# Finding Tiny Faces
## Supplementary Materials

Peiyun Hu, Deva Ramanan
Robotics Institute
Carnegie Mellon University
{peiyunh,deva}@cs.cmu.edu

## 1. Error analysis

**Quantitative analysis** We plot the distribution of error modes among false positives in Fig. 1 and the impact of object characteristics on detection performance in Fig. 2 and Fig. 3.

**Qualitative analysis** We show top 20 scoring false positives in Fig. 4.

## 2. Experimental details

**Multi-scale features** Inspired by the way [3] trains "FCN-8s at-once", we scale the learning rate of predictor built on top of each layer by a fixed constant. Specifically, we use a scaling factor of 1 for res4, 0.1 for res3, and 0.01 for res2. One more difference between our model and [3] is that: instead of predicting at original resolution, our model predicts at the resolution of res3 feature (downsampled by 8X comparing to input resolution).

**Input sampling** We first randomly re-scale the input image by 0.5X, 1X, or 2X. Then we randomly crop a 500x500 image region out of the re-scaled input. We pad with average RGB value (prior to average subtraction) when cropping outside image boundary.

**Border cases** Similar to [2], we ignore gradients coming from heatmap locations whose detection windows cross the image boundary. The only difference is, we treat padded average pixels (as described in **Input sampling**) as outside image boundary as well.

**Online hard mining and balanced sampling** We apply hard mining on both positive and negative examples. Our implementation is simpler yet still effective comparing to [4]. We set a small threshold (0.03) on classification loss to filter out easy locations. Then we sample at most 128 locations for both positive and negative (respectively) from remaining ones whose losses are above the threshold. We compare training with and without hard mining on validation performance in Table 1.

**Loss function** Our loss function is formulated in the same way as [2]. Note that we also use Huber loss as the loss function for bounding box regression.

**Bounding box regression** Our bounding box regression is formulated as [2] and trained jointly with classification using stochastic gradient descent. We compare between testing with and without regression in terms of performance on WIDER FACE validation set.

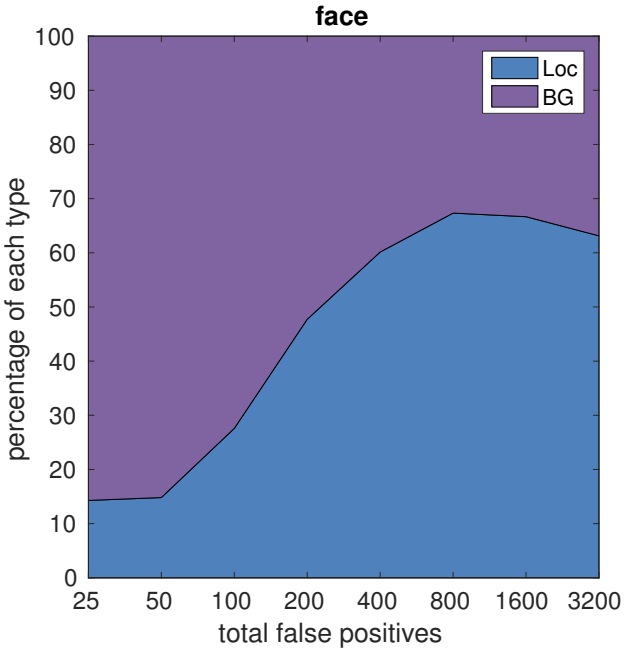


Figure 1: Distribution of error modes of false positives. Background confusion seems the dominating error mode among top-scoring detection, however, we found 15 out of 20 top-scoring false positives, as shown in Fig. 4, are in fact due to missed annotation.

| Method | Easy | Medium | Hard |
|---|---|---|---|
| w/ hard mining | **0.919** | **0.908** | 0.822 |
| w/o hard mining | 0.917 | 0.904 | **0.825** |

Table 1: Comparison between training with and without hard mining. We show performance on WIDER FACE validation set. Both models are trained with balanced sampling and use ResNet-101 architecture. Results suggest hard mining has no noticeable affect the final performance.

| Method | Easy | Medium | Hard |
|---|---|---|---|
| w/ regression | **0.919** | **0.908** | **0.823** |
| w/o regression | 0.911 | 0.900 | 0.798 |

Table 2: Comparison between testing with and without regression. We show performance on WIDER FACE validation set. Both models use ResNet-101 architecture. Results suggest that regression helps slightly more on detecting small faces (2.4%).

**Bounding ellipse regression** Our bounding ellipse regression is formulated as Eq. (1).

$$t^*_{x_c} = (x^*_c - x_c)/w \tag{1}$$
$$t^*_{y_c} = (y^*_c - y_c)/h \tag{2}$$
$$t^*_{r_a} = \log(r^*_a/(h/2)) \tag{3}$$
$$t^*_{r_b} = \log(r^*_b/(w/2)) \tag{4}$$
$$t^*_{\theta} = \cot(\theta^*) \tag{5}$$
$$\tag{6}$$

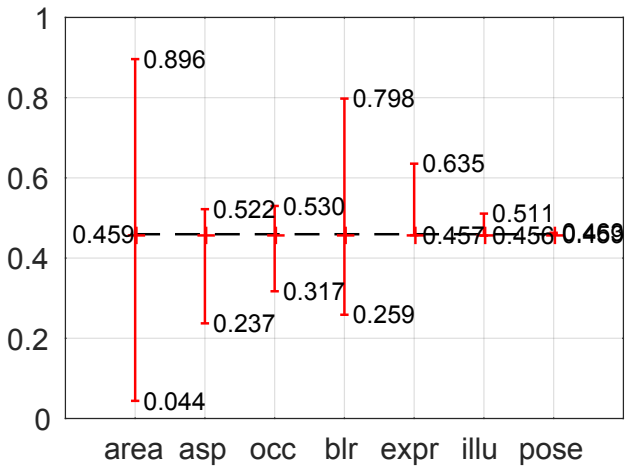


Figure 2: Summary of sensitivity plot. We plot the maximum and minimum of $AP_N$ shown in Figure 3. Our detector is mostly affected by object scale (from 0.044 to 0.896) and blur (from 0.259 to 0.798).

where $x_c^*, y_c^*, r_a^*, r_b^*, \theta^*$ represent center x-,y-coordinate, ground truth half axes, and rotation angle of the ground truth ellipse. $x_c, y_c, h, w$ represent the center x-,y-coordinate, height, and width of our predicted bounding box. We learn the bounding ellipse linear regression offline, with the same feature used for training bounding box regression.

**Other hyper-parameters** We use a fixed learning rate of $10^{-4}$, a weight decay of 0.0005, and a momentum of 0.9. We use a batch size of 20 images, and randomly crop one 500x500 region from the re-scaled version of each image. In general, we train models for 50 epochs and then select the best-performing epoch on validation set.

## References


[1] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *European conference on computer vision*, pages 340–353. Springer, 2012. 4

[2] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[3] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. 1

[4] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. *arXiv preprint arXiv:1604.03540*, 2016. 1

[5] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, June 2016. 4

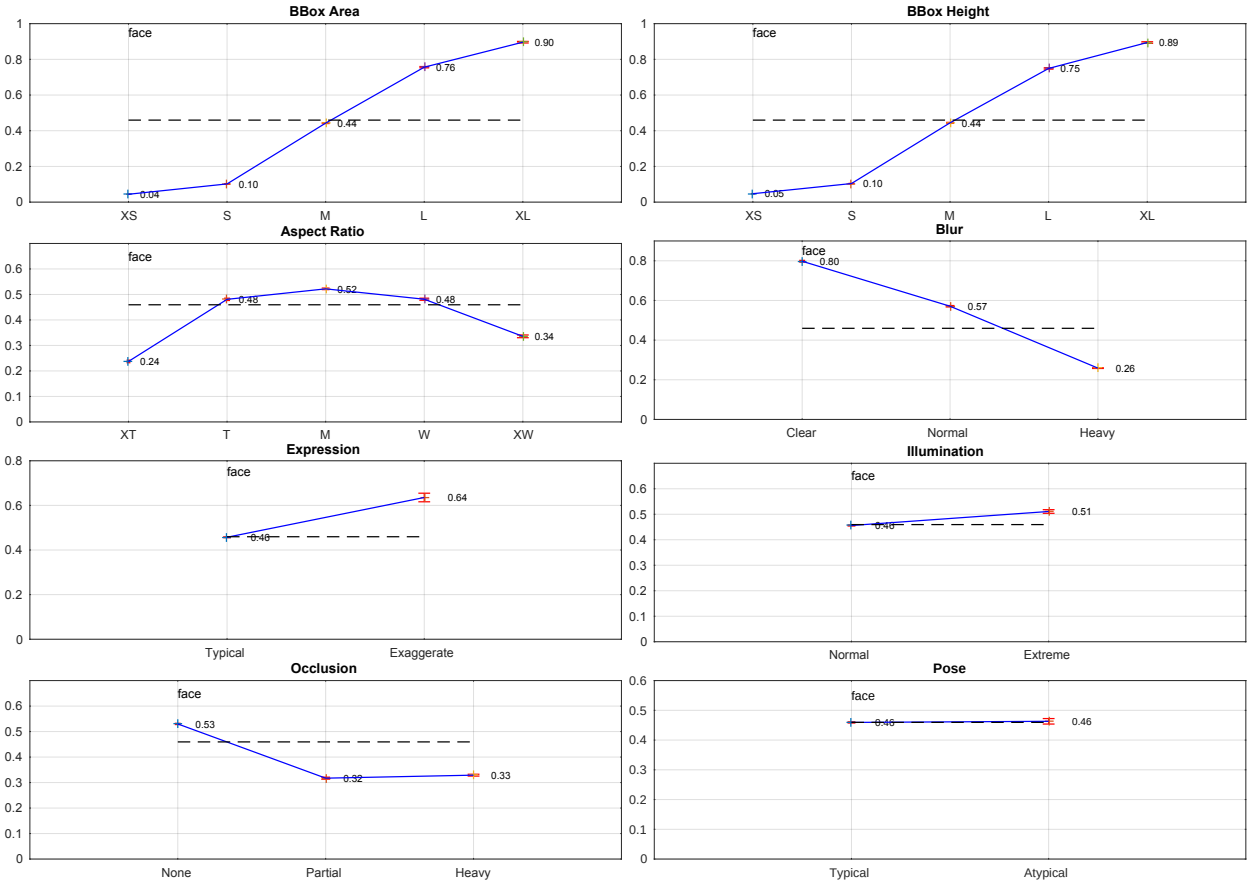


Figure 3: Sensitivity and impact of object characteristics. We show normalized AP[1] for each characteristics. Please refer to [1] for definition of "BBox Area", "BBox Height", and "Aspect Ratio" and also refer to [5] for the definition of per-face attributes "Blur", "Expression", "Illumination", "Occlusion", and "Pose". Our detector performs under average in the case of extremely small scale, extremely skewed aspect ratio, heavy blur, and heavy occlusion. Surprisingly, exaggerated expression and extreme illumination correlate with better performance. Pose variation does not have noticeable affect.

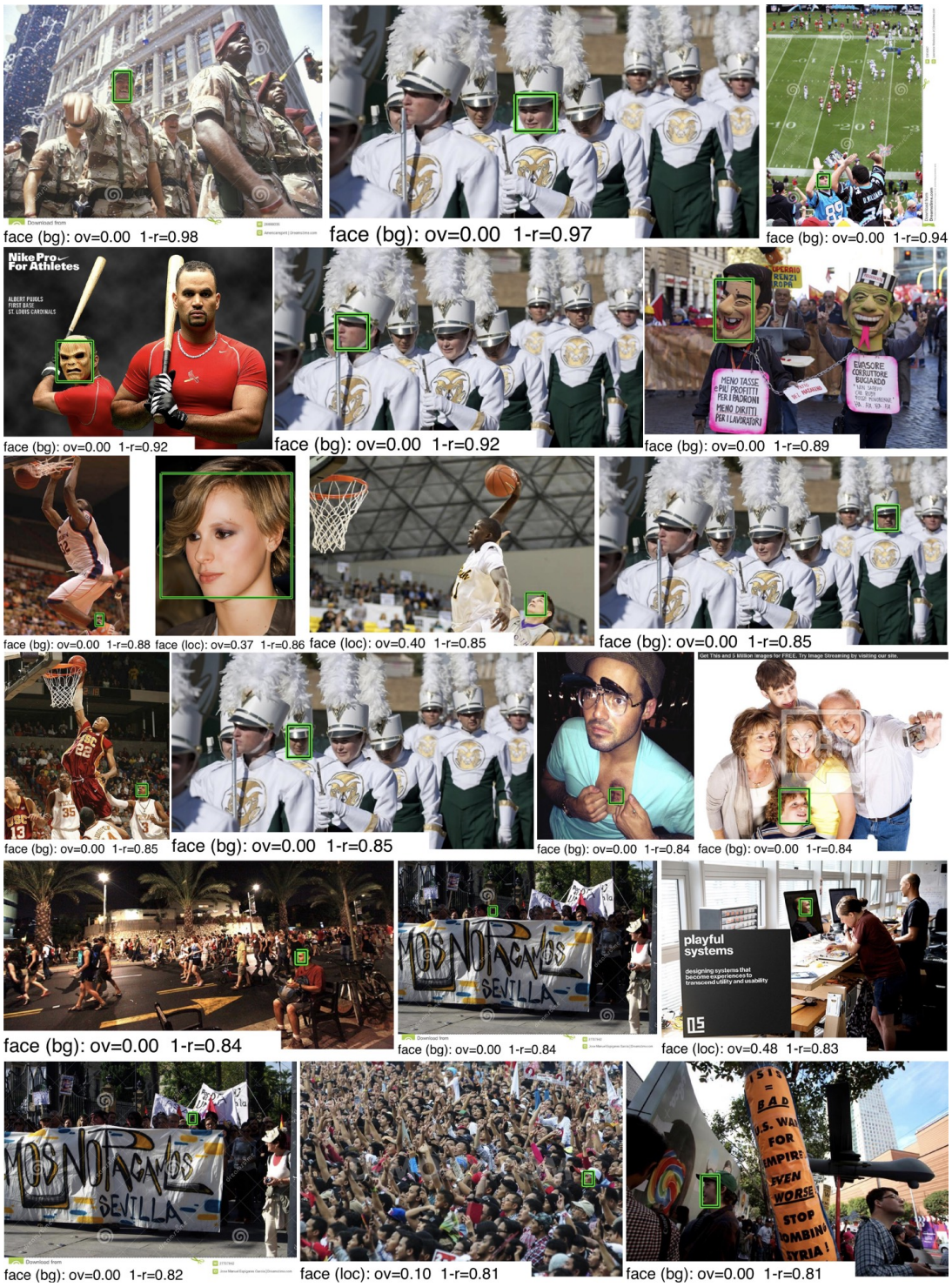


Figure 4: Top 20 scoring false positives on validation set. Error type is labeled at the left bottom of each image. "face(bg)" represents background confusion and "face(loc)" represents inaccurate localization. "ov" represents overlap with ground truth bounding boxes, "1-r" represents the percentage of detections whose confidence is below the current one's. Our detector seems to find faces that were not annotated (when prediction is on the face while "ov" equals to zero).