Object Detection with Partial Occlusion Based on a Deformable Parts-Based Model

Johnson Hsieh (johnsonhsieh@gmail.com), Alexander Chia (alexchia@stanford.edu)

Abstract -- Object occlusion presents a major challenge for robust object detection in static images. We describe an object detection system that explicitly models and accounts for arbitrary but consistent occlusion patterns. Our model builds on the state-of-the-art object detection system based on a deformable parts-based model. We've augmented this model with latent binary visibility variables for each pixel, as well as pairwise consistency visibility potentials. We will show an efficient inference algorithm for matching our visibility model to a test image during detection. In training, we employ a latent SVM learning framework, as well as reusing parts of the inference algorithm that we've developed for matching. Our system is trained and tested on the PASCAL object detection challenge dataset.

1 INTRODUCTION

Occlusion is a common problem in real world images (and videos) and presents a major challenge for object detection. In the Caltech Pedestrian Dataset [1], for example, over 70% of pedestrians are occluded in at least one frame of a video sequence and 19% are occluded in all frames, where the occlusion was categorized as heavy in nearly half of these cases. Dollar et al. [1] showed that detection performance drops significantly even under partial occlusion, and drastically under heavy occlusion. If only fully visible examples are used during training, then many positive examples are simply discarded and the training distribution does not match the test distribution. If partially occluded examples are included, this may lead to tolerance of occlusion in the test set, but the trained model will be more noisy and less robust.

The current state-of-the-art object detection system built by Felzenszwalb et al. represents highly variable objects using mixtures of multiscale deformable part models [2]. Our system builds on top of this model and augments it with latent binary visibility variable for each pixel, as well as pairwise consistency visibility potentials. This augmented object detection system will allow us to explicitly model and account for arbitrary but consistent occlusion patterns

2 BACKGROUND AND RELATED WORK

There have been previous attempts to model occlusion for the object detection task [3,4]. The work done by Vedaldi and Zisserman [5] is the most similar to ours. They propose using binary variables to indicate the visibility of the cells inside a detection window. However, rather than inferring the values of those variables, they treat them as a deterministic function of the position of the bounding boxes, i.e., when the bounding box is partially outside of the image, those variables corresponding to the cells that are outside of the image will be set to invisible. Occlusion is therefore modeled only on image borders, and occlusion within the image frame is ignored.

3 OBJECT MODELS

The Felzenszwalb system is the current state-of-the-art object detection system. In order to see the benefits of modeling latent visibility features, we will analyze three simple baseline models to illustrate the headroom for which we will work towards.

In each of the three baseline models, we used some predetermined visibility mask for each training and test image. The first model uses the ground truth visibility masks, which we obtained through Amazon Mechanical Turk. The second and third models use a precomputed segmentation library to obtain the masks, which we will describe in detail later. Given these predetermined masks, we build a model using the masked training examples, and also use the masks in the testing

phase. We would like to see whether we could actually do better than Felzenszwalb's model given that we now have prior knowledge of the object visibility in each image. We're using predetermined masks for now, so that we don't have to worry about inferring the masks yet, in either training or testing.

Given the pixel-level visibility mask, we convert it into a cell-level visibility mask and apply it to the (cell-level) HOG features [6]. We also take the complement of the mask and apply it to the HOG features to get a set of dual masks. We have done a preliminary study which showed some object classes, such as the 'bicycle' class, gained significant performance improvements from using the positive masked HOG features, while other classes, like the 'horse' class, experienced significant performance improvements from using the inverse masked HOG features (see Fig. 5). Thus we hope that by using a set of dual masked HOG features, we can compensate for this variability and achieve reasonable performance across all object classes.

In order to convert from the pixel-level visibility mask to the cell-level visibility mask, we used the fraction of visible pixels in each cell as the cell-level visibility. Note that our given pixel-level visibility mask is a binary mask, but the cell-level visibility mask is now a weighted mask taking on values between zero to one for each cell. To improve accuracy, we expand the pixel mask by 5 pixels and performed a Gaussian blur before converting it to a cell-level mask. This helps to avoid artificial sharp edges, which could be undesirably learned. The result is that both masks in the dual keeps the edge features of the object, but the positive mask significantly downweights the background features while the inverse mask significantly downweights the interior features of the object.

Formally, Felzenszwalb's HOG features are defined as $\phi(H,p_i) = [h_1^{(i)},...,h_c^{(i)}]$ where H has c cells and $h_k^{(i)}$ is the HOG feature vector of the k^{th} cell in the part p_i.

In our model we modify the HOG features to be

$$\begin{aligned}\phi(H,p_i,v) = \phi(H,p_i,\tilde{v}^{(i)} \mid v) &= \left[\tilde{v}_1^{(i)}h_1^{(i)},...,\tilde{v}_c^{(i)}h_c^{(i)},(1-\tilde{v}_1^{(i)})h_1^{(i)},...,(1-\tilde{v}_c^{(i)})h_c^{(i)}\right] \\ &= \left[\phi_+(H,p_i,\tilde{v}^{(i)} \mid v),\phi_-(H,p_i,\tilde{v}^{(i)} \mid v)\right]\end{aligned}$$

where v is the set of binary pixel-level visibility variables, and $\tilde{v}_k^{(i)}$ is the weighted cell-level visibility variable for the k^{th} cell in the part p_i. Note that $\tilde{v}_k^{(i)}h_k^{(i)}$ corresponds to the positive masked HOG features, and $(1-\tilde{v}_k^{(i)})h_k^{(i)}$ corresponds to the inverse masked HOG features.

3.1 Matching

Once we've modified the HOG features with the visibility information, we run the Felzenszwalb training procedure, which performs optimization using a coordinate descent approach, and employs a latent SVM implemented by stochastic gradient descent.

However, because we've modified the feature vector by incorporating a set of dual masked features, the matching process becomes more complicated. Specifically, how do we score a match to find the best match? In the Felzenszwalb model, the set of filter weights F_i is learnt for each part p_i. The score for the best root and part location match (not accounting for part deformation) is then

$$score(p_0,...,p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H,p_i)$$

In our model, we learn two sets of filter weights, one set for the positive masked features, the other for the inverse masked features. But to calculate the score, we have various options:

Option #1 (sum):
$$score(p_0,...,p_n) = \left[\sum_{i=0}^{n} F_{+i} \cdot \phi_+(H,p_i,\tilde{v}^{(i)} \mid v)\right] + \left[\sum_{i=0}^{n} F_{-i} \cdot \phi_-(H,p_i,\tilde{v}^{(i)} \mid v)\right]$$

Option #2 (overall max):
$$score(p_0,...,p_n) = \max\left(\sum_{i=0}^{n} F_{+i} \cdot \phi_+(H,p_i,\tilde{v}^{(i)} \mid v), \sum_{i=0}^{n} F_{-i} \cdot \phi_-(H,p_i,\tilde{v}^{(i)} \mid v)\right)$$

Option #3 (max per part): $$score(p_0,...,p_n) = \sum_{i=0}^{n} \max\left(F_{+i} \cdot \phi_+(H,p_i,\tilde{v}^{(i)} \mid v), F_{-i} \cdot \phi_-(H,p_i,\tilde{v}^{(i)} \mid v)\right)$$

In our experiments, we found that taking the overall max (option #2) gave best results.

3.2 Training

Again, we notice that the new dual masked features complicate the training procedure. We can reuse the matching logic as described above, but we now have further complications with learning the latent object parts. We have several options: 1) learn two sets of part anchors and deformation costs for each of the positive and inverse masked features; 2) restrict the anchors to be the same, but learn two sets of part deformation costs; 3) learn only one set of part anchors and deformation costs, to be shared by the two sets of masked features in the dual. Due to time constraints, we only explored the first option. Learning one object class over the PASCAL VOC2007 data set took ~6 hours on a 2.8Ghz 6-core Intel Xeon Processor X5650 with 12 GB of memory running Ubuntu Lucid.

3.2 Masks

As mentioned above, we obtained three sets of visibility masks. The first set is obtained via human annotation through Amazon Mechanical Turk (AMT). The second and third sets are obtained through an image segmentation library written by Pawan Kumar, who works in Prof. Daphne Koller's research group. This segmentation library is trained on the VOC2007 data set, and given a test image, returns an output image with segmented regions, with each region labeled with a predicted object class. We use this output in two ways: 1) take the regions with our desired class label as the visibility mask; 2) include all regions within the desired object's bounding box, allowing at most 10% of each region to be outside the bounding box. Figure 1 illustrates an example of these 3 masks.

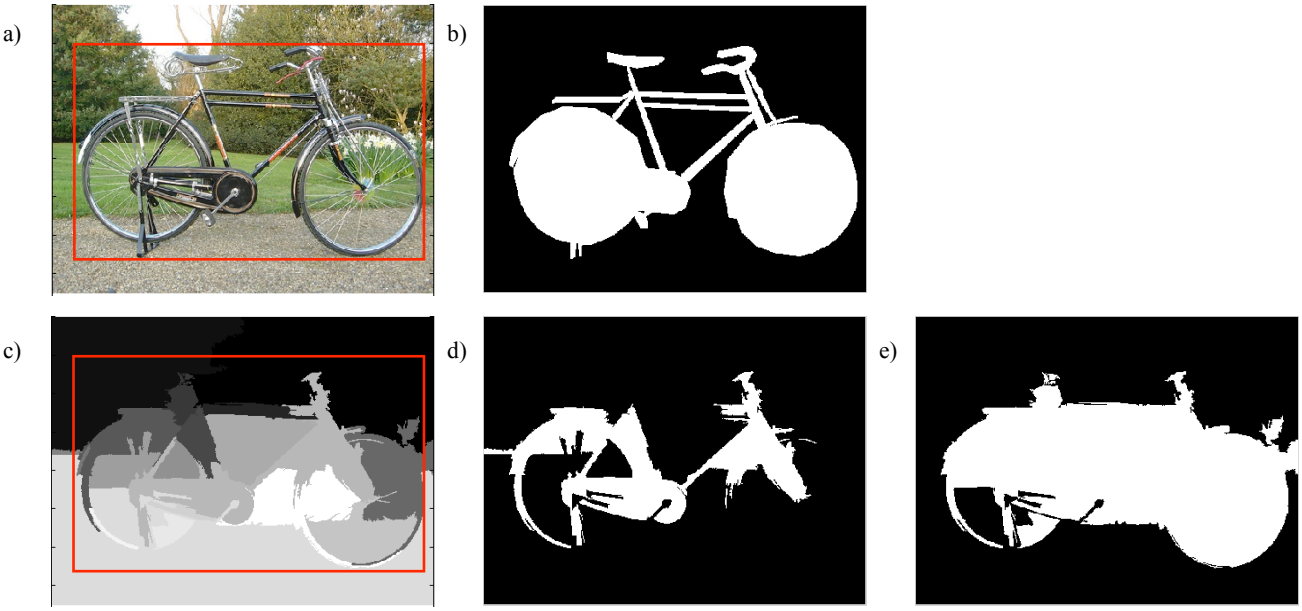


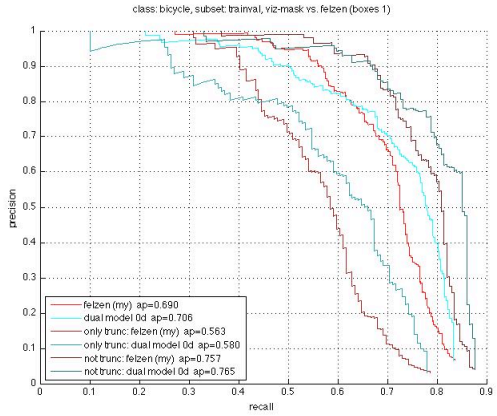
Fig. 1. a) Original image, b) ground truth visibility mask obtained from AMT, c) image segmentation regions, d) segmented regions labeled as bicycle class, e) union of segmented regions with 90% of each region inside the bounding box.

4 RESULTS

In order to evaluate our models, we plotted the precision-recall curves and compared the average precision scores. We also broke down the evaluation and analysis into occluded vs. non-occluded image sets; the VOC2007 dataset included annotations for each object instance indicating whether or not they were occluded (VOC called them "truncated").

Figure 2 shows the PR curves for the bicycle and horse models generated from the ground truth masks. We see that overall, both sets improved decently. More importantly, the occluded subset improved significantly, especially for the horse class. This looks very promising. In Figures 3 and 4, we show the PR curves for models generated from the two precomputed segmentation outputs.

a)



b)

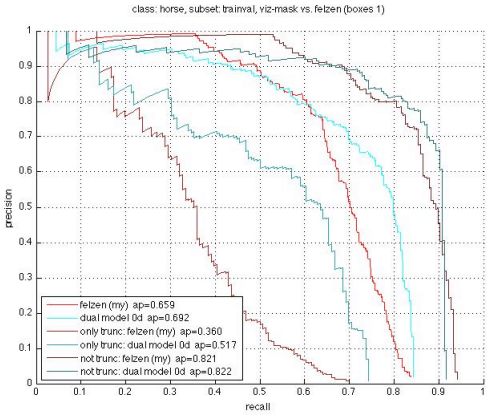
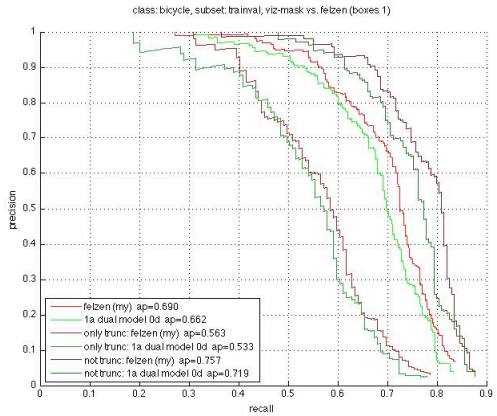


Fig. 2. Bicycle and Horse models generated from the ground truth masks. We breakdown the image set into an occluded subset and a non-occluded subset. a) bicycle, b) horse.

a)



b)

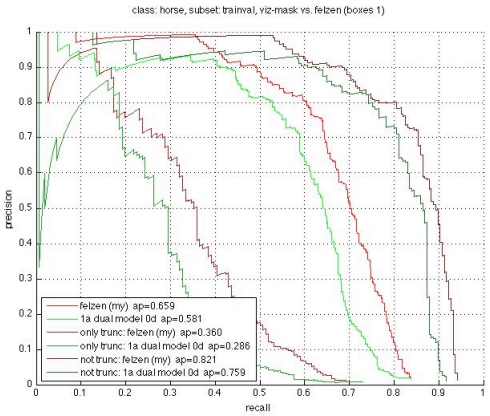
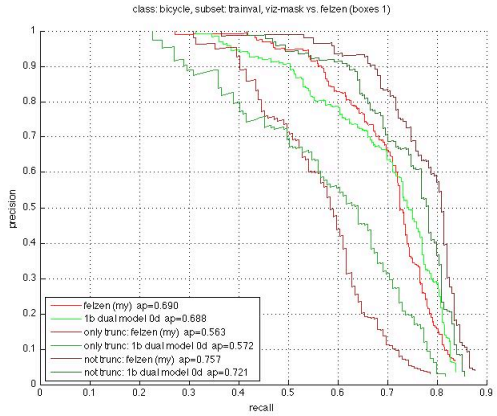


Fig. 3. Bicycle and Horse models generated from the precomputed segmentation masks, using segmented regions with the desired class label as the visibility mask. We breakdown the image set into an occluded subset and a non-occluded subset. a) bicycle, b) horse.

a)



b)

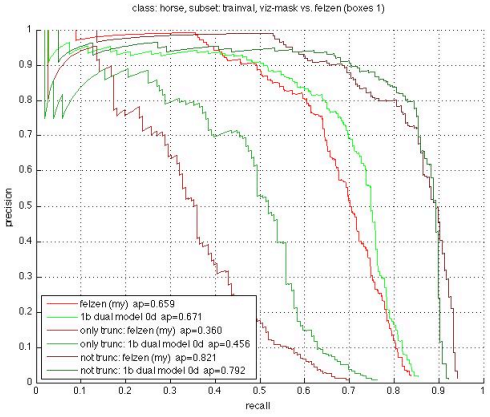


Fig. 4. Bicycle and Horse models generated from the precomputed segmentation masks, using all segmented regions that are 90% within the bounding box. We breakdown the image set into an occluded subset and a non-occluded subset. a) bicycle, b) horse.

The first segmentation model, which used the segmented region class labels, performed quite poorly for both the bicycle and horse object classes. The second segmentation model, which used the regions within the bounding box as the visibility mask, performed much better. We see that the occluded subsets received the biggest gains, and the occluded

subset for the horse class improved the most, by almost 0.1 average precision (AP) points. The overall average precision for the bicycle class stayed about the same compared to Felzenszwalb's results, and for the horse class, we see a small 0.012 gain in AP. However, if our data set included many more occluded images, the overall performance would increase, as shown by our improved detection performance on the occluded subset.

Figure 5 shows our other experimental results. Before coming up with the dual mask model, we investigated a positive-only visibility mask model and an inverse-only visibility mask model. The positive-only mask model gave the bicycle class a slight increase in performance, but performed poorly on the horse class. The inverse-only mask model yielded the exact opposite performance for the two classes. This suggests that different parts of the image contain valuable features for different object classes, and gave us the motivation to explore a dual mask model.

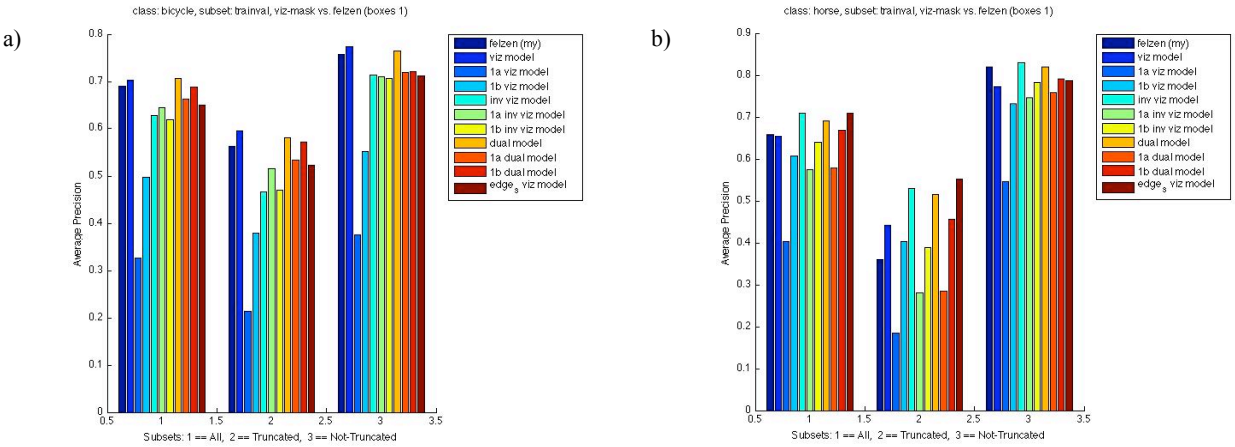


Fig. 5. Other experimental models we built for the bicycle and horse classes. a) bicycle, b) horse.

In Figure 6, we show an example of the bicycle models that we trained. Note that for the visibility part of the dual model, the background regions are significantly downweighted, while for the inverse visibility part of the dual model, the interior regions of the bicycle features are downweighted. Also note that both parts of the dual keep the edge features fully. This is a consequence of how we applied the visibility and inverse visibility masks to the HOG features and is described previously in section 3.

5 CONCLUSIONS AND FUTURE WORK

Our research showed that there is promise in building an object detection model on top of the state-of-the-art object detector by augmenting it with explicit visibility variables. Our results showed that, given decent pixel-level visibility masks, detection performance improves significantly for the occluded subset, and slightly for the overall test set.
In this quarter, we did not have enough time to continue on to implement the inference algorithms that would determine the visibility masks automatically at training and testing. We have worked out most of the math for this inference procedure, which includes pairwise consistency visibility potentials as mentioned in the abstract. In the future, we aim to implement this algorithm and analyze the performance results. In section 3.2 above, we mentioned various other ways to learn the dual feature sets especially for the latent object parts. Part of our future work would also be to explore these options in detail and hopefully find a method that exceeds our current performance.

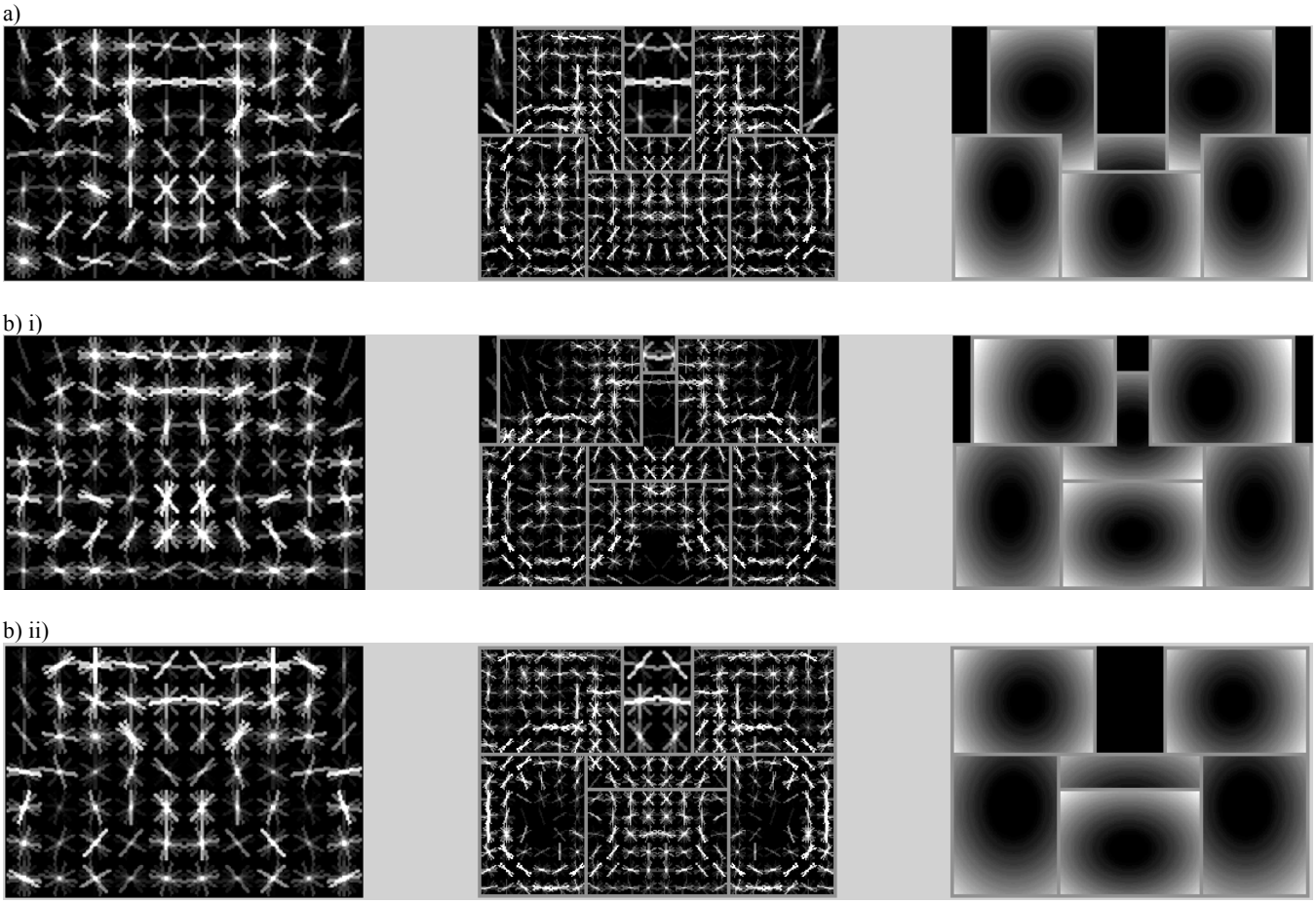


Fig. 6. Bicycle models: a) Felzenszwalb's model; b) Our dual masked model: i) the visibility part of the dual model, ii) the inverse visibility part of the dual model.

6 ACKNOWLEDGEMENTS

The research presented in this paper was work done in conjunction for the CS294A Autumn 2010 class. We worked with Tianshi Gao and Prof. Daphne Koller throughout the quarter, and have received much valuable advice and feedback from them. We also thank Pawan Kumar for his kind permission for the use of his image segmentation library.

References

[1] P. Dollar, C. Wojek, B.S., P. Perona, "Perdestrian detection: A benchmark". In CVPR. (2009)

[2] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models". In PAMI. (2009)

[3] J. Winn, J. Shotton, "The layout consistent random field for recognizing and segmenting partially occluded objects". In CVPR. (2006)

[4] X. Wang, T. Han, S. Yan, "An hog-lbp human detector with partial occlusion handling". In ICCV. (2009)

[5] A. Vedaldi, A. Zisserman, "Structured output regression for detection with partial truncation." In NIPS (2009)

[6] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection". In CVPR. (2005)