Using Segmentation to Predict the Absence of Occluded Parts

Golnaz Ghiasi
gghiasi@ics.uci.edu
Charless C. Fowlkes
fowlkes@ics.uci.edu

Dept. of Computer Science
University of California
Irvine, CA

Abstract

Occlusion poses a significant difficulty for detecting and localizing object keypoints and subsequent fine-grained identification. 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.

1 Introduction

Occlusion, arising from interactions between objects in cluttered scenes or between different parts of a single articulated object, creates drastic changes in local visual appearance. This poses a significant barrier to correctly detecting and performing fine-grained analysis of objects in images. Indeed, detection performance for cluttered images lags behind object classification accuracy for images depicting a single object [21, 24].

Occlusion is naturally represented in object detection models that reason about locations of parts (e.g., [3, 17, 25]). We focus on the problem of detecting and localizing faces where the parts in consideration are represented by facial keypoints. Recent work has shown that pose-regression techniques [6] or deformable part models [14] that include explicit states representing part occlusion can yield better detection and localization accuracy.

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. 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.

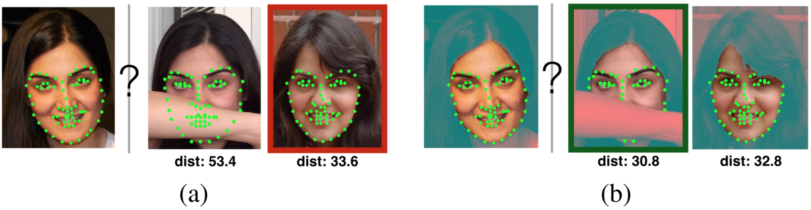


Figure 1: (a) Comparing part appearance for all facial keypoints results in incorrect identification of query image. We propose a model that leverages segmentation to identify which keypoints are occluded, allowing downstream facial analysis to focus on visible regions of the face, yielding the correct 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.

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 segmentation. Highlights include early work on generating consistent object masks during detection [4], producing semantic segmentations of scenes driven by object detector responses (e.g., [13, 27]) and most recently the use of bottom-up segments as proposals for scoring object detectors (e.g., [12, 16]). Our approach is most closely related to GRABCUT [23] and more recent works such as [7, 9, 12] that enforce mutual consistency between detection and segmentation.

The contribution of this paper 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 [9, 12]), 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.

2 A segmentation aware part model

Figure 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. 2.1-2.2) and then return to the problem of inferring segmentations in Sec. 2.3.

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 [14]. Parts of the face (e.g. nose, eyes, eyebrows) are connected to each other using a tree structure

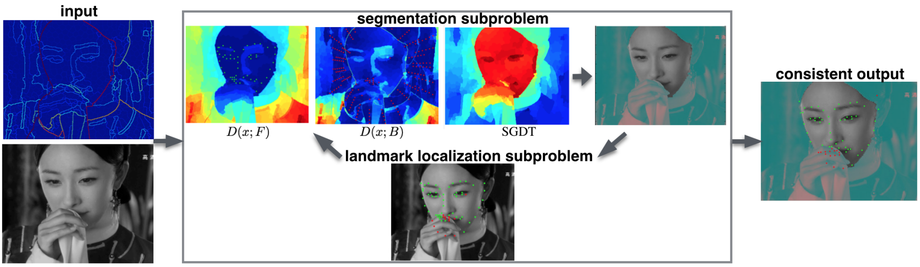


Figure 2: Overall landmark localization and face mask prediction pipeline. First column: input image and a computed superpixel image and boundary weights between them. Our method alternates between landmark localization subproblem and segmentation subproblem.

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. 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:

$$\begin{aligned} S(l,s,o|I,Z) = & \sum_i \alpha_i^{s_i} \cdot \phi_{\text{App}}(l_i, o_i, I) + \sum_i \sum_{j \in child(i)} \beta_{ij}^{s_i,s_j} \cdot \phi_{\text{Shape}}(l_i - l_j) + b_{ij}^{s_i,s_j,o_i,o_j} \\ & + \sum_i \gamma_i^{o_i} \cdot \phi_{\text{Seg}}(l_i, Z) + b_{seg}[Z = \emptyset] \end{aligned} \quad (1)$$

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 $\phi_{\text{Seg}}(l_i, \emptyset) = 0$.

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 [10]. The potentials in our model are linearly parameterized so we can write our scoring function as an inner product

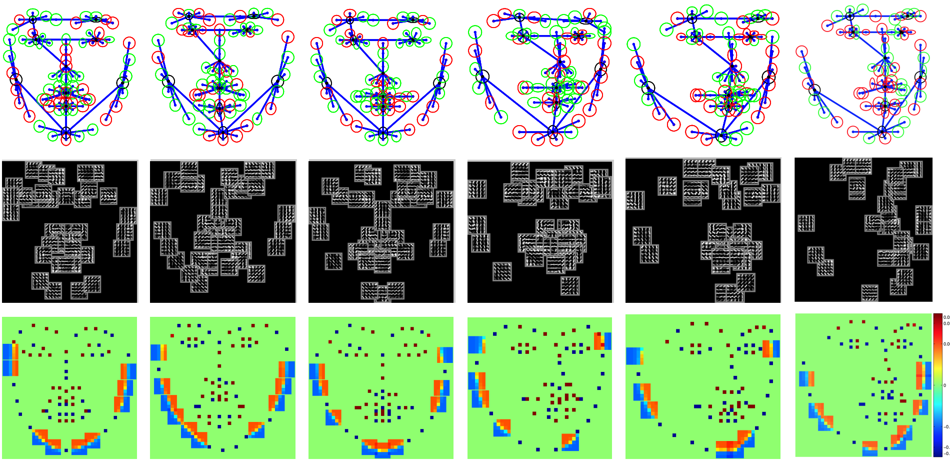


Figure 3: Our model has similar structure as HPM [14]. But, unlike HPM we do not restrict the possible occlusion patterns and each keypoint can be visible (green) or occluded (red) independent of the other keypoints. The examples here show templates corresponding to different choices of part mixtures. 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 [10]. 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 [15]. 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.

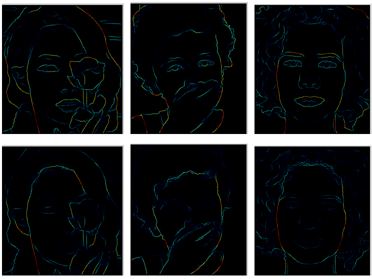
Positive training data We train the model using positive training examples in which all the variables (l,s,o) are fully observed. To learn shape mixtures, we use a similar strategy to [14] and cluster the set of keypoint configurations of each part over the training set to come up with a small number 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 $[\phi_{\text{App}}^T, \phi_{\text{Def}}^T, \phi_{\text{bias}}^T, 1, 0]^T$. The second type of training data include segmentation features and the feature vector is $[\phi_{\text{App}}^T, \phi_{\text{Def}}^T, \phi_{\text{bias}}^T, 0, \phi_{\text{Seg}}^T]^T$. 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

	ODS	AP
Generic BD (BSDS)[8]	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: (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).

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 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.

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.

2.3.1 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 (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 [8] 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(b)).

To learn a class-specific boundary detector it is useful to distinguish the figure-ground orientation 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 [8] 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 (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.

2.3.2 Face segmentation via graph cuts

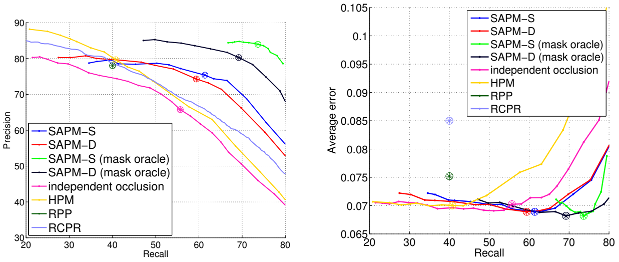
To convert boundary detection into a segmentation, we use the available code of [8] to compute Ultrametric Contour Map (UCM) of boundary image [1]. This provides a set of superpixels and the boundary weights between them. We then use binary graph cut [5] to partition these superpixels into foreground 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

$$S(Z|l,o,I) = \sum_i \hat{\gamma}(l,o)_i Z_i + \lambda \sum_{i,j} \omega_{i,j} [Z_i = Z_j] \tag{2}$$

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. 3) and averaged over the support of superpixel i.

	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 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.

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. 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) [19] 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. 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. 1 + the pairwise graphcut potential in Eqn. 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.

	ave err	FR	F-meaure(P/R)	ave overlap	global	ave(face)
TCDCN[28]	0.080	-	-	-	-	-
R-CR-C[11]	0.073	12.2%	-	-	-	-
DPM[29]	0.079	16.80%	-	-	-	-
structured forest [18]	-	-	[66.56 (80%/57%)]	-	[83.9%]	[88.6%]
RPP[26]	0.075	16.2%	52.88 (78%/40%)	0.724	-	-
RCPR[6]	0.085	20%	53.33 (80%/40%)	-	-	-
HPM[15]	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[6]	0.056	0.01%	-	-	-	-

Table 1: Comparison of landmark localization, occlusion prediction and mask prediction between our model and previous results on the COFW test data [6]. 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 ([18, 26]). [numbers] reported for a subset of image (300 of 507).

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) [6] 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 [18]. Figure 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 grayscale 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 downsampled to match the gray-scale resolution.

Model Training Details To train our model, we used a set of 1758 near-frontal training images taken from the HELEN [20] training set using the 68 keypoint annotations provided by 300-W [22]. 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 [14]. 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.

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 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 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. 5 parts (b) and (c). Previous 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.

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 [2] and average precision (AP) for the evaluation. Results for different detectors are shown in Fig. 4 (a). The first row of the table shows the results for the generic boundary detector of [8]. 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 2.3.1) 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 1 show a comparison between accuracy of our method and previous results. To compare our method with RPP [26], 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 [18], we calculate global and ave(face) metrics which show the percentage of all pixels that are correctly classified and the average

[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.

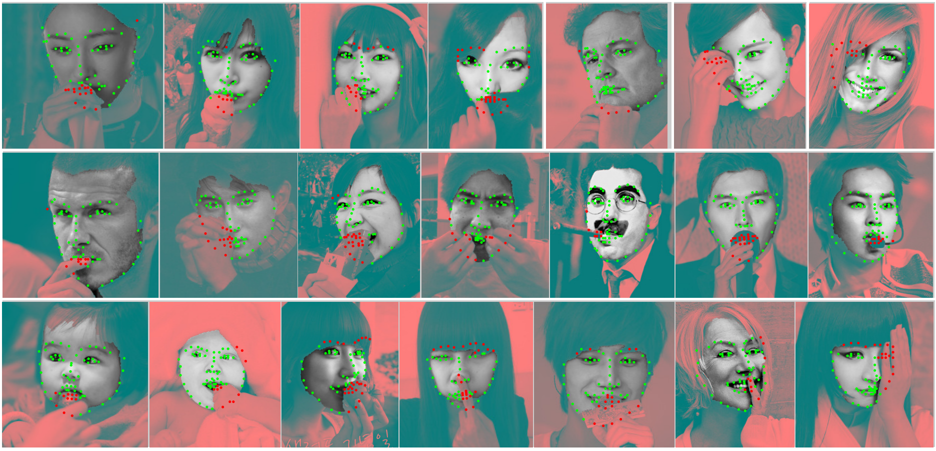


Figure 6: Examples of landmark localization and mask estimation for images from the COFW test data produced by our model.

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 [18] are over a random subset of 300 images (out of 507) of COFW test data.

4 Conclusion

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 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.

Acknowedgements This work was supported by the US National Science Foundation through award IIS-1253538.

References

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

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33(5):898–916, 2011.

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

[4] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV*, pages 109–122. 2002.

[5] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*.

[6] Xavier P Burgos-Artizzu, Pietro Perona, and Piotr Dollár. Robust face landmark estimation under occlusion. In *ICCV*, 2013.

[7] Yuning Chai, Victor Lempitsky, and Andrew Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, pages 321–328, 2013.

[8] Piotr Dollár and C Lawrence Zitnick. Structured forests for fast edge detection. In *ICCV*, pages 1841–1848, 2013.

[9] Jian Dong, Qiang Chen, Shuicheng Yan, and Alan Yuille. Towards unified object detection and semantic segmentation. In *ECCV*, pages 299–314. 2014.

[10] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

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

[12] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, pages 3294–3301, 2013.

[13] Tianshi Gao, Benjamin Packer, and Daphne Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, pages 1361–1368, 2011.

[14] Golnaz Ghiasi and Charless C. Fowlkes. Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model. In *CVPR*, pages 1899–1096, 2014.

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

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[17] Ross B Girshick, Pedro F Felzenszwalb, and David A Mcallester. Object detection with grammar models. In *NIPS*, pages 442–450, 2011.

[18] Xuhui Jia, Heng Yang, Angran Lin, Kwok-Ping Chan, and Ioannis Patras. Structured semi-supervised forest for facial landmarks localization with face mask reasoning. In *BMVC*, 2014.

[19] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *ECCV*, pages 725–739. 2014.

[20] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir Bourdev, and Thomas S Huang. Interactive facial feature localization. In *ECCV*, pages 679–692. 2012.

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. 2014.

[22] Maja Pantic, Georgios Tzimiropoulos, and Stefanos Zafeiriou. 300 faces in-the-wild challenge (300-w). In *ICCV Workshop*, 2013.

[23] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG*, 23(3):309–314, 2004.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[25] Markus Weber, Max Welling, and Pietro Perona. Towards automatic discovery of object categories. In *CVPR*, volume 2, pages 101–108, 2000.

[26] Heng Yang, Xuming He, Xuhui Jia, and Ioannis Patras. Robust face alignment under occlusion via regional predictive power estimation. *Image Processing*, 2015.

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

[28] Chen Change Loy Zhanpeng Zhang, Ping Luo. Learning deep representation for face alignment with auxiliary attributes. *arXiv:1408.3967v2*, 2015.

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