score, it labels the ROI as the object class or background. In order to perform detection, the first task is to represent the object. This representation should be unique for the object and could be used to identify it correctly. This is done using image features, which are described below.

2.1.1 Image features

Image features are basically something peculiar in an image that are particular to an object and are also repeatable among object instances. They can be classified into two classes: (i) textural that defines the appearance of an object, (ii) structural that describes the structure or shape of an object.

Most commonly used textural features are colour ([Moeslund and Granum \(2000\)](#)),

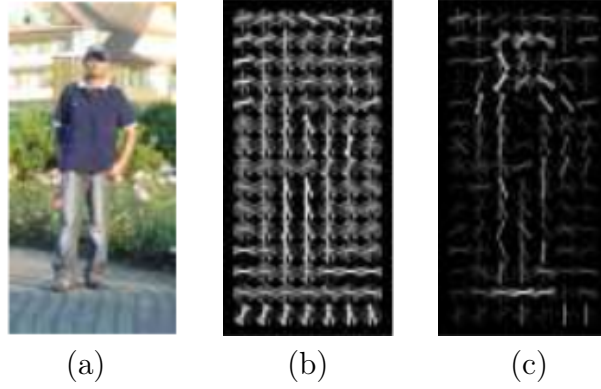


Figure 2.2: **HOG descriptors.** (a) The test image (b) HOG descriptors (c) HOG descriptors after re-weighting as per SVM weights.

filter banks output (Leung and Malik (2001)), Local Binary Pattern (LBP) (Ojala et al. (1996)) and Haar wavelets (Papageorgiou and Poggio (2000); Viola and Jones (2001)). LBP labels every pixel of an image with a binary number by thresholding the neighbourhood. Haar wavelets (adopted in computer vision as Haar-like features) consider the adjacent rectangular regions at a specific location in a detection window, sum up the pixel intensities in these regions and calculate the difference between them (Figure 2.1). Haar wavelets are very effective and are fast to compute using integral images. Other than these, sparse scale invariant feature transform (SIFT) (Lowe (2004)) features are also used to represent local texture and image gradient variations, where sparseness arises from pre-processing with an interest point detector (Mikolajczyk and Schmid (2004)).

Structural features typically involve normalised image gradient orientation histograms (Figure 2.2), computed over local image blocks in histogram of oriented gradients (HOG) (Dalal and Triggs (2005)) and dense SIFT. Another commonly used structural feature is shape context by Belongie and Malik (2002). In this, for every edge point, a coarse histogram of the relative co-ordinates of the remaining

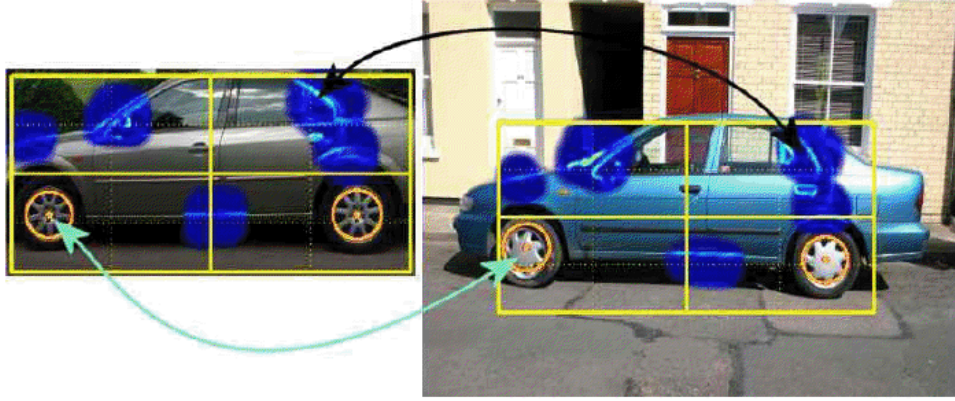


Figure 2.3: **Jumping window method.** An exemplar image and a corresponding car instance. The spatial correspondence between sparse visual word and dense edge distributions is learned. Some corresponding visual words and edges are highlighted in the image.

edge points is constructed.

Other than the above mentioned features, images can also be represented as ‘bag of visual words’ (Sivic and Zisserman (2003)), which capture the frequency of iconic image patches or fragments and represent it as histograms.

2.1.2 ROI selection

The straight-forward way to obtain initial object location hypotheses is the sliding window technique where detector windows at various scales and locations are slid (shifted) over the image (Dalal and Triggs (2005); Felzenszwalb et al. (2010b); Mohan et al. (2001); Papageorgiou and Poggio (2000)).

Another strategy employs interest-point detectors to recover regions with high information content based on local discontinuities of the image brightness that often occur at the object boundaries (Agarwal et al. (2004); Leibe et al. (2005); Lowe (2004)). ‘Jumping window’ is proposed by Chum and Zisserman (2007) in which

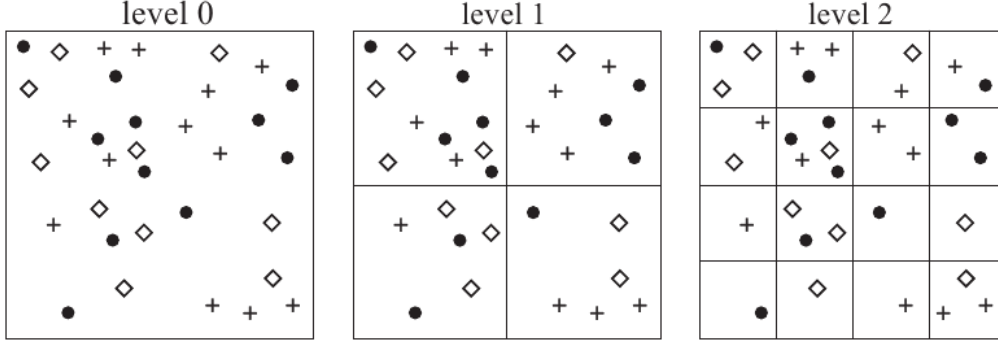


Figure 2.4: **Spatial pyramid model.** The image is subdivided at three different levels of resolution. Each grid is represented as a bag of words. The final output is a weighted concatenation of spatial histograms.

the relation between interest points and the object location is learned to predict the object location in the new image (Figure 2.3). Lampert et al. (2009) developed a branch and bound technique to directly search for the optimum window within an image. Maji and Malik (2009) combine these relations to predict the object location using a Hough transform. A selective search using segmentation is performed by van de Sande et al. (2011) to generate a limited set of object hypotheses.

2.1.3 ROI representation

The appearance of a ROI is represented by the image features (i.e., an array of features). Recently many approaches attempt to break down the complex appearance of the object class into manageable subparts and the ROI is then represented as a combination of parts features. Spatial pyramids were proposed in Lazebnik et al. (2006) (Figure 2.4) for scene classification and applied to object detection by Vedaldi et al. (2009) for different feature channels in a multiple kernel learning framework. A spatial pyramid is a three-layer pyramid where cells at different levels of the grid specify histogram of visual words in the corresponding spatial domain yielding

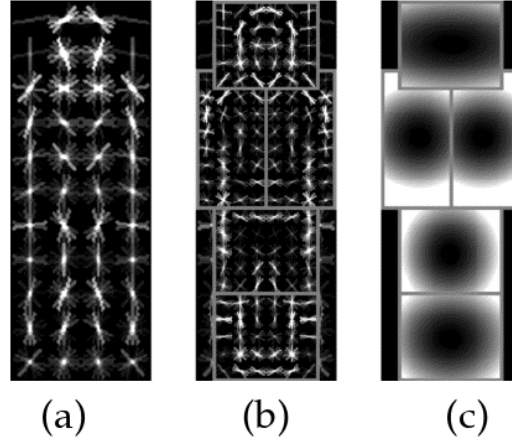


Figure 2.5: **Deformable parts-based model of pedestrian.** (a) The root filter. (b) Spatial model for the location of each part relative to the root. (c) The deformation cost for different part filters relative to the root.

a coarse-to-fine representation. A different approach is the parts-based model of [Felzenszwalb et al. \(2010b\)](#). It is a two-layer structure where the root node represents the entire object while nodes at the second layer correspond to the parts as shown in Figure 2.5. The part nodes are not fixed but are allowed to move relative to the root node to account for large deformations of the object configuration.

[Chen et al. \(2010\)](#) combine the two approaches, and use spatial pyramid and parts-based model in a single representation. In this representation, a ROI is described using a mixture of hierarchical trees where the nodes represent the object and its parts in a pyramid form. [Zhu et al. \(2010\)](#) used a deeper parts-based model having a three-layered tree structure.

2.1.4 Verification

After the initial hypotheses are generated, further verification involves applying appearance models and using spatial cues.

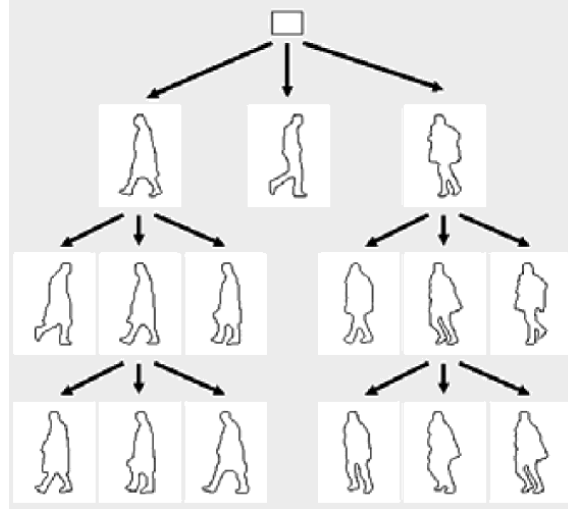


Figure 2.6: A hierarchical structure for pedestrian exemplar shapes used in Gavrilu (2007).

Template matching. These methods follow a generative pipeline, and store an object model as a set of example templates (Gavrilu (2007); Toyama and Blake (2001)). For verification, shapes of the stored examples are matched with the object hypotheses using distance transforms (Breu et al. (1995)). This results in high specificity since only plausible examples are included, but it requires a large amount of examples to sufficiently cover the appearance space of the object (Figure 2.6). Continuous shape models are also proposed which involve a compact parametric representation of the set of training shapes (Heap and Hogg (1996)). Mikolajczyk et al. (2006) learn a code book using SIFT features (Lowe (2004)) extracted over boundary interest points and use hierarchical model for efficient object detection. These methods typically do not require a ROI selection stage as they involve object detection by matching with stored templates.

Discriminative classification. Template matching methods require training data that represents all possible configurations of an object class. This is not feasible for

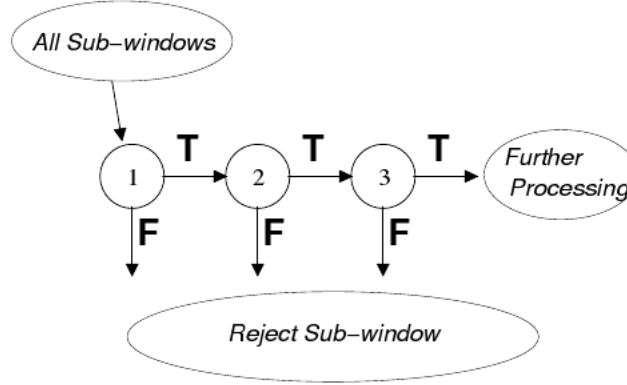


Figure 2.7: Schematic representation of the detection cascade used by [Viola and Jones \(2001\)](#). The complexity of classifier increases as we go from left to right. Initial stage classifiers eliminates a large number of negative examples with a little processing. After many such stages the number of hypotheses windows reduces radically. This results in an accurate detection system with less computational cost.

many practical applications because of limitation on the availability of data. Furthermore, these methods do not generalise easily, which make them less useful for modelling highly deformable objects (for instance, hands). In contrast, discriminative methods show better generalisation capabilities and are less prone to over-fitting ([Bishop \(2006\)](#)). Moreover, they do not require very large amount of training data for model learning. For these reasons, discriminative methods have recently become the most commonly used method for object modelling. Discriminative classifiers are learned using image features in such a way that higher score is assigned to the ROIs having the target object.

Linear Support Vector Machines (SVM) (Section 2.6) are very widely used as the classification method ([Dalal and Triggs \(2005\)](#)). Non-linear SVM classification using polynomial or radial basis kernels yields further improvement in the performance ([Mohan et al. \(2001\)](#); [Papageorgiou and Poggio \(2000\)](#)). [Felzenszwalb et al. \(2010b\)](#) use latent SVM with linear kernel to train their parts-based model, where

the positions of parts are considered as the latent variables. The training is performed by alternating between prediction of the latent variables and optimisation of the learning objective.

Ada-boost (Freund and Schapire (1996)) has also been used to construct strong classifiers as weighted linear combination of the selected weak classifiers, each involving a threshold on a single feature. To incorporate non-linearities and speed up, boosted detector cascades have been introduced by Viola and Jones (2001) (Figure 2.7) and adapted by many others (Mikolajczyk et al. (2004); Romdhani et al. (2001); Viola and Jones (2001); Wu and Nevatia (2006); Zhu et al. (2006)).

In contrast to methods that look for a particular class, Alexe et al. (2010) proposed to search for any object independent of its class. They train a classifier to detect foreground objects on the windows of those objects which have a well-defined shape. The boxes with the highest ‘objectness’ measure serve as the set of object hypotheses.

Among all methods discussed so far, the parts-based model of Felzenszwalb et al. (2010b) is currently the *de facto* approach for the task. Due to its high generalisation ability and capacity to model object deformations, it has been used to model different object categories (PASCAL VOC 2010 results; PASCAL VOC 2011 results). Recent approaches also showed significant improvement in performance by incorporating contextual information about an object (Desai et al. (2009); Harzallah et al. (2009); Torralba et al. (2004)). This motivated us to use the parts-based model of Felzenszwalb et al. (2010b) for modelling body parts and also to incorporate contextual information to further improve the detection results.

2.2 Hand detection

The human hand is a highly deformable object with 27 degrees of freedom (DOF): 4 in each finger, 3 for extension and flexion and one for abduction and adduction; the thumb is more complicated and has 5 DOF, leaving 6 DOF for the rotation and translation of the wrist (ElKoura and Singh (2003)). A robust hand detection has to cope with these huge variations in shape configurations. This is a tremendously challenging task as hand images can be taken from different viewpoints, hands can be closed or open, be partially occluded, have different articulations of the fingers, be grasping other objects or other hands, etc..

Having a reliable hand detector facilitates many other tasks in human visual recognition, such as determining human layout (Andriluka et al. (2009); Ferrari et al. (2008); Kumar et al. (2009); Everingham et al. (2010b)) and actions from static images (Delaitre et al. (2010); Gupta et al. (2009); Everingham et al. (2010b); Yao and Fei-Fei (2010)). It also benefits human temporal analysis, such as recognising sign language (Buehler et al. (2008); Farhadi and Forsyth (2006)), gestures and activities in video.

Hand detection investigation originally started using non-vision based approaches. This involves the use of sensors and relying on measurements from position sensors and data gloves (Sturman and Zeltzer (1994)). With these, detailed measurements of the hand position and of the hand shape can be obtained, including flexing angles of the fingers, as well as yaw, pitch and roll of the hands.

Early vision-based approaches for hand detection consisted of searching for skin-coloured or glove-coloured areas in the images. For example, in Bowden et al. (2004), localisation of hands is performed by locating a specific coloured patch in the image.



Figure 2.8: Multi-camera system of [Vogler and Metaxas \(1998\)](#) to segment out the hands and arms in 3D.

While this makes hand detection trivial, it cannot be performed for a real world image. Alternatively, the hands can be found by identifying one or two clusters of skin-coloured pixels as in [Farhadi et al. \(2007\)](#); [Wu et al. \(2000\)](#); [Zhu et al. \(2000\)](#). However, this technique requires a very accurate global model of skin-coloured pixels and does not perform well when the hands are overlapping or a hand is in front of the face.

Colour segmentation is applied by [Buehler et al. \(2008\)](#) to detect hands in a video sequence. The hand colour model is learned from the first few manually annotated frames of the sequence. Thus, this method is not fully automated. Hand detection has also been performed by leveraging the three-dimensional information of the hand obtained using three orthogonal cameras to segment the arms ([Vogler and Metaxas \(1997, 1998, 1999, 2004\)](#)). However, this system can not give a very precise segmentation of only the hand as it specialises in segmenting out combined hand and arm (Figure 2.8). Hand detection is also performed by frequency spectrum analysis using the Fourier transformation ([Kolsch and Turk \(2004\)](#)). However, this method works only for very limited scenarios.

More recent approaches apply more complex hand detection approaches. For example, boosting methods have been applied for hand detection in [Ong and Bowden](#)

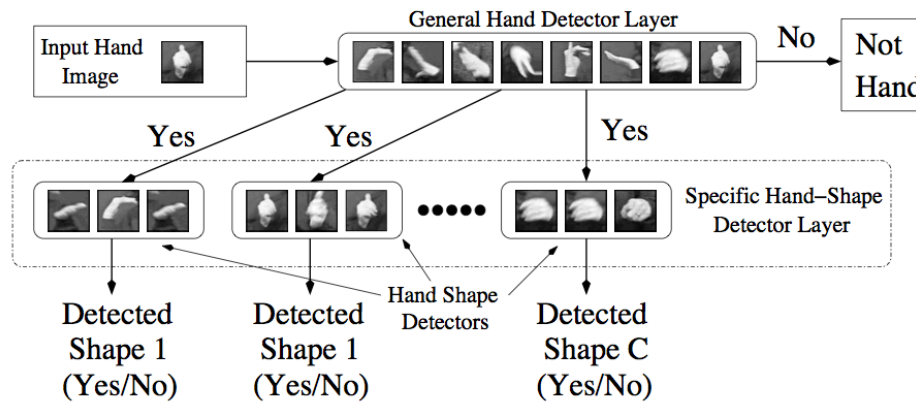


Figure 2.9: The framework of a tree of hand detectors in Ong and Bowden (2004).

(2004); Kadir et al. (2004) where Ada-Boost is employed to form a hand detection classifier (Figure 2.9). In Stenger (2006), colour and motion features are used to hypothesise the hand location. These methods present good results only for training and test data with fairly simple and similar backgrounds. Finally, the hand can also be detected in the process of human layout estimation by some of the systems covered in Section 2.4.

Many hand tracking systems initialise by performing hand detection in the first frame. In [Stenger et al. \(2004, 2006\)](#), a hierarchical (tree-based) classifier is employed at the initialisation stage. The multi-resolution tree corresponds to clusters of similar hand poses. Traversing the tree based on hand template matching provides the hand pose estimate. In [Tomasi et al. \(2003\)](#), real images were manually labelled for hand poses and a binary classifier tree is constructed by unsupervised clustering of features. [Athitsos and Sclaroff \(2003\)](#) introduce indexing methods for hand template matching over a dataset of hand shapes (Figure 2.10). They perform chamfer matching and a probabilistic line matching algorithm for this purpose. In [Rosales et al. \(2000\)](#), a general machine learning approach is proposed to learn a mapping

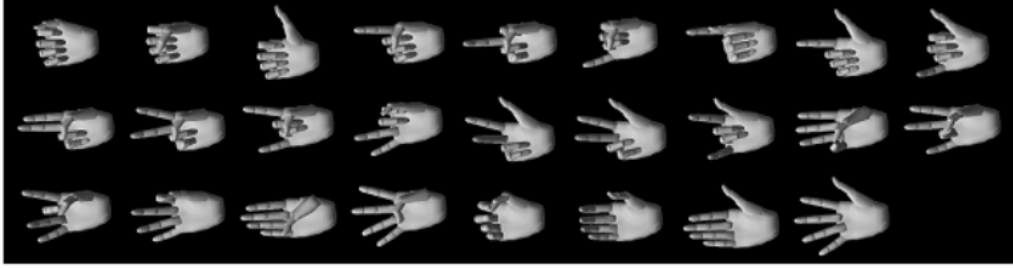


Figure 2.10: 26 basic hand shape templates used for template matching by [Athitsos and Sclaroff \(2003\)](#).

from a 2D feature space to the 3D hand pose parameter space. This is done by employing Specialised Mapping Architecture to implement the mapping of rotation and scale invariant moments of the hand silhouette. Since the tracking systems use detection as the initialisation step, they do not perform a very precise hand localisation. Each of the methods discussed so far specialises for a particular scenario and cannot detect all possible hand configurations. The aim of our research is to make a generic hand detector that works without any limitation as imposed by these methods. We present our method of generic hand detection in Chapter 5.

2.3 Pictorial structure

Many objects, for instance the human body, are a composition of rigid small connected parts. The relative positioning of these parts results in different configurations of the whole object. Such objects are better represented as a collection of interconnected small part-models, instead of a singleton structure ([Felzenszwalb and Huttenlocher \(2005\)](#)). In order to maintain the structural integrity, pairwise constraints are defined between these parts-models which keep parts at stable positions within the complete object. These kinds of parts-based models are known as *pictorial structures*

(Fischler and Elschlager (1973)) and they have the following characteristics: (i) A model for every part defines the likelihood for the part. (ii) Pairwise constraints between the parts give dependencies between the parts. (iii) The joint posterior probability of the object is defined as the product of likelihoods and dependencies over all parts. The optimum configuration of parts is obtained by maximizing the joint posterior probability. This process is known as MAP (maximum a posteriori) probability estimation.

For example, if a human body is defined as a collection of n parts, then the joint posterior probability distribution of the configuration of parts at locations $(L = l_1, l_2 \dots l_n)$ given an image I is defined as (Felzenszwalb and Huttenlocher (2005)):

$$P(L|I) \propto \prod_{i \in \nu} p(I|l_i, u_i) \prod_{(i,j) \in \varepsilon} p(l_i, l_j | c_{ij}) \quad (2.1)$$

where ν is the set of parts and ε is the set of edges indicates the connectivity between the parts. The parameter u_i defines the parameter for the part-model and c_{ij} models the pairwise relations. In the context of human body layout estimation, $p(I|l_i, u_i)$ represents the appearance model for the body parts like shape or colour, and the pairwise term $p(l_i, l_j | c_{ij})$ controls the expected Euclidean distance between the two connected parts.

For human layout, pairwise dependencies are modelled as the deformation between the parts. Thus, for Equation (2.1), $p(l_i, l_j | c_{ij}) = -\lambda f(d_i, d_j)$, where λ is a constant, d_i is the distance of part i from a reference point and f is a convex function of the two distances. One such example is given in Figure 2.11. Felzenszwalb and Huttenlocher (2000) showed that, assuming a tree-like connectivity between parts and

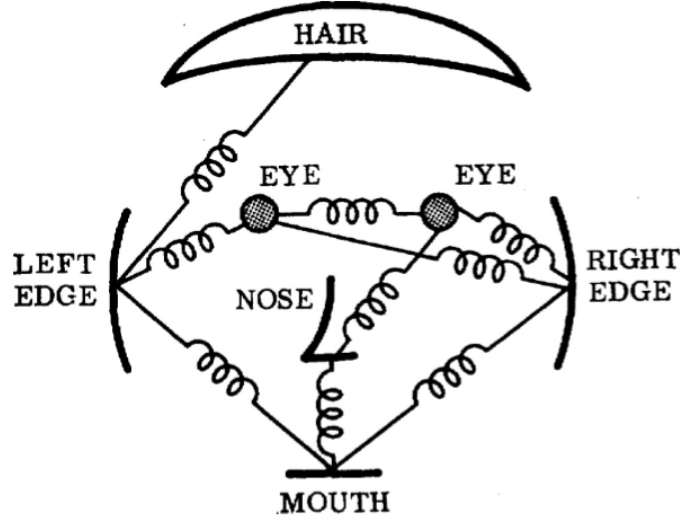


Figure 2.11: Part model used by Fischler and Elschlager (1973) for facial feature detection. Dependencies between parts are depicted as springs which enforces geometric constraints on the part locations.

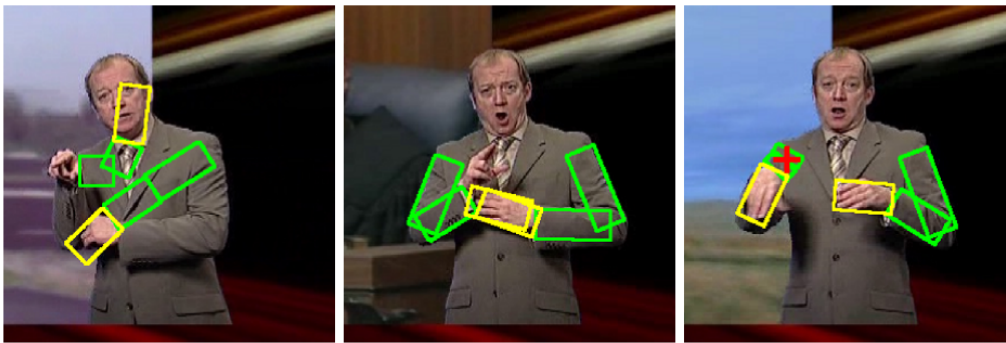
a certain form of the pairwise deformation function f , the configuration L with the highest probability (MAP estimation) can be found in time linear in the number of part locations. For any other kind of pairwise cost, MAP can be computed using a dynamic programming method known as the Viterbi algorithm (Viterbi (1967)) in quadratic time in the number of part locations.

For some applications, it is desirable to draw samples from Equation (2.1) instead of finding the MAP estimate. Felzenszwalb and Huttenlocher (2005) showed that sampling from a pictorial structure is facilitated by the restriction of tree-like topologies and can as a result be performed efficiently. Due to this advantage, many of the state-of-the-art methods for upper body layout detection are based on pictorial structure (Section 2.4).

Drawback of tree-structured pictorial structures. Even though the run-time performance of pictorial structure is very compelling, such an approach has



(a) Images



(b) Arm configurations with high probability under a pictorial structure model

Figure 2.12: Drawbacks of pictorial structure. (a) Example images (b) Highest probability arm configurations obtained by MAP estimation of the pictorial structure model. (b, left) is an example of *ignoring evidence* - the pictorial structure model only explains the foreground, i.e., pixels which are covered by model. Each of the arm parts lie on the appropriate colour, even though this leaves skin pixels of the true hand unexplained. (b, middle) an example of *over-counting of evidence* - each pixel can erroneously contribute evidence for multiple parts. In the example, the left and right hand of the estimated configuration are predicted to occupy the same image area. (b, right) if *occlusions* are not modelled correctly, the estimated upper and lower arms are very unlikely to lie at their true locations, which are occluded by the hand. The estimated position of the left elbow marked with a red cross. These images are taken from [Buehler \(2010\)](#).

several limitations due to the independence assumptions implied by the use of a tree-structured model (an example is shown in Figure 2.12):

- Ignoring evidence: Only pixels which are covered by the model contribute to the overall probability estimation. Thus, any negative evidence in the background is missed. This is evident from the appearance term $p(I|l_i, u_i)$ in Equation (2.1) which is evaluated for each part separately and represents a local measure of a part occurring with a certain configuration.
- Over-counting of evidence: Pixels can contribute more than once to the cost function and hence multiple parts can explain the same image area (Sigal and Black (2006)). This behaviour follows directly from the independence assumption in Equation (2.1) which states that the appearance term of each part can be evaluated individually and is not affected by other parts.
- No modelling of occlusions: In its original form pictorial structure does not allow for modelling occlusion.

2.4 Human layout estimation

Human body layout estimation methods can be broadly categorised into 2 classes, *example-based methods* and *probabilistic assembly of parts*, on the basis of how they model the human body. Methods of the first kind learn the whole human body layout (either 2D or 3D) as a single entity and then detect it on the test images by performing inference. Methods belonging to the later class treat the human body layout as an articulated structure and localise it as an assembly of parts. These methods are explained in greater details below.

2.4.1 Example-based methods

Example-based methods are a top-down approach of human pose estimation where a generative model for the human body is learned, treating the whole body as a single structure. In these methods a large number of human poses are stored in memory for template matching. At the detection time, the template that gives the best matching score is declared as pose for the body. These methods represent the mapping between image and pose space which could even be extended to estimating a 3D pose. They can also handle arbitrarily complex priors on the pose and appearance easily. These are some of the earliest techniques explored due to their simplicity.

A number of example-based methods have been proposed in recent years which compare the observed image with a database of samples. In [Rosales and Sclaroff \(2000\)](#) this technique is used to capture full body human posture. The authors learn a mapping of visual features of segmented persons to static pose using neural networks. In this method, the images from the same viewpoint are clustered together and a neural network is trained for each such cluster. For new visual features, a mapping from each cluster is performed providing a set of possible poses.

In [Mori and Malik \(2002\)](#), the authors use these methods to estimate three-dimensional human body configurations from a single two-dimensional image (Figure [2.13](#)). To do this, the body joints are located in the test image and then using shape context matching ([Belongie et al. \(2000\)](#)), the best matching template is found corresponding to the detected joints. [Shakhnarovich et al. \(2003\)](#) present an example-based approach for viewpoint invariant pose estimation of human upper-body from a single image. A new parameter-sensitive hashing is used that hashes stored templates relevant to a particular task. This methodology enables them to estimate rapidly and accurately full body articulated poses of human figures from a

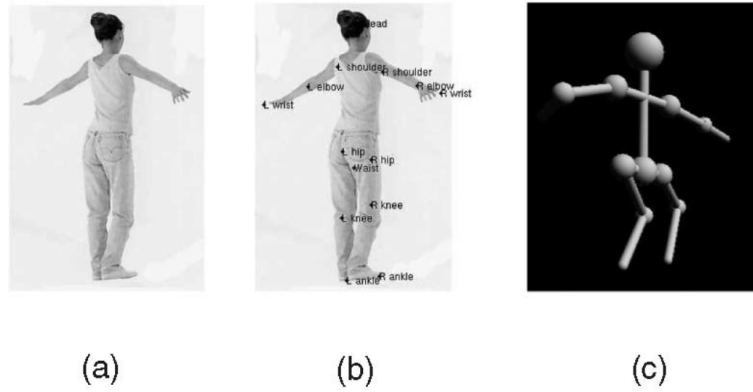


Figure 2.13: Estimation of 3D human body configurations from a single 2D image in [Mori and Malik \(2002\)](#). (a) Input Image. (b) Automatically extracted keypoints. (c) Three dimensional rendering of estimated body configuration.

large database of example images.

In all these methods, human body appearance does not play a major role in modelling. It is used only to segment human figures from images using background subtraction or similar methods. This made researchers focus on estimating human poses out of silhouettes (Figure 2.14). The major advantage of using silhouettes is that they are insensitive to surface attributes like clothing colour and texture and can be extracted reliably from images. [Howe \(2004\)](#) uses silhouettes to track the articulated 3D pose of human beings moving through video sequences. The temporal continuity is modelled in a video sequence with a Markov chain. Using this, a direct look-up technique is proposed mapping silhouettes to the 3D poses.

All methods discussed so far suffer from the problem of storing a significant number of templates from the ground-truth data. Their estimation ability is limited to the poses used in the training data. In [Agarwal and Triggs \(2004, 2006\)](#), a sparse Bayesian non-linear regression is used to distil a large training database into a compact model that has good generalisation to unseen examples (Figure 2.14).



Figure 2.14: Examples for parts localisation inside a silhouette by [Agarwal and Triggs \(2004, 2006\)](#).

Using this technique, the authors were able to reconstruct a long sequence of walking motion from the frames extracted from a monocular video.

3D pose estimation is also performed by learning a mapping between image space and the parameter space of human poses (Figure 2.15). This mapping is learned using a Gaussian process latent variable model ([Ek et al. \(2007\)](#)). In this work, the mapping is represented as the latent space which encapsulates both pose and silhouette features. However, the model developed in this work is not capable of consolidating multiple complementary representations. This method is therefore extended by including additional representation specific information in the latent spaces ([Ek et al. \(2008\)](#)). In [Navaratnam et al. \(2007\)](#) this mapping is learned as a low-dimensional joint manifold model. This enables inference on unlabelled data and also to learn one-to-many mappings between image and pose space.

The major disadvantage associated with example-based methods is that they can infer only poses that have been encountered at training time. Some of the methods

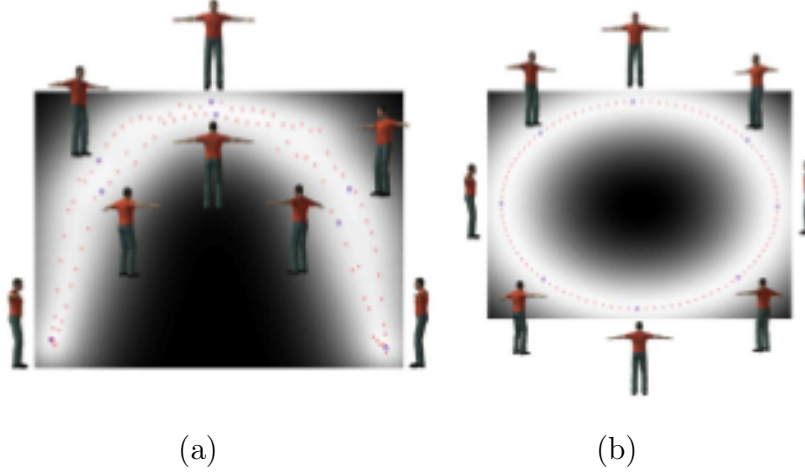


Figure 2.15: Human pose estimation by learning a mapping between feature space and the pose space (Ek et al. (2007, 2008); Navaratnam et al. (2007)). (a) The feature space and (b) The pose space for human layout.

discussed above also require background subtraction, which is not possible in highly cluttered images.

2.4.2 Probabilistic assembly of parts

Probabilistic assembly of parts methods are recent compared to their counterparts and have become very popular in modelling human body layout. In this approach, human body layout is modelled bottom-up, i.e., the body is represented as a combination of individual parts. In these methods, body parts are detected first and are then assembled together to obtain a configuration which best matches the observations. An important contribution of this approach is the pose estimation in cluttered natural scenes from a single view. This overcomes limitations of many previous pose estimation methods that require structured scenes, accurate prior models or multiple views.

In this method, the model can be divided into two parts, a sub-model for the

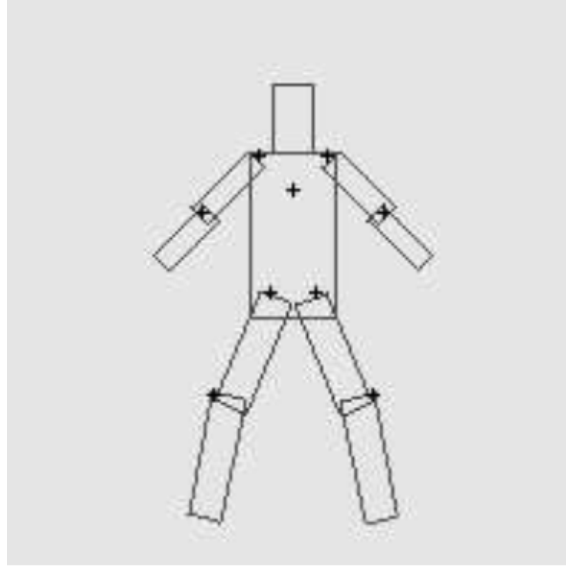


Figure 2.16: Human body representation in Felzenszwalb and Huttenlocher (2005). The anchor positions for the body parts are shown as ‘+’ in the figure. The deformation cost is calculated depending on the displacement of body parts from the anchor position.

appearance of the body parts and a sub-model for structure of the entire body. Now, these sub-models could be generative as well as discriminative. In case of generative model, a template is learned for appearance or structure, while in discriminative approach, a classifier is learned to segregate valid features from the invalid ones.

These methods model human body as the combination of individual body parts. Hence, they require a representation for the structural assembly of the human body. Forsyth and Fleck (1997) introduce the notion of ‘body plans’ to represent people. Following this direction, pictorial structures are introduced (Felzenszwalb and Huttenlocher (2000, 2005); Fischler and Elschlager (1973); Ioffe and Forsyth (1999)) and are used extensively to represent the human body configurations (Section 2.3). They provide a powerful framework to express parts and their combinations (Figure 2.16). Apart from these, combination of part detectors (Mohan et al. (2001); Ron-

[Ranfard et al. \(2002\)](#)) are also used which classify a valid combination of parts from an invalid one. In these methods, structure of the human body is learned rather than expressing it explicitly.

In [Ranfard et al. \(2002\)](#), individual body parts are detected using SVM classifiers (Section 2.6) and joints are learned using adaptive combination of classifiers ([Mohan et al. \(2001\)](#)). In adaptive combination of classifiers framework, a combined classifier is learned which classifies the output of part classifiers and say whether a pattern is a ‘person’ or a ‘non-person’ (Figure 2.17). [Mikolajczyk et al. \(2004\)](#) introduce probabilistic assemblies of robust Ada-Boost body part detectors to locate people in images providing a coarse 2D localisation. The probabilistic assembly of parts models the joint likelihood of a body part configuration.

In all the methods discussed so far, human body parts were modelled as cylinders (3D) or rectangles (2D). However, researchers also worked on representing them as free-shapes. [Mori et al. \(2004\)](#) first segment the image into super-pixels and larger regions or segments. Then perform a combinations search in the space of super-pixels to recover full body configurations. While doing so they prune away impossible configurations of body parts by enforcing the global constraints such as relative scale and symmetry in clothing (Figure 2.18).

[Ren et al. \(2005\)](#) use pairwise constraints between body parts to assemble body parts detections into 2D pose configurations. They formulate the problem as one of integer programming over pairwise constraints between the body parts. This allows them to incorporate much more information than the traditional dynamic programming method used to estimate the most probable human body layout. [Hua et al. \(2005\)](#) present an approach for 2D estimation from a single image using bottom-up feature cues together with a Markov network to model part configurations. In

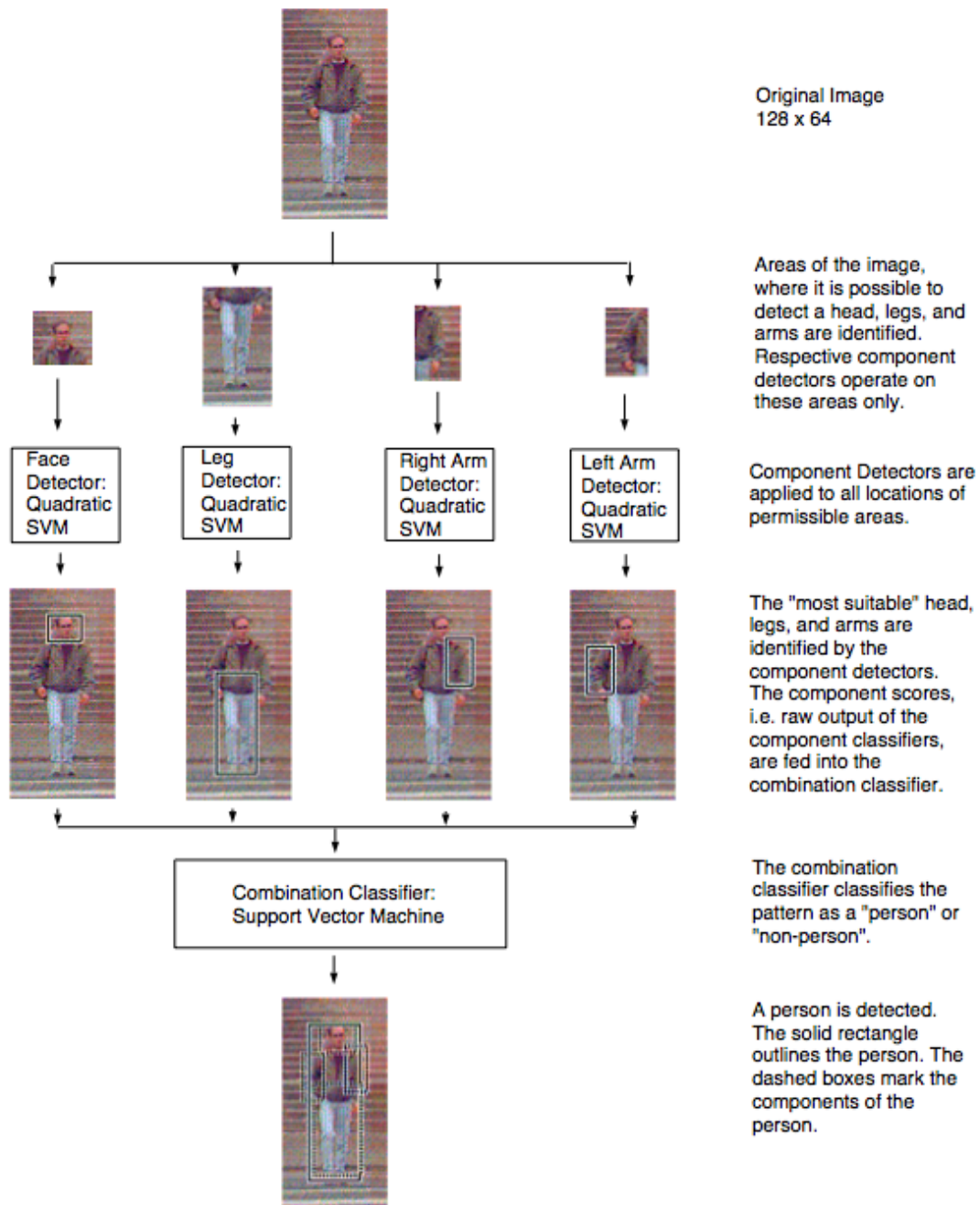


Figure 2.17: Diagrammatic description of adaptive combination of classifiers system developed by [Mohan et al. \(2001\)](#).

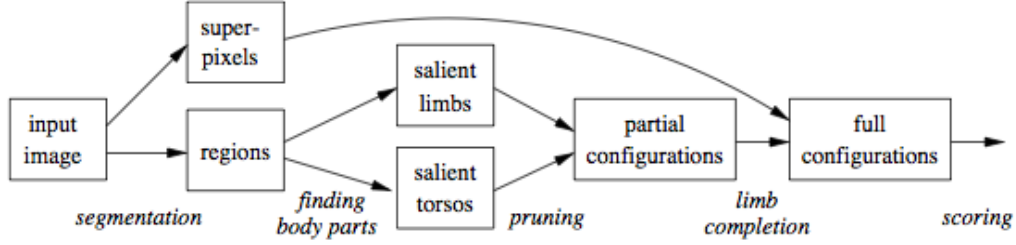


Figure 2.18: Schematic diagram for human body localisation method used by [Mori et al. \(2004\)](#).

their work, they employ a data driven belief propagation Monte Carlo algorithm, utilising importance sampling functions built from bottom-up visual cues for efficient probabilistic inference.

[Navaratnam et al. \(2005\)](#) propose an efficient method of human layout prediction and estimate the human body pose by learning a hierarchical parts-based model. They search for configurations by finding the most reliably detectable part. The rest of the parts are searched based on the detected location of this anchor as they all are kinematically linked. The parts are represented by a set of 2D templates created from a 3D model, hence inherently encoding the 3D joint angles. They use hidden Markov model (HMM) to model this and exploit the temporal coherence of the body motion using a modified Viterbi algorithm ([Viterbi \(1967\)](#)).

Typically when the model is fitted over observable data, appearance likelihood over parts labels is maximised. Alternatively the model could be learned such that when it is run on the training images, it should find the parts at the right location. [Ramanan and Sminchisescu \(2006\)](#) do this by maximizing the conditional likelihood of the training data. While learning, they initialise model parameters with their maximum likelihood estimates, then use gradient ascent to reach the global optimum. Their learning method searches exhaustively over all part locations in

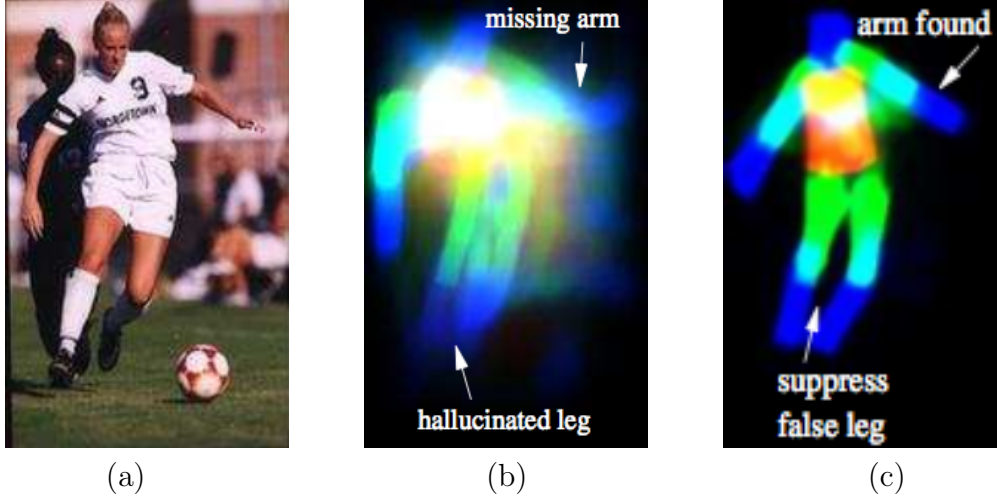


Figure 2.19: Sample results for iterative learning algorithm of [Ramanan \(2006\)](#). (a) Input image. (b) Initial posterior from edges (with missing arm and extra hallucinated leg). (c) Final result after re-parsing.

an image and thus avoids over-fitting issues known with conditional random fields (CRFs).

[Ramanan \(2006\)](#) uses iterative parsing process to learn better features tuned to a particular image. The initial pose estimate of human body is obtained using an edge-based deformable model. A region model for each part is then built around the initial detection to learn foreground and background colour histograms. Using these histograms and initial detector, the images are then re-parsed. It is observed that using this technique, localisation of the body parts improves after every iteration (Figure 2.19). Also, this method learns a colour model from the test image, thus, uses stronger local cues compared to all other existing methods.

The major disadvantage of all the methods discussed so far is that they search over the whole image to localise body parts. If an exhaustive search has to be performed then it might require searching over millions of positions in the image. This may result in high computational and time complexity. In order to avoid

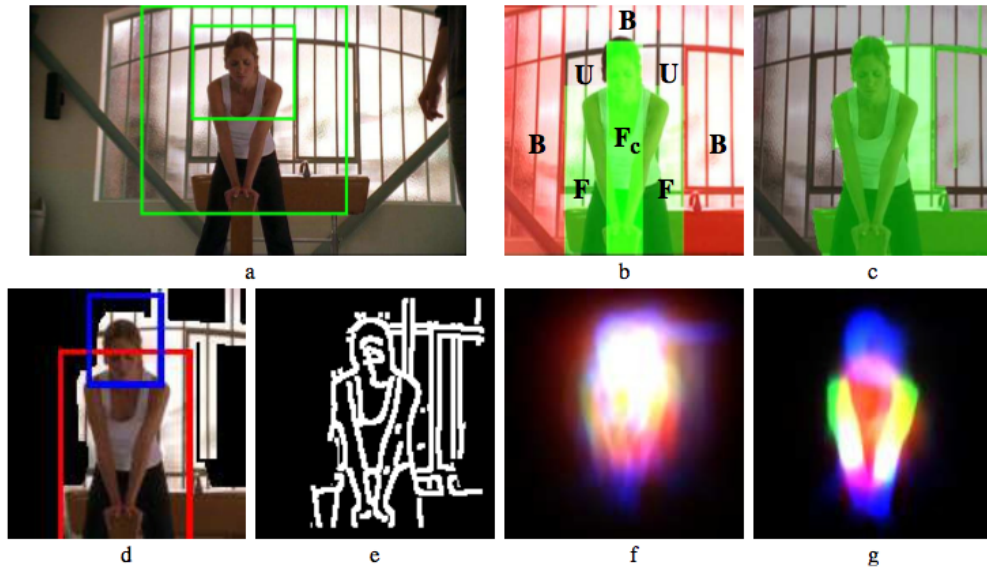


Figure 2.20: Overview of technique in search space reduction method of [Ferrari et al. \(2008\)](#). (a) Upper body detection. (b) Sub-regions for initializing Grab cut. (c) Foreground region output by Grab cut. (d) Area to be parsed. (e) Edges within the area. (f) Posterior of the part positions after the edge-based inference. (g) Posterior after the second inference, based on edges and appearance.

this, [Ferrari et al. \(2008\)](#) use an efficient but weak detector in the initial stage to reduce the search space. Then they use a strong detector with spatio-temporal modelling to get the exact localisation (Figure 2.20). In [Eichner and Ferrari \(2009\)](#), this work is extended further by learning better appearance models. This is done by learning a latent relationship between the appearance of different body parts from the annotated images. The stable body parts are detected first and then the appearance model of other body parts is improved by using an appearance transfer mechanism.

In all the methods discussed so far, either weak detectors (using the edge or colour information etc.) combined with a generative body model were used ([Hua et al. \(2005\)](#); [Ramanan \(2006\)](#); [Ramanan and Sminchisescu \(2006\)](#)), or a strong part detectors (SVM for instance) combined with a discriminative body model were employed ([Mohan et al. \(2001\)](#); [Ronfard et al. \(2002\)](#)). However, in [Andriluka et al. \(2009\)](#), the authors show that if strong part detectors are combined with a generative body model, better results could be achieved. Furthermore, their method does not require iterative parsing of the test image. They use shape context descriptors ([Belongie et al. \(2000\)](#)) with Ada-Boost to train the discriminative part classifiers. They combine this discriminative appearance model with a generative pictorial structures approach and obtain a generic model for people detection and pose estimation.

[Kumar et al. \(2009\)](#) use a similar technique and learn strong part detectors using SVM. They train a non-parametric generative model of the human body to perform human layout estimation. The supervised learning problem is formulated as an optimisation problem, which is then reduced to an equivalent convex formulation. This new convex problem has a number of constraints polynomial in the number of possible parts locations and therefore it is feasible to solve. A globally optimal solu-

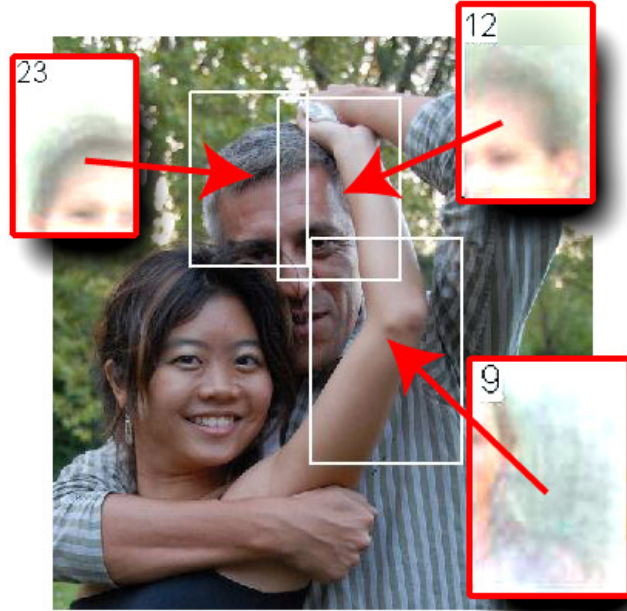


Figure 2.21: Poselets proposed in [Bourdev and Malik \(2009\)](#). They are visually coherent image patches which are also tightly clustered in 3D configuration space.

tion for this problem is obtained using a dual decomposition strategy. We improve this system further to make it work better and faster for efficient human layout estimation as explained in [Section 4.2](#).

In [Bourdev and Malik \(2009\)](#); [Bourdev et al. \(2010\)](#), authors represent the human body as an ensemble of visibly coherent image patches which are also tightly clustered in 3D configuration space ([Figure 2.21](#)). These patches are called as ‘poselets’ and they describe a particular part of the human pose under a given viewpoint. This results in a representation which is easy to detect (because of visual coherence) and also helpful in localisation (due to tight clustering in configuration space). A separate classifier is learned for each of the poselets, which is used at test time to detect these poselets. The results of different poselet detections are combined which give an indication about the configuration of the human body. Since there is no

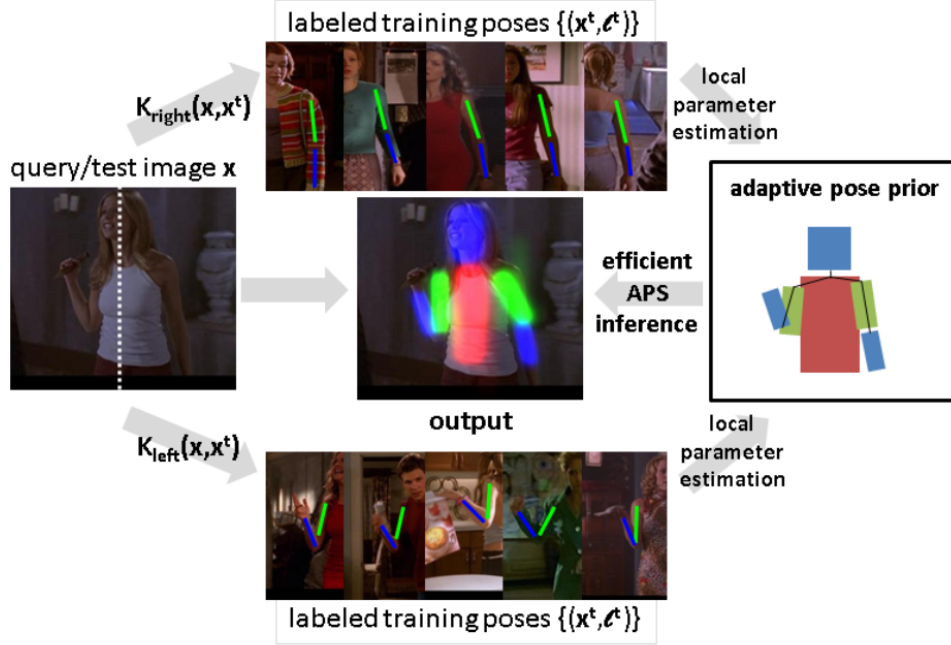


Figure 2.22: Overview of the system of Sapp et al. (2010a). For each test example, pictorial structure parameters are estimated as a kernel-weighted sum of training examples, based on their similarity to the test image.

mapping defined between different body parts and poselets, this method can not be used for body parts localisation.

A pictorial structure model learns different pose parameters from the training data which are then used for inference on the test image. A drawback with this approach is that it results in poor generalisation to the infrequent poses. In Sapp et al. (2010a), an adaptive pictorial structure model is proposed where certain pose parameters are estimated locally for every test example. They are computed as the kernel weighted sum of the training examples depending on their similarity to the test image. Figure 2.22 describes their method. In Sapp et al. (2010b), a faster version of adaptive pictorial structure method is proposed by using a coarse-to-fine cascade of appearance models. This technique is extended in Sapp et al. (2011) to

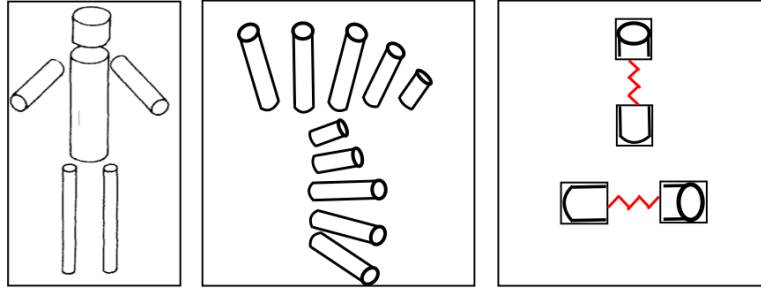


Figure 2.23: Human body model of [Yang and Ramanan \(2011\)](#). Left: A human body is modelled as a deformable parts-based model. Middle: Each part is modelled as a mixture of template denoting different orientations and foreshortening states. Right: Different deformations possible between the mixture of parts.

videos by performing joint parsing of multiple articulated parts over time.

Most of the human layout techniques discussed so far are capable of estimating the human body layout only in frontal poses. [Johnson and Everingham \(2010\)](#) overcome this problem with mixture of trees model. They partition the space of human pose into mixture components and for each such component, they train a separate non-linear appearance model and a Gaussian pose prior. This allows them to cope with the large variation in parts appearance and pose configurations. In [Johnson and Everingham \(2011\)](#), they extend this method to handle much larger quantity of noisy training data. [Yang and Ramanan \(2011\)](#) perform human layout estimation by extending the deformable parts-based model of [Felzenszwalb et al. \(2010b\)](#) (Figure 2.23). They model the human body as a mixture model where each mixture component is a pictorial structure for a specific pose (frontal, lateral etc.). Each part is also modelled as a mixture of templates representing limbs in different orientation and foreshortening states (Figure 2.23 middle). The pairwise relations are defined such that it results in a tree-structured model which is efficiently trained with dynamic programming. However, being a pictorial structure model, Yang and

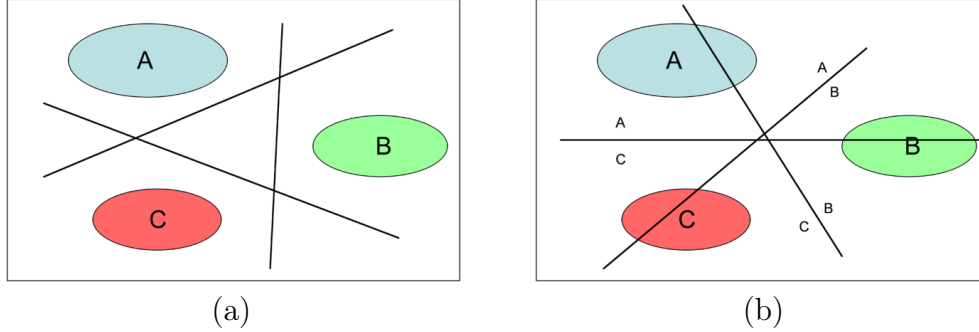


Figure 2.24: (a) One vs. all classifier learns a different classifier separating each class from the remaining. Thus, the number of classifiers learnt in this case is c , the number of class labels. (b) One vs. one classifier learns a classifier between every pair of classes. The number of classifiers learned in this setting is $\frac{c(c-1)}{2}$.

Ramanan’s method suffers from all the limitations of a pictorial structure discussed in Section 2.3. Furthermore, its performance is not good for poses other than near frontal poses. In Chapter 7, we explain our method of human layout estimation, which overcomes all the limitations of current estimation systems and gives good results for images captured from different viewpoints.

2.5 Taxonomic multiclass prediction

Multiclass classification involves learning a system capable of predicting class labels for images, when the number of classes is more than two. This problem has traditionally been solved using neural networks (Bishop (2006)), k-nearest neighbours (Bay (1998)), naïve Bayes classifiers (Rish (2001)), decision trees (Breiman et al. (1984)) and SVM (Section 2.6). Using SVM, this problem is often handled by combining multiple binary classifiers. Most commonly used methods are one-vs-all (Vapnik (1995)) or one-vs-one (voting) classifiers (Kressel (1999)). However, they result in a complex system which does not scale well with the number of classes. Figure 2.24

outlines the difference between both the methods. Another potential drawback is that the class structure is treated as ‘flat’ and the relationships between the classes are not considered, which are commonly expressed as hierarchies or taxonomies.

Taxonomic multiclass prediction involves retrieval of images as per a defined taxonomy or class hierarchy. This results in a multiclass classification system where images are grouped as per the defined taxonomic structure (class hierarchies). Taxonomies are very helpful in supporting tasks like searching or visualisation of object classes and also contributes in better semantic understanding of images. Another advantage associated with taxonomies is that they encourage an efficient information sharing among classes which benefits the overall classification. This helps when the number of classes is very high and the training examples are fewer for individual classes. For example, [Zweig and Weinshall \(2007\)](#) exploited class hierarchies to combine models from different category levels. [He and Zemel \(2008\)](#) used them to cope with missing and roughly specified annotations.

[Deng et al. \(2010\)](#) showed that classification based on hierarchical cost can be significantly more informative. The potential loss of valuable information suffered from ignoring class hierarchies has been pointed out repeatedly in the machine learning literature ([Koller et al. \(1997\)](#); [McCallum et al. \(1998\)](#)).

2.5.1 Building taxonomies

Various systems have been proposed that exploit the advantages associated with using taxonomies. Some of the systems used a pre-defined taxonomic structure ([Binder et al. \(2011\)](#); [Marszalek and Schmid \(2007\)](#)), others tried to build the hierarchical structure from the training images. Such systems can be divided into two groups: (i) Top-down taxonomy construction, (ii) Bottom up taxonomy construction.

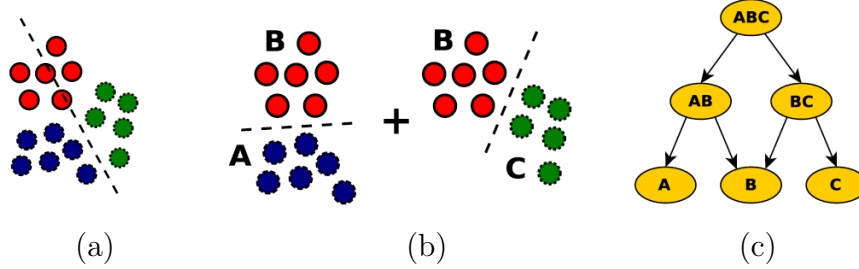


Figure 2.25: (a) A linear decision boundary cannot be learned to separate class ‘B’ from ‘A’ and ‘C’. (b) The overlapping class set solution of Marszalek and Schmid (2008) for this example. Both the classifiers share the label ‘B’ and learns a separation boundary between class ‘B’ and, ‘A’ and ‘C’ respectively. (c) The taxonomy build using this method.

In the top-down approach, a taxonomy is built by recursive partitioning of the set of classes. Chen et al. (2004) perform this by using a relaxed max-cut formulation. Liu et al. (2005) employ k-means clustering over the class labels for the same task. Marszalek and Schmid (2008) identify that the classes are not always separable. They therefore learn the class hierarchy as overlapping class sets (Figure 2.25). Tibshirani and Hastie (2007) learn a greedy decision tree for the taxonomy structure in a top-down manner.

For bottom-up taxonomy construction, different class labels are grouped together depending upon their similarity. For instance, Zhigang et al. (2005) perform this using agglomerative clustering over class labels. Hierarchies are also found by random grouping followed by cross-validation by Yuan et al. (2006). Fan (2005); Griffin and Perona (2008) use spectral clustering to build the hierarchical structure.

Other than these, taxonomies are also built using dependency graph (Lampert and Blaschko (2008)) and co-occurrence statistics (Blaschko and Gretton (2008)) of class labels in a multiple kernel learning setting.

2.5.2 Learning using taxonomies

When a taxonomy is available, a standard way of using it is sequential greedy decision starting from the root node (Marszalek and Schmid (2007)). The most probable child at each node is selected until reaching a leaf node. Thus, for classifying an unseen image only the classifiers on one path of the taxonomy are evaluated.

A potential disadvantage of the above greedy process is that it ignores all other possible paths and decides the downwards path in the taxonomy depending on the decision of the local node classifier. This is improved by using structured output prediction (Section 2.6.2) as proposed by Binder et al. (2011), who report better results than all other techniques. In Section 7.2 we explain our method of taxonomic multiclass prediction using structured output ranking.

2.6 Large margin learning

Support vector machines (SVM) (Cortes and Vapnik (1995); Vapnik (1995); Burges (1998); Scholkopf and Smola (2002)) are the most commonly used large margin classifiers. They make a binary prediction (e.g., true or false) for previously unseen data items. SVM training involves learning the parameters of a hyperplane (w) in a feature space of data-points which separates the two classes.

Let the training set consists of n data-points as instance-label pairs (x_i, y_i) where each instance x_i is a d -dimensional real-value vector ($x_i \in \mathbb{R}^d$) and $y_i \in \{-1, 1\}$. The formulation of SVM is given by:

$$\min_{w, \xi, b} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \right\} \quad (2.2)$$

$$\text{s.t., } y_i(w^T x_i + b) \geq 1 - \xi_i \quad (2.3)$$

$$\xi_i \geq 0, \quad \forall i \quad (2.4)$$

where b is the offset and ξ_i is the hinge loss penalty (slack variable).

The first term $(\frac{1}{2}\|w\|^2)$ of the objective function encourages a large margin between the training examples. The second term ensures that training instances should lie on the correct side of the hyperplane. C is a weighting parameter corresponding to assigning higher penalty to errors. The overall optimisation problem is convex and therefore a globally optimal solution can be found.

The above formulation of SVM is the primal form. The Wolfe dual formulation of SVM is given by:

$$\max_{\alpha} \left\{ \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \right\} \quad (2.5)$$

$$\text{s.t.}, \quad \sum_i \alpha_i y_i = 0 \quad (2.6)$$

$$0 \leq \alpha_i \leq C \quad (2.7)$$

where α_i is the Lagrange multiplier for each constraint in Equation (2.3).

The solution is given by $w = \sum_i \alpha_i y_i x_i$. In the solution, data-points for which $\alpha_i > 0$ are known as ‘support vectors’. These support vectors are the critical elements of the training set. If all other training points are removed and training is repeated, the same hyperplane (w) is obtained as the solution. In Equation (2.5) training data appears as the dot-product of data-points, which further facilitates the application of the kernel trick.

2.6.1 Large-scale training of SVM

Various algorithms have been proposed for efficient large-scale training of SVM. In SVM-light (Joachims), first the original problem is decomposed to a smaller sub-problem by selecting the appropriate working set (of q alpha values) based on the Zoutendijk's method (Joachims (1999)). The method tries to find a direction which is the steepest descent direction with q non-zero elements. The working set selection is done by restricting q to be an even number, and selecting each $q/2$ largest elements (in terms of α values) from the positive and negative training points. After finding the appropriate working set, only the α values for the working set are updated while the rest remain fixed in the current iteration. This smaller sub-problem is solved with a non-linear interior-point solver (Vanderbei (1994)).

Sequential minimal optimisation (SMO) is another technique which solves the large SVM problem into smaller sets (Platt (1999)). To do so, SMO solves sequentially on only two α values. This system has two parts: The first component is an analytic method to solve for the sub-problem of two Lagrange multipliers. The second component is a heuristic for picking which two multipliers to optimise. The analytical method that solves for two Lagrange multipliers can be described as a hill climbing algorithm where the multiplier values are updated iteratively to satisfy the Karush-Kuhn-Tucker (KKT) conditions (Borges (1998)). SMO selects the first multiplier to optimise as the one that violates the KKT conditions. The other multiplier is then picked based on the maximal improvement for the value of the objective function. LibSVM (Chang and Lin (2001)) and SVM-torch (Collobert and Bengio (2001)) use a variant of SMO. This method is also used by some online versions of SVM (Bordes et al. (2005)).

Recent algorithms use a *cutting plane method* to optimise the SVM objective

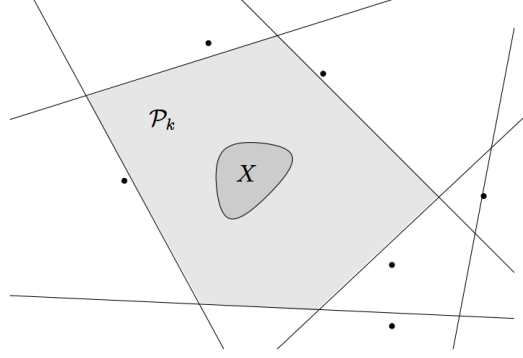


Figure 2.26: **Cutting plane method.** Data-points are shown as dots, and the associated cutting-planes shown as lines. From these cutting-planes, the target set X (shown dark) is restricted inside the localisation polyhedron $\mathcal{P}_k$, shown as lightly shaded. The search for X is limited to $\mathcal{P}_k$. Figure is taken from [Boyd and Vandenberghe \(2007\)](#).

Algorithm 1 Cutting plane algorithm for SVM classifier.

- 1: Input: $(x_1, y_1), \dots, (x_n, y_n), C, \epsilon$
 - 2: $S_i \leftarrow \emptyset$
 - 3: **repeat**
 - 4: **for** $i = 1, \dots, n$ **do**
 - 5: compute $(\hat{x}, \hat{y}) = \arg \max (1 - y_i(w^T x_i + b))$
 - 6: compute $\xi_i = \max\{0, \max_{(x,y) \in S_i} (1 - y_i(w^T x_i + b))\}$
 - 7: **if** $(1 - \hat{y}(w^T \hat{x} + b)) > \xi_i + \epsilon$ **then**
 - 8: $S_i \leftarrow S_i \cup (\hat{x}, \hat{y})$
 - 9: $\alpha_S \leftarrow \text{optimize dual over } S, S = \cup_i S_i$
 - 10: **end if**
 - 11: **end for**
 - 12: **until** no S_i has changed during iteration
-

(Tsochantaridis et al. (2004)). In this method, the objective is optimised by successive addition of constraints. Each of the added constraint corresponds to a cutting plane which cuts off the current solution from the feasible set (Figure 2.26). Algorithm 1 presents a cutting plane method for SVM classification. The algorithm proceeds by finding the ‘most violated’ constraint among all the data-points. If margin violation exceeds the current value of ξ_i by more than some pre-defined small value ϵ , the data-point corresponding to this constraint is added to the working-set and the process is repeated. The SVM optimisation problems differ only in one constraint from iteration to iteration. The optimisation problems are solved using any standard method such as interior-point solver. Most of the time, the number of constraints is not very high, which makes the training process very quick.

2.6.2 Structured output SVM

This variant of SVM facilitates the learning of an output space with a complex internal structure (Lafferty et al. (2001); Taskar et al. (2003); Tsochantaridis et al. (2004)). The problem dealt here is of learning a function with complex outputs, where the prediction is not a single univariate response (e.g., 0/1 for classification or a real number for regression), but a complex multivariate object. For example, the desired prediction is a sequence, tree, lattice or a graph.

This is solved by learning a discriminative function $\mathcal{F} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ over input-output pairs (x_i, y_i) from which a prediction is derived by maximizing $\mathcal{F}$ over the response variable (y_i) for a given input x_i . To do so, a combined feature representation $\phi(x, y)$ is used and the training problem is modified as learning a classifying hyperplane for the joint kernel map.

$$\forall i: \langle w, \phi(x_i, y_i) \rangle - \max_{y \in \mathcal{Y} \setminus y_i} \{ \langle w, \phi(x_i, y) \rangle \} \geq 1 - \xi_i \quad (2.8)$$

which means the correct prediction is separated from the highest scoring wrong prediction by a margin of width 1.

For computer vision problems, $x_i \in \mathbb{R}^d$ is the i^{th} image represented as a d -dimensional feature vector, and $y_i \in \mathcal{Y}$ is the desired structured output space. The domain of $\mathcal{Y}$ is application specific. For person layout problem of the PASCAL VOC challenge (Everingham et al. (2010a)) $\mathcal{Y} \equiv \mathbb{R}^{4r}$, where r is the number of body parts each represented by four co-ordinates of its bounding box. For multiclass prediction, $\mathcal{Y} \equiv \{1, \dots, c\}$, where c is the number of class labels.

In the above constraint (Equation (2.8)), all the data-points are separated from the margin by the same distance (1 in this case). Intuitively, data-points which are more similar to the correct prediction should be separated with smaller margin compared to the data-points which are less similar. This can be accomplished either by multiplying the margin with the difference value ($\Delta(y, y_i)$) or by scaling the slack variable with the inverse of the difference value. This results in the following two different formulations of structured SVM:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \right\} \quad (2.9)$$

$$\text{s.t., } \langle w, \phi(x_i, y_i) \rangle - \max_{y \in \mathcal{Y} \setminus y_i} \{ \langle w, \phi(x_i, y) \rangle \} \geq \Delta(y, y_i) - \xi_i \quad (2.10)$$

$$\text{or, } \langle w, \phi(x_i, y_i) \rangle - \max_{y \in \mathcal{Y} \setminus y_i} \{ \langle w, \phi(x_i, y) \rangle \} \geq 1 - \frac{\xi_i}{\Delta(y, y_i)} \quad (2.11)$$

$$\xi_i \geq 0, \quad \forall i \quad (2.12)$$

Equation (2.10) is known as the margin rescaled formulation and Equation (2.11) is known as the slack rescaled formulation of the structured SVM.

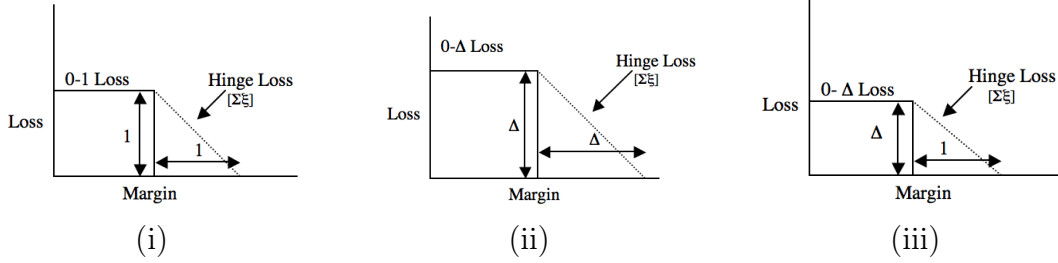


Figure 2.27: (i) Loss function for normal SVM formulation. (ii) Loss function for margin rescaled formulation. (iii) Loss function for slack rescaled formulation.

The margin rescaled formulation has a potential disadvantage that for certain output values which are far from the positive examples, the loss value (Δ) would dominate over the training error (ξ_i), which may result in a sub-optimal model learning. This problem is avoided in the slack rescaling as the width of the margin is always maintained as 1. Figures 2.27 shows the difference between normal SVM loss function and margin and slack rescaled formulations respectively.

The training of structured SVM is performed using the standard cutting-plane technique described in Section 2.6.1.

2.6.3 Ordinal regression SVM

In ordinal regression, the output y_i is a scalar value indicating the ordering (rank) of x_i . Thus for this problem $y_i \in \{1, \dots, R\}$, so that the values $1, \dots, R$ are related on an ordinal scale. R is the number of rank values. The goal is to learn a ranking function $h(x) = w^T x$, such that higher ranked pairs are assigned higher score, i.e.,

$$h(x_i) > h(x_j) \iff y_i > y_j$$

In this case, output space $\mathcal{Y}$ is a non-metric space, therefore, it can not be learned by a regression estimation. Herbrich et al. (2000) suggest to exploit the

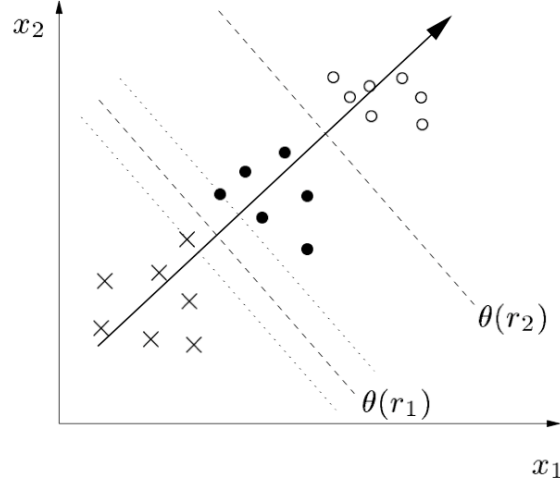


Figure 2.28: **Ordinal regression SVM.** Data-points are of rank r_1 ($\times$), rank r_2 ($\bullet$), and rank r_3 ($\circ$) and the axis of $h(x)$ is shown by the arrow, where $x = (x_1, x_2)^T$. $\theta(r_1)$ and $\theta(r_2)$ are the two coupled thresholds for the data-points which mimic the separating hyperplanes. Figure is taken from [Herbrich et al. \(2000\)](#).

ordinal nature of the elements of $\mathcal{Y}$ by considering the order on the space $\mathcal{X}$ induced by each mapping $f : \mathcal{X} \mapsto \mathcal{Y}$. They formulate it as a max-margin problem given by:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + \frac{C}{m} \sum_{(i,j) \in \mathcal{Q}} \xi_{ij} \right\} \quad (2.13)$$

$$\text{s.t.}, \quad w^T(x_i) \geq w^T(x_j) + 1 - \xi_{ij} \quad (2.14)$$

$$\xi_{ij} \geq 0, \quad \forall (i, j) \in \mathcal{Q} \quad (2.15)$$

where $\mathcal{Q}$ is the set of pairs (i, j) for which example i has a higher rank than example j , i.e., $\mathcal{Q} = \{(i, j) : y_i > y_j\}$ and $m = |\mathcal{Q}|$.

Thus, the above optimisation problem (also known as ranking SVM) finds a large-margin linear function $h(x)$ that minimises the number of pairs of training examples that are swapped with respect to the desired order (Figure 2.28).

The above formulation is a convex quadratic problem and can be trained using any conventional quadratic solver (Tsochantaridis et al. (2004)) or using the cutting plane technique discussed in Section 2.6.1. The only problem is that the above formulation has $\mathcal{O}(n^2)$ constraints and slack variables. The training time complexity is also $\mathcal{O}(n^2)$ using the standard cutting plane method. Even a small dataset of thousand examples is infeasible to train using this method.

2.6.4 1-slack formulation of SVM

A novel formulation for SVM is proposed by Joachims (2006), where for each SVM problem a single slack variable is defined, hence the name 1-slack formulation. The key idea is to replace the n cutting-plane model of the hinge loss - one for each training example - with a single cutting plane model for the sum of the hinge-losses, i.e., $\xi = \frac{1}{n} \sum_{i=1}^n \xi_i$.

1-slack formulation has a solution that is extremely sparse, with the number of non-zero dual variables independent of the number of training examples. The run-time complexity of many algorithms could be reduced using 1-slack formulations (Joachims (2006); Joachims et al. (2009)). Another benefit of using 1-slack formulation is the direct correspondence between ξ and the infeasibility of the set of constraints. The approximate accuracy of the solution is directly related to the training loss, which provides an intuitive precision criterion.

SVM classification. The new optimisation problem for SVM classification for 1-slack formulation is given by:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C\xi \right\} \quad (2.16)$$

$$\text{s.t., } \frac{1}{n} \cdot w^T \sum_{i=1}^n c_i y_i x_i \geq \frac{1}{n} \sum_{i=1}^n c_i - \xi, \quad \xi \geq 0 \quad (2.17)$$

where $c_i \in \{0, 1\}$ is an indicator variable for the active constraints. A constraint is active (i.e., violated) then c_i is set to 1. It could be noted that the above formulation has only 1 slack variable and 2^n constraints. Each constraint in this new formulation corresponds to the sum of subset of constraints from the original n -slack formulation. In [Joachims \(2006\)](#), it is proved that 1-slack and n -slack formulations are equivalent.

For a given w , the ξ_i in n -slack formulation (2.2) can be optimised individually and the optimal solution is: $\xi_i = \max\{0, 1 - y_i(w^T x_i)\}$. For simplicity, the offset b is assumed to be zero.

For 1-slack formulation (2.16), the optimal ξ for a given w is:

$$\xi = \max_{c \in \{0,1\}^n} \left\{ \frac{1}{n} \sum_{i=1}^n c_i - \frac{1}{n} \sum_{i=1}^n c_i y_i (w^T x_i) \right\} \quad (2.18)$$

Each c_i can be optimised independently which gives:

$$\begin{aligned} \min C\xi &= \sum_{i=1}^n \max_{c_i \in \{0,1\}} \left\{ \frac{1}{n} c_i - \frac{1}{n} c_i y_i (w^T x_i) \right\} \\ &= \sum_{i=1}^n \max\left\{0, \frac{1}{n} - \frac{1}{n} y_i (w^T x_i)\right\} = \min \frac{C}{n} \sum_{i=1}^n \xi_i \end{aligned} \quad (2.19)$$

Thus, both the objective functions (2.16) and (2.2) are equal for any w for optimal ξ and ξ_i and hence, 1-slack and n -slack formulations are equivalent. Similar proofs of equivalence for other SVM formulations are given in [Joachims \(2006\)](#).

Training. Algorithm 2 gives the cutting plane solution of the above 1-slack formulation. The major difference between Algorithm 2 and Algorithm 1 is that in each iteration a sum of all active (i.e., violated) constraints of Equation (2.3) is computed,

which forms the most violated constraint as per Equation (2.17) optimised at every iteration. It is proved in [Joachims \(2006\)](#) that the algorithm always terminates after a polynomial number of iterations that does not depend on the size n of the training set.

Algorithm 2 Cutting plane algorithm for 1-slack formulation of SVM classifier.

```

1: Input:  $(x_1, y_1), \dots, (x_n, y_n), C, \epsilon$ 
2:  $S \leftarrow \emptyset$ 
3: repeat
4:    $(w, \xi) \leftarrow \arg \min_{w, \xi \geq 0} \{ \frac{1}{2} \|w\|^2 + C\xi \}$ 
5:   s.t.,  $\forall c \in S : \frac{1}{n} w^T \sum_{i=1}^n c_i y_i x_i \geq \frac{1}{n} \sum_{i=1}^n c_i - \xi$ 
6:   for  $i = 1, \dots, n$  do
7:      $c_i \leftarrow \begin{cases} 1 & y_i(w^T x_i) < 1 \\ 0 & \text{otherwise} \end{cases}$ 
8:   end for
9:    $S \leftarrow S \cup \{c\}$ 
10: until  $\frac{1}{n} \sum_{i=1}^n c_i - \frac{1}{n} \sum_{i=1}^n c_i y_i (w^T x_i) \leq \xi + \epsilon$ 

```

Structured output SVM. Structured output SVM is also formulated as 1-slack formulation in [Joachims et al. \(2009\)](#). The formulation is given as follows:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C\xi \right\} \quad (2.20)$$

$$\text{s.t., } \frac{1}{n} \cdot w^T \sum_{i=1}^n [\phi(x_i, y_i) - \phi(x_i, \bar{y}_i)] \geq \frac{1}{n} \sum_{i=1}^n \Delta(y_i, \bar{y}_i) - \xi, \quad \xi \geq 0 \quad (2.21)$$

$$\text{or, } \frac{1}{n} \cdot w^T \sum_{i=1}^n \Delta(y_i, \bar{y}_i) \cdot [\phi(x_i, y_i) - \phi(x_i, \bar{y}_i)] \geq \frac{1}{n} \sum_{i=1}^n \Delta(y_i, \bar{y}_i) - \xi, \quad \xi \geq 0 \quad (2.22)$$

Here Equation (2.21) represents the margin rescaled formulation and Equation (2.22) represents the slack rescaled formulation. Above optimisation problem has $|\mathcal{Y}|^n$ constraints, one for each possible combination of labels $(\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n) \in \mathcal{Y}^n$. This

can be trained using by adapting the cutting plane algorithm given in Algorithm 2 (Joachims et al. (2009)).

Ordinal regression. The 1-slack formulation for ordinal regression is given by:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C\xi \right\} \quad (2.23)$$

$$\text{s.t.}, \quad \frac{1}{m} \cdot w^T \sum_{(i,j) \in \mathcal{P}} c_{ij} (x_i - x_j) \geq \frac{1}{m} \sum_{(i,j) \in \mathcal{P}} c_{ij} - \xi, \quad \xi \geq 0 \quad (2.24)$$

where $\mathcal{P}$ is the set of pairs (i, j) for which example i has higher rank than example j , and $m = |\mathcal{P}|$.

Similar to the classification problem (2.16), the structural formulation has $\mathcal{O}(2^n)$ constraints, but only a single slack variable ξ . This formulation can be trained efficiently in $\mathcal{O}(n)$ by extending Algorithm 2 as mentioned in Joachims (2006). To do so, the ranking problem is modified into a series of binary classification problems, one for each of the rank values. The classification problem for a given rank value finds all the violated constraints corresponding to that rank value. Thus every iteration of the cutting plane algorithm for ordinal regression iterates additionally over all the rank values to find all the violated constraints. This results in an $\mathcal{O}(Rn) \approx \mathcal{O}(n)$ algorithm, where $R \ll n$ is the number of rank values.

2.6.5 Structured output ranking SVM

In structured output ranking, the goal is to learn a compatibility function $f(x_i, y_i) = w^T \phi(x_i, y_i)$, such that input-output pairs with lower loss (e.g., fewer mispredicted parts in the case of human layout) are assigned higher compatibility score. To do so, a structured output loss value Δ_i is defined for every input-output pair (x_i, y_i) , which represents the loss associated with the prediction y_i for the input x_i . The

compatibility function f is learned such that it holds,

$$f(x_i, y_i) > f(x_j, y_j) \iff \Delta_i < \Delta_j$$

Structured output ranking generalises ordinal regression to structured output space and does so by modifying the hinge loss pair for misordered pairs. The objective function is modified such that it pays a loss proportional to the difference in losses for ranking a worse prediction above the better one. The objective function of structured output ranking is given by:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i,j} \xi_{ij} \right\}, \quad \text{s.t.}, \quad \xi_{ij} \geq 0, \quad (2.25)$$

$$w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) \geq (\Delta_j - \Delta_i) - \xi_{ij} \quad (2.26)$$

$$\text{or, } w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) \geq 1 - \frac{\xi_{ij}}{\Delta_j - \Delta_i} \quad (2.27)$$

where $(i, j) \in \mathcal{P}$ denotes the ordered indices of training samples such that the structured output loss, Δ_i , of sample i is less than the loss, Δ_j , of sample j .

Equation (2.26) is the margin rescaled formulation and Equation (2.27) is the slack-rescaled formulation of structured output ranking SVM. Structured output ranking SVM has recently been used in computer vision for the task of object detection and localisation (Blaschko et al. (2010); Rahtu et al. (2011); Zhang et al. (2011)). The objective function can be solved using a standard cutting-plane technique described in Section 2.6.1. However, the number of constraints defined is $\mathcal{O}(n^2)$, which results in an $\mathcal{O}(n^2)$ solution.

Chapter 3

Datasets and Evaluation

This chapter explains various datasets that are used in evaluating the performance of different methods described in this thesis. We will also explain the evaluation criteria used to measure the performance for each dataset.

An outline of the chapter is given in the following table:

Dataset	Explained in Section	Used in Chapter
Skin dataset	3.1	4
Buffy and PASCAL stickmen	3.2	4
Hand dataset	3.3	5
Signer dataset	3.4	5
Indoor scene dataset	3.5	7
PASCAL VOC dataset	3.6	4, 5 and 7

In the above table, the first column gives name of the dataset, the middle column gives the Section where the dataset is explained and the last column lists the Chapters where the dataset is used.



Figure 3.1: Sample images from the skin dataset (top row) with their corresponding ground-truth skin masks (bottom row).

3.1 Skin dataset

The skin dataset is available from [Jones and Rehg \(2002\)](#) and was originally collected to develop a classifier of pornographic images. It has 13,634 images in total which are divided into 4,670 skin images and 8,964 non-skin images. Each skin image is also provided with a binary mask indicating which areas of the image contain skin, and this is used as the ground-truth data.

We filter out non-nude images, of which there are 452 with accurate skin masks, from the dataset and use them for our experiments. Some sample images from this dataset along with their skin masks are shown in [Figure 3.1](#).

Evaluation criteria. A system is evaluated for the correctness of the skin classification. This is measured using the F-measure which is computed as: $\frac{\text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$. Precision is defined as the fraction of retrieved instances that are correct, while recall is the fraction of correct instances that are retrieved. Thus, higher F-measure value means more ground-truth skin pixels have been identified correctly by the system. This criteria is also employed by other vision benchmarking systems for the



Figure 3.2: Sample images from Buffy stickmen dataset (top) and PASCAL stickmen dataset (bottom). The ground-truth stickmen annotations are overlaid on the images.

segmentation task ([Martin et al. \(2001\)](#)). The mean F-measure over all 452 images is reported as the overall performance of the system.

3.2 Buffy and PASCAL stickmen dataset

The Buffy stickmen and the PASCAL stickmen datasets by Ferrari *et al.* ([Buffy stickmen](#); [PASCAL stickmen](#)) are the most commonly used ones for testing human layout estimation. They consist of images of approximately frontal people and annotation data is provided for 6 body parts (head, torso, upper and lower arms) in the form of stickmen (shown in [Figure 3.2](#)).

The buffy stickmen dataset consists of frames extracted from various episodes of the TV series ‘Buffy the vampire slayer’. A total of 748 frames from episodes 2 to 6 of season 5 are 6-part stickmen annotated. The PASCAL stickmen dataset consists of images from the PASCAL VOC 2008 training and validation dataset ([Section 3.6](#)). In this, 549 images are available with 6-part stickmen annotation.

Evaluation criteria. Performance is measured in terms of the percentage of correctly estimated body parts (PCP) in human layout. An estimated body part is considered correct if its segment end-points lie within 50% of the length of the ground-truth annotated location. Furthermore, only those stickmen are evaluated that correctly localise the upper body, i.e., have an intersection-over-union score greater than 0.5 with the ground-truth bounding box for the upper body. The stickmen that fail to comply are not considered in the evaluation. This gives the detection rate of the model.

3.3 Hand dataset

We introduce a comprehensive dataset of hand images collected from various different public image dataset sources. While collecting the data, no restriction was imposed on the pose or visibility of people, nor was any constraint imposed on the environment. We perform the hand annotations on these collected images ourselves and use it for various hand detection experiments. The different image sources that we used to form this dataset are as follows:

INRIA person. This dataset consists of images of upright people collected from different sources. It was collected for research on detection of upright people in images and videos ([Dalal and Triggs \(2005\)](#)).

H3D dataset. H3D (Humans in 3D) dataset consists of images collected from internet which are annotated for joints and other keypoints (eyes, ears, nose, shoulders, elbows, wrists, hips, knees and ankles). This dataset was introduced by [Bourdev and Malik \(2009\)](#).

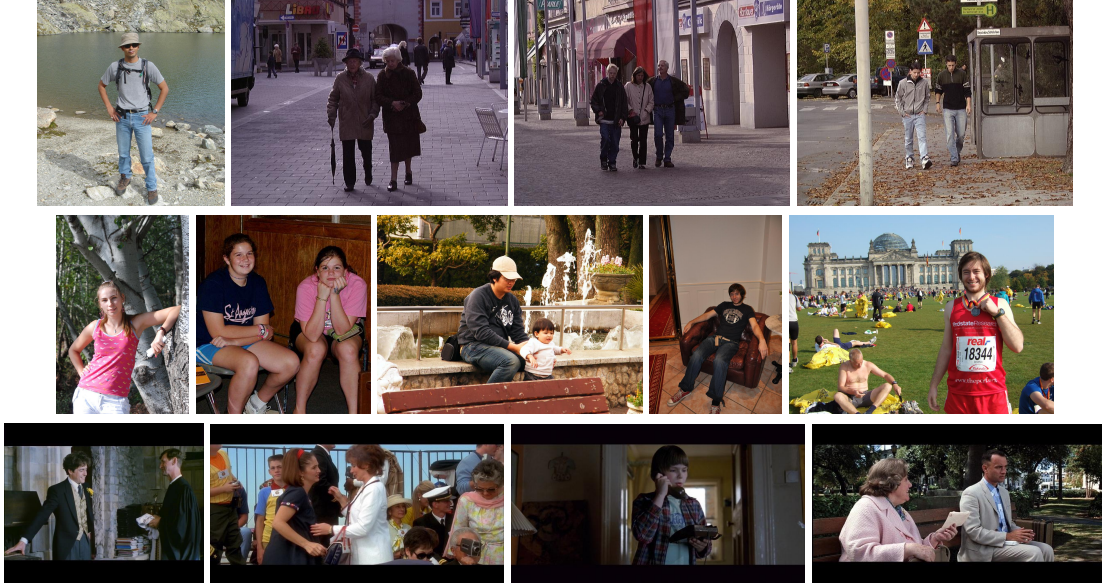


Figure 3.3: Sample images from INRIA person dataset (top row), H3D dataset (middle row) and Movie dataset (bottom row).

Movie dataset. This dataset consists of frames extracted from the films: ‘Four weddings and a funeral’, ‘Apollo 13’, ‘About a boy’ and ‘Forrest Gump’.

Images from the above stated datasets are shown in Figure 3.3.

Other than above mentioned datasets, we also use images from the **Skin dataset**, the **Buffy stickmen dataset** and the **PASCAL VOC dataset** which are described in Section 3.1, 3.2 and 3.6 respectively.

In each image, all the hands that can be perceived clearly by humans are annotated. The annotations consist of a bounding rectangle, which does not have to be axis aligned, oriented with respect to the wrist. Examples are shown in Figure 3.5.

The data is split into training, validation and test sets in such a way that there is no repetition of any given person among these datasets. Hand instances larger than a fixed area of bounding box (1,500 sq. pixels) are used for the experiments in Section 5.4. This gives around 4,170 high quality hand instances. The distribution

Training Dataset			Validation Dataset		
Source	# Ins	# Img	Source	# Ins	# Img
Buffy stickmen	438	346	Movie dataset*	649	406
INRIA person	137	97			
Poselet (H3D)	580	237	Test Dataset		
Skin dataset(Jones and Rehg (2002))	139	87	Source	# Ins	# Img
PASCAL VOC 2007 train and val set	507	345	PASCAL VOC 2010 human layout val set	98	63
PASCAL VOC 2010 train and val set (except human layout set)	1060	732	PASCAL VOC 2007 test set	562	373
<i>Total number</i>	2861	1844	<i>Total number</i>	660	436

Table 3.1: Distribution of larger hand instances in the hand dataset. ‘# Ins’ is the number of hand instances, and ‘# Img’ the number of images. The movie dataset contains frames from the films ‘Four weddings and a funeral’, ‘Apollo 13’, ‘About a boy’ and ‘Forrest Gump’.

of these images into training, validation and test sets is given in Table 3.1. A total of 13,050 hand instances are annotated (including the 4,170 larger instances). The distribution of sizes of hand bounding boxes in the dataset is shown in Figure 3.4. The dataset is available at <http://www.robots.ox.ac.uk/~vgg/data/hands/>.

Evaluation Measure. The performance is evaluated for the correctness of the detected hand bounding boxes. This is measured using average precision (AP) (the area under the precision-recall curve). A precision-recall curve is plotted by connecting the data-points given by the (recall, precision) values corresponding to the retrieved items ordered in terms of their retrieval confidence values. As in the PASCAL VOC challenge (Everingham et al. (2010a)), a hand detection is considered true or false according to its overlap with the ground-truth bounding box. A box is positive if the overlap score is more than 0.5, where the overlap score (O) between

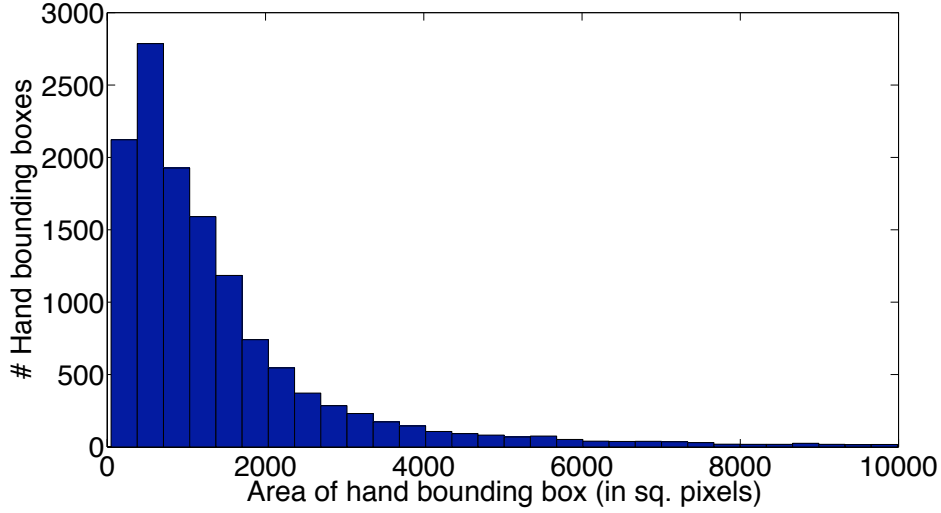


Figure 3.4: **Distribution of sizes of hand bounding boxes in the hand dataset.**

two boxes is defined as: $O = \frac{\text{area}(B_g \cap B_d)}{\text{area}(B_g \cup B_d)}$, where B_g is the axis aligned bounding rectangle around the ground-truth bounding box and B_d is the axis aligned rectangle around the detected bounding box.

3.4 Signer dataset

We use the ‘5-signers’ dataset in our experiment which is a collection of frames from five news sequences (39 frames each) with different signers (Buehler et al. (2008)). Each image is annotated with the bounding box of eight body parts (head, torso, upper and lower arms, hands) of the signer. Figure 3.6 shows some sample images from this dataset.

Evaluation Measure. We use this dataset to evaluate the performance of our hand detector and compare it with other competitive methods. A method is considered to be better if it detects correct hand instances with higher confidence values.

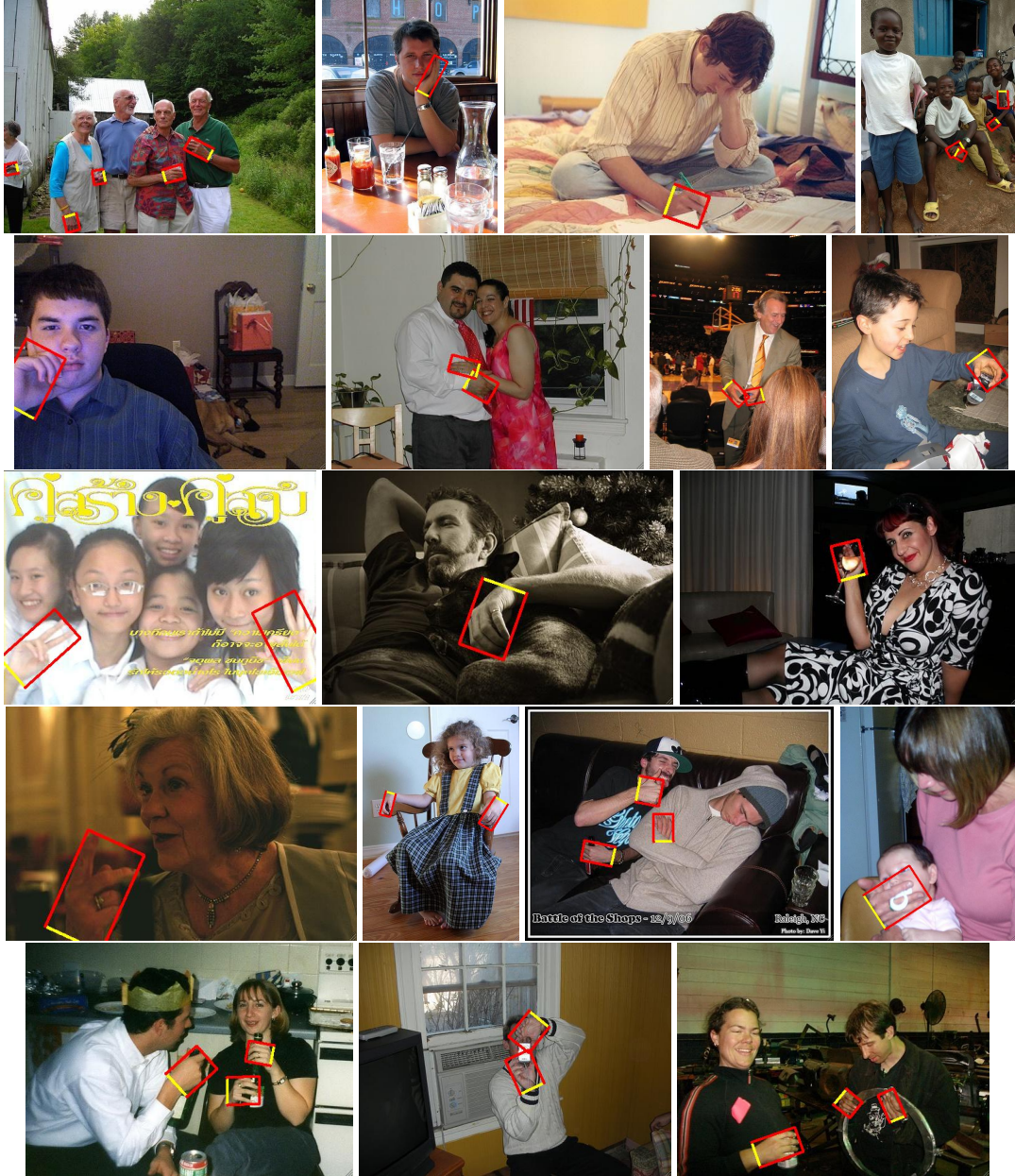


Figure 3.5: Sample images from the hand dataset with bounding box annotations overlaid. In the annotation, rectangle sides are ordered so that the wrist is along the first side shown in yellow.



Figure 3.6: Sample images from ‘5 signers’ dataset.



Figure 3.7: Sample images from different classes and their scene groups of the indoor scene database. Figure is obtained from [Quattoni and Torralba \(2009\)](#).

We adopt the evaluation criteria used by [Karlinsky et al. \(2010\)](#). The performance is computed as the percentage of correct hand detections within the top ‘k’ hand detections per ground-truth hand instance. A hand detection is considered correct if its overlap with the ground-truth bounding box is more than 0.5.

3.5 Indoor scene database

The indoor scene database ([Quattoni and Torralba \(2009\)](#)) consists of 15,620 images for 67 different indoor categories. The categories are further grouped into 5 scene groups which defines a two level taxonomy. The dataset is partitioned into the training set and the test set by choosing 80 and 20 images from each of the classes, respectively. Figure 3.7 shows example images with their class labels.

Evaluation Measure. The classification performance is evaluated in terms of the correctness of labelling of a given class. This is judged by the precision-recall curve. The principal quantitative measure used is the average precision (AP). The mean AP over all the classes is used as the performance measure of the system. This dataset is also used to measure the correctness of taxonomic multiclass predictions. This is done by measuring the mean taxonomic loss which is described in Section 7.2.

3.6 PASCAL VOC dataset

The PASCAL VOC challenge (Everingham et al. (2010a)) is an annual competition which includes various tasks for different problems of computer vision. For every task of the competition, a training and validation dataset is provided and the results are evaluated on the test set. The test set annotations for this competition is not publicly available (after 2007) and the results can be evaluated by submitting them to the evaluation server of the competition.

3.6.1 VOC 2007 classification dataset

The PASCAL VOC 2007 classification dataset (Everingham et al. (2007)) comprises 5,011 images in the combined training and validation set and 4,952 images in the test set. Each image may contain multiple objects which are annotated for 20 different object categories.

As in the indoor scene dataset, the mean average precision is used to measure the classification performance over this dataset. The taxonomic multiclass prediction is evaluated using the mean taxonomic loss described in Section 7.2. A taxonomy is defined with these 20 classes by the organisers (Figure 3.8) which we use for our

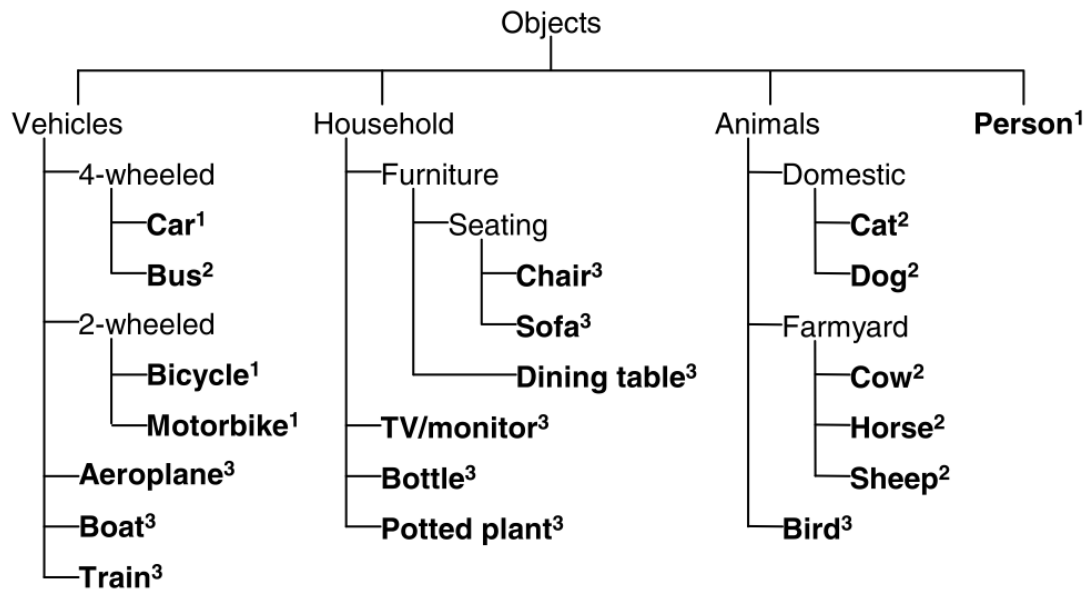


Figure 3.8: Taxonomy for the PASCAL VOC classes as defined in [Everingham et al. \(2010a\)](#). The super-script number indicates the depth of class label in the class hierarchy.

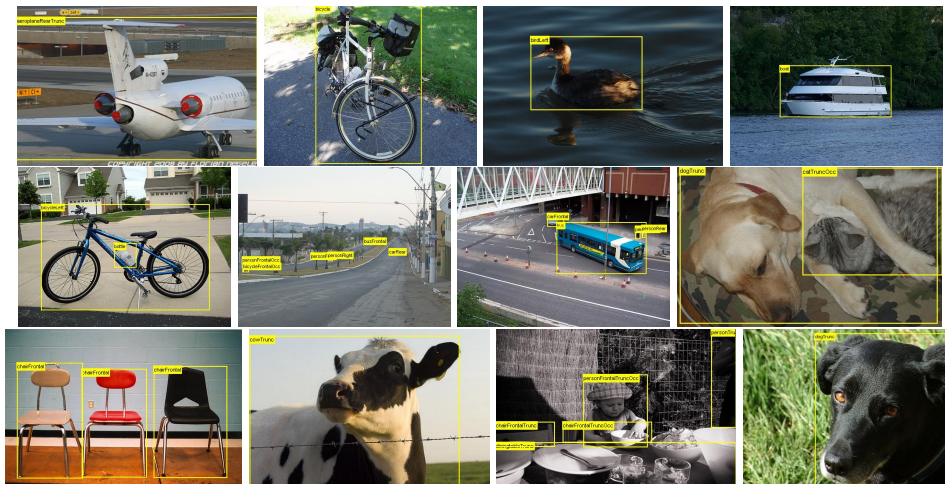


Figure 3.9: Sample images from the PASCAL VOC 2007 dataset with class and object annotations.

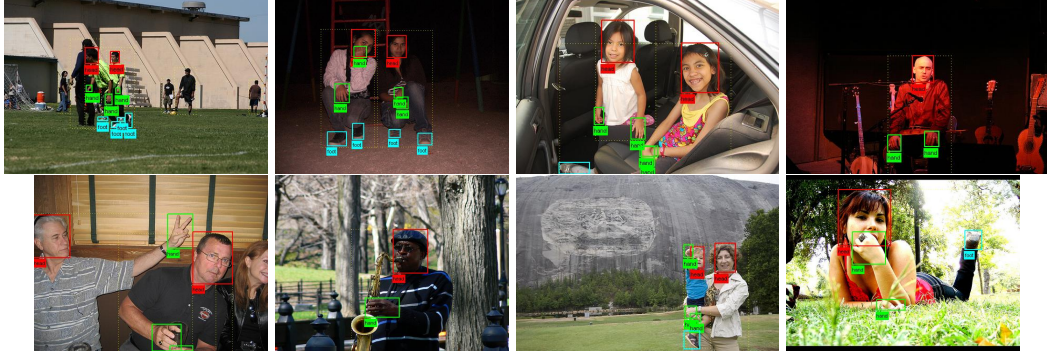


Figure 3.10: Sample images from the PASCAL VOC person layout dataset. The bounding box for the person is shown as a dotted rectangle and different ground-truth body part annotations are shown as solid rectangles.

experiments. Sample images from this dataset are shown in Figure 3.9.

3.6.2 VOC person layout dataset

The person layout competition of the PASCAL VOC measures correctness of different systems for the task of person layout estimation. For the competition, a bounding box (referred hereafter as human ROI) is provided for each ‘person’ object in a test image. The job is now to predict the presence or absence of parts (head, hands and feet), and the bounding boxes of those parts. Figure 3.10 shows some images from the dataset along with the ground-truth annotations of the body parts.

Evaluation criteria. The prediction for a person layout should be output with an associated real-valued confidence of the layout. This confidence score is then used to compute the precision-recall (PR) curve and AP (area under the PR curve) for each of the parts. The success of the layout prediction is judged by the correct prediction for the presence/absence of the parts and also by their correct localisation. A part

is considered to be correctly localised if the overlap score with the ground-truth location is more than 0.5.

We test our system with two different versions of the PASCAL VOC person layout dataset *viz.* 2010 and 2011. In the PASCAL VOC 2010 person layout dataset (Everingham et al. (2010b)), a total of 376 images having 556 human objects are available as the combined training and validation set. The PASCAL VOC 2011 person layout dataset (Everingham et al. (2011)) has 609 images in the combined training and validation set having 850 human ROIs.

Chapter 4

Human body parts localisation using skin detection

In this chapter, we will explain our method for efficient human body parts localisation using detected skin regions in the image. The inspiration behind this work is that some of the human body parts, such as hands and arms, are often found uncovered in images. If the skin regions are known in an image, then they can be used to hypothesise the position and size of these body parts. These hypotheses can be very cheaply obtained, and are later verified using more computationally expensive methods. This results in a very accurate detection system with relatively smaller time complexity.

This chapter is divided into three major parts. In Section 4.1, we will explain our method of skin detection. Our method works for different shades of skin and under illumination changes. We demonstrate the utility of our skin detector method for two very challenging tasks in computer vision: human layout (Section 4.2) and hand detection over real-world images (Section 4.3).

4.1 Skin detection

Skin detection is the process of finding skin coloured pixels in images and videos. A skin detector typically transforms image pixels into a suitable colour space and then uses a skin classifier to label them as skin or non-skin. A skin classifier is a decision function learned from a training database of skin coloured pixels.

For a long time, skin detection has been used as a test for the presence of humans in images. One of the early applications of skin detection was to identify nude pictures on the internet for content filtering (Fleck et al. (1996)). In another application, skin detection was used to detect faces in images (Hsu et al. (2002); Mottaleb and Elgammal (1999)). Skin detection has also been used to track the movement of body limbs and hands (Imagawa et al. (1998)). However, in all these systems, images are captured in controlled environments with man-made background objects and uniform lighting which results in poor performance over real-world images.

Skin detection is a challenging task as different shades of skin must be captured. The appearance of skin in an image also depends on illumination conditions. A good skin detector has to be robust to these variations. Humans are very good at identifying colours under a wide range of illuminations, this is called *colour consistency*. Vision systems still lack this attribute. An important challenge in skin detection is to represent skin colour in a way that either it is invariant to or less sensitive to illumination changes.

We develop our method of skin detection with two improvements: (i) Rather than learning a model for skin from training images in the manner of Jones and Rehg (2002), we use an instance of skin region to determine the colour of skin *in the target image* (Fritsch et al. (2002)). To do so, we detect faces in the target image,

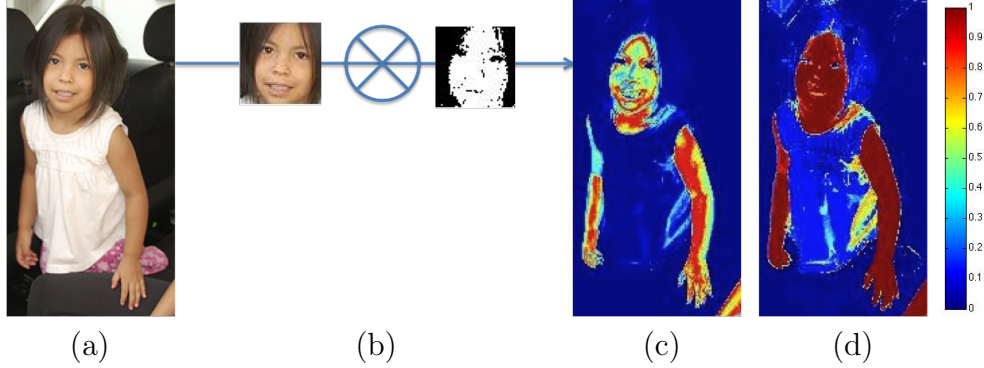


Figure 4.1: **Skin detection.** (a) Input image with the face highlighted. (b) A colour histogram is computed from the face skin pixels. (c) Likelihood values for skin pixels. (d) Likelihood values after bootstrapping. Note that many more of the true skin pixels are now detected compared to (c). In the figure, red colour corresponds to the highest value and blue the lowest.

then use the facial region to learn the local skin colour model. This enables skin to be detected despite unusual lighting and colour balances which may be peculiar to the target image. (ii) In a probabilistic skin detection system, every pixel is assigned a classification score which is then thresholded to a value to get a binary skin mask. We improve upon this by performing hysteresis thresholding ([Canny \(1986\)](#)), i.e. we learn two threshold values (low and high) for the classification score. Pixels which are above the high threshold are classified as skin. Then pixels which are above the low threshold are also classified as skin if they are spatial neighbours of a pixel above the high threshold. Thus, pixels are classified as skin depending on the neighbourhood consensus. Figure 4.1 shows our method of skin detection.

4.1.1 Implementation details

We detect the face region in the target image using the face detector from [Viola and Jones \(2001\)](#). A global skin detector ([Connaire et al. \(2007\)](#)) is then used to detect the skin pixels in the facial region. All the pixels in the facial region which

are identified as skin are used to create the colour histogram of skin pixels. All the pixels in the image are then assigned likelihood score using this histogram.

The likelihood score so obtained is then thresholded using hysteresis thresholding to get the binary mask of skin pixels. The two thresholds (confidences) for score are learnt using ground-truth segmentations available from [Jones and Rehg \(2002\)](#). Finally, a bootstrapping stage is applied to gain additional regions. In this, the colour of the neighbouring pixels is used to update the colour likelihood classifier and the process is repeated. We perform two iterations of this method. The complete method is summarised in Algorithm 3.

Pixels are classified using likelihood here, rather than a posterior or discriminative classifier, as learning a distribution for the background pixels at this stage is error prone due to the possible presence of arms and other people in unknown locations.

We use the HSV colour space for our experiments. HSV colour space represents colour in a cylindrical co-ordinate representation. The three channels stand for Hue (H), Saturation (S) and Value (V) space. It is designed to provide a more intuitive and perceptually relevant representation of the colour space.

For hysteresis thresholding, the first order neighbourhood (above, below, left and right) of pixels is considered. To create the colour histograms, saturation and value components are binned into 4 bins each. We experiment with different number of bins for the hue component and the optimum value we find is 24. This is because the hue component varies the most between different colours hence the number of bins needs more fine tuning.

Algorithm 3 Skin detection method.

```

1: Input: Image  $\mathcal{I}$ , higher threshold  $t_{upper}$ , lower threshold  $t_{lower}$ 
2: Detect face  $\mathcal{F}$  in  $\mathcal{I}$  using the face detector from Viola and Jones (2001)
3: Identify the skin pixels  $\mathcal{S}$  within  $\mathcal{F}$  using Connaire et al. (2007) detector
4:  $\mathcal{M} \leftarrow \mathcal{S}; n \leftarrow 2$ 
5: repeat
6:   Create a colour histogram  $\mathcal{H}$  using pixels in  $\mathcal{M}$ 
7:   Compute the likelihood  $\mathcal{P}$  for all pixels in  $\mathcal{I}$  as the bin value in  $\mathcal{H}$ 
8:   for  $i = 1, \dots, |\mathcal{I}|$  do
9:     if  $(\mathcal{P}_i > t_{upper}) \vee ((MaxNeighbourhood(\mathcal{P}, i) > t_{upper}) \wedge (\mathcal{P}_i > t_{lower}))$  then
10:       $\mathcal{M}_i \leftarrow 1$ 
11:     else
12:       $\mathcal{M}_i \leftarrow 0$ 
13:     end if
14:   end for
15:    $n \leftarrow n - 1$ 
16: until  $n > 0$ 
17: return  $\mathcal{M}$ 

```

Low Threshold	High Threshold	# Hue Bins		
		16	24	32
0.3	0.5	54.24%	53.63%	57.43%
0.3	0.6	56.15%	57.86%	58.68%
0.3	0.7	57.38%	58.65%	59.73%
0.4	0.6	59.86%	60.07%	60.54%
0.4	0.7	60.20%	60.29%	60.79%
0.4	0.8	60.47%	60.80%	61.02%
0.5	0.7	60.34%	61.25%	60.72%
0.5	0.8	59.53%	61.01%	58.96%
0.6	0.8	55.75%	60.82%	56.34%

Table 4.1: Performance of our method in terms of mean F-measure for different values of the parameters.

Threshold values	# Hue Bins		
	16	24	32
0.50	54.81%	54.09%	55.70%
0.55	54.94%	55.60%	55.75%
0.60	55.58%	55.72%	55.63%
0.65	55.22%	55.25%	53.95%
0.70	54.52%	52.85%	51.28%

Table 4.2: Performance of the system in terms of mean F-measure with standard thresholding for different parameter values.

4.1.2 Experimental results

We experiment using the skin dataset and report the performance using the mean F-measure, both of which are described in Section 3.1. Table 4.1 shows the performance of our method for different parameter values. The optimum performance is achieved for the 0.7 value of the high threshold and the low threshold is learned as 0.5.

In Table 4.2, we report the performance when standard thresholding (i.e., only one threshold value) is performed instead of hysteresis. It can be seen that the performance of this system is lower compared to our method. Thus, we gain significantly by doing neighbourhood analysis. Figure 4.2 presents some examples of the skin detection.

4.2 Efficient human upper body layout

We aim to localise human upper body parts (head, torso, upper arms, lower arms, hands) in an image. This problem has been extensively studied with concomitant improvements in efficiency, and the difficulty of the images that can be dealt with.

Methods such as Kumar et al. (2009) learn both appearance and layout configuration priors from training images, and nothing from the target image; whilst at the

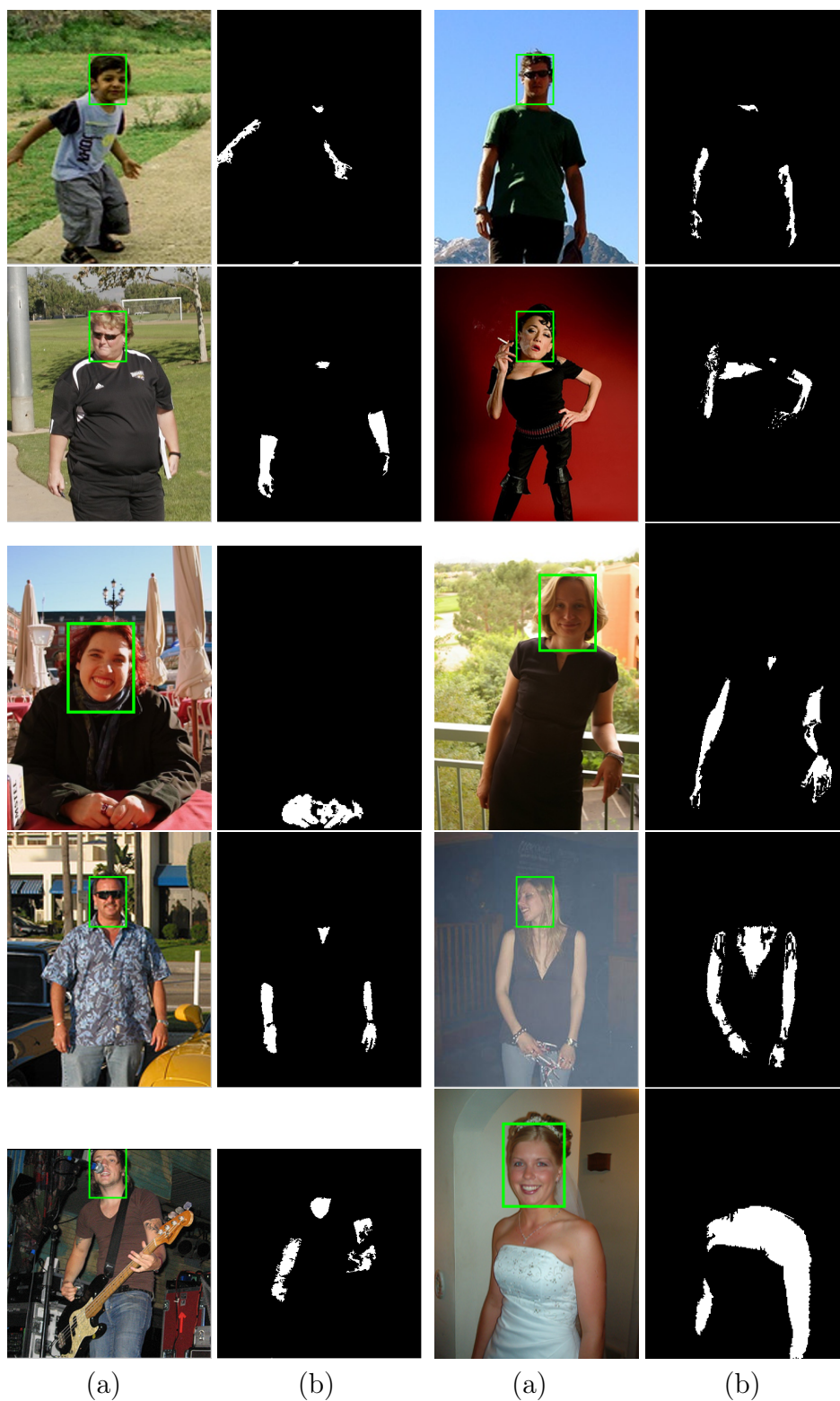


Figure 4.2: **Skin detection results.** (a) Input image with the face highlighted (b) Skin mask. The face has been masked out in all the results.

other extreme methods such as [Ramanan \(2006\)](#) learn the appearance (edges and colour) entirely from the target image. The approach we investigate here borrows ideas from both these extremes. As in [Ferrari et al. \(2008\)](#) our goal is to reduce the search space. To achieve this we determine skin regions in the target image, learning the skin colour from this image. These skin regions are used as hypotheses in an efficient hypothesise-and-verify algorithm, and the verification (and ranking) uses a model trained discriminatively on other images. Figure 4.4 shows an overview of the algorithm on two example images.

In this work, we contribute the following novelties: (i) We show that the search space (and number of features that need to be evaluated) can be reduced significantly using skin regions to hypothesise the hand and arm part locations (positions and orientations). (ii) We develop a light-weight discriminative human upper body model, which we learn as the combination of individual part detectors. Combining these contributions gives state-of-the-art results even after searching over a very reduced space. We demonstrate the performance by evaluations on the [Buffy stickmen](#), and the [PASCAL stickmen](#) standard publicly available datasets.

4.2.1 Overview

Our goal is to fit an 8 part (head, torso, upper arms, lower arms and hands) upper body layout model to near frontal humans. We parametrise the pose of each part as (x, y, ϕ) , where (x, y) is the location and ϕ the orientation (as shown in Figure 4.3). In addition for the arms, three scalings are used to account for foreshortening effects. A common scaling is used for all other parts. In order to perform the layout detection, we perform the following operations:

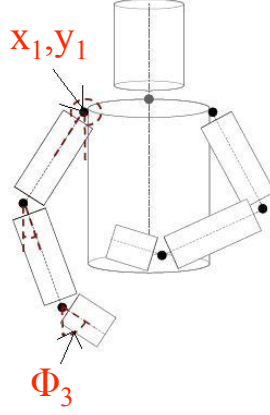


Figure 4.3: Model parameters for the human upper body.

1. Upper body detection. An upper body detector is used to estimate the scale and position of a torso in the image (Figure 4.4(a)). The detection also determines the head position and loosely constrains the shoulders. For the detector we use a method similar to that of [Eichner and Ferrari \(2009\)](#) which combines independent head and upper body detectors. Here we use a weighted combination of head and torso detectors. This two-part model is not learned separately but instead obtained directly from the parts-based model described in 4.2.3. To do so, from the parts-based model, we remove the classifiers for all the body parts except for head and torso.

2. Hypothesis generation from skin regions. The skin regions are used to hypothesise poses for the arms and the hands. This is based on the assumption that in an image either the hand or some portion of the arms is uncovered and hence skin regions identify them (Figure 4.4(e)). It is to be noted that facial skin pixels are masked out to prevent false hypotheses generation from the face region.

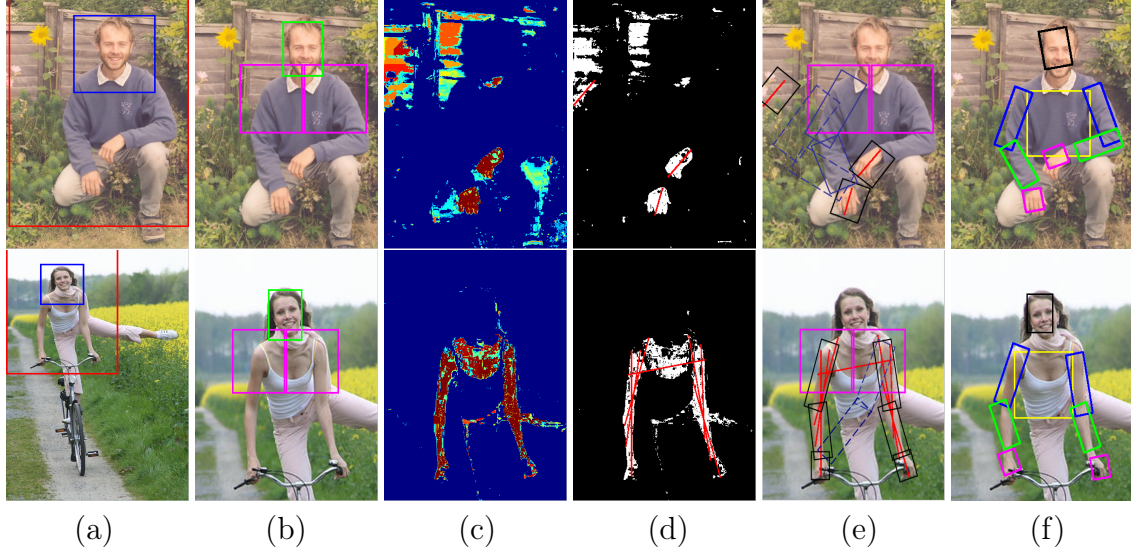


Figure 4.4: **Overview of the pose estimation system.** (a) **Upper body detection:** The inner rectangle shows the upper body detection. The outer rectangle specifies the region used for further search. (b) **Face and shoulder localisation** based on the upper body detector. The shoulder is quite loosely constrained. (c) **Skin detection:** Note the skin regions are correctly detected with few false positives. (d) **Lines fitted to skin regions:** the lines are used to hypothesise the extent and orientation of the arm and hand parts. (e) **Hypothesis generation:** Examples of the hypotheses. The black parts arise from the detected regions, the blue arise from the kinematics chain between the hands and shoulders. Dotted rectangles are those rejected by the model. (f) **Final result.**

3. Kinematic chain hypotheses. Given the hypothesised hands, lower or upper arms, the kinematic chain is completed using the loose constraint on the shoulders if necessary. All those hypotheses are included which are not identified by skin detection, but are valid as per the kinematic chain configurations. In Figure 4.4(e) the blue rectangles illustrate a sub-set of these hypotheses.

4. Verification using the parts-based model. The hypotheses are scored using the parts-based model, and the optimum is selected. In cases where the face is not detected, the layout is simply estimated by exhaustive search with the parts-based

model. Since this model is a tree structured graph the search is efficient (linear in both the number of parts and possible poses for each part (Felzenszwalb and Huttenlocher (2005))) but more expensive than the hypothesis and verify algorithm, and more prone to false positives.

4.2.2 Hypothesis generation

Skin regions are detected following the method described in Section 4.1. Upper body detection provides additional constraints on the position of the face, which results in removal of many false positives and therefore better skin detection. Lines are fitted to the skin region that will act as ‘rails’ for instantiating the hands, and lower and upper arms parts. For example the line orientation will restrict the orientation of the part, and the line length will restrict the extent of the sweep of that part.

Since skin regions can over and under segment the underlying limbs it is necessary to fit multiple lines to each skin region, and several methods are used to propose lines fitted to the skin regions.

Implementation details. Lines are fitted using a Hough transform (Duda and Hart (1972)) (algorithm is given in Appendix A), and also by finding the medial axis of the blob-shaped skin regions (Figure 4.4(d)). The medial axis often produces useful lines in the cases where only hands are visible (and the resulting skin regions are then approximately elliptical). The Hough fitting is more useful in the case of skin from arms (separate lines are generally fitted to the lower and upper arms where visible).

Since we do not know whether the skin region corresponds to the whole part or a portion of it, the hypothesis window (for each part) is slid over and beyond the

end of the line, stopping when only 8 skin-pixels remain in the window. By doing this, we ensure that all the valid locations of the part with respect to the line are included. The axis of the part is restricted to the line orientation, and for hand parts the position is restricted to line ends.

Finally, the kinematic chain is completed between hands and shoulders (Figure 4.4(e)). When completing the chain, the pose space for each part, (x, y, ϕ) , is discretized into 36 values for ϕ and 8 pixels intervals for x and y .

4.2.3 Discriminatively trained parts-based model

Hypotheses are scored using a learnt articulated parts-based model. The model is a tree structured graph whose vertices ν correspond to parts, edges ε define pairwise linkage between parts. The labelling function (for parts) f determines a score:

$$\text{score}(f) = \sum_{a \in \nu} \Theta_{a;f(a)} + \sum_{(a,b) \in \varepsilon} \Theta_{ab;f(a)f(b)} = W^T \Phi \quad (4.1)$$

Here $\Theta_{a;f(a)}$ and $\Theta_{ab;f(a)f(b)}$ are unary and pairwise scores (potentials) respectively, $f(a)$ is the label of a , and specifies the spatial location of the part, Φ represents the combined unary and pairwise features and W is the weight vector. Unary potentials encapsulate the appearance features and the pairwise potential enforces valid pairwise configurations between the parts.

We describe the appearance and pairwise terms below. The model is learned discriminatively from a set of training images using a large margin formulation. In previous approaches (Kumar et al. (2009)) all the parameters are learned simultaneously (e.g. the HOG appearance descriptors and the pairwise terms). Here, the part detectors are learnt first as in Desai et al. (2009), and their scores specify the

unaries. The final model then learns the scaling for the scores of each part detector, as well as the pairwise terms.

4.2.3.1 Appearance features and pairwise potentials

The appearance unary scores are based on linear classifiers for each part. Each of these linear classifiers is trained on the feature vector $(\theta_{a,f(a)})$ formed by concatenating the HOG and segmentation descriptors for a body part. The intuition behind using these two features is to capture both the edge (using HOG) and shape (using segmentation descriptors) information for the body parts.

Histogram of oriented gradient features. HOG features describe the appearance within an image as the distribution of intensity gradients or edge directions and are widely used for human body detection and localisation (Dalal and Triggs (2005)). We employ the implementation of Felzenszwalb et al. (2010b) which uses PCA to reduce dimensionality, and includes both contrast sensitive and insensitive features. Cells of size 24×24 pixels are used for the torso, and 16×16 pixels for all other parts. The histogram of a particular cell is normalised over the 4 possible 2×2 blocks of cells which contain that cell.

The dimensionality of the HOG features for the different body parts are: 1302 for the torso, 744 for the head, 1116 for the upper arms, 744 for the lower arms and 279 for the hands.

Segmentation descriptors. We employ a segmentation descriptor for each part following the method of Ramanan (2007). This represents the part’s segmentation as a binary mask. In our case the segmentation is obtained by thresholding a posterior probability that each pixel belongs to that part based on its colour.

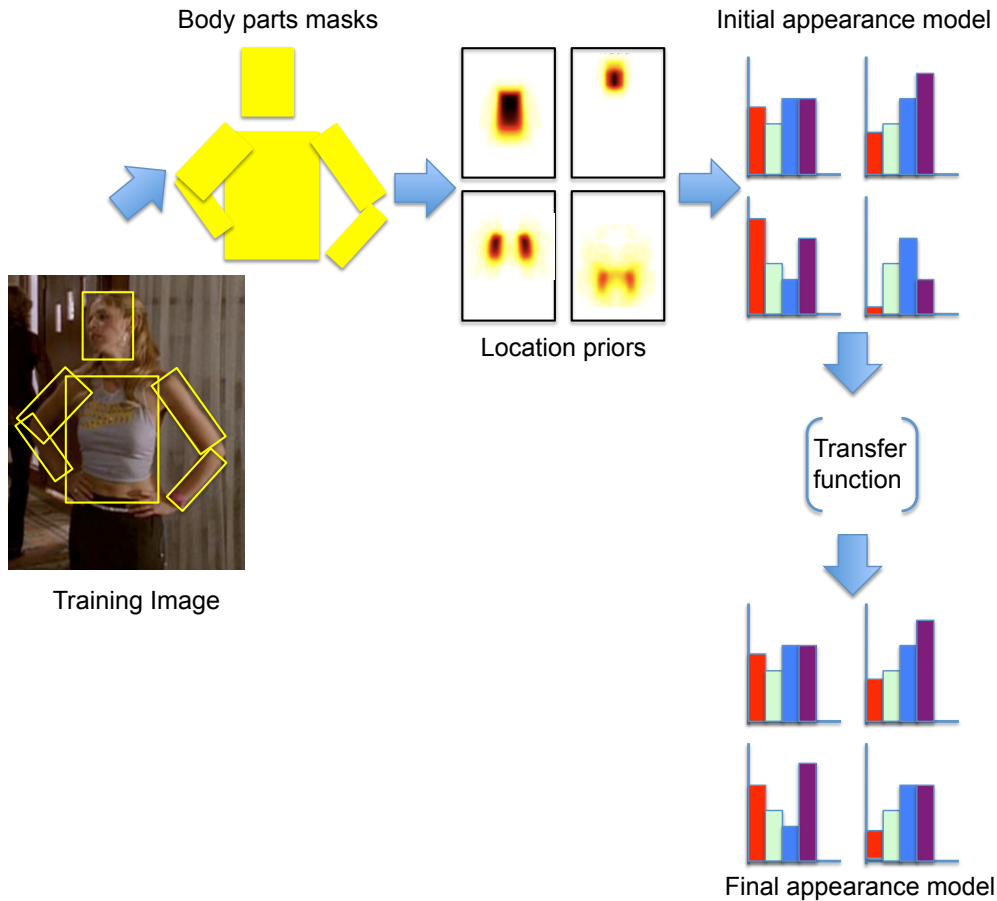


Figure 4.5: **Segmentation Descriptors.** Masks of body parts are obtained from the ground-truth annotation of the training images. The location priors are computed as means of all these masks. The initial appearance model is computed as the colour histogram over the probabilistic masks of body parts. A transfer function is learned which encapsulates similarity between appearance (in terms of colour) of the body parts. Using the transfer function, an updated appearance model for the body parts is obtained. Note: for hands the colour model of the head is used.

To obtain the posterior probability we require the foreground and background colour distributions. For the head and torso, foreground regions are defined by the upper body detector. For the other parts, foreground regions are obtained using the method of [Eichner and Ferrari \(2009\)](#). For each part i , the location prior (LP_i) is learned as the mean of the part masks over all the training images. Thus, $LP_i(x, y)$ is the fraction of training images where part i covers pixel (x, y) . Foreground appearance of the part (fg_i^{LP}) is then computed as the colour histogram over this probabilistic mask. Using a transfer mechanism the appearance of different body parts is combined to refine the foreground appearance model. The new appearance model of the part is given by: $fg_t = \sum_{i \in \nu} w_{it} fg_i^{LP}$, where w_{it} is the mixing weight of part i in the combination with part t . The mixing weight values are taken from [Eichner and Ferrari \(2009\)](#). Figure 4.5 illustrates this method. For all the body parts, the background colour distribution (bg_i) is determined by the pixels which have not been selected as foreground. By using the foreground and background colour distributions, we obtain the probability for every pixel for each of the distributions. The posterior probability is then computed as:

$$\text{posterior probability} = \frac{fg_i}{(fg_i + bg_i)}$$

The posterior probability is thresholded at 0.5 to obtain the binary segmentation masks for the parts.

The dimensionalities of the segmentation descriptors (which is just the binary mask) for the body parts are: 1155 for the torso, 352 for the head, 440 for the upper arms, 286 for the lower arms and 121 for the hands.

Pairwise features. The pairwise relations between the body parts is modelled by Potts' model where

$$\theta_{ab;f(a)f(b)} = \begin{cases} 1 & \text{if } (f(a), f(b)) \in \mathcal{L}_{ab} \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

and $\mathcal{L}_{ab}$ is the set of valid configurations between parts a and b .

4.2.3.2 Learning the model

To learn the model, we proceed in two stages: first a linear discriminative classifier w_a is learnt for each part a using the feature vectors $\theta_{a;f(a)}$, which as mentioned earlier, is the concatenated descriptor for part a computed at location $f(a)$. This gives the score for part a as $w_a^T \theta_{a;f(a)}$. Then second, a single weight vector W is learnt for the entire model where the feature vector is:

$$\Phi = (w_a^T \theta_{a;f(a)}, \forall a \in \nu; \theta_{ab;f(a)f(b)}, \forall (a, b) \in \varepsilon) \quad (4.3)$$

Φ is a 15 dimensional vector where the first 8 dimensions specify the unary scores for the parts and the other 7 refer to the pairwise features. This gives the classifier $W^T \Phi$ as in Equation (4.1).

Training is done using 300 images of episodes 3 and 4 from the Buffy stickmen dataset (described in Section 3.2). Our training system requires the bounding boxes for the 8 body parts (head, torso, upper and lower arms, hands). We manually perform 8 body parts bounding box annotation for all the training images.

The poses of the parts in the form of bounding boxes from the ground-truth annotation is used as the positive training data. We consider all other poses in the

image (other than positive) as the negative examples (i.e., our positive and negative images are the same, only the poses of the parts are different).

To learn the part detector w_a , we extract appearance features for the part from the positive and randomly sampled negative examples (whose overlap score with the positive examples is less than a threshold of 0.25), and learn a linear SVM that separates them with the maximum margin. Performance is improved by bootstrapping. To do this, we use the initial part detectors to get hard negative (i.e., highly ranked) examples (250 in number) and retrain the linear SVM using these newly found hard negative examples with the positive examples. This gives us better part classifiers which are then used to estimate the unary scores for the parts.

The weight vector W for the entire model is learnt using the discriminative training framework developed by [Kumar et al. \(2009\)](#). We are able to learn the model in 3 iterations of their algorithm.

4.2.4 Body parts localisation results

Datasets. All evaluations are performed using two widely used datasets *viz.* Buffy stickmen and PASCAL stickmen described in Section 3.2.

We use 276 images from episodes 2, 5 and 6 of the Buffy stickmen dataset (Section 3.2) for testing (the same images are used by other works to benchmark their performance). All the 549 images of the PASCAL stickmen dataset (Section 3.2) are used for testing.

For evaluation, we require results in the form of stickmen which are obtained here by drawing lines between midpoints of the width edges of the detected bounding boxes. Performance is measured in terms of the percentage of correctly estimated body parts (PCP) and detection rate (DR) explained in Section 3.2.

	Hands	Lower Arms	Upper Arms
Buffy Dataset	96.79%	54.50%	86.30%
PASCAL Dataset	95.63%	60.96%	87.24%

Table 4.3: **Percentage of search space reduction to localise body parts after hypothesis generation.**

We evaluate the efficiency of our approach in two ways: (i) The percentage of search space reduction after hypotheses generation; (ii) The overall performance of the model for the layout detection.

4.2.4.1 Search space reduction

We perform a quantitative analysis of the search space reduction (compared to exhaustive search) using our hypotheses generation technique. Table 4.3 gives the average search space reduction to localise the body parts over all the test images where face was detected. It can be seen that the search space reduction is significant.

4.2.4.2 Layout detection results

(a) Improvement after adding different appearance features. Table 4.4 gives a summary of the improvement in the localisation accuracy we obtain after adding different cues. It can be seen that the results improve consistently as appearance features are added. This increase is smaller for the Buffy dataset (compared to the PASCAL dataset), because just using HOG features is sufficient to localise body parts to a high degree of accuracy.

Some loss in accuracy (over exhaustive search) is due to false detection of faces in images at the initial stage. If we evaluate the results for only true positive instances (face present and detected; face not present and not detected) then the accuracy

Method	Buffy Dataset		PASCAL Dataset	
	DR	PCP	DR	PCP
HOG only	86.23%	85.08%	58.65%	68.81%
HOG and segmentation	87.68%	87.74%	66.30%	71.8%
Search space reduction	89.13%	85.64%	67.94%	70.64%

Table 4.4: **Improvement after adding different features.** Exhaustive search is used for the HOG and ‘HOG and segmentation’ columns. The ‘search space reduction’ column replaces the exhaustive search with the hypothesis and verify algorithm (using the same HOG and segmentation features).

Buffy Dataset				PASCAL Dataset			
Face present (88.41%)		Not present (11.59%)		Face present (85.6%)		Not present (14.4%)	
Face detected (79%)	Face not detected (9.4%)	Face detected (5%)	Face not detected (6.6%)	Face detected (74.1%)	Face not detected (11.5%)	Face detected (8.6%)	Face not detected (5.8%)
87.21%	78.26%	73.80%	80.95%	71.84%	68.96%	63.33%	64.88%

Table 4.5: **Analysis of the performance.** The accuracy (PCP) of the model declines due to false detection of the faces, which results in the computation of wrong skin colour histogram. Note: The number next to the title shows the percentage of images where we encounter such situation. For example, in the PASCAL dataset, face is present but not detected in 11.5% of the total test images. Here, we consider the face to be present in an image if it is fully visible.

(PCP) for the Buffy dataset is 86.80% and for the PASCAL dataset, 71.20%.

Table 4.5 gives an analysis of the performance (PCP) over all the cases for the detection and the visibility of the face in test images. It can be seen that the performance is most affected in the case when we wrongly detect a face in the image (i.e., the ground-truth face is either partly visible or not visible). In this situation, the computed skin colour is wrong and hence the resulting skin regions are not correct. For other cases, we found that the hypothesis and verify method performs poorly in cases where not all the skin regions are detected precisely (incomplete recall). This results in incomplete hypothesising of the hands’ locations.

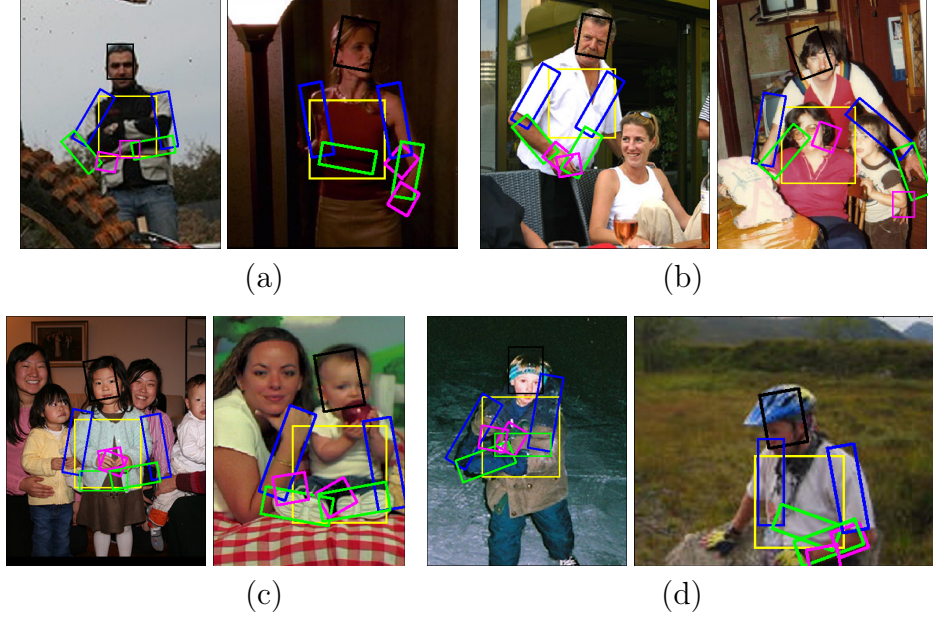


Figure 4.6: False detection cases. (a) Detection failed due to self occlusion of body parts. (b) Occlusion due to other objects. (c) Scaling is not consistent for children as the size of their head is bigger relative to their body. (d) Hands are not detected as they are covered with gloves.

The detection further fails when the body parts are occluded by other parts or objects (Figure 4.6(a,b)). Our method also sometimes does not localise arms correctly for images of children (Figure 4.6(c)). This is because for children, the size of the head is large compared to the body. Due to this, we get the wrong scaling at the initial stage from the upper body detector. Our model localises hands on the basis of identified skin regions. However, this assumption fails when hands of the person are covered in the image (Figure 4.6(d)).

(b) Comparison with state-of-the-art. We compare our results with the state-of-the-art methods over both Buffy and PASCAL datasets (Table 4.6). We are using the same images and adopt the same evaluation criteria as that of [Eichner and Ferrari \(2009\)](#). It can be seen that our method outperforms the state-of-the-art in both the

	Buffy Dataset			PASCAL Dataset	
	Our Method	Eichner and Ferrari (2009)	Andriluka et al. (2009)	Our Method	Eichner and Ferrari (2009)
DR	89.13%	85.14%	-	67.94%	65.57%
PCP	85.64%	80.28%	73.5%	70.64%	69.31%

Table 4.6: **Evaluation of the layout detection results.** We compare the performance of our layout detector with other state-of-the-art methods. In [Andriluka et al. \(2009\)](#), the test was performed on only the Buffy dataset and the detection rate was not made available.



Figure 4.7: **Sample results.** The top two rows of images are from the Buffy test dataset, the bottom two rows are from the PASCAL test dataset. For the 5 images in the last row a face is not detected, and hence exhaustive search is performed. In all other cases the hypothesis and verify algorithm is used.

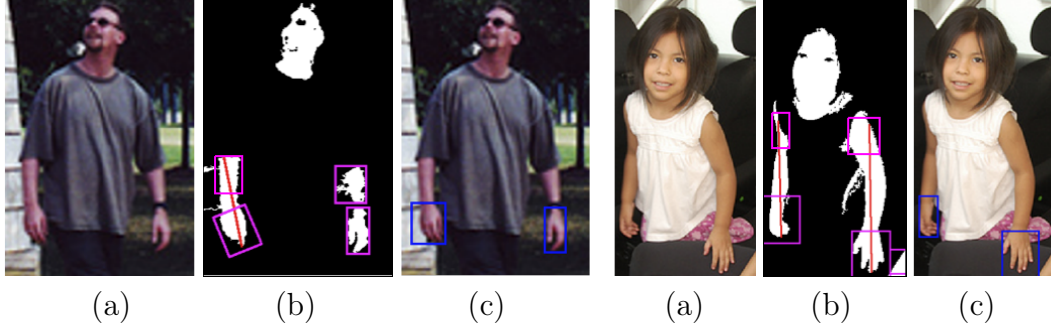


Figure 4.8: **Hypotheses generation from the skin regions.** (a) Original image. (b) Lines are fitted to the skin region which are used to hypothesise the extent and orientation of the hand. Hands are hypothesised at both ends of the fitted lines. If the skin region resembles a blob then the whole skin region is hypothesised as a hand. (c) Hand detections remaining after verification using the model.

detection rate and PCP. Note: The results reported in [Eichner and Ferrari \(2009\)](#) for the PASCAL dataset changed for the current evaluation criteria. The results reported in the Table 4.6 are the latest results available from the authors' website¹. Figure 4.7 shows some qualitative results for our method.

4.3 Hand detection using skin regions

We detect hands in images by following a two step process: first skin regions are obtained as explained in Section 4.1, then they are used to instantiate hypotheses for hands.

Hands are hypothesised from the detected skin regions using a similar technique as described in Section 4.2. Lines are fitted to the skin regions using Hough transform and by finding the medial axis of blob shaped skin regions. The hands are then hypothesised at either end of the fitted lines (Figure 4.8). The size of the box for

¹http://www.vision.ee.ethz.ch/~calvin/ethz_pascal_stickmen/downloads/README.txt

Method	Hand AP%
BCNPCL	3.3%
Our Method	10.4%

Table 4.7: **PASCAL VOC 2010 person layout competition results.**

the hand depends upon the width of the skin region at the end of the line. Hands are also detected as the bounding boxes around blob shaped skin regions. A blob shaped skin region qualifies as a hand depending on its geometric properties, such as: ratio of major and minor axis (< 2.5), eccentricity (< 2) etc., where eccentricity is the measure of how much a shape deviates from being circular. The values for the geometric properties are learned over the validation set. No hypothesis is proposed from the facial skin regions. The set of boxes from all lines and blobs are the proposals from the skin detection, B_{SD} .

Verification of hand hypotheses. The hand hypotheses B_{SD} obtained using the method above are verified and scored using the pictorial structure model described in Section 4.2. The only difference is that the discriminative classifier for the hand, w_{hand} , is trained using an RBF SVM classifier. The hypotheses are scored with their max-marginal values for the given parts-based model, i.e.,

$$\text{score}_{hand;l} = \max_{f, f(hand)=l} \left\{ \sum_{a \in \nu} \Theta_{a;f(a)} + \sum_{(a,b) \in \varepsilon} \Theta_{ab;f(a)f(b)} \right\}, \quad \forall f \quad (4.4)$$

where $\text{score}_{hand;l}$ is the score for the hand bounding box at location l . This is computed by using forward-backward message passing of the belief propagation algorithm (Kumar (2008)).

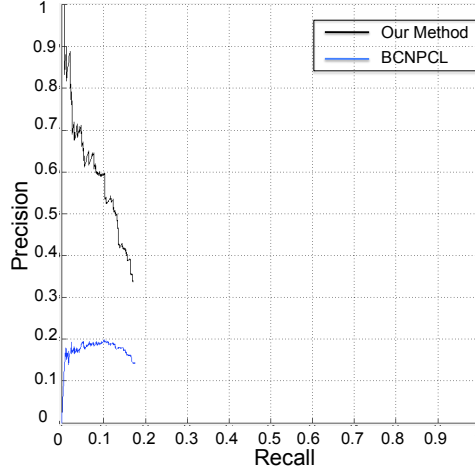


Figure 4.9: PR curve for the hand detection task of PASCAL VOC 2010 person layout competition.

4.3.1 PASCAL VOC 2010 results

The person layout problem of the PASCAL VOC challenge (Everingham et al. (2011)) involves predicting the bounding box and label of each part of the person (head, hands and feet). It is a very challenging task with the real-world images from different viewpoints and highly varying layout configurations of the people.

We submitted hand detection results of our method to the PASCAL VOC 2010 layout competition described in Section 3.6. With our method we were able to win the competition for the hand detection task. Table 4.7 shows the performance of our method compared to the other submitted entry. Figure 4.9 shows the PR curve for hand detection results. Some qualitative results are shown in Figure 4.10.

4.4 Conclusions

In this chapter, we demonstrated that accurate skin detection can help to solve some of the difficult problems in computer vision, such as upper body layout estimation



Figure 4.10: Sample results from the PASCAL VOC 2010 competition.

and hand detection. Upper body layout can be computed very efficiently by searching over a highly reduced pose space. Hand detection can also be performed if skin regions can be detected precisely in images. However, for both these methods, we relied on the fact that hands and some of the arms are usually found uncovered in images. We used a pictorial structure model of the human upper body which fails to perform well when body parts are occluded. Furthermore, the method was specialised for layout estimation in only near frontal poses. In Chapter 7, we will explain a method for layout estimation which overcomes both these problems. In the next chapter we will describe our improved hand detector which uses not only colour but also edges and contextual information from the image.

Chapter 5

Hand detection using multiple proposals

In the previous chapter, we presented the method of performing hand detection using skin regions in the image. In this chapter, we design and evaluate a hand detector using a two-stage hypothesise and classify framework. The previous chapter's skin detector now becomes one (of three) method for proposing hand locations (the hypotheses). In our framework, first, hand hypotheses are proposed from three independent methods: a sliding window hand-shape detector, a context-based detector, and a skin-based detector (explained in Section 5.1). Then, the proposals are scored by all three methods and a discriminatively trained model is used to verify them (described in Section 5.2). Figure 5.1 overviews the detector. The three proposal mechanisms ensure good recall, and the discriminative classification ensures good precision. In addition, we also perform a novel super-pixel based non-maximal suppression in which suppression is done over bounding boxes overlapping the same super-pixel. This is presented in Section 5.3.

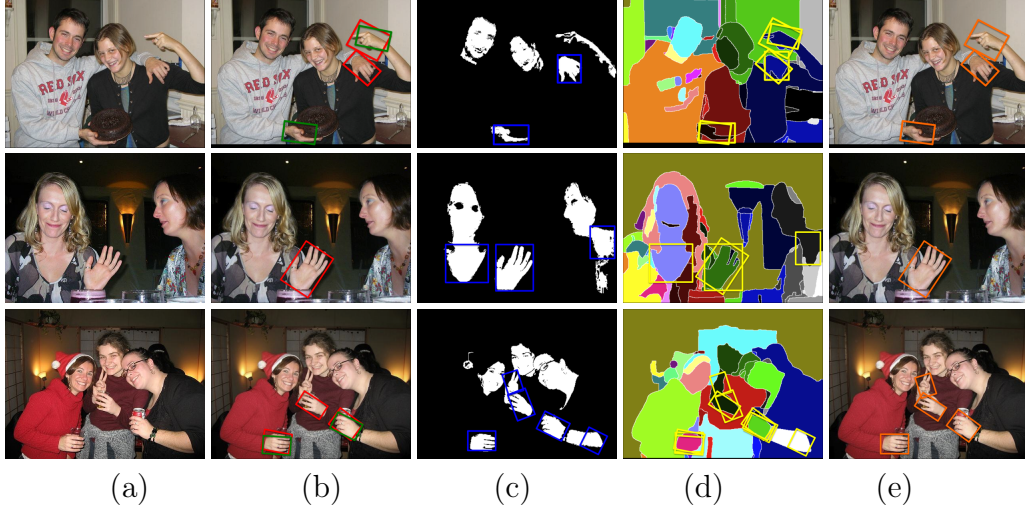


Figure 5.1: **Overview of the method.** (a) Original image. (b) Some of the hypotheses proposed by hand and context detector. Bounding boxes in red are proposed by the hand detector and in green by the context detector. (c) Skin detection and hypotheses generation. (d) Super-pixel segmentation of the image with combined hypothesised bounding boxes from the three proposal schemes. Using super-pixel based non-maximum suppression (NMS), overlapping bounding boxes are suppressed. (e) Final detection after post-processing.

We show in Section 5.4 that this detector can achieve very good recall and precision on unconstrained images from multiple sources, and that it exceeds the state-of-the-art performance of [Karlinsky et al. \(2010\)](#), and the PASCAL VOC person layout challenge ([Everingham et al. \(2010b\)](#)).

In Section 5.5, we show how this result can be improved further if the bounding box of the human is provided as in PASCAL VOC person layout challenge.

5.1 Proposal methods

In this section we describe the three proposal methods: shape, context, and skin colour. Each of these delivers a number of hypotheses for a bounding box for the hand, specified as a rotated rectangle (i.e. it is not axis aligned).

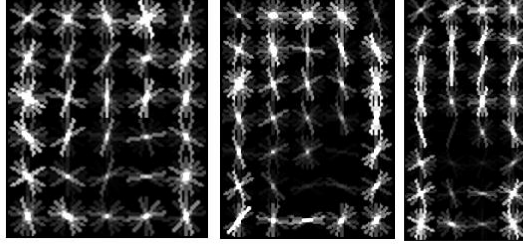


Figure 5.2: Root filters for the three components of the hand-shape detector. The first two filters cover frontal pose and the third filter profile.

5.1.1 Hand shape detector

The hand shape detector proposes bounding boxes for hands using Felzenszwalb et al. (2010b)’s parts-based deformable model based on HOG features (Dalal and Triggs (2005)). The detector is a mixture over three components (Figure 5.2), where each component represent a different aspect of the hand. Learning is done using the training set of the hand dataset (Section 3.3). The training images are rotated, such that all the hand instances are aligned (as shown in Figure 5.3(a)). Testing is performed at 36 different rotations of the image (at standard 10° intervals). For each image the proposed bounding boxes are given by the set $B_{HD} = \{b \in B \mid \beta_{HD}(b) > t_h\}$, where β_{HD} is the scoring function (Felzenszwalb et al. (2010b)) of the hand detector, B is the set of all detected hand bounding boxes, and t_h is the threshold of the hand detector chosen to give 90% recall on the training set.

5.1.2 Context detector

The context detector proposes a hand bounding box depending on its context. The motivation behind this is that the end of arm may be more visible or recognisable than the hand, and could provide vital cues for hand detection. In order to learn the context, a parts-based deformable model (Felzenszwalb et al. (2010b)) is



Figure 5.3: (a) Rotated Training images so that bounding boxes are axis aligned. (b) Context captured around the hand bounding box. The blue rectangle shows the hand bounding box and the red shows the extended box used to capture context around the hand. The context is captured over a region having the same height and twice the width as the hand.

trained from the hand bounding box training annotations extended to include the surrounding region, as shown in Figure 5.3(b). Again, a mixture model with three components is learnt. It should be noted that unlike other methods, which model adjacent body parts such as the arm explicitly, here the area surrounding the hand is instead modelled directly in a discriminative manner. Due to this although the detector is learned over a relatively varied region, it is less altered by occlusion of body parts. For training all the images are rotated so that the bounding boxes have the same orientation (axis aligned) (Figure 5.3(b)), and testing is performed at 10° intervals of rotation. Hand bounding boxes are obtained from the detected context boxes by shrinking them. Thus, for each image the proposed boxes are given by the set $B_{CD} = \{b \in B \mid \beta_{CD}(b) > t_c\}$, where β_{CD} is the scoring function of the context detector (Felzenszwalb et al. (2010b)), B is the set of all hand bounding boxes, and t_c is the threshold of the context detector chosen to give 90% recall on the training set.



Figure 5.4: **Max-pooling of context detector scores.** (a) Detected context boxes. (b) Context box with maximum score.

5.1.3 Skin-based detector

The skin-based detector is our method of proposing hand hypotheses from detected skin regions in the image which is explained earlier in Section 4.3. The set of hypotheses from skin-based detector is denoted by B_{SD} .

5.2 Classification of hand proposals

The hypotheses proposed by the different proposal schemes are combined and are then evaluated using a second stage classifier. The complete set of hypotheses is given by the union $B_h = \{B_{HD} \cup B_{CD} \cup B_{SD}\}$. Three scores are then computed for each hypothesised bounding box ($b \in B_h$) as follows:

Hand detector score. A score (α_1) obtained from the hand detector.

$$\alpha_1 = \beta_{HD}(b) \tag{5.1}$$

where β_{HD} is the scoring function of the hand detector.

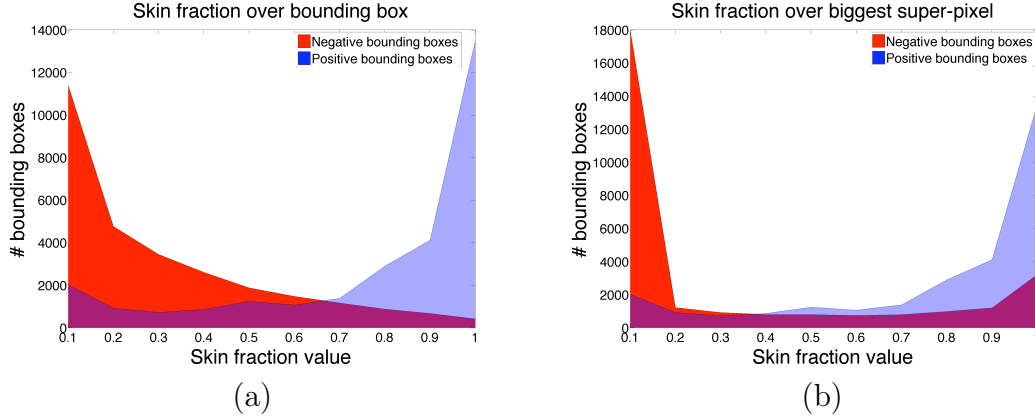


Figure 5.5: **Bayes risk plots.** The distribution of skin fraction value for positive bounding boxes is shown in blue, the negative bounding boxes in red, and the Bayes risk in purple. Bayes risk represents the minimum possible risk (or error) for an estimator system. (a) Skin fraction computed over all the pixels of the bounding box. (b) Skin fraction computed for pixels belonging to the biggest super-pixel in the bounding box. It can be seen that (b) has a far lower Bayes risk and therefore can discriminate better between the positive and negative boxes. The plots are generated using the training set of hand dataset (Section 3.3).

Context detector score. In order to include some deformation between the hand and its context, max-pooling of scores is done over all boxes (translated and rotated) having an overlap with the given bounding box, b , above a 0.5 threshold as shown in Figure 5.4. This gives some degree of invariance to rotation and translation changes. Let B'_h be the set of all bounding boxes having overlap score greater than the threshold value with the given bounding box b . Then the context detector score is given by,

$$\alpha_2 = \max_{b_h \in B'_h} (\beta_{CD}(b_h)) \quad (5.2)$$

where β_{CD} is the scoring function of the context detector.

Skin detection score. For the skin detection score, a straight forward choice would be to use the skin fraction (i.e., the fraction of pixels that are skin in a

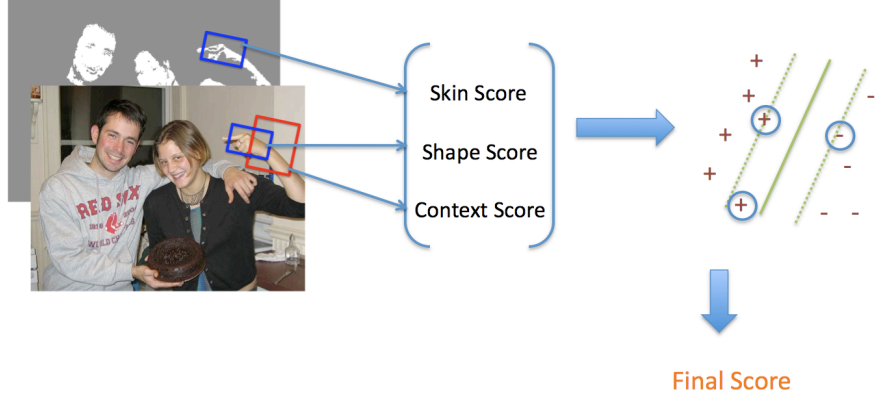


Figure 5.6: Classification of proposals scores.

bounding box b) as the feature. This approach is not suitable for bounding boxes as the boundaries are not tightly aligned with the hand, and may include the arm or other skin regions for example. However, a hand's appearance is often visually coherent and can be obtained as a single super-pixel. Thus, the image is first split into super-pixels (Arbelaez et al. (2009)), then for a bounding box, the skin fraction (α_3) is computed for the biggest super-pixel within it. This gives a better discriminative feature than skin fraction alone (Figure 5.5).

Classification of scores. The three scores for a given bounding box are combined into a single feature vector, $(\alpha_1, \alpha_2, \alpha_3)$, and a linear SVM classifier (Burges (1998)) is learned over the combined feature space using a standard SVM-solver (Joachims (1999)) as shown in Figure 5.6. This final classifier is used to compute confidence score for all the bounding boxes.

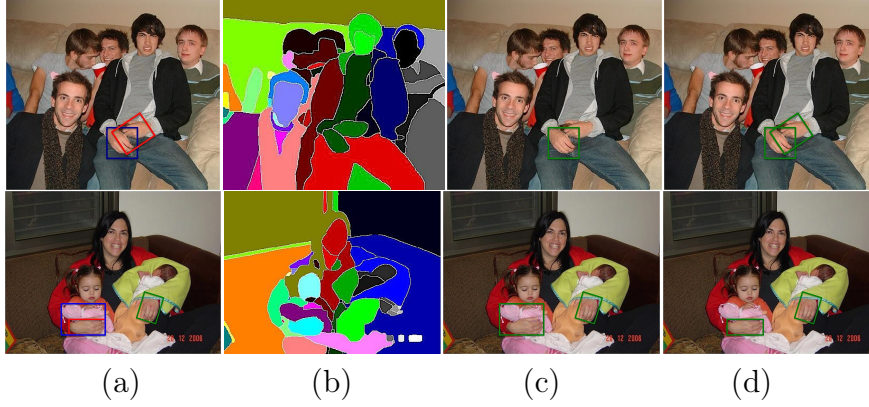


Figure 5.7: **Comparison of conventional NMS with super-pixel based NMS.** (a) Bounding boxes shown in blue and red are overlapping. (b) Super-pixel segmentation of the image. (c) The red bounding box is suppressed by conventional NMS. (d) Super-pixel NMS retains the correct boxes.

5.3 Super-pixel based NMS

Typically a detection algorithm returns a number of overlapping bounding boxes, and non-maximum boxes are then suppressed depending on their overlap with other high-scoring boxes (Canny (1986)). However, this suppression in general does not use any visual information from the image. This sometimes results in losing detections for the objects if multiple partially overlapping instances are present in the image.

We propose a modification of the traditional non-maximum suppression (NMS) technique, and instead incorporate further image information into the NMS process. As in the case of the skin detection score, we make use of super-pixels to capture the visual coherence of the hand. The image is first split into super-pixels, and then NMS is applied over all the boxes overlapping the same super-pixel. The overlap threshold for NMS is 0.4. If a box is overlapping more than one super-pixel then it is associated with the one that it is overlapping with the most. Figure 5.7 shows

some examples where super-pixel based NMS performs better than the conventional NMS technique.

To avoid cases where the detected bounding boxes do not fit tightly around the true hand, the NMS surviving box is fitted tightly around its enclosed super-pixels if the super-pixel resembles a blob. A super-pixel is deemed a blob if the ratio between its major and minor axis is less than a threshold (2.5). Detected hand boxes which overlap with the face regions (localised using the face detector) are also removed as part of the post-processing. By applying these post-processing steps, performance of the system improves significantly (Table 5.1).

Time Analysis. The time taken for the whole detection process is about 2 minutes for an image of size 640×360 pixels on a standard quad-core 2.50 GHz machine. The hand and context detectors employ the efficient cascade implementation of Felzenszwalb et al. (2010a).

5.4 Results

The model is evaluated on the test set of hand dataset (Section 3.3) and two external datasets: signer dataset (Section 3.4) and PASCAL VOC 2010 person layout test dataset (Section 3.6). We compare to the performance of previous work on these external datasets.

5.4.1 Hand dataset

For all of the following experiments the model is trained on the hand training dataset and model parameter values are determined on the hand validation dataset, see

Proposal schemes	Hand score	Hand and context	Hand, context and skin scores	Recall
Hand	<i>33.57 / 36.54</i>	34.23 / 37.78	35.31 / 41.09	74.09
Hand, context	36.19 / 39.22	39.06 / 42.68	41.59 / 47.13	82.12
All three	36.30 / 39.63	39.38 / 43.48	42.25 / 48.20	85.30

Table 5.1: Performance on the hand test dataset in terms of average precision (before post-processing / after post-processing). The final column gives the recall after post-processing. Along a row the variation in performance can be seen for a given proposal scheme after including different scores in the final classifier. The increase in recall with the addition of the different proposal schemes is evident. The baseline is the ‘hand proposal, hand score’ (in *italics*), and the final performance is shown in bold.

Table 3.1.

Parameter estimation. The model parameters include: size of context bounding box around the hand, weight values of the second stage classifier and the SVM parameter C .

For the context box, the following parameter values were investigated: $(\{h, 2w\}, \{2h, 2w\}, \{h, 3w\} \text{ and } \{2h, 3w\})$, where ‘h’ and ‘w’ are the height and width of the hand bounding box respectively. The APs obtained for these different sizes are $[46.13, 38.75, 44.04, 40.97]$. Consequently, a context region of the same height and twice the width as the hand bounding box is used (refer Figure 5.3(b)). The weights learnt for the linear SVM used to blend scores from the proposal schemes are $\mathbf{w} = (1, 0.4, 0.36)^\top$ for hand detector score, context detector score and skin detection score respectively. The value of parameter C was learnt as 1.0.

Test set performance. The basic hand detector (i.e., no context or skin detection) is used as the baseline. Table 5.1 shows the average precision for different proposal schemes after including different scores in the model. The baseline precision is the precision obtained by using just the hand detector as the proposal scheme

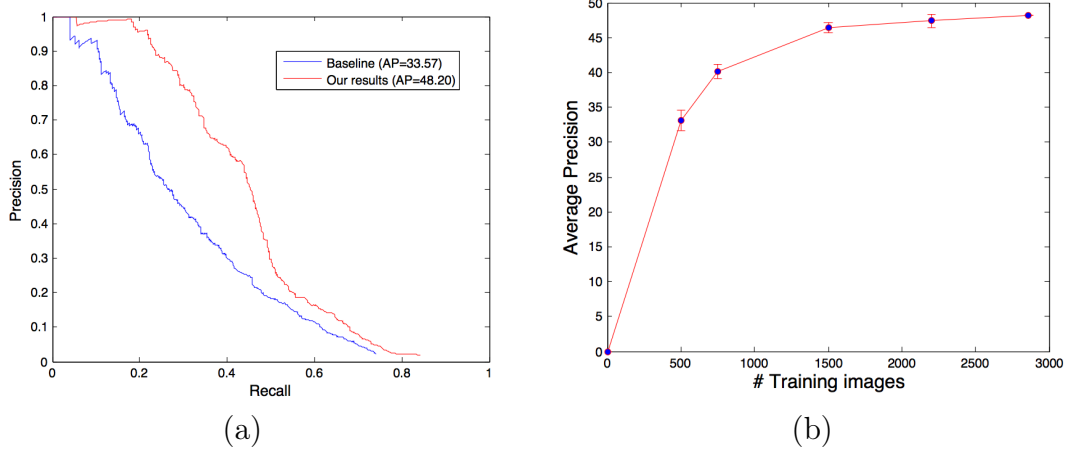


Figure 5.8: (a) Precision-Recall curve comparing the baseline and final results. (b) Variation in Average Precision with the number of training instances. To generate the plot, five sets of fixed size are randomly sampled from the hand training set and a model learnt from each. The graph shows the mean AP and standard deviation obtained over the test set for the five models. For the last data-point no such split is done as all of the training data is used. It can be seen that AP increases with the the size of the training set and reaches saturation after 2500 images.

and the hand score as the only score for the final classifier. Compared to this baseline it can be seen that there is around 15% improvement in average precision and 11% increase in recall by the final model.

If the conventional NMS is used for post-processing, then the AP of the system reduces from 42.25 to 40.79. The PR curve comparing baseline results with our results after post-processing is shown in the Figure 5.8(a). Figure 5.8(b) shows the variation in AP with increase in the training data.

5.4.2 Signer dataset

The model used for this experiment is trained on the hand training dataset (Table 3.1). The test dataset that is used for this experiment is the ‘5-signers’ dataset (Section 3.4).

Setting	1 max	2 max	3 max	4 max
Karlinsky et al. (2010)	84.9	92.8	95.4	96.7
Our method	76.67	90	95.64	97.44

Table 5.2: Comparison of results on the Signer dataset. ‘1 max’, ‘2 max’ etc. are the detection performance within the top ‘k’ hand detections per ground-truth hand instance.

Method	BCNPCL	Oxford	Ours
AP	3.3	10.4	23.18

Table 5.3: Comparison of our method with other submissions for the PASCAL VOC 2010 person layout challenge for hand detection task [Everingham et al. \(2010b\)](#). Scores are obtained by submitting results to the competition evaluation server.

[Karlinsky et al. \(2010\)](#) use the ground-truth position and scale of the head bounding box to fit a chain model originating from head up-to the hand. They consider the hand detection to be correct if it is within half face width from the ground-truth location of the hand. Table 5.2 compares the result from their method with ours. For evaluation the same criteria is used as described in Section 3.4.

It can be seen that for $k = 3$ and 4, our method performs better than [Karlinsky et al. \(2010\)](#). For smaller values of k , [Karlinsky et al.](#)’s method works well because the chain model enables them to disambiguate hands from the background better. However, this can also be a disadvantage of their model as the method requires the position of head at test time, and can not work if the head or some-other body parts are occluded. Our method does not require other body parts to be visible and is therefore not restricted to images having un-occluded humans in frontal poses.

5.4.3 PASCAL VOC 2010 person layout test dataset

For this problem, we train the model on the hand training dataset (Table 3.1) and evaluate it for competition 8 (i.e., training on own data) of the PASCAL VOC 2010



Figure 5.9: Examples of high-scoring detections on the three datasets. Top row: images from the hand dataset; Middle row: images from the Signer dataset; Bottom row: images from the PASCAL VOC 2010 person layout test set.

person layout challenge (Section 3.6). Each of the provided human bounding boxes is re-sized such that the minimum width is at-least 300 pixels. This is done to ensure that the hands in the image are reasonably large for detection. For every figure, either one or two hand bounding boxes are returned depending upon the confidence scores obtained from the model. The final confidence score for the human is the average of the scores for each of the hands.

As shown in Table 5.3, we report a very good performance over this dataset, beating our previous winning entry by a factor of two. A sample of high scoring detections from the three datasets are shown in Figure 5.9. More qualitative results on the hand dataset are given in Appendix B.

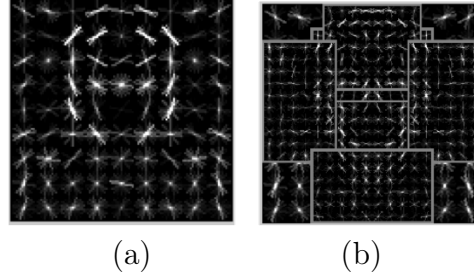


Figure 5.10: (a) Root filter for the head detector. (b) Different parts of the parts-based model of the head.

5.5 Rescoring using position & scale attributes

The hand detector described earlier performs detection independently without using any prior knowledge about the hand position or its size. However, for the PASCAL VOC layout challenge (Section 3.6), human bounding boxes are provided and the task is to detect body parts (head, hands and feet) within the given ROI. This, extra information can be used for improving results along with reducing the search space for detections. Furthermore, a head can be detected more reliably in an image compared to a hand. The size and position of a head also provide vital cues for the position and size of the hand. We include this information by performing head detection as explained below.

Head detection. Head candidates are generated using a parts-based detector of human head trained by [Marin-Jimenez et al. \(2011\)](#) (Figure 5.10). The model is capable of detecting human heads in different poses (for example, frontal, profile, backward). The candidates for the head so obtained are rescored using a linear SVM classifier trained using positional and scale attributes. The confidence value from this classifier is then used as the new detection score.

The linear SVM classifier for head candidates is trained over a feature vector

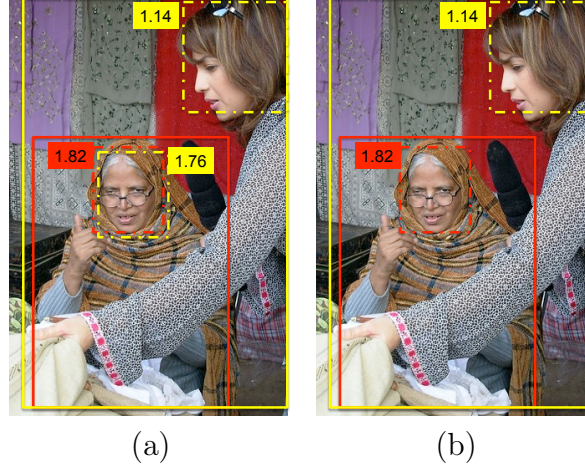


Figure 5.11: Head assignment for overlapping human ROIs. In the image, the two given human ROIs (shown as yellow and red solid rectangles) are overlapping. (a) One of the head detections (shown as dashed yellow) of yellow human ROI is overlapping with the head detection (shown as dashed red) of red human ROI. (b) The overlapping head detection is assigned to red human ROI as it has no second head detection.

(ϕ_{head}) formed by concatenating: (i) confidence score from the detector, (ii) relative size of head and human ROIs, (iii) fraction of area of head ROI in human ROI. The intuition behind using features (ii) and (iii) is to suppress any bounding box which either has an abnormal height with respect to the size of the image, or which does not fully lie within the given bounding box of the person.

In test images there are a few instances where multiple overlapping people are present in the image. Therefore, the same head may be detected twice for different people. In order to resolve this ambiguity, the head is assigned to the bounding box of the person which has either no second head detection (as shown in Figure 5.11), or a lower scoring second head detection.

Rescoring hand detections using position and scale attributes. We train a linear SVM classifier over positional and scale attributes of hand detections com-

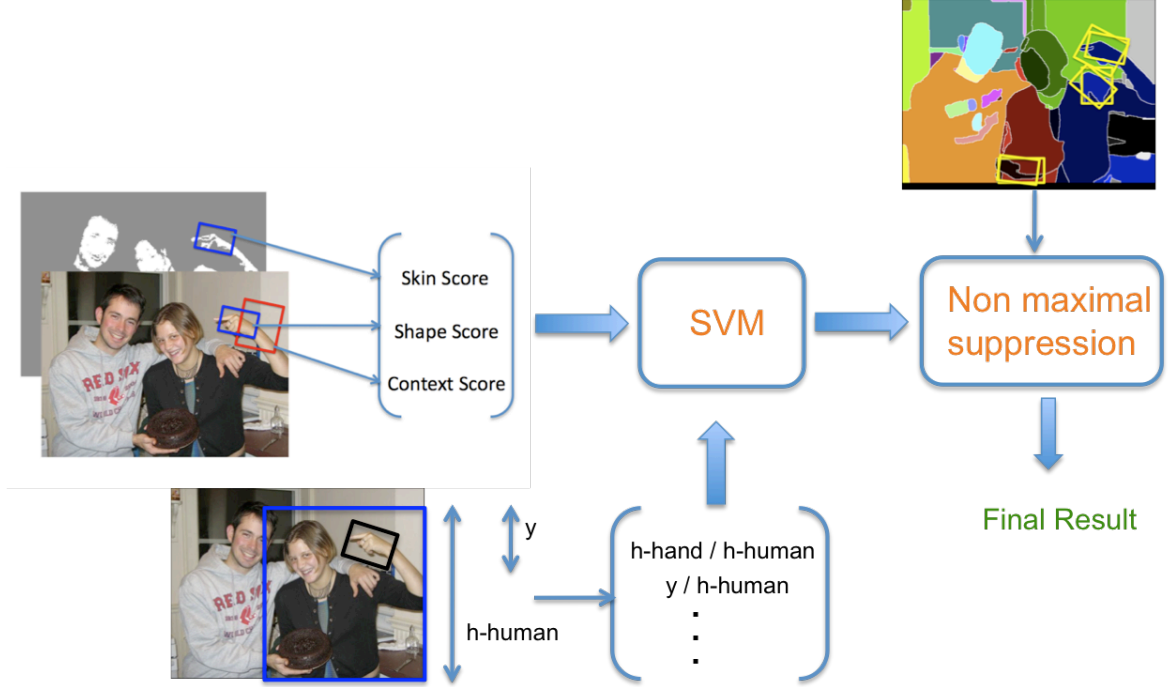


Figure 5.12: New hypotheses generation and classification framework for hand detection.

puted in reference to given human ROI and the detected head. Figure 5.12 shows the modified detection pipeline after adding this extra step.

The linear classifier for hand candidates is trained over a feature vector (ϕ_{hand}) formed by concatenating various positional and scale attributes, which are: (i) the confidence score of hand detector obtained by classifying proposal scores as explained in Section 5.2, (ii) position within human ROI, (iii) relative position of hand and head ROIs within human ROI, (iv) relative size of hand and head ROIs, (v) overlap score of head and hand ROIs, (vi) fraction of area of hand ROI in human ROI.

Implementation details. The classifiers for head and hand are trained using the PASCAL VOC 2011 person layout dataset (Section 3.6) with positive examples as the detection bounding boxes having an overlap score larger than 0.5 with annotated

	Candidates AP%	AP% after rescoring	Yang and Ramanan
Head	68.59±3.14	81.04±1.77	37.71±1.73
Hand	23.97±2.24	28.32±2.12	0.27±0.02

Table 5.4: **Increase in AP after rescoring using a linear classifier.** Candidates AP is the AP obtained by evaluating part candidates obtained from the detectors. Middle column is the result achieved after rescoring part candidates using local features. The experiment was performed over training and validation dataset of VOC 2011 layout dataset. Last column shows the baseline results of Yang and Ramanan (2011).

part ROIs.

For every human ROI the top scoring head candidate is returned as the detected head. The scores of hand candidates are thresholded to give 75% recall on the training set. For every person a maximum of top two scoring hand bounding boxes are returned as the detected hands (zero or one hand may be returned if no image regions score above the threshold).

5.5.1 Results

5-fold cross-validation is performed using the training and validation set of the PASCAL VOC 2011 person layout dataset (Section 3.6). The results are reported in the mean and standard deviation of AP.

Table 5.4 shows the improvement in AP we achieve after rescoring the head and hand candidates. It shows that the modelling of dependencies between the parts improves detection performance. This can be thought of as a (directed) graphical model that encodes the dependencies between parts. Table 5.4 also shows the results of Yang and Ramanan (2011)’s method on the same images. Yang and Ramanan’s method is currently state-of-the-art for person layout detection. They evaluate the detections using the stickman for the person layout (i.e., one line segment indicating

location, size and orientation for each part). To evaluate their detections using the PASCAL criteria, we estimate the bounding boxes from the stickman. The height is given by the length of the stickman line segments and the width is fixed by cross-validation. The hand bounding box is detected by extrapolating along the orientation of the arm from the detected wrist end-point. The poor performance of Yang and Ramanan’s method shows that unmodified pictorial structure approaches do not successfully solve the current problem. The results of Yang and Ramanan are not intended for direct comparison, but are given to indicate the difficulty of the problem.

5.6 Conclusions

We have demonstrated that the proposed two stage hypothesise and classify method is capable of improving recall and precision over state-of-the-art results. The results improved further after incorporating local position and scale attributes. In the next chapter we will present our linear time algorithm of constraint generation for structured output SVM ranking. We will use this in Chapter 7 with head and hand detections from the current chapter to improve the human layout estimation.

Chapter 6

Efficient structured output ranking SVM

In Section 2.6, we gave an overview of structured output ranking SVM. To summarise, structured output *ranking* enforces constraints between pairs of structured output predictions such that better predictions (i.e., having lower loss values) are scored higher than worse ones (having higher loss values). Thus, the number of constraints can be quadratic in the number of predictions.

Although the objective function can contain a number of constraints quadratic in the number of structured output predictions, many vision applications have loss functions that have a fixed number of discrete values, L . For instance, the PASCAL VOC detection overlap score (Everingham et al. (2010a)) is a continuous value lying between 0 and 1. However, if it is rounded to tenths of a decimal then we get 11 discrete values. As will be shown, a limited number of discrete loss values also arise in parts-based models with thresholded loss (Section 7.1) and taxonomic multiclass prediction (Section 7.2). Having a small number of loss values enables us

to adapt the linear time algorithm of [Joachims \(2006\)](#) (for constraint generation in cutting plane optimisation algorithm of ordinal regression) to the structured output setting, resulting in linear time complexity to optimise over a quadratic number of constraints. In Section 6.1 we give the procedure for training the structured output ranking SVM objective in linear time when there are a small number of discrete loss values. In Section 6.2 and 6.3, we give algorithms for slack and margin rescaling formulations respectively developed using the technique explained in Section 6.1.

6.1 Linear time constraint generation

The objective function of structured output ranking simply optimises a scaled loss for all training sample pairs:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i,j} \xi_{ij} \right\}, \quad \text{s.t., } \xi_{ij} \geq 0, \quad (6.1)$$

$$w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) \geq 1 - \frac{\xi_{ij}}{\Delta_j - \Delta_i} \quad (6.2)$$

$$\text{or, } w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) \geq (\Delta_j - \Delta_i) - \xi_{ij} \quad (6.3)$$

where $(i, j) \in \mathcal{P}$ denotes the ordered indices of training samples such that the structured output loss, Δ_i , of sample i is less than the loss, Δ_j , of sample j . Thus, $\Delta_i < \Delta_j$ if $i < j$. As discussed in Section 2.6, Equation (6.2) is referred to as slack rescaling and Equation (6.3) as the margin rescaling.

The above formulation has $m \in \mathcal{O}(n^2)$ number of constraints and slack variables, where n is the number of training samples. We propose an equivalent 1-slack formulation ([Joachims \(2006\)](#)) of structured output ranking that uses a single slack

variable $\xi = \sum_{(i,j) \in \mathcal{P}} \xi_{ij}$ resulting in the cutting plane optimisation problem:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C\xi \right\}, \quad \text{s.t.}, \quad \xi \geq 0, \quad (6.4)$$

$$\frac{1}{m} \sum_{(i,j) \in \mathcal{P}} c_{ij} (w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)) \geq \frac{1}{m} \sum_{(i,j) \in \mathcal{P}} c_{ij} - \frac{\xi}{\Delta_j - \Delta_i} \quad (6.5)$$

$$\text{or, } \frac{1}{m} \sum_{(i,j) \in \mathcal{P}} c_{ij} (w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)) \geq \frac{1}{m} \sum_{(i,j) \in \mathcal{P}} c_{ij} (\Delta_j - \Delta_i) - \xi \quad (6.6)$$

Here $c_{ij} \in \{0, 1\}$ is the indicator variable stating whether the constraint between samples i and j has been included in the summation. The indicator c_{ij} is 1 if the corresponding constraint in Equation (6.5) or Equation (6.6) is violated, and zero otherwise, i.e., $c_{ij} = |(\Delta_i < \Delta_j) \wedge ((w^T \phi(x_i, y_i)) - (w^T \phi(x_j, y_j))) < 1|$ for slack rescaling and for margin rescaling, $c_{ij} = |(\Delta_i < \Delta_j) \wedge ((w^T \phi(x_i, y_i)) - (w^T \phi(x_j, y_j))) < (\Delta_j - \Delta_i)|$. Thus, a constraint is violated when the training samples are scored such that the difference in the scores of lower loss sample and higher loss sample is less than the margin. While both slack and margin rescaled formulations have 2^n constraints (one for each possible value of c_{ij}), this formulation has only one slack variable ξ shared across all constraints. Each constraint in this form corresponds to the sum of a sub-set of constraints from formulations given by Equation (6.2) and Equation (6.3) selected by c_{ij} .

Cutting plane optimisation of Equation (6.5) and Equation (6.6) consists of alternating between optimising the objective with a fixed set of constraints, and finding violated constraints of the current function estimate. The core of the learning algorithm is to determine, for a given w , the most violated constraints.

For a given loss value l , all the violated constraints for pairs (i, j) such that $\Delta_i = l$ can be obtained in linear time using the technique mentioned in [Joachims \(2006\)](#).

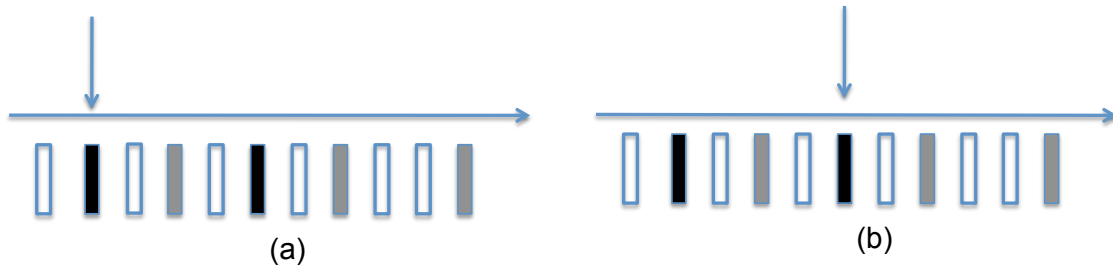


Figure 6.1: **Linear time constraint generation method.** All the data points are ordered in decreasing compatibility score (shown by the horizontal arrow sign). The vertical arrow represents the current position of the scan through the data. Gray bars are all the data-points having the given loss value l . Black bars are the data-points which are having loss $l' > l$. For simplicity of the explanation, the margin is considered to be zero. In this case, a constraint is violated when a black bar is placed earlier than a gray bar in the sorted list. This is because the points are ordered in decreasing compatibility score and a constraint is violated when a higher loss data-point is assigned higher compatibility score. In (a), the first black bar will violate constraints with the three subsequent gray bars and in (b), the second black bar will violate constraints with the two remaining gray bars. If an account is maintained of all the subsequent gray bars in the list, then all the violated constraints (i, j) having $\Delta_i = l$ can be obtained in a single scan through the data.

To do so, first the training samples are sorted in terms of decreasing compatibility scores ($w^T \phi(x_i, y_i)$). If a sample j with $\Delta_j > l$ is scored such that difference between the scores of samples i and j is less than the margin (i.e., it is violated), it will also violate all the subsequent samples, i' with $\Delta_{i'} = l$ in the sorted list (i.e., having $w^T \phi(x_{i'}, y_{i'}) < w^T \phi(x_i, y_i)$). Thus by book-keeping all the samples having loss value l , all the violated constraints for pairs (i, j) having $\Delta_i = l$ can be found in one pass through the training data. Figure 6.1 illustrates the method. If the number of possible loss values is finite, then this gives a linear time solution for generating all the violated constraints (by going through each of the loss values in this manner). The complete learning algorithm is summarised in Algorithms 4 and 5. Source code of a reference implementation is available from our website.

The linear time algorithm for ordinal regression proposed in Joachims (2006) is a special case of our method. Plugging $\phi(x_i, y_i) = x_i$ and replacing $\Delta_j - \Delta_i$ with 1, our method reduces to ordinal regression.

6.2 Slack rescaling formulation

Algorithm 4 describes the algorithm for the slack rescaling formulation of structured output ranking. In the algorithm, we have used variables c_i^+ and c_i^- which are the weighted counts of violations in which the i^{th} sample occurs with positive sign (i.e. $c_{ij} = 1$) and negative sign (i.e. $c_{ji} = 1$) respectively. In slack rescaling, these weights are related to c_{ij} by $\frac{1}{m} \sum_{(i,j) \in \mathcal{P}} (\Delta_j - \Delta_i) c_{ij} = \frac{1}{2m} \sum_{i=1}^n (c_i^+ + c_i^-)$. In each iteration, the algorithm computes optimum over current working set W in line 6. In lines 9–24, it finds all the violated constraints and adds them to the current working set W in line 25. Unlike ordinal regression, the variables c_i^+ and c_i^- are scaled by

Algorithm 4 1-slack optimisation for structured output ranking with slack rescaling.

```

1: Input:  $S = ((\phi(x_1, y_1), \Delta_1), \dots, (\phi(x_n, y_n), \Delta_n)), C, \epsilon$ 
2:  $L = (\Delta_1, \Delta_2, \dots, \Delta_n)$ 
3: sort  $L$  in decreasing order
4:  $W \leftarrow \emptyset$ 
5: repeat
6:    $(w, \xi) \leftarrow \arg \min_{w, \xi \geq 0} \{\frac{1}{2}w^T w + C\xi\}$ 
    $s.t. \forall (c^+, c^-) \in W : \frac{1}{m}w^T \sum_{i=1}^n (c_i^+ - c_i^-)\phi(x_i, y_i) \geq \frac{1}{2m} \sum_{i=1}^n (c_i^+ + c_i^-) - \xi$ 
7:   sort  $S$  by decreasing  $w^T \phi(x_i, y_i)$ 
8:    $c^+ \leftarrow 0; c^- \leftarrow 0$ 
9:   for  $l = l_2, \dots, l_{|L|}$  do
10:     $n_l \leftarrow$  number of examples with  $\Delta_i = l$ 
11:     $i \leftarrow 1; j \leftarrow 1; a \leftarrow 0; b \leftarrow 0; d \leftarrow 0$ 
12:    while  $i \leq n$  do
13:      if  $\Delta_i == l$  then
14:        while  $(j \leq n) \wedge (w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)) < 1$  do
15:          if  $\Delta_j > l$  then
16:             $b++; d \leftarrow d + \Delta_j; c_j^- \leftarrow c_j^- + (n_l - a)(\Delta_j - \Delta_i)$ 
17:          end if
18:           $j++$ 
19:        end while
20:         $a++; c_i^+ \leftarrow c_i^+ + d - b\Delta_i$ 
21:      end if
22:       $i++$ 
23:    end while
24:  end for
25:   $W \leftarrow W \cup \{(c^+, c^-)\}$ 
26: until  $\frac{1}{2m} \sum_{i=1}^n (c_i^+ + c_i^-) - \frac{1}{m} \sum_{i=1}^n (c_i^+ - c_i^-)(w^T \phi(x_i, y_i)) \leq \xi + \epsilon$ 
27: return  $(w, \xi)$ 

```

the loss values, which results in the scaling of hinge loss of the objective function with the difference in loss values of samples. The algorithm iterates from the second highest loss value since no constraint could be violated for the ordered pairs (i, j) when Δ_i is equal to the maximum loss, as to violate a constraint Δ_i has to be lower than Δ_j . As long as the number of loss values, $|L|$, is independent of the number of samples, computation of the outer loop (lines 9–24) is therefore also linear in the number of samples.

6.3 Margin rescaling formulation

The algorithm for margin rescaling formulation of structured output ranking SVM is given in Algorithm 5. In the algorithm, c_i^+ and c_i^- are the weighted counts of violations for i^{th} sample occurring with positive sign (i.e. $c_{ij} = 1$) and negative sign (i.e. $c_{ji} = 1$) respectively. These weights are related to c_{ij} by $\frac{1}{m} \sum_{i=1}^m c_{ij} = \frac{1}{2m} \sum_{(i,j) \in \mathcal{P}} (c_i^+ + c_i^-)$. The optimum over the current working set W is computed in line 6. All the violated constraints are obtained in lines 9–37. They are added to the current working set W in line 38. The condition in line 18 ensures that the margin is rescaled as per difference in the loss values of the training samples.

Experimentally we found that the margin rescaled version gives performance comparable to slack rescaling with the later performing slightly better. The comparison of margin rescaling and slack rescaling variants of structured output ranking for person layout task is given in Section 7.1. They are compared for taxonomic multiclass prediction in Section 7.2.

Time Analysis. For the run-time comparison, we trained our algorithm over the combined training and validation set of the PASCAL VOC 2011 person layout data-

Algorithm 5 1-slack optimisation for structured output ranking with margin rescaling.

```

1: Input:  $S = ((\phi(x_1, y_1), \Delta_1), \dots, (\phi(x_n, y_n), \Delta_n)), C, \epsilon$ 
2:  $L = (\Delta_1, \Delta_2, \dots, \Delta_n)$ 
3: sort  $L$  in decreasing order
4:  $W \leftarrow \emptyset$ 
5: repeat
6:    $(w, \xi) \leftarrow \arg \min_{w, \xi \geq 0} \{\frac{1}{2}w^T w + C\xi\}$ 
    $s.t. \forall (c^+, c^-) \in W : \frac{1}{m}w^T \sum_{i=1}^n (c_i^+ - c_i^-)\phi(x_i, y_i) \geq \frac{1}{m} \sum_{i=1}^n (c_i^- - c_i^+)\Delta_i - \xi$ 
7:   sort  $S$  by decreasing  $w^T \phi(x_i, y_i)$ 
8:    $c^+ \leftarrow 0; c^- \leftarrow 0$ 
9:   for  $l = l_2, \dots, l_{|L|}$  do
10:      $n_l \leftarrow$  number of examples with  $\Delta_i = l$ 
11:      $i \leftarrow 1; j \leftarrow 1; a \leftarrow 0; b \leftarrow 0; k \leftarrow -1$ 
12:     while  $i \leq n$  do
13:       if  $k > -1$  then
14:          $j \leftarrow k; k \leftarrow -1;$ 
15:       end if
16:       if  $\Delta_i == l$  then
17:         while  $(j \leq n) \wedge (w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)) < \Delta_j - l_{|L|}$  do
18:           if  $(w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) < \Delta_j - \Delta_i)$  then
19:             if  $\Delta_j > l$  then
20:               if  $k < 0$  then
21:                  $b ++; c_j^- \leftarrow c_j^- + (n_l - a)$ 
22:               else
23:                  $c_j^- \leftarrow c_j^- + 1; c_i^+ \leftarrow c_i^+ + 1$ 
24:               end if
25:             end if
26:           else
27:             if  $k < 0$  then
28:                $k \leftarrow j$ 
29:             end if
30:           end if
31:            $j ++$ 
32:         end while
33:          $a ++; c_i^+ \leftarrow c_i^+ + b$ 
34:       end if
35:        $i ++$ 
36:     end while
37:   end for
38:    $W \leftarrow W \cup \{(c^+, c^-)\}$ 
39: until  $\frac{1}{m} \sum_{i=1}^n (c_i^- - c_i^+)\Delta_i - \frac{1}{m} \sum_{i=1}^n (c_i^+ - c_i^-)(w^T \phi(x_i, y_i)) \leq \xi + \epsilon$ 
40: return  $(w, \xi)$ 

```

set (Section 3.6) under the settings described in Section 7.1. The dataset contains 850 data-points and the dimensionality of each feature vector is 15. Our method takes on average about $1/80^{th}$ of a second for slack rescaling and about $1/70^{th}$ of a second for margin rescaling formulation per cutting plane iteration for a linear kernel training. A normal $\mathcal{O}(n^2)$ algorithm for structured SVM ranking for the same kernel took around 0.5 seconds. The experiment was performed on a standard single core Intel machine with 2.8 GHz CPU.

6.4 Conclusions

The efficient linear time training algorithm for structured output ranking is applicable whenever the number of loss values is small and independent of the number of training samples. This is applicable in a large variety of practical problems. We will demonstrate in the next chapter this to be the case for person layout and taxonomic multiclass prediction. In person layout, the underlying combinatorics of the loss function means the number of loss values is linear in the number of parts for unoccluded objects, and quadratic in the number of parts when occlusion may be present. Objects are typically described by their decomposition into only a handful of parts, and the objective is consequently tractable. For taxonomic multiclass prediction, a non-degenerate taxonomy results in a number of taxonomic losses at most logarithmic in the number of classes as the number of loss values is bounded by the depth of the tree. As such diverse applications as scene layout, object layout, multi-class and multi-label prediction are characterised by small numbers of loss values, and also methods that rank learning from continuous loss values such as [Li et al. \(2010\)](#) can simply be discretized into a small number of losses, we expect that the

methods proposed here will find wide application across learning based computer vision.

Chapter 7

Person layout and Taxonomic multiclass prediction

In the last chapter, we proposed a method for efficiently solving structured output ranking objective. In this chapter, we demonstrate the applicability of the linear time algorithm to two quite disparate tasks: person layout and taxonomic multiclass prediction.

For person layout, any body pose induces structure in the part configuration space, which governs the relative positioning of body parts. In taxonomic multiclass prediction, the structure is induced by the class hierarchy. Thus, the output space of both the problems is structured and can be solved with structured output SVM (Section 2.6.2). Furthermore, both person layout and taxonomic multiclass prediction exhibit an inherent order among the different instances. In person layout, instances having more correctly predicted parts are ranked higher. In taxonomic multiclass prediction the class taxonomy defines the ordering of class objects. We show that these problems benefit from the use of structured output ranking, which

improves over the structured output SVM when predictions involve ranking results (Blaschko et al. (2010); Rahtu et al. (2011); Zhang et al. (2011)).

In person layout, as described in Section 3.6, the task is to detect interrelated parts, specifically the head, hands and feet belonging to the same person. The natural loss underlying this problem can be computed from the fraction of incorrectly predicted parts. Thus, this loss has a discrete number of values, and in this case is linear in the number of parts. Since the number of parts in a model is typically much smaller than the number of training samples, our approach is highly efficient in this setting.

Taxonomic multiclass prediction involves retrieval of images as per a defined taxonomy. To do this, we employ a standard taxonomic loss that takes into account the path length in the taxonomic tree between predictions, resulting in a number of loss values logarithmic in the number of classes to be predicted (Binder et al. (2011)). This allows us to apply our linear time algorithm for the problem. Taxonomic multiclass prediction is especially important in computer vision when scaling classification and detection to large numbers of categories (Deng et al. (2010)).

We describe our method of layout estimation using structured output ranking in Section 7.1. We also present our results for the PASCAL VOC 2011 person layout competition (Everingham et al. (2011)). In Section 7.2, we explain our procedure of taxonomic multiclass prediction along with various experiments. In both cases we show that learning with structured output ranking consistently surpasses the predictive performance of the SVM, structured output SVM, ordinal regression and other related methods.

7.1 Person layout

The person layout problem of the PASCAL VOC challenge (Everingham et al. (2011)), is described in Section 3.6. It is a very difficult problem having images of people from different viewpoints in different body configurations. The detection results for a part is evaluated using AP computed from the PR curve. The mean AP across all the parts is considered as the AP for the person. A submission is evaluated by the AP for individual parts and also by its summary for the person. Therefore, in order to win the competition, it is imperative to perform well on each of the individual parts categories.

Current layout detection methods tend to model the human body as a pictorial structure, and to predict the pose/layout of the person by maximising the posterior of the joint configuration of the body parts (Eichner and Ferrari (2009); Yang and Ramanan (2011)). However, these methods suffer from a common curse of pictorial structures, the inability to model occlusion of parts and the over-counting of confidence from a given pixel location. In our experiments, we show that unmodified pictorial structure models are not suited to this setting.

Another approach to solve the person layout problem could be to apply individual part detectors and then combine their outputs in some fashion. There are two caveats here: (i) the task needs the confidence score for the whole layout, not the individual parts; (ii) detection of parts is performed independently, which may lead to sub-optimal performance unless some kind of co-occurrence information is introduced into the framework.

The method we implement here proceeds in two stages, (i) part candidates are generated using individual detectors, (ii) candidates for individual parts are com-

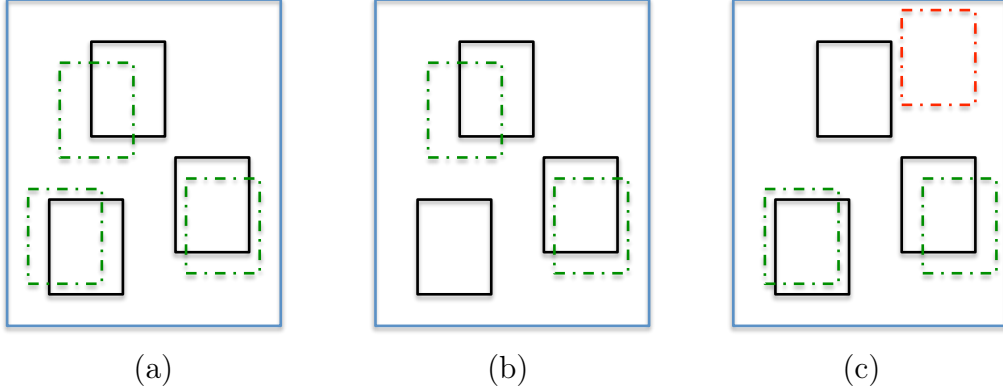


Figure 7.1: **Different scenarios of loss values for human layout.** In the example, there are three body parts and their location is marked with black solid rectangles. The detected bounding boxes are shown as dashed rectangles, green if it is true-positive, red otherwise. In case (a) and (b), all the detected body parts are correct. Hence, the precision is 1 and the loss value is $1 - 1 = 0$. For case (c), the loss value is $1 - 2/3 = 1/3$.

bined and the joint output space is optimised and ranked using the structured output ranking approach. We note that feet are not detected in the current experiments.

For the following experiments, 5-fold cross-validation is performed using the training and validation set of the PASCAL VOC 2011 (Section 3.6) person layout dataset. The results are reported as the mean AP and standard deviation.

7.1.1 Part candidates generation

The candidates for head and hand are generated using individual part detectors explained in Section 5.5. The output of both the detectors is a bounding box around the part (referred to as ROI hereafter) with a confidence value.

7.1.2 Joint learning using structured ranking

The head and hand candidates obtained from the method described above have individual confidence scores. For the person layout, a single confidence score is needed. We obtain this score from a ranking function which is learned by jointly optimising the output spaces of the two parts using structured output ranking. We define the structured output loss value (Δ) for a human layout as: $1 - \text{precision}$. This defines a limited number of loss values corresponding to the fraction of hypothesised part detections that are incorrect. We obtain a substantial improvement over a normal classifier using this technique.

Figure 7.1 demonstrates three different scenarios of loss values for human layout. In case (a), all the three detected body parts are correct (or precision is 1), therefore the loss value is $1 - 1 = 0$. For case (b), loss value is again 0. For case (c), loss value is $1 - 2/3 = 1/3$. Both the cases (a) and (b) have only true positive detections, however the recall for (b) is less than for (a). Both cases (a) and (b) are ranked equally and higher than case (c) using our loss value. This makes sense for a retrieval setting which requires more precise results to be returned earlier irrespective of the individual recall.

We also compare our results with naïve techniques for combining the confidences of different parts. However, they do not optimise the AP for all parts jointly and, as we will show, tend to benefit one of the parts at the expense of others.

7.1.2.1 Implementation details

The ranking objective function is trained on the feature vector formed by concatenating the features used earlier for rescoreing of parts (i.e., by concatenating the feature vectors of head and all the hands). Figure 7.2 shows the procedure followed

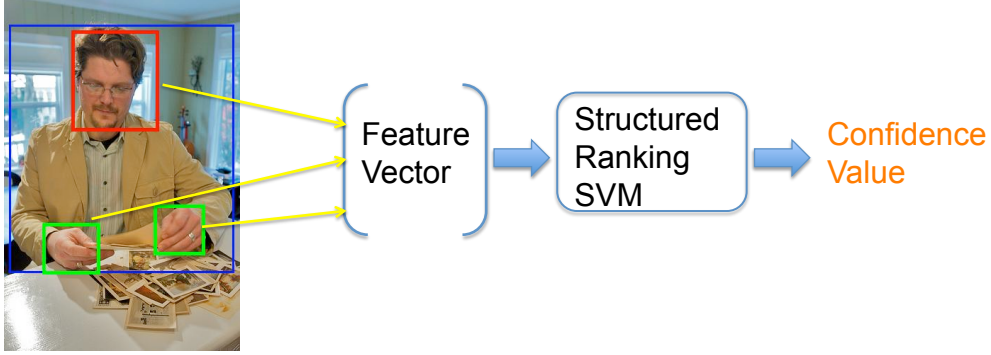


Figure 7.2: Overview of the human layout system.

Ordering by	Head AP%	Hand AP%	Mean
Head score	81.04±1.77	20.98±2.21	51.01±1.23
Max-hand score	74.95±3.16	26.16±2.49	50.55±1.55
Mean-hand score	75.61±2.33	28.32±2.12	51.96±1.15
Mean head, hands	79.77±2.20	22.90±2.27	51.33±1.10
Max head, hands	79.49±2.00	21.53±2.30	50.51±1.25

Table 7.1: **AP for different scoring methods.** The confidence score for the person layout prediction is computed by combining parts scores in different ways. The last column is the mean of head and hand APs, which is the metric to be maximised. Experiments were performed on training and validation set of the PASCAL VOC 2011 layout dataset.

to get the confidence values for the human ROIs.

For the experiments, both linear and non-linear kernels were explored in learning the ranking function. The C parameter of the learning objectives, and the σ parameter of the Gaussian RBF kernel are optimised by a validation step. For this, models were trained on the training set of the PASCAL VOC 2011 person layout dataset (Section 3.6) and tested on the validation set of the same dataset.

7.1.2.2 Results

Table 7.1 shows the inherent tradeoff between the confidences of various body parts. While naïve combination schemes can be employed to compute a joint detection

Method	Head AP%	Hand AP%	Mean
Binary SVM linear	73.92±3.15	20.29±1.76	47.10±1.87
Rank margin linear	79.37±2.74	27.78±1.73	53.57±1.33
Rank slack linear	79.32±2.77	27.88±1.75	53.60±1.28
Rank margin RBF	79.56±2.69	28.10±1.92	53.83±1.11
Rank slack RBF	79.55±2.88	28.22±2.25	53.89±1.29

Table 7.2: **AP scores resulting from different learning techniques.** The last four rows are different variants of the proposed method. They differ only slightly, but improve substantially over the SVM. The dataset used for the experiments was training and validation set of the PASCAL VOC 2011 layout dataset.

Method	Head AP%	Hand AP%	Mean
Our Method	72.85	26.70	49.78
BCNPCL_HumanLayout	74.4	3.3	38.85
OXFORD_SBD	52.7	10.4	31.55

Table 7.3: **AP for PASCAL VOC 2010 person layout test dataset.** We train our method on the training and validation portion of the PASCAL VOC 2010 layout dataset. The evaluation was computed on the competition server. The results for the other methods are as reported on the competition website ([PASCAL VOC 2010 results](#)). We trained our model using slack rescaling formulation with RBF kernel.

score based on the individual part scores, such a strategy will likely choose a sub-optimal score with respect to a combined loss measure. By directly optimising a loss computed over the combination of parts, the structured output ranking objective can better balance the performance of each part detection in scoring the entire person layout. It is only when the detection performance of each part is balanced that the best layout predictions are ranked highest. Table 7.2 shows the improved performance when an appropriate joint scoring objective is applied. We see that the ranking objectives (c.f. Chapter 6) perform substantially better than a binary SVM (first row) and improves the AP for both head and hands.

Table 7.3 compares our results with the best performing results in the PASCAL VOC 2010 layout detection competition. We achieve a substantial improvement over



Figure 7.3: Sample high ranked single person results of person layout detection task for the PASCAL VOC 2010 test dataset. The blue rectangle represents the provided bounding box of the person, green rectangles are the detected hands and red rectangle is the detected head respectively. Our method yields good results despite high variation in pose and occlusion.

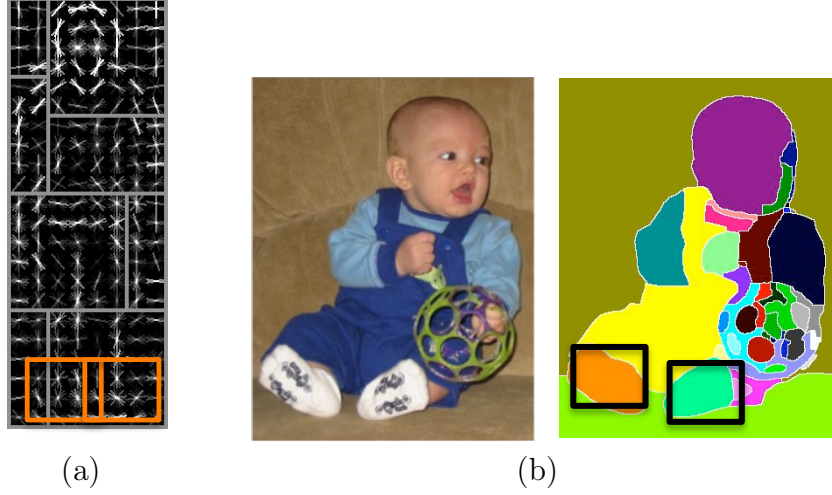


Figure 7.4: **Methods of foot detection.** (a) Foot parts of Felzenszwalb et al. (2010b)’s human detector. (b) Blob shaped super-pixels in the lower quarter of the human ROI.

the winning entries using our method. For the comparison, we trained the model on training and validation set of the PASCAL VOC 2010 layout dataset (Section 3.6). Figure 7.3 shows some sample results for person layout.

Thus, the person layout experiments show both the effects of the prediction architecture and feature design, as well as the benefits of using a structured output ranking formulation.

7.1.3 PASCAL VOC 2011 results

We used our method to participate in the PASCAL VOC 2011 person layout competition. Person layout is one of the most difficult problems in computer vision and as we show earlier, there is no other method that can perform this task accurately. Using the technique that we developed, we achieved good results in the competition.

In our method described earlier, feet were not detected. However, for the challenge, we tried to detect feet in an ad-hoc manner. To do so, we used the following

two different methods:

(i) **Human detector of Felzenszwalb et al. (2010b)** . The human detector of Felzenszwalb et al. (2010b) is a parts-based mixture model having three components. Each of the components model different aspect-ratios of the human body and one of these components models the complete human figure. Two of the parts of this component correspond to the foot which are used by us as foot detectors (Figure 7.4(a)).

(ii) **Super-pixel segmentation.** Blob shaped super-pixels in the lower quarter of the human ROI are treated as foot detections (Figure 7.4(b)). The super-pixels are obtained using the code available from Arbelaez et al. (2011), then they are analysed for their shape measurements, such as: eccentricity, convex area etc. If the shape measurements of a super-pixel are within a range, then its bounding box is returned as the foot detection. The range for the shape measurements is learned over the validation set of the competition.

Training. We used slack-rescaled formulation of SVM ranking with RBF kernels and trained it over the combined training and validation set of the competition. Since the foot was detected in an ad-hoc manner, it does not contribute towards the ranking and the model is trained by concatenating feature vectors of head and all the hands (as shown in Figure 7.2). The structured output loss value is defined as 1-precision (similar to experiments in Section 7.1) and again to compute the precision, foot detection is not considered.



Figure 7.5: Sample results for the PASCAL VOC 2011 person layout competition. Blue and black rectangles are the given human ROIs in the image. Blue rectangle is the human ROI for which layout prediction is done. Red rectangle is the head, green and yellow rectangles are hand and foot respectively.

	Head	Hand	Foot
AP	72.9	26.9	4.1

Table 7.4: Performance in the PASCAL VOC 2011 person layout competition available from the competition website ([PASCAL VOC 2011 results](#)).

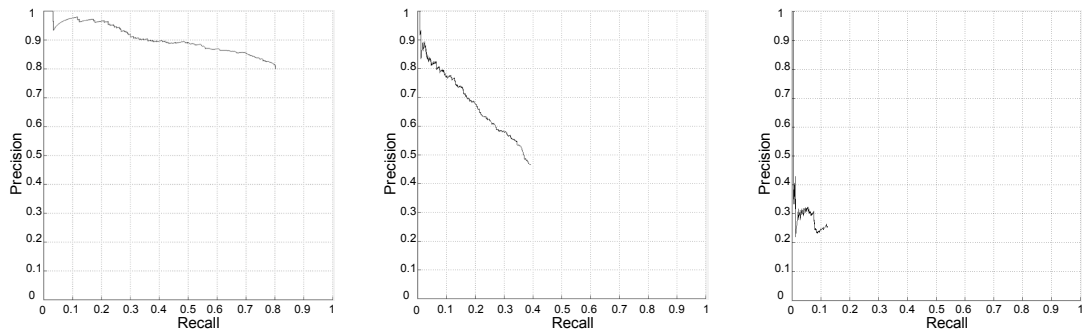


Figure 7.6: Precision/Recall curves for the individual parts. (a) Head (b) Hand (c) Foot.

Results. Figure 7.5 shows some qualitative results for our method. Table 7.4 gives the performance of our method in the competition. Ours was the only entry for the competition (which shows difficulty of the task) and it out-performs all the previous year’s entries. PR curves for the different human body parts are shown in Figure 7.6.

7.2 Taxonomic multiclass prediction

Taxonomic multiclass prediction involves multiclass prediction using class hierarchies (otherwise known as taxonomies). Standard approaches to multi-class image labelling typically penalise incorrect predictions equally: a motorbike misclassified as a bicycle receives the same penalty as a motorbike misclassified as a cow. If taxonomic knowledge is included, a system can be designed to penalise mis-classifications proportional to their taxonomic losses (e.g. the distance along a taxonomic tree, so that for instance misclassifying a car as a van might be better than misclassifying it as a banana). In such a setting, vision systems could learn to associate appearances of images and high-level semantic knowledge such as user tags with greater generalisation ability. Taxonomic multiclass prediction is performed on hierarchical models, therefore it can provide computational benefits at test time by employing a detection strategy logarithmic in the number of classes (Bengio et al. (2010)). We show here that these computational benefits may be extended to the training of such objective functions.

In this work, class hierarchies are learned in a ranking setting. This is done by using the taxonomic loss as the Δ value in the structured output ranking objective. We define the taxonomic loss as the minimum path length between the two classes

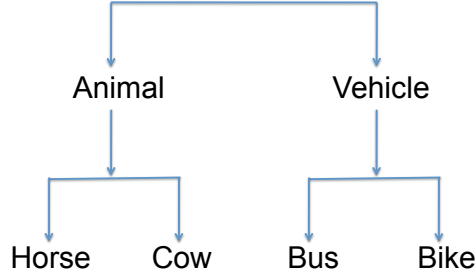


Figure 7.7: **A sample taxonomy.** For the task of taxonomic multiclass prediction, the loss involved in misclassifying ‘cow’ with ‘horse’ is less than misclassifying it as ‘bus’ or ‘bike’.

in the taxonomic graph. A class nearer to the reference class in the taxonomic graph is ranked higher. For example, if the taxonomy is as shown in Figure 7.7, then the loss for misclassifying a ‘cow’ with ‘horse’ will be 2 whereas the loss for mis-classification between ‘cow’ and ‘bike’ will be 4. Thus, the number of possible loss values is $\mathcal{O}(\log c)$, where c is the number of class labels. It should be noted that usually $c \ll n$, where n is the number of training samples.

The joint feature map, ϕ used in this work, is the standard one used in multi-class and taxonomic prediction (Binder et al. (2011); Cai and Hofmann (2004, 2007)). This is given by $\phi(x_i, y_i) = \lambda(y_i) \otimes x_i$. $\lambda(y_i) \in \mathbb{R}^c$ is the class attribute vector, which is defined as: $\lambda_j(y_i) = 1$, if $j = y_i$, zero otherwise. $x_i \in \mathbb{R}^d$ is the i^{th} image represented as a d -dimensional feature vector. Here $\otimes$ denotes a Kronecker product, thus $\phi(x_i, y_i) \in \mathbb{R}^{d \cdot c}$.

The ordinal constraints of Equation (6.5) and Equation (6.6) therefore enforce that the scores of the classes are ordered proportional to their distance in the taxonomic tree to a ground-truth class. Figure 7.8 demonstrates the difference between one vs. all classification approach and structured learning approach with joint feature map.

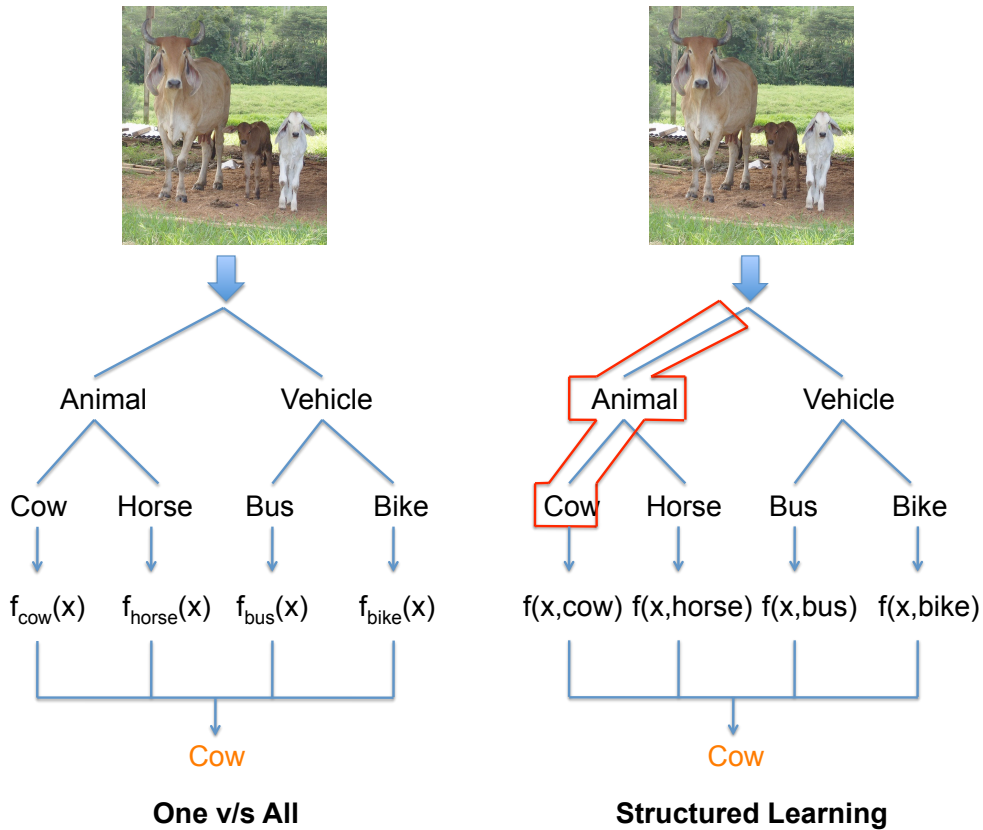


Figure 7.8: Difference between one vs. all classifier minimising 0/1 loss and structured learning classifier having joint feature map, minimising taxonomic loss. In one vs. all method, internal nodes of the taxonomy are ignored and the test image is classified as the class whose classifier assigns it the highest confidence value. All the classifiers for different classes are trained independently and there is no similarity measure is introduced into the system. In structured learning approach, the whole path is considered and the margin is maximised between the correct and wrong paths in training. The margin is decided by the number of edges that are not common in correct and wrong paths.

Evaluation Measure. The performance is evaluated using the cumulative taxonomic loss (Binder et al. (2011)). This is computed by accumulating the taxonomic loss over the top scoring n classes for a given image, where $n \in [1, \dots, c]$. The results are reported as the mean cumulative taxonomic loss which is obtained by averaging the cumulative taxonomic loss for different ranks over all the test images. The ImageNet challenge also uses a hierarchical cost as an evaluation criteria.

In the following experiments, we perform taxonomic multiclass predictions for two publicly available datasets, and show that our approach reduces the taxonomic loss considerably, i.e. gives better semantic classification.

7.2.1 Experiments

We test how well structured ranking performs compared to other methods on the indoor scene dataset (Section 3.5) and the PASCAL VOC 2007 classification dataset (Section 3.6.1). For the PASCAL VOC 2007 classification dataset, we train our models using the training and validation set and test it on the test set following the competition protocol.

7.2.1.1 Implementation details

In all the experiments, slack rescaled versions of structured SVMs are used unless explicitly mentioned.

Image descriptors. SIFT descriptors (Lowe (2004)) are extracted with a spatial stride of 5 pixels, and at four scales, defined by setting the width of the SIFT spatial bins to 4, 6, 8 and 10 pixels respectively. For the indoor scene database experiments, the features are quantised into a visual vocabulary of size 1000. A 2-level spatial

pyramid model (Lazebnik et al. (2006)) is obtained by dividing the image in 1×1 and 2×2 grids, for a total of 5 regions. For the experiments on the VOC 2007 dataset, the visual vocabulary used is of size 10000, and the quantised descriptors are encoded using locality-constraint linear encoding (LLC) (Wang et al. (2010)). The spatial regions are obtained by dividing the image in 1×1 , 3×1 and 2×2 grids, for a total of 8 regions.

Learning Methods. SVM and ordinal regression (SVM-rank) classifiers are trained by minimising the 0/1 loss and are learned in a 1 vs. all fashion. Structured output SVM (SVM-struct) and structured output ranking (Struct rank) algorithms minimise the taxonomic loss and are modelled using the joint feature map. A validation step was employed to set the C parameter where cross-validation was done for (80%-20%) split of the training data for the indoor scene database. For the VOC 2007 dataset, cross-validation is performed over the training and validation set of the dataset. For all formulations linear kernels were used.

7.2.1.2 Results

Figure 7.9(a) and 7.9(c) show the plots of cumulative taxonomic loss for different learning methodologies. Table 7.5 gives the numbers for cumulative loss for the top 1, 5 and 10 ranks for the different methods. It can be seen that the cumulative taxonomic loss decreases for ranking algorithms and it reduces even further by using structured ranking algorithms. Figure 7.9(b) and 7.9(d) show the plots for difference in the cumulative taxonomic loss with the minimum theoretically obtainable loss which is generated by optimally ordering the returned classes by their loss with the ground-truth of the test set. It shows the improvement that is achieved by structured

Method	Obj Loss	Indoor			VOC 2007		
		1	5	10	1	5	10
SVM	0/1	2.75	16.18	33.51	2.83	17.09	39.84
SVM-struct	tax	3.71	18.50	36.97	3.98	20.71	45.19
SVM-rank	0/1	3.14	15.93	32.11	3.02	17.04	39.29
Struct rank slack	tax	2.62	14.75	30.29	2.54	15.07	36.07
Struct rank margin	tax	2.66	14.92	30.53	2.55	15.11	36.26
Min loss		0.00	8.00	18.00	0.00	11.70	33.35

Table 7.5: **Mean cumulative taxonomic loss at different ranks.** The cumulative loss is computed as explained in Section 7.2. ‘Obj loss’ is the loss that is minimised by the respective learning method, where ‘tax’ stands for the taxonomic loss. ‘Min loss’ is the theoretically possible minimum taxonomic loss which is obtained by optimally ordering the class labels. The minimum loss is zero for the top result as the taxonomic loss is zero for correctly labelling an image. For the top two ranked classes, the minimum cumulative loss is the lowest taxonomic loss of a second category. This measure differentiates methods that correctly order multiple predictions by their taxonomic loss. The mean cumulative taxonomic loss for the structured output ranking method is lowest among all the learning methods. There is a very little difference between the accuracy of slack and margin rescaling formulations, with former performing slightly better.

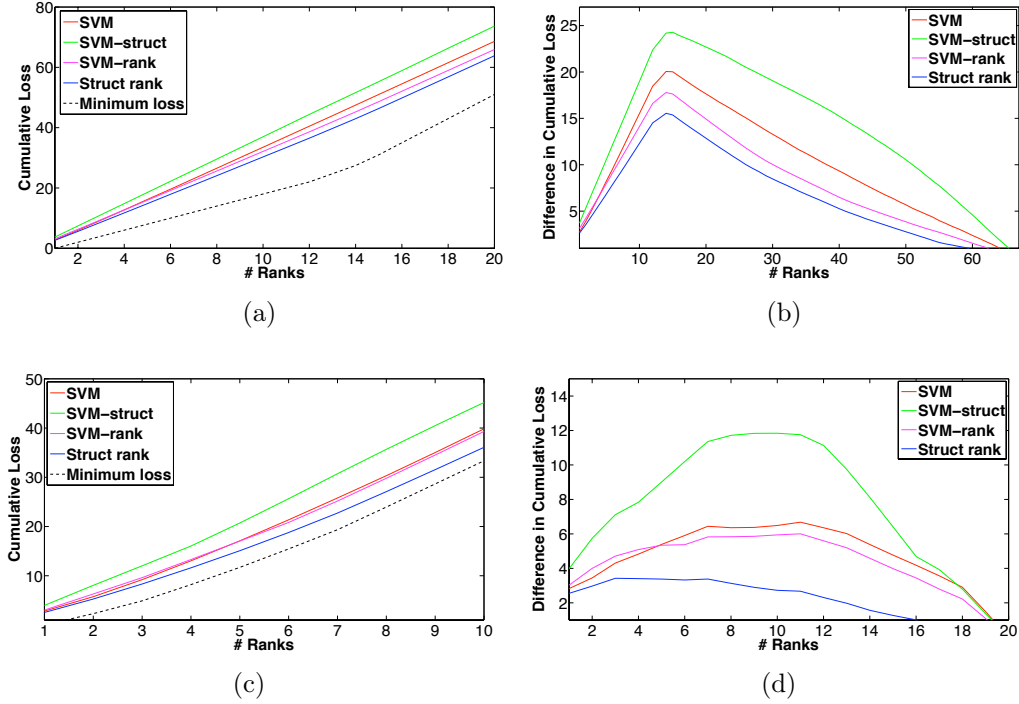


Figure 7.9: **Mean Cumulative Taxonomic loss for different learning methodologies.** (a) Mean Cumulative loss for the indoor scene database, (b) Difference in the mean cumulative losses with the minimum loss for indoor database for different learning techniques, (c) Mean Cumulative loss for the PASCAL VOC 2007 classification dataset, (d) Difference in the mean cumulative loss with the minimum loss for the PASCAL VOC 2007 classification dataset. Curves (b) and (d) are generated from (a) and (c) respectively and show the difference in the mean cumulative taxonomic losses with the minimum possible loss, which is obtained when the classes are optimally ordered. (b) and (d) make the improvement achieved by using structured ranking approach more visible. These graphs show that structured output ranking perform better than all other variations of SVM ranking and classification for the given task.

Indoor Scene			
			
SVM	airport inside, fastfood restaurant, mall	bathroom, elevator, cloister	deli, museum, tv-studio
SVM struct	hairsalon, auditorium, bowling	deli, florist, mall	church inside, florist, tv-studio
SVM rank	video-store, grocery-store, book-store	wine-cellar, corridor, children room	locker room, pool inside, prison cell
Struct rank	mall, grocery-store, book-store	elevator, waiting room, pool inside	museum, locker room, church inside

VOC 2007			
			
SVM	person, chair, tvmonitor	train, cow, bus	car, bus, tvmonitor
SVM struct	cat, car, bus	sheep, horse, cat	chair, sofa, sheep
SVM rank	sofa, chair, dining table	train, bus, bicycle	dining table, tvmonitor, potted plant
Struct rank	chair, sofa, potted plant	train, aeroplane, boat	person, aeroplane, tvmonitor

Figure 7.10: Sample results for the indoor scene and VOC 2007 classification test datasets. The top three labels (in order) predicted by each method are shown for several test images. More semantically meaningful labels are retrieved by structured ranking than the other methods.

learning of the ranking objective more clearly than in Figure 7.9(a) and 7.9(c). Some qualitative results are shown in Figure 7.10.

The performance is evaluated over taxonomic loss, therefore, an objective function minimising taxonomic loss is supposed to give the optimal performance. However, structured output SVM (labelled ‘SVM-struct’ in Table 7.5 and Figure 7.9) performs *worse* on these data sets than a binary SVM even when it optimises taxonomic loss. A SVM optimising ordinal regression with binary loss (labelled ‘SVM-rank’) tends to perform worse than a binary SVM when few labels are returned, but improves over the binary SVM when more labels are returned. However, structured output ranking dominates the performance of all other learning variants, i.e., binary SVM, structured output SVM and ordinal regression. This shows that structured output ranking is the optimal way of optimising taxonomic loss for the given scenario.

The mean cumulative taxonomic loss for the structured output ranking method is closer to the minimum loss for the VOC 2007 dataset than the indoor scene database. This may be because the VOC 2007 dataset has a deeper taxonomy.

7.3 Conclusions

The experiments in Section 7.1 and Section 7.2 lead to several broad conclusions. There is a consistent improvement of structured output ranking over ordinal regression and structured output SVMs. This indicates that structured output ranking is able to encapsulate the benefits of both approaches, leading to better overall performance. While we see a slight improvement of slack rescaling over margin rescaling, this difference does not appear to be substantial on this data. As the

training of structured output ranking is no more computationally expensive than that of binary SVMs or ordinal regression (c.f. Section 6.1), we believe that structured output ranking is appropriate to be applied in a wide variety of settings where those methods are currently employed.

In the next chapter, we will give the generalisation bounds of the structured output ranking algorithms and will present some observations over it.

Chapter 8

Generalisation bounds for structured output ranking

In supervised learning, a training algorithm learns a prediction function (g) such that for the training sample $S = ((x_1, y_1), \dots, (x_n, y_n))$, the output $g(x_i)$ is close to the given input value y_i . However, it is not always possible to learn a function which predicts all the output values correctly due to noise in training data, limited amount of training data and also because of limitations of the training algorithms. This property of training algorithms is characterised by *generalisation error*, which measures how far the prediction function is from the ideal function (the one that predicts all the output values correctly) over the entire set of possible input data. It is called *generalisation error* because it indicates the learning capacity of an algorithm from a finite set of samples. *Generalisation bound* refers to the upper bound on the generalisation error.

Study of generalisation properties of learning algorithms, with increase in the amount of training data, is a quintessential component of learning theory. [Vapnik](#)

and Chervonenkis (1971) were the first to derive the generalisation bound for classification algorithms. Since then, for every learning algorithm, generalisation properties have been studied.

Recently, ranking algorithms have become very popular in machine learning and are used for numerous applications in different fields. In Chapter 7, we also used structured output ranking for person layout and taxonomic multiclass classification. Despite this, the study of generalisation properties of ranking algorithms has been largely limited to the special cases of bipartite ranking (Agarwal and Niyogi (2005); Freund et al. (2003)). In this chapter, we will derive the generalisation bounds of a complete pairwise structured output ranking problem for both the slack and margin rescaled formulations and will present some of our observations about them. In Section 8.1, we will present some characteristics of the slack and margin rescaling formulations. In Section 8.2, generalisation bounds for both these formulations for structured output ranking will be presented.

8.1 Slack rescaling vs. Margin rescaling

The objective function for structured output ranking to regularised risk minimisation is given by:

$$\min_{w, \xi} \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i,j} \xi_{ij} \right\}, \quad \text{s.t., } \xi_{ij} \geq 0, \quad (8.1)$$

$$w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) \geq 1 - \frac{\xi_{ij}}{\Delta_j - \Delta_i} \quad (8.2)$$

$$\text{or, } w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j) \geq (\Delta_j - \Delta_i) - \xi_{ij} \quad (8.3)$$

where $(i, j) \in \mathcal{P}$ denotes the ordered indices of training samples such that the structured output loss, Δ_i , of sample i is less than the loss, Δ_j , of sample j .

As discussed in Section 2.6, Equation (8.2) is referred to as slack rescaling and Equation (8.3) as the margin rescaling. From the above equations, the ranking hinge losses for slack (ℓ_s) and margin (ℓ_m) rescaled formulations are given by:

$$\ell_s = (|\Delta_j - \Delta_i| - (w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)) \cdot \text{sign}(\Delta_j - \Delta_i))_+ \quad (8.4)$$

$$\ell_m = (|\Delta_j - \Delta_i| \cdot (1 - (w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)) \cdot \text{sign}(\Delta_j - \Delta_i)))_+ \quad (8.5)$$

where the output of the ranking algorithms is given by $w^T \phi(x_i, y_i)$.

Given that there are two seemingly natural piecewise linear convex upper bounds (slack and margin rescaled formulations) to the structured output loss, it is of interest to determine whether one is preferable to the other. We explore this here by considering in detail the constraints that result from two samples, i and j , where $\Delta_j - \Delta_i$ varies around zero. Figure 8.1 shows the behaviour of these two variants under such conditions. Margin rescaling has a discontinuity around zero and may overpenalize misorderings of samples with similar loss. Slack rescaling does not suffer from these problems, and vanishes when $\Delta_j - \Delta_i = 0$. This is an important characteristic of slack rescaled formulation, which favours towards its stability. Previous theoretical results from McAllester (2007) were inconclusive in the relative benefits of slack vs. margin rescaling in the structured output SVM setting. We will present theoretical results based on uniform convergence bounds in the following section.

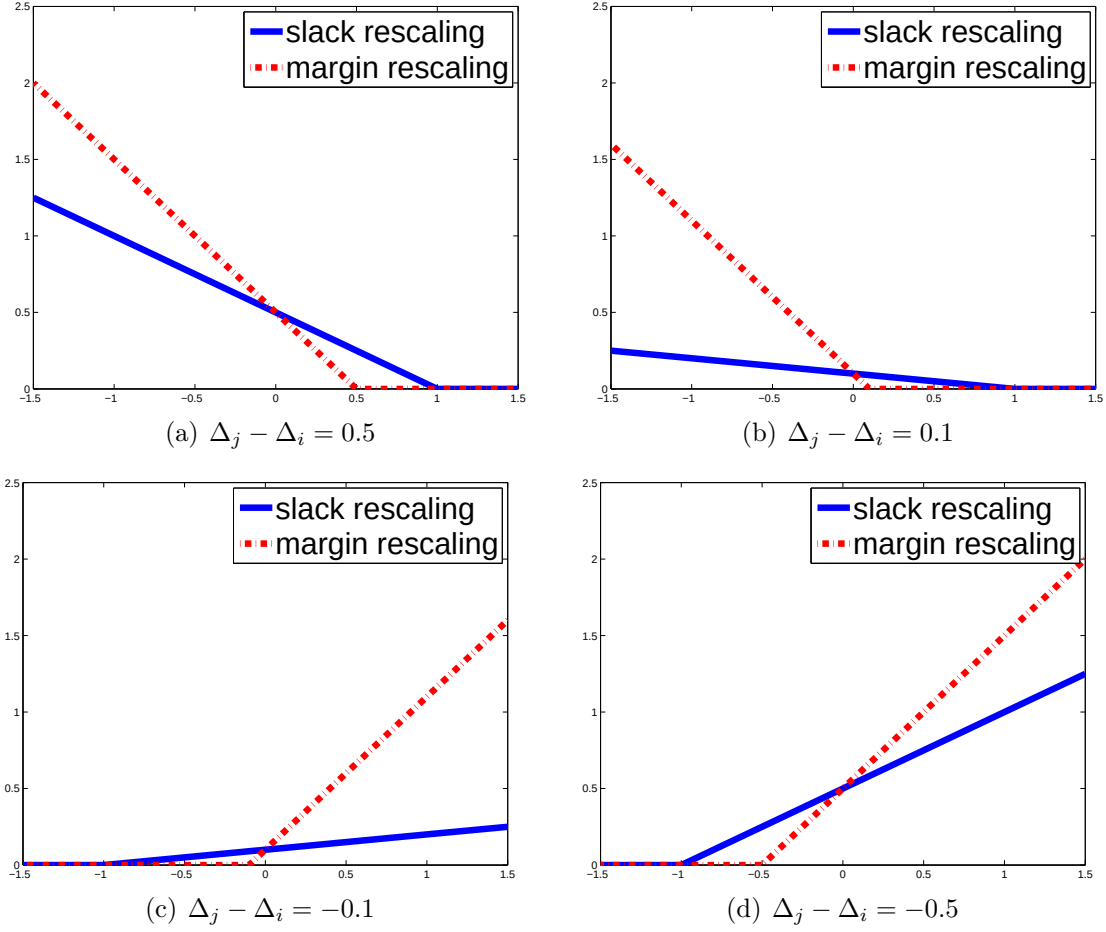


Figure 8.1: **Slack rescaling and margin rescaling as $\Delta_j - \Delta_i$ varies around zero.** The x axis represents $w^T \phi(x_i, y_i) - w^T \phi(x_j, y_j)$, while the y axis represents the convex bound to the step function, i.e., hinge loss. Margin rescaling (dashed red line) has a discontinuity around zero and may overpenalize misorderings of samples with similar loss. Slack rescaling (solid blue line) does not suffer from these problems, and vanishes when $\Delta_j - \Delta_i = 0$.

8.2 Generalisation bounds

Ranking approaches for training are relatively recent in the field of learning theory, and there are few works for analysing asymptotic behaviour of the ranking algorithms. [Cossock and Zhang \(2008\)](#) proved that if the ranking loss function between the data-points $\{(x_i, y_i), (x_j, y_j)\}$ is defined as $\ell = y_i - y_j$, where $y_i > y_j$, then there exists a consistent sub-set ranking formulation. [Duchi et al. \(2010\)](#) showed inconsistency of the slack rescaling variant of ranking, when ranking is defined by a noisy preference graph. This assumption of noisy labels is specific to their application and cannot be generalised to a standard structured output ranking problem.

[Agarwal and Niyogi \(2009\)](#) derived a generalisation bound for ranking algorithms for any kernel based method performing in reducible kernel Hilbert space ([Aronszajn \(1950\)](#)). Their setting explored a convex bound to a scaled step function that can be shown to be equivalent to the margin rescaling formulation presented here. Importantly, their results hold not only for bipartite ranking as in [Agarwal and Niyogi \(2005\)](#), but also are applicable for complete pairwise ranking (our case). We make use of their setting to additionally prove convergence bounds for slack rescaling.

We derive the generalisation bounds of structured output ranking algorithm using the notion of algorithmic stability. Algorithmic stability was first studied for learning algorithms by [Rogers and Wagner \(1978\)](#). It has been used to obtain generalisation bounds for classification and regression algorithms ([Bousquet and Elisseeff \(2002\)](#); [Kutin and Niyogi \(2002\)](#)). A stable algorithm is one whose output does not change significantly with the variation in input. Several different notions of stability have been used by different researchers in the study of classification and regression algorithms ([Bousquet and Elisseeff \(2002\)](#); [Devroye and Wagner \(1979\)](#);

Kearns and Ron (1999); Kuttin and Niyogi (2002); Poggio et al. (2004); Rogers and Wagner (1978)). We are using the notions of stability used by Agarwal and Niyogi (2009), which are defined below:

Definition 1 (Uniform loss stability (β)). *A ranking algorithm which is trained on the sample S of size n has a uniform loss stability β with respect to the ranking loss function ℓ if,*

$$|\ell(S) - \ell(S^k)| \leq \beta(n), \quad \forall n \in \mathbb{N}, 1 \leq k \leq n \quad (8.6)$$

where S^k is a sample resulting from changing the k^{th} element of S , i.e., changing the input training sample by a single example leads to a difference of at most $\beta(n)$ in the loss incurred by the output ranking function on any pair of examples. Thus, a smaller value of $\beta(n)$ corresponds to a greater loss stability.

Definition 2 (Uniform score stability (ν)). *A ranking algorithm with an output g_S on the training sample S of size n , has a uniform score stability ν if*

$$|g_S(x) - g_{S^k}(x)| \leq \nu(n), \quad \forall n \in \mathbb{N}, 1 \leq k \leq n, \forall x \in \mathcal{X} \quad (8.7)$$

i.e., changing an input training sample by a single example leads to a difference of at most $\nu(n)$ in the score assigned by the ranking function to any instance x . A smaller value of $\nu(n)$ represents greater score stability.

Let $g : \mathcal{X} \rightarrow \mathbb{R}$ be a ranking function on $\mathcal{X}$. Let $\ell : \mathbb{R}^{\mathcal{X}} \times (\mathcal{X} \times \mathcal{Y}) \times (\mathcal{X} \times \mathcal{Y}) \rightarrow \mathbb{R}^+ \cup \{0\}$ be the ranking loss function. The expected risk (error) associated with g is given by:

$$R(g) = E_{((X_i, Y_i), (X_j, Y_j))} [\ell(g, (X_i, Y_i), (X_j, Y_j))] \quad (8.8)$$

In practice, we must minimise an empirical ranking error with respect to a finite sample, $S = ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n$,

$$\hat{R}(g; S) = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \ell(g, (x_i, y_i), (x_j, y_j)) \quad (8.9)$$

We found that the notions of algorithmic stability, uniform loss stability and uniform score stability, are related. Their relation is defined by Theorem 3 for slack rescaled formulation.

Theorem 3. *Let $\mathcal{A}$ be a ranking algorithm with slack rescaled formulation whose output on a training sample $S \in (\mathcal{X}, \mathcal{Y})^n$ we denote by g_S . Let $\nu : \mathbb{N} \rightarrow \mathbb{R}$ be such that $\mathcal{A}$ has uniform score stability ν . $\mathcal{A}$ has uniform loss stability β with respect to the slack rescaling loss function ℓ_s , for all $n \in \mathbb{N}$, given by:*

$$\beta(n) = 2\sigma\nu(n) \quad (8.10)$$

where $\sigma \geq \Delta$ is an upper bound on the structured output loss function.

Proof. Without loss of generality we assume that $\ell_s(S) > \ell_s(S^k)$. There are two non-trivial cases.

Case (i): Margin is violated by both g_S and g_{S^k} .

$$\begin{aligned} |\ell_s(S) - \ell_s(S^k)| &= \\ &= |\Delta_j - \Delta_i| \cdot (1 - (g_S(x_i) - g_S(x_j)) \cdot \text{sign}(\Delta_j - \Delta_i)) - \\ &= |\Delta_j - \Delta_i| \cdot (1 - (g_{S^k}(x_i) - g_{S^k}(x_j)) \cdot \text{sign}(\Delta_j - \Delta_i)) \\ &\leq \sigma(|g_S(x_i) - g_{S^k}(x_i)| + |g_S(x_j) - g_{S^k}(x_j)|) \\ &\leq 2\sigma\nu(n) \end{aligned}$$

Case (ii): Margin is violated by either of g_S or g_{S^k} . This is a symmetric case, so we assume that the margin is violated by g_S .

$$\begin{aligned}
|\ell_s(S) - \ell_s(S^k)| &= \\
&|\Delta_j - \Delta_i| \cdot (1 - (g_S(x_i) - g_S(x_j)) \cdot \text{sign}(\Delta_j - \Delta_i)) \\
&\leq |\Delta_j - \Delta_i| \cdot (1 - (g_S(x_i) - g_S(x_j)) \cdot \text{sign}(\Delta_j - \Delta_i)) - \\
&\quad |\Delta_j - \Delta_i| \cdot (1 - (g_{S^k}(x_i) - g_{S^k}(x_j)) \cdot \text{sign}(\Delta_j - \Delta_i)) \\
&\leq \sigma(|g_S(x_i) - g_{S^k}(x_i)| + |g_S(x_j) - g_{S^k}(x_j)|) \\
&\leq 2\sigma\nu(n)
\end{aligned}
\quad \square$$

We will now introduce two lemmas from [Agarwal and Niyogi \(2009\)](#), which will be used for our proof.

Lemma 4. *Let $\mathcal{A}$ be a ranking algorithm whose output on a training sample $S \in (\mathcal{X}, \mathcal{Y})^n$ we denote by g_S . Let $\beta : \mathbb{N} \rightarrow \mathbb{R}$ be such that $\mathcal{A}$ has uniform loss stability β with respect to the ranking loss function ℓ . Let ℓ_r be a rescaled ramp loss function. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of S , the expected ranking error of the function is bounded by:*

$$R(g_S) < \hat{R}_{\ell_r}(g_S; S) + 2\beta(n) + (n\beta(n) + \sigma) \sqrt{\frac{2\ln(1/\delta)}{n}} \quad (8.11)$$

where $\sigma \geq \Delta$ is an upper bound on the structured output loss function.

Proof. The proof is given in Theorem 6 of [Agarwal and Niyogi \(2009\)](#). $\square$

Lemma 5. *Let $\mathcal{H}$ be a RKHS with a joint-kernel¹ k such that $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$, $k((x, y), (x, y)) \leq \kappa^2 < \infty$. Let $\lambda > 0$ and ℓ be the ranking loss function. The training algorithm trained on sample S of size n outputs a ranking function $g_S \in \mathcal{H}$ that satisfies $g_S = \arg \min_{g \in \mathcal{H}} \{\hat{R}_\ell(g; S) + \lambda \|g\|_{\mathcal{H}}^2\}$. Then the uniform score stability*

¹Here we assume a joint kernel map of the form given in [Taskar et al. \(2003\)](#); [Tsochantaridis et al. \(2004\)](#).

for the ranking function is given by:

$$\nu(n) = \frac{8\sigma\kappa^2}{\lambda n} \quad (8.12)$$

where $\sigma \geq \Delta$ is an upper bound on the structured output loss function.

Proof. The proof is given in Theorem 11 of [Agarwal and Niyogi \(2009\)](#). $\square$

Using above notations, convergence bounds for the structured output ranking algorithm for the slack and margin rescaled formulations are given by Theorem 6 and Theorem 7.

Theorem 6 (Slack Rescaling Generalisation Bound). *Let $\mathcal{H}$ be a RKHS with a joint-kernel k such that $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$, $k((x, y), (x, y)) \leq \kappa^2 < \infty$. Let $\lambda > 0$ and ℓ_r be a rescaled ramp loss function. The training algorithm trained on sample S of size n outputs a ranking function $g_S \in \mathcal{H}$ that satisfies $g_S = \arg \min_{g \in \mathcal{H}} \{\hat{R}_{\ell_s}(g; S) + \lambda \|g\|_{\mathcal{H}}^2\}$. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of S , the expected ranking error of the function is bounded by:*

$$R(g_S) < \hat{R}_{\ell_r}(g_S; S) + \frac{32\sigma^2\kappa^2}{\lambda n} + \left(\frac{16\sigma^2\kappa^2}{\lambda} + \sigma \right) \sqrt{\frac{2\ln(1/\delta)}{n}} \quad (8.13)$$

where $\sigma \geq \Delta$ is an upper bound on the structured output loss function.

Proof. From Lemma 5,

$$\nu(n) = \frac{8\sigma\kappa^2}{\lambda n}$$

Substituting this value of $\nu(n)$ in Equation (8.10),

$$\beta(n) = \frac{16\sigma^2\kappa^2}{\lambda n}. \quad (8.14)$$

From Lemma 4,

$$R(g_S) < \hat{R}_{\ell_r}(g_S; S) + 2\beta(n) + (n\beta(n) + \sigma) \sqrt{\frac{2\ln(1/\delta)}{n}}$$

Inequality (8.13) then follows by replacing $\beta(n)$ with its value from Equation (8.14). $\square$

Theorem 7 (Margin Rescaling Generalisation Bound). *Under the conditions of Theorem 6, and a ranking function $f_S \in \mathcal{H}$ that satisfies $f_S = \arg \min_{f \in \mathcal{H}} \{\hat{R}_{\ell_m}(f; S) + \lambda \|f\|_{\mathcal{H}}^2\}$. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of S , the expected ranking error of the function is bounded by:*

$$R(f_S) < \hat{R}_{\ell_r}(f_S; S) + \frac{32\kappa^2}{\lambda n} + \left(\frac{16\kappa^2}{\lambda} + \sigma \right) \sqrt{\frac{2\ln(1/\delta)}{n}} \quad (8.15)$$

Proof. The proof of above theorem is given in Section 5.2.1 of Agarwal and Niyogi (2009). $\square$

From Theorem 6 and Theorem 7, we can make an interesting observation. For $\Delta \in [0, 1)$ we have strictly tighter bounds for slack rescaling as compared to margin rescaling. For $\Delta \in [1, \sigma]$, where $\sigma > 1$, bounds are tighter for margin rescaling. Rescaling losses by a constant multiplier corresponds to selecting a different regulariser in the slack rescaling case. However, it does not indicate preference of one variant over the other *asymptotically*. A ranking algorithm is said to be consistent if in the process of minimising the surrogate ranking loss (hinge loss), asymptotically it also minimises the true ranking loss (0/1 loss). We note that these convergence bounds do not prove or disprove consistency of the algorithms.

8.3 Conclusions

The main difference between ranking problems and classification or regression problems is that the loss function in ranking is ‘pairwise’ (i.e., defined for every pair of data-points) rather than ‘pointwise’. Ranking can therefore be perceived as a weighted classification on pairs of data-points with the weights given by the ranking preferences (structured output loss values in our case). However, generalisation bounds from classification cannot be applied directly to ranking, due to the pairwise dependence. In this chapter, we derived the generalisation bounds for structured output ranking. The bounds show that the empirical error reduces to the lowest possible error with unlimited amount of training data. We further notice that the bounds are tighter for margin rescaling over a larger range of loss values, although it suffers from discontinuity when $\Delta_j - \Delta_i$ reduces to zero. As we pointed out, these observations are not sufficient to favour one formulation over another, but we hope it will open many avenues for future research.

In Chapter 7, we demonstrated applicability of structured output ranking for two major computer vision problems and in this chapter we established the stability of its learning algorithm. We hope with this theoretical support, structured output ranking will find applications for many more computer vision problems.

Chapter 9

Conclusions

In this thesis, we explored human layout estimation task and devised a method for estimating human layout within images for any viewpoint. We generalised this task to a structured multiclass problem where different class labels corresponded to different body parts and the structure of the prediction space is induced by the body layout. We further discovered that in this task, the correct estimations are ranked depending on the number of correct body part labels and hence this problem could be solved accurately using multiclass structured output ranking.

We started our investigation for human layout estimation with pictorial structures, but found that such an approach performs well only for near frontal poses. In order to overcome this shortcoming and make a generic human layout estimation system, we proposed a bottom-up approach where body part candidates are detected first, and are later verified using a system that encapsulates both the appearance of the body parts and pose of the complete human body. We trained independent detection systems for individual body parts using appearance of parts. These part detectors were further improved using their relative positions and sizes, and

ultimately combined using our structured output ranking algorithm to give a valid human body layout. We also applied our structured ranking algorithm for the task of taxonomic multiclass classification and achieved very good results. This shows that the technique that we have developed is not specific to a task but could be generalised to a wide variety of computer vision problems. We could summarise our research as follows:

In Chapter 4, we improved an existing pictorial structure model by adding extra features and reduced the search space using skin colour. We also used our skin detector to localise hands, which enabled us to win PASCAL VOC 2010 person layout challenge hand detection task (Everingham et al. (2010b)).

In Chapter 5, we extended our work on hand detection and improved it by adding information about hand shape and its context. The hand detector that we developed is currently state-of-the-art. As far as we are aware, the hand dataset that we collected for training our hand detector is the most comprehensive dataset of hand images currently available.

In Chapter 6, we proposed our efficient algorithm for structured output ranking. We showed that our algorithm is faster than any other existing method and is applicable to a wide variety of computer vision applications.

Chapter 7 described our method of solving human layout and taxonomic multiclass classification using structured output ranking. We showed that human layout estimation and taxonomic prediction can be generalised to structured multiclass labelling problem with implicit ranking of class labels. Using our method, we achieve better results than all the existing methods and also won the PASCAL VOC 2011 person layout challenge (Everingham et al. (2011)).

In Chapter 8, we gave the convergence bounds on the error for our algorithm

and showed that it converges to the lowest empirical error with unlimited amount of training data.

From our research, we have derived some broad conclusions:

- Inclusion of additional salient information about an object category improves the detection accuracy. For instance, the detection accuracy for hands improved with the addition of shape and context cues. Relative information, such as the position of head, the size of human body etc., improved the detection results further.
- For layout estimation, a bottom-up approach (detecting the body parts first), enabled us to devise a system that works for different viewpoints. On the contrary, a standard top-down method would have required numerous body templates for the same task and may not be extendable to unseen body poses.
- External semantic knowledge, such as class taxonomy, can make image retrieval results more meaningful.

9.1 Future work

9.1.1 Hand tracking

The hand detector described in Chapter 5 could be used for tracking hands in video sequences. A current estimate of hand position and size could be used to reduce the search space for detection in the adjacent frames for efficient tracking. Furthermore, video specific salient features like optical flow (Gibson (1950)) may be included to improve the results.

9.1.2 Markerless motion capture

Our method of human layout estimation could be extended to videos for markerless motion capture. A straight-forward method for this would be to apply layout estimation on the individual video frames. Temporal patterns of pose will be used for the motion estimation. This system could be made efficient by performing layout estimation only at every distinct frame, where distinction is defined on the basis of temporal consistency (Buehler et al. (2008)), and tracking the results for the intermediate frames.

9.1.3 Complete human body layout estimation

Pose estimation of the complete human body (i.e., both the upper and the lower body) could be performed using our framework by incorporating feet detection. In Chapter 7, we tried to do so, however, detection results for feet were poor due to the lack of any reliable feet detection system. Feet detection is more challenging than detecting hands, since most of the time feet are covered with some kind of footwear. However, context around the feet (e.g., detection of legs) would be a strong cue for localising them. If sufficient training data is available, accurate feet detection may be performed.

9.1.4 Consistency proof for structured output ranking

In Chapter 8, we gave the convergence bounds for structured output ranking, which guarantees that our algorithm converges asymptotically to a solution where minimum empirical error is achieved. However, we still need proof for the ranking consistency that verifies that the surrogate empirical error reduces to the true ranking

error.

Structured output SVM has been proved to be inconsistent for both the slack and margin rescaled versions by [McAllester \(2007\)](#). This fundamental shortcoming has not suppressed the popularity of the approach and it is used for a variety of applications (Chapter 2). There is no work that either proves or disproves the consistency of structured output ranking. Analysis of consistency of structured output ranking will be a fundamental contribution, as it will lead to better understanding of the learning theory.

9.1.5 Taxonomy learning

In Chapter 7, using structured output ranking, we demonstrated that semantic multiclass image classification can be performed by learning a multiclass classification system using a pre-defined taxonomy. The caveat here is that the taxonomy should be available beforehand. There are some systems, which try to learn a taxonomy from the available training data (Chapter 2). They commonly use generative methods, such as clustering, to perform this. It will be interesting to investigate discriminative approaches for this task. Discriminative approaches are traditionally less prone to over-fitting to the data and may help in dimensionality reduction as well.

Appendix A

Line fitting algorithm

Algorithm 6 Line fitting algorithm.

```
1: Input: Skin mask  $\mathcal{S}$ 
2:  $\mathcal{L} \leftarrow \emptyset$ 
3:  $\mathcal{L}_{all} \leftarrow HoughTransform(\mathcal{S})$ 
4: repeat
5:    $\mathcal{B} \leftarrow Boundary(\mathcal{S})$ 
6:   for  $i = 1, \dots, |\mathcal{L}_{all}|$  do
7:      $[score_i, boundarypixels_i] \leftarrow CumulativeDistanceTransform(\mathcal{B}, \mathcal{L}_{all_i})$ 
8:   end for
9:    $maxline \leftarrow \max(score, \mathcal{L}_{all})$ 
10:   $\mathcal{L} \leftarrow \mathcal{L} + maxline$ 
11:  for  $j = maxline_{y_1}, \dots, maxline_{y_2}$  do
12:    for  $k = boundarypixels_{x_1}(j), \dots, boundarypixels_{x_2}(j)$  do
13:       $\mathcal{S}(j, k) \leftarrow 0$ 
14:    end for
15:  end for
16:   $\mathcal{L}_{all} \leftarrow HoughTransform(\mathcal{S})$ 
17: until  $|\mathcal{L}_{all}| > 0$ 
18: return  $\mathcal{L}$ 
```

Algorithm 6 describes our method for fitting lines through skin regions. First of all, using Hough transform (Duda and Hart (1972)), lines are hypothesised from the skin mask (line 3). For each of the hypothesised line, a score is computed as

cumulative distance transform ([Breu et al. \(1995\)](#)) over all the pixels belonging to the line. The function ‘CumulativeDistanceTransform’ (line 7) returns this score and co-ordinates of the two boundary pixels at the extreme left and right of the mask for every pixel in the line. The maximum scoring line is picked and added to the set of fitted lines (line 10). Then, the skin mask along this line is erased so that no new line is detected in future iterations from the same region (lines 11-15). This process is repeated until no new line is found using Hough transform.

Appendix B

More qualitative results for hand detection

Following table shows top 100 ordered (in terms of confidence score) detection results for the hand dataset. The number shown below each image is the ranking of the detected hand in terms of the confidence score from the classifier. It can be seen that our hand detector is quite robust and we get only 1 false positive (at rank 58) out of the first 100 detections. False positives are shown by a red bounding box. There are in total 660 hand instances in the test set.



1



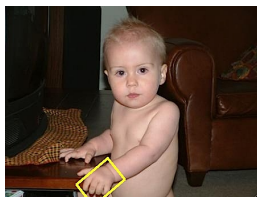
2



3



4



5



6



7



8



9



10



11



12



13



14



15



16



17



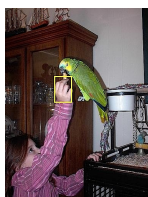
18



19



20



21



22



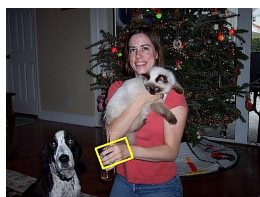
23



24



25



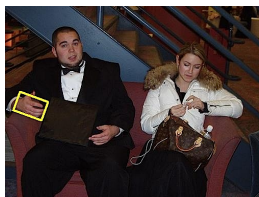
26



27



28



29



30



31



32



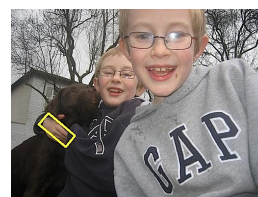
33



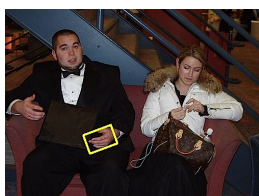
34



35



36



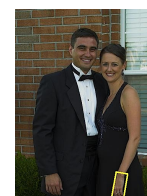
37



38



39



40



41



42



43



44



45



46



47



48



49



50



51



52



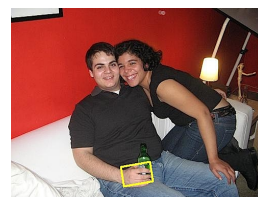
53



54



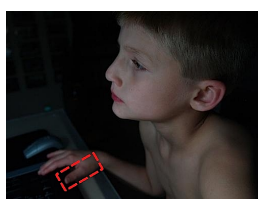
55



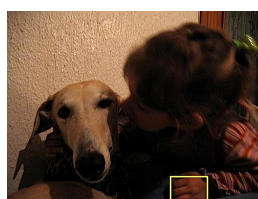
56



57



58



59



60



61



62



63



64



65



66



67



68



69



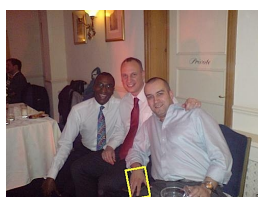
70



71



72



73



74



75



76



77



78



79



80



81



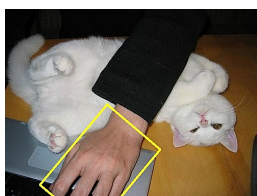
82



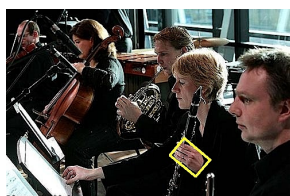
83



84



85



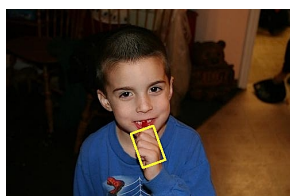
86



87



88



89



90



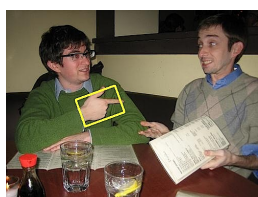
91



92



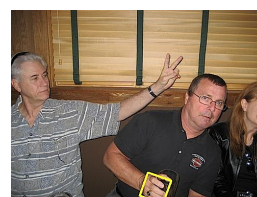
93



94



95



96



97



98



99



100

Bibliography

- A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *Proceedings of the European Conference on Computer Vision*, 2004. 30, 31
- A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. 30, 31
- S. Agarwal and P. Niyogi. Stability and generalization of bipartite ranking algorithms. In *Proceedings of International Conference on Learning Theory*, 2005. 148, 151
- S. Agarwal and P. Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 2009. 151, 152, 154, 155, 156
- S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004. 15
- B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 20
- M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 21, 39, 92
- P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 104
- P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 135

- N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 1950. 151
- V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 23, 24
- S. D. Bay. Combining nearest neighbor classifiers through multiple feature subsets. In *Proceedings of the International Conference on Machine Learning*, 1998. 43
- S. Belongie and J. Malik. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 14
- S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in Neural Information Processing Systems*, 2000. 29, 39
- S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems*, 2010. 137
- A. Binder, K.-R. Müller, and M. Kawanabe. On taxonomies for multi-class image categorization. *International Journal of Computer Vision*, 2011. 44, 46, 127, 138, 140
- C. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. 19, 43
- M. B. Blaschko and A. Gretton. Learning taxonomies by dependence maximization. In *Advances in Neural Information Processing Systems*, 2008. 45
- M. B. Blaschko, A. Vedaldi, and A. Zisserman. Simultaneous object detection and ranking with weak supervision. In *Advances in Neural Information Processing Systems*, 2010. 58, 127
- A. Bordes, R. Vauquelin, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 2005. 48
- L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Journal of Computer Vision*, 2009. 40, 62
- L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *Proceedings of the European Conference on Computer Vision*, 2010. 40

- O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2002. 151
- R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and J. M. Brady. A linguistic feature vector for the visual interpretation of sign language. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. Springer-Verlag, 2004. 21
- S. Boyd and L. Vandenberghe. Localization and cutting-plane methods, 2007. URL <http://www.stanford.edu/class/ee392o/localization-methods.pdf>. 49
- L. Breiman, F. J., R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA, 1984. 43
- H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995. 18, 164
- P. Buehler. *Automatic Learning of British Sign Language from Signed TV Broadcasts*. PhD thesis, University of Oxford, 2010. 27
- P. Buehler, M. Everingham, D. P. Huttenlocher, and A. Zisserman. Long term arm and hand tracking for continuous sign language TV broadcasts. In *Proceedings of the 19th British Machine Vision Conference, Leeds*, 2008. 4, 21, 22, 65, 161
- P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching TV (using weakly aligned subtitles). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 4
- Buffy stickmen. <http://www.robots.ox.ac.uk/~vgg/data/stickmen/>. 61, 79
- C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998. 46, 48, 104
- L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of ACM International conference on Information and knowledge management*, 2004. 138
- L. Cai and T. Hofmann. Exploiting known taxonomies in learning overlapping concepts. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2007. 138
- J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. 74, 105

- C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. URL <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. 48
- Y. Chen, M. Crawford, and J. Ghosh. Integrating support vector machines in a hierarchical output decomposition framework. In *International Geoscience and Remote Sensing Symposium*, 2004. 45
- Y. Chen, L. Zhu, and A. L. Yuille. Active mask hierarchies for object detection. In *Proceedings of the European Conference on Computer Vision*, 2010. 17
- O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis*, 2007. 15
- R. Collobert and S. Bengio. SVMtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 2001. 48
- C. Connaire, N. O'Connor, and A. Smeaton. Detector adaptation by maximising agreement between independent data sources. In *OTCBVS'07 - IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, 2007. 74, 76
- C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 1995. 46
- D. Cossock and T. Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 2008. 151
- N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 14, 15, 19, 62, 84, 100
- V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *Proceedings of the British Machine Vision Conference*, 2010. 21
- J. Deng, A. C. Berg, and F.-F. Li, K. and Li. What does classifying more than 10, 000 image categories tell us? In *Proceedings of the European Conference on Computer Vision*, 2010. 44, 127
- C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proceedings of the International Conference on Computer Vision*, 2009. 20, 83
- L. Devroye and T. J. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 1979. 151

- J. Duchi, L. W. Mackey, and M. I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the International Conference on Machine Learning*, 2010. 151
- R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 1972. 82, 163
- M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proceedings of the British Machine Vision Conference*, 2009. 39, 80, 86, 91, 92, 93, 128
- C. H. Ek, P. H. S. Torr, and N. D. Lawrence. Gaussian process latent variable models for human pose estimation. In *Proceedings of International conference on Machine Learning for Multimodal Interaction*, 2007. 31, 32
- C. H. Ek, J. Rihan, P. H. S. Torr, G. Rogez, and N. D. Lawrence. Ambiguity modeling in latent spaces. In *Proceedings of International conference on Machine Learning for Multimodal Interaction*, 2008. 31, 32
- G. ElKoura and K. Singh. Handrix: Animating the human hand. In *Annual Conference of the European Association for Computer Graphics*, 2003. 21
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>, 2007. 68
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 2010a. 11, 51, 64, 68, 69, 116
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. <http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html>, 2010b. 8, 11, 21, 71, 99, 109, 159
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 (VOC2011). <http://www.pascal-network.org/challenges/VOC/voc2011/>, 2011. 9, 11, 71, 95, 127, 128, 159
- X. Fan. Efficient multiclass object detection by a hierarchy of classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 45

- A. Farhadi and D. Forsyth. Aligning asl for statistical translation using a discriminative word model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 21
- A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 22
- P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 2005. 24, 25, 26, 33, 82
- P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010a. 106
- P. F. Felzenszwalb and D. P. Huttenlocher. Efficient matching of pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000. 25, 33
- P. F. Felzenszwalb, R. B. Grishick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010b. 15, 17, 19, 20, 42, 84, 100, 101, 134, 135
- V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska*, 2008. 21, 38, 39, 79
- V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 5
- M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computer*, 1973. 25, 26, 33
- M. M. Fleck, D. A. Forsyth, and C. Bregler. Finding naked people. In *Proceedings of the European Conference on Computer Vision*. Springer-Verlag, 1996. 73
- D. Forsyth and M. Fleck. Body plans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997. 33
- Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings*, 1996. 20

- Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 2003. 148
- J. Fritsch, S. Lang, M. Kleinhagenbrock, G. A. Fink, and G. Sagerer. Improving adaptive skin color segmentation by incorporating results from face detection. In *IEEE Workshop on Robot and Human Interactive Communication*, 2002. 73
- D. M. Gavrilu. A bayesian exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007. 18
- J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin, Boston, 1950. 160
- G. Griffin and P. Perona. Learning and using taxonomies for fast visual categorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 45
- A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. 21
- H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *Proceedings of the International Conference on Computer Vision*, 2009. 20
- X. He and R. S. Zemel. Latent topic random fields: Learning using a taxonomy of labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 44
- T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *Intl. Conf. on Automatic Face and Gesture Recognition*, 1996. 18
- R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*. MIT Press, 2000. 52, 53
- N. R. Howe. Silhouette lookup for automatic pose tracking. In *Computer Vision and Pattern Recognition Workshop*, 2004. 30
- R. L. Hsu, M. A. Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002. 73
- G. Hua, M. Yang, and Y. Wu. Learning to estimate human pose with data driven belief propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 34, 39

- K. Imagawa, S. Lu, and S. Igi. Color-based hands tracking system for sign language recognition. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 1998. 73
- ImageNet challenge. <http://www.image-net.org/>. 140
- S. Ioffe and D. Forsyth. Finding people by sampling. In *Proceedings of the International Conference on Computer Vision*, 1999. 33
- T. Joachims. SVM^{light}. <http://svmlight.joachims.org/>. 48
- T. Joachims. Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning*, 1999. 48, 104
- T. Joachims. Training linear SVMs in linear time. In *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining (KDD)*, 2006. 12, 54, 55, 56, 57, 117, 118, 120
- T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 2009. 54, 56, 57
- S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2010. 42
- S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 42
- M. J. Jones and J. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 2002. 60, 64, 73, 75
- T. Kadir, R. Bowden, E. J. Ong, and A. Zisserman. Minimal training, large lexicon, unconstrained sign language recognition. In *Proceedings of the 15th British Machine Vision Conference, Kingston*, 2004. 23
- L. Karlinsky, M. Dinerstein, D. Harari, and S. Ullman. The chains model for detecting parts by their context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 67, 99, 108, 109
- M. J. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 1999. 152
- D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tu. Real-time vision-based camera tracking for augmented reality applications. In *ACM Symposium on Virtual Reality, Software and Technology (VRST-97)*, 1997. 44

- M. Kolsch and M. Turk. Robust hand detection. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2004. 22
- U. H.-G. Kressel. Pairwise classification and support vector machines. *Advances in Kernel Methods - Support Vector Learning*, 1999. 43
- M. P. Kumar. *Combinatorial and convex optimization for probabilistic models in computer vision*. PhD thesis, Oxford Brookes University, 2008. 94
- M. P. Kumar, P. H. S. Torr, and A. Zisserman. Efficient discriminative learning of parts-based models. In *Proceedings of the International Conference on Computer Vision*, 2009. 21, 39, 77, 83, 88
- S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2002. 151, 152
- J. Lafferty, A. McCallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning, Williamstown, USA*, 2001. 50
- C. H. Lampert and M. B. Blaschko. A multiple kernel learning approach to joint multi-class object detection. In *Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM)-Symposium*, 2008. 45
- C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. 16
- I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 5
- S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York*, 2006. 16, 141
- B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005. 15
- T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 2001. 14

- F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 124
- S. Liu, H. Yi, L. T. Chia, and D. Rajan. Adaptive hierarchical multi-class svm classifier for texture-based image classification. In *Proceedings of IEEE International Conference on Multimedia and Expo*, 2005. 45
- D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004. 14, 15, 18, 140
- S. Maji and J. Malik. Object detection using a max-margin hough transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 16
- M. Marin-Jimenez, A. Zisserman, and V. Ferrari. "Here's looking at you, kid." Detecting people looking at each other in videos. In *Proceedings of the British Machine Vision Conference*, 2011. 111
- M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 44, 46
- M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *Proceedings of the European Conference on Computer Vision*, 2008. 45
- D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the International Conference on Computer Vision*, 2001. 61
- D. McAllester. Generalization bounds and consistency for structured labeling. In *Predicting Structured Data*. MIT Press, 2007. 149, 162
- A. McCallum, R. Rosenfeld, T. M. Mitchell, and A. Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the International Conference on Machine Learning*, 1998. 44
- Microsoft Xbox Kinect. <http://www.xbox.com/en-US/kinect>. 2
- K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 2004. 14

- K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*. Springer-Verlag, 2004. 20, 34
- K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 18
- T. B. Moeslund and E. Granum. Multiple cues used in model-based human motion capture. In *International Conference on Automatic Face and Gesture Recognition*, 2000. 13
- A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2001. 15, 19, 33, 34, 35, 39
- G. Mori and J. Malik. Estimating human body configurations using shape context matching. In *Proceedings of the European Conference on Computer Vision*, 2002. 29, 30
- G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004. 1, 34, 36
- M. A. Mottaleb and A. M. Elgammal. Face detection in complex environments from color images. In *Proceedings of the IEEE International Conference on Image Processing*, 1999. 73
- R. Navaratnam, A. Thayananthan, P. Torr, and R. Cipolla. Hierarchical part-based human body pose estimation. In *Proceedings of the British Machine Vision Conference*, 2005. 36
- R. Navaratnam, A. W. Fitzgibbon, and R. Cipolla. The joint manifold model for semi-supervised multi-valued regression. In *Proceedings of the International Conference on Computer Vision*, 2007. 31, 32
- T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 1996. 14
- E. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2004. 22, 23

- C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 2000. 14, 15, 19
- PASCAL stickmen. http://www.vision.ee.ethz.ch/~calvin/ethz_pascal_stickmen/. 61, 79
- PASCAL VOC 2010 results. <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2010/results/>. 20, 132
- PASCAL VOC 2011 results. <http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2011/results/>. 20, 136
- A. Patron-Perez, M. Marszałek, A. Zisserman, and I. D. Reid. High five: Recognising human interactions in TV shows. In *Proceedings of the British Machine Vision Conference*, 2010. 5
- J. C. Platt. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods - Support Vector Learning*, 1999. 48
- T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 2004. 152
- A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 67
- E. Rahtu, J. Kannala, and M. B. Blaschko. Learning a category independent object detection cascade. In *Proceedings of the International Conference on Computer Vision*, 2011. 58, 127
- D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems*. MIT Press, 2006. 37, 39, 79
- D. Ramanan. Using segmentation to verify object hypotheses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 84
- D. Ramanan and C. Sminchisescu. Training deformable models for localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2006. 36, 39
- X. Ren, A. C. Berg, and J. Malik. Recovering human body configurations using pairwise constraints between parts. In *Proceedings of the International Conference on Computer Vision*, 2005. 34
- I. Rish. An empirical study of the naive bayes classifier. In *International Joint Conference on Artificial Intelligence workshop on Empirical Methods in AI*, 2001. 43

- W. H. Rogers and T. J. Wagner. A finite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics*, 1978. 151, 152
- S. Romdhani, P. H. S. Torr, B. Schölkopf, and A. Blake. Computationally efficient face detection. In *Proceedings of the International Conference on Computer Vision*, 2001. 20
- R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proceedings of the European Conference on Computer Vision*, 2002. 33, 34, 39
- R. Rosales and S. Sclaroff. Learning and synthesizing human body motion and posture. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2000. 29
- R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff. 3d hand pose reconstruction using specialized mappings. In *Proceedings of the International Conference on Computer Vision*, 2000. 23
- B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010a. 41
- B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proceedings of the European Conference on Computer Vision*, 2010b. 41
- B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 41
- B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002. 46
- G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proceedings of the International Conference on Computer Vision*, 2003. 29
- L. Sigal and M. Black. Predicting 3D people from 2D pictures. In *Proceedings of the International Conference on Articulated Motion and Deformable Objects*. Springer, 2006. 28
- J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, 2003. 15
- Sony Eye Toy. <http://us.playstation.com/ps2/accessories/eyetoy-usb-camera-ps2.html>. 2

- B. Stenger. Template-based hand pose recognition using multiple cues. In *Proceedings of the Asian Conference on Computer Vision*, 2006. 23
- B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Hand pose estimation using hierarchical detection. In *International Workshop on Human-Computer Interaction*, 2004. 23
- B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. 23
- D. J. Sturman and D. Zeltzer. A survey of glove-based input. *IEEE Computer Graphics and Applications*, 1994. 21
- B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems*, 2003. 50, 154
- R. Tibshirani and T. Hastie. Margin trees for high-dimensional classification. *Journal of Machine Learning Research*, 2007. 45
- C. Tomasi, S. Petrov, and A. Sastry. 3d tracking = classification + interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 23
- A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC*, 2004. 20
- K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *Proceedings of the International Conference on Computer Vision*, 2001. 18
- I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning*, 2004. 50, 54, 154
- K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *Proceedings of the International Conference on Computer Vision*, 2011. 16
- R. J. Vanderbei. Loqo: An interior point code for quadratic programming. Technical report, Princeton University, 1994. 48
- V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 1971. 147

- V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995. 43, 46
- A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision*, 2009. 16
- Vicon Motion Capture. <http://www.vicon.com>. 3
- P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 14, 19, 20, 74, 76
- A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 1967. 26, 36
- C. Vogler and D. Metaxas. Adapting hidden Markov models for ASL recognition by using three-dimensional computer vision methods. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1997. 22
- C. Vogler and D. Metaxas. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *Proceedings of the International Conference on Computer Vision*, 1998. 22
- C. Vogler and D. Metaxas. Handshapes and movements: Multiple-channel American sign language recognition. In *Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop*. Springer, 2004. 22
- C. Vogler and D. N. Metaxas. Parallel hidden Markov models for american sign language recognition. In *Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece*, 1999. 22
- J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 141
- B. Wu and R. Nevatia. Tracking of multiple, partially occluded humans based on static body part detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 20
- Y. Wu, Q. Liu, and T. S. Huang. An adaptive self-organizing color segmentation algorithm with application to robust real-time human hand localization. In *Proceedings of the Asian Conference on Computer Vision*, 2000. 22

- Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 42, 114, 128
- B. Yao and L. Fei-Fei. Grouplet: a structured image representation for recognizing human and object interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 21
- X. Yuan, W. Lai, T. Mei, X. Hua, X. Wu, and S. Li. Automatic video genre categorization using hierarchical svm. In *Proceedings of the IEEE International Conference on Image Processing*, 2006. 45
- Z. Zhang, J. Warrell, and P. H. S. Torr. Proposal generation for object detection using cascaded ranking SVMs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 58, 127
- L. Zhigang, S. Wenzhong, Q. Qianqing, L. Xiaowen, and X. Donghui. Hierarchical support vector machines. In *International Geoscience and Remote Sensing Symposium*, 2005. 45
- L. Zhu, Y. Chen, A. L. Yuille, and W. T. Freeman. Latent hierarchical structural learning for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 17
- Q. Zhu, M. C. Yeh, K. T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006. 20
- X. Zhu, J. Yang, and A. Waibel. Segmenting hands of arbitrary color. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2000. 22
- A. Zweig and D. Weinshall. Exploiting object hierarchy: Combining models from different category levels. In *Proceedings of the International Conference on Computer Vision*, 2007. 44