IMAGE PROCESSING, PATTERN RECOGNITION

FaceDetectNet: Face detection via fully-convolutional network

V.S. Gorbatsevich [1], A.S. Moiseenko [1,2], Y.V. Vizilter [1]
[1] State Research Institute of Aviation Systems (GosNIIAS), Moscow, Russia;
[2] Moscow Institute of Physics and Technology (MIPT), Moscow, Russia

Abstract

Face detection is one of the most popular computer vision tasks. There are a lot of face detection approaches proposed including different CNN-based techniques, but the problem of optimal balancing between detection quality and computational speed is still relevant. In this paper we propose new CNN-based solution for face detection called FaceDetectNet. Our CNN architecture is based on ideas of YOLO/DetectNet and GoogleNet architecture supported with some new tools and implementation details created especially for our face detection application. We propose: original iterative proposal clustering (IPC) algorithm for aggregation of output face proposals formed by CNN and the 2-level "weak pyramid" providing better detection quality on the testing sets containing both small and huge images. Our face detection approach is close to previously proposed SSD-based face detection, but the principal difference is that we use the deep features of top hidden CNN layer for forming the face proposals of any size. Thus we utilize the global semantic and context information for improving the detection quality for small faces. Our FaceDetectNet is trained and tested on the most challenging WIDER FACE detection benchmark. Our algorithm achieves the average precision (AP) 0.69 on the WIDER FACE hard level, and thus outperforms all competitive detectors on the Hard level besides the HR state-of-the-art solution. Note that HR solution is based on essentially deeper and slower CNN, while our FaceDetectNet can work in real-time on the NVIDIA GeForce 1080 GPU. On the other hand, SSD-based face detector with comparable CNN parameters provides AP 0.625 only on the WIDER FACE hard level. So, our approach provides the best quality with reasonable computational speed.

Keywords: CNN, face detection, DetectNet, YOLO.

Citation: Gorbatsevich VS, Moiseenko AS, Vizilter YV. FaceDetectNet: Face detection via fully-convolutional network. Computer Optics 2019; 43(1): 63-71. DOI: 10.18287/2412-6179-2019-43-1-63-71.

Acknowledgements: This work is supported by grant from Russian Science Foundation (Project No. 16-11-00082).

Introduction

Face detection is one of the most popular computer vision issues. The objective of face detection stage is to find and locate faces in images providing the facial bounding boxes for further automatic image processing: face recognition, gender recognition, face pose evaluation, expression recognition, 3D face modelling and so on. The main requirements to face detection algorithms are the high quality (robustness and precision) and high computational speed.

Currently, we have a lot of face detection approaches: from classical Voila-Jones like algorithms to modern CNN-based, but the problem of providing the optimal quality/speed ratio is still relevant until these days. For example, Viola-Jones style detectors are fast and correct for frontal faces, but in unconstraint conditions (pose variations, illumination, occlusions, expression variations, blur, etc.) they fail in most cases. Part based models are essentially better in the wild, but best detectors of this type are too slow. CNN-based models are the best in quality, but they are usually extremely slow too.

In this paper we propose the FaceDetectNet – new CNN-based solution for face detection based on ideas of well-known YOLO/DetectNet ("You only look") architecture [1, 2]. Our face detection approach is close to SSD [3], but the principal difference is that we use the deep features of top hidden CNN layer for forming the facial proposals of any size. So, the each cell of FaceDetectNet output 8×8 grid is supported by the input about 400×400 image region. Thus we utilize the global semantic and context information for improving the detection quality for small faces. We also propose two new algorithms for improved implementation of DetectNet scheme in the face detection applications: iterative proposal clustering (IPC) algorithm for aggregation of output face proposals formed by CNN and "weak" image pyramid providing better detection quality. The resultant solution is trained and tested on the WIDER FACE dataset. Our deep network can work in real-time on the NVIDIA GeForce 1080 GPU and we achieve the AP 0,69 on WIDER FACE hard level, which is very close to state-of-the-art results provided by essentially slower approaches [4, 5].

The residual of this paper is organized as follows. In the second section we describe the related works in face and object detection. In the third section we briefly introduce the main ideas of YOLO/DetectNet CNN architecture, and describe data representation and loss function in our FaceDetectNet. In the fourth section we describe our clustering and score normalization techniques. In the last section we present the details of FaceDetectNet implementation and experimental results.

1. Related works

Face detection problem is currently well studied. There are a lot of different face detection approaches including CNN-based algorithms.

We traditionally start our brief overview of face detection algorithms from the Viola-Jones approach [6], which was the first practically applied real-time face detection technique providing the enough accuracy. It is based on adaptive boosting, Haar-like features and cascading computational scheme. Different modifications of this approach [7] are still of use until these days mainly due to their extremely high computational speed. The main disadvantage of this approach is its low robustness relative to face pose and various image acquisition conditions. This was the reason for appearance of other classes of face detection algorithms based on more complex models such as DPM [8], SURF cascades [9] and so on.

In 2012 the revolution in computer vision was started based on deep learning and convolutional neural networks (CNN) [10]. Now CNNs provide the state-of-the-art results in most of computer vision tasks. In particular, CNN-based object detection solutions are successful in different applications starting form R-CNN [11]. Three following ways are usually considered for CNN-based face detection: hybrid "predetection+CNN" scheme, cascaded CNNs, and single shot CNN.

Predetection+CNN. The earliest way for speeding-up of CNN-based algorithms is a combination of final CNN with some essentially faster detector. For example, the R-CNN scheme [11] utilizes the algorithms of selective search for fast object predetection. The further modifications of this scheme such as Faster-R-CNN [12] evolve this idea. In particular, the EdgeBox object detector is successfully applied in [13, 14]. The original derivative of this scheme is presented in [13], where face detection task is substituted by the of task facial parts semantic segmentation.

Cascaded CNNs. Cascaded CNN-based detectors contain some set of CNNs, which sequentially test the hypothesis of face presence in image. The first CNN in cascade in usually trained for detection of faces of some fixed size, so, the image pyramid is applied in such schemes. For maximization of detection speed, the simpler CNNs in a set should be on the earlier stages of cascades. For example, in [15] authors sequentially use three CNNs with input image sizes 12×12, 24×24, and 48×48 correspondingly. Moreover, in this approach CNNs of first and second stages contain just one convolutional layer, and each CNN is learned for fixed set of face positions. Authors of [16] propose the simultaneous learning of all CNNs in cascade, which output contains both the face detection flag and the coordinates of facial bounding box. Authors of MT-CNN approach [17] utilize the paradigm of multi-functional networks described previously in [18]. MT-CNN decides both the face detection task and the facial features detection problem as well. In result of such multi-task learning MT-CNN achieves better face detection rate. Let's note that cascaded CNNs could be not so simple and fast, but in contrary, complex and sophisticated enough.

Single shot CNN. This approach presumes the processing of the whole input image by the entire CNN, which forms the output set of hypotheses about all detected faces in all possible scales simultaneously. One of the first known implementations of this CNN-based approach to object (not face) detection is YOLO architecture [1]. In this technique the output CNN layer is a grid on the input image with cells containing the tag of object or background class and coordinates of corresponding bonding box. Such CNN is learned for generating such grid of hypotheses and solving the multiscale/multiclass detection problem in one pass. This approach was further developed in fully convolutional architectures YOLO v2 [19] and so on. On the other hand, the SSD approach [3, 20] combines the ideas of YOLO and feature/scale pyramid. The SSD-based face detection is performed by the entire CNN too, but hypotheses for objects of different scale are generated via features form different CNN layers with corresponding different feature scale and spatial resolution.

Our approach. In this paper we propose the new face detection CNN called FaceDetectNet based on ideas of YOLO/DetectNet architecture [1, 2]. Our face detection approach is close to SSD, but we use the deep features of top hidden CNN layer for forming the facial proposals of any size. In result, the each 8×8 cell of our FaceDetectNet output grid aggregates the information from the input 400×400 image region. This allows us utilizing the global semantic and context information for improving the detection quality for small faces. The experimental results on the WIDER FACE benchmark confirm the advantage of such approach relative to SDD. We initially base our solution on DetectNet architecture [2], but we propose and implement two important modifications: original iterative proposal clustering (IPC) algorithm for aggregation of output face proposals formed by CNN, and the "weak" image pyramid providing better detection quality in case of highly varying input image size.

2. CNN architecture, data representation and loss function

Object detection via CNN is more complicated task than the classification one because it requires relatively much more information for training. Any label of object in a training set for object detection should contain the coordinates of the corners of object bounding box as well as the label of its class. Let's note that the number of objects can vary for different training images. In the sliding window approach to CNN-based object detection the each window position is presumed to contain only one object (or no objects) inside. In such statement the CNN is applied to each window position separately and simply trained for solving the classification problem (object or background class recognition in a window). Some cascaded face detection algorithms utilize this idea. But such approach requires learning different CNNs for objects (windows) of different size (scale). So, we need to process all image pyramid levels with different CNNs in windows. The other problem here is that we cannot use the semantic or context information from larger region than the current sliding window.

In 2016 the principally other technique to learning CNN for multiscale object detection from "you only look" (YOLO) was proposed [1]. It is primarily based on the special and original labeling information representation for object detection task. YOLO CNN represents the output object detection result (top layer) as a grid of small cells. Each cell contains the class label and bounding box of the object, which covers or partially covers this cell (see Fig. 1). The main advantage of such representation is that it has the same size and structure for any content of image with any number of objects of different shape and size. This allows learning CNN to form such representation on a training set with images of the same size, but with different content.

The original YOLO CNN model contains the full-connected layer, which means that for each cell prediction YOLO use the information from the whole image. So, even for detection of small objects, the information from large regions can be applied.

DetectNet CNN from NVIDIA [2] uses the similar output representation as YOLO. The main difference between DetectNet and YOLO architectures is that DetectNet is a fully convolution network (FCN) without any full-connected layers. Let's note that the second YOLO version YOLO v2 [19] is FCN too, but we'll refer such FCN architecture as DetectNet architecture. FCN DetectNet can process the input image of any size. In such architecture the each cell output is formed based on the analysis of some part of input image only, but this part is essentially larger than the proper cell size. For example, in original DetectNet (based on GoogleNet) the cell size is 16×16 pixels, but the corresponding source image region is more than 400×400.

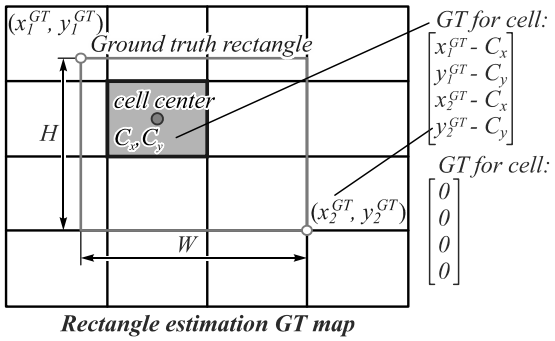
So, DetectNet architecture has the following important features, which are important for our further face detection application: it does not use the image pyramid and small sliding windows; it can use the global semantic and context information for small object detection; it can process the input images of any size. Additionally, let's note that DetectNet provides two different output layers for representation of two different types of output information: object label prediction and bounding box prediction. The each output layer of DetectNet is controlled by its proper loss function, but at the training stage they are learned simultaneously with the complex loss function, which is a linear combination of loss functions for object labels and boxes.

Let's consider the output data representation and loss function applying to our FaceDetectNet. The each output cell of FaceDetectNet contains the binary label face/background and the coordinates of facial bounding box corners – (x_1,y_1,x_2,y_2) = (left, top, right, bottom). Note that $(x_1,y_1,x_2,y_2)=(0,0,0,0)$ if there is no face covering this cell. So, the detection task here can be treated as two sub-problems: two-class semantic segmentation face/background and prediction/estimation of facial bounding box coordinates (see Fig. 1).

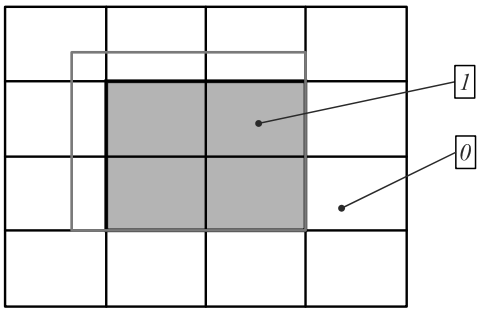
At the training of FaceDetectNet we use the complex loss function, which is a linear combination of loss functions for facial semantic segmentation and of facial bounding boxes estimation:

$$L = \alpha L_{semantic} + L_{bboxes}, \quad (1)$$

where α is tuning parameter, which determines the balance between losses of different type. In this paper we use $\alpha = 2.0$.



Rectangle estimation GT map



Semantic Segmentation GT map

Fig. 1. Data representation in FaceDetectNet

For semantic segmentation controlling we use the following $L_{semantic}$ loss function:

$$L_{semantic} = \sum_{k=1}^{N}\sum_{x=1}^{W}\sum_{y=1}^{H}(\text{label}_k(x,y) - GTmap_k(x,y))^2, \quad (2)$$

where N – number of samples; W, H – horizontal and vertical grid sizes; $label_k$ – predicted class label (face/nonface) map for k-th sample; $GTmap_k$ – ground truth class label map for k-th sample.

The facial size in image may vary in a very wide range – from 10×10 to 600×600 and even more. Due to this, the direct use of L1 or L2 loss for such output values will lead to ignoring of losses connected with smaller faces. Some authors propose to solve this problem via the so called shape priors [19], i.e. separate layers and corresponding losses for objects of different shape and size. But in the learning of our FaceDetectNet we use the L1-loss normalized by the ground truth scale of object, characterized by geometric mean of its ground truth width and heights:

$$\begin{aligned} L_{bboxes} = &\sum_{k=1}^{N}\sum_{x=1}^{W}\sum_{y=1}^{H}\left|x_1^k(\mathrm{x},\mathrm{y}) - \mathrm{x}_{gt_1}^k(\mathrm{x},\mathrm{y})\right| * K_{norm}^k + \\ &+\left|y_1^k(\mathrm{x},\mathrm{y}) - y_{gt_1}^k(\mathrm{x},\mathrm{y})\right| * K_{norm}^k + \\ &+\left|x_2^k(\mathrm{x},\mathrm{y}) - \mathrm{x}_{gt_2}^k(\mathrm{x},\mathrm{y})\right| * K_{norm}^k + \\ &+\left|y_2^k(\mathrm{x},\mathrm{y}) - y_{gt_2}^k(\mathrm{x},\mathrm{y})\right| * K_{norm}^k, \end{aligned} \quad (3)$$

where N – number of samples; W, H – horizontal and vertical grid sizes; $x_1^k, y_1^k, x_2^k, y_2^k$ – prediction of relative coordinates for k-th sample in cell $C_{x,y}$; $x_{gt_1}^k, y_{gt_1}^k, x_{gt_2}^k, y_{gt_2}^k$ – ground truth relative coordinates for k-th sample in cell $C_{x,y}$; $K_{norm}^k = 1/\sqrt{W_k * H_k}$ – scale normalization coefficient for k-th sample, W_k, H_k – ground truth width and height of k-th sample.

Such loss function allows equalizing the account of loss for small and large objects. Note that scale normalization is implemented in our FaceDetectNet architecture as a special "output-like" mask layer with $4 \times W \times H$ cells containing the K_{norm} coefficients if these cells cover faces, and 0 values – otherwise. At the training stage the eltwise multiplication of CNN output and this mask is performed, which results in required scale normalization of coordinate errors in back propagation as well as ignoring the cells without faces in the learning for bounding box prediction (see Fig. 2).

Our FaceDetectNet architecture is based on GoogleNet CNN [21] implemented in Caffe/DIGITS [22] framework and pretrained on the ILSVRC 2012 image base [23]. This basic CNN is transformed to FCN architecture via excluding of full-connected layers.

Finally, it contains 2 convolution layers, 9 inception modules, and 4 pooling layers. The output grid cell of FaceDetectNet is 8×8 pixels of size. We select the GoogleNet as a basic CNN model due to its very wide spreading in a machine learning society, which provides the guaranteed repeatability of our results with the use of any deep learning framework. One can say that nowadays this version of GoogleNet seems to be a little bit "old-school" (especially in comparison to its further modifications: Inception v3, Inception Res-net, Inception v4). Nevertheless, it is still adequate for the face detection applications due to reasonable combination of recognition quality and computational speed.

3. Clustering and weak pyramid

3.1. Clustering of face proposals

The output of FaceDetectNet CNN is a set (grid) of predicted bounding boxes and corresponding map of predicted confidence in semantic segmentation face/background. Note that the large faces may occupy many cells of grid, while the small faces may be covered by one cell only. And hypotheses about large faces will be represented many times – in each covered cell with different predicted confidences and box estimates. So, we need to transform this grid-based set of face proposals (hypotheses about face presence, size and location) into the final output representation – list of unique face proposals ordered by their confidence. In the original DetectNet this transformation called clustering (aggregation of hypotheses form cells) and performed by the OpenCVgroupRectangles [20] alghoritm. This algorithm suppresses smaller objects covered by larger objects. The process of suppression is controlled by the Intersection over Union (IoU) ratio between larger and smaller bounding boxes.

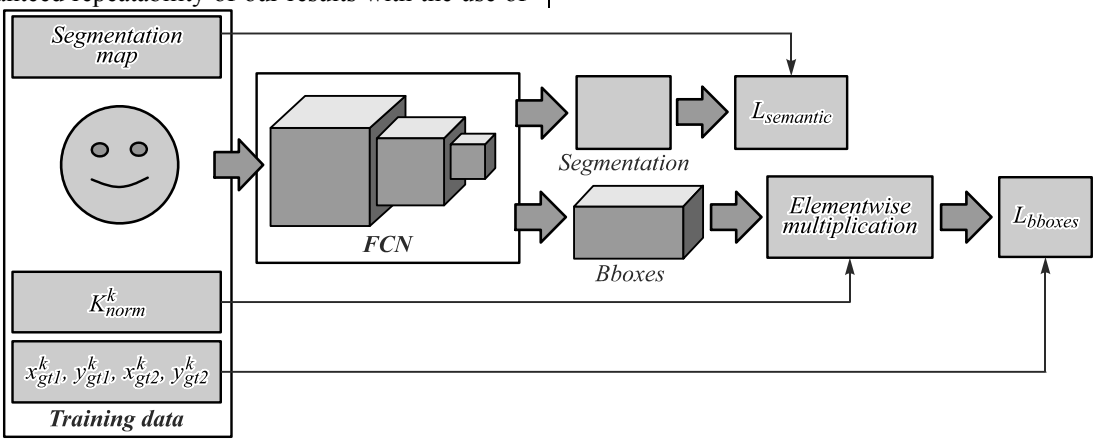


Fig. 2. Network training pipeline in FaceDetectNet

This clustering performs well in various object detection tasks, but our experiments it is not so perfect in case of intensive object overlapping, which is natural for practical face detection, for example, on the "hard" level in WIDER FACE dataset. So, we propose and implement the new iterative proposal clustering (IPC) algorithm for aggregation of output face proposals formed by FaceDetectNet CNN (see Algorithm 1).

In our current implementation we use the maximal number of IPC iterations $N_{iter} = 4$. Our experiments demonstrate that proposed IPC provides the increasing of AP by 0.03 on Wider hard dataset relative to DetectNet clustering. Some qualitative clustering results are shown on Fig. 3.

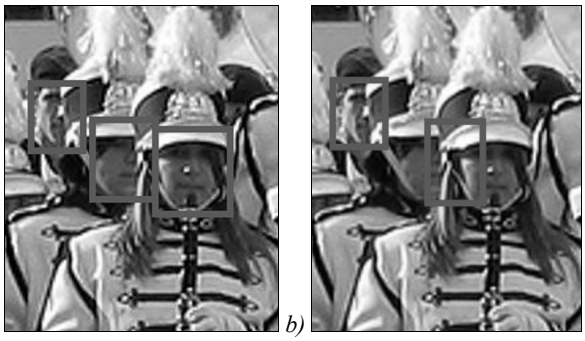
3.2. Weak Pyramid

We know that theoretically any FCN should be applicable to images of any possible size. But in practice YOLO/DetectNet like CNNs are trained on the sets of fixed-size images (in our case – 600×400 of size). Due to this some face sizes in testing images could be essentially bigger than all sample images in the training set. This problem occurs on the hard level of the WIDER FACE benchmark. So, we use the "weak image pyramid" containing the 2 scale levels only – images of original size and images proportionally scaled to be in the 600×400 frame. Applying our FaceDetectNet to test images in these 2 scales provides the reasonable face detection re-

sults for all image sizes in such bases like Hard dataset from Wider.

Algorithm 1: Iterative Proposal Clustering (IPC)

Input:
C – grid $C(x,y)$ of size (W,H) with values:
$C(x,y).Rect$ – bbox prediction in cell (x,y)
$C(x,y).Conf$ – confidence prediction in cell (x,y)
$C(x,y).Cell$ – 8×8 cell (x,y) own bbox
IoUThr – IoU ratio threshold
ConfThr – Confidence threshold
N_{iter} – maximal number of iterations
Output:
R – list of output unique proposals
Initialization:
Step 0. $N:=0$; $C\_cur = C$
Repeat:
Step 1.Initializing the accumulator grid *A*:
for each cell $x=1..W$, $y=1..H$ do
$A(x,y).Rect = (0,0,0,0)$
$A(x,y).Conf = 0$;
Step 2. Grouping the proposals:
for each $x=1..W$, $y=1..H$ do
if $C\_cur(x,y).Conf > ConfThr$ then:
for each $u=1..W$, $v=1..H$ do
if *(C_cur(u,v).Conf >ConfThr)* and ($C\_cur(u,v).Cell$ *in* $C\_cur(u,v).Rect$) then:
$A(u,v).Rect := A(u,v).Rect + C\_cur(x,y).Rect\ C(x,y).Conf$;
$A(u,v)).Conf := A(u,v).Conf + + C(x,y).Conf$;
Step 3. Averaging the proposal parameters:
for each $x=1..W$, $y=1..H$ do
if $A(x,y).Conf > 0$ then:
$A(x,y).Rect := A(x,y).Rect / A(x,y).Conf$
Step 4: Updating current state*:*
$C\_cur := A$;
$N := N + 1$;
until $N > N_{iter}$
Finalization:
Step 5. Forming the list of proposals:
from C_cur with (x,y):
$C\_cur(x\_cur,y\_cur).Conf > ConfThr$.
Step 6. Forming the list of unique proposals:
Merging proposals with $IoU > IoUThr$.
Confidence calculating as a sum of merged cell confidences.



a) *b)*

Fig 3. Clustering results example. Proposed method (a) versus DetectNet default clustering (b)

4. Experimental results

4.1. Training/Testing dataset

We use the WIDER FACE dataset [24] (training subset) only for our CNN training. It is one of the most challenging datasets for now. It contains more than 393,000 faces on 32,203 images with a high degree of face variability in scale, pose and occlusion.

This dataset supports three testing protocols: Easy, Medium and Hard. Testing protocols differ mainly in quality of included face image samples. Note that algorithm testing by Hard protocol presumes the detection of faces with size varying form 10×10 to 1024×1024, which is a really hard problem for most of modern face detection algorithms. In our testing we directly use the evaluation toolbox (source code) provided by the WIDER team.

4.2. CNN Training

For training we use 600×400 fixed size images cropped from Wider images without scaling (image size was limited due to memory limitations of our GPUs). We also use the hard examples mining (bootstrapping) technique to improve our detection rates. This technique contains following simple steps:

1. Learn CNN on the training subset;
2. Validate CNN on large dataset and add images with errors to training set;
3. Go to step one.

In this work we repeat only three iterations until AP stop growing. The improvement of AP for our model on Wider hard level was 0.047.

4.3. Evaluation results

We compare our FaceDetectNet to other face detection algorithms on the WIDER FACE dataset via three supported levels (Table 1): Easy, Medium and Hard. The results of this evaluation are shown in Table 1. Our algorithm outperforms all competitors on the Hard level besides the HR state-of-the-art solution, which is based on the essentially deeper and slower CNNs. In particular, our FaceDetectNet provides the computational speed about 30 ms/frame vs. more than 1 sec/frame for HR solution on the same NVIDIA GeForce 1080 GPU.

Table 1. AP on WIDER FACE dataset

Technique	Easy	Medium	Hard
HR	0.925	0.91	0.806
Our	**0.8**	**0.82**	**0.69**
SSD-based	0.89	0.85	0.62
CMS-RCNN	0.89	0.87	0.62
Multitask cascade CNN	0.84	0.82	0.59
LDCF	0.79	0.76	0.52
Faceness WIDER	0.71	0.63	0.34
Multiscale cascade CNN	0.69	0.66	0.42
Two-stage CNN	0.68	0.61	0.32
ACF-WIDER	0.65	0.54	0.27

Face detector based on SSD is the most close technique to our approach. Note that SSD is a general-purpose object detection architecture, which is developed for multiscale/multiclass object detection as well as YOLO/DetectNet scheme that we base on. The main idea of SSD is the usage of features form different layers of the same CNN for detection of objects of different scale. In contrary, our YOLO/DetectNet style CNN utilizes the

top convolutional layer features only, which are not so local and scale-dependent. These two approaches are directly concurrent in the area of object detection. So, their comparison in the face detection task is especially of interest. Our evaluation on the Wider hard dataset demonstrate that SSD [20] provides AP 0.625, while our approach achieves 0.69.

We conclude that this relative superiority is provided by the fact that the each cell of our FaceDetectNet output grid aggregates the information from the large input image region, which allows utilizing the global semantic and context information for improving the detection quality for small faces in the manner of HR algorithm. For example, the shape of head and shoulders or even the bigger parts of the human body could help in case of small face detection. SSD analyses just the local part of image near the face and cannot utilize such context information.

Conclusion

In this paper we propose the new solution for face detection based on ideas of YOLO/DetectNet. Our FaceDetectNet algorithm performs the face detection in "one look" in the manner of YOLO or SSD detectors.

We use the GoogleNet CNN architecture as a basic pretrained model and transform it to fully convolutional network (FCN) architecture via excluding of full-connected layers. We select this basic CNN model due to its very widespreading in a machine learning society, which provides the guaranteed repeatability of our results with the use of any deep learning framework. Obviously, the use of more powerful GoogleNet modifications (Inception v3, Inception Res-net, Inception v4) could improve the face detection quality, but simultaneously decrease the computational speed. So, we consider this choice of basic CNN model as a reasonable compromise between quality and speed.

For our FaceDetectNet implementation we propose two new algorithmic tricks developed especially for face detection task: iterative proposal clustering (IPC) algorithm for aggregation of output face proposals formed by CNN, and the 2-level "weak pyramid" providing better detection quality on the testing sets with highly varying image sizes.

Testing on the most challenging WIDER FACE hard level demonstrates that our FaceDetectNet outperforms all competitive algorithms besides the HR state-of-the-art solutions, which are based on the deeper and slower CNN models. In particular, our FaceDetectNet provides the computational speed about 30 ms/frame vs. more than 1 sec/frame for HR solution on the same NVIDIA GeForce 1080 GPU.

We provide the fine face detection quality with reasonable processing speed. Our CNN models and algorithms are available on our GitHub page. In practice one can learn or fine-tune our FaceDetectNet via DIGITS framework as a tool "from the box".

Fig. 4 shows some results on the WIDER FACE test dataset.

Fig. 4. Results on WIDERFACE dataset

Continuation of fig. 4

Continuation of fig. 4

References

[1] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 779-788. DOI:10.1109/CVPR.2016.91.

[2] Tao A, Barker J, Sarathy S. DetectNet: Deep neural network for object detection in DIGITS. Source: ⟨https://devblogs.nvidia.com/detectnet-deep-neural-network-object-detection-digits/⟩.

[3] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg A. SSD: Single shot multibox detector. Source: ⟨https://arxiv.org/abs/1512.02325⟩. DOI: 10.1007/978-3-319-46448-0_2.

[4] Hu P, Ramanan D. Finding tiny faces. Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017. Source: ⟨https://arxiv.org/abs/1612.04402⟩.

[5] Zhu C, Zheng Y, Luu K, Savvides M. CMS-RCNN: contextual multi-scale region-based CNN for unconstrained face detection. Source: ⟨https://arxiv.org/abs/1606.05413⟩ 2016.

[6] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features. Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2001; 1. DOI: 10.1109/CVPR.2001.990517.
[7] Bourdev L, Brandt J. Robust object detection via soft cascade. Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2005; 2: 236-243. DOI: 10.1109/CVPR.2005.310.
[8] Chen D, Ren S, Wei Y, Cao X, Sun J. Joint cascade face detection and alignment. In Book: Fleet D, Pajdla T, Schiele B, Tuytelaars T, eds. Computer Vision – ECCV 2014. Cham: Springer; 2014: 109-122. DOI: 10.1007/978-3-319-10599-4_8.
[9] Li J, Wang T, Zhang Y. Face detection using SURF Cascade. Proc IEEE International Conference on Computer Vision Workshops (ICCV Workshops) 2011: 2183-2190. DOI: 10.1109/ICCVW.2011.6130518.
[10] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. NIPS'12 Proc 25th International Conference on Neural Information Processing Systems 2012; 1: 1097-1105.
[11] Girshick R, Donahue J, Darrell T, Malik J. Rich feature-hierarchies for accurate object detection and semantic segmentation. CVPR '14 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014: 580-587. DOI: 10.1109/CVPR.2014.81.
[12] Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. arXiv preprint 2015. Source: ⟨https://arxiv.org/abs/1506.01497⟩.
[13] Yang S, Luo P, Loy CC, Tang X. From facial parts responses to face detection: A deep learning approach. Proc IEEE International Conference on Computer Vision 2015: 3676-3684. DOI: 10.1109/ICCV.2015.419.
[14] Jiang H, Learned-Miller E. Face detection with the faster R-CNN. 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) 2017: 650-657. DOI: 10.1109/FG.2017.82.
[15] Li H, Lin Z, Shen X, Brandt J, Hua G. A convolutional neural network cascade for face detection. Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015: 5325-5334. DOI: 10.1109/CVPR.2015.7299170.
[16] Qin H, Yan J, Li X, Hu X. Joint training of cascaded CNN for face detection. Proc 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 3456-3465. DOI: 10.1109/CVPR.2016.376.
[17] Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multi-task cascaded convolutional networks. IEEE Signal Processing Letters 2016; 23(10): 1499-1503. DOI: 10.1109/LSP.2016.2603342.
[18] Zhang C, Zhang Z. Improving multiview face detection with multi-task deep convolutional neural networks. Proc IEEE Winter Conference on Applications of Computer Vision 2014: 1036-1041. DOI: 10.1109/WACV.2014.6835990.
[19] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. arXiv preprint. Source: ⟨https://arxiv.org/abs/1612.08242⟩.
[20] Yang S, Xiong Y, Change C, Tang LX. Face detection through scale-friendly deep convolutional networks. arXiv preprint. Source: ⟨https://arxiv.org/abs/1706.02863⟩.
[21] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. Proc 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015: 1-9. DOI: 10.1109/CVPR.2015.7298594.
[22] Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. Proc 22nd ACM international conference on Multimedia 2014: 675-678. DOI: 10.1145/2647868.2654889.
[23] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. Proc IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2009: 248-255. DOI: 10.1109/CVPR.2009.5206848.
Yang S, Luo P, Loy CC, Tang X. WIDER FACE: A face detection benchmark. Proc 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016: 5525-5533. DOI: 10.1109/CVPR.2016.596

Authors' information

Vladimir Sergeevich Gorbatsevich (b.1985) graduated from Moscow Aviation Institute (National Research University) in 2009. Currently he works as the head of laboratory at the FGUP "GosNIIAS". Author of 20 scientific papers. Research interests are processing and image analysis, digital photogrammetry, computer vision, mathematical morphology, pattern recognition, machine learning, biometry. E-mail: *gvs@gosniias.ru* .

Anastasia Sergeevna Moiseenko (b.1993) graduated from Moscow Institute of Physics and Technology in 2017. Currently she works as the engineer at the FGUP "GosNIIAS". Research interests are processing and image analysis, computer vision, machine learning, biometry. E-mail: *moiseenko.as@phystech.edu* .

Yury Valentinovich Vizilter (b.1970) graduated from Moscow Aviation Institute (National Research University) in 1992. Since 1997 is the Candidates of Technical Sciences, 2009 - the Doctor of Technical Sciences. Currently he works as the head of department at the FGUP "GosNIIAS". Author of 80 scientific papers. Research interests are processing and image analysis, digital photogrammetry, computer vision, mathematical morphology, pattern recognition, machine learning, biometry. E-mail: *viz@gosniias.ru* .

Code of State Categories Scientific and Technical Information (in Russian – GRNTI)):28.23.15
Received December 7, 2017. The final version – October 30, 2018.