HIERARCHICAL BILINEAR NETWORK FOR HIGH PERFORMANCE FACE DETECTION

Jiangjing Lv[1,2]*, Xiaohu Shao*[1,2*] *,Junliang Xing*[3]*, Pengcheng Liu*[1]*, Xiangdong Zhou*[1]*, Xi Zhou*[1]

[1] Chongqing Institute of Green and Intelligent Technology, CAS, Chongqing, 400714,
[2] China University of Chinese Academy of Sciences, Beijing, 100049, China
[3] National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing, 100190, China

ABSTRACT

Deep Convolutional Networks (DCNs) have achieved great success in face detection. Most architectures of the DCN-based methods, however, suffer from multiple separated steps and large-size models, which increase the training complexity and also slow down the testing speed. In this paper, we propose an efficient end-to-end architecture, called Hierarchical Bilinear Network (HBN), for fast and accurate face detection. It mainly consists of two parts: the Backbone Network and the Bilinear Network. The Backbone Network generates hierarchical feature maps for efficiently characterizing faces of different scales, while the Bilinear Network classifies the regions and regresses the face bounding-boxes on each feature map by introducing the Inception module and weights sharing. Benefited from the characters of the proposed architecture, it obtains a better comprehensive performance regarding the model effectiveness, running efficiency, and parameter size, compared with other DCN-based methods. Extensive experimental results demonstrate that our detector achieves competitive accuracy on both the FDDB database and the WIDER FACE database, while still runs in real time (about 69 FPS on a Titan Black GPU) with a tiny size (2.2 MB) model.

Index Terms— Face detection, end-to-end, hierarchical, bilinear, Inception module.

1. INTRODUCTION

The first step towards automatic face recognition systems is the detection of faces in the image, which has been extensively studied in the area of computer vision [1] in the past couple of decades. Although significant progresses have been made from previous studies [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], especially from recent deep learning based face detection methods [8, 9, 10, 11, 12], it still remains a very difficult problem if we want to fast detect faces from challenging situations like exaggerated facial expressions, large head poses, and / or severe face occlusions [1].

* Corresponding author, e-mail: shaoxiaohu@cigit.ac.cn.

This work is partially supported by the Project from National Natural Science Foundation of China (Grant No. 61672519, 61602433, 61502444), Chongqing Research Program of Basic Research and Frontier Technology (No. cstc2016jcyjA0011), and the CAS "Light of West China" Program.

Many previous methods are motivated to achieve fast and accurate detection results. The seminal work from Viola & Jones [2] presents the first real-time face detection system for near frontal faces, which brings the face detection algorithms from laboratory into practical applications. Based on the framework in [2], many following work are developed to extend the Viola & Jones face detection system to multi-view faces [13, 3, 4, 5, 14], and improve the detection efficiency [3, 4, 15]. One main limitation of traditional face detection algorithms is that, when extending to severer view variations, the corresponding running time and complexity of the model usually increase linearly.

Recently, Deep Convolutional Network (DCN) based models have been employed for face detection [8, 9, 10, 11, 12]. In contrast to traditional face detection models, deep learning based models provide much better extendability to faces with large view variations and other more challenging situations [11, 12], due to their strong feature description ability and large model capacity. Despite these great advantages, most architectures of these DCN-based methods still have two main drawbacks: high model complexity with multiple separated steps (*e.g.*, region proposals, classifiers, and regressors) and large model size with high computation cost, which increase the model learning difficulty during training and also slow down the detection speed during testing.

In order to deal with these two main drawbacks, we propose a novel Hierarchical Bilinear Network (HBN) to provide a fast, accurate, and light-weighted face detector. Within the model architecture, we design a *Backbone Network* to take a single scale image with an arbitrary resolution as input, and generate a convolutional feature pyramid for multi-scale face detection. The Backbone Network does not need to resize the input face image into different scales as in previous works [16, 17], thus making the model size much smaller than previous methods. For each feature map generated from the Backbone Network, we further design a *Bilinear Network* to perform face classification and face bounding-box regression. Compared to the previous work [18] only using one fully connected layer for classification and regression, we introduce two sub-networks in the Bilinear Network to boost the accuracies of the above two tasks. Benefited from the Inception module [19] embedding and the weights sharing among

modules in the Bilinear Networks, our proposed architecture enhances the feature description ability for multi-scale faces without increasing the model size. The HBN consists of the Backbone Network and a series of Bilinear Networks that is able to be trained from end to end.

In contrast to previous DCN based face detection models, the proposed HBN provides a better balance between the model effectiveness, running efficiency, and parameter size. Extensive experimental results on two standard face detection benchmarks, *i.e.*, the FDDB database [6] and the WIDER FACE database [7], have verified the effectiveness and efficiency of the proposed deep face detection architecture.

2. HIERARCHICAL BILINEAR NETWORK

2.1. Network Architecture

Fig. 1 shows the architecture of the proposed model, which mainly consists of two parts, the Backbone Network and the Bilinear Network. Convolution and pooling operations are used alternately in the Backbone Network to generate hierarchical feature maps for observing faces of different sizes. For each feature map, a Bilinear Network is learned for face classification and bounding boxes regression. The HBN provides a fast, accurate, and light-weighted system for face detection.

2.2. Backbone Network

The Backbone Network is designed to build hierarchical feature maps for observing faces of different sizes, the configurations of which is listed in Table 1. The layers before Conv3 encode basic semantic information, which are similar to visual cortex in human vision system. The following convolutional layers are down-sampled feature maps, which are used for multi-scale face detection. Specifically, we use the output of Conv3 and its following convolutional layers as our reference set of feature maps. The expressivity and size of the model is related to the channel number (denoted as *CH* in the rest of this paper) of these convolutional layers. With the increase of *CH*, the network extracts more effective features of faces, but needs more parameters. When *CH* is set to 48, the model has the size of only 0.3MB in each convolutional layer and 45K parameters in total (shown in Table 1). A max-pooling layer with stride 2 is adopted following each convolutional layer. As the spatial resolution is gradually reduced with the pooling operators and the size of receptive field in deeper layers becomes larger, the network is to detect faces with sizes from small to large through hierarchical feature maps.

In order to cover more faces with different sizes, poses, and shapes, we further associate a set of reference bounding boxes called anchors with each feature map cell by following the Region Proposal Network (RPN) [20]. The anchors are pre-defined with different scales and aspect ratios $\{1, 1/2, 2\}$ shown in Table 2, and our architecture can detect faces with the width and height ranging from 8 pixels to 543 pixels.

2.3. Bilinear Network

Compared to the previous work [18] using only one fully connected layer for classification and regression respectively, two sub-networks are introduced in the Bilinear Network for boosting the performance of the above two tasks. For each reference feature map, we associate one Bilinear Network consisting of the confidence sub-network and the localization sub-network. The confidence sub-network predicts whether the anchors contain faces or not, while the localization sub-network regresses the face bounding boxes by predicting shape offsets relative to the anchors.

Since recognition of objects, especially small objects, is sometimes more challenging than locating the object position, we thus adopt the Inception module [19] with the classification sub-network to improve its performance. Furthermore, weights sharing is applied for the Bilinear Networks on different feature maps. It not only enhances face detection accuracy from various scales but also keeps the model light-weighted.

Confidence Sub-network. As revealed in [21], the contextual information with different resolutions is beneficial to detecting faces, especially tiny faces. Considering the performance and the depth of the network, we adopt *inception 3a* as our basic component. It has 1×1, 3×3 and 5×5 convolutions, which correspond to different sizes of receptive field in the reference feature map. With the help of Inception module followed by 3×3 convolutions, more semantic features can be extracted for confidence score prediction.

Localization Sub-network. For each feature map cell correlated with a set of anchors, which is similar to sliding window processors covering various locations and sizes, we just adopted a single 3×3 convolutional layer to predict offset information for the corresponding anchor point location.

Loss function. For the classification loss, we use SoftMax loss to separate the positive and negative training samples. We take the anchors that have intersection over union (IoU) overlapping with each ground-truth box larger than 0.5 as positive samples, and others as negative ones. For the localization loss, we employ the Smooth L1 loss [18] for bounding-box regression and only make regression on positive samples.

Table 1. The configurations of the Backbone Network, where *CH* is set to 48 here and Pad is the spatial padding of convolutional layer.

Layer Name	Filter Size	Stride	Pad	Parameter Number
Conv1	$16 \times 5 \times 5$	2	2	1.2K
Pool_1	2×2	2	0	0
Conv2	$24 \times 3 \times 3$	1	1	3.4K
Pool_2	2×2	2	0	0
Conv3	$CH \times 3 \times 3$	1	1	10.1K
Pool_3	2×2	2	0	0
Conv4	$CH \times 3 \times 3$	1	1	10.1K
Pool_4	2×2	2	0	0
Conv5	$CH \times 3 \times 3$	1	1	10.1K
Pool_5	2×2	2	0	0
Conv6	$CH \times 3 \times 3$	1	1	10.1K

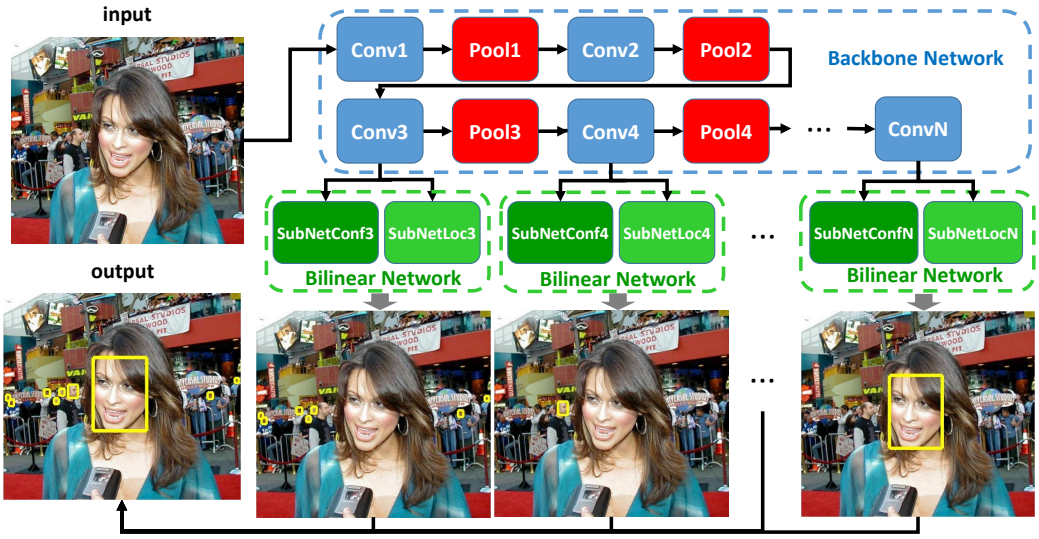


Fig. 1. The framework of our Hierarchical Bilinear Network. It is consists of a Backbone Network and a series of Bilinear Networks. The layers before Conv3 of the Backbone Network without connecting Bilinear Networks are used to encode basic semantic information. The variable N, maximum number of Bilinear Networks, is set to 6 by default in our experiments.

Table 2. The scales and proposals of the reference bounding boxes for each feature map.

Layer	Conv3		Conv4		Conv5		Conv6	
Scale	12^2	24^2	48^2	96^2	144^2	192^2	288^2	384^2
Proposal	12×12	24×24	48×48	96×96	144×144	192×192	288×288	384×384
	8×17	17×34	34×68	68×136	102×204	136×272	204×407	272×543
	17×8	34×17	68×34	136×68	204×102	272×136	407×204	543×272

3. EXPERIMENTS

3.1. Experimental settings

We use the FDDB benchmark [6] and the WIDER FACE's testing set [7] to evaluate our model. Our model is implemented using the Caffe platform and trained on the training set of the WIDER FACE database using input face images of arbitrary sizes. The standard Stochastic Gradient Descent (SGD) algorithm is adopted for updating weights. We use a weight decay of 0.0001 and a momentum of 0.9. The learning rate is set to 0.001 for the first 40k iterations, and 0.0001 for the next 40k iterations. During training, we repeat the training procedure three times for each network and use the last trained model as initialization.

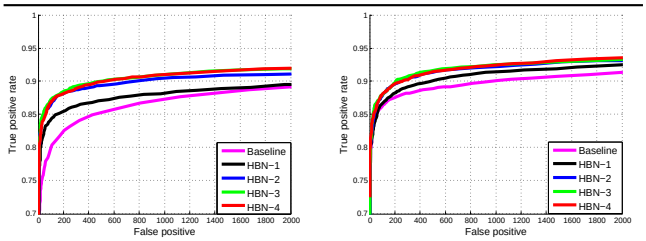
3.2. Model Analyses

For evaluating the influence of the variable *CH* in Backbone Network on accuracy of face detection, we set it to 48 and 64 respectively. In order to verify the efficiency of Inception modules and weights sharing in Bilinear Network, we train 5 networks with different confidence sub-networks (listed in Table 3) for comparison.

We evaluate performances of these networks with different values of *CH* on the FDDB database. The ROC curves are shown in Fig. 2, and the comparison of model sizes and face detection speeds of these networks are listed in Table 4.

Table 3. The configuration of different confidence sub-networks, where I and C represent the Inception module and the convolutional layer respectively.

Model	Conv3	Conv4	Conv5	Conv6
Baseline	C	C	C	C
HBN-1	$I+C$	C	C	C
HBN-2	$I+C$	$I+C$	C	C
HBN-3	$I+C$	$I+C$	$I+C$	C
HBN-4	$I+C$	$I+C$	$I+C$	$I+C$



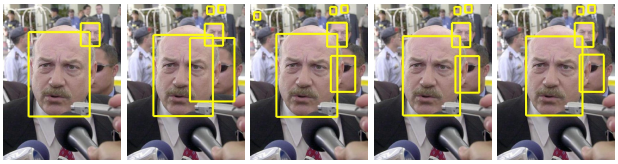
(a) $CH = 48$ (b) $CH = 64$

Fig. 2. ROCs of HBN with different confidence sub-networks on the FDDB database under (a) $CH = 48$ (b) $CH = 64$.

The speed is tested on VGA-resolution images by Titan Black GPU with the Matlab Interface of the Caffe platform.

Analysis of *CH*. Comparing the curves with the same method *e.g.*, HBN-1, in (a) and (b) of Fig. 2, we note that with the increase of *CH* from 48 to 64, the network achieves the better accuracy on face detection, however, it has the larger size of the model and the lower detection speed (see in Table 4).

Analysis of Inception module. We note that adding Inception module significantly improves the face detection performance, especially for tiny face detection (see in Fig. 2). The fact that HBN-2, HBN-3 and HBN-4 get almost the same performance, reveals that higher layer has enough semantic



(a) Baseline (b) HBN-1 (c) HBN-2 (d) HBN-3 (e) HBN-4

Fig. 3. Examples of HBN with different confidence sub-networks under $CH = 64$ on the FDDB database, they show that Inception module significantly improves the accuracy of tiny face detection.

Table 4. Comparison of model sizes and speeds among different confidence sub-networks under different values of *CH*, where *CS* and *TS* represent the size of model storaged by the Caffe platform and the theoretical size of model respectively.

	Model	Baseline	HBN-1	HBN-2	HBN-3	HBN-4
CH=48	*CS*(MB)	0.6	1.3	2.0	2.7	3.4
	TS(MB)	0.4	1.1	1.1	1.1	1.1
	Speed (FPS)	110	79	72	67	62
CH=64	*CS*(MB)	0.8	1.5	2.2	3.0	3.7
	TS(MB)	0.6	1.3	1.3	1.3	1.3
	Speed (FPS)	97	71	69	65	58

feature for large faces as Inception module has little improvement for accuracy of large face detection. Examples of these models are shown in Fig. 3.

Analysis of weights sharing. In Table 4, it is noted that weights sharing does not keep the model size small while introducing more Inception modules, and the practical model sizes are larger than the theoretical ones. The reason is that the Caffe platform stores the shared parameters followed by each feature map repeatedly. In the future work, we will implement our method on other platforms to achieve the theoretical size.

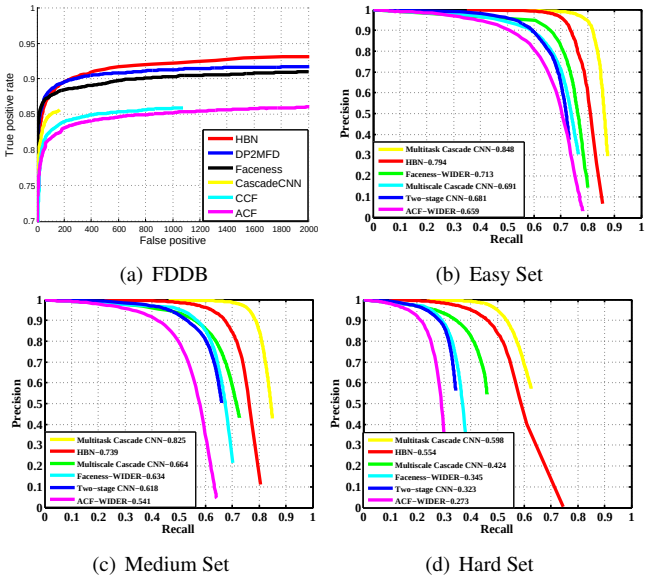
Considering the effectiveness and efficiency, we select HBN-2 with $CH = 64$ for the practical application and perform the following comparisons.

3.3. Comparison with other methods

We compare HBN against the previous face detection methods [10, 9, 8, 16, 22] on the FDDB database, and [17, 9, 7, 22] on the WIDER FACE database respectively, the results are shown in Fig. 4. It is noted that our method achieves much better performance than most of other methods. Although Multitask Cascade CNN [17] which benefits from its cascaded structure and multi-task learning has the superior accuracy, it runs at only 15 FPS on GPU for VGA image comparing with 69 FPS of our method. Examples of our method on FDDB and WIDER FACE are shown in Fig. 5.

4. CONCLUSION

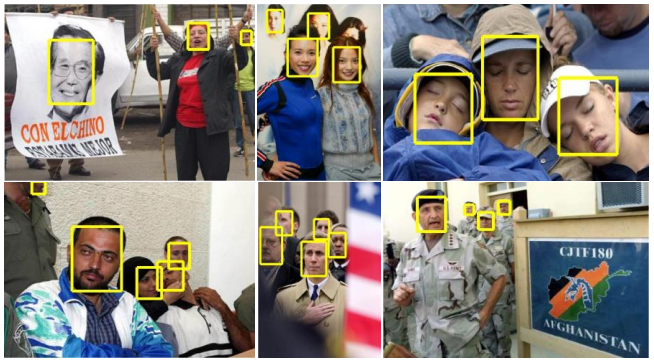
We have presented an end-to-end deep architecture for fast, accurate, and light-weighted face detection. Our method detects faces hierarchically from small faces to large ones without resizing input images into different scales. When comparing with previous methods, it provides promising results on the FDDB and WIDER FACE databases. Benefited from its tiny model size and fast detection speed, it can be deployed for real-time face detection in practical applications.



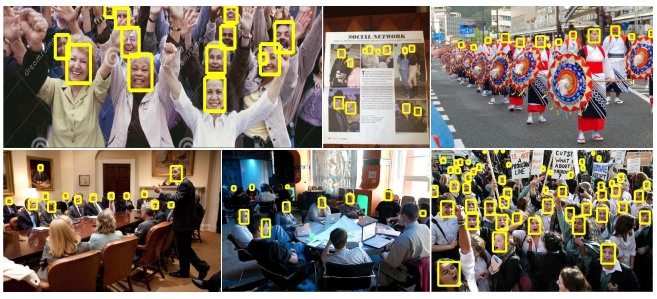
(a) FDDB (b) Easy Set

(c) Medium Set (d) Hard Set

Fig. 4. Comparisons with other methods: (a) ROCs on FDDB; (b)-(d) show Precision Recall Curves (PRC) on the easy/medium/hard validation subsets of WIDER FACE, respectively.



(a) FDDB



(b) WIDER FACE

Fig. 5. Examples of our method on FDDB (a) and WIDER FACE (b).

5. REFERENCES

[1] Stefanos Zafeiriou, Cha Zhang, and Zhengyou Zhang, "A survey on face detection in the wild: Past, present and future," *Computer Vision and Image Understanding*, vol. 138, pp. 1 – 24, 2015.

[2] Paul Viola and Michael J Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[3] Bo Wu, Haizhou Ai, Chang Huang, and Shihong Lao, "Fast rotation invariant multi-view face detection based on real adaboost," in *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, 2004, pp. 79–84.

[4] Chang Huang, Haizhou Ai, Yuan Li, and Shihong Lao, "High-performance rotation invariant multiview face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 671–686, 2007.

[5] Xiangxin Zhu and Deva Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2879–2886.

[6] Vidit Jain and Erik G Learned-Miller, "FDDB: A benchmark for face detection in unconstrained settings," *UMass Amherst Technical Report*, 2010.

[7] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.

[8] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li, "Convolutional channel features," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 82–90.

[9] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang, "From facial parts responses to face detection: A deep learning approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3676–3684.

[10] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa, "A deep pyramid deformable part model for face detection," in *Proceedings of the IEEE International Conference on Biometrics Theory, Applications and Systems*, 2015, pp. 1–8.

[11] Huaizu Jiang and Erik Learned-Miller, "Face detection with the faster R-CNN," *arXiv preprint arXiv:1606.03473*, 2016.

[12] Dong Chen, Gang Hua, Fang Wen, and Jian Sun, "Supervised transformer network for efficient face detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 122–138.

[13] Stan Z Li, Long Zhu, ZhenQiu Zhang, Andrew Blake, HongJiang Zhang, and Harry Shum, "Statistical learning of multi-view face detection," in *Proceedings of European Conference on Computer Vision*, 2002, pp. 67–81.

[14] Haoxiang Li, Gang Hua, Zhe Lin, Jonathan Brandt, and Jianchao Yang, "Probabilistic elastic part model for unsupervised face detector adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 793–800.

[15] Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool, "Face detection without bells and whistles," in *Proceedings of European Conference on Computer Vision*, 2014, pp. 720–735.

[16] Haoxiang Li, Zhe Lin, Xiaohui Shen, Jonathan Brandt, and Gang Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5325–5334.

[17] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[18] Ross Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[21] Peiyun Hu and Deva Ramanan, "Finding tiny faces," *arXiv preprint arXiv:1612.04402*, 2016.

[22] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li, "Aggregate channel features for multi-view face detection," in *Proceedings of the IEEE International Joint Conference on Biometrics*, 2014, pp. 1–8.