Neurocomputing 284 (2018) 119–127

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Detecting Face with Densely Connected Face Proposal Network

Shifeng Zhang[a,b], Xiangyu Zhu[a,b], Zhen Lei[a,b,*], Xiaobo Wang[a,b], Hailin Shi[a,b], Stan Z. Li[a,b]

[a] CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] University of Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 6 July 2017
Revised 3 January 2018
Accepted 10 January 2018

Communicated by Dr Xiaoming Liu

Keywords:
Face detection
Small face
Region proposal network

ABSTRACT

Accuracy and efficiency are two conflicting challenges for face detection, since effective models tend to be computationally prohibitive. To address these two conflicting challenges, our core idea is to shrink the input image and focus on detecting small faces. Reducing the image resolution can significantly improve the detection speed, but it also results in smaller faces that need to pay more attention. Specifically, we propose a novel face detector, dubbed the name Densely Connected Face Proposal Network (DCFPN), with high accuracy as well as CPU real-time speed. Firstly, we subtly design a lightweight-but-powerful fully convolution network with the consideration of efficiency and accuracy. Secondly, we present a dense anchor strategy and a scale-aware anchor matching scheme to improve the recall rate of small faces. Finally, a fair L1 loss is introduced to locate small faces well. As a consequence, our proposed method can detect faces at 30 FPS on a single 2.60 GHz CPU core and 250 FPS using a GPU for the VGA-resolution images. We achieve state-of-the-art performance on the common face detection benchmark datasets.

1. Introduction

Face detection is one of the fundamental problems in computer vision and pattern recognition. It plays an important role in face based applications, since accurate and efficient face detection usually needs to be done first. With the great progress, face detection has been successfully applied in our daily life. However, there are still some tough challenges in the uncontrolled face detection problem. The challenges mainly come from two requirements for face detectors: (1) The large variation of facial changes requires face detectors to accurately address a complicated face and non-face classification problem; (2) The large search space of arbitrary face positions and sizes further imposes a time efficiency requirement. These two requirements are conflicting, since high-accuracy face detectors tend to be computationally expensive.

To meet these challenges, face detection has been studied mainly in two different ways. One way is the cascade based methods and it starts from the pioneering work of Viola-Jones face detector [1]. Since then, the boosted cascade with simple features becomes the most popular and effective design for practical face detection. A number of improvements to the Viola-Jones face detector have been proposed in the past decade [2], which can be seen as a history of more efficiently sampling the output space to a solvable scale and more effectively evaluation per configuration. The other way is Convolutional Neural Network (CNN) [3] based methods and with the development of deep learning techniques, the CNN has been successfully applied in face detection tasks. Recently, some works based on R-CNN [4] demonstrate state-of-the-art performance on face detection tasks.

However, these two ways focus on different aspects. The former pays more attention to efficiency while the latter cares more about accuracy. To make face detector perform well on both speed and accuracy, one natural idea is to combine the advantages of them. Therefore, cascade CNN based methods [5] are proposed that put features learned by CNN into cascade framework so as to boost the performance and keep efficient. However, there are three problems in cascaded CNN based methods: (1) Their speed is negatively related to the number of faces on the image. The speed would dramatically degrade as the number of faces increases; (2) The cascade based detectors optimize each component separately, making the training process extremely complicated and the final model sub-optimal; (3) For the VGA-resolution images, their runtime efficiency on the CPU is about 14 FPS, which is not fast enough to reach the real-time speed (25 FPS).

Therefore, it is still one of the remaining open issues for practical face detectors to achieve CPU real-time speed as well as maintain high performance. In this work, we develop a state-of-the-art face detector with CPU real-time speed. The core idea is to shrink the input image and focus on detecting small faces.

* Corresponding author at: CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
E-mail addresses: shifeng.zhang@nlpr.ia.ac.cn (S. Zhang), xiangyu.zhu@nlpr.ia.ac.cn (X. Zhu), zlei@nlpr.ia.ac.cn (Z. Lei), xiaobo.wang@nlpr.ia.ac.cn (X. Wang), hailin.shi@nlpr.ia.ac.cn (H. Shi), szli@nlpr.ia.ac.cn (S.Z. Li).

https://doi.org/10.1016/j.neucom.2018.01.012
0925-2312/

Reducing the high-resolution input image into the low-resolution image can significantly improve the detection speed, but it also results in smaller faces that need to pay more attention in order to maintain high performance. Specifically, our DCFPN has a lightweight-but-powerful network with the consideration of efficiency and accuracy. To improve the recall rate of small faces, a dense anchor strategy and a scale-aware anchor matching scheme are introduced. Besides, we present a fair L1 loss to locate small faces well. Consequently, for VGA images to detect faces bigger than 40 pixels, our face detector can run at 30 FPS on a single CPU core and 250 FPS on a GPU card. More importantly, the speed of DCFPN is invariant to the number of faces on the image.

A preliminary version of this work has been published on Chinese Conference on Biometric Recognition (CCBR) 2017.[1] Comparing with the preliminary version, this paper proposes a new scale-aware anchor matching scheme and further improves the state-of-the-art performance. For clarity, the main contributions of this work can be summarized as four-fold:

- We design a lightweight-yet-powerful fully convolution network with the consideration of efficiency and accuracy for the face detection task.
- We present a dense anchor strategy and a scale-aware anchor matching scheme to improve the recall rate of small faces.
- We introduce a fair L1 loss function that directly regresses box's relative center and size in order to locate small faces well.
- We achieve state-of-the-art performance on the common face detection benchmark datasets with CPU real-time speed.

2. Related work

Face detection approaches can be roughly divided into two different categories. One is based on hand-craft features, and the other one is built on CNN. This section briefly reviews them and refer more detailed survey to [2,6,7].

Hand-craft based methods. Previous face detection systems are mostly based on the hand-crafted features. The milestone work of Viola-Jones [1] proposes to use Haar feature, Adaboost learning and cascade inference for face detection. After that, many subsequent works focus on new local features [8,9], new boosting algorithms [10–12] and new cascade structures [13–15]. Besides the cascade framework, the seminal work deformable part model (DPM) [16] is introduced into the face detection task by [17–21] , which use supervised parts, more pose partition, better training or more efficient inference to achieve better performance.

CNN based methods. Recently, CNN based methods have showed advantages in face detection. CCF [22] uses boosting on top of CNN features for face detection. Farfade et al. [23] fine-tune CNN model trained on 1 k ImageNet classification task for face and background classification task. Faceness [24] trains a series of CNNs for facial attribute recognition to detect partially occluded faces. CascadeCNN [5] uses six cascaded CNNs to efficiently reject backgrounds in three stages. STN [25] proposes a new Supervised Transformer Network and a ROI convolution for face detection. Similar to Chen et al. [26], MTCNN [27] presents a multi-task cascaded CNNs based framework for joint face detection and alignment. UnitBox [28] introduces a new intersection-over-union loss function. CMS-RCNN [29] uses Faster R-CNN in face detection with body contextual information. Convnet [30] integrates CNN with 3D face model in an end-to-end multi-task learning framework.

Generally, hand-craft based methods are able to achieve CPU real-time speed, but they are not accurate enough for the uncontrolled face detection problem. With learned feature and classifier directly from the image, CNN based methods can differentiate faces from highly cluttered backgrounds, while they are too time-consuming to reach real-time speed. Notably, our proposed DCFPN is able to achieve real-time speed on the CPU devices as well as maintain state-of-the-art detection performance.

Table 1
The receptive field of the last convolutional layer and the default anchor of our DCFPN.

Receptive field	75×75, 107×107, 139×139, 171×171, 203×203, 235×235
Default anchor	16×16, 32×32, 64×64, 128×128, 256×256

3. Densely connected face proposal network

This section presents detail of DCFPN. It includes four key contributions that make it accurate and efficient for face detection: lightweight-but-powerful architecture, dense anchor strategy, scale-aware anchor matching scheme and fair L1 loss.

3.1. Lightweight-but-powerful architecture

The architecture of DCFPN encourages feature reuse and leads to a substantial reduction of parameters. As illustrated in Fig. 1 , it consists of two parts.

Rapidly Digested Convolutional Layers. It is designed for high efficiency via quickly reducing the input image spatial size by 16 times with narrow but large kernels. On one side, face detection is a two classification problem and does not require very wide network, hence the narrow kernels is powerful enough and can result in faster running speed, especially for CPU devices. On the other side, the large kernels are to alleviate the information loss brought by spatial size reducing.

Densely Connected Convolutional Layers. Inspired by Huang et al. [31], each layer in DCCL is directly connected to every other layer in a feed-forward fashion. It ends with two micro inception layers. There are two motivations behind the design of DCCL. Firstly, the DCCL is designed to enrich the receptive field of the last convolutional layer that is used to predict the detection results. As listed in Table 1, the last convolutional layer of DCFPN has a large scope of receptive field from 75 to 235 pixels, which is consistent with our default anchors and is important for the network to learn visual patterns for different scales of faces. Secondly, the DCCL aims at combining coarse-to-fine information across deep CNN models to improve the recall rate and precision of detection. Deep and shallow CNN features are really complementary for detection task, since the information of the interest region is distributed over all levels of the convolution network with multiple level abstraction, and they should be well organized.

To sum up, our lightweight-but-powerful architecture consists of RDCL and CCL. The former is designed to achieve CPU real-time speed. The latter aims at enriching the receptive fields and combining coarse-to-fine information across different layers to handle faces of various scales.

3.2. Dense anchor strategy

As listed in Table 1, we use 5 default anchors that are associated with the last convolutional layer. Hence, these 5 default anchors have the same tiling interval on the image (i.e., 16 pixels). It is obviously that there is a tiling density imbalance problem. Comparing with large anchors (i.e., 64×64, 128×128 and 256×256), small anchors (i.e., 16×16 and 32×32) are too sparse, which results in low recall rate of small faces.

To improve the recall rate of small faces, we proposed the dense anchor strategy for small anchor. Specifically, without our dense anchor strategy, there are 5 anchors for every receptive filed center

[1] http://ccbr2017.org/

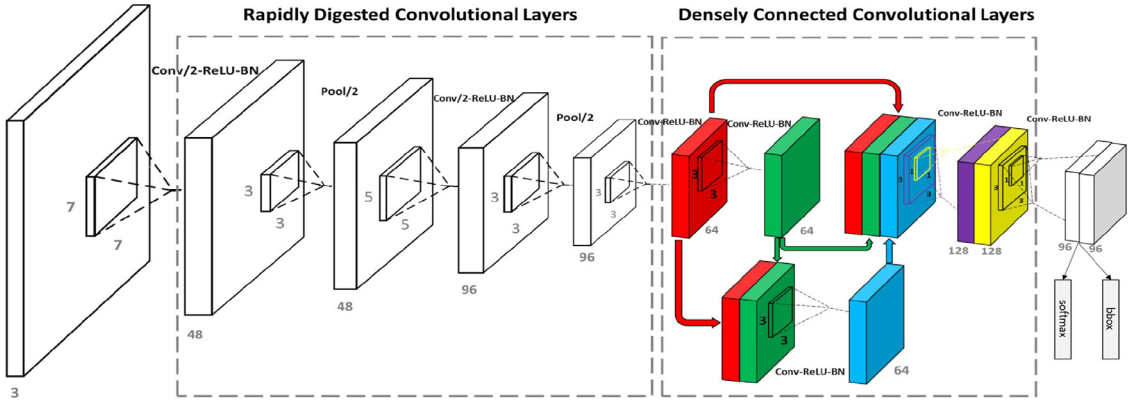


Fig. 1. Illustration of the structure of our Densely Connected Face Proposal Network (DCFPN). It consists of the Rapidly Digested Convolutional Layers (RDCL) and the Densely Connected Convolutional Layers (DCCL). RDCL is designed for high efficiency by quickly reducing the input image spatial size by 16 times with narrow but large kernels; DCCL is aimed at high accuracy by extracting information from different sizes of receptive field at multiple levels of abstraction.

(Fig. 2(a)). To densify one type of anchors, our strategy uniformly tiles several anchors around the center of one receptive field instead of only tiling one. As illustrated in Fig. 2(b) and2(c), the sampling interval of 16×16 and 32×32 anchor are densed to 4 and 8 pixels, respectively. Consequently, for every receptive filed center, there are total 23 anchors (16 from 16×16 anchor, 4 from 32×32 anchor and 3 from the rest three anchors). The dense anchor strategy is crucial to improve the recall rate of small faces.

3.3. Scale-aware anchor matching scheme

During training, a binary label (i.e., positive or negative) need to be assigned to each anchor. Current anchor matching method first matches the anchors to the faces with the best Jaccard overlap and then matches the anchors to any face with Jaccard overlap higher than a threshold θ. However, anchor scales are discrete while face scales are continuous, these faces whose scales distribute away from anchor scales cannot match enough anchors, especially for small faces, leading to their low recall rate. To solve this problem, we propose a scale-aware anchor matching scheme, which treats small and large faces differently. It uses the existing matching method for large faces, while applies the proposed matching method for small faces. The procedure is as follow:

Algorithm: Scale-aware anchor matching scheme

Input: n anchor boxes $\rightarrow A_{n\times 4}$, m gt boxes $\rightarrow G_{m\times 4}$
Output: all matched pairs $\rightarrow P$
1: Compute IoU between anchor and gt boxes: $O_{n\times m} = IoU(A, G)$
2: $R_{n\times 1} = ArgmaxAlongRow(O)$, $C_{m\times 1} = ArgmaxAlongColumn(O)$
3: Match each gt to the $anchor$ with the best IoU: $P \leftarrow MakePair(A(C), G)$
4: for $a \in A$ do
5: for $g \in G$ do
/*- - - - - - - - Large faces: the Jaccard overlap condition - - - - - - - -*/
6: if $\sqrt{area(g)} > 20$ then
7: if $O(a, g) \geq 0.5$ and $R(a) = g$ then
8: $P \leftarrow MakePair(a, g)$
/*- - - - - - - - Small faces: the center distance condition - - - - - - - -*/
9: if $\sqrt{area(g)} \leq 20$ and $size(a) = 16 \times 16$ then
10: if $center(a)$ within the scaled down 0.75 box of g and $R(a) = g$ then
11: $P \leftarrow MakePair(a, g)$

Our scheme is roughly the same as existing matching method except that small faces use the center distance condition (clarified at lines 9–11) instead of the Jaccard overlap. Specifically, for the small face (< 20 pixels), we first scale it down by 0.75 times to get a shrunk box, then match those 16×16 anchors whose center are within the shrunk box to this small face. This scheme ensures that small faces can match enough positive anchor. Anchors that do not be matched are negative anchors.

3.4. Fair L1 loss function

As formulated in Eq. (1), our model is jointly optimized by two loss functions, L_{cls} and L_{reg}, which compute errors of score and coordinate, respectively.

$$L(\{p_i\}, \{t_i\}) = \frac{\lambda}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{1}$$

Where i is the anchor index and p_i is the predicted probability that the anchor i is a face. The groundtruth label p_i^* is 1 if the anchor is positive, 0 otherwise. As formulated in Eq. (2), t_i and t_i^* is a vector representing the 4 parameterized coordinates of the predicted box and the GT box. $p_i^* L_{reg}$ means the regression loss is activated only for positive anchors and disabled otherwise. The two terms are normalized by N_{cls} and N_{reg} and weighted by a balancing parameter λ. In our implementation, the *cls* and *reg* term are normalized by the number of positive and negative anchors, and the number of positive anchors, respectively. Because of the imbalance between the number of positive and negative anchors, we set $\lambda = 10$ to balance these two loss terms.

We adopt a 2-class softmax loss for L_{cls}. As for L_{cls}, to locate small faces well, we propose the fair L1 loss that directly regresses the predicted boxâ;;s relative center coordinate and its width and height as follows:

$$\begin{aligned} t_x &= x - x^a, & t_y &= y - y^a, & t_w &= w, & t_h &= h \\ t_x^* &= x^* - x^a, & t_y^* &= y^* - y^a, & t_w^* &= w^*, & t_h^* &= h^* \end{aligned} \tag{2}$$

where x, y, w, and h denote the box's center and its size. Variables x, x^a, and x^* are for the predicted box, anchor box, and GT box, respectively (likewise for y,w,h). The scale normalization is implemented to have scale-invariance loss value as Eq. (3):

$$L_{reg}(t, t^*) = \sum_{j\in\{x,y,w,h\}} fair_{L_1}(t_j - t_j^*) \tag{3}$$

in which

$$fair_{L_1}(z_j) = \begin{cases} |z_j| \ / \ gt_w & if j \in \{x, w\} \\ |z_j| \ / \ gt_h & otherwise \end{cases} \tag{4}$$

where gt_w and gt_h denote the GT box's width and height. Compare with [32], the fair L1 loss directly regresses box's relative center and size, and implements scale normalization when computing loss value, which is crucial to locate small faces well.

3.5. Training data and implementation details

Training data. Our model is trained on 12,880 images from the WIDER FACE training set. To enrich the training dataset, each

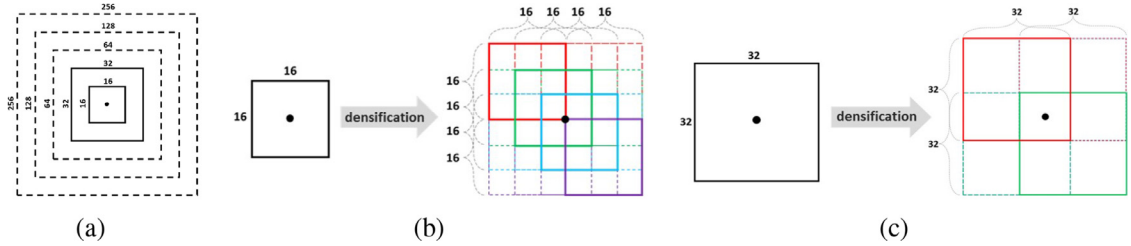


Fig. 2. (a) 5 default anchors at one receptive filed center. (b) 16×16 anchor densification. (c) 32×32 anchor densification. In (b) and (c), only part of anchors are drawn with solid line of different colors, while the others are with dotted line of corresponding color. Best viewed in color. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article).

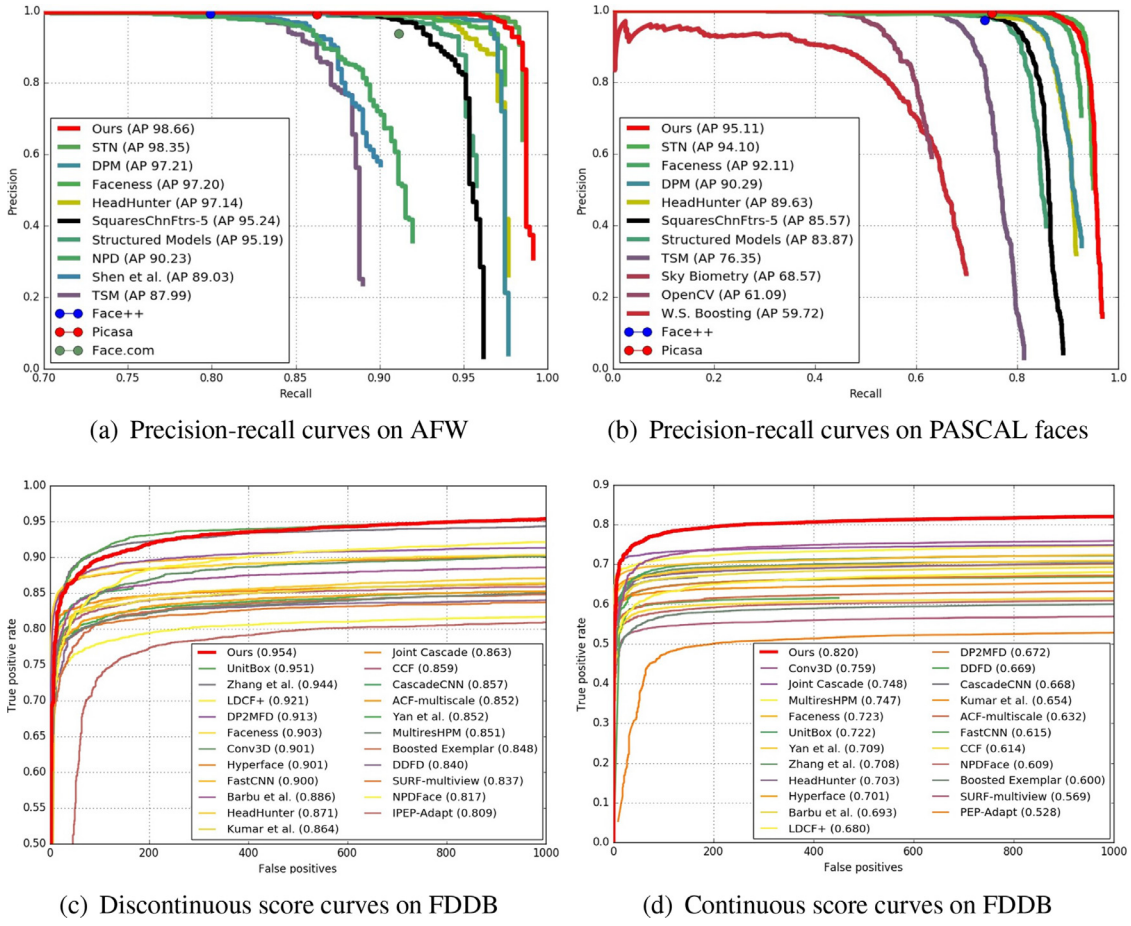


Fig. 3. Evaluation results of DCFPN.

training image is sequentially processed by the color distortion, random cropping, scale transformation and horizontal flipping, eventually getting a 512×512 square sub-image from original image. The GT bounding box is ignored if its center coordinate is located outside of the square sub-image. In the training process, each mini-batch is collected randomly from 48 images. For each mini-batch, all of the positive anchors and half of the negative anchors are used to train our model.

Implementation details. The DCFPN is trained end-to-end via using the standard back-propagation and stochastic gradient descent (SGD). We randomly initialize all layers by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. We use 0.9 momentum and 0.0005 weight decay. The maximum number of iterations is 100 k, and the initial learning rate is set to 0.1 and multiplied by 0.1 every 20 k iterations. Our model is implemented in Caffe framework [33].

4. Experiments

In this section, we firstly analyze our model in an ablative way, then evaluate it on the common face detection benchmarks, finally introduce its runtime efficiency.

4.1. Model analysis

We carry out extensive ablation experiments on the FDDB dataset to analyze our model. For all the experiments, we use the same settings, except for specified changes to the components. To

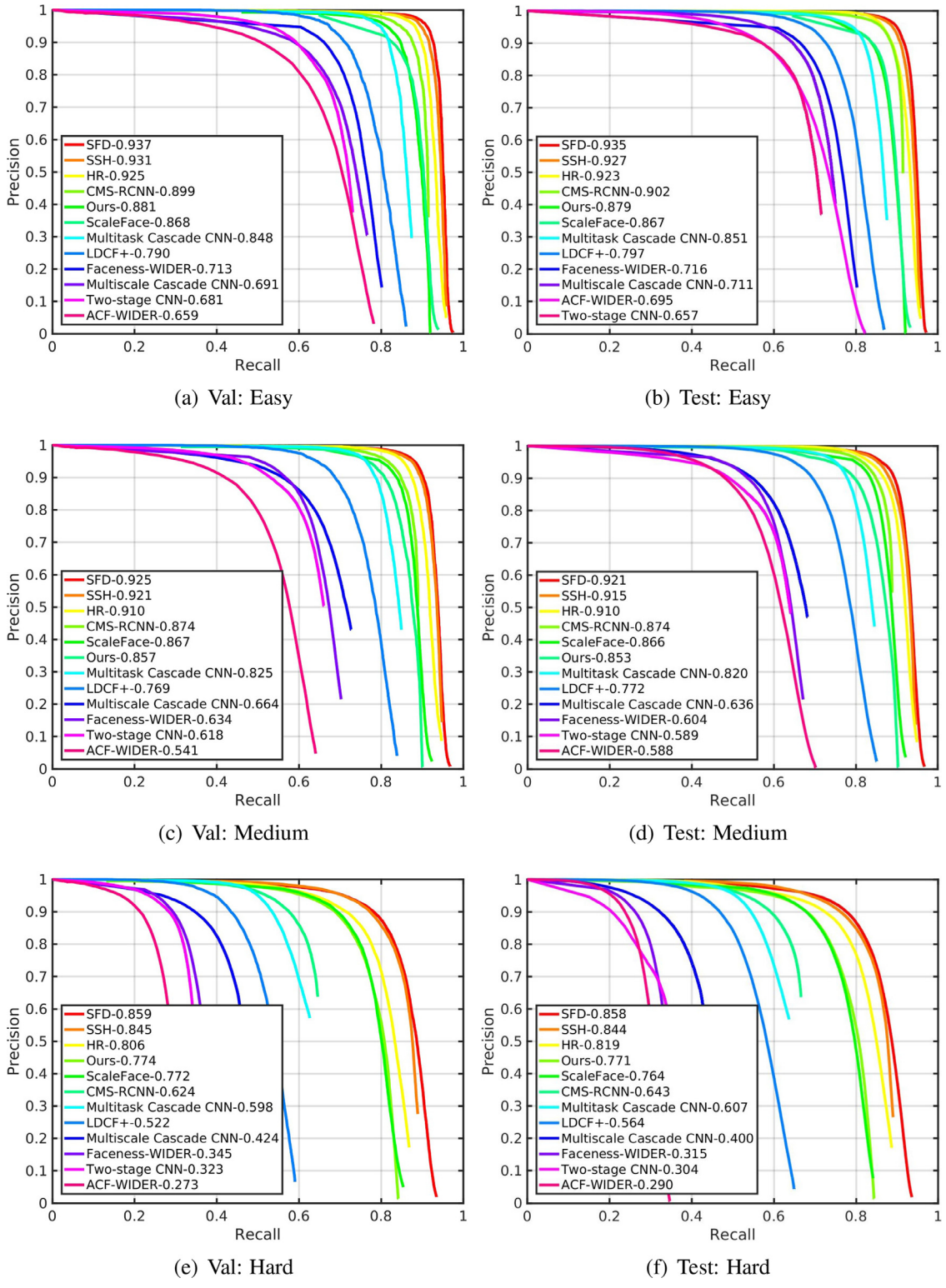
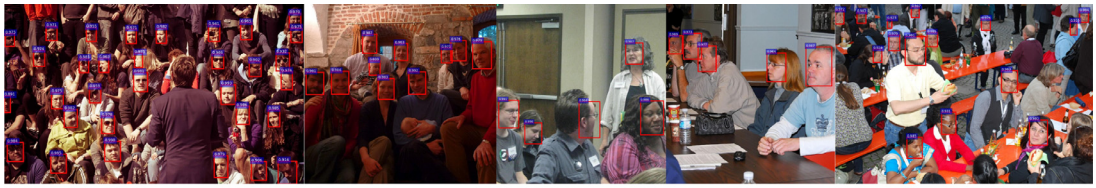


Fig. 4. Precision-recall curves on WIDER FACE validation and test sets.

better understand DCFPN, we ablate each component one after another to examine how each proposed component affects the final performance. Firstly, we replace the fair L1 loss with smooth L1 loss. Meantime, the target of regression is the same as RPN. Secondly, we do not use the scale-aware anchor matching scheme during training stage. Finally, we ablate the dense anchor strategy.

Some promising conclusions can be summed up according to the ablative results. Firstly, the comparison between the first and second columns in Table 2 indicates that the fair L1 loss effectively increases the TPR performance by 0.4%, owning to locating small faces well. Secondly, the scale-aware anchor matching scheme is proposed to ensure small faces to match enough anchors, and the comparison between the second and third columns in Table 2 demonstrates that it rises the TPR performance from 94.5 to 95.0%, attributing to the higher recall rate of small faces. Finally, our dense anchor strategy is used to increase the density



(a) AFW

(b) PASCAL face

(c) FDDB

(d) WIDER FACE

Fig. 5. Qualitative results of DCFPN.

of small anchors (i.e., 16×16 and 32×32). From the results listed in Table 2, we can observe that the TPR on FDDB is reduced from 94.5 to 93.7% after ablating the dense anchor strategy. The sharp decline (i.e., 0.8%) demonstrates the effectiveness of the proposed dense anchor strategy.

4.2. Evaluation on benchmark

This section presents the face detection bechmarking using our proposed DCFPN approach. We compare our results with those of other leading methods.

Table 2
Ablative results on FDDB. Accuracy means the true positive rate at 1000 false positives.

Contribution	DCFPN			
Designed architecture?	✓	✓	✓	✓
Dense anchor strategy?	✓	✓	✓	
Scale-aware scheme?	✓	✓		
Fair L1 loss?	✓			
Accuracy	95.4	95.0	94.5	93.7

Table 3
Overall CPU inference time and TPR compared on different methods. TPR(%) means the true positive rate at 1000 false positives on FDDB dataset. For STN [25], its TPR is the true positive rate at 179 false positives and with ROI convolution, it can speed up from 10 to 30 FPS with only 0.6% recall rate drop.

Approach	Resolution	Device	GHz	TPR(%)	FPS
ACF [8]	640×480	Intel i7-3770	3.40	85.2	20
CasCNN [5]	640×480	Intel E5-2620	2.00	85.7	14
FaceCraft [44]	640×480	N/A	N/A	90.8	10
STN [25]	640×480	Intel i7-4770K	3.50	91.5	10
MTCNN [27]	N/A	N/A	2.60	94.4	16
Ours	640×480	Intel E5-2660v3	2.60	95.4	30

AFW database [21]. It contains 205 images with 473 labeled faces from Flickr. We evaluate our detector on this dataset and compare with well known research and commercial face detectors. Research detectors include [18,20,21,24,34,35]. Commercial detectors include Face.com, Face++ and Google Picasa. As can be observed from Fig. 3(a), our method outperforms strong all others by a large margin. Fig. 5(a) shows some examples of the face detection results using the proposed DCFPN on the AFW dataset.

PASCAL face database [20]. It consists of 851 images with 1335 labeled faces and is collected from the test set of PASCAL person layout dataset, which is a subset of PASCAL VOC. There are large face appearance and pose variations in this dataset. Note that this dataset is designed for person layout detection and head annotation is used as face annotation. The cases when the face is occluded are common. Fig. 3(b) shows the precision-recall curves on this dataset and our DCFPN method outperforms all other detectors. Fig. 5(b) shows some examples of the face detection results using the proposed DCFPN on the PASCAL face dataset.

FDDB database [36]. It has 5171 faces in 2845 images taken from news articles on Yahoo websites. FDDB uses ellipse face annotations while our DCFPN outputs rectangle outputs. This inconsistency has a great impact to the continuous score. For a more fair comparison under the continuous score evaluation, we regress a transformation matrix according to the ellipse and rectangle annotations, and then transform our rectangle outputs to ellipse outputs. As shown in Fig. 3(c) and3(d)), our DCFPN performs better than all of the published face detection methods, demonstrating that DCFPN is able to robustly detect unconstrained faces. Fig. 5(c) shows some examples of the face detection results using the proposed DCFPN on the FDDB dataset.

WIDER FACE database [37]. It has 32,203 images and 393,703 faces with a high degree of variability in scale, pose and occlusion. The database is divided into training (40%), validation (10%) and testing (50%) set with three levels of detection difficulty (Easy, Medium and Hard subset). The images and annotations of training and validation set are available online, while the annotations of testing set are not released and the results are sent to the database server for receiving the precision-recall curves. Our DCFPN is trained only on the training set and tested on both validation and testing set against recent face detection methods [8,24,27,29,37–43]. Fig. 4 shows the precision-recall curves and mAP values. Our DCFPN, based on an extremely lightweight network, achieves promising performance, i.e. 0.881 (Easy), 0.857 (Medium) and 0.774 (Hard) for validation set, and 0.879 (Easy), 0.853 (Medium) and 0.771 (Hard) for testing set. Among the lightweight detectors, our model outperforms others by a large margin across the three subsets, especially on the hard subset which mainly consists of small faces. Besides, our DCFPN performs better than some detectors based on ResNet, such as CMS-RCNN [29] and ScaleFace [42]. These results demonstrate that DCFPN achieves good trade-off between effectiveness and efficiency via detecting small faces. Fig. 5(d) shows some examples of the face detection results using the proposed DCFPN on the WIDER FACE dataset.

4.3. Runtime efficiency

CNN based methods have always been accused of their runtime efficiency, since in most CPU based applications, they are not fast enough. As listed in Table 3, comparing with other methods, our DCFPN is efficient and accurate enough to meet practical requirements. Specifically, due to the great ability to detect small faces, our proposed DCFPN can shrink the test images by a few times and detect small faces, in order to reach real-time speed as well as maintain high performance. This means that faces can be efficiently detected by shrinking the test image and detecting smaller ones. With this advantage, our DCFPN can detect faces bigger than 40 pixels at 30 FPS on a 2.60 GHz CPU for the VGA-resolution images. Besides, our method with only 3.2 M parameter can directly run on a GPU card at 250 FPS for the VGA-resolution images.

5. Conclusion

In this paper, we propose a novel face detector with CPU real-time speed as well as high performance. On the one hand, our DCFPN has a lightweight-but-powerful framework that can well incorporate CNN features from different sizes of receptive field at multiple levels of abstraction. On the other hand, the dense anchor strategy, the scale-aware anchor matching strategy and the fair L1 loss function are proposed to handle small faces well. The state-of-the-art performance on common face detection datasets shows its ability to detect faces in the uncontrolled environment. The proposed detector is very fast, achieving 30 FPS to detect faces bigger than 40 pixels on CPU and can be accelerated to 250 FPS on GPU for the VGA-resolution images.

Acknowledgs

This work was supported by the National Key Research and Development Plan (Grant No. 2016YFC0801002), the Chinese National Natural Science Foundation Projects Nos. 61473291, 61572501, 61502491, 61572536 and AuthenMetric R&D Funds.

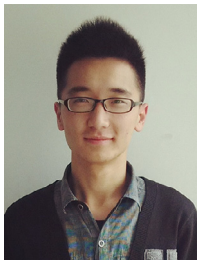
Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.neucom.2018.01.012

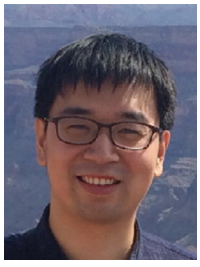
References

[1] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[2] C. Zhang, Z. Zhang, A survey of recent advances in face detection, Technical Report MSR-TR-2010-66, 2010.
[3] Y. Lecun, Y. Bengio, Convolutional networks for images, speech, and time-series, in: The Handbook of Brain Theory and Neural Networks, MIT press, 1995.
[4] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[5] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.
[6] M.H. Yang, D.J. Kriegman, N. Ahuja, Detecting faces in images: a survey, IEEE Tran. Pattern Anal. Mach. Intell. 24 (1) (2002) 34–58.
[7] S. Zafeiriou, C. Zhang, Z. Zhang, A survey on face detection in the wild: past, present and future, Comput. Vis. Image Underst. 138 (2015) 1–24.
[8] B. Yang, J. Yan, Z. Lei, S.Z. Li, Aggregate channel features for multi-view face detection, in: Proceedings of the International Joint Conference on Biometrics, 2014, pp. 1–8.
[9] L. Zhang, R. Chu, S. Xiang, S. Liao, S.Z. Li, Face detection based on multi-block LBP representation, in: Proceedings of the International Conference on Biometrics, 2007, pp. 11–18.
[10] C. Huang, H. Ai, Y. Li, S. Lao, High-performance rotation invariant multiview face detection, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 671–686.
[11] M. Jones, P. Viola, Fast multi-view face detection, MERL 3 (2003) 14.
[12] C. Zhang, J.C. Platt, P.A. Viola, Multiple instance boosting for object detection, in: Proceedings of the Neural Information Processing Systems, 2005, pp. 1417–1424.
[13] L. Bourdev, J. Brandt, Robust object detection via soft cascade, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2, 2005, pp. 236–243.
[14] S.Z. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, H. Shum, Statistical learning of multi-view face detection, in: Proceedings of the European Conference on Computer Vision, 2002.
[15] R. Xiao, L. Zhu, H.-J. Zhang, Boosting chain learning for object detection, in: Proceedings of the International Conference on Computer Vision, 2003, pp. 709–715.
[16] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[17] G. Ghiasi, C.C. Fowlkes, Occlusion coherence: detecting and localizing occluded faces, arXiv:1506.08347 (2015).
[18] M. Mathias, R. Benenson, M. Pedersoli, L. Van Gool, Face detection without bells and whistles, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 720–735.
[19] J. Yan, Z. Lei, L. Wen, S.Z. Li, The fastest deformable part model for object detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2014a, pp. 2497–2504.
[20] J. Yan, X. Zhang, Z. Lei, S.Z. Li, Face detection by structural models, Image Vis. Comput. 32 (10) (2014b) 790–799.
[21] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2012, pp. 2879–2886.
[22] B. Yang, J. Yan, Z. Lei, S.Z. Li, Convolutional channel features, in: Proceedings of the International Conference on Computer Vision, 2015, pp. 82–90.
[23] S.S. Farfade, M.J. Saberian, L.-J. Li, Multi-view face detection using deep convolutional neural networks, in: Proceedings of the ACM International Conference on Multimedia Retrieval, 2015, pp. 643–650.
[24] S. Yang, P. Luo, C.-C. Loy, X. Tang, From facial parts responses to face detection: a deep learning approach, in: Proceedings of the International Conference on Computer Vision, 2015.
[25] D. Chen, G. Hua, F. Wen, J. Sun, Supervised transformer network for efficient face detection, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 122–138.
[26] D. Chen, S. Ren, Y. Wei, X. Cao, J. Sun, Joint cascade face detection and alignment, in: Proceedings of the European Conference on Computer Vision, 2014, pp. 109–122.
[27] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, Joint face detection and alignment using multitask cascaded convolutional networks, IEEE Signal Process. Lett. 23 (10) (2016) 1499–1503.
[28] J. Yu, Y. Jiang, Z. Wang, Z. Cao, T. Huang, UnitBox: an advanced object detection network, in: Proceedings of the ACM International Conference on Multimedia, 2016, pp. 516–520.
[29] C. Zhu, Y. Zheng, K. Luu, M. Savvides, CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection, arXiv:1606.05413 (2016).
[30] Y. Li, B. Sun, T. Wu, Y. Wang, Face detection with end-to-end integration of a convnet and a 3d model, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 420–436.
[31] G. Huang, Z. Liu, K.Q. Weinberger, L. van der Maaten, Densely connected convolutional networks, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2017.
[32] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Proceedings of the Neural Information Processing Systems, 2015.
[33] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in: Proceedings of the ACM International Conference on Multimedia, 2014.
[34] S. Liao, A.K. Jain, S.Z. Li, A fast and accurate unconstrained face detector, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 211–223.
[35] X. Shen, Z. Lin, J. Brandt, Y. Wu, Detecting and aligning faces by image retrieval, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2013, pp. 3460–3467.
[36] V. Jain, E.G. Learned-Miller, FDDB: a benchmark for face detection in unconstrained settings, UMass Amherst Report, 2010.
[37] S. Yang, P. Luo, C.C. Loy, X. Tang, Wider face: a face detection benchmark, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2016, pp. 5525–5533.
[38] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S.Z. Li, S^3FD: single shot scale-invariant face detector, in: Proceedings of the International Conference on Computer Vision, 2017.
[39] M. Najibi, P. Samangouei, R. Chellappa, L.S. Davis, SSH: single stage headless face detector, in: Proceedings of the International Conference on Computer Vision, 2017.
[40] P. Hu, D. Ramanan, Finding tiny faces, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2017, pp. 1522–1530.
[41] C. Zhu, Y. Zheng, K. Luu, M. Savvides, CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection, CoRR (2016). abs/1606.05413
[42] S. Yang, Y. Xiong, C.C. Loy, X. Tang, Face detection through scale-friendly deep convolutional networks, arXiv:1706.02863 (2017).
[43] E. Ohn-Bar, M.M. Trivedi, To boost or not to boost? On the limits of boosted trees for object detection, in: Proceedings of the Computer Vision and Pattern Recognition, 2016, pp. 3350–3355.
[44] H. Qin, J. Yan, X. Li, X. Hu, Joint training of cascaded CNN for face detection, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2016, pp. 3456–3465.



Shifeng Zhang received the B.S. degree from the University of Electronic Science and Technology of China (UESTC), in 2015. Since September 2015, he has been a Ph.D. candidate at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science (CASIA). His research interests include computer vision, pattern recognition, especially for object detection, face detection, pedestrian detection.



Xiangyu Zhu received the B.S. degree in Sichuan University (SCU) in 2012, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences, in 2017, where he is currently an assistant professor. His research interests include pattern recognition and computer vision, in particular, image processing, 3D face model, face alignment and face recognition.

Zhen Lei received the B.S. degree in automation from the University of Science and Technology of China, in 2005, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2010, where he is currently an associate professor. He has published more than 100 papers in international journals and conferences. His research interests are in computer vision, pattern recognition, image processing, and face recognition in particular. He served as an area chair of the International Joint Conference on Biometrics in 2014, the IAPR/IEEE International Conference on Biometric in 2015, 2016, 2018, and the IEEE International Conference on Automatic Face and Gesture Recognition in 2015. He is a senior member of the IEEE.

Xiaobo Wang received the B.S. and M.E. degrees from the School of Science, Tianjin University, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include machine learning, deep learning, data mining, and computer vision.

Hailin Shi is currently an assistant researcher at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Science (CASIA). He receives his B.S. and M.S. degrees from University of Paris 6. He received his Ph.D. degree from Institute of Automation, Chinese Academy of Science (CASIA) in 2017. His research interests include deep learning, computer vision, face recognition, person re-identification.

Stan Z. Li received the B.Eng. degree from Hunan University, China, the M.Eng. degree from National University of Defense Technology, China, and the Ph.D. degree from Surrey University, United Kingdom. He is currently a professor and the director of Center for Biometrics and Security Research (CBSR), Institute of Automation, Chinese Academy of Sciences (CASIA). He was with Microsoft Research Asia as a researcher from 2000 to 2004. Prior to that, he was an associate professor in the Nanyang Technological University, Singapore. His research interests include pattern recognition and machine learning, image and vision processing, face recognition, biometrics, and intelligent video surveillance. He has published more than 300 papers in international journals and conferences, and authored and edited eight books. He was an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and is acting as the editor-in-chief for the Encyclopedia of Biometrics. He served as a program co-chair for the International Conference on Biometrics 2007, 2009, 2013, 2014, 2015, 2016 and 2018, and has been involved in organizing other international conferences and workshops in the fields of his research interest. He was elevated to IEEE fellow for his contributions to the fields of face recognition, pattern recognition and computer vision and he is a member of the IEEE Computer Society.