FaceBoxes: A CPU Real-time Face Detector with High Accuracy

Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang and Stan Z. Li

CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China

① Motivation

Accuracy Efficiency

One of the remaining open challenges of face detection is to ***achieve CPU real-time speed as well as maintain high performance***, since effective models for face detection tend to be computationally prohibitive. To address this challenge, we propose a novel face detector, named FaceBoxes, with superior performance on both speed and accuracy.

Method

- **A lightweight yet powerful network architecture**
  - Rapidly Digested Convolutional Layers (RDCL) is designed to enable FaceBoxes to achieve CPU real-time speed.
  - Multiple Scale Convolutional Layers (MSCL) aims at enriching the receptive fields and discretizing anchors over different layers to handle faces of various scales.
- **A novel anchor densification strategy**
  - Making different types of anchors have the same density to significantly improves the recall rate of small faces.

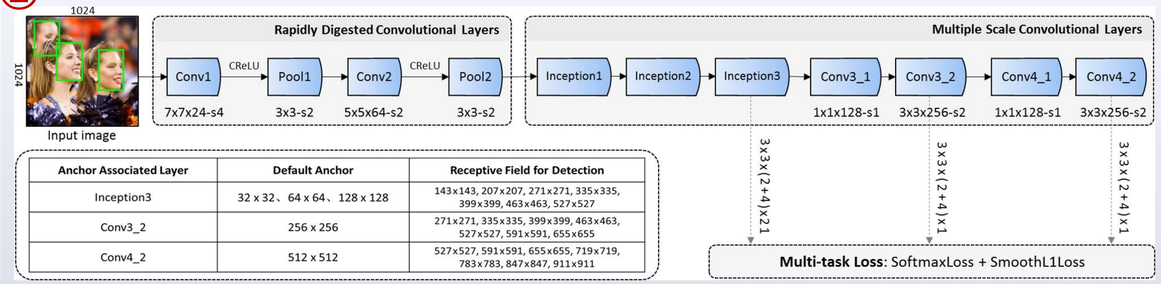
Result

- **Efficiency:** For VGA-resolution images, 20 FPS on a single CPU core and 125 FPS using a GPU. Moreover, **the speed of FaceBoxes is invariant to the number of faces**.
- **Accuracy:** Presenting state-of-the-art detection performance on several face detection benchmarks, including the AFW, PASCAL face, and FDDB.

Contribution

- We design the Rapidly Digested Convolutional Layers (RDCL) to enable face detection to achieve CPU real-time speed;
- We introduce the Multiple Scale Convolutional Layers (MSCL) to handle various scales of face via enriching receptive fields and discretizing anchors over layers.
- We present a novel anchor densification strategy to improve the recall rate of small faces;
- We further improve the state-of-the-art performance on the AFW, PASCAL face, and FDDB datasets.

② Architecture



Anchor Associated Layer	Default Anchor	Receptive Field for Detection
Inception3	32 x 32、64 x 64、128 x 128	143x143, 207x207, 271x271, 335x335, 399x399, 463x463, 527x527
Conv3_2	256 x 256	271x271, 335x335, 399x399, 463x463, 527x527, 591x591, 655x655
Conv4_2	512 x 512	527x527, 591x591, 655x655, 719x719, 783x783, 847x847, 911x911

Rapidly Digested Convolutional Layers

- Shrinking the spatial size of input with a series of large stride sizes
  The total stride size of RDCL is 32, which means the input spatial size is reduced by 32 times quickly.
- Choosing suitable kernel size
  The suitable kernel size can not only speed up, but also alleviate the information loss brought by the spatial size reducing.
- Reducing the number of output channels by utilizing C.ReLU
  It significantly increases speed with negligible decline in accuracy.

Multiple Scale Convolutional Layers

- Multi-scale design along the dimension of network width
  Consisting of multiple convolution branches with different kernels, Inception module can enrich the receptive fields to learn visual patterns for different scales of faces.
- Multi-scale design along the dimension of network depth
  Associated with multi-scale feature maps, the default anchors are discretized over multiple layers with different resolutions to naturally handle faces of various sizes.

③ Anchor densification strategy

Problem

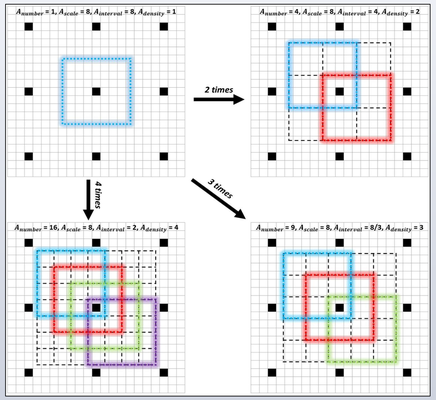
- We define that the tiling density of anchor $A_{density}$ is:

$$A_{density} = A_{scale} / A_{interval}$$

- Where A_{scale} is the scale of anchor and $A_{interval}$ is the tiling interval of anchor
- Because some different scales of default anchors are associated with the same layer, there is a tiling density imbalance problem between anchors of different scales

Solution

- To eliminate this imbalance, we propose a novel anchor densification strategy
- To densify one type of anchors **n** times, we uniformly tile $A_{number} = n^2$ anchors around the center of one receptive field instead of only tiling one at the center of this receptive field to predict
- This strategy can be used to guarantee that different scales of anchor have the same density, so that various scales of faces can match almost the same number of anchors



④ Efficiency VS Accuracy

Approach	CPU-model	mAP(%)	FPS
ACF	i7-3770@3.40	85.2	20
CasCNN	E5-2620@2.00	85.7	14
FaceCraft	N/A	90.8	10
STN	i7-4770K@3.50	91.5	10
MTCNN	N/A@2.60	94.4	16
Ours	E5-2660v3@2.60	**96.0**	**20**

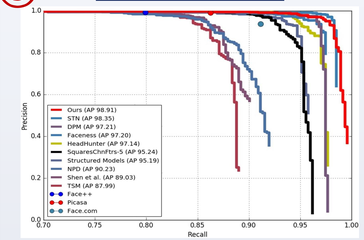
- **CPU inference time and mAP compared between different methods**
  - **FPS** is for VGA images on CPU
  - **mAP** means the true positive rate at 1000 false positives on FDDB.
  - For STN, its mAP is the true positive rate at 179 false positives and with ROI convolution, its FPS can be accelerated to 30 with 0.6% recall rate drop
- **The proposed FaceBoxes: efficiency & accuracy**
  - **20** FPS on the CPU devices
  - **125** FPS on a single GPU
  - **3.87** MB in model size
  - **state-of-the-art** performance

⑤ Model analysis

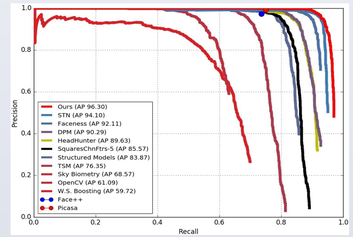
Contribution	FaceBoxes			
RDCL				×
MSCL			×	×
Strategy		×	×	×
Accuracy (mAP)	96.0	94.9	93.9	94.0
Speed (ms)	50.98	48.27	48.23	67.48

- **Anchor densification strategy is crucial:** increasing the density of small anchors to improve the recall rate of small faces
- **MSCL is better:** enriching the receptive fields and discretizing anchors over different layers to handle faces of various scale
- **RDCL is efficient and accuracy-preserving:** enabling FaceBoxes to achieve CPU real-time speed as well as keep the accuracy unchanged

⑥ AFW dataset result



PASCAL face dataset result



FDDB dataset result