Poselets and Their Applications in High-Level Computer Vision

Lubomir Bourdev

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-52
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-52.html

May 1, 2012

Acknowledgement

I would like to acknowledge my adviser Jitendra Malik, the members of my Dissertation Committee and Quals Committees Trevor Darrell, Bruno Olshausen and Ruzena Bajcsy, my collaborators Subhransu Maji, Thomas Brox, Pablo Arbelaez, Chunhui Gu and the rest of my friends in the vision group, my employer Adobe Systems, who sponsored me throughout my Ph.D. and my wife Ina and son Martin who sacrificed a lot over the past four years.

Poselets and Their Applications in High-Level Computer Vision

by

Lubomir Bourdev

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jitendra Malik, Chair
Professor Trevor Darrell
Professor Bruno Olshausen

Spring 2011

Poselets and Their Applications in High-Level Computer Vision

Abstract

Poselets and Their Applications in High-Level Computer Vision

by

Lubomir Bourdev

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Jitendra Malik, Chair

Part detectors are a common way to handle the variability in appearance in high-level computer vision problems, such as detection and semantic segmentation. Identifying good parts, however, remains an open question. Anatomical parts, such as arms and legs, are difficult to detect reliably because parallel lines are common in natural images. In contrast, a visual conjunction such as "half of a frontal face and a left shoulder" may be a perfectly good discriminative visual pattern. We propose a new computer vision part, called a *poselet*, which is trained to respond to a given part of the object at a given viewpoint and pose. There is a wide variety of poselets – a frontal face, a profile face, a head-and-shoulder configuration, etc. A requirement for training poselets is that the visual correspondence of object parts in the training images be provided. We create a new dataset, H3D, in which we annotate the locations of keypoints of people, infer their 3D pose and label their parts (the face, hair, upper clothes, etc.). Our richly annotated dataset allows for creation of poselets as well as other queries not possible with traditional datasets.

To train a poselet associated with a given image patch, we find other patches that have the same local configuration of keypoints and use them as positive training examples. We use HOG features and linear SVM classifiers. The resulting poselet is trained to recognize the visual patterns associated with the given local configuration of keypoints, which, in turn, makes it respond to a specific pose under a specific viewpoint regardless of the variation in appearance.

High-level computer vision is challenging because the image is a function of multiple somewhat independent factors, such as the appearance model of the object, its pose, and the camera viewpoint. Poselets allow us to "untie the knot", i.e. decouple the pose from the appearance and model them separately. We show that this property helps in a variety of high-level computer vision tasks. Our person detector based on poselets is the leading method on the PASCAL VOC 2009 and 2010 person detection competitions and naturally extends to other visual classes. We currently have the best semantic segmentation engine for person and several other categories on the PASCAL 2010 segmentation datasets. We report

competitive performance for pose and action recognition and we are the first method to do attribute classification for people under any viewpoint and pose.

Contents

List of Figures **iii**

List of Tables **viii**

1 Introduction **1**
1.1 The case for extra supervision 2
1.2 What parts should we search for? 3

2 The H3D dataset **6**
2.1 The role of datasets in computer vision 6
2.2 The Humans in 3D Dataset (H3D) 7
2.3 The Human Annotation Tool 8
2.3.1 Interaction Model 8
2.3.2 3D reconstruction 9
2.3.3 Region Labeling Tool 11
2.4 Applications of H3D 11
2.4.1 Viewpoint Extraction 11
2.4.2 3D Pose Statistics 12
2.4.3 Appearance Queries 13
2.5 Conclusion 14

3 Poselets **15**
3.1 Finding corresponding patches 15
3.1.1 Distance in configuration space 15
3.1.2 Determining the right number of training examples 17
3.2 Training a poselet 18
3.3 Poselet selection 18
3.4 Training poselet prediction parameters 21
3.5 Consistent poselet activations 22
3.6 Enhancing poselets using context 23
3.7 Combining poselet activations 24

3.8 Poselets beyond people . . . 27

4 Applications of Poselets **31**
4.1 Introduction . . . 31
4.2 Object Recognition with Poselets . . . 31
4.3 Attribute Classification with Poselets . . . 33
4.4 Semantic Segmentation with Poselets . . . 33
4.5 Pose Estimation with Poselets . . . 35
4.6 Action Classification with Poselets . . . 36

5 Attribute Classification with Poselets **41**
5.1 Introduction . . . 41
5.2 Related work . . . 43
5.3 The Attributes of People dataset . . . 44
5.4 Algorithm Overview . . . 46
5.5 Training and using poselets . . . 47
5.6 Poselet-level features ϕ^i . . . 49
5.7 Classifiers . . . 49
5.7.1 Poselet-level attribute classifier r_j^i . . . 49
5.7.2 Person-level attribute classifier s_j . . . 50
5.7.3 Context-level attribute classifier S_j . . . 51
5.8 Experimental results . . . 51
5.8.1 Performance *vs.* baselines . . . 51
5.8.2 Performance from different viewpoints . . . 54
5.8.3 Optimal places to look for an attribute . . . 54
5.8.4 Gender recognition performance . . . 57
5.8.5 Comparisons to human visual system . . . 58
5.9 Describing people with poselets . . . 58
5.10 Discussion . . . 61

6 Conclusion **62**

Bibliography **63**

List of Figures

1.1 Challenges of object recognition 1
1.2 The Latent SVM model by Felzenszwalb et al. 3
1.3 Examples of poselets 4

2.1 Examples of annotations from the H3D dataset. 7
2.2 HAT Java3D annotation tool 8
2.3 HAT Region Labeling Tool 11
2.4 Examples of 3D pose statistics made possible by H3D. 12
2.5 Examples of 2D image statistics made possible by H3D 13
2.6 Examples of appearance queries generated by H3D. 14
2.7 Part color models trained from H3D. 14

3.1 A poselet describing a frontal face and five of its examples. 15
3.2 Example of the swamping phenomenon for a profile face poselet. 17
3.3 Examples of poselet activations and their associated labels. 19
3.4 Poselet coverage matrix. 20
3.5 The first ten poselets for the person category. 21
3.6 Empirical keypoint distribution for a poselet. 22
3.7 Consistency of poselet activations. 23
3.8 Poselet classification errors due to lack of context. 24
3.9 The top activations of a poselet with and without context. 25
3.10 ROC curves for poselets with and without context. 25
3.11 Poselet clustering algorithm. 26
3.12 Examples of poselets from various visual categories. 29
3.13 More examples of poselets from various visual categories. 30

4.1 Detection examples. 37
4.2 Examples of our person segmentation. 38
4.3 Examples of segmenting two categories. 38
4.4 Segmentation examples from multiple visual categories. 39
4.5 Keypoint localization performance. 39

4.6 Error in predicting yaw across views. 40
4.7 Examples of poselets trained to recognize specific actions. 40

5.1 We can easily infer the gender using a cropped image of a person. 42
5.2 Attribute classification is much easier when aspect is fixed. 43
5.3 Fifty images drawn at random from our attributes test set. 45
5.4 Overview of our attribute classification algorithm. 46
5.5 Examples of a poselet and its mask for the parts. 48
5.6 Example of matching detected bounds to truth bounds. 48
5.7 Computing skin-specific features. 50
5.8 The six highest and lowest scoring examples of each attribute on our test set. 52
5.9 Examples of most confused attributes. 53
5.10 Views we provide to our attribute baselines. 54
5.11 Precision-recall curves of the attribute classifiers. 56
5.12 Optimal poses and viewpoints to look for evidence of a given attribute. . . . 57
5.13 Correlation between human performance and the poselets algorithm. 59
5.14 Precision-recall curves on gender recognition. 59
5.15 Describing people with poselets. 60

List of Tables

3.1 Class-specific variations in the poselet training parameters. 28

4.1 Average precision on the PASCAL competitions on the person category. . . 33
4.2 AP on PASCAL 2007 as a function of context and number of poselets. . . . 33
4.3 Detection and segmentation results for poselets on PASCAL 2010. 34
4.4 Average precision on the PASCAL VOC 2010 action classification task. . . . 36

5.1 Number of positive and negative labels for our attributes. 45
5.2 Average precision of baselines relative to our attribute classification model. . 55
5.3 Average precision for the attributes as a function of viewpoint. 55

Acknowledgments

First and foremost I would like to thank my advisor Prof. Jitendra Malik, who has completely changed my views on learning, vision and research in general. Let us take, for example, the topic of deep learning. Jitendra has made his view on deep learning well known, so as a new graduate student I came prepared with (or, dare I say, surrendered to) the expectation that "there is no deep learning in Jitendra's vision group". Over the years I have come to realize that I was wrong. It turns out that there is a lot of deep learning research in our lab, although of a slightly different kind. It happens during the countless hours we spend looking at data, visualizing failure cases and trying to get intuition about what is important. It is our intuition that we should develop first – that is the unwritten rule in our lab. Jitendra has a unique teaching style: he doesn't tell us the answers; he steers us towards them so that we can build our intuition in the process. He doesn't directly ask us to do work. Instead, he uses his infinite enthusiasm to motivate us, and it is quite infectious! I am grateful for the countless hours he spent working with me on poselets and teaching me so much in the process. Jitendra has led me to conclude that the critical component in vision is *features*, not learning, and that we should avoid learning something we can specify. H3D, poselets and my entire thesis is a manifestation of this philosophy.

I would like to thank Prof. Trevor Darrell and Prof. Bruno Olshausen for serving on my qualification and dissertation committees and Prof. Ruzena Bajcsy for serving on my qualification committee. Trevor's Visual Object and Activity Recognition class and Bruno's Neural Computation class were two of the most fun and inspiring classes I have ever taken.

I would like to thank all my colleagues at work who made it possible for Adobe to sponsor me and keep me on payroll during my Ph.D. Without their support, at this point in my life I would not have been able to pursue a Ph.D. Specifically I would like to thank Martin Newell (my former manager), David Salesin (my current manager), Sara Perkovic and Leslie Bixel, who designed and managed the university sabbatical program, Tom Malloy (head of Adobe's research organization) who sponsored the program and Adobe's former and current presidents Bruce Chizen and Shantanu Narayen who approved the program and encouraged me to pursue a Ph.D.

I would like to thank my wife Ina and my 9-year-old son Martin who sacrificed tremendously over the past four years so I can get my Ph.D. Martin spent nearly half of his current life with his father coming only on weekends and Ina was almost like a single mom having to deal both with a full-time job and taking care of our son. I am looking forward to spending more time with them after graduation!

Last but not least I would like to thank my friends and collaborators at Berkeley. Subhransu was my most frequent collaborator, and I admire his intelligence, intuition and knowledge. He has a bright future! Thomas Brox gave us the much needed German sense of order and organization, critical for some of our papers. I also enjoyed the company of the rest my friends Pablo Arbelaez, Chunhui Gu, Patrick Sundberg, Jon Barron, Chetan Nandakumar, Joseph Lim, the older and wiser generation (Michael Maire, Alex Berg, Ashley

Eden) and the newer generation (Georgia Gkioxari and Bharath Hariharan). I will cherish the memories of the time we spent together climbing the mountains in Crete, strolling the streets of Tokyo, swimming in the warm waters at Miami Beach, or playing volleyball in our good old Soda court. I am looking forward to hanging out with the "Big J crew" both at Berkeley and during future conferences.

Chapter 1

Introduction

Our long-term goal is to detect and describe the objects in Flickr-style photographs, such as the ones shown on Figure 1.1. For example, we would like to determine that the first image contains a woman with long hair who is sitting behind a table and looking to the right. We would like to segment the shape of the woman and the table. We would like to determine that the second image contains a person riding a motorcycle, wearing a helmet and facing right.

Describing these images requires solving several problems in high-level vision – detection, semantic segmentation, pose estimation, action recognition and attribute classification. This thesis is far from providing a comprehensive solution to these central computer vision problems; the war on solving vision is sure to go on for a long time, but we provide a new weapon, a part based model called *poselets*, and we have won a few small battles with it: Poselets are the basis of the current leading methods in both detection and segmentation of people and several other visual categories on the PASCAL VOC 2010 test set. We show competitive results on the PASCAL action recognition challenge and we are the first method to provide attribute classification for people under arbitrary viewpoints and poses.

Our work differs from the traditional approaches in two aspects – our use of extra supervision and our choice of parts.

Figure 1.1: Challenges of object recognition

1.1 The case for extra supervision

Figure 1.1 illustrates some of the challenges of person detection, which are representative of other categories and other high-level vision problems: occlusion (1), large articulation (4), varied clothes with sharp edges inside (2), no edges along the person's outline (4), the presence of glasses or hat/helmet (2,3,6,7), camera viewpoint (3), wrinkles on the clothes (6) and many other factors. Despite the huge complexity introduced by all of these factors, the mainstream computer vision setup, adopted by nearly all datasets, is to use the visible bounding box as the only degree of supervision. Given just the bounding box, how is a person classifier expected to make sense of the patterns comprising the humans shown on Figure 1.1? The only hope is for it, with enough data, to discover common patterns and their most likely locations, such as the fact that faces tend to have an oval shape and most commonly appear in the upper part of the bounding box. This is, in fact, the strategy employed by the Latent SVM detector by Felszenswalb et al. [16], one of the most popular object detectors today. This detector consists of a global low-resolution template and a set of smaller higher-resolution template parts. The appearance and the relative location of the parts are learnt jointly. The benefit of unsupervised methods is that they require no extra annotations, but the disadvantage is that they have to discover the patterns on their own, which could lead to patterns with mixed semantics and imprecise localization. Figure 1.2 shows two examples of the Latent SVM detector instantiated on two pedestrians. As the figure shows, the model has discovered that the head is a common pattern and has correctly initialized and localized a head part. Notice, however, that the "left part of the body" part could sometimes cover half of the face and in other cases it could be instantiated below the shoulder area. Similarly, the leg part can fire when the legs are near each other or when only one leg is present. Parts with such mixed semantics are both harder to train and not as informative as parts with precise semantics.

We believe that the visible bounds provide a poor and insufficient level of supervision and make it difficult to train object detectors. The key information that we would like to help our classifiers with is *visual correspondence*. That is, we would like to associate each pixel from the body of one person to the corresponding pixel, if any, from the body of another. To allow for such dense correspondence, we created a dataset in which we have labelled the locations of keypoints, such as the eyes, nose, shoulders, hips, etc. Our dataset, H3D (Humans in 3D), is described in Chapter 2. Poselets are an example of a new part that would not be possible to construct without the extra supervision provided by H3D.

The common counter-argument to extra supervision is its lack of scalability: if we want to train classifiers for thousands of visual categories, it would be difficult to add keypoints to each training example of each object class. This argument, while still valid, is less convincing today than it was ten years ago due to the emergence of crowdsourcing. Using Amazon Mechanical Turk we were able to annotate tens of thousands of examples of the 20 visual categories in the PASCAL competition in less than a week and using less than $500.

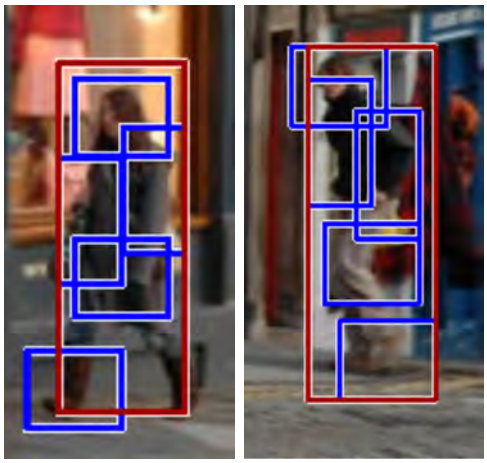

Figure 1.2: The Latent SVM model by Felzenszwalb et al.

1.2 What parts should we search for?

The second key difference of our work from traditional approaches is our choice of parts, which are inspired by the extra supervision. While holistic detectors have been effective for detecting faces [45] and pedestrians [11],[35], the variation in viewpoint, articulation and the presence of occlusion associated with closeups of people have led to the development of part-based approaches, which tend to fall into two main categories:

1. Work in the pictorial structure tradition, from Felzenszwalb and Huttenlocher [17] and others [37, 36, 18, 1], picks a natural definition of part in the 3D configuration space of the body, guided by human anatomy. Even earlier work with "stick figure" representations using generalized cylinders to model various body parts made essentially the same choice [34, 40]. While these parts are the most natural if we want to construct kinematic simulations of a moving person, they may not correspond to the most salient features for visual recognition. It may be that "half of a frontal face and a left shoulder" or "the legs of a person making a step in a profile view" are particularly discriminative visual patterns for detecting a human–does it matter that these are not "parts" in an anatomical sense, or that English doesn't have single words for them?

2. Work in the appearance-based window classification tradition. A flagship example is the aforementioned work of Felzenszwalb, McAllester and Ramanan [16] who allow an intermediate layer of "parts" that can now be shifted with respect to each other, rendering the overall model deformable. The templates for these parts emerge as part of the overall discriminative training. Such approaches, however, are not suitable for pose extraction or localization of the anatomical body parts or joints. An alternative way to provide flexibility is by the use of point descriptors as in the work of Mori and

Malik [33], or Leibe *et al.* [27]. What is common to all these approaches is the parts or point descriptors are chosen based purely on appearance.

Finally, there are now some hybrid approaches which have stages of one type followed by a stage of another type. Ferrari *et al.* [18] start with holistic upper-body detection based purely on appearance, followed by the application of a pictorial structure model in regions of interest. Andriluka *et al.* [1] train part detectors for anatomically defined body parts which then are combined using pictorial structures. However, none of the previous approaches use parts that emerge from strong supervision.

Figure 1.3: Examples of poselets

Examples of our parts, the poselets, are shown on Figure 1.3 and the training algorithm is described in Chapter 3. As the figure shows, poselets capture part of the pose at a given viewpoint. They have clear semantics that can be described with a noun phrase. The first poselet corresponds to "the head and shoulders of a person whose left hand is lifted near the face" and the last is "a frontal view of the torso of a person with their hands crossed". Notice that poselets do not correspond to individual anatomical parts; a poselet can capture a local configuration of anatomical parts. This results in patterns that have rich structure and are therefore easier to discriminate. Notice also that the examples of a poselet are not necessarily very similar visually. For instance, the second and the fourth example of a

back-facing poselet (3rd row) are visually very different even in the gradient domain. People have different hair styles, they wear different clothes and appear on different backgrounds. If we were to construct our parts in an unsupervised way, entirely based on appearance, it would have been impossible to keep these varied examples within the same poselet without including lots of visually similar but unrelated patterns. The extra supervision allows us to have a clean training set of examples, and *it allows us to train classifiers that learn the visual differences associated with a common underlying semantics.*

That latter property of poselets means that, for example, a frontal head-and-shoulders poselet will fire for people with long or short hair, with or without sunglasses and with or without a hat as long as they are facing the camera, but it will not fire if the same person is looking sideways. This allows us to use poselets as an engine to decompose the pose from the appearance, which is key to many high-level vision tasks. For instance, the pose is a latent parameter in detection; we would like to detect objects regardless of their pose and camera viewpoint. Pose is key for pose estimation and action recognition – to find out whether a person is facing the camera, or if they are reading a book, we need to find out their articulation and pose regardless of the types of clothes they are wearing. Attribute classification, on the other hand, treats pose as a latent parameter and requires discriminating the variations in appearance associated with a given pose: to determine whether a person wears glasses we would need to train glasses detectors for a variety of poses. As we show in Chapters 4 and 5 poselets are effective in all these visual tasks.

Chapter 2

The H3D dataset

In this section we introduce our strongly-supervised dataset of people H3D, Humans in 3D, which allows us to train poselets.

2.1 The role of datasets in computer vision

Datasets are important for making progress in computer vision research. One might argue that the CMU-MIT Face Dataset[1] contributed to significant research in face detection and was central to solving the frontal face detection problem, and the Berkeley Segmentation Dataset [31] was key to the progress in bottom-up segmentation research. The Internet revolution gave us access to millions of images and allowed for new kinds of datasets. Torralba et al. demonstrated that with a large enough dataset even hard vision problems can be trivially solved using a nearest neighbor classifier [44].

Another advantage of the Internet era is the possibility for collaboration on a large scale, which could be used for construction of large datasets at very low cost or even for free. LabelMe [38] is a dataset of images with regions labelled by the computer vision community at large. Currently it includes more than 50000 annotated images. Google Image Labeller [2] is an effort by Google to create a large collection of images labelled by people. To encourage people to label images they turned the task into a game where each person is rewarded points for using the same labels chosen by other annotators. Sorokin and Forsyth [42] proposed using Amazon's Mechanical Turk for generating large datasets. They report that annotations can be generated for as little as $0.01 per image. Having web users annotate the images is appealing, as long as one can enforce quality control. LabelMe uses multiple annotations per image and also measures precision by analyzing the level of detail of the annotated region. Mechanical Turk allows the dataset administrator to review each annotation and decide whether to pay for it or not.

[1] `http://vasc.ri.cmu.edu/idb/html/face/frontal_images`

[2] `http://images.google.com/imagelabeler`

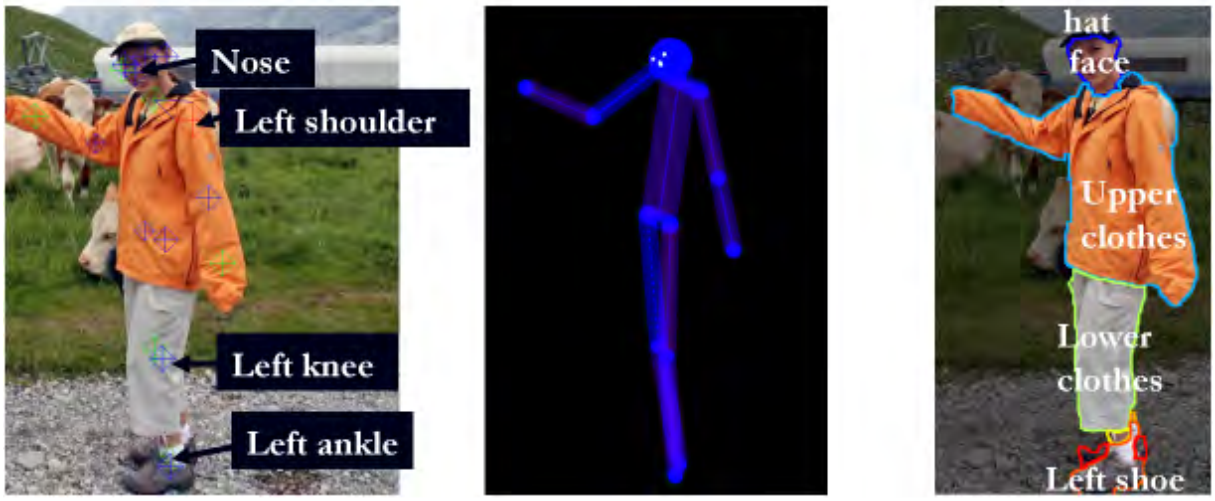

Figure 2.1: Examples of annotations from the H3D dataset. **Left:** Keypoint annotations. **Center:** Inferred 3D pose. **Right:** Part labels

2.2 The Humans in 3D Dataset (H3D)

The existing datasets vary significantly in their size, degree of annotations and annotation quality. Our emphasis is on the degree of annotation and quality. We believe that the higher the degree of supervision the better. Our dataset currently consists of 7053 annotations coming from 3250 images. These include the PASCAL trainval 2009 images as well as approximately 500 additional higher resolution images of people collected from Flickr. We have provided the following annotations as shown on Figure 2.1:

1. **Keypoint annotations.** We have annotated the following 20 keypoints: The eyes, nose, ears, back of head, shoulders, elbows, wrists, hips, knees, ankles and toes. For each keypoint we have specified the coordinates as well as the visibility status. Occluded keypoints are specified as long as their location can reasonably be estimated.

2. **Depth information.** We have connected the keypoints into a skeleton and have specified the relative depth of each pair of keypoints connected by an edge. This allows us to infer the 3D pose of people using a variation of the Taylor algorithm [43]

3. **Part labels.** For 1217 of our annotations we have also provided labelled segmentations of the body parts. The image is segmented into regions and regions are marked with one of the following 20 labels: occluder, face, sunglasses, hair, neck, hat, upper clothes, lower clothes, bag, dress, left/right arm skin, left/right leg skin, left/right sock, left/right shoe, left/right glove.

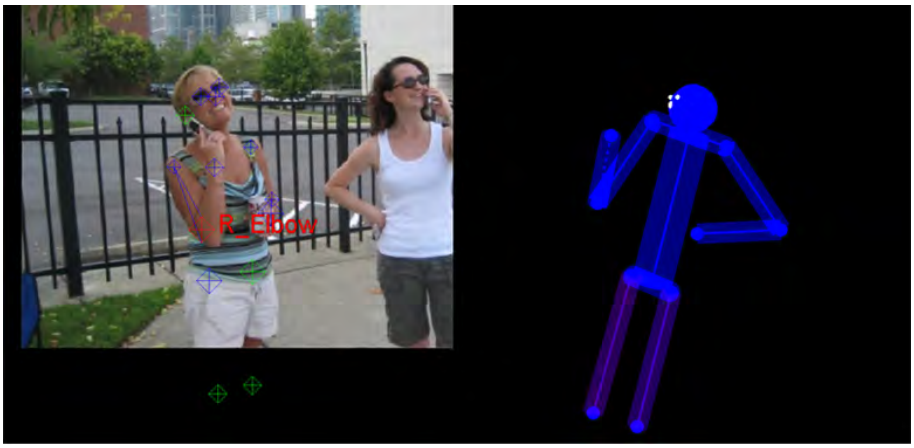

Figure 2.2: Our Java3D annotation tool allows the user to mark keypoints and displays the 3D pose in real time. Users can mark the image while looking at the 3D pose from another view. Our 3D extraction is based on the Taylor method [43] which we have extended to further ease accurate reconstruction. We have introduced extra constraints and global controls that use gradient descent in the overconstrained space to help the user adjust the pose, straighten the torso, etc.

4. **Gender/age.** We have categorized the annotations into three cases: man, woman and child. This categorization allows us to use a different parameters resulting in higher accuracy of our 3D estimation.

To ensure high quality and consistent bias we have trained paid annotators and we have manually verified and corrected the annotations. We would like to thank Hewett Packard for sponsoring the creation of H3D.

2.3 The Human Annotation Tool

Our annotation environment consists of two main modules - a 3D pose module that allows us to navigate the dataset, specify locations of joints and derive the 3D pose of people, and a region labelling module that segments the human body and allows us to label the associated regions. The 3D pose module is a Java 3D application shown on Figure 2.2. The left part of the screen shows the image containing a person and the locations of the joints and the right part shows the 3D pose.

2.3.1 Interaction Model

To label a person, the user can zoom and scroll the left image and then place and adjust the keypoints associated with the shoulders, elbows, wrists, hips, knees, ankles, as well as

the ears, eyes and nose. The names of the keypoints are shown when the mouse hovers over them. The user can also mark joints as visible or occluded. Occluded joints and joints outside the image are also annotated, as long as we have a rough idea where they are.

Most joints have an associated 'parent' joint. For example, the parents of the ankles are the knees, and their parents are the hips. The user can also specify whether a given joint is further away, roughly equidistant, or closer to the camera relative to its parent joint. This information is necessary for 3D reconstruction.

As the user labels joints in the left window, our tool provides real-time 3D pose estimation in the right window. The user can orbit the right window and view the 3D pose from an arbitrary view and see how changes in the keypoint locations of the 2D image affect the 3D pose. Figure 2.2 shows how the user is adjusting the right elbow while looking at the pose from the left profile. The figure also shows that the user is dragging the right elbow (highlighted in red). A perspective link connects the right elbow to the right shoulder indicating that the shoulder is the parent of the elbow and that the shoulder is further away from the camera. Note that the knees fall outside the image and are marked in green, which means they are occluded.

2.3.2 3D reconstruction

Most existing 3D datasets of people consist of motion capture data. One reason is that full 3D modeling is a very time consuming task. Making 3D modeling easier is critical for creating large 3D datasets and we have put a lot of effort to ease the process.

Luckily, for the case of people, Taylor [43] has proposed an algorithm that can recover the 3D pose from little more than the 2D locations of joints. In our project we have used and extended Taylor's algorithm, which we briefly describe here. To infer the 3D pose, the algorithm assumes a scaled orthographic projection and also that all people have the same body proportions. We have found that these assumptions are reasonable in practice[3]. These two assumptions allow us to reconstruct the 3D length of segments up to an unknown scale s. Using the 2D coordinates of joints we can compute the image-space length of a segment. The relative depth of the two endpoints of the segment can then be computed using the Pythagorean theorem:

$$l^2 = (X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2 \quad (2.1)$$

$$(u_1 - u_2) = s(X_1 - X_2) \quad (2.2)$$

$$(v_1 - v_2) = s(Y_1 - Y_2) \quad (2.3)$$

$$Z_1 - Z_2 = \sqrt{l^2 - ((u_1 - u_2)^2 + (v_1 - v_2)^2)/s^2} \quad (2.4)$$

[3]Note that excess body mass does not materially affect the skeletal structure and thus the body proportions. Our tool also allows the user to categorize the body as male, female or baby and we have separate body proportions for each class.

The sign of the relative depth cannot be derived by the above formulas and we ask the user to specify whether a point is closer or further away from the camera relative to its parent point. In the future we could determine a good default of the sign based on statistics of poses. While the minimum scale s is unspecified, we have lower bounds for it because the 3D length cannot be shorter than the 2D length. For every segment the following must hold:

$$s \geq \frac{\sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2}}{l} \tag{2.5}$$

Setting s to its lower bound is reasonable. That implies that at least one of the segments is parallel to the image plane, which is almost always the case for typical 3D poses. We refer to that segment as the critical segment and we show it with a dashed line in the 3D view (Figure 2.2 top-right) because small changes to it make big changes to the overall pose.

While in theory Taylor's algorithm is sufficient to reconstruct the 3D pose, we found that in practice it is difficult and time consuming to get the pose just right, because the algorithm has the same number of constraints as variables and doesn't allow for any margin of error. We introduced additional constraints to ease 3D reconstruction. In particular, we allow segments to be marked as equidistant to the camera. In practice, frontal views of upright people have many segments parallel to the image plane. These extra constraints allow the tool to determine whether some segments are too short. It marks segments whose size is inconsistent as red, with intensity proportional to the degree of inconsistency. We also provide three global controls that the user can apply by pressing and holding a key. We adjust the 2D coordinates by following the gradient descent of the following energy function:

$$x \longrightarrow \arg\min_x \sum_i w_i(\tilde{x_i} - x_i)^2 + F_j(x) \tag{2.6}$$

where $\tilde{x_i} = (u_i, v_i)$ denote the human marked 2D coordinates, x_i are the true 2D coordinates and w_i are the weights associated with the data fidelity terms. We use different weights depending on the type of joint. For example, annotators are less accurate in specifying hips than they are in specifying wrists. We also take into account the visibility: a keypoint marked as occluded is less precise than a visible keypoint. F_j is the specific control method:

- Control 1 changes the length of the critical segment and affects the degree of foreshortening of the pose.
- Control 2 moves keypoints of segments marked as parallel to the image plane a small amount in a direction that makes their 2D lengths closer to their 3D lengths.
- Control 3 straightens up the torso. It moves the hips and shoulders in a direction that moves the torso towards an isosceles trapezoid. Most of the time people's torsos are not twisted but it is hard to get straight torsos without using this explicit control.

Since the 3D pose is defined up to a similarity transform, we subtract similarity transforms from all of the optimizations and adjust the keypoints only by the remaining error.

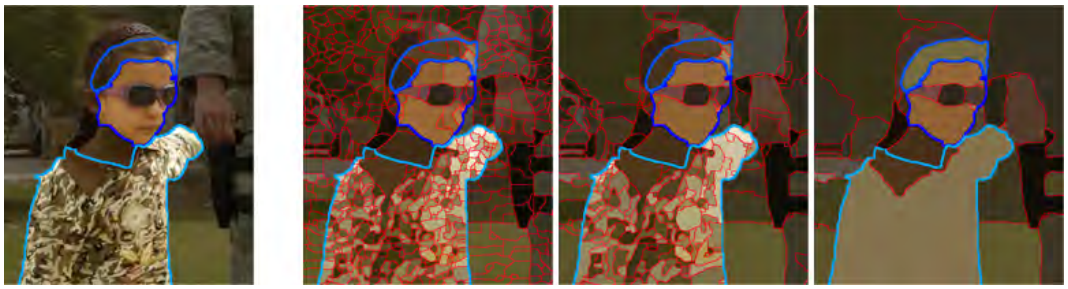

Figure 2.3: Our region labeling tool performs hierarchical oversegmentation of the image using [28] to allow the user to efficiently and accurately assign region labels. Users start labelling a rough version and refine the labels.

2.3.3 Region Labeling Tool

Our labeling tool allows us to label the pixels of the image with labels such as "left hand", "upper clothes", "face", etc. It is important to have high quality pixel-level masks while allowing for fast labeling. To achieve this goal we segment the image into superpixels using [28] and allow the user to label superpixels instead of pixels. This significantly speeds up the labeling while achieving high quality results, but is not sufficient for large images that have thousands of superpixels. We construct a hierarchy of the superpixels based on the strength of the edges between superpixels. Using a slider the user can select a suitable level of the hierarchy which corresponds to the level of detail for segmentation. In a typical workflow the user starts with rough segmentation, labels large regions and then increases the resolution and refines the labels. Figure 2.3 shows our labeling tool at three levels of segmentation.

The time to create an annotation varies on the difficulty of the annotation and the expertise of the annotator, but on average it takes in the order of five minutes to specify the keypoints, set the 3D pose and label the regions.

2.4 Applications of H3D

In this section we give examples of the types of queries H3D allows for.

2.4.1 Viewpoint Extraction

Since our 3D poses are defined up to a scale, we cannot distinguish between distance from the camera and the size of the person, but we can extract the camera viewpoint (azimuth and elevation angles). To compute the viewpoint we define the center of the human to be the midpoint of the shoulders and we let the X-axis be along the shoulder line. The Y-axis is in a direction orthogonal to the X-axis and lies in the same plane as the midpoint of the

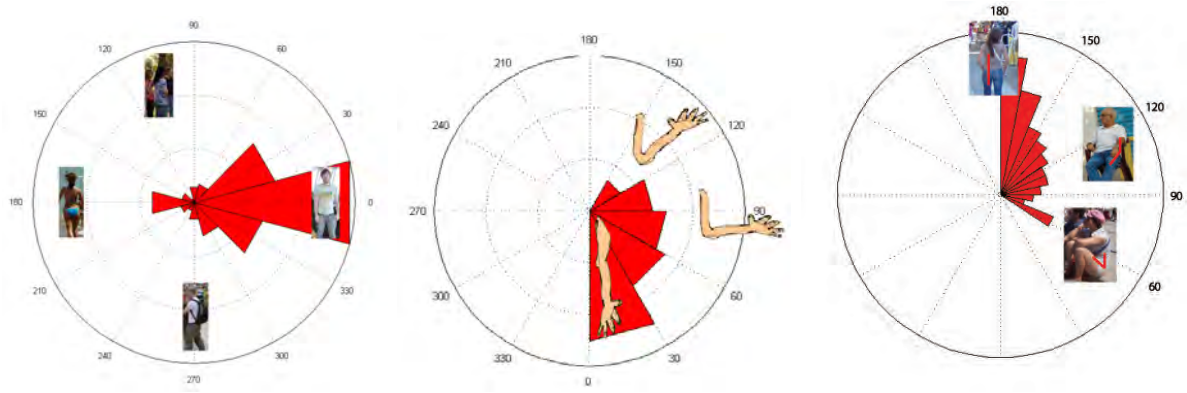

Figure 2.4: Examples of 3D pose statistics made possible by H3D. **Left:**Camera azimuth angle relative to frontal pose.**Center:**Expected bending angle of the left arm. **Right:** Expected bending angle between the left thigh and the torso (log polar graph). This angle typically distinguishes sitting from standing people.

hips. Figure 2.4 (left) shows the camera azimuth distribution. We see that 39.6% of the time the human pose is frontal (between $-15°$ and $15°$).

2.4.2 3D Pose Statistics

In Figure 2.4 (center) we have explored the distribution of angles of between the upper and lower arm segments. As the figure shows, physical constraints of the body are implicitly incorporated into our statistics - there are no arms that bend backwards. Also the figure shows that 33% of the time the arm is almost straight, bent less than 30% degrees. Figure 2.4 (right) shows the expected angle between the left thigh and the torso. It is a log-polar graph; 63% of the people in our dataset have this angle almost flat (between 175 and 185 degrees) which usually means that they are standing. The figure again shows how the human body limits are incorporated into our statistics, as people cannot bend their legs backwards. Note that these statistics are performed in 3D space. Most other datasets do not contain 3D information, which would make it impossible to derive these statistics due to foreshortening ambiguities.

Using the annotated keypoint locations, we can determine the expected image locations of a set of keypoints conditioned on the locations of other keypoints. Such distributions would be valuable for any part-based human detector. Figure 2.5 (left) shows our prediction for the locations of the left ankle and right elbow conditioned on the shoulder locations.

We can also use H3D to determine the label probability of pixels conditioned on any property we want. For example, Figure 2.5 (right) shows the expected location of the upper and lower clothes conditioned on the location of the two eyes and of the two hips. As expected, the label prediction probability is higher near the fixed points and goes down for

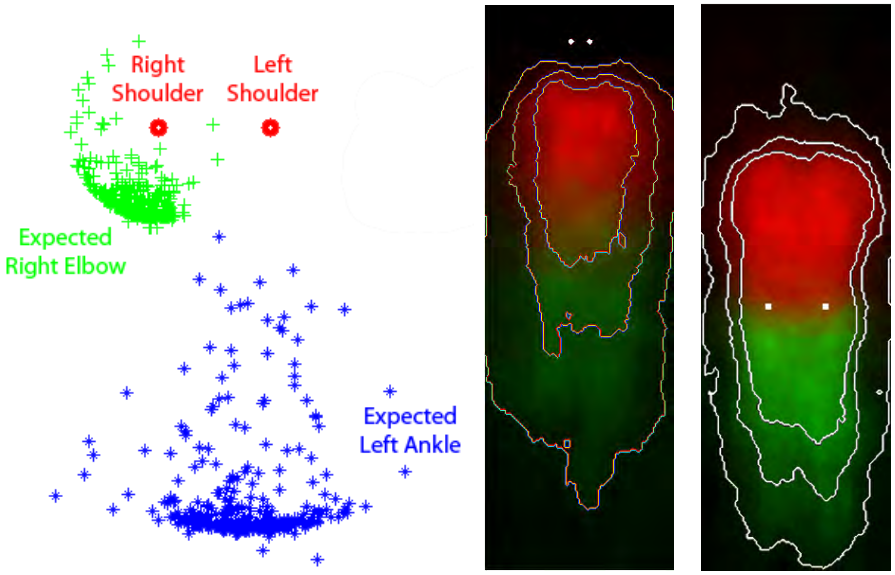

Figure 2.5: **Left:** Scatters of the 2D screen locations of the right elbow and left ankle given the locations of both shoulders. **Right:** Probability of upper clothes (red) and lower clothes (green) given the location of the eyes (left picture) or hips (right picture). Contours at 0.1, 0.3 and 0.5 are shown.

pixels further away. We could, of course, compute conditional probabilities on more variables if the size of our dataset allows for meaningful predictions.

2.4.3 Appearance Queries

Registering 3D views with 2D images is powerful, as it allows us to query for the appearance of parts. Given the normalized locations of two keypoints (which define a similarity transform), a target aspect ratio and resolution, H3D can extract patches from the annotated images.

For example, to generate examples of raised hands (Figure 2.6 left) we specify the left wrist to be in the center of the patch, the left elbow to be vertically below it (outside the patch) and request patches for which the out-of-plane angle is small (to avoid hands pointing towards or away from the camera).

H3D can leverage our region annotations to include or exclude specific regions. Figure 2.6(center) shows an eerie collection of frontal heads with the faces masked out. Figure 2.6 (right) shows the result of displaying people whose hip-to-torso angle is less than 130 degrees (i.e. sitting people), sorted by the hip-to-torso angle (i.e. most sitting to least sitting). We show them with the background and any occluders masked out.

We can use H3D to learn the prior probability of the color of each part. For example Figure 2.7 shows a GMM fit for the colors of skin, hair and upper clothes.

Figure 2.6: Examples of appearance queries generated by H3D.

Figure 2.7: Means and proportions of Gaussian Mixtures trained to fit the colors of skin, hair and clothes. Variances not shown.

2.5 Conclusion

Combining appearance, 3D pose and viewpoint allows for a virtually unlimited set of questions we can ask H3D for. One can compute statistics for questions such as: "How often are both hands occluded?", "What fraction of the women are blonde?" or "How often do sitting people wear a hat?". Each of the examples in this section was done using less than half a dozen lines of Matlab via the H3D toolset.

The H3D dataset, the Matlab H3D toolbox and the Java3D annotation tool are available on our website, together with a video tutorial and the associated papers. Poselets are one example of a novel computer vision part that is made possible by H3D. We hope that our richly supervised dataset will inspire other new methods and new ways of thinking about data.

Chapter 3

Poselets

In this chapter we describe an algorithm for training poselets and for combining them to form object hypotheses.

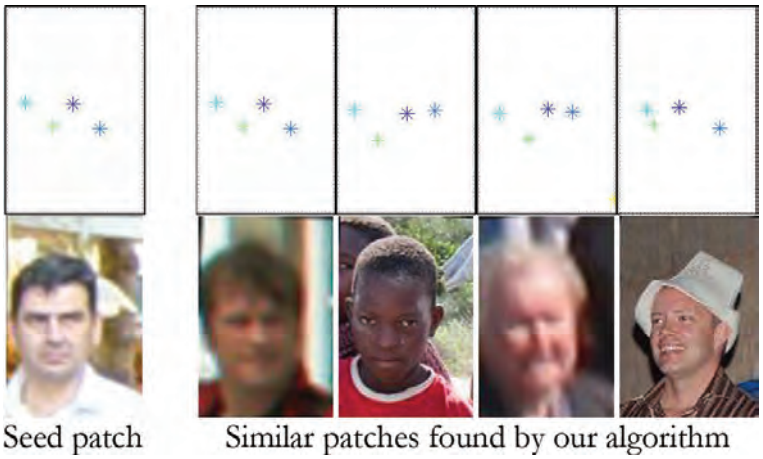

Figure 3.1: A poselet describing a frontal face and five of its examples. **Top row:** The configuration spaces showing the eyes, nose and left ear keypoints. **Bottom row:** the corresponding image patches. By construction all examples of a poselet have similar configurations and are therefore semantically similar.

3.1 Finding corresponding patches

3.1.1 Distance in configuration space

A poselet is specified with an image patch, such as a frontal face poselet shown on the left of Figure 3.1. The first step in training a poselet is finding positive training examples, which

are patches similar in configuration space. In the example on Figure 3.1 we find four similar patches by transforming the annotated people in our training set so that the locations of their eyes, nose and left ear match those in the seed, as shown in the first row of Figure 3.1.

Specifically, given the local keypoint configuration within a seed window, we extract patches from other training examples that have similar local keypoint configurations. Following [5] we compute a similarity transform that aligns the keypoints of each annotated image of a person with the keypoint configuration within the seed window and we discard any annotations whose residual error is too high. We use the following distance metric which we introduced in [4]:

$$D(P1, P2) = D_{proc}(P1, P2) + \lambda D_{vis}(P1, P2), \tag{3.1}$$

where D_{proc} is the Procrustes distance between the common keypoints in the seed and destination patch and D_{vis} is a visibility distance, set to the intersection over union of the keypoints present in both patches. D_{vis} has the effect of ensuring that the two configurations have a similar aspect, which is an important cue if 3D information is not available. We empirically set the tradeoff between the two criteria to $\lambda = 0.1$. This equation is efficient to compute as the Procrustes distance involves solving a least squares system. Thus we can find all patches similar to the seed, searching over thousands of training examples, in the order of a couple of seconds.

One issue is whether to use a full similarity transform, or whether to restrict it to just rotation and scale. The advantage of using a full similarity transform is that we can find more training examples: For instance, to train a tilted head poselet we don't have to restrict our examples to people whose heads are tilted; we could also rotate upright faces. On the other hand, some visual categories have strong rotational prior (we rarely see cars rotated at 45 degrees). In such cases disabling the rotation can improve the match, because rotation is misused to minimize the distance between keypoint configurations of different aspects. It is better to keep parts from different aspects in different poselets. Currently we have a flag for each visual category. We allow rotation for people, birds, cats, etc, and we don't allow it for "grounded" categories, such as cars and buses.

Equation 3.1 is only one possible way to define the correspondence between patches. Variations of this equation involve using a different weight for each keypoint, for example by giving higher weight to keypoints closest to the center of the patch and giving non-zero weight to nearby keypoints that fall outside of the patch, or using distance in 3D if available [5]. We could associate a penalty if the visibility status of keypoints does not match, or we could take into account other optional cues, such as aligning the outlines of the two objects. The specifics of the distance transform are not important; what is key is that we are defining a notion of a distance between pairs of patches that would allow clustering in configuration space.

3.1.2 Determining the right number of training examples

Our distance transform is defined for every training example, but we certainly don't want to use as training examples patches that vary significantly from the seed. While the residual error D is a reliable way to order the patches from most to least similar, we need to find a threshold of the residual error above which we would discard the examples. Determining this threshold is not an easy task. There is no universal threshold that works for all poselets; the distance metric D is not on the same scale for different poselets. If we use a threshold that is too small we would have very few training examples and our poselet classifiers will not train well. Using too large a threshold would result in including bad examples into the training set and would similarly result in suboptimal training.

Figure 3.2: Example of the swamping phenomenon for a profile face poselet. The examples are shown sorted by residual error. The rank of each example is indicated.

There is a more subtle phenomenon here which we call *swamping* of a poselet. It is illustrated on Figure 3.2. The figure shows patches corresponding to a profile face poselet sorted by increasing residual error. The first hundred or so examples are good profile faces, but once we run out of profile faces we start including tilted frontal faces. They are good matches because the right eye and the nose are at the same position as in a profile face. Since there are an order of magnitude more frontal faces, if our threshold allows many of them to be included they will end up dominating the training examples and we will end up training a poselet to respond to tilted frontal faces as opposed to profile faces. Swamped poselets are harder to train and have poor pose estimation capabilities.

To avoid swamping we use the following strategy: We first use a conservatively low threshold that guarantees there is no swamping. Empirically we found that the first $k_1 N$ examples have no swamping, where N is the size of our training set and k_1 is set to 3%. We train poselets using the smaller set of "unswamped" examples. We then evaluate them on the training set and count the number of patches they are able to find. Let that number be P. In the example on Figure 3.2 we would find the number of profile faces in our training dataset. We then do a second round of training of the poselet using exactly P training examples. Effectively we are bootstrapping the poselets based on the appearance of the unswamped examples.

3.2 Training a poselet

The previous section describes a method for generating positive training examples for a poselet. We generate negative examples by randomly sampling patches from images that don't contain the visual category. We then compute features of each patch and train a classifier. Our features of choice are Histograms of Oriented Gradients first introduced by Dalal and Triggs [11] and we use a linear SVM classifier[1].

After the initial training round, we perform a bootstrapping procedure: We run the classifiers on a set of images not containing the object in question, we collect the false positives with highest scores and we retrain the classifier. We perform this bootstrapping procedure on the training images in the PASCAL dataset that don't contain instances of the object we are training to detect.

Due to the disproportional size of negative vs positive training examples, we found that the training procedure does not set the SVM bias threshold reliably. We would like a threshold that is not too low (so we detect instances of the pattern) but not too high (so we don't generate too many false positives). We found that the optimal way to set the threshold is to run the poselet classifier in a window scanning manner on the training set, collect the top activations and set the threshold so that precisely $k_2 N$ of them have a positive score. We used $k_2 = 2$.

3.3 Poselet selection

The previous two sections provide an algorithm that given a seed image patch would train the corresponding poselet classifier. In this section we describe a method for selecting suitable poselets. A good set of poselets must satisfy the following criteria: Each poselet must be trained well, the poselets must be complementary, they must provide a good "coverage" of the training examples and they must be as few as possible. We do poselet selection as follows:

1. We pick 1200 random seed patch windows from the training images. We make sure that each window sufficiently overlaps a training example. We sample with uniform distribution over location and log scale. This results in higher probability of sampling patterns that are common, such as frontal faces. We take equal number of samples from each of the following normalized dimensions: 64x96, 64x64, 96x64, and 128x64 pixels.

2. We then use the algorithm in the previous sections to extract training patches, threshold them, and train poselet classifiers for each of the seed patch windows.

[1] While we might train better with more powerful classifiers, we use linear SVMs due to their simplicity and speed.

3. We apply the poselets in a scanning-window fashion on our training set and collect their top $k_2 N$ activations.

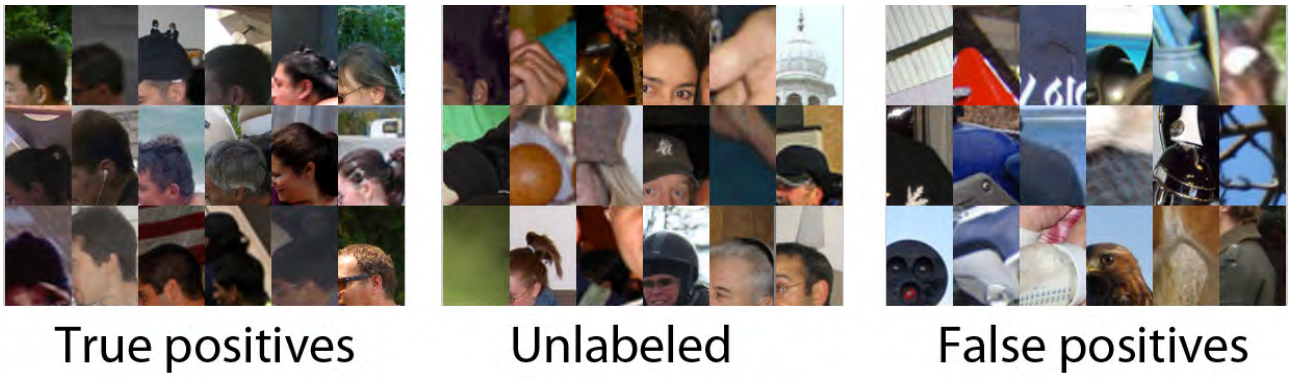

Figure 3.3: Examples of a left-back-head-profile poselet activations classified into true positives, false positives and unlabeled.

4. We assign labels (true positive, false positive, unlabeled) to each poselet activation (Figure 3.3). To assign a label we use the bounds of the patches we extracted in step 2 and their rank according to the distance metric (equation 3.1) with disabled rotation. We partition the bounds into two classes: the top-rank patches (which we used for training) are treated as ground truth; the lower-rank patches are treated as secondary ground truth. Any activation that has intersection over union overlap of more than 0.35 with a ground truth is assigned a true positive label. If the overlap with a secondary ground truth is less than 0.1 or none, it is assigned a false positive label. All other cases remain unlabeled.

5. The previous step determines if a poselet activation is a true positive and, if so, which training example it activates on. We now construct a coverage matrix that captures which poselets activate on which training example. That is, entry $c_{i,j}$ is true if and only if poselet type i activates on a training example j (Figure 3.4).

6. We use a greedy selection algorithm (Algorithm 1) to pick a small set of poselets that cover as many training examples as possible while allowing for some controlled redundancy:

This algorithm starts with the poselet that covers the largest number of examples and after that picks poselets that cover the largest number of so-far-uncovered examples. For the person category we used $N = 1200$, $T = 150$ and $\gamma = 5$. The first ten poselets chosen for the person category are shown on Figure 3.5. The algorithm prefers head-and-shoulders poselets as they train well and cover a large number of the training examples. Other high ranking poselets are a frontal face and a pedestrian. Leg poselets come later in the ranking as they

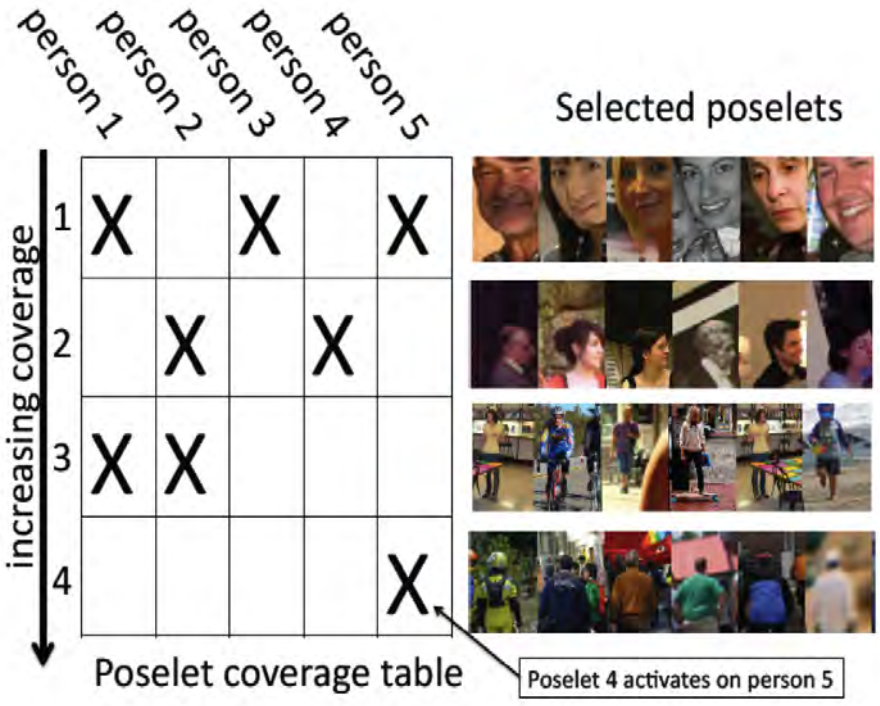

Figure 3.4: Poselet coverage matrix.

Algorithm 1 Poselet selection algorithm

Input: $N > 0$ Number of poselet candidates
$T \in [1, N]$ Number of poselets to select
$\gamma > 0$ Smoothing constant
$c_{i,j}$ Coverage table, $c_{i,j} = true$ iff poselet i activates on example j
Output: M : a set of selected poselets
$\forall j, r_j \Leftarrow \gamma$
$M \Leftarrow \emptyset$
for $t = 1$ to T **do**
$k \leftarrow \underset{i \in [1,N]: i \notin M}{\operatorname{argmax}} \sum_{j: c_{i,j} = true} r_j$
$M \leftarrow M \cup \{k\}$
for all j such that $c_{k,j} = true$ **do**
$r_j \leftarrow \max(0, r_j - 1)$
end for
end for

are harder to train and have weaker coverage (in many cases the legs are occluded.) Hand poselets are rare.

For the person category, our training set is H3D trainval + PASCAL train09 (a total of 3809 training examples). The first 10 poselets cover 63.8% of the training examples and the first 150 cover 76.8%. The maximum coverage of 77.5% of the training set would be achieved using 245 poselets. It is important to note that a poselet activated on a person is necessary but not sufficient for us to detect that person. Person detection, as defined in the PASCAL competitions, requires correctly predicting the visible bounds of the person.

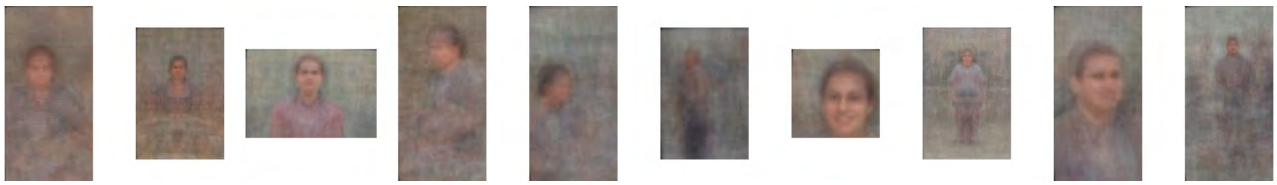

Figure 3.5: The first ten poselets for the person category. Each poselet is represented as the mean of its top 40 training examples.

3.4 Training poselet prediction parameters

Given poselet i and its labelled activations on the training set we train the following:

1. We fit a logistic over the positive and negative activations and the associated scores to convert SVM scores into probabilities q_i.

2. We set a threshold for the SVM score that ensures 90% of the positive and unlabeled examples are above the threshold. This allows each poselet's detection rate to match the frequency of the pattern it has learned to detect.

3. We fit a model for the keypoint predictions conditioned on each poselet by observing the keypoint distributions of the true positive activations of each poselet type. An example is shown in Figure 3.6. We model the distributions using a 2D Gaussian associated with each keypoint.

4. We fit the prediction of the visible bounds of the human relative to the poselet in a similar way using the true positive activations. We find the mean and variance of x_{min}, y_{min}, x_{max}, and y_{max} of the visible bounding box.

5. We fit a foreground probability mask: Using the foreground/background region annotations in the training images we estimate the probability of each pixel within the poselet patch to be an object pixel *vs.* one coming from the background. This mask (shown on Figure 3.12) will be useful in the segmentation task.

Figure 3.6: Empirical keypoint distribution: locations of the shoulders (left), shoulder and ear (middle), and shoulders and hips (right) over true positive poselet activations of three different poselets.

3.5 Consistent poselet activations

One of the key steps of processing poselet activations in a test image is determining whether two activations refer to the same object instance. Two activations are called *consistent* if they are true positive and detect parts of the same object instance. Consistent activations are often nearby in space, but sometimes they can be far apart: a frontal face and legs activations can be consistent while being spatially far apart.

We measure consistency between two poselet activations using the symmetrized KL-divergence of their empirical keypoint distributions $\mathcal{N}_i^k$ and $\mathcal{N}_j^k$:

$$D_{SKL}(\mathcal{N}_i^k, \mathcal{N}_j^k) = D_{KL}(\mathcal{N}_i^k || \mathcal{N}_j^k) + D_{KL}(\mathcal{N}_j^k || \mathcal{N}_i^k) \tag{3.2}$$

$$D_{i,j} = \frac{1}{k} \sum_k D_{SKL}(\mathcal{N}_i^k, \mathcal{N}_j^k) \tag{3.3}$$

Since we represent these keypoint distributions as 2D Gaussians, D_{SLK} has a closed-form solution. Since not all keypoints are available for all poselets, we average over those keypoints predicted by both poselets.

We treat two activations i and j as consistent if $D_{i,j} < \tau$. We set τ as the threshold that best separates distances among consistent activations from distances among inconsistent activations on the training set. Note that for all pairs of labeled activations on the training set we can determine whether they are consistent or not - namely, two activations i and j are consistent if they are both true positive and refer to the same training example. An example of consistent and not consistent activations is shown on Figure 3.7.

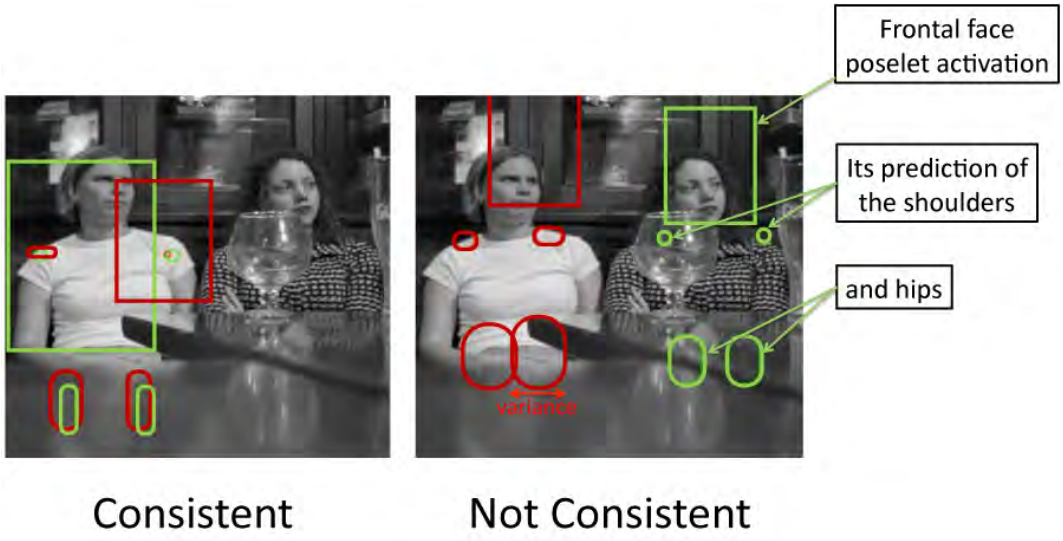

Figure 3.7: An example of two poselet activations that are consistent (left) and not consistent (right). Consistent activations have similar predictions for the locations of the keypoints. In this case we show their predictions of the hips and shoulders.

The symmetrized KL-divergence is an estimate of the likelihood that two poselet activations are consistent. It is not the only such estimate. In general one could define a distance by measuring the discrepancy in their predictions of various object properties. For example, each poselet activation can predict the visible bounds of its reference object. The intersection over union of the predicted visible bounds of two activations defines an alternative estimate. The intersection over union of the soft foreground masks predicted by the two activations define yet a third estimate.

3.6 Enhancing poselets using context

Poselet classifiers can make mistakes because the pattern can be hard to detect or it could be rare and we don't have enough training examples. There are also "near-metamers"; patterns that can be distinguished by a human observer using additional context, but are indistinguishable given the signal inside the image patch encoded in the HOG feature vector. For example, a back-facing head-and-torso pattern is similar in appearance to a front-facing head-and-torso pattern (Figure 3.8 left), and thus a back facing poselet will often fire on front-facing people as well. False positives are common (Figure 3.8 center). Another example is a left leg, which in isolation looks very similar to a right leg (Figure 3.8 right).

One can resolve these ambiguities by exploiting context – the signal within a patch may be weak, but there is strong signal outside the patch or at a different resolution. This is no different from the case of object detection where the surrounding scene – nearby objects –

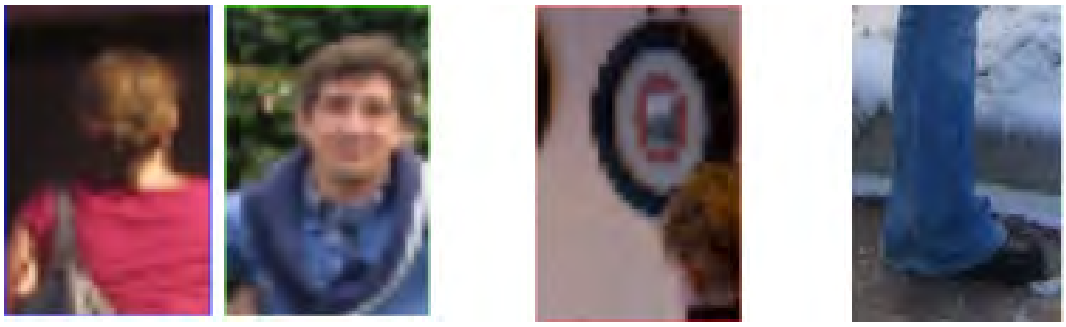

Figure 3.8: Poselet classifiers can make mistakes for many different reasons. **Left:** The head-and-shoulders pattern looks similar for front and back facing people. **Center:** An oval shape is often mistaken for a frontal face. **Right:** Due to body symmetry it is hard even for a person to determine if this is a left or a right leg.

provide the disambiguating signal. In our setup we use nearby consistent activations as a source of context. In the examples on Figure 3.8 the presence of a frontal face activation at the right location and scale is a strong indicator that the person is front-facing. The lack of head-and-shoulders poselet or a pedestrian poselet is an indication that the middle example is a false positive. The location of a pedestrian poselet can help disambiguate the left from the right leg.

We refer to the score of a poselet activation based on its classifier only as **q-score** and one enhanced by its consistent activations as **Q-score**. For each poselet activation i in the training set we construct a context feature vector f of size the number of poselet types. The value of f_p is the maximum q-score of any activation of poselet type p that is consistent with activation i (or zero if none). We train a linear SVM on the context feature vectors of activations in the training set using their true and false positive labels. We then fit a logistic to convert the SVM score into a probability. The result is what we call Q-score. Figure 3.9 shows an example of the effect our context classifier has on improving the activations and Figure 3.10 shows ROC curves for various poselets on the PASCAL val09 set which has not been used in training.

3.7 Combining poselet activations

Once we detect the poselet activations in the test image and enhance them using context, the next logical step is to cluster them so that all activations that refer to the same object are in the same cluster. Our initial approach in [5] is build upon the Max Margin Hough Transform from [30] in order to group poselet activations to consistent people detections. This comes with the assumption that the object has a stable central part and the relative position of all other parts has very small variance – an assumption that is not satisfied for articulated

Figure 3.9: The top activations of a poselet trained to find back facing people.**Top row:** Sorted by q score. **Bottom row:** Sorted by Q score.The correct activations have a green frame and the wrong ones have a red one. The Q-scores are computed using context to disambiguate front-facing from back facing people so nearly all top examples are correct, while the q-scores are based on the poselet classifier alone and result in 6/10 mistakes.

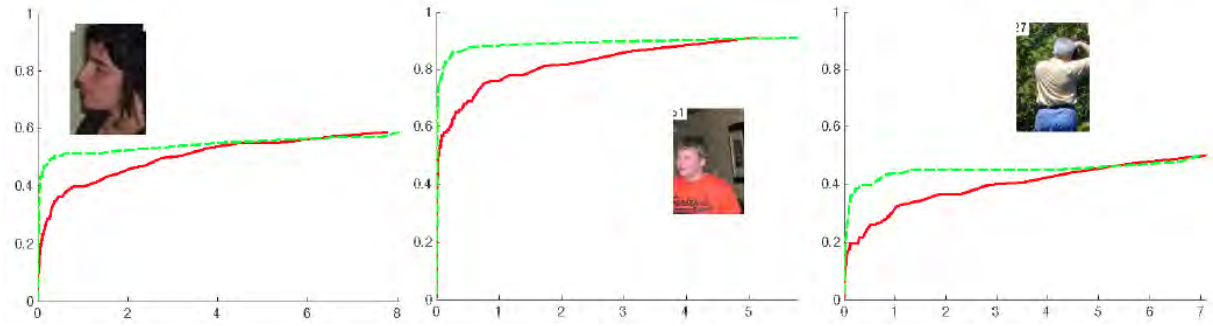

Figure 3.10: ROC curves of activations of three poselets on our validation dataset. Red continuous lines use q-score and green dashed lines use Q-score. These performance improvements are representative for the rest of the poselets.

objects, such as people. To this end we propose an alternative clustering algorithm:

Algorithm 2 Poselet clustering algorithm

Input: $T > 0$ Maximum number of hypothesis clusters
$\tau > 0$ Clustering threshold
a : A set of poselet activations in the image sorted by decreasing Q-score
Output: M : a set of activation clusters
$M_1 \Leftarrow \{a_1\}$
for $i = 2$ to $|a|$ **do**
$k \leftarrow \operatorname{argmin}_j d(a_i, M_j)$
if $d(a_i, M_k) < \tau$ **then**
$M_k \leftarrow M_k \cup a_i$ {Append the poselet to a current cluster}
else if $|M| < T$ **then**
$M \leftarrow M \cup \{a_i\}$ {Start a new cluster}
end if
end for
return M

In the end the poselet activations are grouped into clusters each corresponding to an object detection hypothesis. In addition some poselets with low scores that are inconsistent with any clusters are marked as false positives and are discarded. The parameter T is a tradeoff between speed and false positive rate. We set $T = 100$, i.e. we collect at most 100 hypotheses from each image. We use approximate average linkage, i.e. $d(a_i, M_j) = E_{a_k \in M_j} D_{a_i, a_k}$ where $D_{i,j}$ is the symmetrized KL-divergence (Equation 3.3). For performance, if cluster M_j contains more than K activations, we sample a random set of K of them. We found empirically that we can set $K = 5$ without affecting the accuracy.

Figure 3.11: Example of a detection cluster for a person. **Left:** The highest probability activation creates the first cluster. **Center:** The second activation is compatible so it falls in the same cluster. **Right:** The third activation is incompatible with the first two so it forms a second cluster.

This algorithm is a form of greedy clustering by considering the highest scored detections first. Compared to other schemes such as spectral clustering or agglomerative clustering, the proposed algorithm has computational advantages because it processes the most salient information first. The algorithm runs in linear time. We do not spend compute cycles measuring distances between low scoring detections, and the algorithm can be terminated at any time with a good list of the most-salient-so-far hypothesis M. Furthermore, by starting with the highest probability detections we are less likely to be mislead by false positives. Example of the clustering algorithm in action is shown on Figure 3.11.

This algorithm returns a set of clusters of poselet activations in the test image. A cluster of poselets is the output of the poselet detection algorithm and is the input to all poselet-based high-level vision methods as described in Chapter 4.

3.8 Poselets beyond people

We extended the above algorithm for training and detection of poselets to all 20 visual categories in the PASCAL challenge. The first challenge is deciding which keypoints to use. This is fairly straightforward for other animal categories but becomes more complicated for categories, such as a chair, a boat and an airplane, whose examples have large structural variations. There are chairs with four legs or one stem and a wide base. Some chairs have armrests, and others don't. Military airplanes look very different from commercial ones, and sail boats have little in common with cruise ships. We decided to split the categories into a few common subcategories and provide separate keypoints for each subcategory. This allows us to train separate poselets for the pointed front of a military airplane, the round tip of a commercial airliner and the propeller blades of a propeller plane.

The second challenge is that some categories do not have a principal orientation, which makes it difficult to assign keypoints in the reference frame of the object. For example, it is clear what the front left leg is in the case of a horse, but what is the front left leg of a table? Other categories have round parts and thus have no extrema points, such as the base of a bottle or a potted plant. Our solution in these cases is to introduce view-dependent keypoints. For example, we have a keypoint for the bottom left corner of a bottle, and we define the front left leg of a table based on the current camera view.

With the exception of the person category, our keypoint locations are defined in 2D. The absence of 3D information can cause certain ambiguities in configuration space. For example, a front and a back view of a bicycle will have the same coordinates of most of the keypoints (with the exception of the left/right handle which would switch places). Such ambiguities would result in a poselet containing examples of front and back views of a bicycle; something we certainly don't want to allow. To prevent such scenarios we annotate the view of our examples (front, left, right and back) and we disallow examples to match seeds of the opposing view.

As mentioned in Section 3.1.1 for some categories we have disabled rotation when search-

Class	#Keypoints	#Poselets	Rot	AR	Training set
Aeroplane	16x3	200	Y	64x128	trainval10
Bicycle	11	150	N	64x128	trainval10
Bird	12x2	200	Y	32x32	trainval10
Boat	11x5	200	N	64x128	trainval10
Bottle	8	100	Y	-	trainval10
Bus	8	100	N	64x128, 64x256	trainval10
Car	14	100	N	-	trainval10
Cat	16	150	Y	-	trainval10
Chair	10	100	Y	-	trainval10
Cow	16	150	N	64x128	trainval10
Dining table	8	100	N	-	trainval10
Dog	16	150	Y	-	trainval10
Horse	19	100	Y	-	trainval07+trainval09
Motorbike	10	100	N	64x128	trainval10
Person	20	150	Y	128x64	H3D+train09
Potted plant	6	150	N	-	trainval10
Sheep	16	150	N	-	trainval10
Sofa	12	100	N	64x128, 64x256	trainval10
Train	7	100	N	64x128, 64x256	trainval10
TV monitor	8	100	N	-	trainval10

Table 3.1: Class-specific variations in the poselet training parameters.**#Keypoints** is the number of keypoints, and the number of subcategories. **# Poselets** is the number of selected poselets. **Rot** denotes whether we fit over rotation when finding training examples of a given poselet. **AR** is class-specific poselet aspect ratios, in addition to the standard ones of 64x96, 64x64 and 96x64

ing for similar patches (Equation 3.1) as shown on Table 3.1. Lastly, the visual categories vary widely in aspect ratios and using poselets of a fixed size and aspect ratio is suboptimal. We extended the algorithm to support poselets of variable class-specific aspect ratios, as well as trained different number of poselets for each category. The differences are listed on Table 3.1. Examples of poselets from various categories are shown on Figures 3.12 and 3.13.

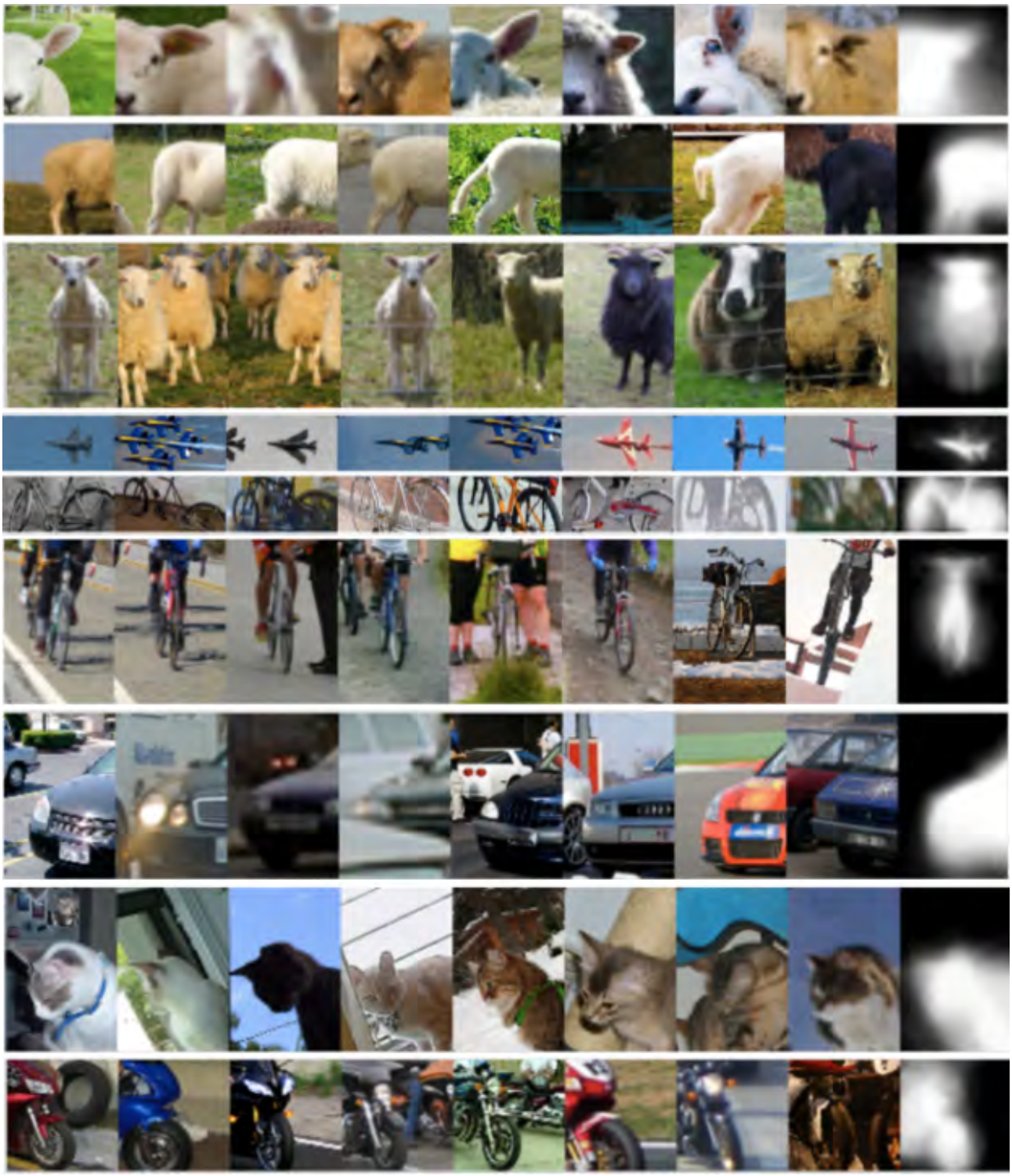

Figure 3.12: Examples of poselets from various visual categories. The last column shows the foreground probability mask.

Figure 3.13: More examples of poselets from various visual categories.

Chapter 4

Applications of Poselets

4.1 Introduction

The image of a person is a complex function of the camera view, the pose, the lighting conditions, the clothes of the person and other variables. Changing just one of these parameters in isolation can result in a very different image pattern. Poselets are effective in handling this complexity because they decompose the camera view and pose from the appearance: each poselet responds to a given part of the pose under a given viewpoint regardless of the appearance parameters. This decomposition of pose from appearance is essential for high-level computer vision tasks because it allows us to treat pose and appearance separately. This section gives an overview of how poselets help in several computer vision problems. The next section describes in more detail the use of poselets in attribute classification. The input for all of these problems is a cluster of poselet activations in the test image, as described in Section 3.7.

4.2 Object Recognition with Poselets

The goal of object recognition is to detect instances of the object regardless of viewpoint, pose and appearance. Poselets can be used almost "out of the box" here: each cluster of activations corresponds to a hypothesis. All we need to do is train a regression for estimating the bounding box, and a classifier for scoring each cluster. We use the following steps:

1. **Initial bounds prediction**.

   - **For categories other than person:** Each poselet in the cluster has an estimate for the location of each side of the visible bounding box $(x_{min}, y_{min}, x_{max}, y_{max})$. We take the weighted average of the predictions, weighting each activation by its probability. We use the probability without context (i.e. q_i) here. We include all poselets within the cluster, as well as any poselet consistent with them. We then

perform non-max suppression: we use agglomerative clustering to merge clusters whose intersection-over-union of the initial bounds prediction is greater than 0.6. We interpolate the bounds of the merged clusters.

- **For person:** Instead of predicting the person bounds we predict the location of the torso, as the torso is a more stable region to predict. The torso is defined with a center, size and angle. We predict the locations of the hips and shoulders as the average prediction of each poselet activation, weighted by the score of the activation. The center is the midpoint of the hips and shoulders and the size is the length between the midpoint of the shoulders and the midpoint of the hips. We use a fixed aspect ratio of 1.5. We then perform non-max suppression on the torso bounds the same way as above, using agglomerative clustering to merge torso predictions whose intersection over union is greater than 0.6. This results in a refined set of clusters. We then predict the person bounds from the poselet activations in a given cluster as above, i.e. we predict separately the value of each of $x_{min}, y_{min}, x_{max}, and y_{max}$ as a weighted average of the predictions of each poselet activation, weighting each by its probability q_i.

2. **Improving the predicted bounds**. The above generative bounding box prediction is not very accurate and we enhance it using a linear regression similar to [16]. Specifically we transform $[x_{min}\ y_{min}\ x_{max}\ y_{max}]^T$ with a 4x4 regression matrix T. To train T, we perform step 1 on the training set, we match the bounds predictions to the ground truths using intersection over union overlap of 0.45 and collect the true positives. We then fit T using the predicted bounds and the associated ground truth bounds via linear regression.

3. **Computing the score of a poselet cluster.** We follow [5] to predict the score of the poselet cluster, i.e., we train a linear discriminative classifier with positivity constraints on its weights to predict the scores based on the context-adjusted scores (Q_i) of the activations within the cluster. For our positive examples we use detections on the training set whose bounds intersection over union overlap is over 0.5. For negative examples we use detections that do not intersect the truth or whose overlap is less than 0.1. Our feature vector has dimensionality equal to the number of poselet types. The feature value for each poselet type is the maximum of all activations of that poselet type within the cluster.

We participated in the PASCAL person recognition competitions [13] for 2009 and 2010 and we also report results on the 2007 and 2008 datasets. In all datasets we are currently the leading method among all methods competing in Competition 3 and Competition 4. (Table 4.1)[1]. On Table 4.2 we show the effect of using context and the effect of varying the

[1]We did not participate in the 2007 and 2008 competitions. We computed the numbers for 2007 since the test set is available. The numbers for 2008 came as a result of our submission to 2010.

Dataset	Poselets	Second-best
VOC 2010	48.5%	47.5%
VOC 2009	48.6%	47.9%
VOC 2008	54.1%	43.1%
VOC 2007	46.9%	43.2%

Table 4.1: Average precision on the PASCAL competitions (best of comp 3 and comp 4) on the person category. The second-best results were obtained with various releases of the method by Felzenszwalb *et al.* [16].

Num. poselets	no context (q)	context (Q)
10	36.9%	37.8%
40	43.7%	44.3%
100	45.3%	45.6%
200	45.7%	46.9%

Table 4.2: AP on PASCAL VOC 2007 test set for the person category for various number of poselets without and with context.

number of poselets. Table 4.3 shows our performance for other visual categories. Figure 4.1 shows some examples of detecting people.

4.3 Attribute Classification with Poselets

We have used poselets for the task of inferring attributes of people from a static image, such as the gender, the hair style, the presence of a hat or glasses and the style of clothes. In this problem the pose and camera viewpoint are latent parameters: for example, we would like to detect the presence of glasses regardless of whether the person is in a frontal or a profile view. However, the huge variability in appearance introduced by pose variation prevents us from training a universal detector for glasses. Poselets are a natural way to address this problem: we train attribute classifiers for each poselet type and combine them into a single robust classifier. We are the first method to infer attributes for people under arbitrary pose and viewpoint variations. We also report state-of-the-art results for our gender classifier. This work is discussed in detail in Chapter 5.

4.4 Semantic Segmentation with Poselets

Semantic segmentation is the task of inferring which pixels in the image come from the object of interest and which ones come from the background. Poselets are useful for this task because if we know the pose and viewpoint at a given part of the object we can predict

Category	Detection AP	Segmentation AP
background	N/A	82.2%
aeroplane	33.2%	43.8%
bicycle	51.9%	23.7%
bird	8.5%	30.4%
boat	8.2%	22.2%
bottle	34.8%	45.7%
bus	39.0%	56.0%
car	48.8%	51.9%
cat	22.2%	30.4%
chair		**9.2%**
cow	20.6%	27.7%
dining table		6.9%
dog	18.5%	29.6%
horse	48.2%	**42.8%**
motorbike	44.1%	37.0%
person	**48.5%**	**47.1%**
potted plant	9.1%	15.1%
sheep	28.0%	35.1%
sofa	13.0%	**23.0%**
train	22.5%	37.7%
TV monitor	33.0%	36.5%

Table 4.3: Detection and segmentation results for poselets on PASCAL 2010, competitions 4 and 6. Our detection results were part of the competition. Our segmentation results are obtained after the competition and are reported in [7]. The results shown in bold are the ones on which our method is currently the leading one for the category.

the foreground mask. We leverage the foreground probability masks associated with each poselet (Figure 3.12). We combine our top-down prediction based on poselets with the bottom-up segmentation engine of Maire *et al.* [28]. We use an optical flow method of Brox and Malik [6] to align the mask to the underlying segmentation. On Table 4.3 we show our segmentation results on the PASCAL VOC 2010 competition 6. We report the best performance for person, chair, horse and sofa. Examples of our method are shown on Figures 4.2, 4.3, and 4.4. Full details of our method are described in Brox, Bourdev, Maji and Malik [7].

4.5 Pose Estimation with Poselets

Pose estimation is the task of inferring the articulation parameters of an object. In the case of a person we would like to infer the joint locations, 3D skeleton, the torso or head angles. For this task the appearance parameters, such as the clothes and the hair style are noise and all the signal is in the pose parameters. Poselets are useful as they are trained to respond to pose configurations regardless of appearance.

As we describe in Chapter 3 each poselet is trained to predict a spatial probability distribution for the location of each keypoint. To test the ability of poselets to infer the locations of individual keypoints, we created a simple keypoint predictor by taking the mean of the (x,y) predictions of a given keypoint from all poselets within the same cluster. Each poselet prediction is weighted according to the poselet probability and the variance in its Gaussian prediction. We tested the predictions on the true positive poselet clusters of the H3D test set. The results are shown on Figure 4.5. As expected, our prediction is best for keypoints on the head where we have lots of poselets, deteriorates for predicting the shoulders and hips, and it would deteriorate further in prediction of the outermost keypoints, such as wrists and ankles, since their location variability is highest. Notice also that keypoints along the axis of symmetry (the nose and the neck) are better predicted compared to other keypoints (left/right shoulder, left/right eye). The reason is that due to axial symmetry sometimes back-facing poselets fire on front-facing patterns and vice versa. This results in swapping the predictions of the left and right keypoint, which has the effect of lowering the prediction accuracy. We report these results in [5].

In general, we can use the cluster of poselet activations to predict the angle of the head and the torso. Specifically, we constructed a feature vector consisting of the score of each poselet type (or 0 if missing). We split the full circle into eight 45-degree pies and we trained eight view-specific classifiers. We did the same for the head and for the torso. The results are shown on Figure 4.6. On average we correctly predict the yaw of the head 62.1% of the time and the yaw of the torso 61.7%. Our average error is 26.3° for the head and 23.4° for the torso. More detail can be found in the work of Maji, Bourdev and Malik [29].

Category	Poselet AP
Phoning	49.6%
Playing an instrument	43.2%
Reading	27.7%
Riding a bike	**83.7%**
Riding a horse	**89.4%**
Running	85.6%
Taking a photo	31.0%
Using a computer	59.1%
Walking	67.9%

Table 4.4: Average precision on the PASCAL VOC 2010 action classification task. The results are generated after the competition. Results in bold are the current best for the category.

4.6 Action Classification with Poselets

We consider the task of inferring the action of a person from a static image. In this problem there is signal both in the pose and, sometimes, in the appearance. Specifically, the pose is very discriminative for actions, such as taking a picture, using a computer, walking, etc. The appearance can be useful for actions which are typically performed in specific outfits, such as biking and horse riding. The ability of poselets to decompose the pose from the appearance plays a key role for action recognition.

We trained action classifiers for the 9 actions in the PASCAL 2010 action classification competition. For each action we trained action-specific poselets by restricting the positive examples to come only from images of the given action and by selecting a small subset of the potential poselets based on their action-discrimination capabilities. Figure 4.7 shows examples of action-specific poselets. Our full model takes into account the responses of action-specific poselets, the responses of object detectors for horse, bicycle and TV monitor in the vicinity, as well as context based on the predicted actions of other people in the image. Our performance on the PASCAL 2010 action recognition dataset is shown on Table 4.4. We report the best results on two categories: biking and riding a horse. More detail can be found in the work of Maji, Bourdev and Malik [29].

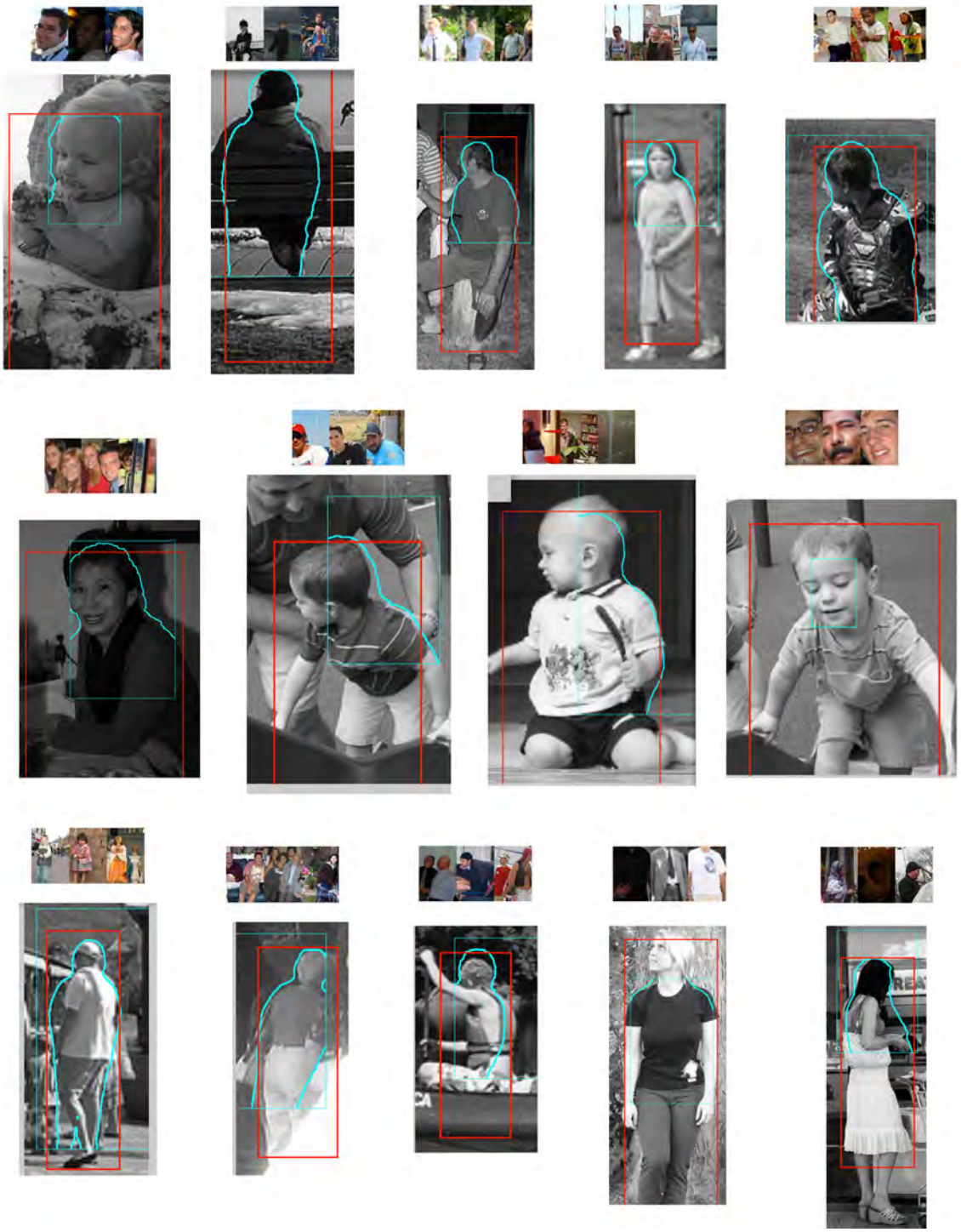

Figure 4.1: Detection examples. The proposed bounding box of each person is shown in red. The highest probability poselet activation is shown in a cyan bounding box and a figure-ground outline. Above each image we show three training examples from the poselet that was activated.

Figure 4.2: Examples of our person segmentation.

Figure 4.3: Examples of segmenting multiple categories.

Figure 4.4: Segmentation examples from multiple visual categories.

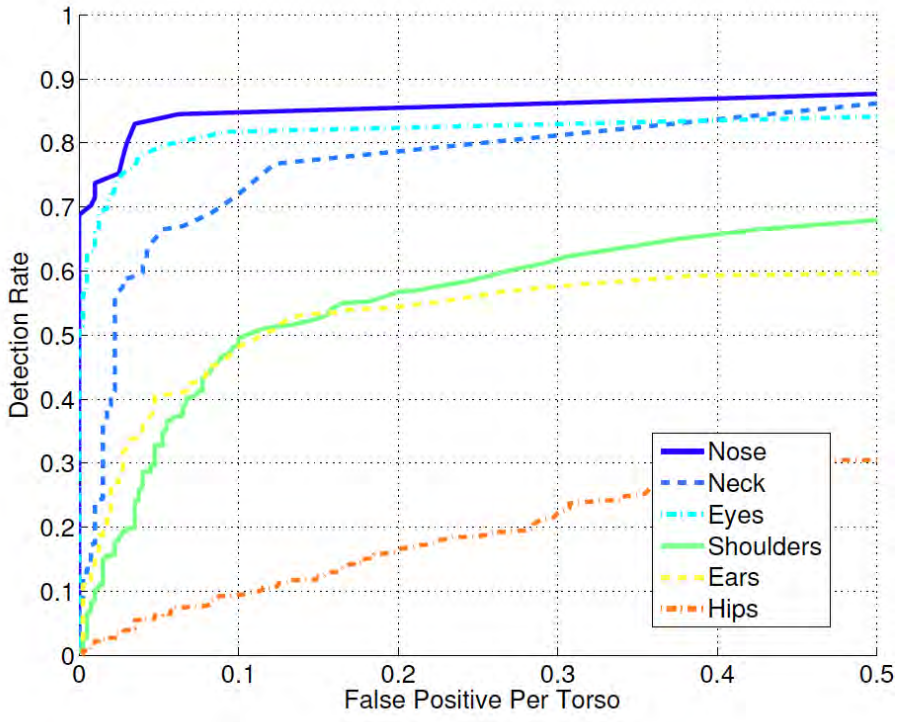

Figure 4.5: Detection rate of some keypoints conditioned on true positive torso detection on the H3D test set. We consider a detection as correct if it is within 0.2S of its annotated location, where S is the 3D distance between the two shoulders.

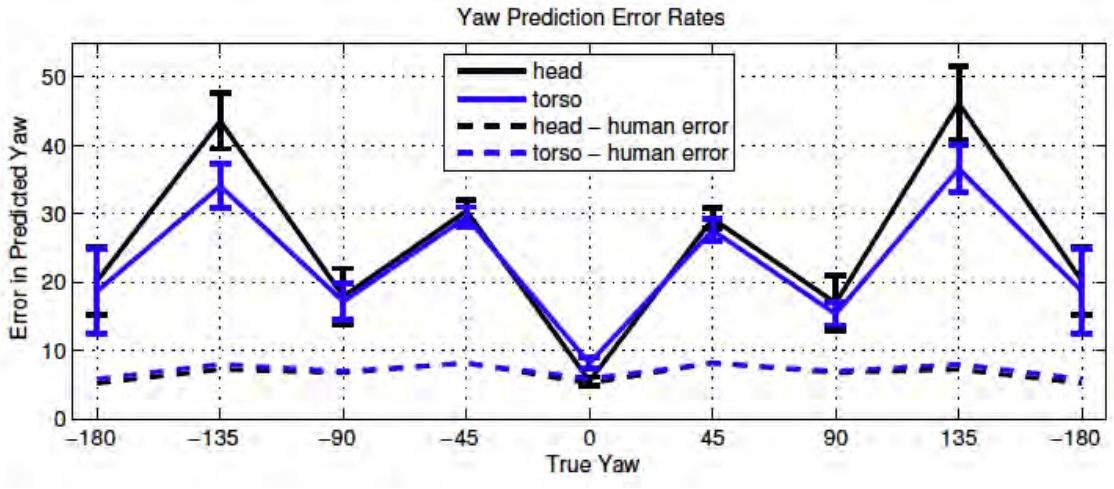

Figure 4.6: Error in predicting yaw across views.

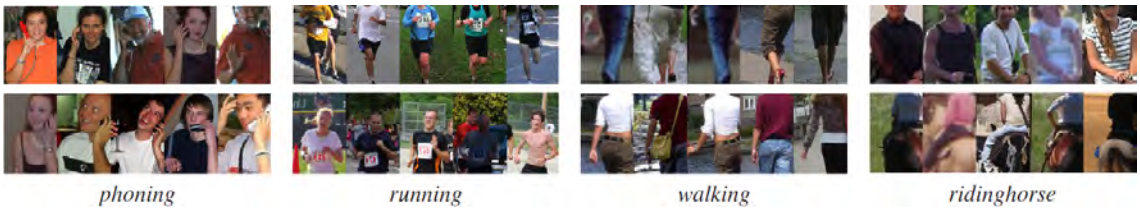

Figure 4.7: Examples of poselets trained to recognize specific actions.

Chapter 5

Attribute Classification with Poselets

In this chapter we explore the problem of attribute classification of people under arbitrary viewpoints and we develop a method based on poselets.

5.1 Introduction

We have an impressive ability to reliably recognize the gender of people under arbitrary viewpoint and articulation, even when presented with a cropped part of the image (Figure 5.1). Clearly we don't rely on the appearance of a single body part; gender can be inferred from the hair style, body proportions, types of clothes and accessories. We use different cues depending on the pose and viewpoint, and the same is true for other attributes, such as the hair style, presence of glasses and types of clothes.

Let us consider how we might build a system for classifying gender and other attributes. If we could somehow isolate image patches corresponding to the same body part from the same viewpoint then attribute classification becomes much easier. If we are not able to detect and align the parts well, however, the effect of nuisance variables, such as the pose, viewpoint and localization will affect the feature vector much more than the relevant signal (Figure 5.2). The visual cues associated with the attribute "has glasses", for example, are very subtle and different for a person facing the camera *vs.* a person looking sideways. As we show on Table 5.2, a generic classifier for has-glasses performs only slightly better than chance when trained on the entire person, but works much better when trained on aligned frontal faces.

Localizing body parts, however, is in itself a very hard problem, e.g. [18]. The PASCAL 2010 person layout challenge had only two contestants and the best AP for detecting hand is just 10.4% and foot just 1.2%! Frontal face is an exception, which is why virtually all state-of-the-art gender recognition approaches rely on carefully aligned frontal faces.

We develop an approach to solve this problem for gender as well as for other attributes, such the hair style, presence of glasses or hat, and the style of clothes. Specifically, we decom-

Figure 5.1: People can easily infer the gender based on the face, the hair style, the body proportions and the types of clothes. A robust gender classifier should take into account all such available cues.

pose the image into a set of parts, *poselets* [5], each capturing a salient pattern corresponding to a given viewpoint and local pose, such as the one shown in Figure 5.2 (right). This decomposition allows us to combine evidence from different parts of the body at different scales. **The activations of different poselets give us a robust distributed representation of a person from which attributes can be inferred without explicitly localizing different body parts.**

Prior work on gender recognition has focused on high resolution frontal faces or pedestrians and requires a face detector and alignment modules. Not only do we not need such modules, our method gracefully deals with profiles, back-facing people or even when the face is occluded or at too low a resolution, because we leverage information at multiple scales and aspects. Even though we use standard HOG and color features we outperform a leading commercial gender identification system that relies on proprietary biometric analysis. Furthermore, the same mechanism allows us to handle not just gender but any other attribute.

We illustrate our approach on the task of determining nine attributes of people – is-male, has-hat, has-t-shirt, has-shorts, has-jeans, has-long-hair, has-glasses, has-long-sleeves, has-long-pants. The training input is a set of images in which the people of interest are specified via their visible bounds and the values of their attributes. We use a three layer feed-forward network (Figure 5.4). In the first layer we predict the attribute value conditioned on each poselet type, such as the gender given a frontal face. In the second layer we combine the

Figure 5.2: The problem of determining the people wearing hats (top) *vs.* no hats (bottom) is difficult in unconstrained setup (left). If we can detect and align parts from the same view (right) the problem becomes much easier.

information from all such predictions (such as the gender given the face, the legs and the full body) into a single attribute classification. In the third layer we leverage dependencies between different attributes, such as the fact that gender is correlated with the presence of long hair.

We also collected a new dataset for attribute classification of people in unconstrained settings consisting of 8035 examples labelled with the nine attributes (Section 5.3). Although attribute recognition of people has been studied for frontal faces [24] and pedestrians [8], our dataset is significantly harder; it exhibits a large variation in viewpoint, pose, occlusion and self-occlusion, close proximity to other people, variable resolution, etc. (Figure 5.3).

5.2 Related work

Prior research on attributes has generally followed two directions. One line of work has used attributes as an intermediate representation layer with the goal of transfer learning as well as describing properties of objects [25, 14]. Farhadi *et al.* propose a method for localizing part-based attributes, such as a head or a wheel [15]. Recognition and localization of low-level attributes in a generative framework has also been proposed by Ferrari and Zisserman [19]. Joint learning of classes and attributes has been explored using Multiple Instance Learning [46] and latent SVMs [48]. Automated discovery of attributes from text and associated images has also been explored [19, 2, 47]. The key advantage of our method is that our parts implicitly model the pose and camera view, which we believe results in more powerful discrimination capabilities.

A second line of work has focused on attributes of people. Gender recognition methods using neural networks date back to the early 1990s [10, 21]. Support vector machines [32] and AdaBoost classifiers on Haar features [39] have been proposed for gender and race recognition. Kumar *et al.* propose using face attributes for the purpose of face recognition [24] as well as visual search [23]. Gallagher and Chen have explored inferring gender and age from visual features combined with names [20]. Gender, age and weight attributes have also been successfully extracted from 3D motion capture data [41]. These approaches generally require careful alignment of the data, and most of them apply to frontal faces only. We leverage the full body under any articulation without the need for alignment.

Our solution is based on poselets, which have been used effectively for recognition and segmentation of people [4], [5]. These problems are similar to ours, because the articulation and camera views are also latent parameters when recognizing and segmenting people. Thus, as we stated in Chapter 4, we can think of poselets as a general purpose engine for decomposing the viewpoint and pose from the appearance.

5.3 The Attributes of People dataset

There are several existing datasets of attributes of people but we did not find any suitable for the context in which our method is used. FaceTracer [23] uses 15000 faces and full body, but provides only URLs to images and many of the images are no longer available. Other datasets, such as PubFig [24] and the Labeled Faces in the Wild [22] include only frontal faces.

We propose a new dataset of 8035 images, each centered at a full body of a person. The images are collected from the H3D [5] dataset and the PASCAL VOC 2010 [12] training and validation datasets for the person category, but instead of the low-resolution versions used in PASCAL, we collected the full resolution equivalents on Flickr. For each person we cropped the high resolution image around that person, leaving sufficient background around the visible bounds and scaled it so the distance between hips and shoulders is 200 pixels. For each such image we provide the visible bounds of the person in the center and a list of bounds of all other people in the background.

We used Amazon Mechanical Turk[1] to provide labels for all attributes on all annotations by five independent annotators. A label was considered as ground truth if at least 4 of the 5 annotators agreed on the value of the label. We discarded 501 annotations in which less than two attributes were specified as ground truths which left us with 8035 images. Table 5.1 shows the distribution of labels. We split the images into 2003 training, 2010 validation and 4022 test images by ensuring that no cropped images of different set come from the same source image and by maintaining a balanced distribution of the H3D and PASCAL images in each set. Figure 5.3 shows 50 examples drawn at random from our test set.

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

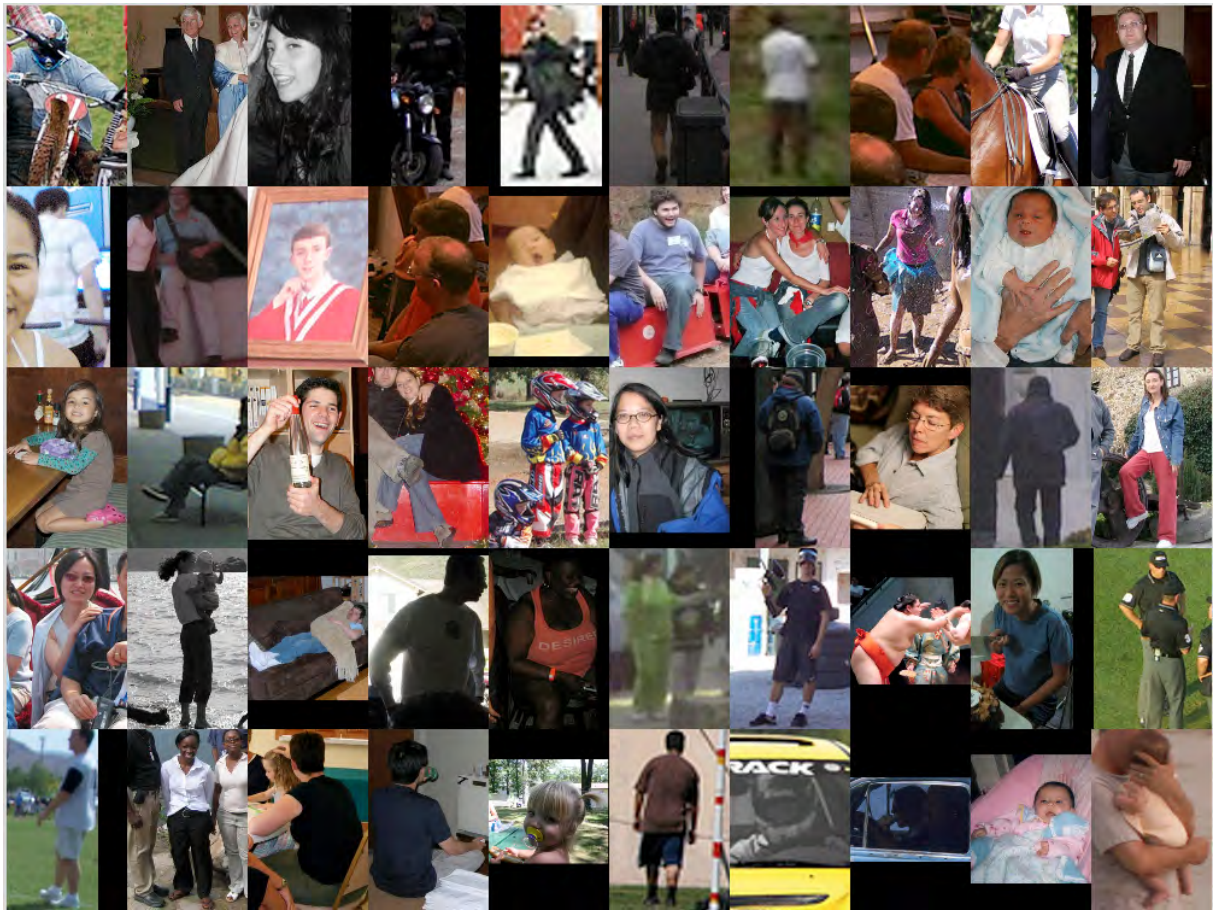

Figure 5.3: Fifty images drawn at random from our test set and slightly cropped to the same aspect ratio. Each image is centered at a target person. Our dataset is challenging as it has a large variability of viewpoints, poses, and occlusions. In some cases people are close to each other which makes identifying the correct person challenging as well. To aid identification we provide the visible bounds of the target person, as well as the bounds of all other people in the image.

Attribute	True	False	Attribute	True	False
is male	3395	2365	long hair	1456	3361
has hat	1096	5532	glasses	1238	4083
has t-shirt	1019	3350	long sleeves	3045	3099
has shorts	477	2020	long pants	2020	760
has jeans	771	1612			

Table 5.1: Number of positive and negative labels for our attributes.

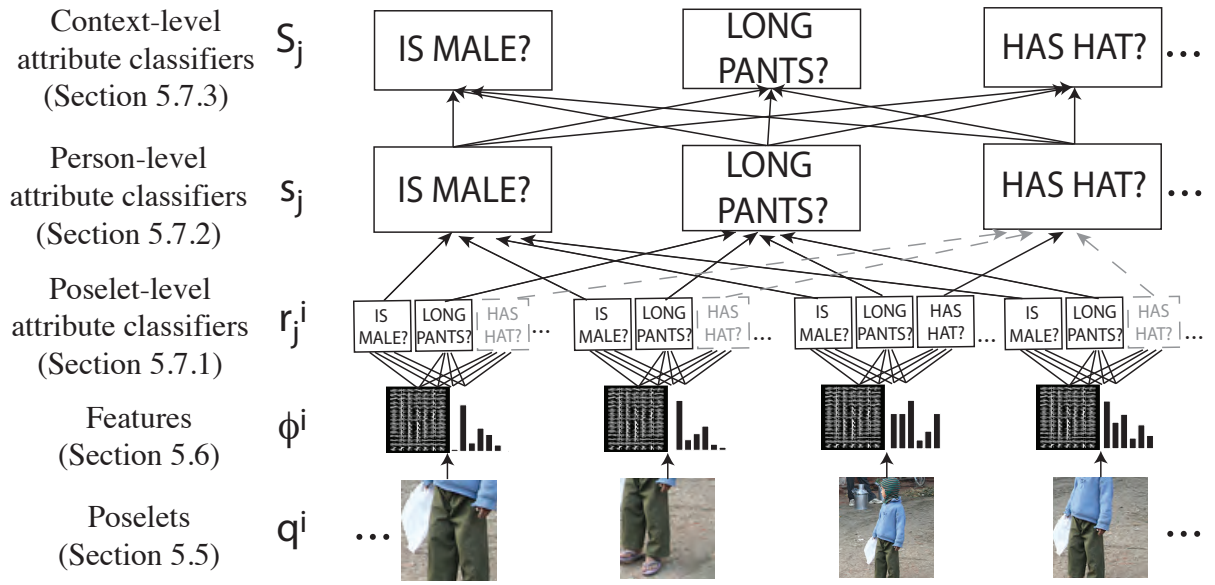

Figure 5.4: Overview of our algorithm at test time. Poselets are detected on the test image; detection scores q^i are computed and features ϕ^i are extracted. Poselet-level attribute classifiers r_j^i are evaluated for every poselet activation i and attribute j (unless the attribute is part-specific and the poselet does not cover the part, such as the has-hat for three of the four shown poselets). A person-level attribute classifier s_j for every attribute combines the feedback of all poselet-level classifiers. A context-level classifier S_j for the attribute takes into account predictions of the other attributes. This picture uses 4 poselets and 3 attributes, but our system uses 1200 poselets and 9 attributes.

5.4 Algorithm Overview

Our algorithm at test time is shown on Figure 5.4 and can be summarized as follows:

Step 1 We detect the poselets on the test image and determine which ones are true positives referring to the target person (Section 5.5). Let q^i denote the probability of poselet type i. q^i is the score of the poselet classifier transformed by a logistic, with zero mean, or 0 if the poselet was not detected.

Step 2 For each poselet type i we extract a feature vector ϕ^i from the image patch of the activation, as described in Section 5.6. The feature vector consists of HOG cells at three scales, a color histogram and skin-mask features.

Step 3 For each poselet type i and each attribute j we evaluate a classifier r_j^i for attribute j conditioned on the poselet i. We call these the *poselet-level attribute classifiers*. We use a

linear SVM followed by a logistic g:

$$r_j^i = g(w_j^{iT}\phi^i + b_j^i) \tag{5.1}$$

where w_j^i and b_j^i are the weight vector and the bias term of the SVM. These classifiers attempt to determine the presence of an attribute from a given part of the person under a given viewpoint, such as the has-hat classifier for a frontal face shown on Figure 5.2.

Step 4 We zero-center the outputs of the poselet-level attribute classifiers, modulate them by the poselet detection probabilities and we use them as an input to a second-level classifier for each attribute j, called a *person-level attribute classifier*, whose goal is to combine the evidence from all body parts. It emphasizes poselets from viewpoints that are more frequent and more discriminative. It is also a linear classifier with a logistic g:

$$\psi_j^i = q^i(r_j^i - 0.5) \tag{5.2}$$

$$s_j = g(w_j'^T\psi_j + b_j') \tag{5.3}$$

Step 5 Finally, for each attribute j, we evaluate a third-level classifier which we call the *context-level attribute classifier*. Its feature vector is the scores of all person-level classifiers for all attributes, s_j. This classifier exploits the correlations between the attributes, such as gender *vs.* the presence of a skirt, or short-sleeves *vs.* short-pants. We use an SVM with quadratic kernel which we found empirically to work best. We denote the score of this classifier with S_j, which is the output of our algorithm.

5.5 Training and using poselets

We trained poselets as described in Chapter 3. For each poselet, during training, we build a soft mask for the probability of each body component (such as hair, face, upper clothes, lower clothes, etc) at each location within the normalized poselet patch (Figure 5.5) using body component annotations on the H3D dataset (Chapter 2). We did not use poselet selection to reduce the number of poselets; we used all 1200 poselets. In addition we did not use context (Section 3.6) for this problem. After clustering the activations (Section 3.7) we predicted the hypothesis bounding box as in the detection task (Section 4.2).

We now need to decide which cluster of poselets refers to the person in the center of the image. This is not a trivial problem and simply picking the bounding box closest to the center of the image is not always correct because sometimes people are close to each other and their visible bounds largely overlap. In addition some clusters may refer to false positive detections. We found that it is better to find the optimally global assignment of all

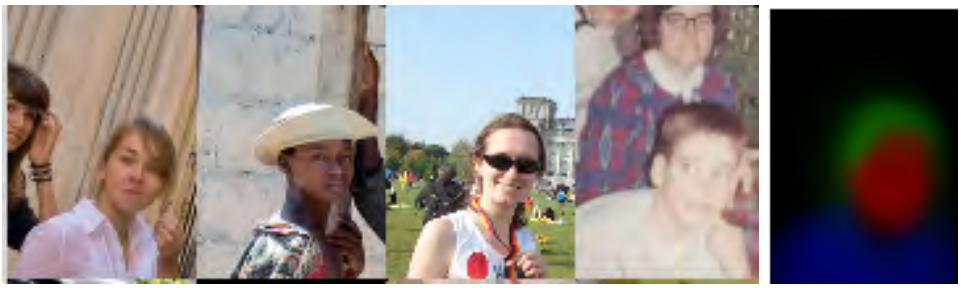

Figure 5.5: **Left:**Examples of a poselet. **Right:** The poselet soft mask for the hair, face and upper clothes.

Figure 5.6: Example of matching detected bounds to truth bounds. This image is centered at the woman and the man behind her is part of the background. The thick rectangles are the truth bounds and the thin ones are the detected bounds. The colors indicate the assignment. The goal is to find which detected bounds corresponds to the woman. A simple rule, such as "the most central one" or "the one with the highest score" does not aways work well. We use the Hungarian algorithm to provide optimal assignment.

hypotheses to all truth bounding boxes by preferring to assign a bounding box to a given truth if its intersection over union is high, and by giving preference to hypotheses with higher scores, which are less likely to be false positives. Figure 5.6 shows an example where visible bounds overlap significantly and using the strongest or the one nearest the center results in assigning to the wrong person. We formulate this problem as finding the maximum flow in a bipartite graph and we used the Hungarian algorithm to find a solution. Once we do the matching, we associate the poselets with the cluster matched to the ground truth bounds and discard all other poselet activations as false positives or belonging to other people. The result is a set of poselet activations q^i that refer to the foreground person.

5.6 Poselet-level features ϕ^i

In this section we describe our poselet-level features ϕ^i, which consist of HOG features, color histogram and skin-specific features.

For the HOG features we use the same parameters as described in [11]. In addition to the 8x8 cells we extract HOG at two coarser levels - 16x16 and 32x32. Depending on the patch dimensions this feature is of size between 2124 and 4644. The color histogram is constructed with 10 bins in each of the H, S and B dimensions.

For the skin-specific features we trained a skin classifier, which is a GMM with 5 components fit from the LAB-transformed patches of skin collected from various skin tones and illuminations. We use three skin features: hands-skin, legs-skin and neck-skin. Each feature is the fraction of skin pixels in the corresponding part. Figure 5.7 describes how the feature is computed using the hand-skin feature of an upper-body-torso poselet as an example.

5.7 Classifiers

5.7.1 Poselet-level attribute classifier r_j^i

We train a separate classifier for each of the 1200 poselet types i and for each attribute j. We used the 2003 training images for training these classifiers.

We construct a feature vector from all activations of poselet i on the training set. The label of a given activation is the label associated with the ground truth to which the poselet activation is assigned. We discard any activations on people that don't have a label for the given attribute. Figure 5.2(right) shows instances of positive (top row) and negative (bottom row) examples for the frontal face poselet and the "has-hat" attribute.

Some attributes have associated parts and poselets in which these parts don't appear are excluded from training of the attribute. For example, as shown on Figure 5.4 it doesn't make sense to use a legs poselet to train the "has-hat" attribute.[2] To determine if a poselet covers

[2]Poselets away from the useful part can sometimes be effective in training the part attribute. For example,

Figure 5.7: Computing skin-specific features. The skintone classifier is applied to the poselet activation patch (A) to obtain the skintone probability mask (B). The poselet part soft mask (C), in this case, a mask for the hands, is used to modulate the skintone mask and the result is shown in (D). While for this poselet the positions of the hands vary, as evidenced by the widespread hands mask, we are still able to exclude most non-hand skin areas. The hand-skin feature is the fraction of skin pixels in the modulated mask (D). This feature is especially useful for determining if a person wears short or long sleeves.

a given part, we check to see if its mask (Figure 5.5) has presence of that part. This spatial selection reduces the dimensionality of our classifiers and the opportunity for overfitting and we found that it improves performance.

Our classifiers are linear SVMs trained with weighted examples. The weight of each training example is the probability of the corresponding poselet activation q^i.

5.7.2 Person-level attribute classifier s_j

The person-level attribute classifier for attribute j combines all poselet-level classifiers for the given attribute. The feature vector has one dimension for each poselet type. Our features are zero-centered responses of the poselet-level attribute classifiers, see Equation 5.2. Our classifier is similar to a linear SVM, except we impose positivity constraints on the weights. The positivity constraints restrict the solution space to the correct half-space. A negative weight of a classifier would mean that the SVM takes the opposite of the classifier's advice. This could only happen due to overfitting so we prevent it explicitly. Since the input of the classifier is trained on the training set, we use the 2010 validation images to train the person-level attribute classifier.

a leg poselet can be trained for the "has-long-sleeves" attribute, because short pants are correlated with short sleeves. However we leverage the correlations between attributes in a later level of our hierarchy.

5.7.3 Context-level attribute classifier S_j

There are strong correlations among various attributes. Long hair is much more common in women than in men; when people wear short pants they also tend to wear short sleeves, etc. We would like to adjust each attribute based on the values of the others. This is especially helpful when the direct evidence for the attribute is non-salient.

To take advantage of the correlations among attributes we train an SVM with a quadratic kernel for each attribute. The features are the scores of all person-level attribute classifiers for a given person. We trained the context-level classifier on the training + validation sets.

5.8 Experimental results

The highest/lowest scoring examples for each attribute on the test set are shown on Figure 5.8. Our system performs very well, correctly classifying 103 of the top 108 examples across attributes. The most confused examples are on Figure 5.9. The confusions are often due to unusual examples, such as men with long hair, errors in the ground truths, severe occlusion and assignment to the wrong person.

5.8.1 Performance *vs.* baselines

To validate the design choices of our approach we tested the effect of disabling portions of our model. Specifically, we measured the effect of disabling the skin features and the context classifier. The results are shown on Table 5.2, columns 7-9. As expected, skin features are essential for clothes-style attributes (the bottom five on Table 5.2) and without skin their mean AP drops from 63.18 to 55.10. The other attributes, such as gender and hairstyle are largely unaffected by skin. The context classifiers help on seven of the attributes and decrease performance on two, boosting the overall mean AP from 61.5 to 65.2.

Our baseline method uses Canny-modulated Histogram of Oriented Gradients [3] with Spatial Pyramid Matching kernel [26] which is effective for image classification in Caltech-101 as well as gender classification on MIT pedestrians [9]. The results of training it on the full bounds of the person are in column 6 of Table 5.2. We handily outperform SPM across all attributes with a mean AP of 65.18 *vs.* 45.91 for the SPM. We believe this is partly due to the fact that the generic spatial model used in the SPM is insufficient and the implicit pose-specific alignment provided by the poselets is necessary. Our examples have large degree of articulation and a generic classifier would suffer from localization errors, especially for location-sensitive attributes such as has-glasses. To help SPM with localization we extracted higher resolution views of the people, zoomed on the head, upper body and lower body (Figure 5.10). Columns 3-5 on Table 5.2 show the results of using an SPM trained on each of the zoomed views. As expected, the head zoom improves detection of gender, hairstyle, presence of glasses and a hat. However, even if we used the best view for each

Figure 5.8: The six highest and lowest scoring examples of each attribute on our test set. Of the 108 examples, five are classified incorrectly and marked with an X in the upper right corner. Three of them are women wearing hats misclassified as men. The gender attribute is the only one negatively affected by the context classifier and the effect applies only for the lowest recall mode, shown here.

Figure 5.9: Examples of most confused attributes. Many of the most confused males have long hair and the most confused females hide their hair under a hat. Results are affected by incorrect ground truth labels (has t-shirt, has-shorts), occlusion (has-jeans), and confusion with another person (has-shorts, not has-long-pants).

Figure 5.10: To help with localization, we provide our baselines the full bounds (left), as well as zoomed and aligned views of the head, upper body and lower body.

attribute, we would get a mean AP of 51.87, which, despite the extra supervision, remains substantially lower than our AP of 65.18.

5.8.2 Performance from different viewpoints

As the examples on Figure 5.8 show, the classifiers are most confident for people facing the camera. To test the robustness of our method to different viewpoints we partitioned the test set into three partitions – frontal, profile and back-facing people and we tested the performance for each view. To automatically partition the data we made use of the keypoint annotations that come with our images. Specifically, images for which both eyes are present are treated as frontal; if only one eye is present the image is treated as a profile, and if no eyes are present, and the left shoulder is to the left of the right shoulder we assign the image to the back-facing category. Approximately 61% of our test data consists of frontal images, 18% is profile images and 11% is back-facing people. Around 9% of the data did not fall into any of these categories. In some cases this is due to missing annotation data, and in other cases the head is not visible. Table 5.3 shows the average precision of the attributes on all the data and on each partition. As expected, performance is highest for frontal examples, followed by back-facing and then profile examples.

5.8.3 Optimal places to look for an attribute

It is not obvious exactly which part of the image is most discriminative for a given attribute. Consider the attribute has-long-hair. Clearly we should look at the face, but what is the optimal zoom level and pose? What if the person is in a profile or back-facing view? Our method automatically determines the optimal location, scale and viewpoint to look for evidence for a given attribute. This is a function of both the frequency of the given pose in the training set and the ease of discrimination given the pose. Specifically, the person-level classifier ranks each poselet type according to its predictive power. Figure 5.12 shows the

Attribute(1)	Freq(2)	Spatial Pyramid Match				Our Method		
		Head(3)	Lower(4)	Upper(5)	BBox(6)	No ctxt(7)	No skin(8)	Full(9)
is male	59.3	74.9	63.9	71.3	68.1	82.9	82.5	82.4
has long hair	30.0	60.1	34.0	45.2	40.0	70.0	73.2	72.5
has glasses	22.0	33.4	22.6	25.5	25.9	48.9	56.1	55.6
has hat	16.6	53.0	24.3	32.3	35.3	53.7	60.3	60.1
has t-shirt	23.5	32.2	25.4	30.0	30.6	43.0	48.4	51.2
has long sleeves	49.0	53.4	52.1	56.6	58.0	74.3	66.3	74.2
has shorts	17.9	22.9	24.8	22.9	31.4	39.2	33.0	45.5
has jeans	33.8	38.5	38.5	34.6	39.5	53.3	42.8	54.7
long pants	74.7	79.9	80.4	76.9	84.3	87.8	85.0	90.3
Mean AP	36.31	49.81	40.66	43.94	45.91	61.46	60.84	**65.18**

Table 5.2: Average precision of baselines relative to our model. **Freq** is the label frequency. We trained separate SPM models on the head (**Head**), lower body (**Lower**), upper body (**Upper**) and full bounding box (**BBox**) as shown on Figure 5.10. We tested our method by disabling the skin features (**No skin**), the context classifiers (**No ctxt**) and on the full system (**Full**).

Attribute	All	Frontal	Profile	Back
is male	82.4	82.9	82.9	83.2
has long hair	72.5	81.3	31.3	47.2
has glasses	55.6	59.8	33.9	18.8
has hat	60.1	66.4	54.8	41.9
has long sleeves	74.2	76.1	70.6	75.1
has t-shirt	51.2	55.7	43.3	46.7
has long pants	90.3	89.9	92.9	94.2
has jeans	54.7	53.0	46.9	70.0
has shorts	45.5	47.8	48.6	45.3
Mean AP	65.18	68.11	56.12	58.05
Num. examples	4022	2449	736	459

Table 5.3: Average precision for the attributes using all test annotations as well as using frontal-only, profile-only and back facing-only ones. The has-glasses attribute is most affected by the head orientation, and it drops to chance level for the back-facing case.

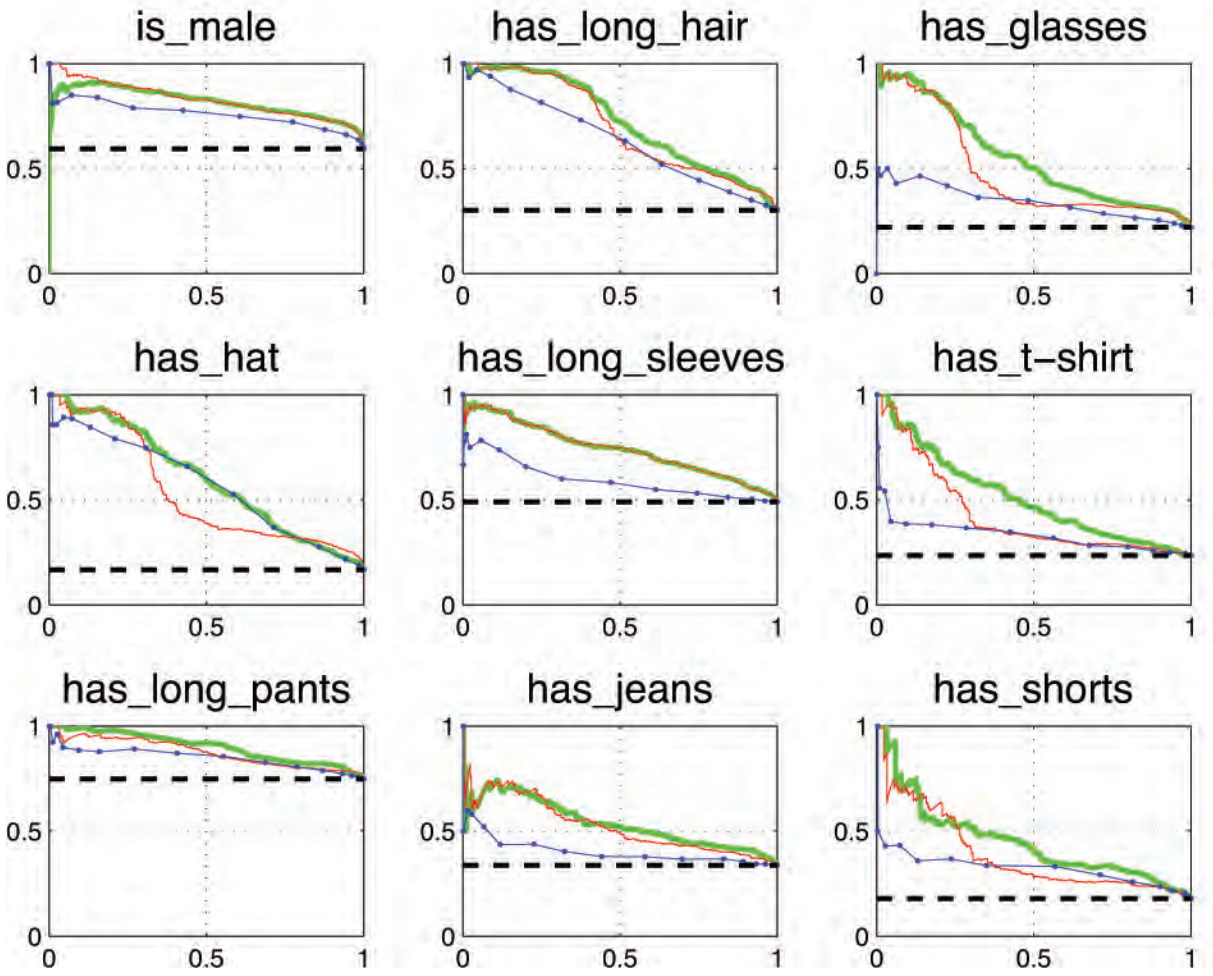

Figure 5.11: Precision-recall curves of the attribute classifiers on the test set. Our full result (column 9 in Table 5.2) is shown in thick green. Our performance without context classifiers (column 7) is shown in red; the SPM using the optimal view per attribute (max of columns 3-6) is shown in blue and the frequency of the label (column 2) is the dashed black horizontal line.

Figure 5.12: Our algorithm automatically determines the optimal poses and viewpoints to look for evidence of a given attribute. **First row:** The top poselets for is-male. **Second row:** The top poselets for has-long-hair. **Third row:** The top poselets for has-glasses. These three attributes require progressively higher zoom, which is reflected in the choice of poselets. The poselets are drawn by averaging their top ten training examples.

top five poselets used for determining the gender, hair length and presence of glasses. Since more than half of the people in our training set are facing the camera, and frontal view is usually more discriminative, the top poselets all come from frontal view.

5.8.4 Gender recognition performance

Comparison with other methods is challenging because the vast majority of person-specific attribute classification methods operate on frontal faces only [24, 32, 39]. If we applied our method on their datasets, our three-level hierarchy would reduce to a single frontal poselet and the comparison will reduce to the effectiveness of HOG features for gender classification,

a problem that is interesting but not directly relevant to our work[3]. In addition, other methods use different attributes, with the exception of gender.

Fortunately we have access to the Cognitec face recognizer, which is the winner of FRVT 2002 and one of the leading commercial face recognizers according to MBE 2010, the latest NIST test[4]. Cognitec can also report gender. As with other methods, it operates on frontal faces only. The Cognitec API does not allow for training of gender, so we could not train it on our training set. For optimal performance, we applied the engine on the zoomed head views (Figure 5.10b). Cognitec failed to find the face in 38.0% of the images (not all of them have frontal faces) and it failed to predict gender of another 20.0%. If we use mean score for the missing predictions we get AP of 75.0% for Cognitec *vs.* our AP of 82.4%. The precision-recall curve is shown on Figure 5.14. If we restrict the test to the faces for which Cognitec predicts gender, we get AP of 83.72% for Cognitec and 83.74% for our method, essentially equal, even though we aid Cognitec by providing a zoomed centered view of the head. Note that we use simple HOG features and linear SVMs and Cognitec uses careful alignment and advanced proprietary biometric analysis. We believe that our method benefits from the power of combining many view-dependent poselet classifiers.

We don't have access to other leading methods, such as [24], but we can give an upper bound to their performance since they all require frontal faces. In our dataset 60.9% of the faces are frontal. If other methods use perfect face detector, perfect alignment and perfect recognition for frontal faces and perform at chance level for other cases, their AP would be 60.9*1 + 39.1*0.5 = 80.5 *vs.* our AP of 82.4.

5.8.5 Comparisons to human visual system

Are the cues used by humans similar to the ones exploited by our system? To help answer this question we conducted an experiment using 10 representative poselets chosen to cover various parts of the body at various zoom levels. For each poselet we picked 100 examples, 50 male and 50 female. We flashed a random poselet example for an average of 200ms followed by a random image and asked each of the 8 subjects to immediately choose the gender of the example. We then sorted the 10 poselets using their mean AP averaged over all subjects, and we also sorted them according to their AP of discriminating gender in our system. The results are shown on Figure 5.13. The figure shows that there is a strong correlation between poselets preferred by humans and those preferred by our system.

5.9 Describing people with poselets

We can easily combine the attribute predictions that have high confidence values to generate complete descriptions of the person. Some of the better predictions are shown on Figure 5.15.

[3]Our skin features are only useful for attributes not visible from the frontal face

[4]http://www.cognitec-systems.de/FaceVACS-Performance.23.0.html

Figure 5.13: The poselets that performed best (left) to worst (right) for people (top row) and the computer algorithm (bottom).

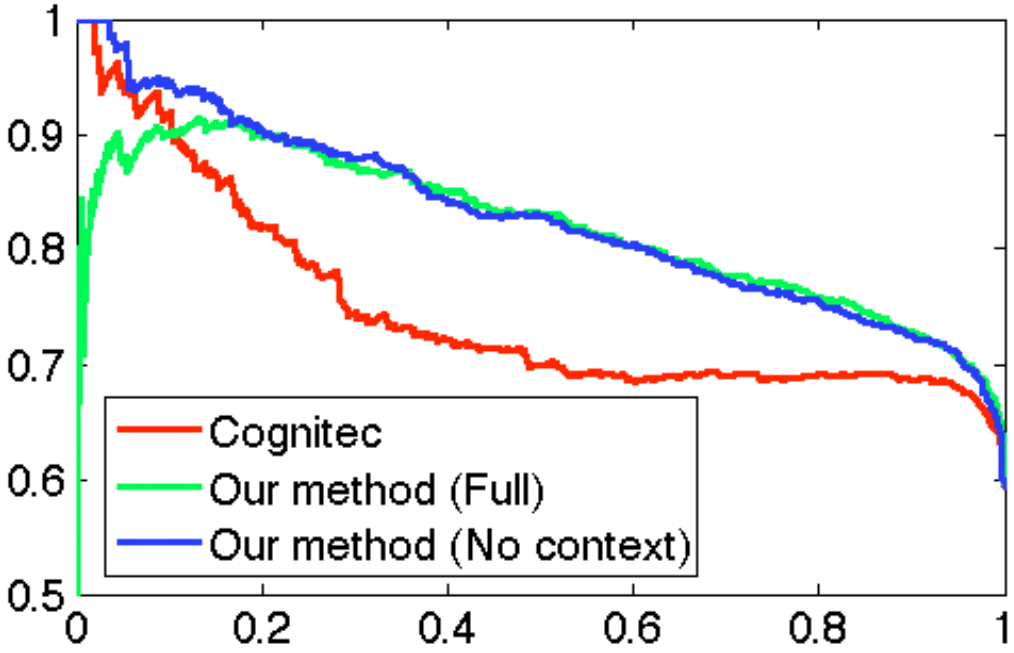

Figure 5.14: Precision-recall curves on gender recognition using our full method (AP=82.4), our method without context classifiers (AP=82.9) and Cognitec (AP=75.0).

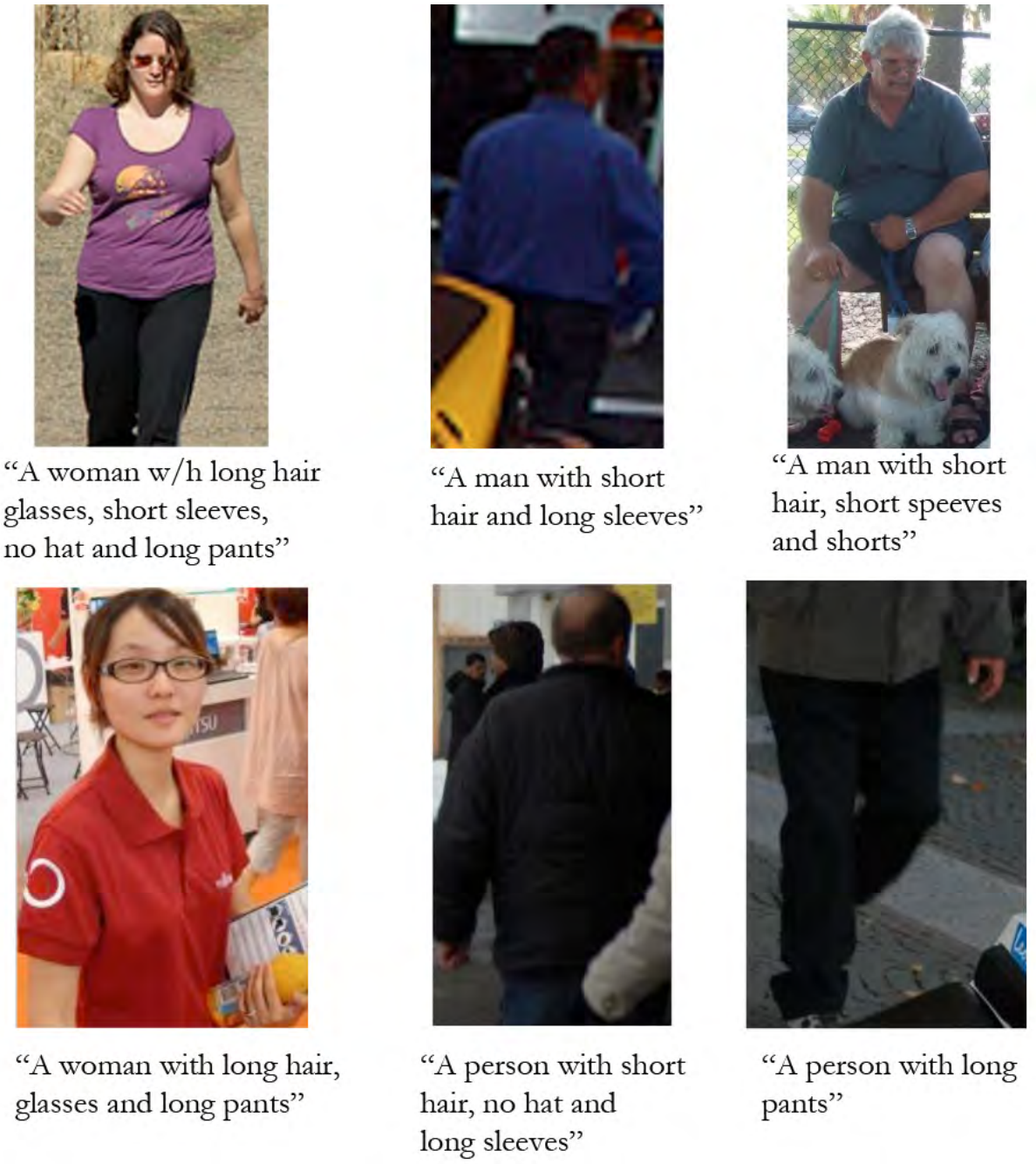

Figure 5.15: We combine the attributes with higher confidence obtained for a given person into complete descriptions.

5.10 Discussion

We are the first to address an important but challenging problem with many practical applications - attribute classification of people "in the wild". Our solution is simple and effective. It is robust to partial occlusion, articulation and camera view. It draws cues from any part of the body at any scale and it leverages the power of alignment without explicitly inferring the pose of the person. While we have demonstrated the technique using nine attributes of people, it trivially extends to other attributes and other visual categories. We provide a large dataset of 8035 people annotated with 9 attributes, which we hope will inspire others to follow with better methods.

Chapter 6

Conclusion

No single method currently dominates all high-level computer vision problems. Depending on the task, the dataset the currently leading methods could be based on a Latent SVM, Spatial Pyramid Matching, Multiple Kernel Learning, or others, and are sometimes custom-designed for the dataset. We propose a single approach that, almost out-of-the-box, applies to a variety of problems and often achieves state-of-the-art performance.

Our approach can be thought of as a feed-forward network. The neurons on the first layer are poselet classifiers operating on the HOG features of the image and the neurons on the second level are context-enhanced poselets. Our choice of linear SVMs followed by logistic classifiers makes our setup equivalent to a standard multi-layer neural network. Our pose and action classifiers essentially add a third layer to the network. Our attribute classifiers add a few layers on the network – one layer of aspect-specific attribute classifiers, followed by a layer of aspect-independent classifiers, followed by a layer of context-enhanced aspect-independent attribute classifiers.

Our system is a hierarchy in which every layer is more abstract and has more invariance than the previous one. It starts with simple gradient orientations (the HOG features), followed by local pattern recognizers that don't rely on context and it ends with nodes that detect the gender of a person or her current action independent of her pose and camera viewpoint.

There is one important difference from other neural networks: we don't rely on back-propagation for training. Instead we train each node in the hierarchy in a supervised manner, starting from the bottom layer. This supervision at every level means that training is simple, efficient and there is no danger of getting stuck in a local minima. Moreover, the system is not a black box: every node has associated semantics that can be described with words, such as "probability of back-view of a person's hand next to a hip" or "probability of long hair given a left-profile view of the head and shoulders". It is easy to measure performance, look at failure cases and visualize the output of every node. Semi-supervised and unsupervised methods have been the subject of much attention in computer vision. In this thesis we argue for the opposite: extra supervision is good. It makes the problem easier, allows us to train better classifiers and, with the rise of crowdsourcing, is scalable and inexpensive.

Bibliography

[1] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

[2] Tamara L. Berg, Alexander C. Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[3] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *ACM ICIVR*, 2007.

[4] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, sep 2010.

[5] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, sep 2009.

[6] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 2010.

[7] Thomas Brox, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011.

[8] L. Cao, M. Dikmen, Y. Fu, and T.S. Huang. Gender recognition from body. In *Proceeding of the 16th ACM international conference on Multimedia*. ACM, 2008.

[9] M. Collins, J. Zhang, P. Miller, and H. Wang. Full body image feature representations for gender profiling. In *ICCV Workshop*, 2010.

[10] Garrison W. Cottrell and Janet Metcalfe. Empath: face, emotion, and gender recognition using holons. In *NIPS*, NIPS, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 2, June 2005.

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge (2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[13] 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*, 88(2):303–338, June 2010.

[14] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, 0, 2009.

[15] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010.

[16] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *PAMI, to appear*, June 2008.

[17] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, V61(1), January 2005.

[18] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, June 2008.

[19] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, December 2007.

[20] A. Gallagher and T. Chen. Estimating age, gender and identity using first name priors. In *CVPR*, 2008.

[21] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *NIPS*, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[22] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[23] N. Kumar, P. N. Belhumeur, and S. K. Nayar. FaceTracer: A Search Engine for Large Collections of Images with Faces. In *ECCV*, Oct 2008.

[24] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, Oct 2009.

[25] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[26] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*. IEEE, June 2006.

[27] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Combined object categorization and segmentation with an implicit shape model. In *In ECCV workshop on statistical learning in computer vision*, 2004.

[28] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[29] S. Maji, L. Bourdev, and J. Malik. Action recognition using a distributed representation of pose and appearance. In *CVPR*, 2011.

[30] Subhransu Maji and Jitendra Malik. Object detection using a max-margin hough tranform. In *CVPR*, 2009.

[31] 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 *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[32] Baback Moghaddam and Ming hsuan Yang. Learning gender with support faces. *IEEE TPAMI*, 24, 2002.

[33] Greg Mori and Jitendra Malik. Estimating human body configurations using shape context matching. *ECCV*, 2002.

[34] R. Nevatia and T. O. Binford. Description and recognition of curved objects. In *Artificial Intelligence*, volume 8, pages 77–98, 1977.

[35] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio. Pedestrian detection using wavelet templates. In *CVPR*, 1997.

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

[37] Xiaofeng Ren, Alexander C. Berg, and Jitendra Malik. Recovering human body configurations using pairwise constraints between parts. In *ICCV*, 2005.

[38] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: A database and web-based tool for image annotation. *IJCV*, 77(1-3), 2008.

[39] Gregory Shakhnarovich, Paul A. Viola, and Baback Moghaddam. A unified learning framework for real time face detection and classification. In *In Proceedings of International Conference on Automatic Face and Gesture Recognition*, 2002.

[40] Hedvig Sidenbladh and Michael J. Black. Learning the statistics of people in images and video. In *IJCV*, 2003.

[41] Leonid Sigal, David J. Fleet, Nikolaus F. Troje, and Micha Livne. Human attributes from 3d pose tracking. In *ECCV*, 2010.

[42] A. Sorokin and D.A. Forsyth. Utility data annotation with amazon mechanical turk. In *First IEEE Workshop on Internet Vision, CVPR*, 2008.

[43] Camillo Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *Comp. Vision & Image Understanding*, 80(10), Oct 2000.

[44] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, 30(11), 2008.

[45] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[46] Gang Wang and David Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[47] Josiah Wang, Katja Markert, and Mark Everingham. Learning models for object recognition from natural language descriptions. In *BMVC*, September 2009.

[48] Yang Wang and Greg Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.