SSH: Single Stage Headless Face Detector

Mahyar Najibi* Pouya Samangouei* Rama Chellappa Larry S. Davis
University of Maryland
najibi@cs.umd.edu {pouya,rama,lsd}@umiacs.umd.edu

Abstract

We introduce the Single Stage Headless (SSH) face detector. Unlike two stage proposal-classification detectors, SSH detects faces in a single stage directly from the early convolutional layers in a classification network. SSH is headless. That is, it is able to achieve state-of-the-art results while removing the "head" of its underlying classification network – i.e. *all fully connected layers in the VGG-16 which contains a large number of parameters. Additionally, instead of relying on an image pyramid to detect faces with various scales, SSH is scale-invariant by design. We simultaneously detect faces with different scales in a single forward pass of the network, but from different layers. These properties make SSH fast and light-weight. Surprisingly, with a headless VGG-16, SSH beats the ResNet-101-based state-of-the-art on the WIDER dataset. Even though, unlike the current state-of-the-art, SSH does not use an image pyramid and is* $5X$ *faster. Moreover, if an image pyramid is deployed, our light-weight network achieves state-of-the-art on all subsets of the WIDER dataset, improving the AP by* 2.5%*. SSH also reaches state-of-the-art results on the FDDB and Pascal-Faces datasets while using a small input size, leading to a speed of* 50 *frames/second on a GPU.*

1. Introduction

Face detection is a crucial step in various problems involving verification, identification, expression analysis, *etc*. From the Viola-Jones [29] detector to recent work by Hu *et al*. [7], the performance of face detectors has been improved dramatically. However, detecting small faces is still considered a challenging task. The recent introduction of the *WIDER* face dataset [35], containing a large number of small faces, exposed the performance gap between humans and current face detectors. The problem becomes more challenging when the speed and memory efficiency of the detectors are taken into account. The best performing face detectors are usually slow and have high memory foot-prints (*e.g*. [7] takes more than 1 second to process an image, see Section 4.5) partly due to the huge number of parameters as well as the way robustness to scale or incorporation of context are addressed.

Figure 1: *SSH* is able to detect various face sizes in a single CNN feed-forward pass and without employing an image pyramid in ~ 0.1 second for an image with size 800×1200 on a GPU.

State-of-the-art CNN-based detectors convert image classification networks into two-stage detection systems [4, 24]. In the first stage, early convolutional feature maps are used to propose a set of candidate object boxes. In the second stage, the remaining layers of the classification networks (*e.g*. *fc6~8* in *VGG-16* [26]), which we refer to as the network "*head*", are deployed to extract local features for these candidates and classify them. The head in the classification networks can be computationally expensive (*e.g*. the network head contains $\sim$ 120M parameters in *VGG-16* and $\sim$ 12M parameters in *ResNet-101*). Moreover, in the two stage detectors, the computation must be performed for all proposed candidate boxes.

Very recently, Hu *et al*. [7] showed state-of-the-art results on the *WIDER* face detection benchmark by using a similar approach to the Region Proposal Networks (*RPN*) [24] to directly detect faces. Robustness to input scale is achieved by introducing an image pyramid as an integral

*Authors contributed equally

part of the method. However, it involves processing an input pyramid with an up-sampling scale up to 5000 pixels per side and passing each level to a very deep network which increased inference time.

In this paper, we introduce the Single Stage Headless (*SSH*) face detector. *SSH* performs detection in a single stage. Like *RPN* [24], the early feature maps in a classification network are used to regress a set of predefined anchors towards faces. However, unlike two-stage detectors, the final classification takes place together with regressing the anchors. *SSH* is headless. It is able to achieve state-of-the-art results while removing the head of its underlying network (*i.e*. all fully connected layers in *VGG-16*), leading to a light-weight detector. Finally, *SSH* is scale-invariant by design. Instead of relying on an external multi-scale pyramid as input, inspired by [14], *SSH* detects faces from various depths of the underlying network. This is achieved by placing an efficient convolutional detection module on top of the layers with different strides, each of which is trained for an appropriate range of face scales. Surprisingly, *SSH* based on a headless *VGG-16*, not only outperforms the best-reported *VGG-16* by a large margin but also beats the current *ResNet-101*-based state-of-the-art method on the *WIDER* face detection dataset. Unlike the current state-of-the-art, *SSH* does not deploy an input pyramid and is 5 times faster. If an input pyramid is used with *SSH* as well, our light-weight *VGG-16*-based detector outperforms the best reported *ResNet-101* [7] on all three subsets of the *WIDER* dataset and improves the mean average precision by 4% and 2.5% on the validation and the test set respectively. *SSH* also achieves state-of-the-art results on the *FDDB* and *Pascal-Faces* datasets with a relatively small input size, leading to a speed of 50 frames/second.

The rest of the paper is organized as follows. Section 2 provides an overview of the related works. Section 3 introduces the proposed method. Section 4 presents the experiments and Section 5 concludes the paper.

2. Related Works

2.1. Face Detection

Prior to the re-emergence of convolutional neural networks (*CNN*), different machine learning algorithms were developed to improve face detection performance [29, 39, 10, 11, 18, 2, 31]. However, following the success of these networks in classification tasks [9], they were applied to detection as well [6]. Face detectors based on *CNN*s significantly closed the performance gap between human and artificial detectors [12, 33, 32, 38, 7]. However, the introduction of the challenging *WIDER* dataset [35], containing a large number of small faces, re-highlighted this gap. To improve performance, *CMS-RCNN* [38] changed the *Faster R-CNN* object detector [24] to incorporate context information. Very recently, Hu *et al*. proposed a face detection method based on proposal networks which achieves state-of-the-art results on this dataset [7]. However, in addition to skip connections, an input pyramid is processed by re-scaling the image to different sizes, leading to slow detection speeds. In contrast, *SSH* is able to process multiple face scales simultaneously in a single forward pass of the network, which reduces inference time noticeably.

2.2. Single Stage Detectors and Proposal Networks

The idea of detecting and localizing objects in a single stage has been previously studied for general object detection. *SSD* [16] and *YOLO* [23] perform detection and classification simultaneously by classifying a fixed grid of boxes and regressing them towards objects. *G-CNN* [19] models detection as a piece-wise regression problem and iteratively pushes an initial multi-scale grid of boxes towards objects while classifying them. However, current state-of-the-art methods on the challenging *MS-COCO* object detection benchmark are based on two-stage detectors[15]. *SSH* is a single stage detector; it detects faces directly from the early convolutional layers without requiring a proposal stage.

Although *SSH* is a detector, it is more similar to the object proposal algorithms which are used as the first stage in detection pipelines. These algorithms generally regress a fixed set of *anchors* towards objects and assign an objectness score to each of them. *MultiBox* [28] deploys clustering to define anchors. *RPN* [24], on the other hand, defines anchors as a dense grid of boxes with various scales and aspect ratios, centered at every location in the input feature map. *SSH* uses similar strategies, but to localize and at the same time detect, faces.

2.3. Scale Invariance and Context Modeling

Being scale invariant is important for detecting faces in unconstrained settings. For generic object detection, [1, 36] deploy feature maps of earlier convolutional layers to detect small objects. Recently, [14] used skip connections in the same way as [17] and employed multiple shared *RPN* and classifier heads from different convolutional layers. For face detection, *CMS-RCNN* [38] used the same idea as [1, 36] and added skip connections to the *Faster RCNN* [24]. [7] creates a pyramid of images and processes each separately to detect faces of different sizes. In contrast, *SSH* is capable of detecting faces at different scales in a single forward pass of the network without creating an image pyramid. We employ skip connections in a similar fashion as [17, 14], and train three detection modules jointly from the convolutional layers with different strides to detect small, medium, and large faces.

In two stage object detectors, context is usually modeled by enlarging the window around proposals [36]. [1] models context by deploying a recurrent neural network. For

face detection, *CMS-RCNN* [38] utilizes a larger window with the cost of duplicating the classification head. This increases the memory requirement as well as detection time. *SSH* uses simple convolutional layers to achieve the same larger window effect, leading to more efficient context modeling.

3. Proposed Method

SSH is designed to decrease inference time, have a low memory foot-print, and be scale-invariant. *SSH* is a single-stage detector; *i.e*. instead of dividing the detection task into bounding box proposal and classification, it performs classification together with localization from the global information extracted from the convolutional layers. We empirically show that in this way, *SSH* can remove the "head" of its underlying network while achieving state-of-the-art face detection accuracy. Moreover, *SSH* is scale-invariant by design and can incorporate context efficiently.

3.1. General Architecture

Figure 2 shows the general architecture of *SSH*. It is a fully convolutional network which localizes and classifies faces early on by adding a *detection module* on top of feature maps with strides of 8, 16, and 32, depicted as $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ respectively. The *detection module* consists of a convolutional binary classifier and a regressor for detecting faces and localizing them respectively.

To solve the localization sub-problem, as in [28, 24, 19], *SSH* regresses a set of predefined bounding boxes called *anchor*s, to the ground-truth faces. We employ a similar strategy to the *RPN* [24] to form the anchor set. We define the anchors in a dense overlapping sliding window fashion. At each sliding window location, K anchors are defined which have the same center as that window and different scales. However, unlike *RPN*, we only consider anchors with aspect ratio of one to reduce the number of anchor boxes. We noticed in our experiments that having various aspect ratios does not have a noticeable impact on face detection precision. More formally, if the feature map connected to the detection module $\mathcal{M}_i$ has a size of $W_i \times H_i$, there would be $W_i \times H_i \times K_i$ anchors with aspect ratio one and scales $\{S_i^1, S_i^2, \dots S_i^{K_i}\}$.

For the detection module, a set of convolutional layers are deployed to extract features for face detection and localization as depicted in Figure 3. This includes a simple context module to increase the effective receptive field as discussed in section 3.3. The number of output channels of the context module, (*i.e*. "X" in Figures 3 and 4) is set to 128 for detection module $\mathcal{M}_1$ and 256 for modules $\mathcal{M}_2$ and $\mathcal{M}_3$. Finally, two convolutional layers perform bounding box regression and classification. At each convolution location in $\mathcal{M}_i$, the classifier decides whether the windows at the filter's center and corresponding to each of the scales $\{S_i^k\}_{k=1}^K$ contains a face. A 1×1 convolutional layer with $2 \times K$ output channels is used as the classifier. For the regressor branch, another 1×1 convolutional layer with $4 \times K$ output channels is deployed. At each location during the convolution, the regressor predicts the required change in scale and translation to match each of the *positive* anchors to faces.

3.2. Scale-Invariance Design

In unconstrained settings, faces in images have varying scales. Although forming a multi-scale input pyramid and performing several forward passes during inference, as in [7], makes it possible to detect faces with different scales, it is slow. In contrast, *SSH* detects large and small faces simultaneously in a single forward pass of the network. Inspired by [14], we detect faces from three different convolutional layers of our network using detection modules $\mathcal{M}_1, \mathcal{M}_2$, and $\mathcal{M}_3$. These modules have strides of 8, 16, and 32 and are designed to detect small, medium, and large faces respectively.

More precisely, the detection module $\mathcal{M}_2$ performs detection from the *conv5-3* layer in *VGG-16*. Although it is possible to place the detection module $\mathcal{M}_1$ directly on top of *conv4-3*, we use the feature map fusion which was previously deployed for semantic segmentation [17], and generic object detection [14]. However, to decrease the memory consumption of the model, the number of channels in the feature map is reduced from 512 to 128 using 1×1 convolutions. The *conv5-3* feature maps are up-sampled and summed up with the *conv4-3* features, followed by a 3×3 convolutional layer. We used bilinear up-sampling in the fusion process. For detecting larger faces, a max-pooling layer with stride of 2 is added on top of the *conv5-3* layer to increase its stride to 32. The detection module $\mathcal{M}_3$ is placed on top of this newly added layer.

During the training phase, each detection module $\mathcal{M}_i$ is trained to detect faces from a target scale range as discussed in 3.4. During inference, the predicted boxes from the different scales are joined together followed by Non-Maximum Suppression (*NMS*) to form the final detections.

3.3. Context Module

In two-stage detectors, it is common to incorporate context by enlarging the window around the candidate proposals. *SSH* mimics this strategy by means of simple convolutional layers. Figure 4 shows the context layers which are integrated into the detection modules. Since anchors are classified and regressed in a convolutional manner, applying a larger filter resembles increasing the window size around proposals in a two-stage detector. To this end, we use 5×5 and 7×7 filters in our context module. Modeling the context in this way increases the receptive field proportional to the stride of the corresponding layer and as a result the tar-

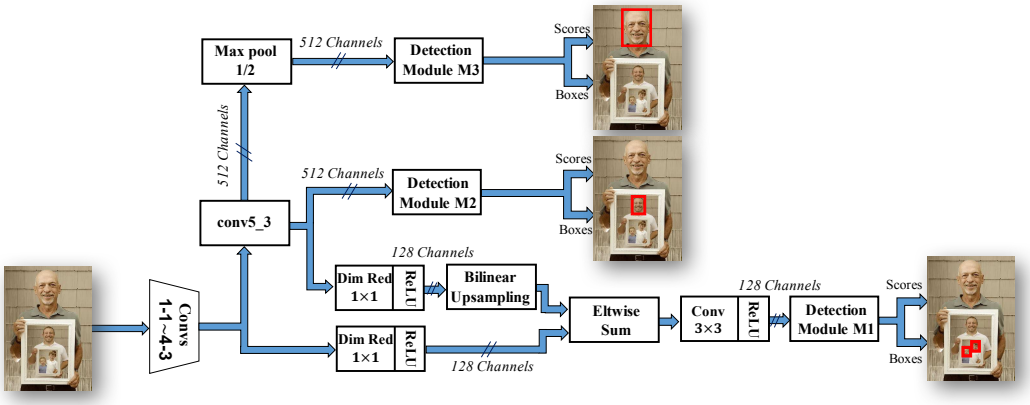


Figure 2: The network architecture of *SSH*.

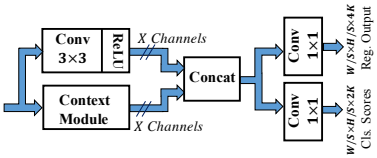


Figure 3: *SSH* detection module.

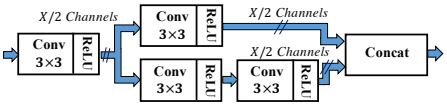


Figure 4: *SSH* context module.

get scale of each detection module. To reduce the number of parameters, we use a similar approach as [27] and deploy sequential 3×3 filters instead of larger convolutional filters. The number of output channels of the detection module (*i.e.* "X" in Figure 4) is set to 128 for $\mathcal{M}_1$ and 256 for modules $\mathcal{M}_2$ and $\mathcal{M}_3$. It should be noted that our detection module together with its context filters uses fewer of parameters compared to the module deployed for proposal generation in [24]. Although, more efficient, we empirically found that the context module improves the mean average precision on the *WIDER* validation dataset by more than half a percent.

3.4. Training

We use stochastic gradient descent with momentum and weight decay for training the network. As discussed in section 3.2, we place three detection modules on layers with different strides to detect faces with different scales. Consequently, our network has three multi-task losses for the classification and regression branches in each of these modules as discussed in Section 3.4.1. To specialize each of the three detection modules for a specific range of scales, we only back-propagate the loss for the anchors which are assigned to faces in the corresponding range. This is implemented by distributing the anchors based on their size to these three modules (*i.e.* smaller anchors are assigned to $\mathcal{M}_1$ compared to $\mathcal{M}_2$, and $\mathcal{M}_3$). An anchor is assigned to a ground-truth face if and only if it has a higher IoU than 0.5. This is in contrast to the methods based on *Faster R-CNN* which assign to each ground-truth at least one anchor with the highest IoU. Thus, we do not back-propagate the loss through the network for ground-truth faces inconsistent with the anchor sizes of a module.

3.4.1 Loss function

SSH has a multi-task loss. This loss can be formulated as follows:

$$\sum_k \frac{1}{N_k^c} \sum_{i \in \mathcal{A}_k} \ell_c(p_i, g_i) + \lambda \sum_k \frac{1}{N_k^r} \sum_{i \in \mathcal{A}_k} \mathcal{I}(g_i = 1) \ell_r(b_i, t_i) \quad (1)$$

where ℓ_c is the face classification loss. We use standard multinomial logistic loss as ℓ_c. The index k goes over the *SSH* detection modules $\mathcal{M} = \{\mathcal{M}_k\}_1^K$ and $\mathcal{A}_k$ represents the set of anchors defined in $\mathcal{M}_k$. The predicted category for the i'th anchor in $\mathcal{M}_k$ and its assigned ground-truth label are denoted as p_i and g_i respectively. As discussed in Section 3.2, an anchor is assigned to a ground-truth bounding box if and only if it has an IoU greater than a threshold (*i.e.* 0.5). As in [24], negative labels are assigned to anchors with IoU less than a predefined threshold (*i.e.* 0.3) with any ground-truth bounding box. N_k^c is the number of anchors in module $\mathcal{M}_k$ which participate in the classification loss computation.

ℓ_r represents the bounding box regression loss. Following [6, 5, 24], we parameterize the regression space

with a log-space shift in the box dimensions and a scale-invariant translation and use smooth ℓ_1 loss as ℓ_r. In this parametrized space, p_i represents the predicted four dimensional translation and scale shift and t_i is its assigned ground-truth regression target for the i'th anchor in module $\mathcal{M}_k$. $\mathcal{I}(.)$ is the indicator function that limits the regression loss only to the positively assigned anchors, and $N_k^r = \sum_{i \in \mathcal{A}_k} I(g_i = 1)$.

3.5. Online hard negative and positive mining

We use online negative and positive mining (*OHEM*) for training *SSH* as described in [25]. However, *OHEM* is applied to each of the detection modules ($\mathcal{M}_k$) separately. That is, for each module $\mathcal{M}_k$, we select the negative anchors with the highest scores and the positive anchors with the lowest scores with respect to the weights of the network at that iteration to form our mini-batch. Also, since the number of negative anchors is more than the positives, following [4], 25% of the mini-batch is reserved for the positive anchors. As empirically shown in Section 4.8, *OHEM* has an important role in the success of *SSH* which removes the fully connected layers out of the *VGG-16* network.

4. Experiments

4.1. Experimental Setup

All models are trained on 4 GPUs in parallel using stochastic gradient descent. We use a mini-batch of 4 images. Our networks are fine-tuned for $21K$ iterations starting from a pre-trained ImageNet classification network. Following [4], we fix the initial convolutions up to *conv3-1*. The learning rate is initially set to 0.04 and drops by a factor of 10 after $18K$ iterations. We set momentum to 0.9, and weight decay to $5e^{-4}$. Anchors with IoU>0.5 are assigned to positive class and anchors which have an IoU<0.3 with all ground-truth faces are assigned to the background class. For anchor generation, we use scales $\{1, 2\}$ in $\mathcal{M}_1$, $\{4, 8\}$ in $\mathcal{M}_2$, and $\{16, 32\}$ in $\mathcal{M}_3$ with a base anchor size of 16 pixels. All anchors have aspect ratio of one. During training, 256 detections per module is selected for each image. During inference, each module outputs 1000 best scoring anchors as detections and *NMS* with a threshold of 0.3 is performed on the outputs of all modules together.

4.2. Datasets

WIDER dataset[35]: This dataset contains $32,203$ images with $393,703$ annotated faces, $158,989$ of which are in the train set, $39,496$ in the validation set and the rest are in the test set. The validation and test set are divided into "easy", "medium", and "hard" subsets cumulatively (*i.e.* the "hard" set contains all images). This is one of the most challenging public face datasets mainly due to the wide variety of face scales and occlusion. We train all models on the train set of the *WIDER* dataset and evaluate on the validation and test sets. Ablation studies are performed on the the validation set (*i.e.* "hard" subset).

Table 1: Comparison of *SSH* with top performing methods on the validation set of the *WIDER* dataset.

Method	easy	medium	hard
CMS-RCNN [38]	89.9	87.4	62.9
HR(VGG-16)+Pyramid [7]	86.2	84.4	74.9
HR(ResNet-101)+Pyramid [7]	92.5	91.0	80.6
SSH(VGG-16)	91.9	90.7	81.4
SSH(VGG-16)+Pyramid	**93.1**	**92.1**	**84.5**

FDDB[8]: *FDDB* contains 2845 images and 5171 annotated faces. We use this dataset only for testing.

Pascal Faces[30]: *Pascal Faces* is a subset of the *Pascal VOC* dataset [3] and contains 851 images annotated for face detection. We use this dataset only to evaluate our method.

4.3. WIDER Dataset Result

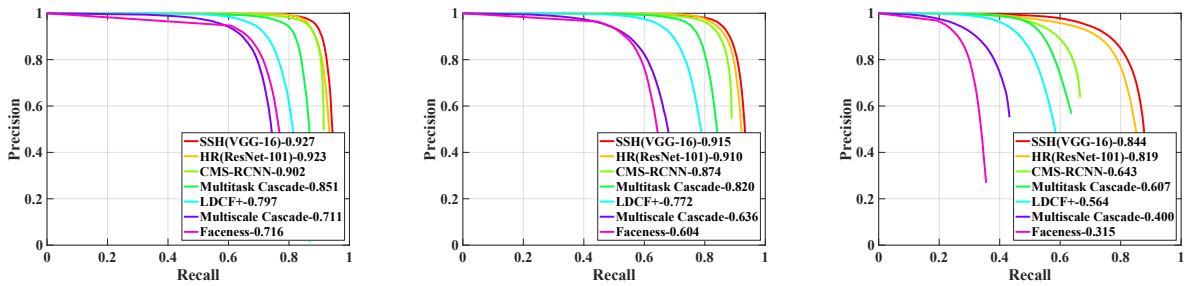
We compare *SSH* with *HR* [7], *CMS-RCNN* [38], *Multitask Cascade CNN* [37], *LDCF* [20], Faceness [34], and *Multiscale Cascade CNN* [35]. When reporting *SSH* without an image pyramid, we rescale the shortest side of the image up to 1200 pixels while keeping the largest side below 1600 pixels without changing the aspect ratio. *SSH+Pyramid* is our method when we apply *SSH* to a pyramid of input images. Like *HR*, a four level image pyramid is deployed. To form the pyramid, the image is first scaled to have a shortest side of up to 800 pixels and the longest side less than 1200 pixels. Then, we scale the image to have min sizes of $500, 800, 1200$, and 1600 pixels in the pyramid. All modules detect faces on all pyramid levels, except $\mathcal{M}_3$ which is not applied to the largest level.

Table 1 compares *SSH* with best performing methods on the *WIDER* validation set. *SSH* without using an image pyramid and based on the *VGG-16* network outperforms the *VGG-16* version of *HR* by $5.7\%, 6.3\%$, and 6.5% in "easy", "medium", and "hard" subsets respectively. Surprisingly, *SSH* also outperforms *HR* based on *ResNet-101* on the whole dataset (*i.e.* "hard" subset) by 0.8. In contrast *HR* deploys an image pyramid. Using an image pyramid, *SSH* based on a light *VGG-16* model, outperforms the *ResNet-101* version of *HR* by a large margin, increasing the state-of-the-art on this dataset by $\sim 4\%$.

The precision-recall curves on the *test* set is presented in Figure 5. We submitted the detections of *SSH* with an image pyramid only once for evaluation. As can be seen, *SSH* based on a headless *VGG-16*, outperforms the prior methods on all subsets, increasing the state-of-the-art by 2.5%.

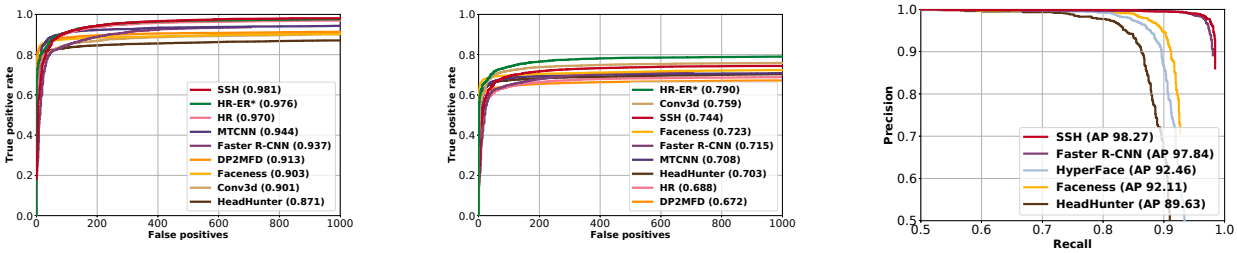
4.4. FDDB and Pascal Faces Results

In these datasets, we resize the shortest side of the input to 400 pixels while keeping the larger side less than



(a) Easy (b) Medium (c) Hard

Figure 5: Comparison among the methods on the test set of *WIDER* face detection benchmark.



(a) FDDB discrete score. (b) FDDB continuous score. (c) Pascal-Faces.

Figure 6: Comparison among the methods on FDDB and Pascal-Faces datasets. (*Note that unlike *SSH*, *HR-ER* is also trained on the FDDB dataset in a 10-*Fold Cross Validation* fashion.)

800 pixels, leading to an inference speed of more than 50 frames/sec. We compare *SSH* with *HR*[7], *HR-ER*[7], Conv3D[13], Faceness[34], Faster R-CNN(*VGG-16*)[24], MTCNN[37], DP2MFD[21], and Headhunter[18]. Figures 6a and 6b show the ROC curves with respect to the discrete and continuous measures on the *FDDB* dataset respectively.

It should be noted that *HR-ER* also uses *FDDB* as a training data in a 10-fold cross validation fashion. Moreover, *HR-ER* and *Conv3D* both generate ellipses to decrease the localization error. In contrast, *SSH* does not use *FDDB* for training, and is evaluated on this dataset out-of-the-box by generating bounding boxes. However, as can be seen, *SSH* outperforms all other methods with respect to the discrete score. Compare to *HR*, *SSH* improved the results by 5.6% and 1.1% with respect to the continuous and discrete scores.

We also compare *SSH* with *Faster R-CNN*(*VGG-16*)[24], *HyperFace*[22], *Headhunter*[18], and *Faceness*[34] on the *Pascal-Faces* dataset. As shown in Figure 6c, *SSH* achieves state-of-the-art results on this dataset.

4.5. Timing

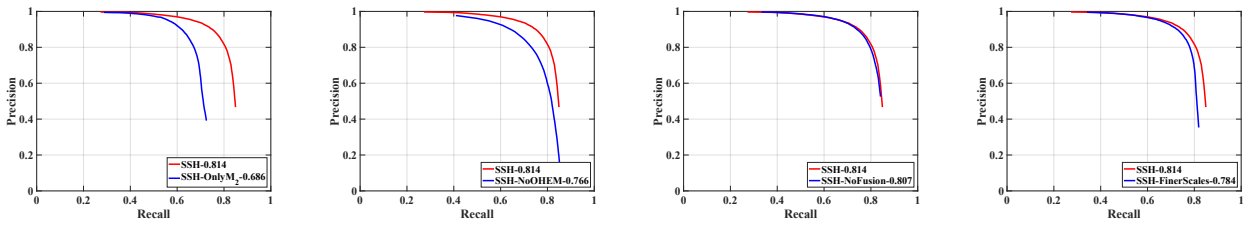
SSH performs face detection in a single stage while removing all fully-connected layers from the *VGG-16* network. This makes *SSH* an efficient detection algorithm. Table 2 shows the inference time with respect to different input sizes. We report average time on the *WIDER* validation set. Timing are performed on a *NVIDIA Quadro P6000* GPU. In column with max size $m \times M$, the shortest side of the images are resized to "m" pixels while keeping the longest side less than "M" pixels. As shown in section 4.3, and 4.4, *SSH* outperforms *HR* on all datasets without an image pyramid. On *WIDER* we resize the image to the last column and as a result detection takes 182 ms/image. In contrast, *HR* has a runtime of 1010 ms/image, more than $5X$ slower. As mentioned in Section 4.4, a maximum input size of 400×800 is enough for *SSH* to achieve state-of-the-art performance on *FDDB* and *Pascal-Faces*, with a detection speed of 50 frames/sec. If an image pyramid is used, the runtime would be dominated by the largest scale.

Table 2: *SSH* inference time with respect to different input sizes.

Max Size	400×800	600×1000	800×1200	1200×1600
Time	48 ms	74 ms	107 ms	182 ms

4.6. Ablation study: Scale-invariant design

As discussed in Section 3.2, *SSH* uses each of its detections modules, $\{\mathcal{M}_i\}_{i=1}^{3}$, to detect faces in a certain range of scales from layers with different strides. To better understand the impact of these design choices, we compare the results of *SSH* with and without multiple detection mod-



(a) Effect of multi-scale design. (b) Effect of *OHEM*. (c) Effect of feature fusion. (d) Effect of increasing #anchors.

Figure 7: Ablation studies. All experiments are reported on the Wider Validation set.

Table 3: The effect of input size on average precision.

Max Size	600×1000	800×1200	1200×1600	1400×1800
AP	68.6	78.4	81.4	81.0

ules. That is, we remove $\{\mathcal{M}_1, \mathcal{M}_3\}$ and only detect faces with $\mathcal{M}_2$ from *conv5-3* in *VGG-16*. However, for fair comparison, all anchor scales in $\{\mathcal{M}_1, \mathcal{M}_3\}$ are moved to $\mathcal{M}_2$ (*i.e.* we use $\cup_{i=1}^{3}\mathbf{S}_i$ in $\mathcal{M}_2$). Other parameters remain the same. We refer to this simpler method as *"SSH-OnlyM_2"*. As shown in Figure 7a, by removing the multiple detection modules from *SSH*, the *AP* significantly drops by $\sim 12.8\%$ on the *hard* subset which contains smaller faces. Although *SSH* does not deploy the expensive head of its underlying network, results suggest that having independent simple detection modules from different layers of the network is an effective strategy for scale-invariance.

4.7. Ablation study: The effect of input size

The input size can affect face detection precision, especially for small faces. Table 3 shows the *AP* of *SSH* on the *WIDER* validation set when it is trained and evaluated with different input sizes. Even at a maximum input size of 800×1200, *SSH* outperforms *HR-VGG16*, which up-scales images up to 5000 pixels, by 3.5%, showing the effectiveness of our scale-invariant design for detecting small faces.

4.8. Ablation study: The effect of OHEM

As discussed in Section 3.5, we apply hard negative and positive mining (*OHEM*) to select anchors for each of our detection modules. To show its role, we train *SSH*, with and without *OHEM*. All other factors are the same. Figure 7b shows the results. Clearly, *OHEM* is important for the success of our light-weight detection method which does not use the pre-trained head of the *VGG-16* network.

4.9. Ablation study: The effect of feature fusion

In *SSH*, to form the input features for detection module $\mathcal{M}_1$, the outputs of *conv4-3* and *conv5-3* are fused together. Figure 7c, shows the effectiveness of this design choice. Although it does not have a noticeable computational overhead, as illustrated, it improves the *AP* on the *WIDER* validation set.

4.10. Ablation study: Selection of anchor scales

As mentioned in Section 4.1, *SSH* uses $\mathbf{S}_1 = \{1, 2\}$, $\mathbf{S}_2 = \{4, 8\}$, $\mathbf{S}_3 = \{16, 32\}$ as anchor scale sets. Figure 7d compares *SSH* with its slight variant which uses $\mathbf{S}_1 = \{0.25, 0.5, 1, 2, 3\}$, $\mathbf{S}_2 = \{4, 6, 8, 10, 12\}$, $\mathbf{S}_3 = \{16, 20, 24, 28, 32\}$. Although using a finer scale set leads to a slower inference, it also reduces the *AP* due to the increase in the number of *False Positives*.

4.11. Qualitative Results

Figure 8 shows some qualitative results on the Wider validation set. The colors encode the score of the classifier. Green and blue represent score 1.0 and 0.5 respectively.

5. Conclusion

We introduced the *SSH* detector, a fast and lightweight face detector that, unlike two-stage proposal/classification approaches, detects faces in a single stage. *SSH* localizes and detects faces simultaneously from the early convolutional layers in a classification network. *SSH* is able to achieve state-of-the-art results without using the "*head*" of its underlying classification network (*i.e. fc* layers in *VGG-16*). Moreover, instead of processing an input pyramid, *SSH* is designed to be scale-invariant while detecting different face scales in a single forward pass of the network. *SSH* achieves state-of-the-art performance on the challenging *WIDER* dataset as well as *FDDB* and *Pascal-Faces* while reducing the detection time considerably.

Acknowledgement This research is based upon work supported by 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 the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Figure 8: Qualitative results of *SSH* on the validation set of the *WIDER* dataset. Green and blue represent a classification score of 1.0 and 0.5 respectively.

References

[1] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2883, 2016. 2

[2] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer, 2014. 2

[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[4] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 1, 5

[5] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 4

[6] 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, 4

[7] P. Hu and D. Ramanan. Finding tiny faces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 3, 5, 6

[8] V. Jain and E. G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. *UMass Amherst Technical Report*, 2010. 5

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[10] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang. Probabilistic elastic part model for unsupervised face detector adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–800, 2013. 2

[11] H. Li, Z. Lin, J. Brandt, X. Shen, and G. Hua. Efficient boosted exemplar-based face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1843–1850, 2014. 2

[12] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua. A convolutional neural network cascade for face detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5325–5334, 2015. 2

[13] Y. Li, B. Sun, T. Wu, and Y. Wang. face detection with end-to-end integration of a convnet and a 3d model. In *European Conference on Computer Vision*, pages 420–436. Springer, 2016. 6

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2, 3

[15] 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. 2

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 2

[17] 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

[18] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European Conference on Computer Vision*, pages 720–735. Springer, 2014. 2, 6

[19] M. Najibi, M. Rastegari, and L. S. Davis. G-cnn: an iterative grid based object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2369–2377, 2016. 2, 3

[20] E. Ohn-Bar and M. M. Trivedi. To boost or not to boost? on the limits of boosted trees for object detection. *23rd International Conference on Pattern Recognition*, 2016. 5

[21] R. Ranjan, V. M. Patel, and R. Chellappa. A deep pyramid deformable part model for face detection. In *7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–8. IEEE, 2015. 6

[22] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *arXiv preprint arXiv:1603.01249*, 2016. 6

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 779–788, 2016. 2

[24] 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, 2, 3, 4, 6

[25] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 5

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 4

[28] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe. Scalable, high-quality object detection. *arXiv preprint arXiv:1412.1441*, 2014. 2, 3

[29] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I. IEEE, 2001. 1, 2

[30] J. Yan, X. Zhang, Z. Lei, and S. Z. Li. Face detection by structural models. *Image and Vision Computing*, 32(10):790–799, 2014. 5

[31] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2014. 2

[32] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Convolutional channel features. In *Proceedings of the IEEE international conference on computer vision*, pages 82–90, 2015. 2

[33] 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. 2

[34] 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. 5, 6

[35] S. Yang, P. Luo, C.-C. Loy, and X. Tang. Wider face: A face detection benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5525–5533, 2016. 1, 2, 5

[36] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár. A multipath network for object detection. *arXiv preprint arXiv:1604.02135*, 2016. 2

[37] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 5, 6

[38] C. Zhu, Y. Zheng, K. Luu, and M. Savvides. Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection. pages 57–79, 2017. 2, 3, 5

[39] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2879–2886. IEEE, 2012. 2