UC Irvine
UC Irvine Electronic Theses and Dissertations

Title
Recognizing and Segmenting Objects in the Presence of Occlusion and Clutter

Permalink
https://escholarship.org/uc/item/1139s1ns

Author
Ghiasi, Golnaz

Publication Date
2016

Copyright Information
This work is made available under the terms of a Creative Commons Attribution License, availalbe at https://creativecommons.org/licenses/by/4.0/

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

Recognizing and Segmenting Objects in the Presence of Occlusion and Clutter

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Golnaz Ghiasi

Dissertation Committee:
Professor Charless Fowlkes, Chair
Professor Deva Ramanan
Professor Alexander Ihler

2016

© 2016 Golnaz Ghiasi

DEDICATION

To my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

It has been my great pleasure and honor to work with my advisor, Charless Fowlkes. I always admire his wonderful attitude, his enthusiasm toward research and his vast knowledge and wisdom. I owe him a big thank for his patient guidance and generous help over the past few years.

I would also like to thank Deva Ramanan, his excitement and enthusiasm towards all papers and discussions in the group meetings were always inspiring for me. Also thanks to my other thesis committee, Alex Ihler for his time and help with this thesis.

Many thanks to all my great friends at vision lab: Hamed, Chaitanya, Dennis, Xiangxin, Yi, Julian, Mohsen, Sam, Maryam, James, Phuc, Peiyun, Shu, Minhaeng, Zhe, also many thanks to my wonderful friends that I met during the journey in UCI.

I would never be able to finish the degree without the support and love from my close friends and my family.

CURRICULUM VITAE

Golnaz Ghiasi

EDUCATION

Doctor of Philosophy in Computer Science **2016**
University of California, Irvine *Irvine, California*

Master of Science in Computer Engineering **2010**
Amirkabir University of Technology *Tehran, Iran*

Bachelor of Science in Computer Engineering **2007**
Amirkabir University of Technology *Tehran, Iran*

REFEREED PUBLICATIONS

Parsing Occluded People **Jun 2014**
Computer Vision and Pattern Recognition (CVPR)

Occlusion Coherence: Localizing Occluded Faces with a Hierarchical Deformable Part Model **Jun 2014**
Computer Vision and Pattern Recognition (CVPR)

Using Segmentation to Predict the Absence of Occluded Parts **Sep 2015**
British Machine Vision Conference (BMVC)

Occlusion Coherence: Detecting and Localizing Occluded Faces **2016**
in revision for CVIU

Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation **Oct 2016**
European Conference on Computer Vision (ECCV)

ABSTRACT OF THE DISSERTATION

Recognizing and Segmenting Objects in the Presence of Occlusion and Clutter

By

Golnaz Ghiasi

Doctor of Philosophy in Computer Science

University of California, Irvine, 2016

Professor Charless Fowlkes, Chair

One of the fundamental problems of computer vision is to detect and localize objects such as humans and faces in images. Object detection is a building block for a wide range of applications including self-driving cars, robotics and face recognition. Though significant progress has been achieved in these tasks, it is still challenging to obtain robust results in unconstrained images. Real world scenes usually contain more than one object and it is very likely that some parts of an object are occluded by other objects in the scene. To tackle occlusion, image features generated by occlusion should be explicitly modeled rather than treated as noise. In this thesis, a deformable part model for detection and keypoint localization is introduced that explicitly models part occlusion. The proposed model structure makes it possible to augment positive training data with large numbers of synthetically occluded instances. This allows us to easily incorporate the statistics of occlusion patterns in a discriminatively trained model. To exploit bottom-up cues such as occluding contours and image segments, we extend the proposed model to utilize bottom-up class-specific segmentation in order to jointly detect and segment out the foreground pixels belonging to the object.

In these approaches, a detector for a single object category is trained which operates independently of other detections in the scene. An appealing alternative approach for detection

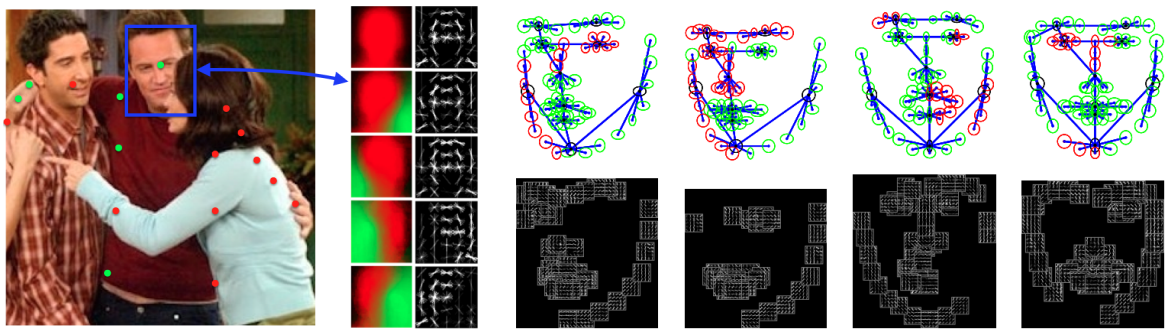
in cluttered images is to move from single object detection to whole-image parsing. The presence of occlusion can then be explained away by the presence of an occluding object. We model multi-object detection by classifying each pixel of the image (semantic segmentation) using Convolutional Neural Network. CNN architectures have terrific recognition performance but rely on spatial pooling which makes it difficult to adapt them to tasks that require dense, pixel-accurate labeling. We demonstrate that while the apparent spatial resolution of convolutional feature maps is low, the high-dimensional feature representation contains significant sub-pixel localization information. We describe a multi-resolution reconstruction architecture based on a Laplacian pyramid that uses skip connections from higher resolution feature maps and multiplicative gating to successively refine segment boundaries reconstructed from lower-resolution maps. We demonstrate that this approach yields state-of-the-art semantic segmentation results without resorting to more complex random-field inference or instance detection driven architectures.

Chapter 1

Introduction

Object detection and segmentation is one of the most important tasks in computer vision where the goal is to enable machines to detect and localize objects. Successful instance segmentation is a building block for many applications such as self-driving cars, robotics and human-computer interaction. However, the input image for these technologies is an unconstrained real word scene which usually contains many objects and, as a result, it is very likely that some parts of an object are occluded by other objects in the scene. Partially occluded objects are harder to detect and analyze due to the missing parts.

To tackle occlusion, many previous approaches have treated occluders as outliers and simply ignored image evidence in hypothesized occluded regions. However, such an approach easily confuses occluded parts with parts that are present but simply hard to detect due to unusual appearance or weak discriminability. In Chapters 2 to 4 of this thesis, we introduce methods that model image features generated by occlusion. Figure-ground cues such as the presence and shape of occluding contours as well as prototypical appearances corresponding to self-occlusion serve as positive evidence for an occlusion event. Furthermore, occlusions are induced by other objects and surfaces in the scene and hence should exhibit occlusion



(a) (b)

Figure 1.1: (a) Our goal is to develop appearance models that explain figure-ground cues generated by occlusion such as the presence and shape of occluding contours as well as prototypical appearances corresponding to self-occlusion. Our model of human body pose has a collection of templates for each part to span the range partially occluded appearances. (b) Example visualizations of our face model for different choices of shape and occlusion states for the parts. Circles represent uncertainty in the location of the part (red indicates hypothesized occlusion) and the blue lines indicate geometric constraints between those parts. The graph of these constraints forms a tree.

coherence. The statistics of typical patterns of occlusion should be learned and exploited.

An appealing alternative approach for detection in cluttered images is to move from single object detection to whole-image parsing. The presence of occlusion can then be explained away by the presence of an occluding object. Towards this goal in Chapter 5, we introduce a Convolutional Neural Network method to label every pixel of the input image with a semantic object class label.

1.1 Detection of Partially Occluded Objects

Chapter 2 and 3 introduce robust models for object detection and localization in the presence of occlusion. The introduced methods are based on a part based model which decomposes the appearance of an object into a set of local templates for parts and a set of geometric constraints between pairs of parts. Each part can have different types corresponding to different shapes (e.g. an open versus closed eye). When the connections between neighboring parts form a tree, it is possible to efficiently find the optimal locations and shapes of parts

using dynamic programming [104]. These models can capture variation in the appearance of an object and it has been shown they are successful for object detection and pose estimation task [39, 116, 104]. But, simple part-based models still suffer from occlusion: the best location or shape of the occluded parts can be estimated incorrectly and the pairwise geometric constraints can cause the optimal locations of visible parts to be pushed to incorrect locations.

In our proposed models, each part has different possible occlusion states in addition to different shapes. When the parts are relatively small as in our face model (Chapter 3), each part has two occlusion states: fully visible and occluded. If the model has N parts (for us N=68) there are 2^N possible occlusion patterns. Fortunately, not all of these occlusion patterns are plausible and the number of common ones is much smaller. For face detection, we learn a set of common occlusion patterns from training data using groupings of keypoints such as eye or nose. When the parts are larger, such as parts corresponding to a shoulder or elbow (used in our model of human body pose in Chapter 2), the part itself may only be partially visible. In this case, we can use a collection of templates to span the range partially occluded appearances based on the location of the occluding contour (Fig. 1.1(a)).

An advantage of these models is that we can compose different part types together and generate different occlusion patterns and shapes. The visualization of our face model for different choices of templates is shown in Fig. 1.1(b). We have shown we can automatically learn co-occurrence biases between part mixtures in order to favor particular patterns of part shapes and occlusion patterns.

Specifying training data from which to learn feasible occlusions poses difficulties of its own. Practically speaking, existing datasets have focused primarily on fully visible objects. Moreover, it seems unlikely that any reasonable sized set of training images would serve to densely probe the space of possible occlusions. Beyond certain weak contextual constraints, the location and identity of the occluder itself are arbitrary and largely independent of the occluded

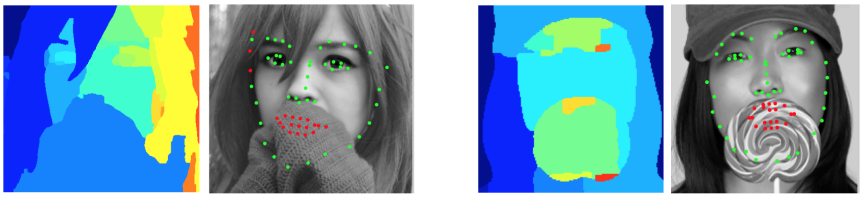


Figure 1.2: Examples of image segmentations and landmark localization results of HPM (introduced in Chapter 3) that only uses top-down cues about occlusion. Visibility predictions are not correct for all the parts in these cases. However, the foreground and occluders consist of disjoint segments, hence bottom-up segmentation should help to fix the occlusion prediction.

object. To overcome this difficulty of training data, we propose an approach for generating synthetically occluded positive training examples [44, 48]. Since we can generate an essentially infinite supply of training data, our ability to model the long-tail of rare occlusion patterns is only limited by computation.

With these models we have achieved outstanding performances. Our face model has demonstrated state-of-the-art performance for face landmark localization on the COFW [12] dataset which has significant amount of occlusion.

1.2 Combining Detection with Segmentation

The introduced method in Chapter 3 relies heavily on top-down hypotheses about occlusion to drive alignment of the model with the image. However, it does not exploit generic bottom-up cues such as detection of occluding contours. Examples of landmark localization result for two occluded faces and segmentations of the images are shown in the Fig. 1.2. Our method fails to make the correct visibility prediction for all the parts in this case. However, the foreground and occluders consist of disjoint segments, hence bottom-up segmentation should help to fix the occlusion prediction. We propose to incorporate segmentation into the detection framework by using bottom-up cues to generate candidate figure-ground seg-

mentations, providing a short list of hypothesized occluders. This should serve to reduce the combinatorial search space of possible occlusion patterns (2^N), because whole set of keypoints that are in one segment are either visible or occluded.

In Chapter 4, we formulate a joint objective that simultaneously attempts to localize parts and determine their occlusion state in a manner that is consistent with image segments suggested by edges in the image. We introduce a simple optimization approach for combining explicit part occlusion in a detection model with object-specific segmentation using an alternating minimization to fit the model to the image [45]. The integration of bottom-up segmentation cues yields further improvement over the top-down detection model, improving precision of keypoint occlusion prediction on the challenging COFW dataset.

1.3 Scene Parsing

Another approach for detection in a cluttered scene is to parse the whole image by simultaneously detect multiple objects and reason jointly about their segmentations. Toward this goal, we model multi-object detection by by jointly classifying all pixels of the image (semantic segmentation) using a Convolutional Neural Network. Deep convolutional neural networks (CNNs) have proven highly effective at semantic segmentation due to the capacity of discriminatively pre-trained feature hierarchies to robustly represent and recognize objects. A key difficulty in the adaption of CNN features to segmentation is that feature pooling layers, which introduce invariance to spatial deformations required for robust recognition, result in high-level representations with reduced spatial resolution. In Chapter 5, we investigate a "spatial-semantic uncertainty principle" for CNN hierarchies and introduce reconstruction and masking techniques that yield substantially improved segmentations [46].

1.4 Thesis Contributions

The main contributions of this thesis can be summarized in the following points:

- Modeling of occlusion in deformable part models by associating each part with different occlusion states and learning the statistics of occlusion patterns of two neighboring parts (Chapter 2, for human pose estimation).

- Modeling of occlusion in deformable part models with relatively small parts (parts are either fully visible or occluded) by adding hierarchical structure to the model. The introduced model includes intermediate part nodes which encode an intermediate representation of the occlusion state. The statistics of occlusion patterns of two neighboring part nodes are learned (Chapter 3, for face detection and landmark localization).

- Generating synthetically occluded training data from which to learn feasible occlusions poses (Chapters 2 and 3).

- Leveraging both top-down and bottom-up features for localization by combining an efficient part-based model and a binary segmentation method. Occluder boundaries and segments are detected via object-specific boundary detector and parts configurations whose occlusion pattern is consistent with the segmentation are encouraged by the scoring function of the part based model (Chapter 4).

- Reconstructing high-resolution prediction maps from low-resolution feature maps of Convolutional Neural Networks by encoding spatial information in the low-resolution high dimensional feature maps (Chapter 5).

- Introducing a multi-level Laplacian pyramid reconstruction architecture that efficiently combines semantic-rich low-resolution CNN feature map predictions with spatial details from high-resolution feature maps. It leverages accurate semantic predictions of

low-resolution feature map predictions to remove noisy class predictions from high-resolution feature maps (Chapter 5).

Chapter 2

Parsing Occluded People

Occlusion poses a significant difficulty for object recognition due to the combinatorial diversity of possible occlusion patterns. We take a strongly supervised, non-parametric approach to model occlusion by learning deformable models with many local part mixture templates using large quantities of synthetically generated training data. This allows the model to learn the appearance of different occlusion patterns including figure-ground cues such as the shapes of occluding contours as well as the co-occurrence statistics of occlusion between neighboring parts. The underlying part mixture-structure also allows the model to capture coherence of object support masks between neighboring parts and make compelling predictions of figure-ground-occluder segmentations. We test the resulting model on human pose estimation under heavy occlusion and find it produces improved localization accuracy.

2.1 Introduction

Occlusion poses a significant barrier to good recognition performance in complex cluttered scenes such as that shown in Fig. 2.1. Even when the type of occluder is known (e.g.,

Figure 2.1: The image above depicts a scene where low-level feature descriptors are dominated by occlusions. We aim to model such appearances by training models with large numbers of local mixtures that capture these occlusion statistics, yielding improvements for the task of pose estimation and visibility prediction.

other people in a crowded street) the relative layout of occluder and object is unconstrained resulting in a huge variety of possible appearances for a partially occluded object. Many approaches to detection and pose estimation treat occluders as outliers and simply ignore image evidence in hypothesized occluded regions. However, such an approach easily confuses occlusion with features that are simply hard to detect due to unusual appearance or weak discriminability.

Our goal is to develop appearance models that explain image features generated by occlusion rather than ignoring them, coupling pose estimation to segmentation. Figure-ground cues such as the presence and shape of occluding contours as well as prototypical appearances corresponding to self-occlusion serve as positive evidence for an occlusion event. To achieve this, we utilize part models in which local appearances are represented by a large library of discriminatively trained templates and their associated segmentations. Our system predicts the presence and pose of the object as well as detailed segmentation masks that contain figure, background, and occluder labels.

Unfortunately, full joint training of such high-dimensional models requires large amounts of hand-segmented training data that are representative of the huge variety of possible occlusion

patterns. Since such training data is not readily available, we approach this difficulty through the extensive use of synthetically generated data. We generate tens of thousands of images of partially occluded objects which are then used to train deformable human templates. Each such generated example comes with a complete annotation of both the object and a segmentation of the occluder. We use occluder segmentation masks as a supervisory signal to group (cluster) the space of possible occlusions and infer occlusion by enumeration over such groups. Since we can generate an essentially infinite supply of training data, our ability to model the "long-tail" of rare occlusion patterns is only limited by computation.

Inference and learning in our model is based on now-standard approaches to discriminative training of pictorial structures. In particular, training our model is quite similar to flexible part model [104, 30] which also uses local mixtures. However, to capture occlusion states requires a model with an order of magnitude more parameters and trained with several orders of magnitude more training data. This poses significant computational burden during learning and inference – using the off-the-shelf code of [104] would require 2 weeks of computation. We present several improvements to training that make such learning feasible.

Finally to demonstrate the value of modeling occlusion, we carry out an analysis of the effect of occlusion on model performance using images from the H3D [9] and "We are Family" datasets [35]. In addition to evaluating joint localization accuracy, we also evaluate occlusion/visibility prediction. We show that by modeling the appearance of occlusion, the model achieves improved accuracy over existing pose estimation techniques.

2.2 Related Work

We posit that a shortfall of many proposed occlusion models for detection is that they don't model the visual appearance of occlusion. Instead the occluded portions of the object are

described with the same model used for all background/non-object pixels. Algorithmically this means that a part is assumed occluded if it scores lower than some learned threshold. If this threshold is too high, unoccluded objects are predicted as being occluded. If this threshold is too low, occluded objects are easily confused with background. Instead we argue that occlusion should only be hypothesized if there is image evidence to support it.

Occlusion Modeling: One popular approach is to treat visibility as a binary variable that is inferred at test-time. Modeling part-level occlusion is a natural fit for models with an explicit representation of part detection. For example, the generative constellation models of Weber et. al. [95] and Fergus et. al. [41] exhaustively enumerated and scored all possible occlusion hypotheses. The supervised part models described in [5] includes templates for an occluded version of each part in the model but imposes no pairwise constraints on visibility of different parts in the model. The grammar-based model described by [51] also includes explicit occlusion part templates but enforces more structure in the pattern of occlusion, specifying a person detector that includes a variable number of parts arrayed in a vertical chain followed by an occluder part. While this grammar could be implemented with a local mixture model formally equivalent to our approach, the grammar provides an elegant and compact description of parameter sharing within the model. A major difference with our model is that [51] requires that allowed patterns of occlusion be specifically designed into the grammar. The idea we describe here sidesteps this structure learning problem, automatically learning valid occlusion patterns from data in a non-parametric way.

One drawback of part-level occlusion is that it doesn't capture the fine-scale pattern of occlusion within a part. An alternate family of techniques apply occlusion reasoning at the level of image feature maps or individual pixels that make up a template [93, 43, 94]. Spatial coherence is often enforced in such models by a Markov random field aligned with the pixel grid. However, in natural scenes the spatial statistics of occlusion patterns are not translation invariant and depend on the environment and imaging geometry (see e.g., the

Figure 2.2: We synthesize a large corpus of training data by compositing segmented objects at random locations over a base training image. The position and scale of the occluder is tied to the occluded object by a weak ground-plane assumption. When combined with keypoint annotations and segmentations from the original dataset, this yields a limitless supply of strongly supervised training data that includes keypoint visibility and occlusion segmentation masks.

results in [58]). Our modeling approach describes occlusion within each part template but enforces consistency at the level of parts rather than pixels or HOG cells which allows the dependence structure between occlusions to adapt to articulated shapes. Our model thus implicitly learns the the spatial statistics of occlusion but with the benefit of a tree-structured distribution which makes hypothesis enumeration computationally efficient.

Image Parsing: An appealing alternative is to move from single object detection to whole-image parsing. The presence of occlusion can then be "explained away" by the presence of an occluding object. For example, [103] describe a layered segmentation model for reasoning about occlusions between detected objects at the pixel level. [43] enforce mutual exclusion in the assignment of HOG cells while [11] use competitive smoothing between shape masks

associated with detectors. Our work is also closely inspired by a family of approaches that build templates for detecting the complicated appearances associated with typical object-object interactions. This includes multi-person [102, 35, 89, 77] or other multi-object [84, 30] models that implicitly capture occlusion interactions between objects.

An inherent difficulty with image parsing approaches is that they require detecting the occluding object. In a real world setting where the occluder could be arbitrary, this involves training and scoring a huge bank of object detectors on every image. Learning explicit multi-object "visual phrase" detectors may be feasible for some small set of human-object interactions, but it seems unlikely to scale to occlusion where interactions are far less constrained. *There are a limited number of ways one can reasonably ride a horse but many ways to hide a horse.* The complexity of our modeling approach lies between that of single object models and multi-object models. We train a detector for a single object category which operates independently of other detections in the scene. However, we model the appearance of occluders in a generic manner, relying on a large corpus of synthetic training data to capture the generic statistics of occlusion appearance.

Synthetic training data: Several papers have explored the use of synthetic data in training systems for recognition and pose estimation. In [86] the authors use a large set of synthetically rendered poses spanning the space of articulations in order to perform nearest-neighbor (pose) regression. [67] use green-screening to augment training data with synthetic renderings of real objects on cluttered backgrounds and [59] generated a 3-million frame dataset of synthetic images of articulated models in real backgrounds. Our work differs in using an "image-based rendering" approach, cutting and pasting existing images to yield novel ones. This is most related to [81], who fit 3D articulated models to real images, and generate synthetic renderings by slightly perturbing joint angles.

2.3 Modeling Local Occlusion Patterns

We model the appearance of occluded people by a pictorial structure with local mixtures, similar to the flexible part model of [104]. In this section we describe how the local mixture labels for each part are derived. In the next section we describe how the appearance templates are learned and combined into a joint model.

Generating synthetic images: We generate a large corpus of synthetic occlusion data by compositing segmented objects over a base training data set that has been annotated with part locations and figure-ground masks. This process automatically produces examples of occluded appearance along with supervisory information including the pixel-level support of the occluding object. In our experiments we use the H3D dataset [9] which provides segmentation masks as well as joint locations for $\sim$1500 people. We scale occluders based on object annotations in the base image to produce realistic spatial distributions (e.g., people's heads are unlikely to be occluded by others feet). The bottom of the occluder is placed below the base object and scaled linearly as a function of relative y-offset. Fig. 2.2 shows examples of such synthetic training images.

Learning part appearances: We exploit this highly-annotated synthetic training data to find clusters of training examples that capture the appearance of each part under different pose and partial occlusion conditions. Generating a large number of quality clusters is a surprisingly hard problem; typical approaches of clustering image patches [31], clustering keypoint annotation [9], or even manual grouping [117] have shown only modest performance increases with increasing numbers of clusters. Some of these difficulties are due to insufficient data (given a finite dataset, the amount of training data per cluster decreases with more clusters) and poor metrics for clustering. We found simple appearance-based clustering performs poorly since the space of occlusion patterns is high-dimensional due to the arbitrary placement and appearance of the occluder.

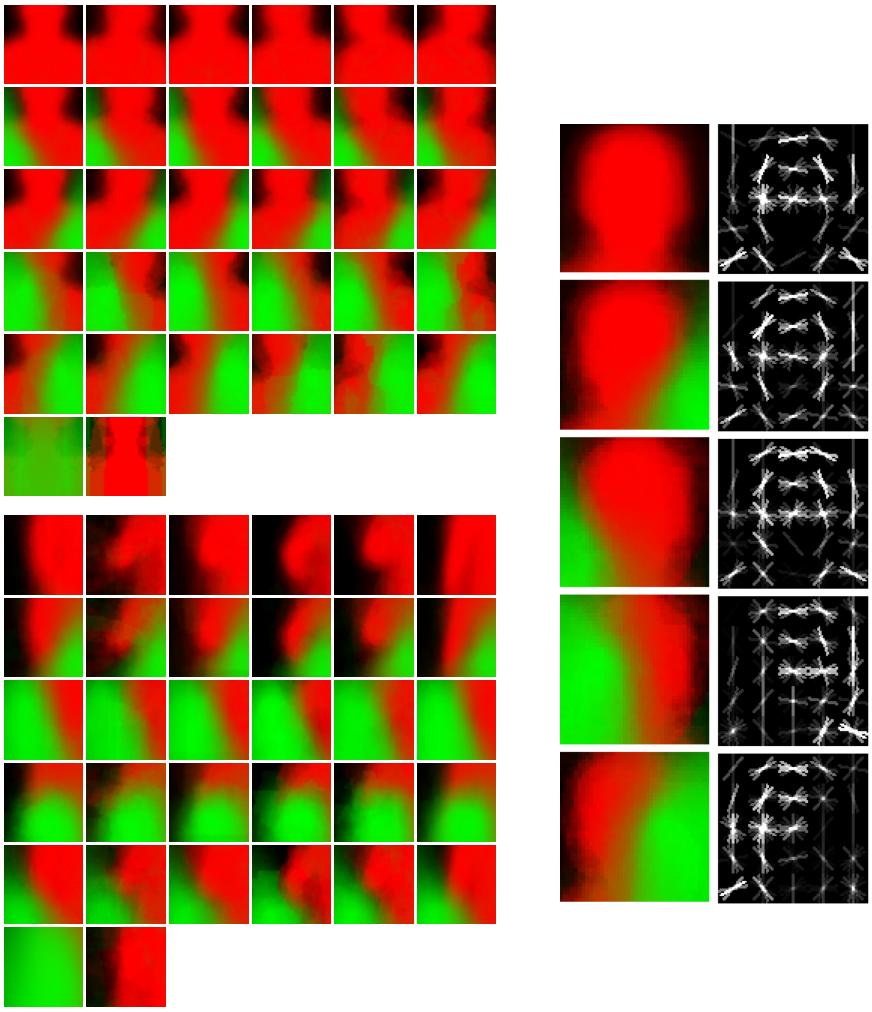
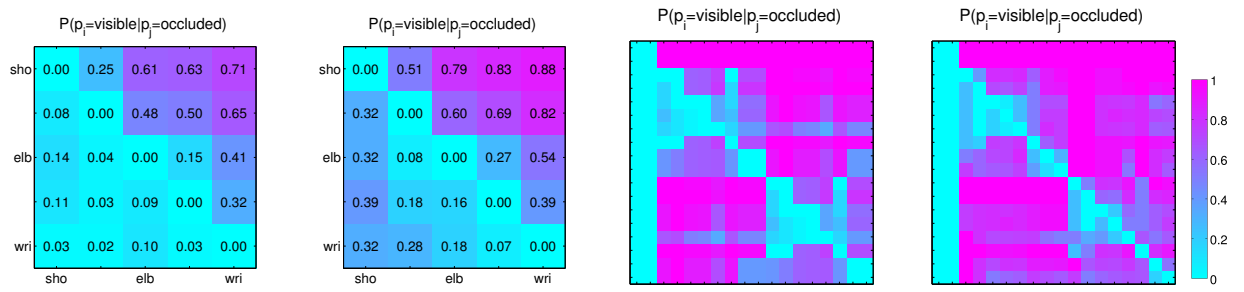


Figure 2.3: We cluster part appearances using a factored model that independently captures variation in pose (clustered on keypoint location) and occlusion patterns (clustering based on segmentation). Our model also includes separate fully-occluded and self-occluded components. Example factored clusterings for neck and elbow are shown on the left. Each row corresponds to a separate occlusion pattern and each column a separate pose. Colors show the average segmentation masks associated with each cluster. The right shows visualizations of part templates associated with different head occlusions. Relative to the unoccluded head, the partial occlusion templates include more edges oriented along the occluding contour, aiding detection.

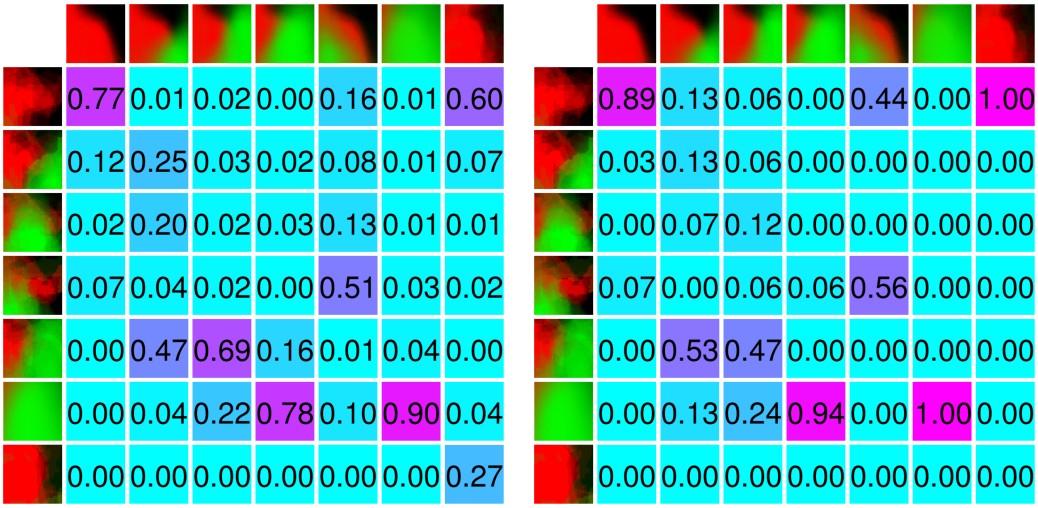
Synthetic training data addresses these difficulties in two ways. First, synthetic training data generation allows us to increases the amount of training data per cluster. Second, synthetic data comes with supervisory information in the form of occluder-object-background segmentation masks which can provide stronger metrics for clustering.



(a) training examples (b) test predictions (c) training examples (d) test predictions

Figure 2.4: Here we show co-occurrence structure between the occlusion state (visibility) of each part. The jth column contains the probability that a part i is visible conditioned on j being occluded. Panels (a) and (b) show the co-occurrence for a simple 5-part, 32-mixture model trained to localize arms. (a) gives the ground-truth visibility statistics for test data while (b) shows the statistics of the labels produced from running the model on test data. The discriminative SVM training produces a fairly good quantitative match with a slight bias towards increased visibility. Panels (c) and (d) show similar statistics for the whole upper-body model. The prominent block-structure corresponds to the head, left and right halves of the body respectively. Within each limb, occlusion at the top of the arm (e.g. shoulder) makes visibility of the lower arm unlikely while occlusion of the wrist does not strongly constrain visibility of the upper arm.

Factored occlusion-pose clustering: We separately cluster training patches for each keypoint. We label each training patch i using both a geometric pose feature g_i and a figure-background-occluder segmentation o_i. The pose feature vector describes the spatial offset of a part relative to its neighbors in the pictorial structure. The segmentation o_i is a collection of three binary masks in a window surrounding the keypoint that indicate the local segmentation of the object (as in Fig. 2.2). A naive approach is to apply a standard clustering algorithm (e.g., K-means) on concatenated descriptors with some relative weighting between the two types of features. However, our training set is severely biased due to our synthetic training data, all of which contain significant occlusions. We do not want this to adversely affect our grouping. Furthermore, from a generative perspective, we expect that the pattern of occlusion and the object pose are largely independent (one very important exception being self-occlusion). For this reason, we use a factored clustering algorithm. We generate one clustering using geometric pose features into K_g clusters with K-means. By construction, these clusters are not affected by the amount of synthetic training data. We also generate



(a) training examples (b) test predictions

Figure 2.5: Conditional probabilities for occlusion states of the elbow given the shoulder in (a) the ground-truth synthetic training data and (b) mixtures selected by the model on test data. The model learns coherency of the segmentation. E.g., if the left side of the shoulder is occluded (5th column) then the elbow tends to be visible (1st row) or also occluded on the left (4th row). The 7th row corresponds to self-occlusion.

a second independent clustering of the occluder masks into K_o clusters. We finally assign each training example to an element of the "cross-product" space of $K_g \times K_o$ clusters, or to fully or self-occluded mixtures.

Fig. 2.3 shows an example of such clusterings for several different object parts. Each row corresponds to different occlusion clusters while each column corresponds to pose clusters. We also include two additional clusters, a fully-occluded cluster and a self-occluded cluster. The appearance of a fully-occluded part is assumed to be independent of the pose since it only includes image features arising from the occluder. An example is assigned to the self-occluded cluster when the part is invisible in the image even though there is no occluder present (e.g., hands clasped behind the head). Self-occlusion could presumably benefit from multiple pose clusters but this requires sufficient training data and our synthesis approach cannot automatically generate such self-occlusions

Cluster statistics: Occlusions of parts are not independent. If a person's elbow is occluded

by the elbow of another person, then shoulder may also be occluded (by that occluders' shoulder). This implies that cluster labels across neighboring joints may have very specific co-occurrence statistics. We visualize examples of such statistics in Fig. 2.4 and Fig. 2.5.

2.4 Occlusion-aware Part Models

We now describe a method for training deformable pictorial structures with tens of thousands of images of human poses (under heavy occlusion).

Deformable Templates: Our model consists of a set of parts V and pairwise relations E which encode joint constraints on part locations and appearances. Let I be an image, $p_i = (x, y)$ be the location for part $i \in V$ and m_i be the mixture component of part i. Each local mixture m_i corresponds to a occlusion-pose cluster learned for that part from the synthetic training data. If part i corresponds to the left elbow, and m_i selects the coarse elbow orientation along with a particular local pattern of occlusion (including full and self-occlusion). Each choice of m_i is associated with an average figure-ground-occluder mask for the cluster which can be used to predict keypoint visibility and segmentation at test time.

Given an image, we score a collection of hypothesized part locations and local mixture selections with the following objective:

$$S(I, p, m) = \sum_{i \in V} \left[\alpha_i^{m_i} \cdot \phi(I, p_i) \right] + \sum_{ij \in E} \left[\beta_{ij}^{m_i, m_j} \cdot \psi(p_i - p_j) + \gamma_{ij}^{m_i, m_j} \right] \tag{2.1}$$

The first term scores the appearance evidence for placing a template $\alpha_i^{m_i}$ for part i, tuned for mixture m_i, at location p_i. We write $\phi(I, p_i)$ for the feature vector (e.g., HOG descriptor [27]) extracted from pixel location p_i in image I. Note that we define a separate template for

each mixture, even occluded states as such templates will capture visual features associated with occlusions.

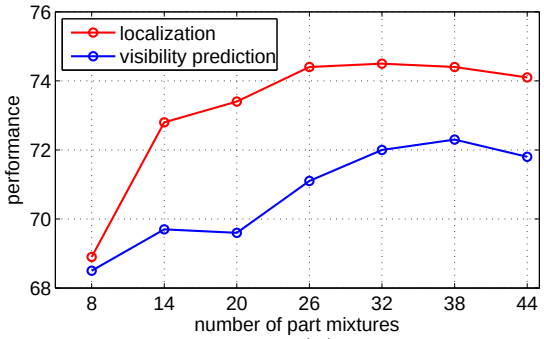
The second term scores relational constraints between pairs of parts. The feature $\psi(p_i - p_j) = \begin{bmatrix} dx & dx^2 & dy & dy^2 \end{bmatrix}$ is a vector of relative offsets between part i and part j and the parameters $\beta_{ij}^{m_i,m_j}$ specify the relative rest location and quadratic spring penalty for deviating from that rest location. Both the spring and the bias, $\gamma_{ij}^{m_i,m_j}$, depend on the local mixtures m_i and m_j selected for parts i and j. This allows the relational model to capture dependencies between visibility of neighboring parts in the model (as in Fig. 2.4) as well as providing a much richer, non-parametric description of pose and appearance than is possible with a single local template and spring.

Learning and Inference: Given a test image, we seek the maximum scoring part arrangement p and mixture assignments m. When E is tree-structured, this solution can be computed efficiently with dynamic programming [38, 104].

Let (p^n, m^n) be the ground-truth part locations and mixture labels provided for the nth positive training example. We learn model parameters $w = (\alpha, \beta, \gamma)$ using a variant on the structured SVM.

$$\begin{aligned} \underset{w,\xi_i \geq 0}{\text{argmin}} \quad & \frac{1}{2}||w||^2 + C\sum_n \xi_n \\ \text{s.t.} \quad \forall n \in \text{pos} \quad & w \cdot \Phi(I^n, p^n, m^n) \geq 1 - \xi_n \\ \forall n \in \text{neg}, \forall p, m \quad & w \cdot \Phi(I^n, p, m) \leq -1 + \xi_n \end{aligned} \tag{2.2}$$

The above quadratic program (QP) attempts to learn a low-norm w that scores positive examples above 1 (evaluated at ground-truth part locations and mixture labels) and scores negative examples below -1 (for any setting of part locations and mixtures). We use a standard cutting-plane approach to incrementally add negative constraints by running the



(a)

	H3D		H3D Occluded		H3D Synthetic	
	pck	ocl	pck	ocl	pck	ocl
FMP6 [104]	71.5	80.1	55.6	64.2	50.4	57.8
FMP6.1+syn	70.5	79.4	60.6	69.0	56.2	62.5
OMP32	71.5	78.4	55.5	62.0	59.0	67.4
OMP32+syn	68.7	72.6	68.5	72.0	70.4	73.5
OMP32+syn+struct	70.0	74.4	68.5	72.8	71.1	74.5

(b)

Figure 2.6: (a) Localization and visibility prediction accuracy on synthetically occluded test data as a function of the number of mixture model components. We found that performance saturated above 32 mixtures. (b) Evaluation of performance on H3D test images, a subset containing heavy occlusion and a set of synthetically occluded examples. The benefits of modeling occlusion are more pronounced on the synthetic and heavily occluded subsets. FMP6.1 is a baseline model with a single mixture representing occlusion so it can exploit synthetic training data.

detector on negative training images in order to find a subset of constraints that are active at the optimum.

Mislabeled positives: In order to improve localization and occlusion prediction, we also added incorrectly-labeled positive images as negative examples. To do so, we include the following negative constraints to our QP:

$$\forall p, n \in \text{pos}, m \nsim m^n \quad w \cdot \Phi(I^n, p, m) \leq -1 + \xi_n \tag{2.3}$$

We say that two sets of part-mixture assignments m and m' are in the same equivalence class iff they predict the same set of parts are visible. We use the notation $m \nsim m^n$ to mean that the mixture assignments m are in a different equivalence class than the ground-truth. This constraint thus enforces that for a given positive example n, all poses associated with *incorrect* mixture assignments corresponding to an incorrect occlusion prediction score below -1. These constraints are similar to those found in traditional structured prediction (where the true label should outscore incorrect labels by a margin), but split into positive and

negative constraints. We find that this splitting speeds up optimization without sacrificing performance.

Semi-latent learning: Although the part locations and mixture labels are given in our positive training data, we found it was useful to perform re-estimation of part locations. This is particularly important for occluded parts where the discriminative appearance features being learned (the presence of an occluding contour) is more dependent on the position of the occluder than on the the part keypoint location.

We used a standard latent learning approach [37] to alternately train a model using convex optimization and then re-estimate the locations p^n for the set of parts. For each positive example n, let $\Omega_n = \{p : |p_i - p_i^n| < r\}$ denote a set of possible part locations that lie near the ground-truth location p^n. We learn a model w and update part locations p^n with a coordinate descent algorithm:

1. *Model update:* Learn w with a QP (2.2) using the inferred positive part locations p^n.
2. *Latent assignment:* Compute $p^n = \max_{p \in \Omega_n} w \cdot \phi(I^n, p, m^n)$ for $n \in$ pos.

Note that during this learning we do not update the mixture assignments, instead relying on the reliable ground-truth clustering.

Computational bottleneck: Typically, the computational bottleneck of latent SVM training[37] is Step 1, which requires passing over a large set of negative images (typically on the order of a thousand) and performing "hard negative" mining. In our case, the computational bottleneck is Step 2, since we now have *hundreds of thousands* of positive examples in our synthetic dataset. A standard approach for latent updating of positive examples is to evaluate the model w as a "detector" on each positive example. A simple but crucial observation is that in the semi-latent setting the mixture label m^n is known and *not* latently updated, so only a *single* filter per part need be evaluated. Modifying the released code of [104] to allow for

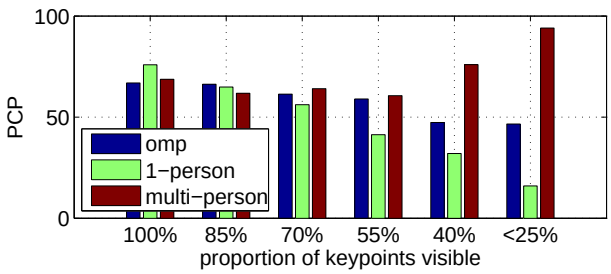
this efficient semi-latent update produced an order-of-magnitude speed up, reducing training time from over a week to under a day. We also note the latent assignment can be trivially parallelized.

2.5 Experimental Results

Dataset: We use H3D as our primary source of training and testing data. Since H3D contains many challenging poses with different points of view and our baseline model [104] still struggles with non-frontal poses, we selected a subset of 668 images with frontal facing people. For our synthetic training experiments, each original training image was augmented with 100 different synthetic occlusions, yielding a training set of half a million positive images. We use negative training images from the INRIAPerson database [27] and evaluate models using 190 test images from H3D. Additionally, we also evaluate our model on the "We Are Family" dataset (WAF) and benchmark provided by [35].

To better understand model performance, we considered variants of each test dataset which were enriched for occlusion. We evaluated on 3 different variants of the H3D dataset. The original 190 test images, a subset of 60 which were selected as heavily occluded (many invisible ground-truth keypoints), and an a synthetically occluded version of the 190 original test examples. For WAF, we considered six subsets of data based on the proportion of visible keypoints. In the WAF dataset, there are 6 parts or "sticks" (head, torso, upper arms and right arms) labeled in the data set, each with a visibility variable. We build six subsets according to the number of total visible sticks.

Evaluation: For evaluation on H3D we use the percentage of correctly localized keypoints (PCK) criteria used by [104]. A predicted keypoint is considered as correctly localized when it lies within a scale-normalized threshold distance (half the head height) of the ground-truth



	WeAreFamily	
	pcp	ocl
FMP6 [104]	58.0	74.5
FMP6.1+syn	60.4	74.2
OMP32+syn	61.9	75.2
OMP32+syn+WAF$_{train}$	63.6	74.0
1-Person [35]	58.6	73.9
Multi-Person [35]	69.4	80.0

Figure 2.7: Performance on subsets of the WeAreFamily dataset as a function of the amount of occlusion present. Our model (OMP) achieves a better PCP score than the 1-person model baseline in [35], primarily due to better handling of occluded examples. The much more complicated multi-person model of [35] outperforms our model for heavy occlusion ($< 50\%$ of keypoints visible) but does so at a loss of localization accuracy relative to the 1-person baseline. Table shows overall PCP and occlusion prediction accuracy.

keypoint location. To extend the PCK criteria for occluded body parts, we require that any keypoint marked invisible in the ground-truth must correctly be predicted as occluded by the model. To separate out errors in localization from errors in visibility prediction, we also compute the accuracy of part visibility prediction as a binary classification task.

For WAF we used the percentage of correctly localized parts (PCP) criteria of [35] which is similar to PCK but measures the localization of the sticks rather than their endpoints. Similar to H3D we also measure stick occlusion prediction based on stick visibility marked in the ground-truth. We use the set of upper-body detections provided with the WAF dataset which is based on a combined face and body detector and achieves an 86% detection rate.

Model Complexity: In order to choose number of mixtures per part, we evaluated performance while varying K_o to generate models with up to 44 mixtures per part. Fig. 2.6 shows test localization and visibility prediction accuracy as a function of the number of part mixtures. We chose $K_o = 5$ which yielded 32 mixture components and offered a good trade-off between performance and running time. Increasing the amount of synthetic training data did not seem to change the saturation point. In fact using $10x$ data gave similar performance to $100x$ data in many cases. This suggests that we may need more variety in the shapes of our synthetic occluders in order to usefully grow the number of part mixtures further.

Synthetic Training Data: Fig. 2.6 shows the performance of our 32-mixture model (OMP32) under a variety of training and test conditions. Training the model without synthetic occlusion data (OMP32) yielded better performance on the original H3D data but including synthetic occlusions (OMP32+syn) gave very substantial improvements on the occlusion enriched datasets. Training the model with mis-labeled positive examples (OMP32+syn+struct) gave significant improvements in part localization accuracy which boosted performance on the un-enriched original dataset.

Comparative Evaluation: Fig. 2.6 also compares the proposed model (OMP32+syn+struct) to several baselines. FMP6 is based on the code of [104]. We train a baseline version of the FMP on the H3D dataset which always predicts all parts visible at test time. Excluding occluded data during training produced a model which was slightly worse (PCK=69.6). To allow the FMP to predict visibility we also trained a version with the addition of a single occluded mixture component for each part (FMP6.1). This model achieved improved occlusion accuracy since it could predict occlusions at test time. With only 1 occlusion mixture, the addition of synthetic data offered some further improvement in occlusion prediction but at the expense of localization accuracy.

Fig. 2.7 compares the performance of several models on the "We are Family" dataset including the FMP baseline, the proposed OMP model, as well as a single and multi-person model proposed by [35]. By training the OMP model with synthetic data, we outperform the 1-Person model despite training on a completely different dataset. Including the WAF positive examples in our training set improved OMP performance further. We also examined performance as a function of occlusion level. For all but the most extreme occlusions, our model achieves a similar PCP to the Multi-Person model. This is particularly surprising since the Multi-Person detector performs joint inference over a collection of detector outputs and incorporates other image cues in order to find a depth ordering of detections.

2.6 Conclusions

In this chapter, we have presented a method for modeling occlusion that is aimed at explicitly learning the appearance and statistics of occlusion patterns. Our system produces models which are more robust to heavy occlusion than existing approaches. As an added benefit, our model explicitly represents part occlusions and hence can predict not only part locations but a local segmentation mask. The combination of synthetic training data and flexible models with many part appearance mixtures is in some sense "brute force" and perhaps less elegant that some more parametric approach. However, it has the distinct advantage of being amenable to discriminative learning and, as we have shown, capable of not only learning detailed occlusion statistics but also achieving competitive performance at the task of human pose estimation.

The introduced model in this chapter has a collection of templates for each part to span the range of partially occluded appearances. But when the parts are relatively small as in a face model [116], each part is either fully visible or occluded. In this case, learning co-occurrence statistics of occlusion between neighboring parts of a tree-structured model is not sufficient to model likely patterns of occlusion. In the next chapter, we introduce a hierarchical part based model for face landmark localization and detection which enforces occlusion coherence across neighboring landmarks using a set of intermediate nodes.

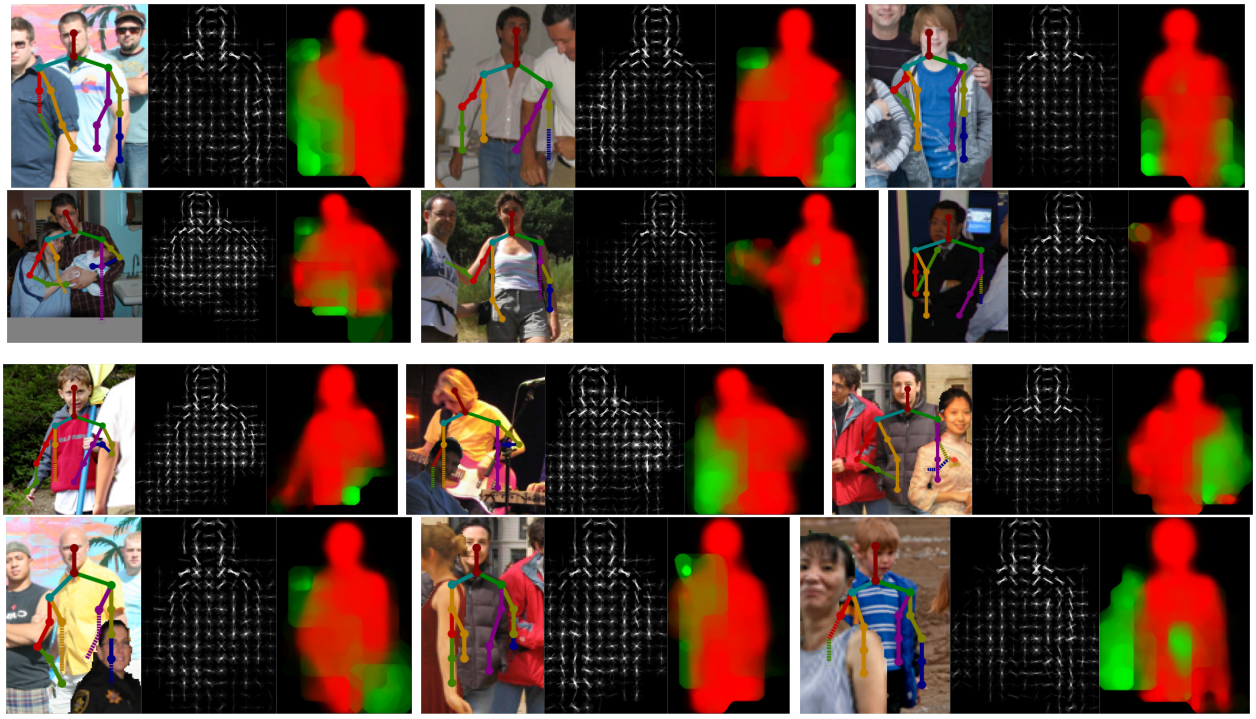


Figure 2.8: Examples of pose and occlusion estimation for images from the H3D test dataset. Each image shows the keypoint localization (dashed lines indicate occluded), a visualization of the deformed HOG template and a occluder-figure-ground segmentation estimated by compositing the predictions of individual part mixtures. Each of the 18 parts can take on one of 32 mixture (occlusion) states allowing for 32^{18} possible occlusion patterns. The top two rows show examples from H3D, the bottom two from H3D Synthetic.

Chapter 3

Occlusion Coherence: Detecting and Localizing Occluded Faces

In this chapter, we describe a hierarchical deformable part model for face detection and landmark localization that explicitly models part occlusion. In contrast with our part based model for human pose estimation (Chapter 2) that parts may only be partially visible, parts of our face model are relatively small and they are either fully visible or occluded. To enforce coherence across neighboring landmarks, we add hierarchical structure to the model. Our model includes intermediate part nodes which encode an intermediate representation of the occlusion state. The statistics of occlusion patterns of two neighboring part nodes are learned.

We augment positive training data with large numbers of synthetically occluded instances. This allows us to easily incorporate the statistics of occlusion patterns in a discriminatively trained model. We test the model on several benchmarks for landmark localization and detection including challenging new data sets featuring significant occlusion. We find that the addition of an explicit occlusion model yields a detection system that outperforms existing

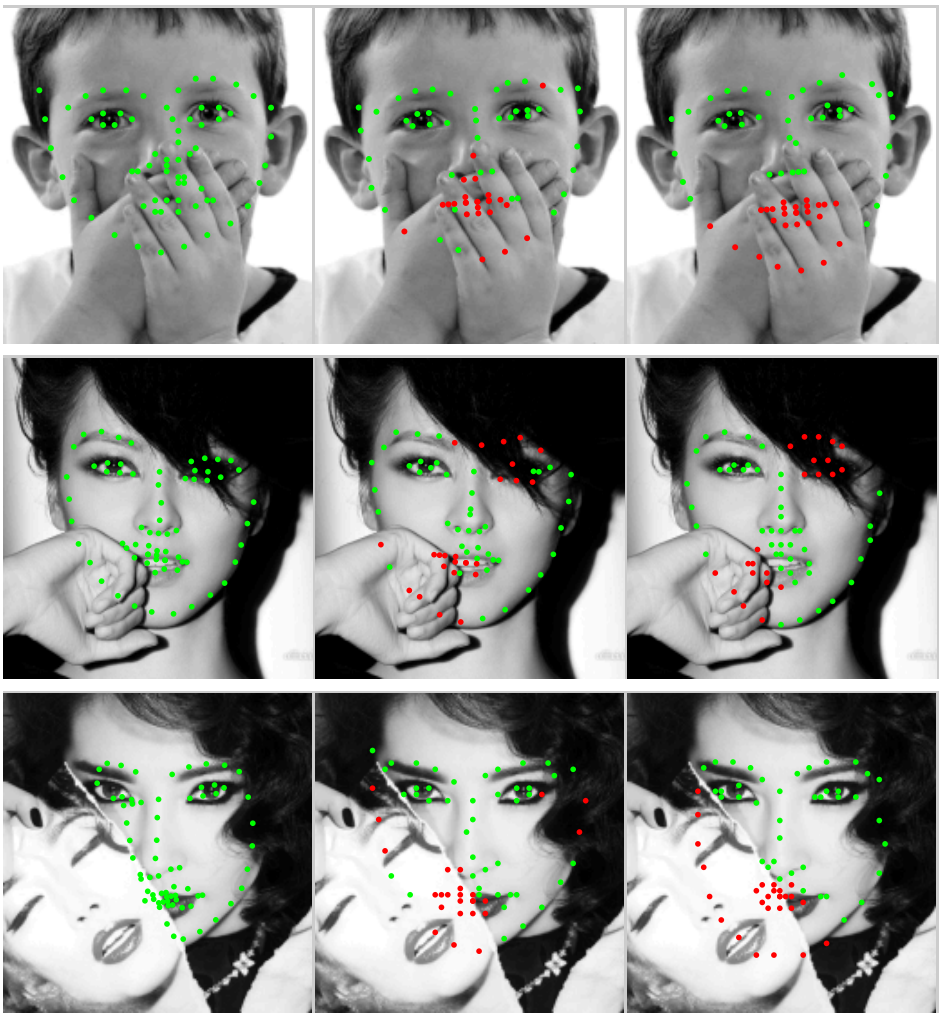
approaches for occluded instances while maintaining competitive accuracy in detection and landmark localization for unoccluded instances.

3.1 Introduction

Accurate localization of facial landmarks provides an important building block for many applications including identification [7] and analysis of facial expressions [72]. Significant progress has been made in this task, aided in part by the fact that faces have less intra-category shape variation and limited articulation compared to other object categories of interest. However, feature point localization tends to break down when applied to faces in real scenes where other objects in the scene (hair, sunglasses, other people) are likely to occlude parts of the face. Fig. 3.1(a) depicts the output of a deformable part model [116] where the presence of occluders distorts the final alignment of the model.

A standard approach to handling occlusion in part-based models is to compete part feature scores against a generic background model or fixed threshold (as in Fig. 3.1(b)). However, setting such thresholds is fraught with difficulty since it is hard to distinguish between parts that are present but simply hard to detect (e.g., due to unusual lighting) and those which are genuinely hidden behind another object.

Treating occlusions as an unstructured source of noise ignores a key aspect of the problem, namely that occlusions are induced by other objects and surfaces in the scene and hence should exhibit **occlusion coherence**. For example, it would seem very unlikely that every-other landmark along an object contour would happen to be occluded. Yet many occlusion models make strong independence assumptions about occlusion, making it difficult to distinguish *a priori* likely from unlikely patterns. Ultimately, an occluder should not be inferred simply by the lack of evidence for object features, but rather by positive evidence for the



(a) (b) (c)

Figure 3.1: Occlusion impacts part localization performance. In panel (a) the output of a deformable part model [116] is distorted by the presence of occluders, disrupting localization even for parts that are far from the site of occlusion. (b) Introducing independent occlusion of each part results in better alignment but occlusion is treated as an outlier process and prediction of occlusion state is inaccurate. (c) The output of our hierarchical part model, which explicitly models likely patterns of occlusion, shows improved localization as well as accurate prediction of which landmarks are occluded.

occluding object that **explains away** the lack of object features.

The contribution of this chapter is an efficient hierarchical deformable part model that encodes these principles for modeling occlusion and achieves state-of-the-art performance on benchmarks for occluded face localization and detection (depicted in Fig. 3.1(c)). We model the face by an arrangement of parts, each of which is in turn composed of local

landmark features. This two-layer model provides a compact, discriminative representation for the appearance and deformations of parts. It also captures the correlation in shapes and occlusion patterns of neighboring parts (e.g., if the chin is occluded it would seem more likely the bottom half of the mouth is also occluded). In addition to representing the face shape, each part has an associated occlusion state chosen from a small set of possible occlusion patterns, enforcing coherence across neighboring landmarks and providing a sparse representation of the occluder shape where it intersects the part. We describe the details of this model in Section 3.3.

Specifying training data from which to learn feasible occlusion patterns comes with an additional set of difficulties. Practically speaking, existing datasets have focused primarily on fully visible faces. Moreover, it seems unlikely that any reasonable sized set of training images would serve to densely probe the space of possible occlusions. Beyond certain weak contextual constraints, the location and identity of the occluder itself are arbitrary and largely independent of the occluded object. To overcome this difficulty of training data, we propose a unique approach for generating synthetically occluded positive training examples. By exploiting the structural assumptions built into our model, we are able to include such examples as "virtual training data" without explicitly synthesizing new images. This in turn leads to an interesting formulation of discriminative training using a loss function that depends on the latent occlusion state of the parts for negative training examples which we describe in Section 4.2.2.

We carry out an extensive analysis of this model performance in terms of landmark localization, occlusion prediction and detection accuracy. While our model is trained as a detector, the internal structure of the model allows it to perform high-quality landmark localization, comparable in accuracy to pose regression, while being more robust to initialization and occlusions (Section 3.5.1). To carry out an empirical comparison to recently published models, we provide a new set of 68-landmark annotations for the Caltech Occluded Faces in

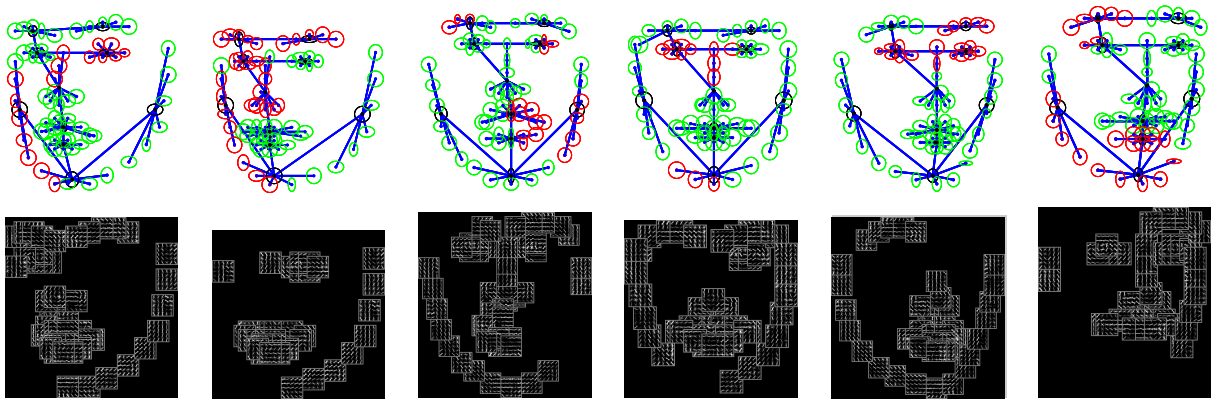


Figure 3.2: Our model consists of a tree of parts (black circles) each of which is connected to a set of landmarks (green or red) in a star topology. The examples here show templates corresponding to different choices of part shape and occlusion patterns. Red indicate occluded landmarks. Shape parameters are independent of occlusion state. Landmark appearance is modeled with a small HOG template (2nd row) and occluded landmarks are constrained to have an appearance template fixed to 0. Note how the model produces a wide range of plausible shape configurations and occlusion patterns.

the Wild (COFW) benchmark dataset. We find that not only the localization but also the prediction of which landmarks are occluded is improved over simple independent occlusion models (Section 3.5.2). Unlike landmark regression methods, our model does not require initialization and achieves good performance on standard face detection benchmarks such as FDDB [60]. Finally, to illustrate the impact of occlusion on existing detection models, we evaluate performance on a new face detection dataset that contains significant numbers of partially occluded faces (Section 3.5.3).

3.2 Related Work

Face Detection and Localization There is a long history of face detection in the computer vision literature. A classic approach treats detection as problem aligning a model to a test image using techniques such as Deformable Templates [107], Active Appearance Models (AAMs) [23, 73, 74] and elastic graph matching [96]. Alignment with full 3D models provides even richer information at the cost of additional computation [53, 7]. A key difficulty in many

of these approaches is the dependence on iterative and local search techniques for optimizing model alignment with a query image. This typically results in high computational cost and the concern that local minima may undermine system performance.

Recently, approaches based on *pose regression*, which train regressors that predict landmark locations from both appearance and spatial context provided by other detector responses, has also shown impressive performance [91, 34, 6, 12, 14, 28, 98, 82, 115]. While these approaches lack an explicit model of face shape, stage-wise pose-regression models can be trained efficiently in a discriminative fashion and thus sidestep the optimization problems of global model alignment while providing fast, feed-forward performance at test time.

Pose-regression is flexible in the choice of features and regressors used. Supervised Descent Method (SDM) [98] employs linear regression on SIFT features to compute shape increments. ESR [14] and RCPR [12] predict shape increments using simple pixel-difference features and boosted ferns. LBF [82] learns a set of binary features and a regression function using random forest regression. Zhu et al. proposed a Coarse-to-Fine Shape Searching method (CFSS) [115] in which at each stage a cascade of linear regressors are used to calculate a finer sub-space (represented as a center and scope). The incorporation of Deep Convolutional Neural Network features has allowed further improvements by using raw image pixels as input instead of hand-designed features and allows end-to-end training. Zhang et al. proposed successive auto-encoder networks (CFAN) to perform coarse-to-fine alignment [110]. TCDCN [111] train a multi-task DCNN jointly for landmark localization along with prediction of other facial attributes. They show that facial attributes such as gender and expression can help in learning a robust landmark detector.

Our model is most closely related to the work of [116], which applies discriminatively trained deformable part models (DPM) [39] to face analysis. This offers an intermediate between the extremes of model alignment and landmark regression by utilizing mixtures of simplified shape models that make efficient global optimization of part placements feasible while

exploiting discriminative training criteria. Similar to [104], we use local part and landmark mixtures to encode richer multi-modal shape distributions. We extend this line of work by adding hierarchical structure and explicit occlusion to the model. We introduce intermediate part nodes that do not have an associated "root template" but instead serve to encode an intermediate representation of occlusion and shape state. The notion of hierarchical part models has been explored extensively as a tool for compositional representation and parameter sharing (see e.g., [114, 51]). While the intermediate state represented in such models can often be formally encoded in by non-hierarchical models with expanded state spaces and tied parameters, our experiments show that the particular choice of model structure proves essential for efficient representation and inference.

Occlusion Modeling Modeling occlusion is a natural fit for recognition systems with an explicit representation of parts. Work on generative constellation models [95, 41] learned parameters of a full joint distribution over the probability of part occlusion and relied on brute force enumeration for inference, a strategy that doesn't scale to large numbers of landmarks. More commonly, part occlusions are treated independently which makes computation and representation more efficient. For example, the supervised detection model of [5] associates with each part a binary variable indicating occlusion and learns a corresponding appearance template for the occluded state.

The authors of [51] impose a more structured distribution on the possible occlusion patterns by specifying grammar that generates a person detector as a variable length vertical chain of parts terminated by an occluder template, while [22] allows "flexible compositions" which correspond to occlusion patterns that leave visible a connected subgraph of the original tree-structure part model. Our approach provides a stronger model than full independence, capturing correlations between occlusions of non-neighboring landmarks. Unlike the grammar-based approach, occlusion patterns are not specified structurally but instead

learned from data and encoded in the model weights.

Pose regression approaches have also been adapted to incorporate explicit occlusion modeling. For example, the face model of [85] uses a robust m-estimator which serves to truncate part responses that fall below a certain threshold. In our experiments, we compare our results to the recent work of [12] which uses occlusion annotations when training a cascade of regressors where each layer predicts both part locations and occlusion states.

3.3 Hierarchical Part Model

In this section we develop a hierarchical part model that simultaneously captures face appearance, shape and occlusion. Fig. 3.2 shows a graphical depiction of the model structure. The model has two layers: the face consists of a collection of parts (nose, eyes, lips) each of which is in turn composed of a number of landmarks that specify local edge features making up the part. Landmarks are connected to their parent part nodes with a star topology while the connections between parts forms a tree. In addition to location, each part takes one of a discrete set of shape states (corresponding to different facial shapes or expressions) and occlusion states (corresponding to different patterns of visibility). The model topology which groups facial features into parts was specified by hand while the shape and occlusion patterns are learned automatically from training data (see Section 4.2.2). This model, which we term a hierarchical part model (HPM) is a close cousin of the deformable part model (DPM) of [39] and the flexible part model (FMP) of [116]. It differs in the addition of part nodes that model shape but don't include any "root filter" appearance term, and by the use of mixtures to model occlusion patterns for each part. In this section we introduce some formal notation to describe the model and some important algorithmic details for performing efficient message passing during inference.

3.3.1 Model Structure

Let l, s, o denote the hypothesized locations, shape and occlusion of N_p parts and N_l landmarks describing the face. Locations $l \in \mathbb{R}^{2N}$ range over the whole image domain and $o \in \mathcal{O}_1 \times \mathcal{O}_2 \ldots \times \mathcal{O}_N$ indicates the occlusion states of parts and landmarks and $N = N_p + N_l$. The shape $s \in \mathcal{S}_1 \times \mathcal{S}_2 \ldots \times \mathcal{S}_N$ selects one of a discrete set of shape mixture components for each part. We define a tree-structured scoring function by:

$$Q(l, s, o|I) = \sum_i \phi_i(l_i, s_i, o_i|I) \tag{3.1}$$
$$+ \sum_i \sum_{j \in child(i)} \psi_{ij}(l_i, l_j, s_i, s_j) + b_{ij}(s_i, s_j, o_i, o_j)$$

where the potential ϕ scores the consistency of the local image appearance around location l_i, ψ is a quadratic shape deformation penalty, and b is a co-occurrence bias.

The first (unary) term scores the appearance evidence. We linearly parameterize the unary appearance term with filter weights $w_i^{s_i}$ that depend on the discrete shape mixture selected

$$\phi_i(l_i, s_i, o_i|I) = w_i^{s_i} \cdot \phi(l_i, o_i|I)$$

Appearance templates are only associated with the leaves (landmarks) in the model so the unary term only sums over those leaf nodes. The occlusion variables o_i for the landmarks are binary, corresponding to either occluded or visible. If the ith landmark is unoccluded, the appearance feature ϕ is given by a HOG [27] feature extracted at location l_i, otherwise the feature is set to 0. This is natural on theoretical grounds since the appearance of the occluder is arbitrary and hence indistinguishable from background based on its local appearance. Empirically we have found that unconstrained occluder templates learned with sufficiently varied data do in fact have very small norms, further justifying this choice [48].

The second (pairwise) term in Eq. 1 scores the placement part j based on its location relative to its parent i and the shape mixtures of the child and parent. We model this with a linearly parameterized function:

$$\psi_{ij}(l_i, l_j, s_i, s_j) = w_{ij}^{s_i,s_j} \cdot \psi(l_i - l_j)$$

where the feature ψ includes the x and y displacements and their cross-terms, allowing the weights w_{ij} to encode a standard quadratic "spring". We assume that the shape of the parts is independent of any occluder so the spring weights do not depend on the occlusion states. [1] The pairwise parameter b_{ij} encodes a bias of particular occlusion patterns and shapes to co-occur. Formally, each landmark has the same number of occlusion states and shape mixtures as its parent part, but we fix the bias parameters between the part and its constituent landmarks to impose a hard constraint that the mixture assignments are compatible.

3.3.2 Efficient Message Passing

The model above can be made formally equivalent to the FMP model used in [104] by introducing local mixture variables that live in the cross-product space of o_i and s_i. However, this reduction fails to exploit the structure of the occlusion model. This is particularly important due to the large size of the model. Naive inference is quite slow due to the large number of landmarks and parts (N=68+10), and huge state space for each node which includes location, occlusion pattern and shape mixtures. Consider the message passed from one part to another where each part has L possible locations, S shape mixtures and O occlusion patterns. In general this requires minimizing over functions of size $(LSO)^2$ or $L(SO)^2$ when using the distance transform. In the models we test, $SO = 12$ which poses a

[1] In practice we find it is sufficient for the deformation cost to only depend on the child shape mixture, i.e. $\psi_{ij}(l_i, l_j, s_i, s_j) = w_{ij}^{s_j} \cdot \psi(l_i - l_j)$ which gives a factor S speedup with little decrease in performance.

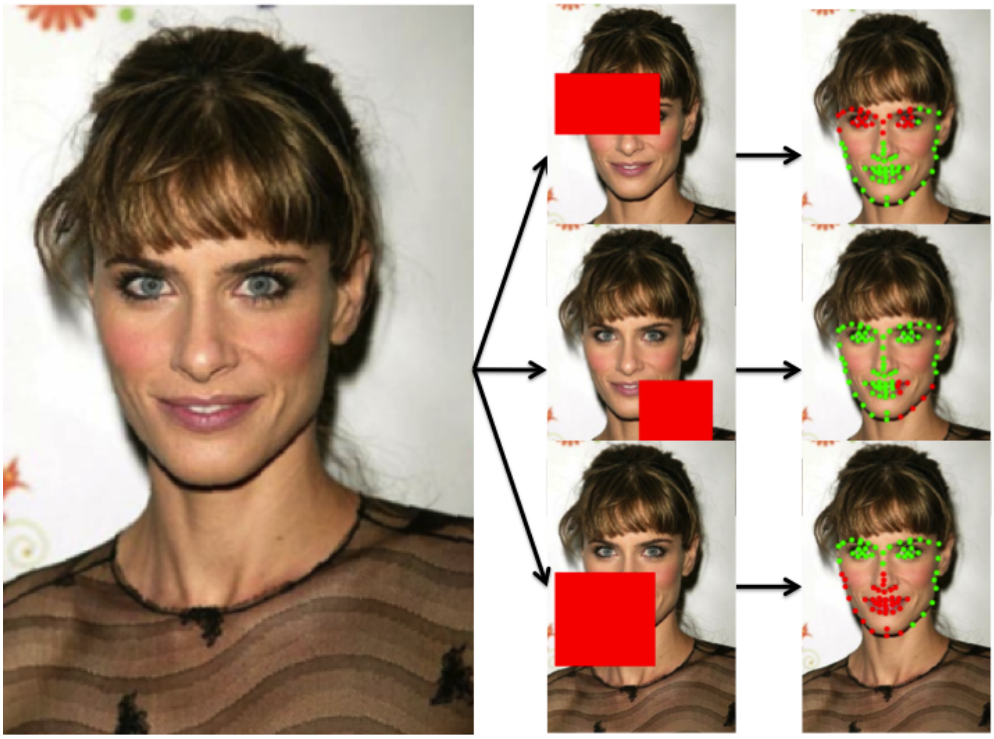


Figure 3.3: Virtual positive examples are generated synthetically by starting with a fully visible training example and sampling random coherent occlusion patterns.

substantial computation and memory cost, particularly for high-resolution images where L is large.

Part-Part messages While the factorization of shape and occlusion doesn't change the asymptotic complexity, we can reduce the runtime in practice by exploiting distributivity of the distance transform over max to share computations. Standard message passing from part j to part i requires that we compute:

$$\mu_{j\to i}(l_i, s_i, o_i) = \max_{l_j, s_j, o_j} \Big[\psi_{ij}(l_i, l_j, s_i, s_j) + \sum_{k \in child(j)} \mu_{k\to j}(l_j, s_j, o_j) + b_{ij}(s_i, s_j, o_i, o_j) \Big]$$

where we have dropped the unary term ϕ_j which is 0 for parts. Since the bias doesn't depend on the location of parts we can carry out the computation in two steps:

$$\nu_{ij}(l_j, s_i, s_j, o_j) = \max_{l_j} \left[\psi_{ij}(l_i, l_j, s_i, s_j) + \sum_{k \in child(j)} \mu_{k \to j}(l_j, s_j, o_j) \right]$$

$$\mu_{j \to i}(l_i, s_i, o_i) = \max_{s_j, o_j} \left[\nu_{ij}(l_j, s_i, s_j, o_j) + b_{ij}(s_i, s_j, o_i, o_j) \right]$$

which only requires computing S^2O distance transforms.

Landmark-Part messages In our model the occlusion and shape variables for a landmark are determined completely by the parent part state. Since the score is known for an occluded landmark in advance, it is not necessary to compute distance transforms for those components. We write this computation as:

$$\nu_{jk}(l_j, s_j, o_j) = \begin{cases} 0 & \text{if k occluded in } o_j \\ \max_{l_k} \psi_{jk}(l_j, l_k, s_j, s_j) + \phi_k(l_k, s_j, o_j | I) \end{cases}$$

$$\mu_{k \to j}(l_j, s_j, o_j) = \nu_{jk}(l_k, s_j, o_j) + b_{jk}(s_j, o_j, s_j, o_j)$$

Where we have used the notation to explicitly capture the constraint that landmark shape and occlusion mixtures (s_k, o_k) must match those of the parent part (s_j, o_j). In our models, this reduces the memory and inference time by roughly a factor of 2, a savings that becomes increasingly significant as the number of occlusion mixtures grows.

3.3.3 Global Mixtures for Viewpoint and Resolution

Viewpoint and image resolution are the largest sources of variability in the appearance and relative location of landmarks. To capture this, we use a mixture over head poses. These "global" mixtures can be represented with the same notation as above by expanding the state-space of the shape variables to be the cross product of the set of local shapes for part i and the global viewpoint for the model (i.e., $s_i \in \mathcal{S}_i \times \mathcal{V}$) and fixing some entries of the bias b_{ij} to be $-\infty$ to prevent mixing of local shapes from different viewpoints. In our implementation we tie parameters to enforce the left- and right-facing models to be mirror symmetric.

The HPM model we have described includes a large number of landmarks. While this is appropriate for high resolution imagery, it does not perform well in detecting and modeling low resolution faces (< 150 pixels tall). To address this we introduce an additional global mixture component for each viewpoint that corresponds to low-resolution HPM model consisting of a single half-resolution template for each part and no landmark templates. This mixture is trained jointly with the full resolution model using the strategy described in [79].

3.4 Model Training and Inference

The potentials in our shape model are linearly parameterized, allowing efficient training using an SVM solver [39]. Face viewpoint, landmark locations, shape and occlusion mixtures are completely specified by pre-clustering the training data so that parameter learning is fully supervised. We first describe how these supervised labels are derived from training data and how we synthesize "virtual" positive training examples that include additional occlusion. We then discuss the details of the parameter learning and test-time prediction.

3.4.1 Training Data

We assume that a training data set of face images has been annotated with landmark locations for each face. From such data we automatically generate additional mixture labels specifying viewpoint, shape, and occlusion. We also generate additional virtual training examples by synthesizing plausible coherent occlusion patterns.

Viewpoint and Resolution Mixtures To cluster training examples into a set of discrete viewpoints, we make use of the MultiPIE dataset [52] which provides ground-truth viewpoint annotations for a limited set of faces. We perform Procrustes alignment between each training example and examples in the MultiPIE database and then transfer the viewpoint label from nearest MultiPIE example to the training example. In our experiments we used either 3 or 7 viewpoint clusters (each viewpoint spans 15 degrees). In addition to viewpoint, alignment to MultiPIE also provides a standard scale normalization and removes in-plane rotations from the training set. To train the low-resolution mixture components, we use the same training data but down-sample the input image by a factor of 2.

Part Shape and Occlusion Mixtures For each part and each viewpoint, we cluster the set of landmark configurations in the training data in order to come up with a small number of shape mixtures for that part. The part shapes in the final model are represented by displacements relative to a parent node so we subtract off the centroid of the part landmarks from each training example prior to clustering. The vectors containing the coordinates of the centered landmarks are clustered using k-means. We imagine it would be efficient to allocate more mixtures to parts and viewpoints that show greater variation in shape, but in the final model tested here we use fixed allocation of $S = 3$ shape mixtures per part per viewpoint. Fig. 3.4 shows example clusterings of part shapes for the center view.

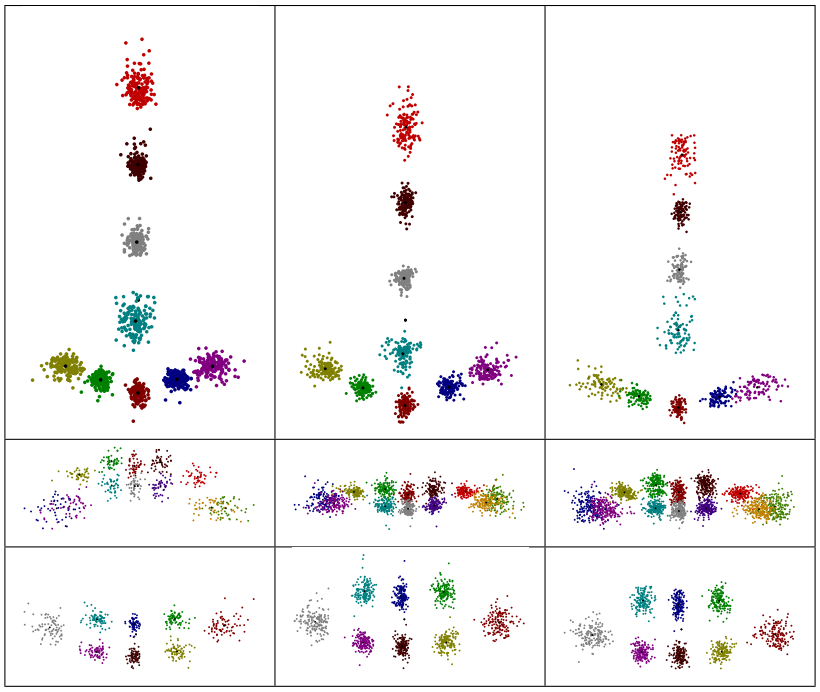


Figure 3.4: Example shape clusters for face parts (nose, upper lip, lower lip). Co-occurrence biases for combinations of part shapes are learned automatically from training data. Different colored points correspond to location of each landmark relative to the part (centroid).

Synthetic Occlusion Patterns In the model each landmark is fully occluded or fully visible. The occlusion state of a part describes the occlusion of its constituent landmarks. If there are N_l landmarks then there are 2^{N_l} possible occlusion patterns. However, many of these occlusions are quite unlikely (e.g., every other landmark occluded) since occlusion is typically generated by an occluder object with a regular, compact shape.

To model spatial coherence among the landmark occlusions, we synthetically generate "valid" occlusions patterns by first sampling mean part and landmark locations from the model and then randomly sampling a quarter-plane shaped occluder and setting as occluded those landmarks that fall behind the occluder. Let a, b be uniformly sampled from a tight box surrounding the face. This selected origin point induces a partition of the image into quadrants (i.e., $(x < a) \wedge (y < b)$, $(x \geq a) \wedge (y < b)$, etc.). We choose a quadrant at random and mark all landmarks falling in that landmark as occluded. While our occluder is somewhat

"boring", it is straightforward to incorporate more interesting shapes, e.g., by sampling from a database of segmented objects. Fig. 3.3 shows example occlusions generated for a training example.

In our experiments we generate 8 synthetically occluded examples for each original training example. For each part in the model we cluster the set of resulting binary vectors in order to generate a list of valid part occlusion patterns. The occlusion state for each landmark in a training example is then set to be consistent with the assigned part occlusion pattern. In our experiments we utilized only $O = 4$ occlusion mixtures per part, typically corresponding to unoccluded, fully occluded and two half occluded states whose structure depended on the part shape and location within the face.

Figure 3.5: Examples of landmark localization and occlusion estimation for images from the HELEN (row 1) and COFW (rows 2-3) test datasets. Red indicates those landmarks which are predicted as being occluded by the HPM.

3.4.2 Parameter learning

Recall that our model (Eqn. 3.1) is parameterized by a set of weights and biases, which we collect into a parameter vector w. Each weight is multiplied by some corresponding feature that depends on the hypothesized model configuration (l, s, o) and input image I. Collecting these features a feature vector $\Psi(l, s, o|I)$, we write the scoring function as an inner product with the model weights $Q(l, s, o) = w \cdot \Psi(l, s, o|I)$. We learn the model weights using a regularized SVM objective:

$$\begin{aligned}
&\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{t} \eta_t \\
&\quad w \cdot \Psi(l^t, s^t, o^t|I^t) \geq 1 - \eta_t && \forall t \in \mathcal{P} \\
&\quad w \cdot \Psi(l, s, o|I^t) \leq -(1 - m\delta(o) - \eta_t) && \forall l, s, o \;\; \forall t \notin \mathcal{P}
\end{aligned}$$

where (l^t, s^t, o^t) denotes the supervised model configuration for a positive training example, $\delta(o)$ is a margin scaling function that measures the fraction of occluded landmarks and C and m are hyper-parameters (described below). The constraint on positive images $t \in \mathcal{P}$ encourages that the score of the correct model configuration be larger than 1 and penalizes violations using slack variable η_t. The second constraint encourages the score to be low on all negative training images $t \notin \mathcal{P}$ for all configurations of the latent variables.

Margin scaling for occlusion This formulation differs from standard supervised DPM training in the treatment of negative training examples. Since landmarks can be occluded in our model, fully or partially occluded faces can be detected by our model in the negative training images. These images do not contain any faces and we would like our model generates low scores for these detections. However, a landmark which is detected as occluded in a negative image is in some sense correct. *There is no real distinction between a negative image and a positive image of a fully occluded face!* Thus we penalize negative detections

(false positives) with significant amounts of occlusion less than fully-visible false positives.

For this purpose, we scale the margin for negative examples in proportion to the number of occluded landmarks. We specify the margin for a negative example as $1 - m\delta(o)$, where the function $\delta(o)$ measures the fraction of occluded landmarks and m is a hyper-parameter. As the number of occluded landmarks increases the margin decreases and the model score for that example can be larger without violating the constraint. The margin for a fully occluded example is equal to $1 - m$. Setting $m = 0$ corresponds to standard classification where all the negatives have the same margin of 1. In this case the biases learned for occluded landmarks tend to be low (otherwise many fully or partially occluded negative examples will violate the constraint). As a result, models trained with $m = 0$ tend not to predict occlusion. As we increase m, the scores of fully or partially occluded negative examples can be larger without violating the constraint and the training procedure is thus free to learn larger bias parameters associated with occluded landmarks. As we show in our experimental evaluation, this results in higher recall of occluded landmarks and improved test-time performance.

We use a standard hard-negative mining or cutting-plane approach to find a small set of active constraints for each negative image. Given a current estimate of the model parameters w, we find the model configuration (s, l, o) that maximizes $w \cdot \Psi(l, s, o|I^t) - m\delta(o)$ on a negative window I^t. Since the loss $m\delta(o)$ can be decomposed over individual landmarks, this loss-augmented inference can be easily performed using the same inference procedure introduced in section 3.3. We simply subtract $\frac{m}{N_l}$ from the messages sent by occluded landmarks where $N_l = 68$ is the number of landmarks. During training we make multiple passes through the negative training set and maintain a pool of hard negatives for each image. We share the slack variable η_t for all such negatives found over a single window I^t.

3.4.3 Test-time Prediction

Scale and In-plane Rotation We use a standard sliding window approach to search over a range of locations and scales in each test image. In our experiments, we observed that part models with standard quadratic spring costs are surprisingly sensitive to in-plane rotation. Models that performed well on images with controlled acquisition (such as MultiPIE) fared poorly "in the wild" when faces were tilted. The alignment procedure described above explicitly removes scale and in-plane rotations from the set of training examples. At test time, we perform an explicit search over in-plane rotations (-30 to 30 degrees with an increment of 6 degrees).

Landmark Prediction The number of landmarks in our model was chosen based on the availability of 68-landmark ground-truth annotations. In cases where it was useful to benchmark landmark localization of our model on datasets using different landmark annotation standards (e.g., COFW 29-landmark data), we used additional held-out training data to fit a simple linear map from the part locations returned by our hierarchical part model to the desired output space. This provided a more stable procedure than simpler heuristics such as hand selecting a subset of landmarks.

Let $l^i \in \mathbb{R}^{2N_l}$ be the vector of landmark locations returned at the top scoring detection when running the model on a training example i. Let $\hat{l}^i \in \mathbb{R}^{2M}$ a vector of ground-truth landmark locations for that image based on some other annotation standard (i.e., $M \neq N_l$). We train a linear regressor

$$\min_{\beta} \sum_{i} \|\hat{l}^i - \beta^T l^i\|^2 + \lambda \|\beta\|^2$$

where $\beta \in \mathbb{R}^{2N_l \times 2M}$ is the matrix of learned coefficients and λ is a regularization parameter. To prevent overfitting, we restrict β_{pq} to be zero unless the landmark p belongs to the same

part as q.

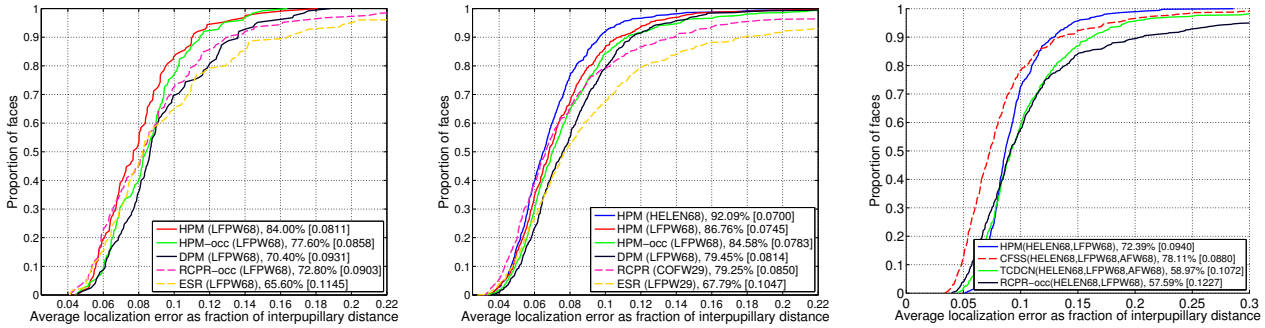
To predict landmark occlusion, we carried out a similar mapping procedure using regularized logistic regression. However, in this case we found that simply specifying a fixed correspondence between the two sets of landmarks based on their average locations and transferring the occlusion flag from the model to benchmark landmark space achieved the same accuracy.

3.5 Experimental Evaluation

Figure 4.6 shows example outputs of the HPM model run on example face images. The model produces both a detection score and estimates of landmark locations and occlusion states. While the possible occlusion patterns are quite limited (4 occlusions patterns per part shape), the final predicted occlusions (marked in red) are quite satisfying in highlighting the support of the occluder for many instances. We evaluate the performance of the model on three different tasks: landmark localization, landmark occlusion prediction, and face detection. In our experiments we focus on test datasets that have significant amounts of occlusion and emphasize the ability of the model to generalize well across datasets.

3.5.1 Landmark Localization

Datasets We evaluate performance of our method and related baselines on three benchmark datasets for landmark localization: the challenging portion of the IBUG dataset which contains a range of poses and expressions [78], a subset of the HELEN dataset [66] containing occlusions, and the Caltech Occluded faces in the Wild (COFW) [12] dataset. We evaluate on IBUG to provide a baseline for localization in the absence of occlusion. The latter two datasets were selected to evaluate the ability of our model in the presence of substantial natural occlusion which is not well represented in many benchmarks. The authors of [12]



(a) Occluded HELEN68 (b) COFW29 (c) COFW68

Figure 3.6: Panels show cumulative error distribution curves (the proportion of test images that have average landmark localization error below a given threshold) on three test sets: an occlusion rich subset of HELEN, COFW29 and COFW68. The legend indicates the training set (in parentheses), the success rate % at a localization threshold of 0.1 and the average error [in brackets]. The HPM shows good localization performance, especially on more difficult datasets with significant occlusion. In general regression models (dashed lines) have better performance for a low localization threshold compare to part based models (solid lines). However, the success rates for regression models increase more slowly and eventually cross over those for part models (solid lines) as the allowable localization error threshold increases.

estimate that COFW contains 23% occluded landmarks. Fig. 4.6 depicts selected results of running our detector on example images from the HELEN and COFW test datasets.

68 Landmark annotations for COFW We note there is a variety of annotation conventions across different face landmark datasets. COFW is annotated with 29 landmarks while HELEN includes a much denser set of 194 landmarks. The 300 Faces in-the-wild Challenge (300-W) [78] has produced several unified benchmarks in which HELEN dataset have been re-annotated with a set of 68 standard landmarks. To allow for a greater range of comparisons and further this standardization, we manually re-annotated the test images from the COFW dataset with 68 landmarks and occlusion flags. We also generated face bounding boxes (using a similar detection method that used for the 300-W datasets [4]) for evaluating pose regression methods that require initialization. We bootstrapped our annotations from the 29-landmark annotations using a custom annotation tool. The annotations and

benchmarking code are publicly available[2].

Localization Evaluation Metrics To evaluate landmark localization independent of detection accuracy, we follow a standard approach that assumes that detection has already been performed and evaluates performance on cropped versions of test images. While our model is capable of both detecting and localizing landmarks, this protocol is necessary to evaluate pose regression methods that require good initialization. We thus follow the standard protocol (see e.g., [78]) of using the bounding boxes provided for each dataset (usually generated from the output of a face detector) by evaluating the localization accuracy for the highest scoring detection that overlaps the given bounding box by at least 70%.

We report the average landmark localization error across each test set as well as the "success rate", the proportion of test images with average landmark localization error below a given threshold. Distances used in both quantities are expressed as a proportion of the interpupillary distance (distance between centers of eyes) specified by the ground-truth. Computing the success rate across a range thresholds yields a cumulative error distribution curve (CED) (Fig. 4.5). When a single summary number is desired, we report the success rate at a standard threshold of 0.1 interpupillary distance (IPD).

Training and baselines To train our model, we used training data from LFPW (811 images) and/or HELEN (2000 images) annotated with 68 landmarks. The training set is specified in parenthesis in figure legends. From each training image we generate 8 synthetically occluded "virtual positives". To fit linear regression coefficients for mapping from the HPM predicted landmark locations to 29 landmark datasets, we ran the trained model on the COFW training data set and fit regression parameters β that mapped from the 68 predicted points to the 29 annotated.

[2] `https://github.com/golnazghiasi/cofw68-benchmark`

Method	average error
DRMF [4]	0.1979
CDM [106]	0.1954
RCPR [12]	0.1726
ESR [14]	0.1700
CFAN [110]	0.1678
SDM [98]	0.1540
CFSS [115]	0.1200‡ / 0.0998
TCDCN [111]	0.1121† / 0.0860
LBF [82]	0.1198
HPM	0.1310

Table 3.1: Average errors as a fraction of IPD on IBUG68 [78] dataset. Results with ‡/† are obtained by testing the method with the standard detector bounding boxes provided by 300-W, using either the published model (†) or retraining (‡).

For diagnostic purposes, we trained several baseline models including a version of our model without occlusion mixtures (HPM-occ) and the (non-hierarchical) deformable part model [3] (DPM) described by [116]. We also evaluated variants of the robust cascaded pose regression (RCPR) described in [12] as well as their implementation of explicit shape regression (ESR) [14] using both pre-trained models provided by the authors and models retrained to predict 68 landmarks. Unlike HPM which uses virtual occlusion, RCPR requires training examples with actual occlusions and corresponding annotations. For training sets that featured no occlusion, we thus trained a variant that does not model occlusion (RCPR-occ).

Localization Results (Occluded HELEN 68) We evaluated on a subset of the HELEN dataset [66] consisting of 126 images which were selected on the basis having some significant amount of occlusion [4]. We do not report results of the HPM (HELEN68) model on this dataset as there was overlap between training and testing images. Fig. 4.5(a) shows the error distribution. The HPM achieves an average error of 0.0811, beating out the DPM baseline (0.0931) and RCPR-occ (0.0903). Removing explicit occlusion from the model (HPM-occ)

[3]The originally published DPM model of [116] was trained on the very constrained MultiPIE dataset [52]. Retraining the model of Zhu et al. and including in-plane rotation search at test time yielded significantly better performance than reported elsewhere (c.f., [12])

[4]`https://github.com/golnazghiasi/Occluded-HELEN-image-list`

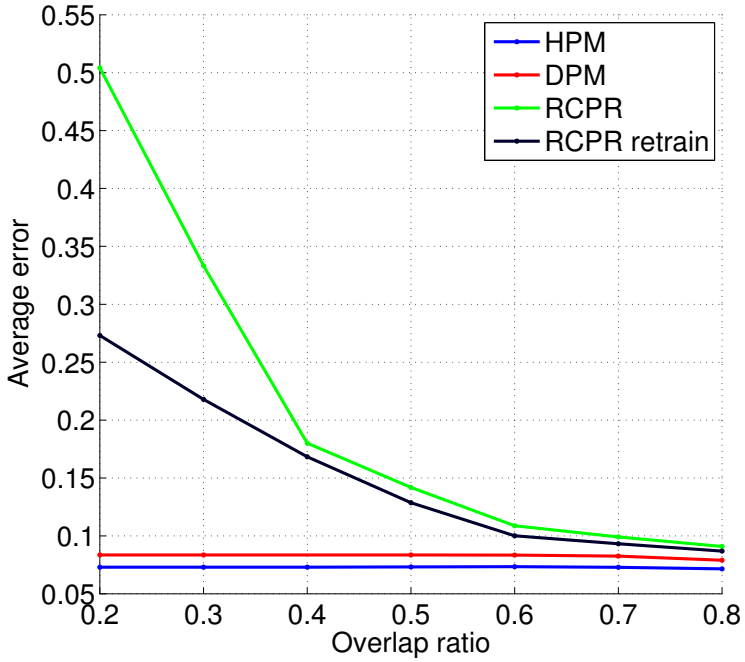


Figure 3.7: We analyze the landmark localization average error of RCPR, HPM and DPM for different overlap ratio with the ground-truth face boxes. For RCPR we change the minimum overlap ratio of the initial bounding boxes and the ground-truth face boxes. For HPM and DPM, we change the minimum overlap threshold of the returned detections and ground-truth boxes. RCPR is very sensitive to the amount of overlap and its performance decreases rapidly as the overlap ratio decreases. But, HPM and DPM are robust to the overlap threshold and they can maintain the same performance over different thresholds.

results in lower success rates for a range of thresholds.

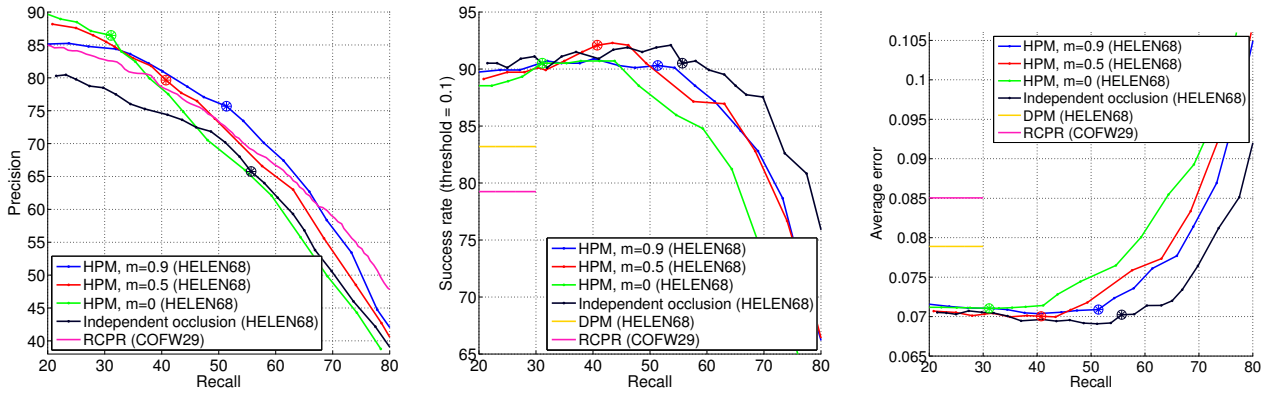
Localization Results (COFW29) To facilitate diagnostic comparison to previously published results, we evaluated our model on the original COFW 29-landmark test set [12] consisting of 507 internet photos depicting a wide variety of more difficult poses and includes a significant amount of occlusion. Since COFW training only contains 29 landmarks (we only performed additional annotations on test data), we evaluated models trained on LFPW68 and HELEN68. Fig. 4.5(c) shows that HPM achieves a significantly lower average error than RCPR and higher success rates for all but the smallest (< 0.06) localization success thresholds.

		LFPW (29)		COFW (29)	
model	training dataset	SR	AE	SR	AE
RCPR-occ	LFPW29	88.95	0.073	63.44	0.115
RCPR-occ	LFPW29+	98.95	0.038	63.64	0.096
RCPR-occ	COFW29	89.01	0.071	76.28	0.091
RCPR	COFW29	91.05	0.064	79.25	0.085
HPM	LFPW68,INR-	97.37	0.050	86.76	0.075
HPM	HELEN68,INR-	98.42	0.049	90.71	0.072
HPM	HELEN68,PAS-	98.95	0.048	92.09	0.070

Table 3.2: We find HPM generalizes well across datasets while pose regression has a strong dependence on training data. Localization performance is measured by success rate (SR) and average error (AE). The RCPR model trained on COFW performs much better on COFW test data compared to RCPR-occ trained on LFPW29+ (79% SR vs 64% SR) but has much worse performance on LFPW test data compared to that model (91% SR vs 99% SR). Good performance on LFPW also depends heavily on including additional warped positive instances (LFPW29+ vs LFPW29). The HPM trained on LFPW68 has high success rates on both COFW (87%) and LFPW (%97) test data. Last two rows of the table show the performance of HPM when a different training data set (HELEN68) is used for training. This dataset has more variation and more images (1758) compared to LFPW68 (682) and improves performance of HPM on both test datasets. Training on more negative images (6000 images from PASCAL) decreases localization error of our model compared to using only INRIA negatives.

Localization Results (COFW68) We tested our model trained on LPFW68 and HELEN68 training data on this benchmark and compared with CFSS, TCDCN and RCPR-occ (Fig. 4.5 (c)). For CFSS and TCDCN we used the publicly available pre-trained models which were trained on HELEN68, LFPW68 and AFW68 (TCDCN is also pretrained on MAFL dataset). For RCPR-occ we used the authors' code to train a model on HELEN68 and LFPW68 training sets. Note we that couldn't train the full RCPR 68-landmark model with occlusion since HELEN68 and LFPW68 do not have occlusion and COFW train is only labeled with 29 landmarks.

Localization Results (IBUG 68) This dataset contains 68 landmark annotations for 135 faces in difficult poses and expression [78]. For testing our method on this dataset, we follow previous work and trained our model on combined HELEN68 and LFPW68 training



(a) Occlusion prediction accuracy

(b) Success rate vs. occlusion recall

(c) Localization error vs. occlusions recall

Figure 3.8: Occlusion prediction accuracy on the COFW test dataset for variants of our model. Using a suitable margin scaling function (see Sec. 3.4.2) allows for significantly better occlusion prediction accuracy over an independent occlusion model (a) with minimal loss in localization performance (b,c). Localization performance of DPM and RCPR are included for reference.

data provided by 300-W. Since IBUG includes many side view faces we trained a variant of our model with 7 viewpoints. We compare our model with published performance of several state-of-the-art methods in Table 3.1 and achieve comparable performance.

In addition to reporting values from the published literature, we also re-evaluated two recent top-performing models: TCDN [111] and CFSS [115]. Since these methods operate in the general framework of pose regression, performing iterative refinement of predicted landmark locations, they are sensitive to initial bounding box location. We tested both models using the standardized detection bounding boxes provided by the 300-W benchmark [78] rather than tight cropping images to the ground-truth landmark locations. We used the pre-trained TCDCN model available online while for CFSS we retrained the model using the standard detector bounding boxes. In both cases, average error was significantly worse than previously reported results, highlighting the sensitivity of these methods to initialization.

Dependence of Localization on Detection A key benefit of the HPM (and DPM [116]) approach is that the same model serves to both detect and localize the landmarks. In

contrast, pose regression methods such as RCPR, TCDN or CFSS require that the face already be detected. This distinction becomes particularly important for occluded faces since detection is significantly less accurate (see Detection experiments below).

To highlight the dependence of landmark localization on accurate detection, we benchmarked average localization error for varying degrees of overlap between the hypothesized detection and ground-truth bounding box on the COFW test set. As shown in Fig. 3.7, decreasing the overlap ratio has no affect HPM / DPM performance since there are never false positives in the vicinity of the face that score higher than one with high overlap ratio. In contrast, RCPR performs significantly worse when initialized from bounding boxes that do not have high overlap with the face. Since the area over which RCPR searches is learned from training data, we also retrained a version of RCPR for each degrees of overlap. This yielded improved performance but still shows a significant fall off in performance compared to the HPM. As noted above, we encountered similar behavior when evaluating other methods such as TCDNN and CFSS on realistic detector-generated bounding boxes.

Dependence of Localization on Training data One advantage of the HPM model is robustness to the choice of training data set. Table 3.2 highlights a comparison of HPM and RCPR in which the training set is varied. HPM performs well on LFPW and COFW regardless of training set specifics. In contrast, RCPR shows better performance on COFW when the training data is also taken from COFW. Training data augmentation is also important to achieve good performance with RCPR, while HPM works well even when trained on the relatively smaller LFPW training set.

3.5.2 Occlusion Prediction

To evaluate the ability of the model to correctly determine which landmarks are occluded, we evaluate the accuracy of occlusion as a binary prediction task. For a given test set, we compute precision and recall of occlusion predictions relative to the ground-truth occlusion labels of the landmarks.

For HPM, we trace out a precision-recall curve for occlusion prediction by adjusting the model parameters to induce different predicted occlusions. As described in Section 3.3, the bias parameter $b_{ij}(s_i, s_j, o_i, o_j)$ favors particular co-occurrences of part types. By increasing (decreasing) the bias for occluded configurations we can encourage (discourage) the model to use those configurations on test. Let $b_{ij}(s_i, s_j, o_i, o_j)$ be a learned bias parameter between an occluded leaf and its parent. To make the model favor occluded parts, we modify this parameter to $b_{ij}(s_i, s_j, o_i, o_j) + abs(b_{ij}(s_i, s_j, o_i, o_j)) \times \alpha$.

Fig. 3.8(a) depicts occlusion precision-recall curves generated by running the HPM model for different bias α offsets. The crosses mark the precision-recall for the default operating point when $\alpha = 0$. We compare performance of the HPM model with different values of the margin scaling hyper-parameter m as well as RCPR and a baseline independent occlusion model. Fig. 3.8 (b) and (c) show the corresponding average errors and success rates for these models parameterized by the recall of occlusion. For large values of α, the model predicts more occlusions, resulting in improved recall at the expense of precision (a) and ultimately lower localization accuracy (b,c).

Margin scaling As described in section 3.4.2, we can change the learning parameter m to produce models with different recall of occlusions at the trained operating point ($\alpha = 0$). When $m = 0$ all the negative examples including fully or partially occluded configurations are penalized equally. Therefore, model learns small biases for occluded configurations, reducing

the total loss over occluded negative examples and decreasing default recall of occlusion. When driven to predict more occlusion by increasing α the model localization performance degrades rapidly. Training the model with larger values of m yields a model which naturally predicts occlusion more frequently and degrades more gracefully for larger values of α. We found that choosing a value of $m = 0.5$ provided a good compromise, improving both recall and localization accuracy.

Independent occlusion baseline We compared the results of HPM with a model that had the same architecture but in which there are no occlusion mixtures at the part level and each landmark is allowed to be independently set to visible or occluded depending on learned biases. We refer to this as "independent occlusion" since the model does not capture any correlations between the occlusion of different landmarks. We found that this independent occlusion model has many of the same benefits as the HPM model in terms of landmark localization accuracy (Fig. 3.8). However, occlusion prediction accuracy is significantly worse in the independent model with precisions typically 5% lower than HPM($m = 0.9$) over a range of recall values.

3.5.3 Detection

Pose regression requires good initialization provided by a face detector to accurately locate landmarks. In contrast, part-based models have the elegant advantage of performing detection and localization simultaneously. In this section, we compare the detection performance of our approach and other top methods on two datasets: FDDB [60] and our own Occluded Face Detection (UCI-OFD) dataset.

Multi-resolution HPM Since many face detection datasets such as FDDB contain many low-resolution faces, we trained a multi-resolution variant of our model [79]. This model

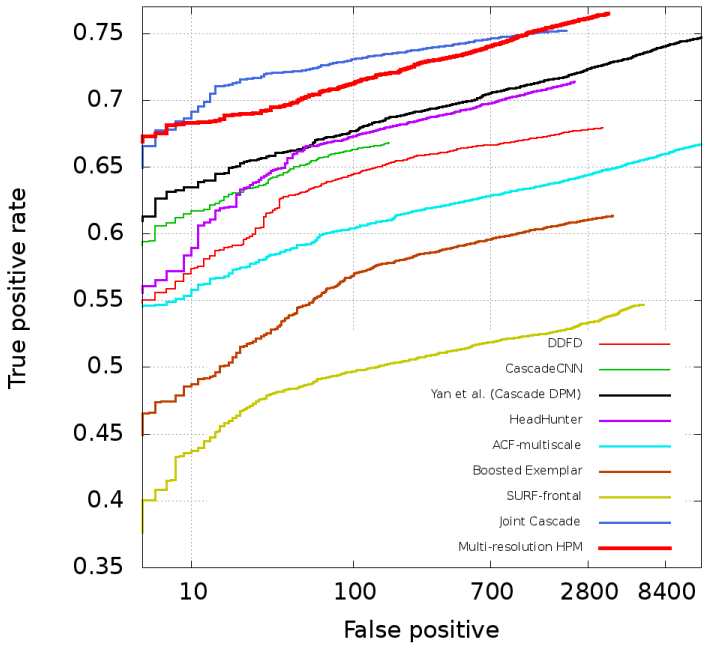
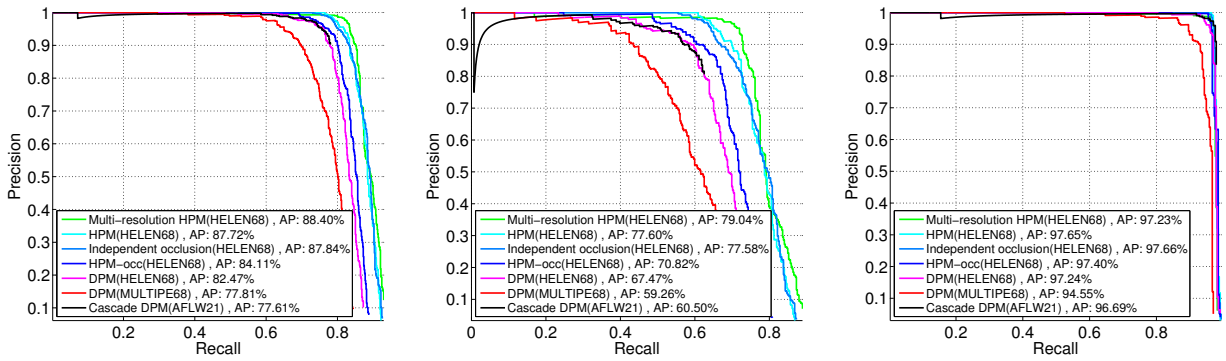


Figure 3.9: Face detection performance of HPM and state-of-the-art methods [1] on the continuous-ROC FDDB benchmark [60].

has a high and a low-resolution model for each viewpoint. The high resolution model has the same structure as our trained model for landmark localization except that parts are represented as 3x3 HoG cells rather than 5x5. The low-resolution model has 7 parts (right eye, left eye, nose, mouth, chin, left jaw and right jaw) each of which is represented by 7x7 HoG cells with the spatial bin size of 4. Each part has one shape mixture and 2 occlusion mixtures (visible or occluded). The heights (eyebrow to chin) of the large model and small model are about 100 and 60 pixels respectively. To detect even smaller images, we upsample input images by a factor of 2 to allow for detection of faces as small as 30 pixels. We trained this model using the same 1758 positive examples from HELEN68 and generated 8 virtual positive examples per example. For negative images we used 6000 images from the PASCAL VOC 2010 train-val set which do not contain people.



(a) UCI-OFD (b) UCI-OFD occluded (c) UCI-OFD visible

Figure 3.10: Precision-Recall curves of face detection on our UCI-OFD dataset (a) for all of the faces, (b) occluded subset and (c) visible subset. On the visible subset our model, DPM retrained on HELEN68 and Cascade DPM [99] have almost similar performances, but our model significantly outperforms these methods on the occluded subset and it has a better overall performance. Cascade DPM uses many accelerate techniques, which may reject some of the faces. Its maximum recall for the visible faces is near 100%, while its maximum recall for the occluded faces is only 60%. The initial drop in the Precision-Recall of this method for the occluded subset is because its returned bounding boxes for some of the high scored occluded faces are not accurate and do not have the minimum 0.5 overlap with the ground-truth bounding boxes.

Detection on FDDB We evaluated our multi-resolution model on the widely used FDDB dataset. This dataset contains 5171 faces in a set of 2845 images. Faces are annotated by ellipses in this dataset and are as small as 20 pixels in height. To match that, we map our predicted landmark locations to ellipses using a linear regression model. FDDB has 10 folds and the ROC curves are the average over the results of these folds. To compute ellipses for each fold, we learned the linear regression coefficients using examples from the other 9 folds.

We used the standard evaluation protocol for this dataset and compared our method with the top published results available on the FDDB website [1]. The continuous ROC curve for our method and leading methods are shown in Fig. 3.9 plotted on a semi-log scale. Our result is highly competitive with the top results. The model has better performance on the continuous ROC evaluation relative to other methods because it can predict location of parts and compute accurate bounding ellipses around the faces.

UCI Occluded Face Detection Dataset (UCI-OFD) In order to better measure the ability of our model to handle detection of occluded faces, we assembled a preliminary dataset for occluded face detection. This dataset and benchmarking code are publicly available [5]. It consists of 61 images from Flickr containing 766 labeled faces. Of the faces in these images, 430 include some amount of occlusion. Most of the faces are near frontal and vertical. Height (eyebrow to chin) of the smallest face is about 40 pixels.

Precision/Recall curves of face detection of multi-resolution HPM, HPM, HPM-occ, DPM and Cascade DPM [99] are shown in Fig. 3.10(a). We further break down performance, plotting Precision/Recall curves for the subset of faces with some amount of occlusion in (b) and fully visible in (c). Precision and recall for occluded subset of faces are calculated as below:

$$\text{Precision}_o = \frac{tp_o}{tp_o + fp}, \text{Recall}_o = \frac{tp_o}{tp_o + fn_o}$$

where tp_o and fn_o show number of correct detection and miss detection of occluded faces, respectively. Our method significantly outperforms other methods on the occluded subset and the performance of all of the methods are almost equal on the visible subset. Fig. 3.11 shows example detection results produced by the model on cluttered scenes containing many overlapping faces.

3.6 Conclusion

Our experimental results demonstrate that adding coherent occlusion and hierarchical structure allows for substantial gains in performance for landmark localization and detection in part models. In images with relatively little occlusion, the HPM gives similar detection and

[5] `https://github.com/golnazghiasi/hpm-detection-code/tree/master/UCI_OFD`

localization performance to other part-based approaches, e.g. DPM, but is significantly more robust to occlusion. Our results also suggest that when it is useful to determine exactly which parts are occluded (e.g., for later use in face identification), independent occlusion makes weaker predictions than our part occlusion mixtures which enforce coherence between neighboring landmarks. While not specifically trained for landmark estimation, the final HPM is competitive with pose regression techniques in terms of landmark localization accuracy on unoccluded faces (IBUG) and outperforms many such methods on occluded faces (Occluded HELEN, COFW).

In comparing pose regression and part-based models, there seem to be several interesting trade-offs. In our experiments, we see a general trend in which error distribution curves for pose regression and part-based models cross, suggesting that pose regression yields very accurate localization for a subset of images relative to the HPM but fails for some other proportion even at very large error thresholds. Unlike pose regression, the part model performs detection, eliminating the need for detection as a pre-process and improving robustness. In particular, we are able to detect many heavily occluded faces which would not be detected by a standard cascade detector and hence inaccessible to pose regression. We find that the HPM tends to generalize well across datasets suggesting it avoids some overfitting problems present in pose regression.

This flexibility currently comes with a computational cost. The run-time of our model implementation built on dynamic programming lags significantly behind those of regression-based, feed-forward approaches. Our implementation takes $\sim$10s to run on a typical COFW image, roughly 100x slower than RCPR or DCNN based approaches. However, the HPM is amenable to implementation on a GPU which may address most of this runtime gap.

Finally, we note several avenues for future work. Performance depends on the graphical independence structure of the model which should ideally be learned from data. While our model implicitly represents the pattern of part occlusions, it does not integrate local

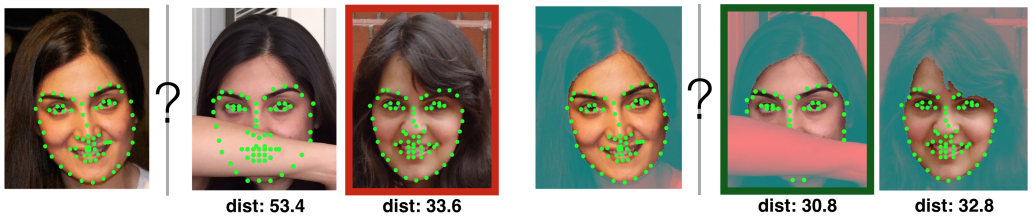
image evidence for the occluder itself. A natural extension would be to add local filters that detect the presence of an occluding contour between the occluded and non-occluded landmarks. Such filters could be shared across parts to avoid increasing too much the overall computation cost while moving closer to our goal of explaining away missing object parts using positive evidence of coherent occlusion. In the next chapter we instead take a slightly different approach that attempts to address the same problem. Rather than utilizing filters to model occlusion boundary appearance locally, we use a bottom up segmentation model to find face/non-face segments. This segmentation is then utilized to find parts configurations whose occlusion pattern is consistent with the location of the occluder.

Figure 3.11: Examples of detection and localization for images from our UCI-OFD dataset (rows 1-2) and images containing occlusion from FDDB dataset (rows 3-4). Detections indicated with only 7 landmarks correspond to responses from the low-resolution model component. Ellipses are predicted on FDDB images by linear regression from landmark locations to ellipse parameters.

Chapter 4

Using Segmentation to Predict the Absence of Occluded Parts

We show in the previous chapters that deformable part models that include explicit states representing part occlusion are more robust in the present of occlusion and yield better part localization. However, the introduced methods do not exploit generic bottom-up cues such as detection of occluding contours which are helpful for detection of occlusion. In this chapter, we propose a part-based face detection model that utilizes bottom-up class-specific segmentation in order to jointly detect and segment out the foreground pixels belonging to the face. The model explicitly represents occlusion of parts at the detection phase, allowing for hypothesized figure-ground segmentation to suggest coherent patterns of part occlusion. We show that this bi-directional interaction between recognition and grouping results in state-of-the-art part localization accuracy for challenging benchmarks with significant occlusion and yields substantial gains in the precision of keypoint occlusion prediction.



(a) (b)

Figure 4.1: (a) Comparing part appearance for all facial keypoints results in incorrect identification of query image. We propose a model that leverages segmentation to identify which keypoints are occluded, allowing downstream facial analysis to focus on visible regions of the face, yielding the correct identity match shown in (b). Numbers below the images indicate the similarity distances between the image and the query image computed as the mean pixel differences over the patches of (a) all the keypoints (b) visible keypoints.

4.1 Introduction

In this chapter similar to the Chapter 3 we focus on the problem of detecting and localizing faces where the parts in consideration are represented by facial keypoints. The introduced part based model in Chapter 3 that include explicit states representing part occlusion can yield better detection and localization accuracy. But a key difficulty with part-based occlusion reasoning is that such methods rely heavily on local image appearance in the vicinity of a part to predict whether it is occluded. This is easily confounded by lighting or other appearance variations and ignores long-range dependencies in patterns of occlusion (e.g., the occluding object comprises an extended region of the image demarcated by enclosing contours). For example, in Fig. 4.1, facial keypoints which are occluded have similar local appearance to the face (they are both skin-colored). We argue that bottom-up segmentation provides a valuable mechanism by which subsets of occluded or visible keypoints can be grouped in a way that is not easily captured by standard pose-regression or deformable part models.

We formulate a joint objective that simultaneously attempts to localize parts and determine their occlusion state in a manner that is consistent with image segments suggested by edges in the image. There is a large body of work on combining detection and segmenta-

tion. Highlights include early work on generating consistent object masks during detection [8], producing semantic segmentations of scenes driven by object detector responses (e.g., [43, 103]) and most recently the use of bottom-up segments as proposals for scoring object detectors (e.g., [50, 42]). Our approach is most closely related to GRABCUT [83] and more recent works such as [33, 42, 17] that enforce mutual consistency between detection and segmentation.

The contribution of this chapter is in combining explicit part occlusion in a detection model with object-specific segmentation using a simple alternating minimization. Unlike previous work that focused primarily on segmenting objects from background, our model solves the problem of identifying occluders with high accuracy. We do not perform inference over a large pool of segmentation proposals (unlike [42, 33]), instead generating a consistent segmentation with only two iterations of segmentation and detection, even when the occluder has similar texture and color to the object. These claims are supported by tests on standard benchmarks showing this approach achieves far more precise occlusion prediction at high recall while maintaining precise part localization.

4.2 A Segmentation Aware Part Model

Figure 4.2 gives an overview of our model which carries out an initial detection followed by alternating segmentation and refinement of detector keypoint locations. These two models are coupled by a unary potential function that enforces agreement between the location and occlusion state of keypoints in the detector and face/non-face assignment of superpixels in the segmentation model. We first describe the detection model assuming a segmentation is given (Sec. 4.2.1-4.2.2) and then return to the problem of inferring segmentations in Sec. 4.2.3.

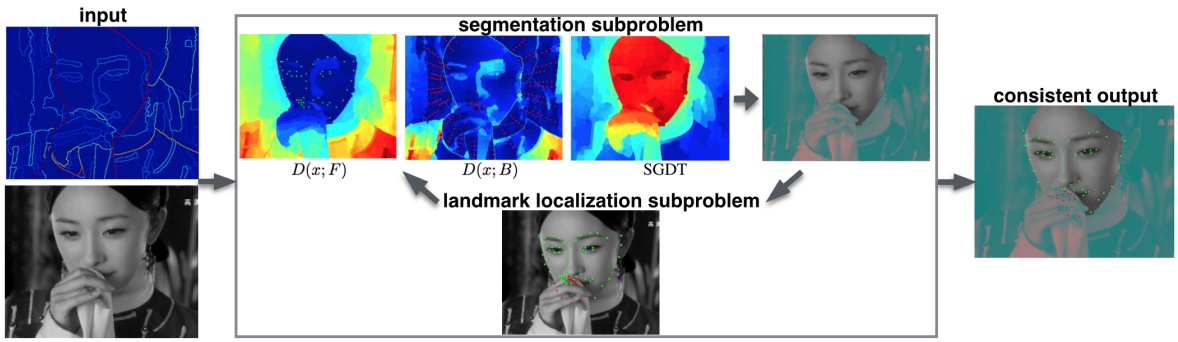


Figure 4.2: Overall landmark localization and face mask prediction pipeline. First column: input image and a computed superpixel image and boundary weights between them. Our method alternates between landmark localization subproblem and segmentation subproblem.

4.2.1 Landmark Localization Subproblem

We use a deformable part model framework to capture the relative layout of facial keypoints. We use the same tree topology used in the hierarchical part model (HPM) described in previous chapter ([44]). Parts of the face (e.g. nose, eyes, eyebrows) are connected to each other using a tree structure and each part is in turn composed of a set of keypoints connected with a star topology. Each part takes on one of a discrete set of shape states (e.g. corresponding to different facial expressions). Unlike the HPM work we do not restrict the possible occlusion patterns for the parts. Each keypoint can be visible or occluded independent of the other keypoints, allowing us to represent a much larger space of possible occlusion patterns. Instead we make use of bottom-up segmentation to guide the detector towards consistent patterns of occlusion. Fig. 4.3 shows some templates of our model corresponding to different choices of part mixtures.

Let l, s, o denote the locations, shape and occlusion states of the parts and keypoints and Z denote a binary segmentation on the image I into face/non-face pixels. We score part

configuration (l, s, o) given image I and segmentation Z as:

$$S(l,s,o|I,Z) = \sum_i \alpha_i^{s_i} \cdot \phi_{\text{App}}(l_i, o_i, I) + \sum_i \sum_{j \in child(i)} \beta_{ij}^{s_i,s_j} \cdot \phi_{\text{Shape}}(l_i - l_j) + b_{ij}^{s_i,s_j,o_i,o_j} \\ + \sum_i \gamma_i^{o_i} \cdot \phi_{\text{Seg}}(l_i, Z) + b_{seg}[Z = \emptyset] \tag{4.1}$$

where α, β, γ and b are the model parameters to be learned. First term scores local appearance of each part with ϕ_{App} denoting a set of appearance features (e.g. HOG) at location l_i. If a keypoint is occluded we set the appearance feature to $\bar{0}$. $\phi_{\text{Shape}}(l_i - l_j)$ contains linear and quadratic expansions of the displacement $l_i - l_j$. This allows the second term to compute a quadratic deformation penalty for locations l_j of the child j given its parent i. b_{ij} is the co-occurrence biases for each combination of occlusion and shape mixtures of parts i and child j.

The third term ϕ_{Seg} scores the consistency of the part locations and their occlusion states with an underlying segmentation Z. The feature $\phi_{\text{Seg}}(l_i, Z)$ is is a subwindow extracted from the segmentation Z centered at location l_i and $\gamma_i^{o_i}$ is a foreground/background probability template for the keypoint i when it is in occlusion state o_i. If a segmentation mask is not available (e.g., for unsegmented training images or the initial detection pass at test-time) we include an additional bias b_{seg} that is added to the score when Z is empty and define $\phi_{\text{Seg}}(l_i, \emptyset) = 0$.

4.2.2 Part Model Parameter Learning and Inference

Conditioned on the segmentation Z our scoring function is tree-structured allowing efficient inference using dynamic programming and distance transforms [39]. The potentials in our model are linearly parameterized so we can write our scoring function as an inner product

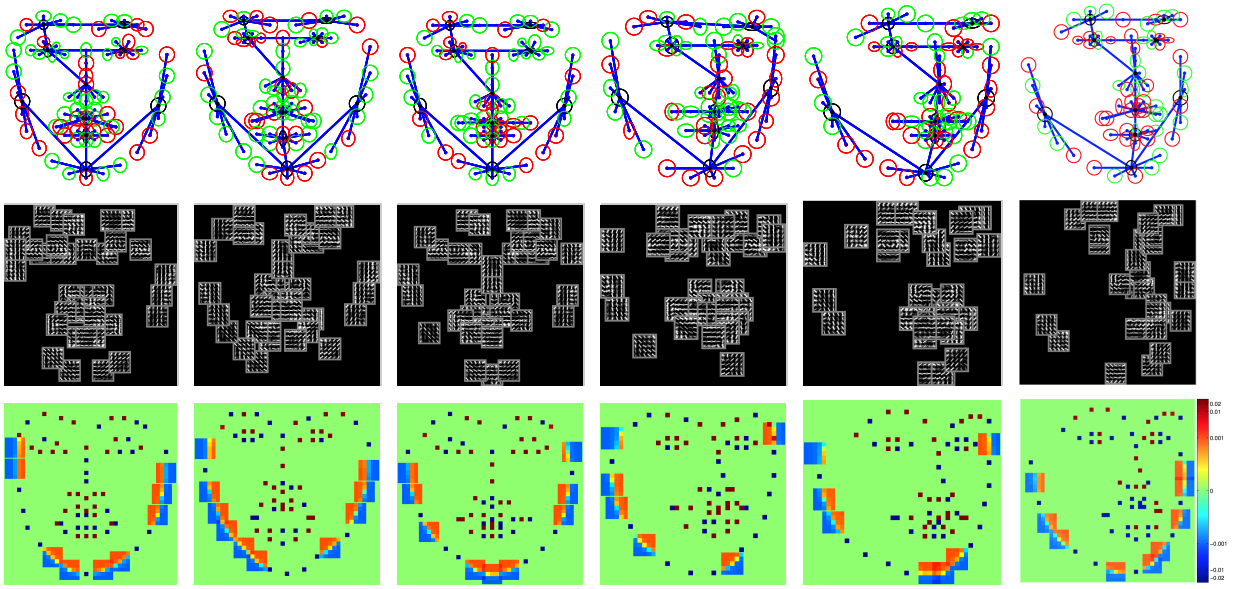


Figure 4.3: Our model has similar structure as HPM [44]. But, unlike HPM we do not restrict the possible occlusion patterns and each keypoint can be visible (green) or occluded (red) independent of the other keypoints. The examples here show templates corresponding to different choices of part mixtures. The appearance of visible keypoints are modeled with HOG templates (2nd row). Each keypoint of our model has a foreground/background probability mask (3rd row). The segmentation masks for the visible keypoints on the jaw are 5×5, but all the other masks are 1×1.

of weights and features $S(l, s, o, I, Z) = w \cdot \phi(l, s, o|I, Z)$. To efficiently learn the model's parameters, we solve a regularized SVM with a set of constraints over scores of positive and negative examples [39]. The constraints state that positive examples should score better than 1, while negative examples should score less than -1. We scale the margin for negative examples in proportion to the number of occluded keypoints [47]. Which means we penalize negative detections (false positives) with significant amounts of occlusion less than fully visible false positives. This helps us to learn the model's parameters for a high recall of occlusion.

Positive training data We train the model using positive training examples in which all the variables (l, s, o) are fully observed. To learn shape mixtures, we use a similar strategy to the model described in the previous chapter ([44]) and cluster the set of keypoint configurations of each part over the training set to come up with a small number of shape

mixture labels for each part. Since our training dataset does not include occluded faces, we synthesize "virtual" positive training examples with occlusions by starting from a training example and setting a subset of the keypoints in the occluded state. The occlusion state for the part nodes (parents of the keypoints) is not used.

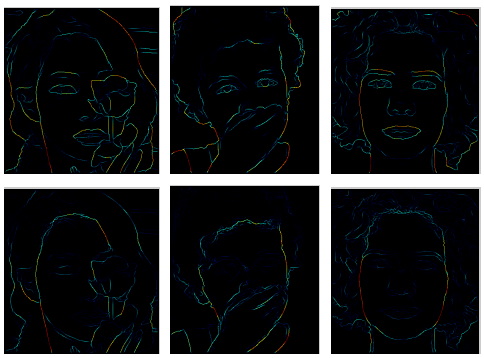
We would like our model to be able to detect faces with and without segmentation of the image. For each positive training example we generate two types of feature vectors: one with and one without segmentation features. The first set of training data does not include segmentation features and assigns 1 for the corresponding feature of b_{seg}. Hence, the feature vector is $[\phi^T_{\text{App}}, \phi^T_{\text{Def}}, \phi^T_{\text{bias}}, 1, 0]^T$. The second type of training data include segmentation features and the feature vector is $[\phi^T_{\text{App}}, \phi^T_{\text{Def}}, \phi^T_{\text{bias}}, 0, \phi^T_{\text{Seg}}]^T$. We learn all the weights jointly so that the appearance, deformation and biases weights are over the two types of training data.

Mining hard negatives We use images that do not contain any faces as the negative images. We would like our model to compute a low score for all possible locations and mixtures of the parts and also for all segmentations of the image into foreground and background. Our part localization model can find the optimal locations and mixtures of the parts, but it does not generate a segmentation. Because the space of possible segmentations is very large, we can not run our part based model for every possible segmentation. Also, we should run our segmentation procedure on many negative images for various locations and scales, so we need an efficient technique for finding such segmentations.

Rather than considering all possible segmentations, we generate two different sets of candidate negatives. First, we use the ability of our model to find faces without providing any segmentation to generate higher scoring negatives training examples on a pool of non-face images. Given the keypoint locations and visibility flags, we greedily assign superpixels of the image to the foreground or background to generate the most consistent segmentation

	ODS	AP
Generic BD (BSDS)[32]	0.288	0.160
Face BD (gray COFW)	0.336	0.242
Face BD (gray COFW), seg feature	0.351	0.268
Face BD (rgb COFW)	0.358	0.272
Face BD (rgb COFW), seg feature	0.379	0.300

(a)



(b)

Figure 4.4: (a) Boundary detection results on the COFW test data where ground truth boundaries just include boundaries around the faces. As a result, generic boundary detector that detects all kinds of boundaries has worse performance in compare to face boundary detector (Face BD) approaches. Names inside the parenthesis refer to the training dataset. (b) Illustration of boundary detection results on COFW test images: (top row) results of generic boundary detector (trained on BSDS), (second row) results of our boundary detector which trained to detect boundaries of faces. Our model suppresses edges belonging to the parts of the face (e.g. lips, eyes, nose) and boosts edges around the faces' masks (e.g. boundaries between face and hairs and boundaries between chin and neck).

with the found detection. Second, we use images from the PASCAL VOC 2010 segmentation dataset and include each segmented (non-face) object as a candidate binary mask, running our current model to find the optimal locations and mixtures of the keypoints for that segmentation. Although faces have specific shapes and the foreground segmentation of the other objects may look very different, mining non-face segments produces additional useful hard negatives in which keypoints are detected as occluded in order to match the segment shape.

Our hard negative mining method is not optimal and it may not find the highest scoring negatives. But, we found that we are still able to optimize the model parameters by including the two types of training data (with and without segmentation features). The first type of training data forces the learning procedure to find a good appearance, deformation and biases weights, while the second type of training data helps the learning procedure to calibrate the masks' weights with the other weights of the model.

4.2.3 Part Detection-guided Segmentation

Our landmark localization model predicts visibility of each keypoint but these predictions are independent and may be spatially inconsistent. Given an estimation of the keypoints' locations and their visibility flags, we can use bottom-up cues to calculate an estimation of the visible face region which can then provide propagate visibility information among distant keypoints that are not neighbors in the part-model tree topology. In this section we describe how this segmentation is computed, guided by a detection.

Class Specific Boundary Detection

Our goal is to correctly segment out the image region corresponding to the face from low-level cues. A generic boundary detector finds all the boundaries in the given image which typically includes internal contours in the face region (e.g. eyes, lips) which are not helpful for the segmentation of the whole face. Also, a generic boundary detector may not detect some specific kind of boundaries around the face like boundaries between the hair and the face (top row of Fig. 4.4 (b)).

To generate a high-quality segmentation, we train a random forest to specifically detect those boundaries relevant for face segmentation. To make a better boundary detector for our purpose, we train a structured random forests [32] on images from the COFW training dataset in which we manually labeled face foreground masks. The ground-truth boundaries for each training patch were computed based on the segmentation masks. Hence, edges arising from parts of the face such as eyes and lips are labeled as non-boundaries and the boundary detector learns to not return strong boundaries for them (bottom row of Fig. 4.4(b)).

To learn a class-specific boundary detector it is useful to distinguish the figure-ground ori-

entation of each boundary. Faces regions have specific texture and color patterns that distinguish them from background. However, the original entropy-based splitting function proposed in [32] treats the inside and outside symmetrically. We thus modified the distance function for clustering training examples to operate on the binary segmentation mask rather than on the boundary map. As shown in Fig. 4.4 (a), this modification yielded a small but significant gain in the boundary detection accuracy of face specific boundaries. We also note that while generic boundary detectors are typically run at multiple image resolutions, the detection output provides a canonical scale for each face so we can run at a fixed object resolution during training and testing.

Face Segmentation via Graph Cuts

To convert boundary detection into a segmentation, we use the available code of [32] to compute Ultrametric Contour Map (UCM) of boundary image [2]. This provides a set of superpixels and the boundary weights between them. We then use binary graph cut [10] to partition these superpixels into foreground and background where the unary potentials are based on the location and occlusion state of the detector part masks $\{(l_i, o_i)\}$ and the pairwise potential between superpixels i and j with boundary weight $w_{i,j}$ is computed as $\omega_{i,j} = \exp(-w_{i,j}^{\beta})$. We can write the resulting scoring function as

$$S(Z|l, o, I) = \sum_i \hat{\gamma}(l, o)_i Z_i + \lambda \sum_{i,j} \omega_{i,j}[Z_i = Z_j] \tag{4.2}$$

where $\gamma(l, o)$ is the accumulation of the segmentation masks γ_j associated with all of the keypoints placed down at locations l with visibilities o (visualized in bottom row of Fig. 4.3) and averaged over the support of superpixel i.

Interpolating unary potentials Given the location of the parts estimated by the detector, our scoring function only provides sparse information about which pixels belong to

the occluder (see Fig. 4.3). To interpolate these sparse estimates, we make use of the UCM boundary image and set the unary potential for that superpixel to be the difference between the distance to the nearest visible keypoint and the nearest occluded keypoint where distance is computed as the shortest path length through the superpixel graph. [1]

More formally, we calculate the Signed Geodesic Distance Transform (SGDT) [65] of visible and occluded keypoints:

$$D(i, F, B) = \min_{j \in B}[d(i, j)] - \min_{j \in F}[d(i, j)]$$

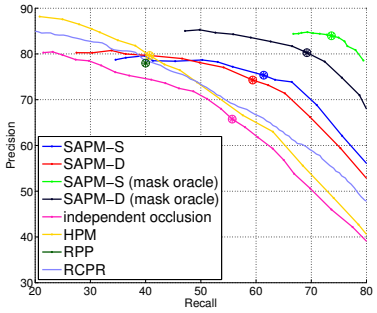
where $d(i, j)$ is the geodesic distance or length of shortest path between superpixel i and the keypoint j, F is the set of visible keypoints and B is the set of occluded keypoints. We set the unary in our final graph cut problem to be $\hat{\gamma}(l, o)_i = D(i, F, B)$ with $F = [\gamma(l, o) > 0]$ and $B = [\gamma(l, o) < 0]$.

The top row in Fig. 4.2 shows the distance to the F (green points) and the distance to the B (red points). When computing the segmentation, we offset the seed points relative to detected keypoint locations for all keypoints on the object boundary. For each visible jaw keypoint we include a foreground seed which is offset towards the estimate of the face center and background seeds generated by an offset outward from the face center. The third image shows the resulting difference of the first two images (SGDT) which we use as our unary potential for the graph cut problem. As can be seen, the non-occluded face region has higher values compared to the other regions of the image.

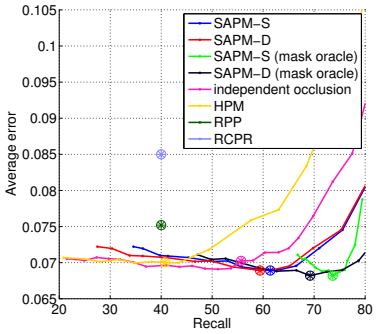
[1] We note that without interpolation, our detection scheme can be viewed as alternating minimization of a single unified objective (Eqn. 4.1 + the pairwise graphcut potential in Eqn. 4.2). Including interpolation in the coupling term γ results in all-pairs quadratic interactions between part locations we cannot efficiently optimize in the detection phase. However, in practice the addition of interpolation gives better performance which we judge well worth the loss of easy theoretical convergence guarantees.

	iter 1	iter 2	iter 3
ave err	0.070	0.069	0.069
FR	8.9%	7.9%	7.7%
F-meas	58.3	65.6	66.8
overlap	-	0.832	0.835

(a)



(b)



(c)

Figure 4.5: (a) Average keypoint localization error, failure rate, occlusion prediction F-measure and segmentation overlap for our model as a function iteration. In the second and third iterations, the estimated segmentation influences the keypoint localization and visibility, substantially improving precision and recall of occlusion. (b) Occlusion prediction accuracy for our segmentation-aware part model (SAPM) based on either using occlusion predicted by the keypoint detector (SAPM-D) or the segmentation mask (SAPM-S). We also plot performance when the model is provided the the ground-truth segmentation mask by an oracle. This improves occlusion prediction but does not improve the keypoint localization as shown in in (c) which plots average error parameterized by recall.

4.3 Experimental Evaluation

We evaluate the performance of our method and related baselines on the 507 test images from the Caltech Occluded Faces in the Wild (COFW) [12] dataset which was designed to evaluate landmark localization performance in the presence of occlusion. To evaluate our predicted face segmentations, we compare our predicted masks with the manually annotated masks for COFW testing data provided by the authors of [61]. Figure 4.6 shows example outputs of our model run on example COFW images. The model produces both a foreground mask and estimates of the keypoint locations and occlusion states.

The COFW dataset includes both high-resolution color images and down-sampled gray-scale images. Our detection and localization code was run on the gray-scale images but we evaluated versions of our segmentation method on both gray-scale and color images down-sampled to match the gray-scale resolution.

	ave err	FR	F-meaure(P/R)	ave overlap	global	ave(face)
TCDCN[112]	0.080	-	-	-	-	-
R-CR-C[40]	0.073	12.2%	-	-	-	-
DPM[116]	0.079	16.80%	-	-	-	-
structured forest [61]	-	-	[66.56 (80%/57%)]	-	[83.9%]	[88.6%]
RPP[100]	0.075	16.2%	52.88 (78%/40%)	0.724	-	-
RCPR[12]	0.085	20%	53.33 (80%/40%)	-	-	-
HPM[47]	0.072	9.29%	53.86 (79.6%/40.7%)	-	-	-
our model(gray)	0.070	8.70%	65.07 (73.7%/58.3%)	0.828	88.0%	86.5%
our model(rgb)	0.069	7.71%	67.69 (75.4%/61.4%)	0.835	88.6%	87.1%
human[12]	0.056	0.01%	-	-	-	-

Table 4.1: Comparison of landmark localization, occlusion prediction and mask prediction between our model and previous results on the COFW test data [12]. Our method outperforms all the previous methods in the average error, failure rate and F-measure for occlusion prediction of keypoints. We also achieve better segmentation accuracy based on previously used segment overlap and recall metrics ([100, 61]). [numbers] reported for a subset of image (300 of 507).

Model Training Details To train our model, we used a set of 1758 near-frontal training images taken from the HELEN [66] training set using the 68 keypoint annotations provided by 300-W [78]. From each training image, we generate 8 synthetically occluded "virtual positives" yielding a final training set of 15822 positives. For negative images we used 6000 images that does not contain person from PASCAL VOC 2010 trainval set. To benchmark the keypoint localization of this 68 keypoint model on COFW (which only has 29 landmark points), we used linear regression to learn a mapping from the set of locations returned by our part model [44]. To fit the linear regression coefficients, we ran the model on the COFW training data set which has 29 keypoint annotations.

For segmentation training, we augmented 500 images of the COFW training data with manually labeled face masks and used the cropped images around the faces for training our boundary detector. We used cross-validation on the COFW training data to set the graphcut parameters λ and β.

4.3.1 Keypoint Localization and Occlusion Prediction

We report the average landmark localization error across the entire test set as well as the percentage of "failures" of, test images that had landmark localization error above 0.1. Landmark localization error is computed as the average of landmark distances to the ground-truth' landmarks, normalized with the distance between the centers of eyes (inter-ocular distance). To evaluate occlusion prediction, we compute precision and recall of the model keypoint predictions relative to the ground-truth occlusion for each keypoint.

Figure 4.5(a) shows the evaluations of various metrics for different iterations of our method. The average error and failure rate slightly improves in the second and third iterations while precision and recall of occlusion significantly improves in the second iteration once the part model has an estimate of the face masks.

The first three columns of Table 4.1 show a comparison of the landmark localization accuracy and the occlusion prediction accuracy between our model and previous results on the COFW test data. Our method outperforms the other approaches and has a better average error, failure rate and precision/recall of occlusion.

We generate a precision/recall curve for occlusion prediction by manually changing the model parameters to induce more predicted occlusions. The bias parameter $b_{ij}^{s_i,s_j,o_i,o_j}$ favors particular co-occurrences of part types. By increasing the bias for occluded configurations we can encourage the model to use those configurations on test.

Let $b_{ij}^{s_i,s_j,o_i,o_j}$ be a learned bias parameter between an occluded leaf and its parent. To make the model favor occluded parts, we modify this parameter to $b_{ij}^{s_i,s_j,o_i,o_j} + abs(b_{ij}^{s_i,s_j,o_i,o_j}) \times \delta$. Thus, we can vary δ to have different precsions and recalls of occlusion.

Precision/recall of occlusion prediction and the corresponding average errors parameterized by recall for our model and some baselines are shown in Fig. 4.5 parts (b) and (c). Previ-

Figure 4.6: Examples of landmark localization and mask estimation for images from the COFW test data produced by our model.

ous methods have a poor performance when recall of occlusion is high. Their precision of occlusion and average error drop very fast, as the recall of occlusion increases. However, our model performs very well in the challenging regime of high occlusion recall.

4.3.2 Segmentation Prediction

Boundary Detection To evaluate boundary detectors for our purpose, we benchmarked boundary prediction on the cropped COFW test images using ground-truth boundaries that only include boundaries around the face masks (no internal contours or background segments). We computed ODS [3] and average precision (AP) for the evaluation. Results for different detectors are shown in Fig. 4.4 (a). The first row of the table shows the results for the generic boundary detector of [32]. By just re-training this method on COFW training data (4th row), the average precision increases from 0.16 to 0.27. Using the segmentation feature for the clustering of patches (as described in Section 4.2.3) increases the average precision to 0.30. We also found that using color images (4th and 5th row) gave a noticeable performance boost over gray-scale (2nd, 3rd row).

Mask prediction We evaluate the segmentation accuracy of our method on the COFW test images. The last three columns of Table 4.1 show a comparison between accuracy of our method and previous results. To compare our method with RPP [100], we compute average overlap between the ground truth masks and the predicted masks inside the face bounding boxes. [2] Our method has significantly higher overlap (0.835) compared to RPP (0.724).

To compare our mask prediction result with [61], we calculate global and ave(face) metrics which show the percentage of all pixels that are correctly classified and the average recall of face pixels, respectively inside the COFW bounding box. Our method has better performance according to global metric (88.6 vs 83.9) and close performance according to the ave(face) metric (87.1 vs 88.6). We note that the results of [61] are over a random subset of 300 images (out of 507) of COFW test data.

4.4 Conclusion

In this chapter, we have presented a method that uses both top-down and bottom-up features to estimate the occluded parts of the face. Our method combines an efficient part-based model and binary segmentation method to accurately localize landmarks and segment out the visible portion of the face. Unlike approaches to detection and pose estimation that treat occluders as outliers and ignore image evidence in occluded regions, our model leverages the appearance of occluder boundaries throughout the image via object-specific segmentation.

Unlike HPM or other part models, which can only enforce consistent occlusion patterns for keypoints if they are nearby in the tree topology, our scoring function couples all keypoints globally by encouraging configurations whose occlusion pattern is consistent with

[2] We use the same protocol as previous work but note that because the COFW bounding boxes are tight, some areas of face masks are outside of the bounding boxes and computing the average overlap over the entire image and over bounding boxes is not equivalent. Computing over the whole image yielded 0.796 and 0.787 for our color and gray-scale models respectively.

some bottom-up segmentation. This coupling results in a joint segmentation and detection system with state-of-the-art performance for simultaneous landmark localization, occlusion prediction and face segmentation.

Chapter 5

Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation

CNN architectures have terrific recognition performance but rely on spatial pooling which makes it difficult to adapt them to tasks that require dense, pixel-accurate labeling. This chapter makes two contributions: (1) We demonstrate that while the apparent spatial resolution of convolutional feature maps is low, the high-dimensional feature representation contains significant sub-pixel localization information. (2) We describe a multi-resolution reconstruction architecture based on a Laplacian pyramid that uses skip connections from higher resolution feature maps and multiplicative gating to successively refine segment boundaries reconstructed from lower-resolution maps. This approach yields state-of-the-art semantic segmentation results on the PASCAL VOC and Cityscapes segmentation benchmarks without resorting to more complex random-field inference or instance detection driven architectures.

5.1 Introduction

Deep convolutional neural networks (CNNs) have proven highly effective at semantic segmentation due to the capacity of discriminatively pre-trained feature hierarchies to robustly represent and recognize objects and materials. As a result, CNNs have significantly outperformed previous approaches (e.g., [87, 16, 15]) that relied on hand-designed features and recognizers trained from scratch. A key difficulty in the adaption of CNN features to segmentation is that feature pooling layers, which introduce invariance to spatial deformations required for robust recognition, result in high-level representations with reduced spatial resolution. In this chapter, we investigate this *spatial-semantic uncertainty principle* for CNN hierarchies (see Fig.5.1) and introduce two techniques that yield substantially improved segmentations.

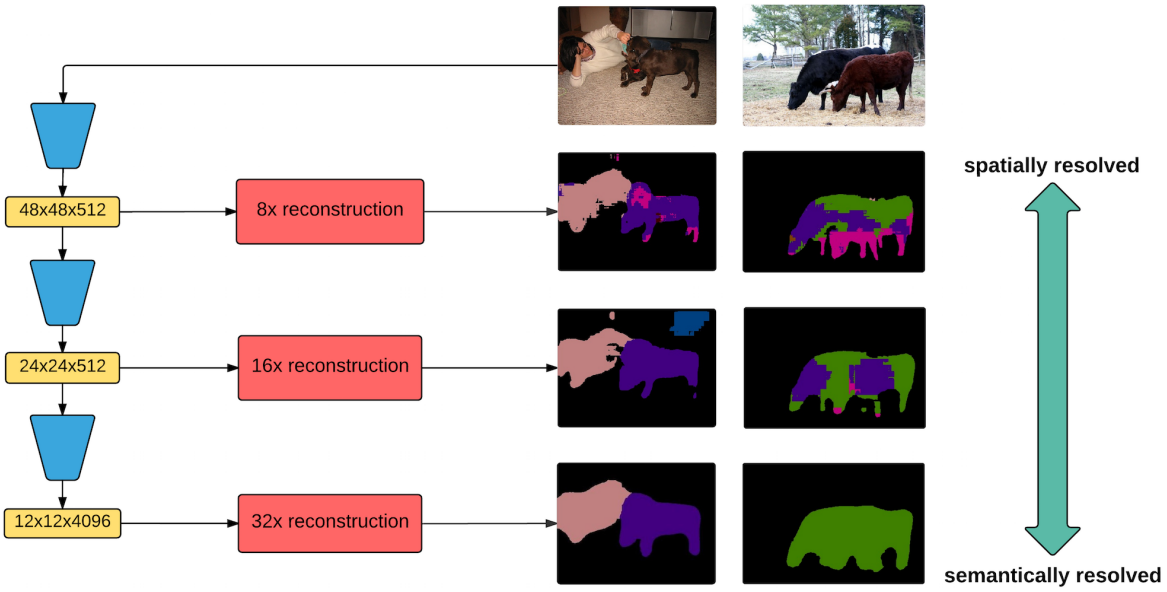
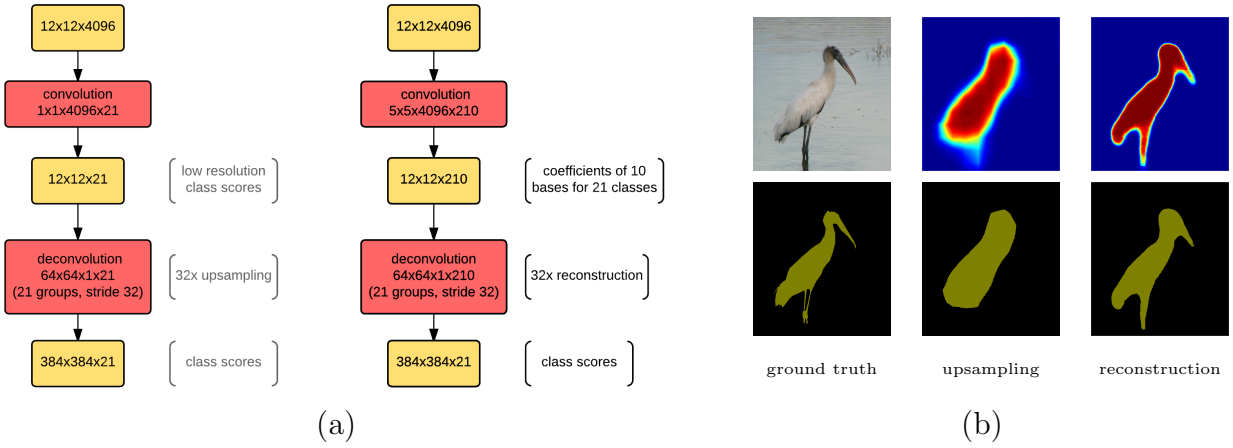


Figure 5.1: In this chapter, we explore the trade-off between spatial and semantic accuracy within CNN feature hierarchies. Such hierarchies generally follow a spatial-semantic uncertainty principle in which high levels of the hierarchy make accurate semantic predictions but are poorly localized in space while at low levels, boundaries are precise but labels are noisy. We develop reconstruction techniques for increasing spatial accuracy at a given level and refinement techniques for fusing multiple levels that limit these tradeoffs and produce improved semantic segmentations.

First, we tackle the question of how much spatial information is represented at high levels of the feature hierarchy. A given spatial location in a convolutional feature map corresponds to a large block of input pixels (and an even larger "receptive field"). While max pooling in a single feature channel clearly destroys spatial information in that channel, spatial filtering prior to pooling introduces strong correlations across channels which could, in principle, encode significant "sub-pixel" spatial information across the high-dimensional vector of sparse activations. We show that this is indeed the case and demonstrate a simple approach to spatial decoding using a small set of data-adapted basis functions that substantially improves over common upsampling schemes (see Fig. 5.2).

Second, having squeezed more spatial information from a given layer of the hierarchy, we turn to the question of fusing predictions across layers. A standard approach has been to either concatenate features (e.g., [55]) or linearly combine predictions (e.g., [71]). Concatenation is appealing but suffers from the high dimensionality of the resulting features. On the other hand, additive combinations of predictions from multiple layers does not make good use of the relative spatial-semantic content tradeoff. High-resolution layers are shallow with small receptive fields and hence yield inherently noisy predictions with high pixel-wise loss. As a result, we observe their contribution is significantly down-weighted relative to low-resolution layers during linear fusion and thus they have relatively little effect on final predictions.

Inspired in part by recent work on residual networks [56, 57], we propose an architecture in which predictions derived from high-resolution layers are only required to correct residual errors in the low-resolution prediction. Importantly, we use multiplicative gating to avoid integrating (and hence penalizing) noisy high-resolution outputs in regions where the low-resolution predictions are confident about the semantic content. We call our method *Laplacian Pyramid Reconstruction and Refinement* (LRR) since the architecture uses a Laplacian reconstruction pyramid [13] to fuse predictions. Indeed, the class scores predicted at each level of our architecture typically look like bandpass decomposition of the full resolution



(a) (b)

Figure 5.2: (a) Upsampling architecture for FCN32s network (left) and our 32x reconstruction network (right). (b) Example of Class conditional probability maps and semantic segmentation predictions from FCN32s which performs upsampling (middle) and our 32x reconstruction network (right).

segmentation mask (see Fig. 5.3).

5.2 Related Work

The inherent lack of spatial detail in CNN feature maps has been attacked using a variety of techniques. One insight is that spatial information lost during max-pooling can in part be recovered by unpooling and deconvolution [109] providing a useful way to visualize input dependency in feed-forward models [108]. This idea has been developed using learned deconvolution filters to perform semantic segmentation [76]. However, the deeply stacked deconvolutional output layers are difficult to train, requiring multi-stage training and more complicated object proposal aggregation.

A second key insight is that while activation maps at lower-levels of the CNN hierarchy lack object category specificity, they do contain higher spatial resolution information. Performing classification using a "jet" of feature map responses aggregated across multiple layers has been successfully leveraged for semantic segmentation [71], generic boundary detection [97], simultaneous detection and segmentation [55], and scene recognition [101]. Our architecture

shares the basic skip connections of [71] but uses multiplicative, confidence-weighted gating when fusing predictions.

Our techniques are complementary to a range of other recent approaches that incorporate object proposals [26, 76], attentional scale selection mechanisms [21], and conditional random fields (CRF) [19, 68, 70]. CRF-based methods integrate CNN score-maps with pairwise features derived from superpixels [25, 75] or generic boundary detection [63, 18] to more precisely localize segment boundaries. We demonstrate that our architecture works well as a drop in unary potential in fully connected CRFs [64] and would likely further benefit from end-to-end training [113].

5.3 Reconstruction with Learned Basis Functions

A standard approach to predicting pixel class labels is to use a linear convolution to compute a low-resolution class score from the feature map and then upsample the score map to the original image resolution. A bilinear kernel is a suitable choice for this upsampling and has been used as a fixed filter or an initialization for the upsampling filter [55, 71, 19, 113, 21, 26, 49]. However, upsampling low-resolution class scores necessarily limits the amount of detail in the resulting segmentation (see Fig. 5.2 (a)) and discards any sub-pixel localization information that might be coded across the many channels of the low-resolution feature map. The simple fix of upsampling the feature map prior to classification poses computational difficulties due to the large number of feature channels (e.g. 4096). Furthermore, (bilinear) upsampling commutes with 1x1 convolutions used for class prediction so performing per-pixel linear classification on an upsampled feature map would yield equivalent results unless additional rounds of (non-linear) filtering were carried out on the high-resolution feature map.

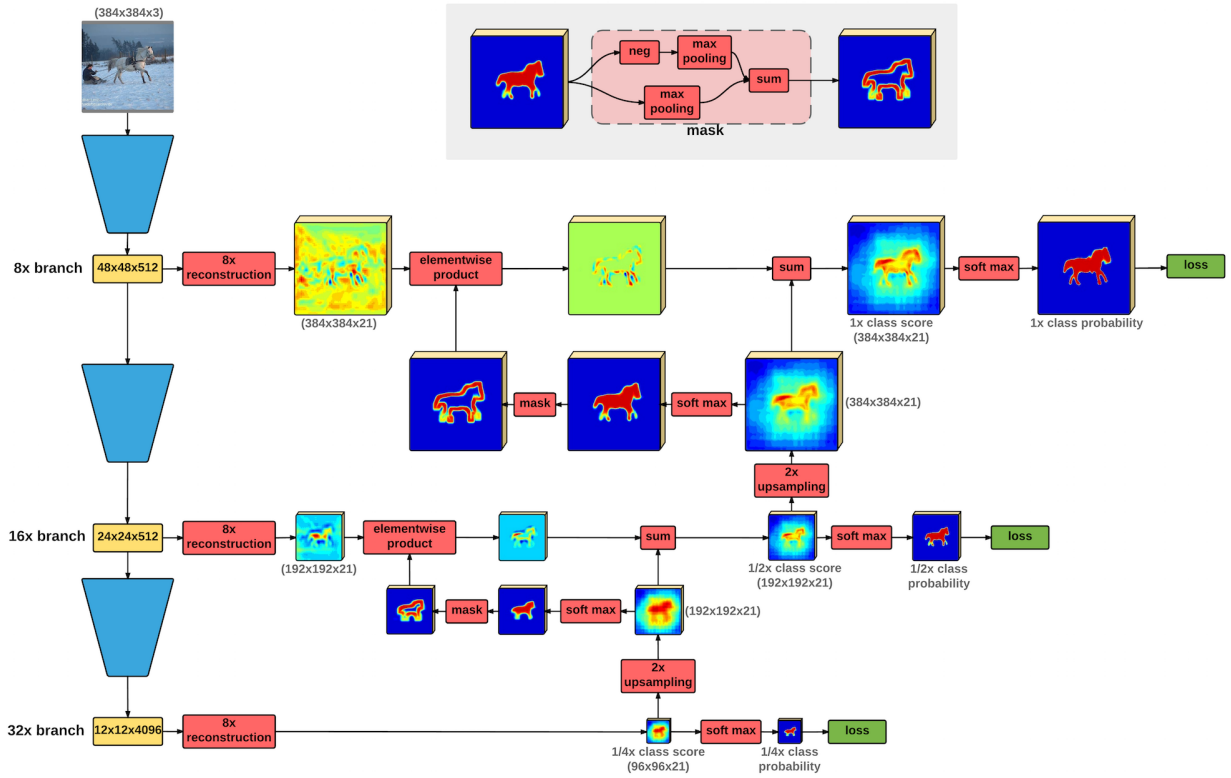


Figure 5.3: Overview of our Laplacian pyramid reconstruction network architecture. We use low-resolution feature maps in the CNN hierarchy to reconstruct a coarse, low-frequency segmentation map and then refine this map by adding in higher frequency details derived from higher-resolution feature maps. Boundary masking (inset) suppresses the contribution of higher resolution layers in areas where the segmentation is confident, allowing the reconstruction to focus on predicting residual errors in uncertain areas (e.g., precisely localizing object boundaries). At each resolution layer, the reconstruction filters perform the same amount of upsampling which depends on the number of layers (e.g., our LRR-4x model utilizes 4x reconstruction on each of four branches). Standard 2x bilinear upsampling is applied to each class score map before combining it with higher resolution predictions.

To extract more detailed spatial information, we avoid immediately collapsing the high-dimensional feature map down to low-resolution class scores. Instead, we express the spatial pattern of high-resolution scores using a linear combination of high-resolution basis functions whose coefficients are predicted from the feature map (see Fig. 5.2 (a)). We term this approach "reconstruction" to distinguish it from the standard upsampling (although bilinear upsampling can clearly be seen as special case with a single basis function).

Reconstruction by deconvolution: In our implementation, we tile the high-resolution score map with overlapping basis functions (e.g., for 4x upsampled reconstruction we use basis functions with an 8x8 pixel support and a stride of 4). We use a convolutional layer to predict K basis coefficients for each of C classes from the high-dimensional, low-resolution feature map. The group of coefficients for each spatial location and class are then multiplied by the set of basis function for the class and summed using a standard deconvolution (convolution transpose) layer.

To write this explicitly, let s denote the stride, $q_s(i) = \lfloor \frac{i}{s} \rfloor$ denote the quotient, and $m_s(i) = i \, mod \, s$ the remainder of i by s. The reconstruction layer that maps basis coefficients $X \in \mathbb{R}^{H \times W \times K \times C}$ to class scores $Y \in \mathbb{R}^{sH \times sW \times C}$ using basis functions $B \in \mathbb{R}^{2s \times 2s \times K \times C}$ is given by:

$$Y_c[i,j] = \sum_{k=0}^{K-1} \sum_{(u,v) \in \{0,1\}^2} B_{k,c}\left[m_s(i) + s \cdot u, m_s(j) + s \cdot v\right] \cdot X_{k,c}\left[q_s(i) - u, q_s(j) - v\right]$$

where $B_{k,c}$ contains the k-th basis function for class c with corresponding spatial weights $X_{k,c}$. We assume $X_{k,c}$ is zero padded and Y_c is cropped appropriately.

Connection to spline interpolation: We note that a classic approach to improving on bilinear interpolation is to use a higher-order spline interpolant built from a standard set of non-overlapping polynomial basis functions where the weights are determined analytically to assure continuity between neighboring patches. Our approach using learned filters and basis functions makes minimal assumptions about mapping from high dimensional activations to the coefficients X but also offers no guarantees on the continuity of Y. We address this in part by using larger filter kernels (i.e., $5 \times 5 \times 4096$) for predicting the coefficients $X_{k,c}$ from the feature activations. This mimics the computation used in spline interpolation of introducing linear dependencies between neighboring basis weights and empirically improves

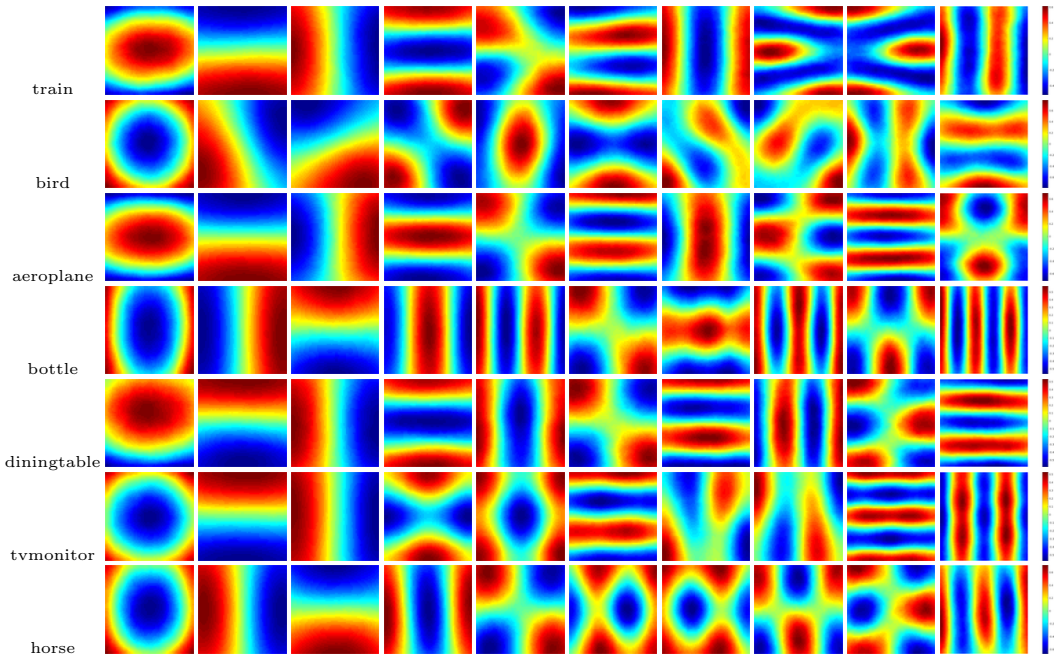


Figure 5.4: Category-specific basis functions for reconstruction are adapted to modeling the shape of a given object class. For example, airplane segments tend to be elongated in the horizontal direction while bottles are elongated in the vertical direction.

continuity of the output predictions.

Learning basis functions: To leverage limited amounts of training data and speed up training, we initialize the deconvolution layers with a meaningful set of filters estimated by performing PCA on example segment patches. For this purpose, we extract 10000 patches for each class from training data where each patch is of size 32×32 and at least 2% of the patch pixels are members of the class. We apply PCA on the extracted patches to compute a class specific set of basis functions. Example bases for different categories of PASCAL VOC dataset are shown in Fig. 5.4. Interestingly, there is some significant variation among classes due to different segment shape statistics. We found it sufficient to initialize the reconstruction filters for different levels of the reconstruction pyramid with the same basis set (downsampled as needed). In both our model and the FCN bilinear upsampling model, we observed that end-to-end training resulted in insignificant ($< 10^{-7}$) changes to the basis functions.

We experimented with varying the resolution and number of basis functions of our reconstruction layer built on top of the ImageNet-pretrained VGG-16 network. We found that 10 functions sampled at a resolution of 8×8 were sufficient for accurate reconstruction of class score maps. Models trained with more than 10 basis functions commonly predicted zero weight coefficients for the higher-frequency basis functions. This suggests some limit to how much spatial information can be extracted from the low-res feature map (i.e., roughly 3x more than bilinear). However, this estimate is only a lower-bound since there are obvious limitations to how well we can fit the model. Other generative architectures (e.g., using larger sparse dictionaries) or additional information (e.g., max pooling "switches" in deconvolution [109]) may do even better.

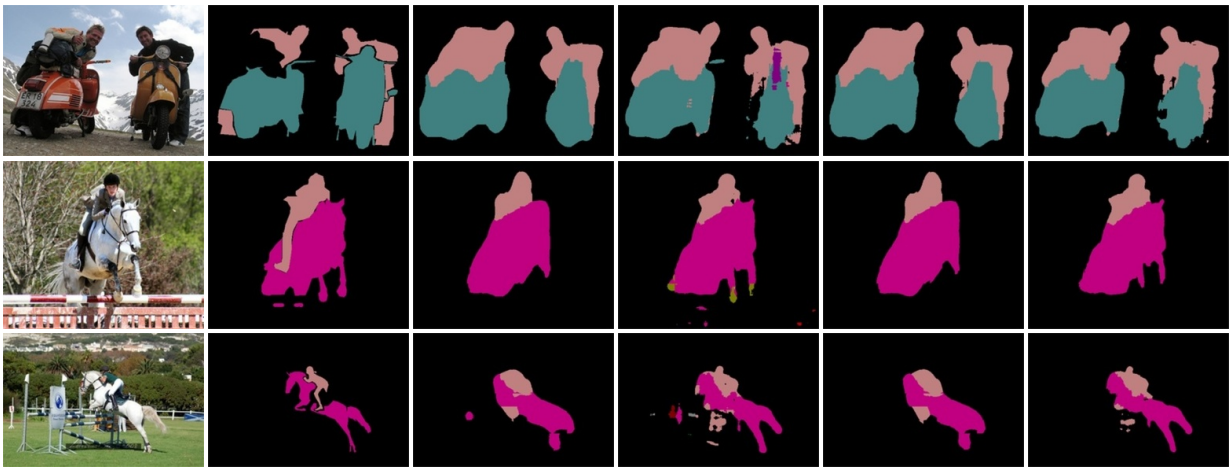


Figure 5.5: Visualization of segmentation results produced by our model with and without boundary masking. For each row, we show the input image, ground-truth and the segmentation results of 32x and 8x layers of our model without masking (middle) and with masking (right). The segmentation results for 8x layer of the model without masking has some noise not present in the 32x output. Masking allows such noise to be repressed in regions where the 32x outputs have high confidence.

5.4 Laplacian Pyramid Refinement

The basic intuition for our multi-resolution architecture comes from Burt and Adelson's classic Laplacian Pyramid [13], which decomposes an image into disjoint frequency bands

using an elegant recursive computation (analysis) that produces appropriately down-sampled sub-bands such that the sum of the resulting sub-bands (synthesis) perfectly reproduces the original image. While the notion of frequency sub-bands is not appropriate for the non-linear filtering performed by standard CNNs, casual inspection of the response of individual activations to shifted input images reveals a power spectral density whose high-frequency components decay with depth leaving primarily low-frequency components (with a few high-frequency artifacts due to disjoint bins used in pooling). This suggests the possibility that the standard CNN architecture could be trained to serve the role of the analysis pyramid (predicting sub-band coefficients) which could then be assembled using a synthesis pyramid to estimate segmentations.

Figure 5.3 shows the overall architecture of our model. Starting from the coarse scale "low-frequency" segmentation estimate, we carry out a sequence of successive refinements, adding in information from "higher-frequency" sub-bands to improve the spatial fidelity of the resulting segmentation masks. For example, since the 32x layer already captures the coarse-scale support of the object, prediction from the 16x layer does not need to include this information and can instead focus on adding finer scale refinements of the segment boundary. [1]

Boundary masking: In practice, simply upsampling and summing the outputs of the analysis layers does not yield the desired effect. Unlike the Laplacian image analysis pyramid, the high resolution feature maps of the CNN do not have the "low-frequency" content subtracted out. As Fig.5.1 shows, high-resolution layers still happily make "low-frequency" predictions (e.g., in the middle of a large segment) even though they are often incorrect. As a result, in an architecture that simply sums together predictions across layers, we found the learned parameters tend to down-weight the contribution of high-resolution predictions to the sum

[1] Closely related architectures were used in [29] for generative image synthesis where the output of a lower-resolution model was used as input for a CNN which predicted an additive refinement, and in [80], where fusing and refinement across levels was carried out via concatenation followed by several convolution+ReLU layers.

VOC 2011-val	pixel acc.	mean acc.	mean IoU
FCN-32s	89.1%	73.3%	59.4%
FCN-16s	90.0%	75.7%	62.4%
FCN-8s	90.3%	75.9%	62.7%
LRR-32x (w/o aug)	90.7%	78.9%	64.1%
LRR-32x	91.5%	81.6%	66.8%
LRR-16x	91.8%	81.6%	67.8%
LRR-8x	92.4%	83.2%	69.5%
LRR-4x	92.2%	83.7%	69.0%
LRR-4x-ms	92.8%	84.6%	71.4%

Figure 5.6: Comparison of our segment reconstruction model, LRR (without boundary masking) and the baseline FCN model[71] which uses upsampling. We find consistent benefits from using a higher-dimensional reconstruction basis rather than upsampling class prediction maps. We also see improved performance from using multi-scale training augmentation, fusing multiple feature maps, and running on multiple scales at test time. Note that the performance benefit of fusing multiple resolution feature maps diminishes with no gain or even decrease performance from adding in the 4x layer. Boundary masking (cf. Fig.5.7) allows for much better utilization of these fine scale features.

in order to limit the potentially disastrous effect of these noisy predictions. However, this hampers the ability of the high-resolution predictions to significantly refine the segmentation in areas containing high-frequency content (i.e., segment boundaries).

To remedy this, we introduce a masking step that serves to explicitly subtract out the "low-frequency" content from the high-resolution signal. This takes the form of a multiplicative gating that prevents the high-resolution predictions from contributing to the final response in regions where lower-resolution predictions are confident. The inset in Fig.5.3 shows how this boundary mask is computed by using a max pooling operation to dilate the confident foreground and background predictions and taking their difference to isolate the boundary. The size of this dilation (pooling size) is tied to the amount of upsampling between successive layers of the pyramid, and hence fixed at 9 pixels in our implementation.

5.5 Experiments

We now describe a number of diagnostic experiments carried out using the PASCAL VOC [36] semantic segmentation dataset. In these experiments, models were trained on train-

	VOC			VOC+COCO			
	VGG-16			VGG-16			ResNet
VOC 2011-val	unmasked	masked	masked+DE	umasked	masked	masked+DE	masked+DE
LRR-4x(32x)	67.1%	67.0%	68.8%	71.3%	71.2%	72.9%	76.7%
LRR-4x(16x)	68.6%	69.2%	70.0%	72.1%	72.4%	73.9%	78.0%
LRR-4x(8x)	69.3%	70.3%	70.9%	72.9%	73.4%	74.9%	78.3%
LRR-4x	69.3%	70.5%	71.1%	72.9%	73.6%	75.1%	78.4%
LRR-4x-ms	71.9%	73.0%	73.6%	74.0%	75.0%	76.6%	79.2%
LRR-4x-ms-crf	73.2%	74.1%	74.6%	75.0%	76.1%	77.5%	79.9%

Figure 5.7: Mean intersection-over-union (IoU) accuracy for intermediate outputs at different levels of our Laplacian reconstruction architecture trained with and without boundary masking (value in parentheses denotes an intermediate output of the full model). Masking allows us to squeeze additional gains out of high-resolution feature maps by focusing only on low-confidence areas near segment boundaries. Adding dilation and erosion losses (DE) to the 32x branch improves the accuracy of 32x predictions and as a result the overall performance. Running the model at multiple scales and performing post-processing using a CRF yielded further performance improvements.

ing/validation set split specified by [54] which includes 11287 training images and 736 held out validation images from the PASCAL 2011 val set. We focus primarily on the average Intersection-over-Union (IoU) metric which generally provides a more sensitive performance measure than per-pixel or per-class accuracy. We conduct diagnostic experiments on the model architecture using this validation data and test our final model via submission to the PASCAL VOC 2012 test data server, which benchmarks on an additional set of 1456 images. We also report test benchmark performance on the recently released Cityscapes [24] dataset.

5.5.1 Parameter Optimization

We augment the layers of the ImageNet-pretrained VGG-16 network [88] or ResNet-101 [56] with our LRR architecture and fine-tune all layers via back-propagation. All models were trained and tested with Matconvnet [92] on a single NVIDIA GPU. We use standard stochastic gradient descent with batch size of 20, momentum of 0.9 and weight decay of 0.0005. The models and code are available at `https://github.com/golnazghiasi/LRR`.

Stage-wise training: Our 32x branch predicts a coarse semantic segmentation for the in-

put image while the other branches add in details to the segmentation prediction. Thus 16x, 8x and 4x branches are dependent on 32x branch prediction and their task of adding details is meaningful only when 32x segmentation predictions are good. As a result we first optimize the model with only 32x loss and then add in connections to the other layers and continue to fine tune. At each layer we use a pixel-wise softmax log loss defined at a lower image resolution and use down-sampled ground-truth segmentations for training. For example, in LRR-4x the loss is defined at 1/8, 1/4, 1/2 and full image resolution for the 32x, 16x, 8x and 4x branches, respectively.

Dilation erosion objectives: We found that augmenting the model with branches to predict dilated and eroded class segments in addition of the original segments helps guide the model in predicting more accurate segmentation. For each training example and class, we compute a binary segmentation using the ground-truth and then compute its dilation and erosion using a disk with radius of 32 pixels. Since dilated segments of different classes are not mutually exclusive, a k-way soft-max is not appropriate so we use logistic loss instead. We add these Dilation and Erosion (DE) losses to the 32x branch (at 1/8 resolution) when training LRR-4x. Adding these losses increased mean IoU of the 32x branch predictions from 71.2% to 72.9% and also the overall multi-scale accuracy from 75.0% to 76.6 (see Fig.5.7, built on VGG-16 and trained on VOC+COCO).

Multi-scale Data Augmentation: We augmented the training data with multiple scaled versions of each training examples. We randomly select an image size between 288 to 704 for each batch and then scale training examples of that batch to the selected size. When the selected size is larger than 384, we crop a window with size of 384×384 from the scaled image. This augmentation is helpful in improving the accuracy of the model and increased mean IoU of our 32x model from 64.07% to 66.81% on the validation data (see Fig.5.6).

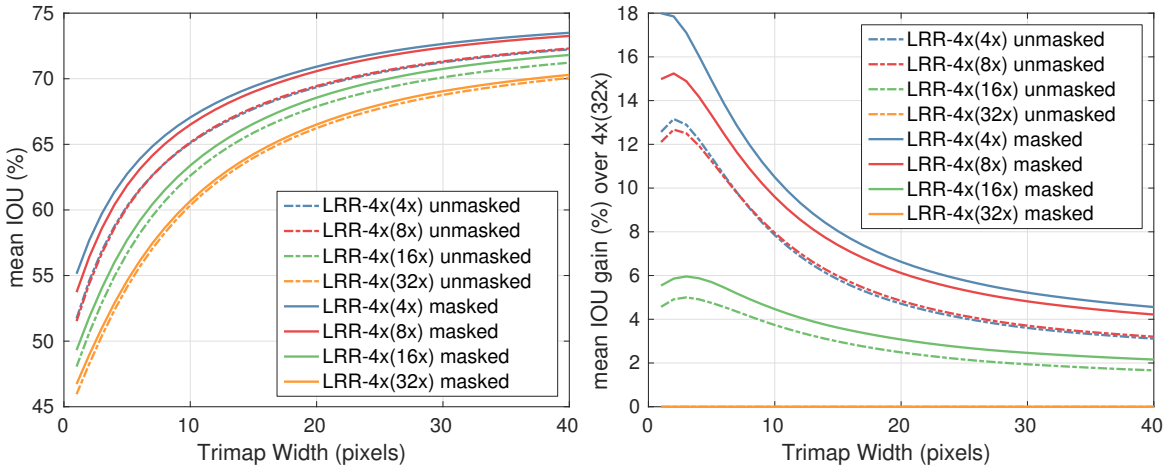


Figure 5.8: The benefit of Laplacian pyramid boundary refinement becomes even more apparent when focusing on performance near object boundaries. Plots show segmentation performance within a thin band around the ground-truth object boundaries for intermediate predictions at different levels of our reconstruction pyramid. (right) Measuring accuracy or mean IoU relative to the baseline 32x output shows that the high-resolution feature maps are most valuable near object boundaries while masking improves performance both near and far from boundaries.

5.5.2 Reconstruction vs Upsampling

To isolate the effectiveness of our proposed reconstruction method relative to simple upsampling, we compare the performance of our model without masking to the fully convolutional net (FCN) of [71]. For this experiment, we trained our model without scale augmentation using exactly same training data used for training the FCN models. We observed significant improvement over upsampling using reconstruction with 10 basis filters. Our 32x reconstruction model (w/o aug) achieved a mean IoU of 64.1% while FCN-32s and FCN-8s had a mean IoU of 59.4% and 62.7%, respectively (Fig. 5.6).

5.5.3 Multiplicative Masking and Boundary Refinement

We evaluated whether masking the contribution of high-resolution feature maps based on the confidence of the lower-resolution predictions resulted in better performance. We anticipated that this multiplicative masking would serve to remove noisy class predictions from

high-resolution feature maps in high-confidence interior regions while allowing refinement of segment boundaries. Fig. 5.5 demonstrates the qualitative effect of boundary masking. While the prediction from the 32x branch is similar for both models (relatively noise free), masking improves the 8x prediction noticeably by removing small, incorrectly labeled segments while preserving boundary fidelity. We compute mean IoU benchmarks for different intermediate outputs of our LRR-4x model trained with and without masking (Table 5.7). Boundary masking yields about 1% overall improvement relative to the model without masking across all branches.

Evaluation near Object Boundaries: Our proposed model uses the higher resolution feature maps to refine the segmentation in the regions close to the boundaries, resulting in a more detailed segmentation (see Fig. 5.11). However, boundaries constitute a relatively small fraction of the total image pixels, limiting the impact of these improvements on the overall IoU performance benchmark (see, e.g. Fig. 5.7). To better characterize performance differences between models, we also computed mean IoU restricted to a narrow band of pixels around the ground-truth boundaries. This partitioning into figure/boundary/background is sometimes referred to as a tri-map in the matting literature and has been previously utilized in analyzing semantic segmentation performance [19, 62].

Fig. 5.8 shows the mean IoU of our LRR-4x as a function of the width of the tri-map boundary zone. We plot both the absolute performance and performance relative to the low-resolution 32x output. As the curves confirm, adding in higher resolution feature maps results in the most performance gain near object boundaries. Masking improves performance both near and far from boundaries. Near boundaries masking allows for the higher-resolution layers to refine the boundary shape while far from boundaries the mask prevents those high-resolution layers from corrupting accurate low-resolution predictions.

	mean	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Only using VOC training data																					
FCN-8s[71]	62.2	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1
Hypercol[55]	62.6	68.7	33.5	69.8	51.3	70.2	81.1	71.9	74.9	23.9	60.6	46.9	72.1	68.3	74.5	72.9	52.6	64.4	45.4	64.9	57.4
Zoom-out[75]	69.6	85.6	37.3	**83.2**	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3
EdgeNet[18]	71.2	83.6	35.8	82.4	63.1	68.9	86.2	79.6	84.7	31.8	74.2	61.1	79.6	76.6	83.2	80.9	58.3	82.6	49.1	74.8	65.1
Attention[21]	71.5	86.0	38.8	78.2	63.1	70.2	89.6	84.1	82.9	29.4	75.2	58.7	79.3	78.4	83.9	80.3	53.5	82.6	51.5	79.2	64.2
DeepLab[19]	71.6	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7
CRFRNN[113]	72.0	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1
DeconvN[76]	72.5	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0
DPN [70]	74.1	87.7	**59.4**	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0
Adelaide[68]	75.3	90.6	37.6	80.0	**67.8**	**74.4**	92.0	85.2	86.2	**39.1**	81.2	58.9	83.8	83.9	84.3	**84.8**	**62.1**	83.2	**58.2**	80.8	**72.3**
LRR	74.7	89.2	40.3	81.2	63.9	73.1	91.7	86.2	87.2	35.4	80.1	62.4	82.6	84.4	84.8	81.7	59.5	83.6	54.3	83.7	69.3
LRR-CRF	**75.9**	**91.8**	41.0	83.0	62.3	74.3	**93.0**	**86.8**	**88.7**	36.6	**81.8**	**63.4**	**84.7**	**85.9**	**85.1**	83.1	62.0	**84.6**	55.6	**84.9**	70.0
Using VOC and COCO training data																					
EdgeNet[18]	73.6	88.3	37.0	**89.8**	63.6	70.3	87.3	82.0	87.6	31.1	79.0	61.9	81.6	80.4	84.5	83.3	58.4	86.1	55.9	78.2	65.4
CRFRNN[113]	74.7	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1
BoxSup[26]	75.2	89.8	38.0	89.2	**68.9**	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7
SBound[63]	75.7	90.3	37.9	89.6	67.8	74.6	89.3	84.1	89.1	35.8	83.6	66.2	82.9	81.7	85.6	84.6	60.3	84.8	60.7	78.3	68.3
Attention[21]	76.3	93.2	41.7	88.0	61.7	74.9	92.9	84.5	90.4	33.0	82.8	63.2	84.5	85.0	87.2	85.7	60.5	87.7	57.8	**84.3**	68.2
DPN [70]	77.5	89.0	**61.6**	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4
Adelaide[68]	77.8	**94.1**	40.4	83.6	67.3	**75.6**	93.4	84.4	88.7	41.6	**86.4**	63.3	85.5	**89.3**	85.6	86.0	**67.4**	**90.1**	62.6	80.9	**72.5**
LRR	77.9	91.4	43.2	87.9	64.5	75.0	93.1	86.7	90.6	42.4	82.9	68.1	85.2	87.8	88.6	86.4	65.4	85.0	62.2	83.3	71.6
LRR-CRF	**78.7**	93.2	44.2	89.4	65.4	74.9	**93.9**	**87.0**	**92.0**	**42.9**	83.7	**68.9**	**86.5**	88.0	**89.0**	**87.2**	67.3	85.6	**64.0**	84.1	71.5
ResNet + Using VOC and COCO training data																					
DeepLab[20]	**79.7**	**92.6**	**60.4**	91.6	63.4	**76.3**	95.0	88.4	**92.6**	32.7	**88.5**	67.6	**89.6**	**92.1**	87.0	**87.4**	63.3	**88.3**	**60.0**	**86.8**	74.5
LRR	78.7	90.8	44.4	94.0	**65.8**	75.8	94.4	88.6	91.4	**39.1**	84.7	70.0	87.5	88.7	88.3	85.8	64.1	85.6	56.6	85.1	76.8
LRR-CRF	79.3	92.4	45.1	**94.6**	65.2	75.8	**95.1**	**89.1**	92.3	39.0	85.7	**70.4**	88.6	89.4	**88.6**	86.6	**65.8**	86.2	57.4	85.7	**77.3**

Figure 5.9: Per-class mean intersection-over-union (IoU) performance on PASCAL VOC 2012 segmentation challenge test data. We evaluate models trained using only VOC training data as well as those trained with additional training data from COCO. We also separate out a high-performing variant built on the ResNet-101 architecture.

5.5.4 CRF Post-processing

To show our architecture can easily be integrated with CRF-based models, we evaluated the use of our LRR model predictions as a unary potential in a fully-connected CRF [64, 18]. We resize each input image to three different scales (1,0.8,0.6), apply the LRR model and then compute the pixel-wise maximum of predicted class conditional probability maps. Post-processing with the CRF yields small additional gains in performance. Fig. 5.7 reports the mean IoU for our LRR-4x model prediction when running at multiple scales and with the integration of the CRF. Fusing multiple scales yields a noticeable improvement (between 1.1% to 2.5%) while the CRF gives an additional gain (between 0.9% to 1.4%).

	unmasked+DE	masked+DE
LRR-4x(32x)	64.7%	64.7%
LRR-4x(16x)	66.7%	67.1%
LRR-4x(8x)	68.5%	69.3%
LRR-4x	68.9%	70.0%

(a)

	IoU class	iIoU class	IoU cat	iIoU cat
FCN-8s [71]	65.3%	41.7%	85.7%	70.1%
CRF-RNN [113]	62.5%	34.4%	82.7%	66.0%
Dilation10 [105]	67.1%	42.0%	86.5%	71.1%
DPN [70]	66.8%	39.1%	86.0%	69.1%
Pixel-level Encoding [90]	64.3%	41.6%	85.9%	73.9%
DeepLab(ResNet) [20]	70.4%	42.6%	86.4%	67.7%
Adelaide_Context [68]	**71.6%**	**51.7%**	87.3%	74.1%
LRR-4x(VGG16)	69.7%	48.0%	**88.2%**	**74.7%**

(b)

Figure 5.10: (a) Mean intersection-over-union (IoU class) accuracy on Cityscapes validation set for intermediate outputs at different levels of our Laplacian reconstruction architecture trained with and without boundary masking. (b) Comparison of our model with state-of-the-art methods on the Cityscapes benchmark test set.

5.5.5 Benchmark Performance

PASCAL VOC Benchmark: As the Table 5.9 indicates, the current top performing models on PASCAL all use additional training data from the MS COCO dataset [69]. To compare our approach with these architectures, we also pre-trained versions of our model on MS COCO. We utilized the 20 categories in COCO that are also present in PASCAL VOC, treated annotated objects from other categories as background, and only used images where at least 0.02% of the image contained PASCAL classes. This resulted in 97765 out of 123287 images of COCO training and validation set.

Training was performed in two stages. In the first stage, we trained LRR-32x on VOC images and COCO images together. Since, COCO segmentation annotations are often coarser in comparison to VOC segmentation annotations, we did not use COCO images for training the LRR-4x. In the second stage, we used only PASCAL VOC images to further fine-tune the LRR-32x and then added in connections to the 16x, 8x and 4x layers and continue to fine-tune. We used the multi-scale data augmentation described in section 5.5.1 for both stages. Training on this additional data improved the mean IoU of our model from 74.6% to 77.5% on PASCAL VOC 2011 validation set (see Table 5.7).

Cityscapes Benchmark: The Cityscapes dataset [24] contains high quality pixel-level annotations of images collected in street scenes from 50 different cities. The training, vali-

dation, and test sets contain 2975, 500, and 1525 images respectively (we did not use coarse annotations). This dataset contains labels for 19 semantic classes belonging to 7 categories of ground, construction, object, nature, sky, human, and vehicle.

The images of Cityscapes are high resolution (1024×2048) which makes training challenging due to limited GPU memory. We trained our model on a random crops of size 1024×512. At test time, we split each image to 2 overlapping windows and combined the predicted class probability maps. We did not use any CRF post-processing on this dataset. Fig. 5.10 shows evaluation of our model built on VGG-16 on the validation and test data. It achieves competitive performance on the test data in comparison to the state-of-the-art methods, particularly on the category level benchmark. Examples of semantic segmentation results on the validation images are shown in Fig. 5.11

5.6 Conclusions

In this chapter, we have presented a system for semantic segmentation that utilizes two simple, extensible ideas: (1) sub-pixel upsampling using a class-specific reconstruction basis, (2) a multi-level Laplacian pyramid reconstruction architecture that uses multiplicative gating to more efficiently blend semantic-rich low-resolution feature map predictions with spatial detail from high-resolution feature maps. The resulting model is simple to train and achieves performance on PASCAL VOC 2012 test and Cityscapes that beats all but two recent models that involve considerably more elaborate architectures based on deep CRFs. We expect the relative simplicity and extensibility of our approach along with its strong performance will make it a ready candidate for further development or direct integration into more elaborate inference models.

Figure 5.11: Examples of semantic segmentation results on PASCAL VOC 2011 (top) and Cityscapes (bottom) validation images. For each row, we show the input image, ground-truth and the segmentation results of intermediate outputs of our LRR-4x model at the 32x, 16x and 8x layers. For the PASCAL dataset we also show segmentation results of FCN-8s [71].

Bibliography

[1] `http://vis-www.cs.umass.edu/fddb/results.html`.

[2] P. Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *CVPR*, pages 182–182, 2006.

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

[4] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *CVPR*, pages 3444–3451, 2013.

[5] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, pages 836–849. 2012.

[6] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, pages 545–552, 2011.

[7] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 25(9):1063–1074, 2003.

[8] E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, pages 109–122. 2002.

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

[10] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, pages 1124–1137, 2004.

[11] T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011.

[12] X. P. Burgos-Artizzu, P. Perona, and P. Dollár. Robust face landmark estimation under occlusion. In *ICCV*, pages 1513–1520, 2013.

[13] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[14] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression. In *CVPR*, pages 2887–2894, 2012.

[15] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 98(3):243–262, 2012.

[16] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7):1312–1328, 2012.

[17] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328, 2013.

[18] L.-C. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *CVPR*, 2016.

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

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

[21] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2015.

[22] X. Chen and A. Yuille. Parsing occluded people by flexible compositions. In *CVPR*, pages 3945–3954, 2015.

[23] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *PAMI*, 23(6):681–685, 2001.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[25] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. *arXiv preprint arXiv:1412.1283*, 2014.

[26] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015.

[27] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[28] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *CVPR*, pages 2578–2585, 2012.

[29] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494, 2015.

[30] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. *ECCV*, 2012.

[31] S. K. Divvala, A. A. Efros, and M. Hebert. How important are deformable parts in the deformable parts model? In *European Conference on Computer Vision (ECCV), Parts and Attributes Workshop*, 2012.

[32] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, pages 1841–1848, 2013.

[33] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards unified object detection and semantic segmentation. In *ECCV*, pages 299–314. 2014.

[34] B. Efraty, C. Huang, S. K. Shah, and I. A. Kakadiaris. Facial landmark detection in uncontrolled conditions. In *IJCB*, pages 1–8, 2011.

[35] M. Eichner and V. Ferrari. We are family: Joint pose estimation of multiple persons. In *ECCV*, pages 228–242. 2010.

[36] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, pages 98–136, 2015.

[37] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32:1627–45, 2010.

[38] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[40] Z. Feng, P. Huber, J. Kittler, B. Christmas, and X. Wu. Random cascaded-regression copse for robust facial landmark detection. *Signal Processing Letters*, 2015.

[41] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, page 264271, 2003.

[42] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, pages 3294–3301, 2013.

[43] T. Gao, B. Packer, and D. Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, pages 1361–1368, 2011.

[44] G. Ghiasi and C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, pages 1899–1906. 2014.

[45] G. Ghiasi and C. Fowlkes. Using segmentation to predict the absence of occluded parts. In *BMVC*, 2015.

[46] G. Ghiasi and C. Fowlkes. Laplacian reconstruction and refinement for semantic segmentation. In *ECCV*, pages 519–534, 2016.

[47] G. Ghiasi and C. C. Fowlkes. Occlusion coherence: Detecting and localizing occluded faces. In *arXiv preprint arXiv:1506.08347*, 2015.

[48] G. Ghiasi, Y. Yang, D. Ramanan, and C. Fowlkes. Parsing occluded people. In *CVPR*, pages 2401–2408, 2014.

[49] S. Gidaris and N. Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*, pages 1134–1142, 2015.

[50] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[51] R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester. Object detection with grammar models. In *NIPS*, pages 442–450, 2011.

[52] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *IVC*, 28(5):807–813, 2010.

[53] L. Gu and T. Kanade. 3d alignment of face in a single image. In *CVPR*, pages 1305–1312, 2006.

[54] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011.

[55] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, pages 447–456, 2015.

[56] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[57] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[58] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*, pages 3146–3153, 2012.

[59] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. Technical report, Institute of Mathematics of the Romanian Academy and University of Bonn, September 2012.

[60] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[61] X. Jia, H. Yang, A. Lin, K.-P. Chan, and I. Patras. Structured semi-supervised forest for facial landmarks localization with face mask reasoning. In *BMVC*, 2014.

[62] P. Kohli, L. Ladicky, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009.

[63] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. In *ICLR*, 2016.

[64] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[65] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, pages 725–739. 2014.

[66] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *ECCV*, pages 679–692. 2012.

[67] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, volume 2, pages II–97, 2004.

[68] G. Lin, C. Shen, A. v. d. Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016.

[69] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[70] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *ICCV*, pages 1377–1385, 2015.

[71] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[72] A. Martinez and S. Du. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *JMLR*, 13:1589–1608, 2012.

[73] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.

[74] S. Milborrow and F. Nicolls. Locating facial features with an extended active shape model. In *ECCV*, pages 504–513. 2008.

[75] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, pages 3376–3385, 2015.

[76] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.

[77] W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. CVPR, 2013.

[78] M. Pantic, G. Tzimiropoulos, and S. Zafeiriou. 300 faces in-the-wild challenge (300-w). In *ICCV Workshop*, pages 397–403, 2013.

[79] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *ECCV*, pages 241–254. 2010.

[80] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016.

[81] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *CVPR*, pages 3178–3185, 2012.

[82] S. Ren, X. Cao, Y. Wei, and J. Sun. Face alignment at 3000 fps via regressing local binary features. In *CVPR*, pages 1685–1692, 2014.

[83] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG*, 23(3):309–314, 2004.

[84] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, pages 1745–1752, 2011.

[85] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable model fitting by regularized landmark mean-shift. *IJCV*, 91(2):200–215, 2011.

[86] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *CVPR*, pages 750–757, 2003.

[87] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1):2–23, 2009.

[88] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[89] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. In *BMVC*, pages 1–11, 2012.

[90] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. *arXiv preprint arXiv:1604.05096*, 2016.

[91] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, pages 2729–2736, 2010.

[92] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *ICML*, 2015.

[93] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial occulsion. In *NIPS*. Citeseer, 2009.

[94] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *CVPR*, pages 32–39, 2009.

[95] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR*, pages 101–108, 2000.

[96] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von Der Malsburg. Face recognition by elastic bunch graph matching. *PAMI*, 19(7):775–779, 1997.

[97] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015.

[98] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013.

[99] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *CVPR*, pages 2497–2504, 2014.

[100] H. Yang, X. He, X. Jia, and I. Patras. Robust face alignment under occlusion via regional predictive power estimation. *Image Processing*, 2015.

[101] S. Yang and D. Ramanan. Multi-scale recognition with dag-cnns. In *ICCV*, pages 1215–1223, 2015.

[102] Y. Yang, S. Baker, A. Kannan, and D. Ramanan. Recognizing proxemics in personal photos. In *CVPR*, pages 3522–3529, 2012.

[103] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation. *PAMI*, 34(9):1731–1743, 2012.

[104] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures-of-parts. *PAMI*, 35(12):2878–2890, 2013.

[105] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[106] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas. Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In *ICCV*, pages 1944–1951, 2013.

[107] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *IJCV*, 8(2):99–111, 1992.

[108] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[109] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, pages 2018–2025, 2011.

[110] J. Zhang, S. Shan, M. Kan, and X. Chen. Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In *ECCV*, pages 1–16, 2014.

[111] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Learning deep representation for face alignment with auxiliary attributes. *PAMI*, 38(5):918–930, 2016.

[112] C. C. L. Zhanpeng Zhang, Ping Luo. Learning deep representation for face alignment with auxiliary attributes. *arXiv:1408.3967v2*, 2015.

[113] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015.

[114] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *PAMI*, 32(6):1029–1043, 2010.

[115] S. Zhu, C. Li, C. Change Loy, and X. Tang. Face alignment by coarse-to-fine shape searching. In *CVPR*, pages 4998–5006, 2015.

[116] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886, 2012.

[117] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012.