SSH: Single Stage Headless Face Detector

Mahyar Najibi, Pouya Samangouei, Rama Chellapa,
Larry Davis (ICCV 2017)

Previous Works

- Detecting small faces is a challenging task with high inference time and low memory footprint becoming essential requirements.
- Most of the previous works for object detection use a 2-stage pipeline with bounding box proposals followed by classification task on all proposed bounding boxes.
- Most 2-stage detectors use context information by:
  - enlarging the windows around proposals (Multipath Network)
  - employing a recurrent neural network (Inside Outside Networks)
- Presence of fully-connected layers at the “head” of the network is computationally expensive and adds to the memory requirements.

Previous Works (contd...)

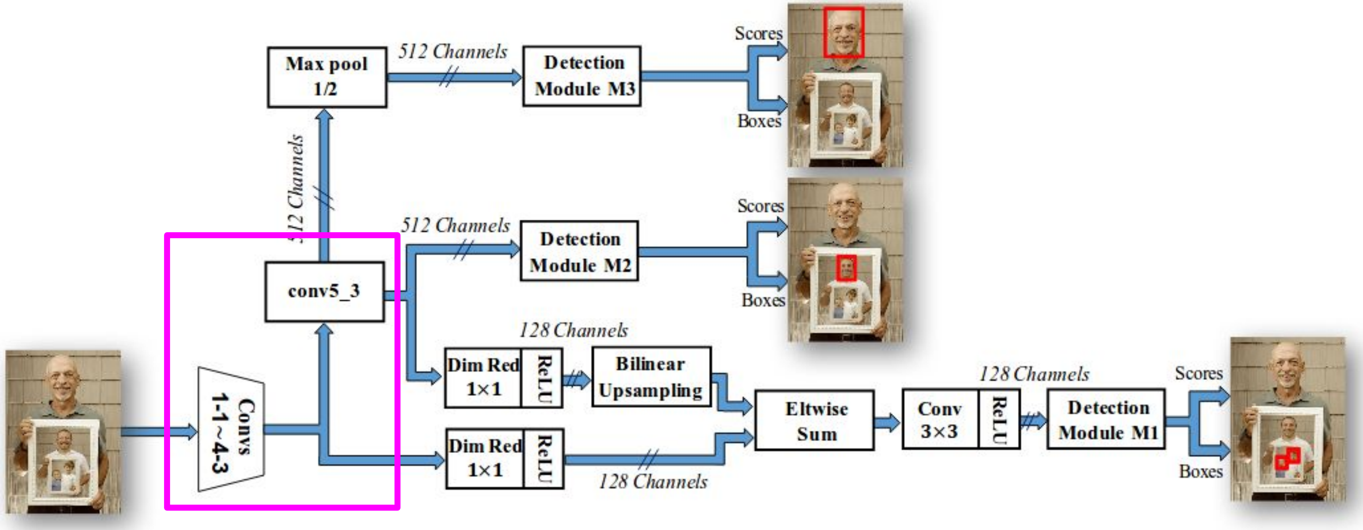
- An improvement, the previous state-of-the-art ("Finding Tiny Faces") used RPN-like model based on Faster RCNN to directly detect faces. But using an image pyramid as input , reduces detection speed.
- CMS-RCNN, based on Faster RCNN, incorporated context information and added skip connections to the Faster RCNN. It also has a large memory requirement.
- SSD, YOLO used an approach to classify and regress boxes simultaneously

Introduction

- employs single stage run for detecting tiny faces. Classification and Regression on proposed boxes done simultaneously without any proposal stage.
- scale invariant, as it does not generate an input pyramid of scaled images, uses 3 detection modules M1, M2, M3 with steps 8,16, 32 respectively for detecting small, medium and large faces.
- Light -weight network achieved by removing the fully-connected layers at the "head" of the network. Also, it contains lesser parameters for detection and context modules than Faster RCNN's proposal generation

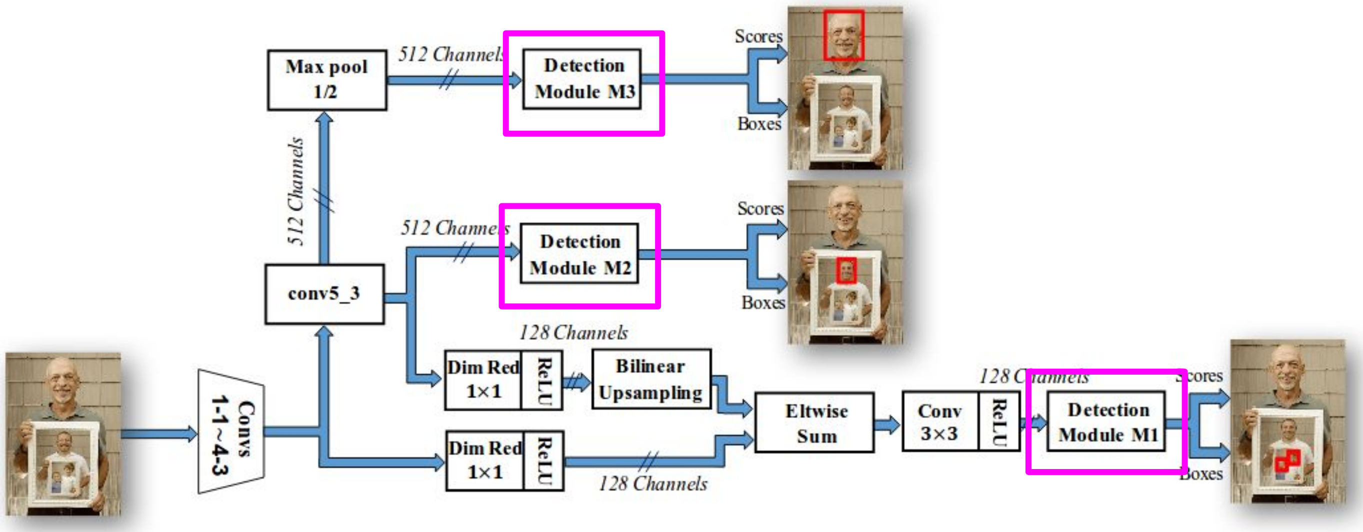
Architecture

- Base Network : VGG16 with the fully connected layers removed.



Architecture

- Base Network : VGG16 with the fully connected layers removed.
- Detection Module: 3 detection modules for different scales [M1, M2, M3]
    - use **RPN** to build set of anchors. Each location defines K anchors with different scales.
    - consists of a binary classifier and regressor.



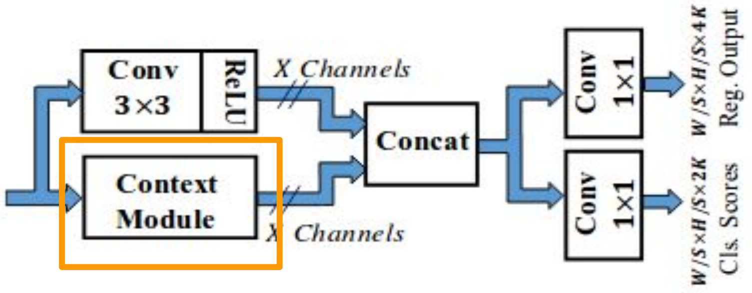
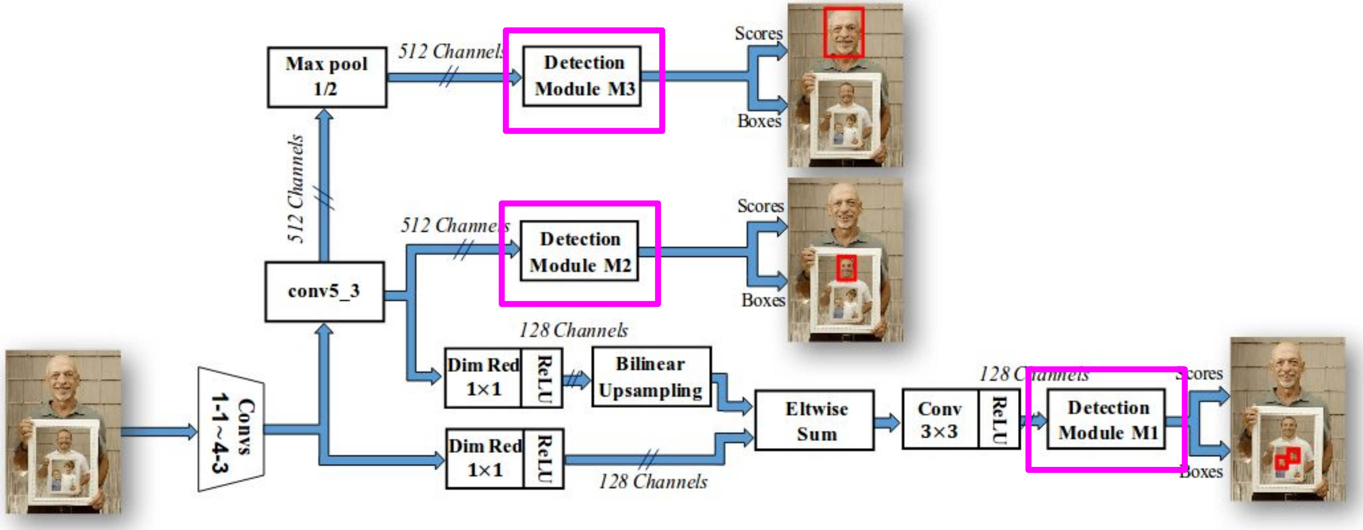


Figure 3: *SSH* detection module.

Architecture

- Base Network : VGG16 with the fully connected layers removed.
- Detection Module: 3 detection modules for different scales [M1, M2, M3]
  - use **RPN** to build set of anchors. Each location defines K anchors with different scales.
  - consists of a binary classifier and regressor.
  - Context Module: incorporating context by enlarging the window around the candidate proposals.



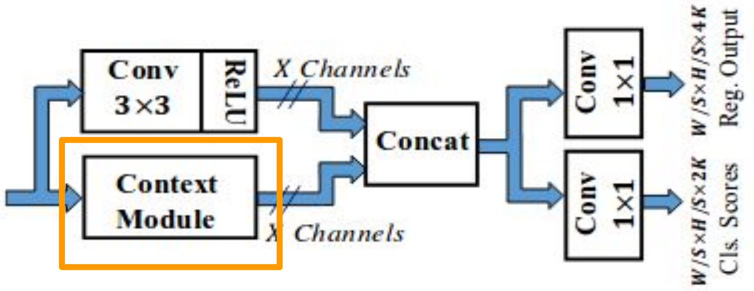
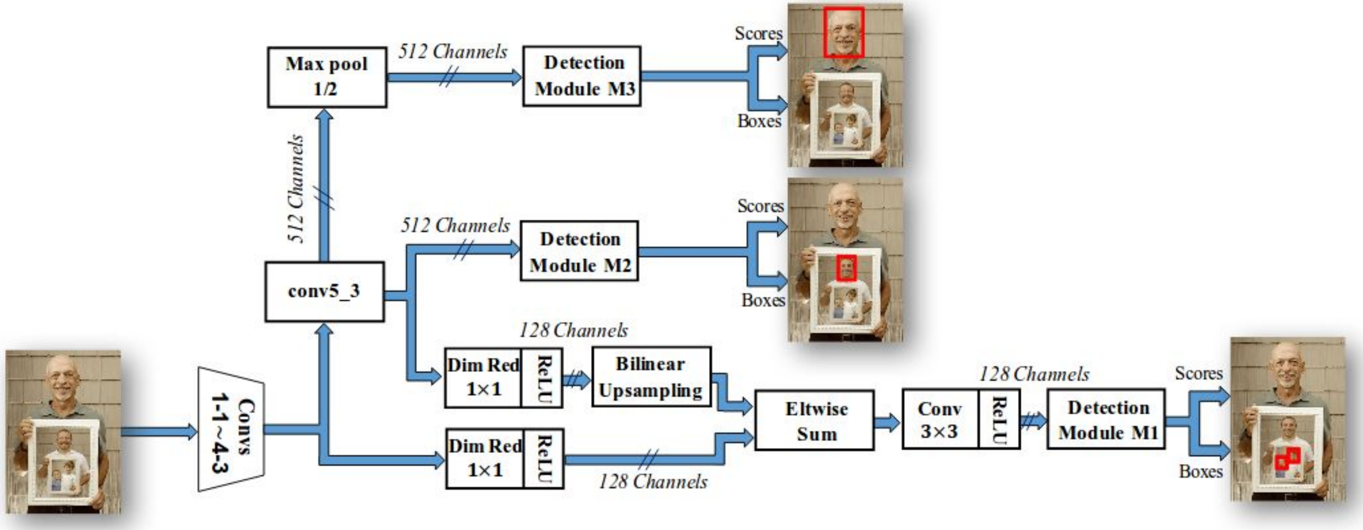


Figure 3: *SSH* detection module.

Architecture

- Base Network : VGG16 with the fully connected layers removed.
- Detection Module: 3 detection modules for different scales [M1, M2, M3]
  - use **RPN** to build set of anchors. Each location defines K anchors with different scales.
  - consists of a binary classifier and regressor.
  - Context Module: incorporating context by enlarging the window around the candidate proposals.



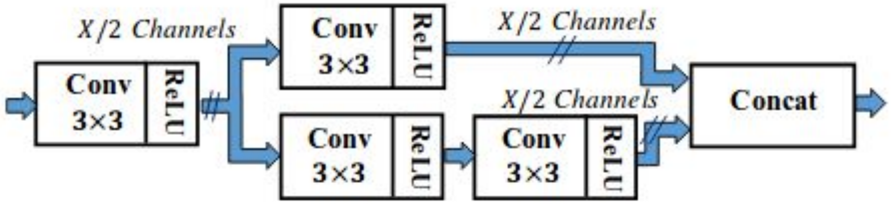


Figure 4: *SSH* context module.

Multi-task Loss function

- Loss is calculated based on face classification loss and bounding-box regression loss.

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

Multi-task Loss function

- Loss is calculated based on face classification loss and bounding-box regression loss.
- Face Classification Loss: calculated as multinomial logistic loss(cross-entropy) on predicted class scores(p_i) and ground truth labels(g_i) per anchor (k)

$$\boxed{\sum_k \frac{1}{N_k^c} \sum_{i \in A_k} \ell_c(p_i, g_i) +}$$

$$\lambda \sum_k \frac{1}{N_k^r} \sum_{i \in A_k} \mathcal{I}(g_i = 1) \ell_r(b_i, t_i) \quad (1)$$

Multi-task Loss function

- Loss is calculated based on face classification loss and bounding-box regression loss.
- Face Classification Loss: calculated as multinomial logistic loss(cross-entropy) on predicted class scores(p_i) and ground truth labels(g_i) per anchor (k)
- Bounding Box Regression Loss: calculated as SmoothL1Loss on predicted (x,y,w,h) bbox representation (b_i) and ground truth regression targets(t_i) on the anchors representing the face class per anchor (k).

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

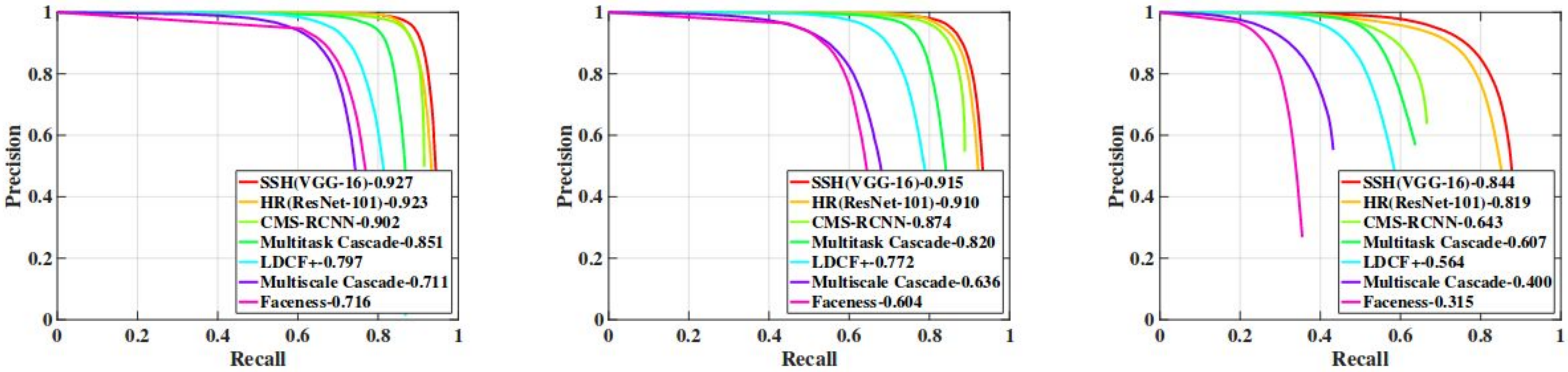
Results (Precision, Recall*)

- Trained on 3 Datasets:
  - WIDER dataset (Training, Testing)
  - FDDB dataset (Testing only)
  - Pascal Faces (Evaluation)
- Validation and Test sets are divided into **easy**, **medium** and **hard** subsets of the data.

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**

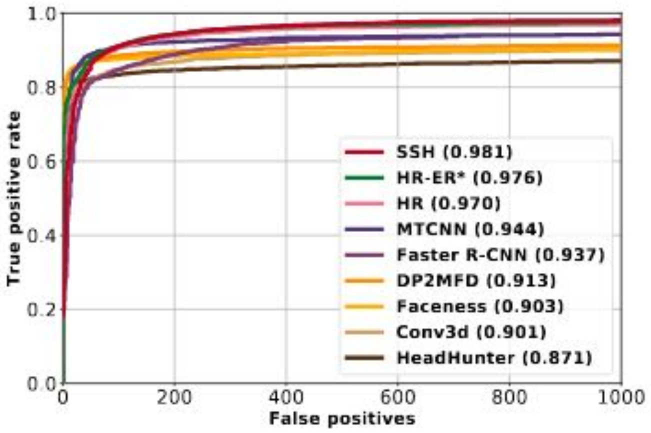
More Results (WIDER test set)



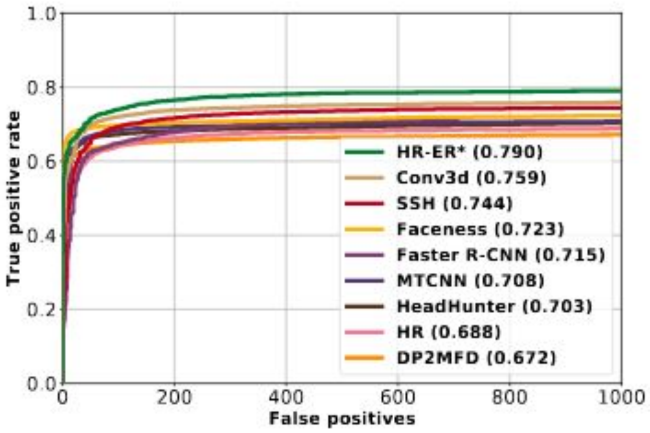
(a) Easy (b) Medium (c) Hard

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

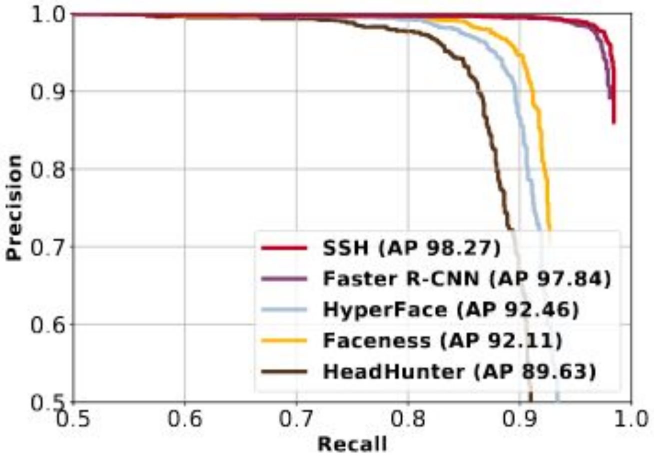
More Results (FDDB, Pascal faces)



(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.)

More Results (Timing)

- Timinig Results are based on WIDER validation set.
- Max Size (m x M) where image is resized to “m” pixels while the longest side is < “M” pixels.

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

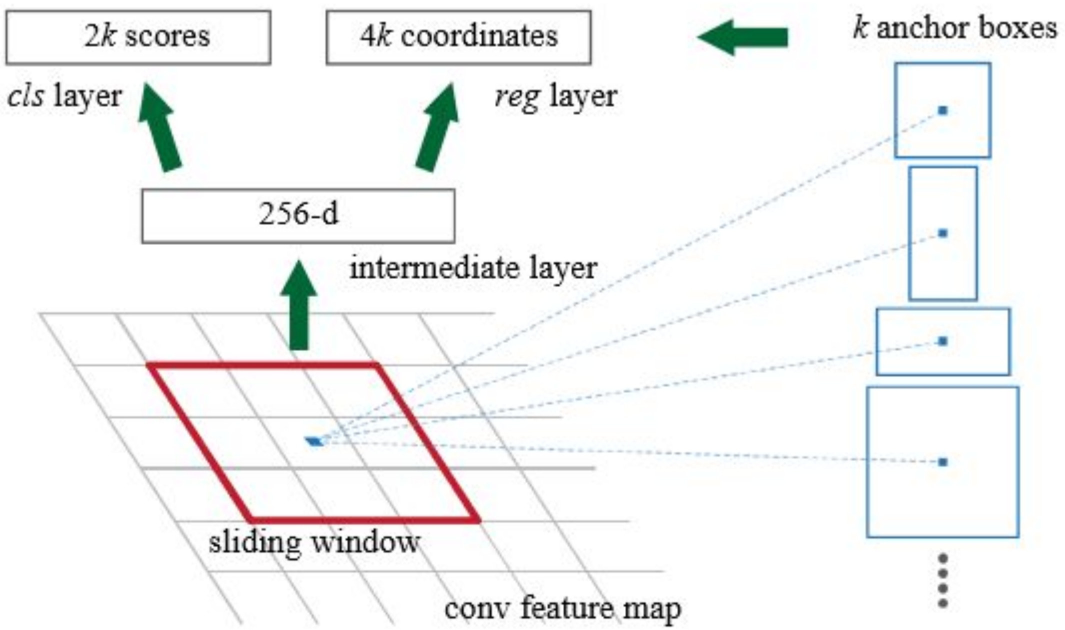
Conclusion

- SSH is single stage, scale invariant face detector with low memory requirements.
- Achieves state-of-the-art without using “head” of the base network
- Uses efficient convolution based context model in contrast to using image pyramid
- Uses detection modules to identify faces of varying scales in an image
- Tested against WIDER dataset, FDDB dataset & Pascal-Faces with reduced detection time

Additional Slides

Region Proposal Network

- Contains anchors as a dense grid of boxes with various scales and aspect ratios,centered at each location in the feature map
- RPN predicts the possibility of an anchor being background or foreground, and refine the anchor.



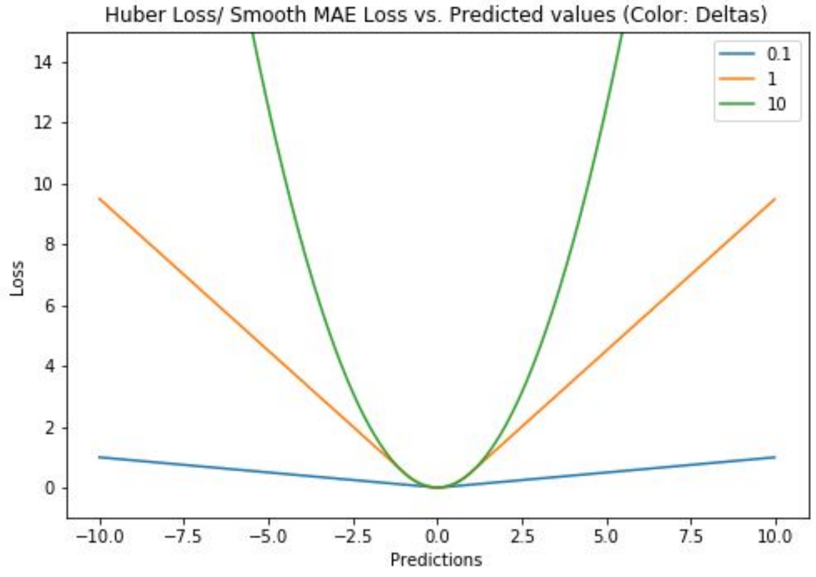
SmoothL1Loss

- Also known as Huber Loss or Smooth Mean Absolute Error
- Less sensitive to outliers than L_2 loss
- Differentiable at 0
- **Hyperparameter**: δ (delta), determines the threshold to consider an outlier
  - for $\boldsymbol{\delta}$ **~ 0**, SmoothL1Loss ~ MAE
  - for $\boldsymbol{\delta}$ **~ inf.** , SmoothL1Loss ~ MSE

+ Using MSE can lead to missing minima when training NN with Large Gradients. Combines the goodness of both MSE & MAE.

- Need to train the hyperparameter which can be an iterative process.

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



Plot of Hoss Loss (Y-axis) vs. Predictions (X-axis). True value = 0