Finding Tiny Faces

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

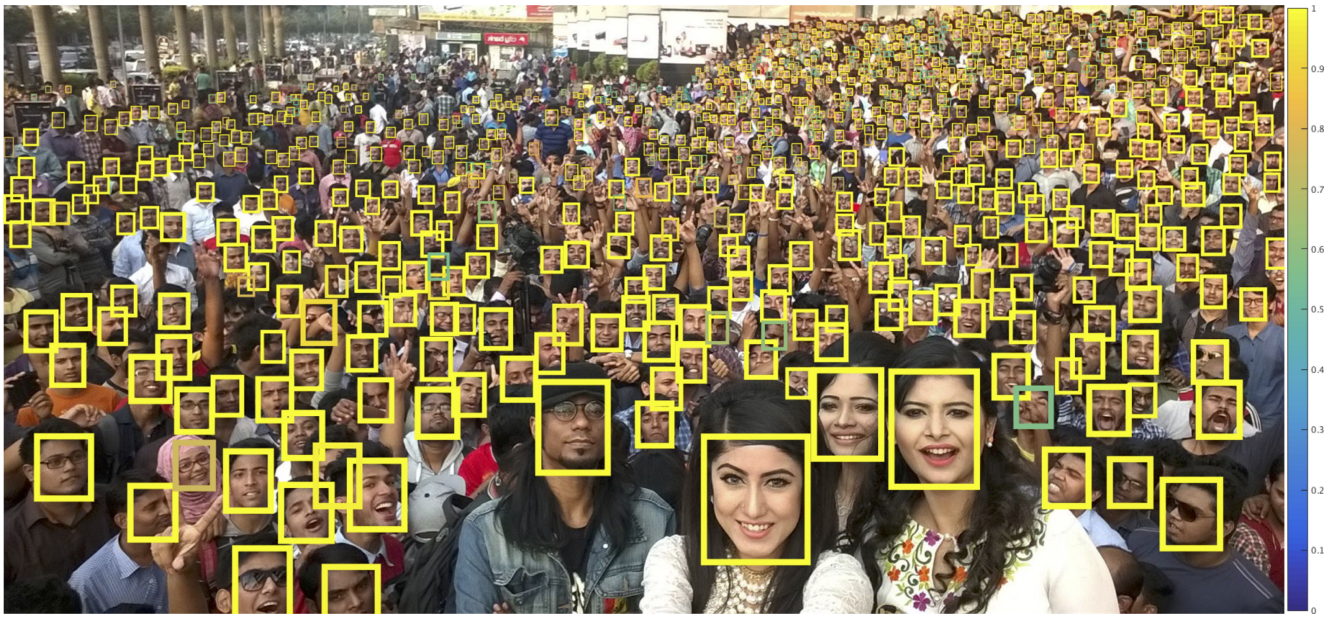

Figure 1: We describe a detector that can find around 800 faces out of the reportedly 1000 present, by making use of novel characterizations of scale, resolution, and context to find small objects. Detector confidence is given by the colorbar on the right: can you confidently identify errors?

Abstract

Though tremendous strides have been made in object recognition, one of the remaining open challenges is detecting small objects. We explore three aspects of the problem in the context of finding small faces: the role of scale invariance, image resolution, and contextual reasoning. While most recognition approaches aim to be scale-invariant, the cues for recognizing a 3px tall face are fundamentally different than those for recognizing a 300px tall face. We take a different approach and train separate detectors for different scales. To maintain efficiency, detectors are trained in a multi-task *fashion: they make use of features extracted from multiple layers of single (deep) feature hierarchy. While training detectors for large objects is straightforward, the crucial challenge remains training detectors for small objects. We show that context is crucial, and define templates that make use of* massively-large receptive fields *(where 99% of the template extends beyond the object of interest). Finally, we explore the role of scale in pre-trained deep networks, providing ways to extrapolate networks tuned for limited scales to rather extreme ranges. We demonstrate state-of-the-art results on massively-benchmarked face datasets (FDDB and WIDER FACE). In particular, when compared to prior art on WIDER FACE, our results* **reduce error by a factor of 2** *(our models produce an AP of 82% while prior art ranges from 29-64%).*

1. Introduction

Though tremendous strides have been made in object recognition, one of the remaining open challenges is detecting small objects. We explore three aspects of the prob-

(a) (b) (c) (d) (e)

Figure 2: Different approaches for capturing scale-invariance. Traditional approaches build a single-scale template that is applied on a finely-discretized image pyramid (a). To exploit different cues available at different resolutions, one could build different detectors for different object scales (b). Such an approach may fail on extreme object scales that are rarely observed in training (or pre-training) data. We make use of a coarse image pyramid to capture extreme scale challenges in (c). Finally, to improve performance on small faces, we model additional context, which is efficiently implemented as a fixed-size receptive field across all scale-specific templates (d). We define templates over features extracted from multiple layers of a deep model, which is analogous to foveal descriptors (e).

lem in the context of face detection: the role of scale invariance, image resolution and contextual reasoning. Scale-invariance is a fundamental property of almost all current recognition and object detection systems. But from a practical perspective, scale-invariance cannot hold for sensors with finite resolution: the cues for recognizing a 300px tall face are undeniably different that those for recognizing a 3px tall face.

Multi-task modeling of scales: Much recent work in object detection makes use of scale-normalized classifiers (e.g., scanning-window detectors run on an image pyramid [5] or region-classifiers run on "ROI"-pooled image features [7, 18]). When resizing regions to a canonical template size, we ask a simple question –*what should the size of the template be?* On one hand, we want a small template that can detect small faces; on the other hand, we want a large template that can exploit detailed features (of say, facial parts) to increase accuracy. Instead of a "one-size-fits-all" approach, we train separate detectors tuned for different scales (and aspect ratios). Training a large collection of scale-specific detectors may suffer from lack of training data for individual scales and inefficiency from running a large number of detectors at test time. To address both concerns, we train and run scale-specific detectors in a *multi-task* fashion : they make use of features defined over multiple layers of single (deep) feature hierarchy. While such a strategy results in detectors of high accuracy for large objects, finding small things is still challenging.

How to generalize pre-trained networks? We provide two remaining key insights to the problem of finding small objects. The first is an analysis of how best to extract scale-invariant features from pre-trained deep networks. We demonstrate that existing networks are tuned for objects of a characteristic size (encountered in pre-training datasets such as ImageNet). To extend features fine-tuned from these networks to objects of novel sizes, we employ a simply strategy: resize images at test-time by interpolation and decimation. While many recognition systems are applied in a "multi-resolution" fashion by processing an image pyramid, we find that interpolating the lowest layer of the pyramid is particularly crucial for finding small objects [5]. Hence our final approach (Fig. 2) is a delicate mixture of scale-specific detectors that are used in a scale-invariant fashion (by processing an image pyramid to capture large scale variations).

How best to encode context? Finding small objects is fundamentally challenging because there is little signal on the object to exploit. Hence we argue that one must use image evidence beyond the object extent. This is often formulated as "context". In Fig. 3, we present a simple human experiment where users attempt to classify true and false positive faces (as given by our detector). It is dramatically clear that humans need context to accurately classify small faces. Though this observation is quite intuitive and highly explored in computer vision [16, 22], it has been notoriously hard to quantifiably demonstrate the benefit of context in recognition [4, 6, 23]. One of the challenges appears to be how to effectively encode large image regions. We demonstrate that convolutional deep features extracted from multiple layers (also known as "hypercolumn" features [8, 14]) are effective "foveal" descriptors that capture both high-resolution detail and coarse low-resolution cues across large receptive field (Fig. 2 (e)). We show that high-resolution components of our foveal descriptors (extracted from lower convolutional layers) are crucial for such accurate localization in Fig. 5.

Our contribution: We provide an in-depth analysis of image resolution, object scale, and spatial context for the purposes of finding small faces. We demonstrate state-of-the-art results on massively-benchmarked face datasets (FDDB and WIDER FACE). In particular, when compared to prior art on WIDER FACE, our results **reduce error by a factor of 2** (our models produce an AP of 82% while prior art ranges from 29-64%).

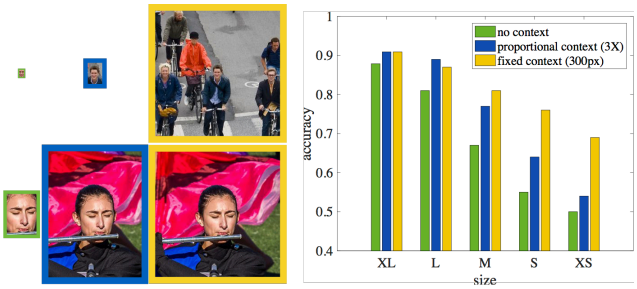

Figure 3: On the **left**, we visualize a large and small face, both with and without context. One does not need context to recognize the large face, while the small face is dramatically unrecognizable without its context. We quantify this observation with a simple human experiment on the **right**, where users classify true and false positive faces of our proposed detector. Adding *proportional* context (by enlarging the window by 3X) provides a small improvement on large faces but is insufficient for small faces. Adding a *fixed* contextual window of 300 pixels dramatically reduces error on small faces by 20%. This suggests that context should be modeled in a scale-*variant* manner. We operationalize this observation with foveal templates of massively-large receptive fields (around 300x300, the size of the yellow boxes).

2. Related work

Scale-invariance: The vast majority of recognition pipelines focus on scale-invariant representations, dating back to SIFT[15]. Current approaches to detection such as Faster RCNN [18] subscribe to this philosophy as well, extracting scale-invariant features through ROI pooling or an image pyramid [19]. We provide an in-depth exploration of scale-variant templates, which have been previously proposed for pedestrian detection[17], sometimes in the context of improved speed [3]. SSD [13] is a recent technique based on deep features that makes use of scale-variant templates. Our work differs in our exploration of context for tiny object detection.

Context: Context is key to finding small instances as shown in multiple recognition tasks. In object detection, [2] stacks spatial RNNs (IRNN[11]) model context outside the region of interest and shows improvements on small object detection. In pedestrian detection, [17] uses ground plane estimation as contextual features and improves detection on small instances. In face detection, [28] simultaneously pool ROI features around faces and bodies for scoring detections, which significantly improve overall performance. Our proposed work makes use of large local context (as opposed to a global contextual descriptor [2, 17]) in a scale-variant way (as opposed to [28]). We show that context is mostly useful for finding low-resolution faces.

Multi-scale representation: Multi-scale representation has been proven useful for many recognition tasks. [8, 14, 1] show that deep multi-scale descriptors (known as "hypercolumns") are useful for semantic segmentation. [2, 13] demonstrate improvements for such models on object detection. [28] pools multi-scale ROI features. Our model uses "hypercolumn" features, pointing out that fine-scale features are most useful for localizing small objects (Sec. 3.1 and Fig. 5).

RPN: Our model superficially resembles a region-proposal network (RPN) trained for a specific object class instead of a general "objectness" proposal generator [18]. The important differences are that we use foveal descriptors (implemented through multi-scale features), we select a range of object sizes and aspects through cross-validation, and our models make use of an image pyramid to find extreme scales. In particular, our approach for finding small objects make use of scale-specific detectors tuned for interpolated images. Without these modifications, performance on small-faces dramatically drops by more than 10% (Table 1).

3. Exploring context and resolution

In this section, we present an exploratory analysis of the issues at play that will inform our final model. To frame the discussion, we ask the following simple question: *what is the best way to find small faces of a fixed-size (25x20)?*. By explicitly factoring out scale-variation in terms of the desired output, we can explore the role of context and the canonical template size. Intuitively, context will be crucial for finding small faces. Canonical template size may seem like a strange dimension to explore - given that we want to find faces of size 25x20, why define a template of any size other than 25x20? Our analysis gives a surprising answer of when and why this should be done. To better understand the implications of our analysis, along the way we also ask the analogous question for a large object size: *what is the best way to find large faces of a fixed-size (250x200)?*.

Setup: We explore different strategies for building scanning-window detectors for fixed-size (e.g., 25x20) faces. We treat fixed-size object detection as a *binary heatmap prediction problem*, where the predicted heatmap at a pixel position (x, y) specifies the confidence of a fixed-size detection centered at (x, y). We train heatmap predictors using a fully convolutional network (FCN) [14] defined over a state-of-the-art architecture ResNet [9]. We explore multi-scale features extracted from the last layer of each res-block, i.e. (res2cx, res3dx, res4fx, res5cx) in terms of ResNet-50. We will henceforth refer to these as (res2, res3, res4, res5) features. We discuss the remaining particulars of our training pipeline in Section 5.

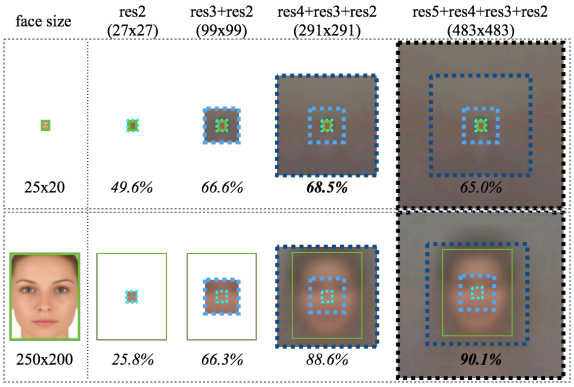

Figure 4: Modeling additional context helps, especially for finding small faces. The improvement from adding context to a tight-fitting template is greater for small faces (18.9%) than for large faces (1.5%). Interestingly smaller receptive fields do better for small faces, because the entire face is visible. The green box represents the actual face size, while dotted boxes represent receptive fields associated with features from different layers (cyan = res2, light-blue = res3, dark-blue = res4, black = res5). Same colors are used in Figures 5 and 7.

3.1. Context

Fig. 4 presents an analysis of the effect of context, as given by the size of the receptive field (RF) used to make heatmap prediction. Recall that for fixed-size detection window, we can choose to make predictions using features with arbitrarily smaller or larger receptive fields compared to this window. Because convolutional features at higher layers tend to have larger receptive fields (e.g., res4 features span 291x291 pixels), smaller receptive fields necessitate the use of lower layer features. We see a number of general trends. Adding context almost always helps, though eventually additional context for tiny faces (beyond 300x300 pixels) hurts. We verified that this was due to over-fitting (by examining training and test performance). Interestingly, smaller receptive fields do better for small faces, because the entire face is visible - it is hard to find large faces if one looks for only the tip of the nose. More importantly, we analyze the impact of context by comparing performance of a "tight" RF (restricted to the object extent) to the best-scoring "loose" RF with additional context. Accuracy for small faces improves by 18.9%, while accuracy for large faces improves by 1.5%, consistent with our human experiments (that suggest that context is most useful for small instances). Our results suggest that we can build multi-task templates for detectors of different sizes with identical receptive fields (of size 291x291), which is particularly simple to implement as a *multi-channel* heatmap prediction problem (where each scale-specific channel and pixel position has its own binary loss). In Fig. 5, we compare between descriptors with and without foveal structure, which shows that high-resolution components of our foveal descriptors are crucial for accurate detection on small instances.

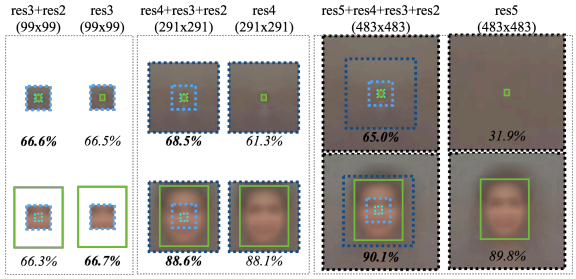

Figure 5: Foveal descriptor is crucial for accurate detection on small objects. The small template (**top**) performs 7% worse with only res4 and 33% worse with only res5. On the contrary, removing foveal structure does not hurt the large template (**bottom**), suggesting high-resolution from lower layers is mostly useful for finding small objects!

3.2. Resolution

We now explore a rather strange question. What if we train a template whose size intentionally differs from the target object to be detected? In theory, one can use a "medium"-size template (50x40) to find small faces (25x20) on a 2X upsampled (interpolated) test image. Fig. 7 actually shows the surprising result that this noticeably boosts performance, from 69% to 75%! We ask the reverse question for large faces: can one find large faces (250x200) by running a template tuned for "medium" faces (125x100) on test images downsampled by 2X? Once again, we see a noticeable increase in performance, from 89% to 94%!

One explanation is that we have different amounts of training data for different object sizes, and we expect better performance for those sizes with more training data. A recurring observation in "in-the-wild" datasets such as WIDER FACE and COCO [12] is that smaller objects greatly outnumber larger objects, in part because more small things can be labeled in a fixed-size image. We verify this for WIDER FACE in Fig. 8 (gray curve). While imbalanced data may explain why detecting large faces is easier with medium templates (because there are more medium-sized faces for training), it does not explain the result for small faces. There exists *less* training examples of medium faces, yet performance is still much better using a medium-size template.

We find that the culprit lies in the distribution of object scales in the *pre-trained* dataset (ImageNet). Fig. 6 reveals that 80% of the training examples in ImageNet contain objects of a "medium" size, between 40 to 140px. Specifically, we hypothesize that the pre-trained ImageNet model (used

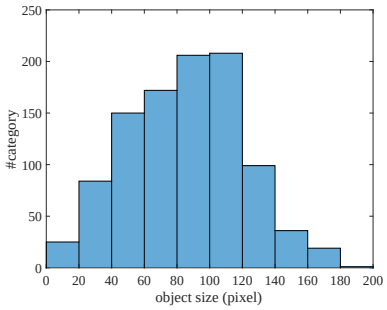

Figure 6: The distribution of average object scales in the ImageNet dataset (assuming images are normalized to 224x224). More than 80% categories have an average object size between 40 and 140 pixel. We hypothesize that models pre-trained on ImageNet are optimized for objects in that range.

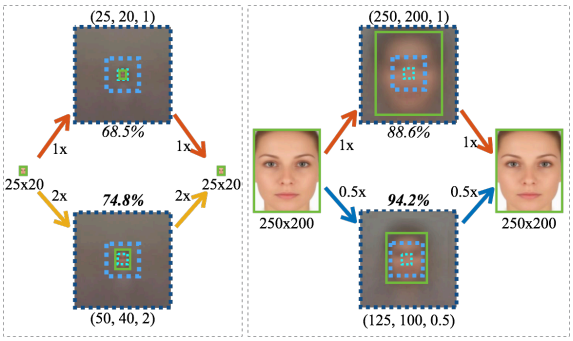

Figure 7: Building templates at original resolution is not optimal. For finding small (25x20) faces, building templates at 2x resolution improves overall accuracy by 6.3%; while for finding large (250x200) faces, building templates at 0.5x resolution improves overall accuracy by 5.6%.

for fine-tuning our scale-specific detectors) is optimized for objects in that range, and that one should bias canonical-size template sizes to lie in that range when possible. We verify this hypothesis in the next section, where we describe a pipeline for building scale-specific detectors with varying canonical resolutions.

4. Approach: scale-specific detection

It is natural to ask a follow-up question: is there a general strategy for selecting template resolutions for particular object sizes? We demonstrate that one can make use of multi-task learning to "brute-force" train several templates at different resolution, and greedily select the ones that do the best. As it turns out, there appears to be a general strategy consistent with our analysis in the previous section.

First, let us define some notation. We use $t(h, w, \sigma)$ to represent a template. Such a template is tuned to detect objects of size $(h/\sigma, w/\sigma)$ at resolution σ. For example, the right-hand-side Fig 7 uses both $t(250, 200, 1)$ (top) and $t(125, 100, 0.5)$ (bottom) to find 250x200 faces.

Given a training dataset of images and bounding boxes, we can define a set of canonical bounding box shapes that roughly covers the bounding box shape space. In this paper, we define such canonical shapes by clustering, which is derived based on Jaccard distance d(Eq. (1)):

$$d(s_i, s_j) = 1 - \mathrm{J}(s_i, s_j) \tag{1}$$

where, $s_i = (h_i, w_i)$ and $s_j = (h_j, w_j)$ are a pair of bounding box shapes and J represents the standard Jaccard similarity (intersection over union overlap).

Now for each target object size $s_i = (h_i, w_i)$, we ask: *what σ_i will maximize performance of $t_i(\sigma_i h_i, \sigma_i w_i, \sigma_i)$?* To answer, we simply train separate multi-task models for each value of $\sigma \in \Sigma$ (some fixed set) and take the max for each object size. We plot the performance of each resolution-specific multi-task model as a colored curve in Fig. 8. With optimal σ_i for each (h_i, w_i), we retrain one multi-task model with "hybrid" resolutions (referred to as HR), which in practice follows the upper envelope of all the curves. Interestingly, there exist natural regimes for different strategies: to find large objects (greater than 140px in height), use 2X smaller canonical resolution. To find small objects (less than 40px in height), use 2X larger canonical template resolution. Otherwise, use the same (1X) resolution. Our results closely follow the statistics of ImageNet (Fig. 6), for which most objects fall into this range.

Pruning: The hybrid-resolution multitask model in the previous section is somewhat redundant. For example, template $(62, 50, 2)$, the optimal template for finding 31x25 faces, is redundant given the existence of template $(64, 50, 1)$, the optimal template for finding 64x50 faces. Can we prune away such redundancies? Yes! We refer the reader to the caption in Fig. 9 for an intuitive description. As Table 1 shows, pruning away redundant templates led to some small improvement. Essentially, our model can be reduced to a small set of scale-specific templates (tuned for 40-140px tall faces) that can be run on a coarse image pyramid (including 2X interpolation), combined with a set of scale-specific templates designed for finding small faces (less than 20px in height) in 2X interpolated images.

4.1. Architecture

We visualize our proposed architecture in Fig. 10. We train binary multi-channel heatmap predictors to report object confidences for a range of face sizes (40-140px in height). We then find larger and smaller faces with a coarse image pyramid, which importantly includes a 2X upsampling stage with special-purpose heatmaps that are predicted only for this resolution (e.g., designed for tiny faces

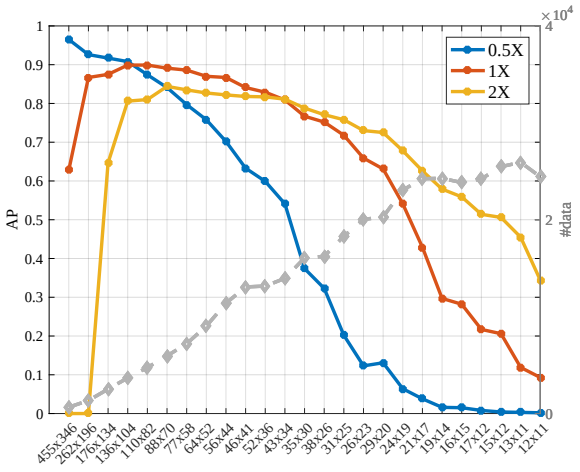

Figure 8: Template resolution analysis. X-axis represents target object sizes, derived by clustering. Left Y-axis shows AP at each target size (ignoring objects with more than 0.5 Jaccard distance). Natural regimes emerge in the figure: for finding large faces (more than 140px in height), build templates at 0.5 resolution; for finding smaller faces (less than 40px in height), build templates at 2X resolution. For sizes in between, build templates at 1X resolution. Right Y-axis along with the gray curve shows the number of data within 0.5 Jaccard distance for each object size, suggesting that more small faces are annotated.

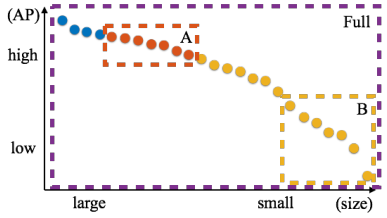

Figure 9: Pruning away redundant templates. Suppose we test templates built at 1X resolution (A) on a coarse image pyramid (including 2X interpolation). They will cover a larger range of scale except extremely small sizes, which are best detected using templates built at 2X, as shown in Fig. 8. Therefore, our final model can be reduced to two small sets of scale-specific templates: (A) tuned for 40-140px tall faces and are run on a coarse image pyramid (including 2X interpolation) and (B) tuned for faces shorter than 20px and are only run in 2X interpolated images.

shorter than 20 pixels). For the shared CNNs, we experimented with ResNet101, ResNet50, and VGG16. Though ResNet101 performs the best, we included performance of all models in Table 2. We see that *all* models achieve substantial improvement on "hard" set over prior art, including CMS-RCNN[28] , which also models context, but in a proportional manner (Fig. 3).

Method	Easy	Medium	Hard
RPN	0.896	0.847	0.716
HR-ResNet101 (Full)	0.919	0.908	0.823
HR-ResNet101 (A+B)	**0.925**	**0.914**	**0.831**

Table 1: Pruning away redundant templates does not hurt performance (validation). As a reference, we also included the performance of a vanilla RPN as mentioned in Sec. 2. Please refer to Fig. 9 for visualization of (Full) and (A+B).

Method	Easy	Medium	Hard
ACF[24]	0.659	0.541	0.273
Two-stage CNN[26]	0.681	0.618	0.323
Multiscale Cascade CNN[25]	0.691	0.634	0.345
Faceness[25]	0.713	0.664	0.424
Multitask Cascade CNN[27]	0.848	0.825	0.598
CMS-RCNN[28]	0.899	0.874	0.624
HR-VGG16	0.862	0.844	0.749
HR-ResNet50	0.907	0.890	0.802
HR-ResNet101	**0.919**	**0.908**	**0.823**

Table 2: Validation performance of our models with different *architectures*. ResNet101 performs slightly better than ResNet50 and much better than VGG16. Importantly, our VGG16-based model already outperforms prior art by a large margin on "hard" set.

Details: Given training images with ground-truth annotations of objects and templates, we define positive locations to be those where IOU overlap exceeds 70%, and negative locations to be those where the overlap is below 30% (all other locations are ignored by zero-ing out the gradient). Note that this implies that each large object instance generates many more positive training examples than small instances. Since this results in a highly imbalanced binary classification training set, we make use of balanced sampling [7] and hard-example mining [21] to ameliorate such effects. We find performance increased with a post-processing linear regressor that fine-tuned reported bounding-box locations. To ensure that we train on data similar to test conditions, we randomly resize training data to the range of Σ resolution that we consider at test-time (0.5x,1x,2x) and learn from a fixed-size random crop of 500x500 regions per image (to take advantage of batch processing). We fine-tune pre-trained ImageNet models on the WIDER FACE training set with a fixed learning rate of 10^{-4}, and evaluate performance on the WIDER FACE validation set (for diagnostics) and held-out testset. To generate final detections, we apply standard NMS to the detected heatmap with an overlap threshold of 30%. We discuss more training details of our procedure in the Appendix B. Both our code and models are available online at https://www.cs.cmu.edu/~peiyunh/tiny.

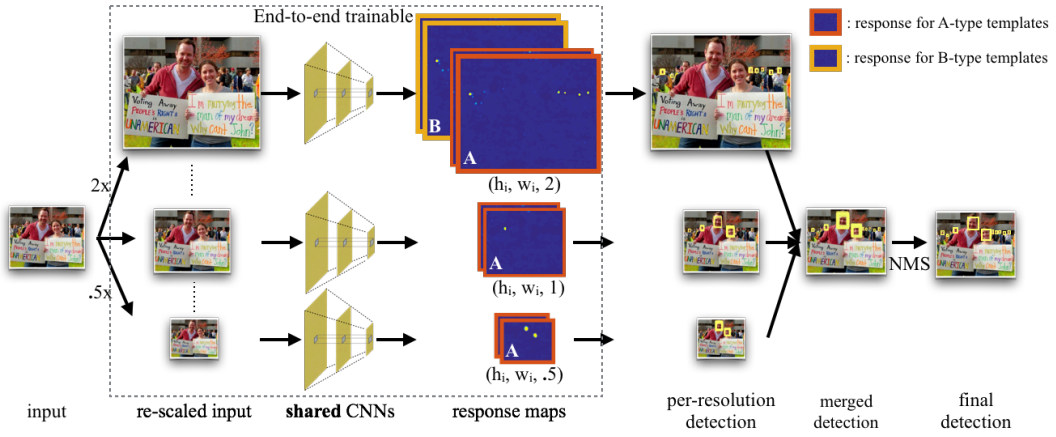

Figure 10: Overview of our detection pipeline. Starting with an input image, we first create a coarse image pyramid (including 2X interpolation). We then feed the scaled input into a CNN to predict template responses (for both detection and regression) at every resolution. In the end, we apply non-maximum suppression (NMS) at the original resolution to get the final detection results. The dotted box represents the end-to-end trainable part. We run A-type templates (tuned for 40-140px tall faces) on the coarse image pyramid (including 2X interpolation), while only run B-type (tuned for less than 20px tall faces) templates on only 2X interpolated images (Fig. 9)

5. Experiments

WIDER FACE: We train a model with 25 templates on WIDER FACE's training set and report the performance of our best model *HR-ResNet101 (A+B)* on the held-out test set. As Fig. 11 shows, our hybrid-resolution model (HR) achieves state-of-the-art performance on all difficulty levels, but most importantly, reduces error on the "hard" set by 2X. Note that "hard" set includes *all* faces taller than 10px, hence more accurately represents performance on the full testset. We visualize our performance under some challenging scenarios in Fig. 13. Please refer to the benchmark website for full evaluation and our Appendix A for more quantitative diagnosis [10].

FDDB: We test our WIDER FACE-trained model on FDDB. Our out-of-the-box detector (HR) outperforms all published results on the discrete score, which uses a standard 50% intersection-over-union threshold to define correctness. Because FDDB uses bounding ellipses while WIDER FACE using bounding boxes, we train a post-hoc linear regressor to transform bounding box predictions to ellipses. With the post-hoc regressor, our detector achieves state-of-the-art performance on the continuous score (measuring average bounding-box overlap) as well. Our regressor is trained with 10-fold cross validation. Fig. 12 plots the performance of our detector both with and without the elliptical regressor (ER). Qualitative results are shown in Fig. 14. Please refer to our Appendix B for a formulation of our elliptical regressor.

Run-time: Our run-time is dominated by running a "fully-convolutional" network across a 2X-upsampled image. Our Resnet101-based detector runs at 1.4FPS on 1080p resolution and 3.1FPS on 720p resolution. Importantly, our run-time is *independent* of the number of faces in an image. This is in contrast to proposal-based detectors such as Faster R-CNN [18], which scale *linearly* with the number of proposals.

Conclusion: We propose a simple yet effective framework for finding small objects, demonstrating that both large context and scale-variant representations are crucial. We specifically show that massively-large receptive fields can be effectively encoded as a foveal descriptor that captures both coarse context (necessary for detecting small objects) and high-resolution image features (helpful for localizing small objects). We also explore the encoding of scale in existing pre-trained deep networks, suggesting a simple way to extrapolate networks tuned for limited scales to more extreme scenarios in a scale-variant fashion. Finally, we use our detailed analysis of scale, resolution, and context to develop a state-of-the-art face detector that significantly outperforms prior work on standard benchmarks.

Acknowledgments: This research is based upon work supported in part by NSF Grant 1618903, the Intel Science and Technology Center for Visual Cloud Systems (ISTC-VCS), Google, and the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R & D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is

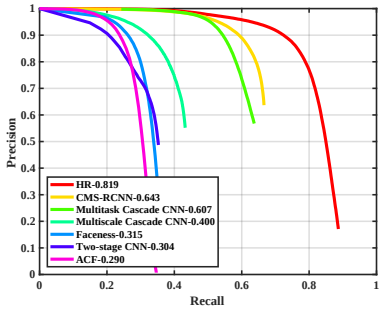

Figure 11: Precision recall curves on WIDER FACE "hard" testset. Compared to prior art, our approach (HR) reduces error by 2X.

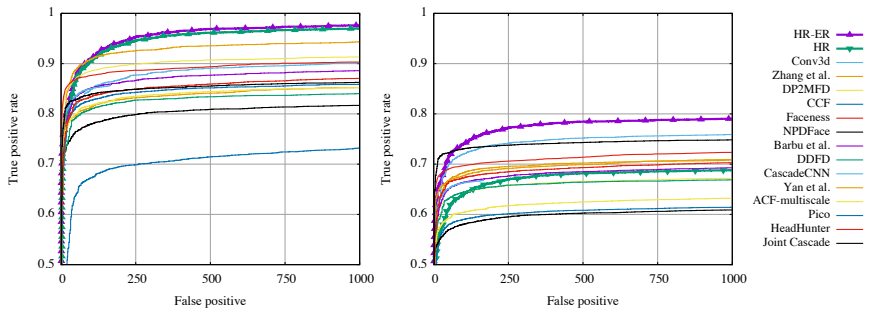

Figure 12: ROC curves on FDDB-test. Our pre-trained detector (HR) produces state-of-the-art discrete detections (**left**). By learning a post-hoc regressor that converts bounding boxes to ellipses, our approach (HR-ER) produces state-of-the-art continuous overlaps as well (**right**). We compare to only published results.

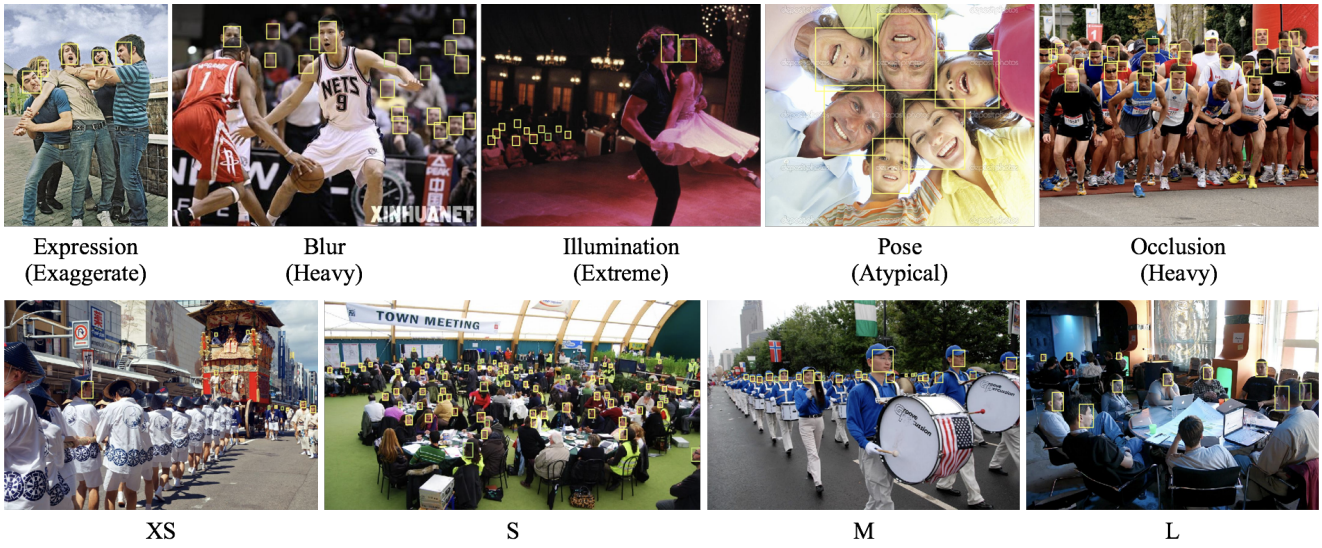

Figure 13: Qualitative results on WIDER FACE. We visualize one example for each attribute and scale. Our proposed detector is able to detect faces at a continuous range of scales, while being robust to challenges such as expression, blur, illumination etc. Please zoom in to look for some very small detections.

A. Error analysis

Quantitative analysis We plot the distribution of error modes among false positives in Fig. 15 and the impact of object characteristics on detection performance in Fig. 16 and Fig. 17.

Qualitative analysis We show top 20 scoring false positives in Fig. 18.

B. Experimental details

Multi-scale features Inspired by the way [20] 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 [20] 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.

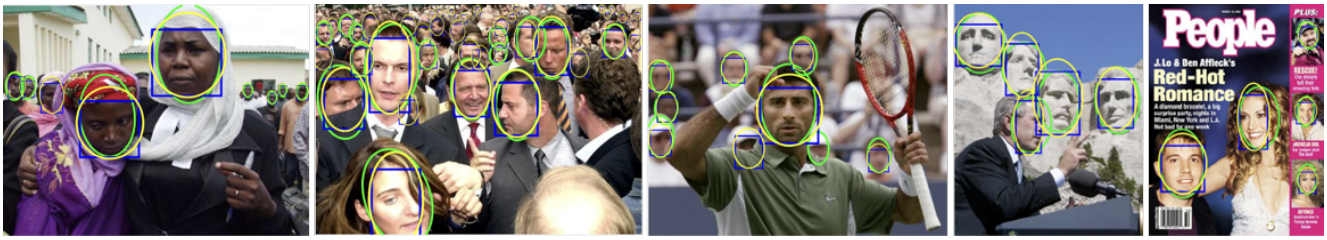

Figure 14: Qualitative results on FDDB. Green ellipses are ground truth, blue bounding boxes are detection results, and yellow ellipses are regressed ellipses. Our proposed detector is robust to heavy occlusion, heavy blur, large appearance and scale variance. Interestingly, many faces under such challenges are not even annotated (second example).

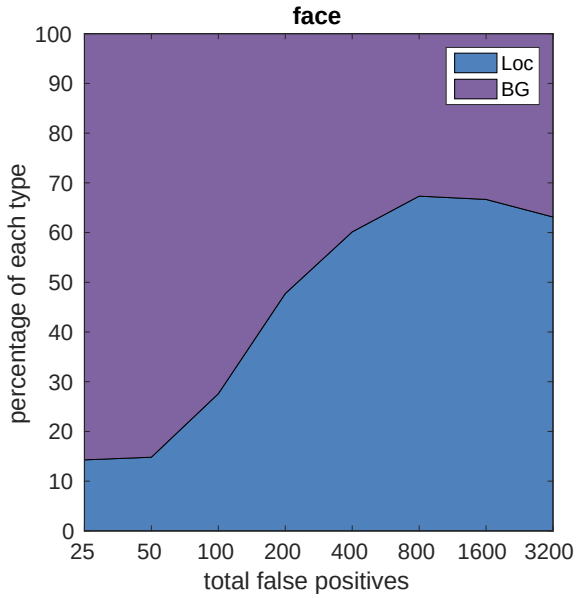

Figure 15: Distribution of error modes of false positives. Background confusion seems the dominating error mode among top-scoring detection, however, we found 15 out of 20 top-scoring false positives, as shown in Fig. 18, are in fact due to missed annotation.

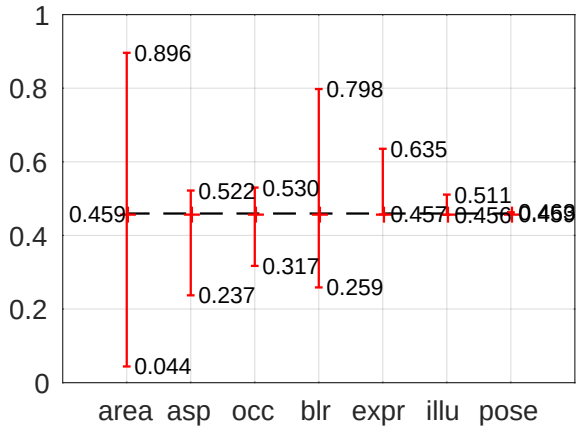

Figure 16: Summary of sensitivity plot. We plot the maximum and minimum of AP_N shown in Figure 17. Our detector is mostly affected by object scale (from 0.044 to 0.896) and blur (from 0.259 to 0.798).

Method	Easy	Medium	Hard
w/ hard mining	**0.919**	**0.908**	0.823
w/o hard mining	0.917	0.904	**0.825**

Table 3: 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.

Border cases Similar to [18], 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 [21]. 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 3.

Loss function Our loss function is formulated in the same way as [18]. 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 [18] 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.

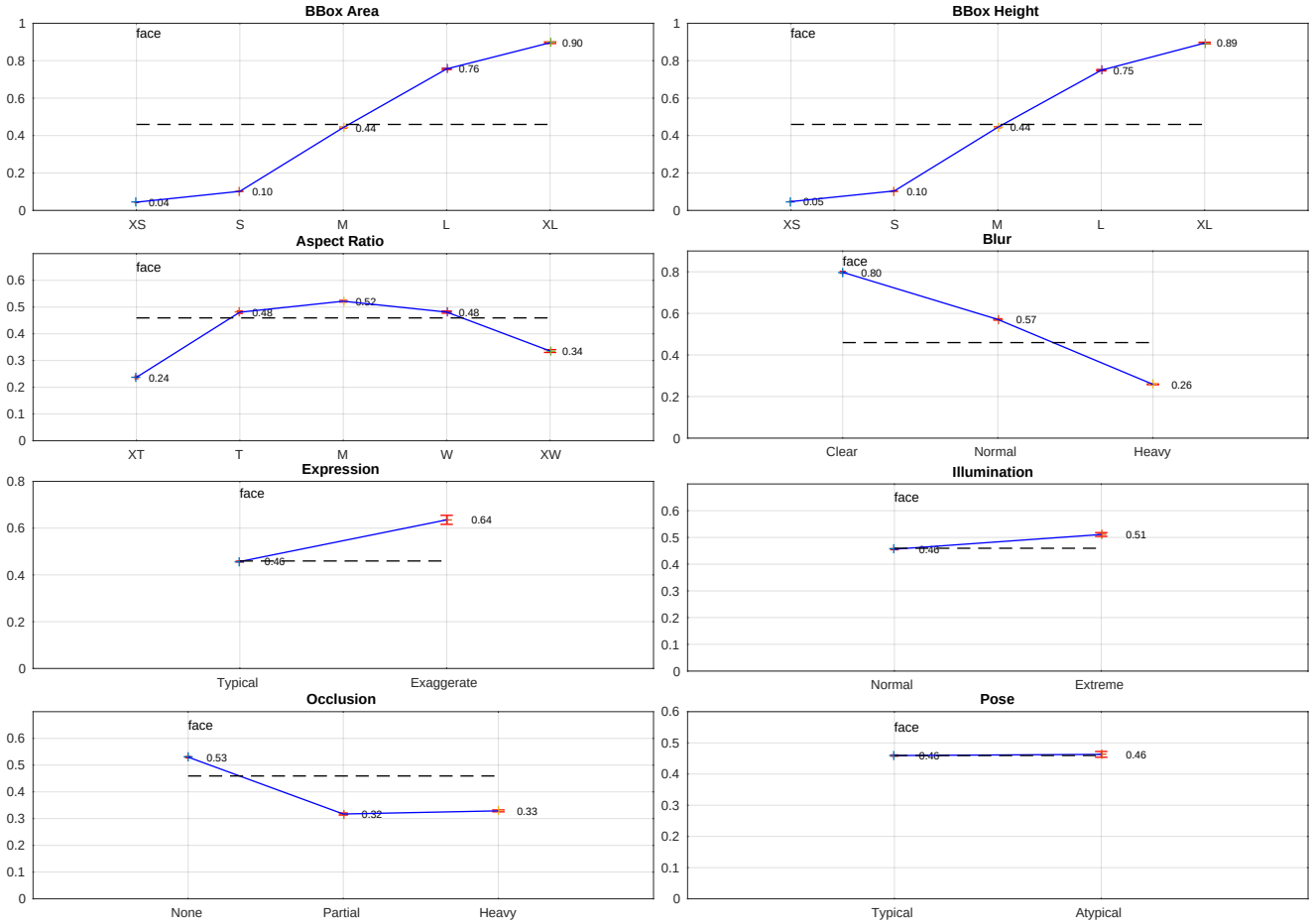

Figure 17: Sensitivity and impact of object characteristics. We show normalized AP[10] for each characteristics. Please refer to [10] for definition of "BBox Area", "BBox Height", and "Aspect Ratio" and also refer to [26] for the definition of per-face attributes "Blur", "Expression", "Illumination", "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.

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

Table 4: 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. (2).

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

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.

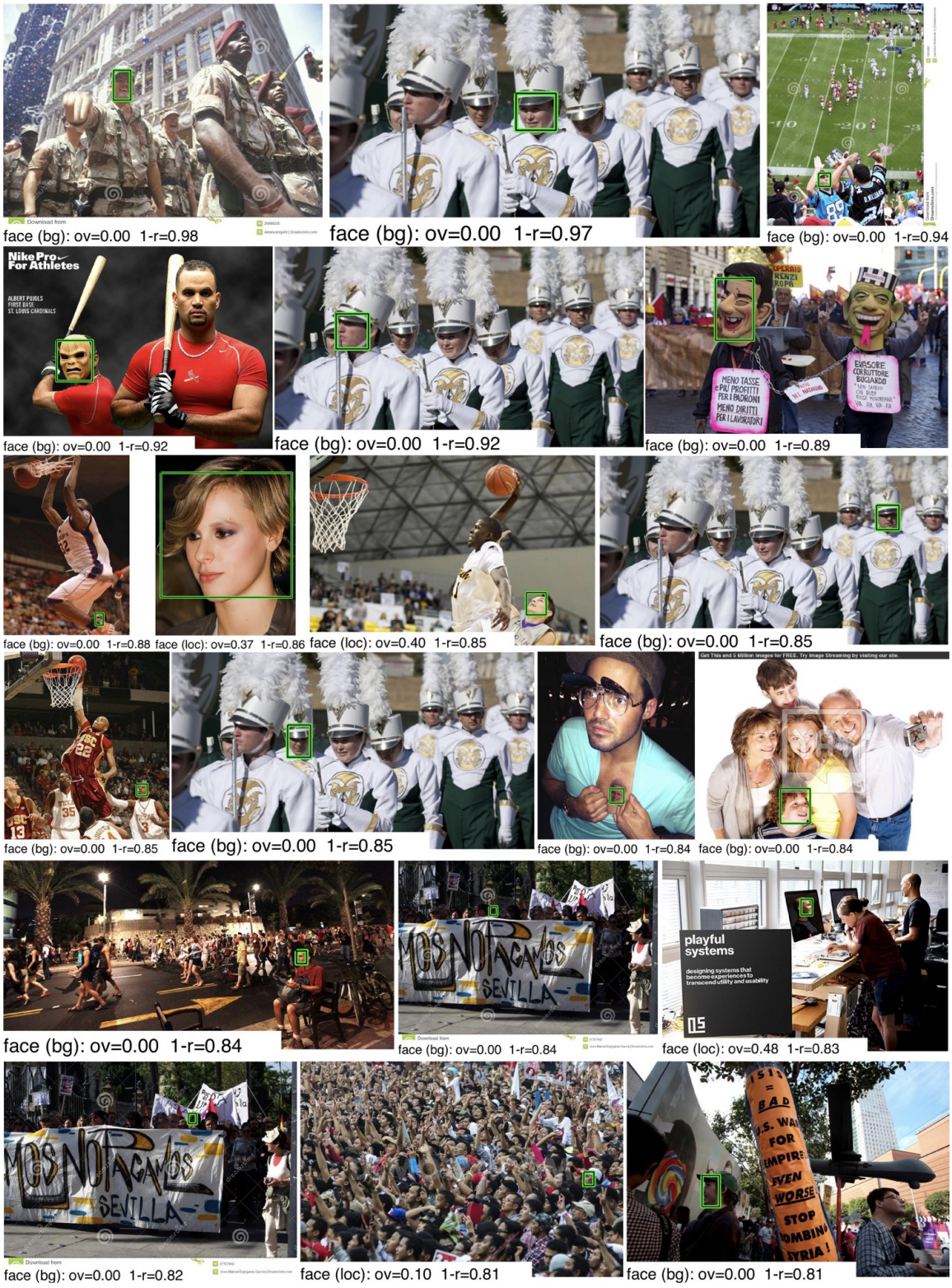

Figure 18: Top 20 scoring false positives on validation set. Error type is labeled at the left bottom of each image. "face(bg)" represents background confusion and "face(loc)" represents inaccurate localization. "ov" represents overlap with ground truth bounding boxes, "1-r" represents the percentage 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).

References

[1] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan. Pixelnet: Towards a General Pixel-level Architecture. *arXiv preprint arXiv:1609.06694*, 2016. 3

[2] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *arXiv preprint arXiv:1512.04143*, 2015. 3

[3] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2903–2910. IEEE, 2012. 3

[4] S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1271–1278. IEEE, 2009. 2

[5] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. 2

[6] C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010. 2

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2, 6

[8] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 447–456, 2015. 2, 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[10] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *European conference on computer vision*, pages 340–353. Springer, 2012. 7, 10

[11] Q. V. Le, N. Jaitly, and G. E. Hinton. A simple way to initialize recurrent networks of rectified linear units. *arXiv preprint arXiv:1504.00941*, 2015. 3

[12] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 4

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 3

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2, 3

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 3

[16] A. Oliva and A. Torralba. The role of context in object recognition. *Trends in cognitive sciences*, 11(12):520–527, 2007. 2

[17] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *European conference on computer vision*, pages 241–254. Springer, 2010. 3

[18] 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. 2, 3, 7, 9

[19] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. *arXiv preprint arXiv:1504.06066*, 2015. 3

[20] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. 8

[21] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. *arXiv preprint arXiv:1604.03540*, 2016. 6, 9

[22] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 273–280. IEEE, 2003. 2

[23] L. Wolf and S. Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, 2006. 2

[24] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *Biometrics (IJCB), 2014 IEEE International Joint Conference on*, pages 1–8. IEEE, 2014. 6

[25] S. Yang, P. Luo, C.-C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015. 6

[26] 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. 6, 10

[27] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multi-task cascaded convolutional networks. *arXiv preprint arXiv:1604.02878*, 2016. 6

[28] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: Contextual multi-scale region-based cnn for unconstrained face detection. *arXiv preprint arXiv:1606.05413*, 2016. 3, 6