Detecting Face with Densely Connected Face Proposal Network

Shifeng Zhang

2017.10.28

Center for Biometric Security Research

National Laboratory of Pattern Recognition

Institute of Automation, Chinese Academy of Sciences

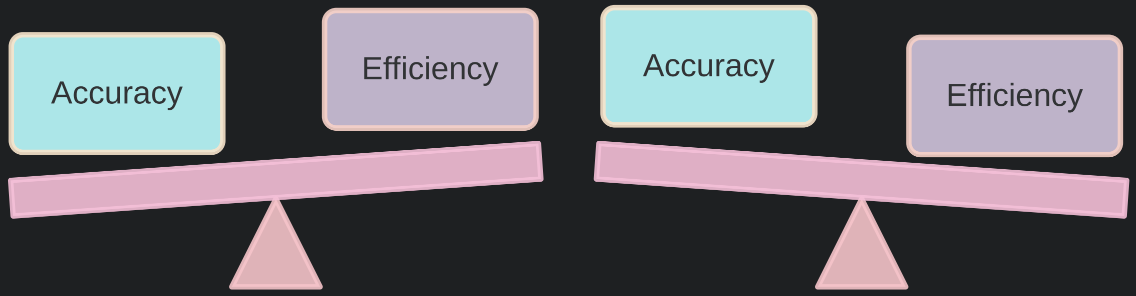
Content

- Introduction
- Proposed method
- Experiments
- Summary

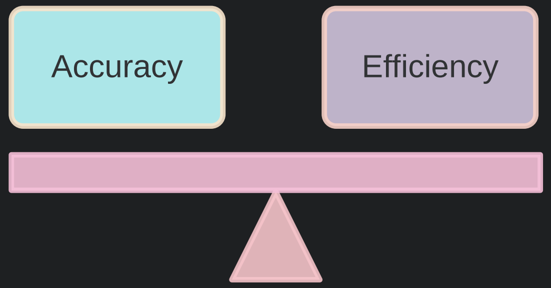
Content

Motivation

- ***Dilemma***: Accuracy and efficiency are two conflicting challenges for face detection
  - Effective models[1] tend to be time-consuming
  - Efficient models tend to be not robust enough



- ***Solution***: shrink the input image and focus on detecting small faces
  - Trade-off between accuracy and efficiency



[1] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z. S3FD: Single Shot Scale-invariant Face Detector. ICCV (2017)

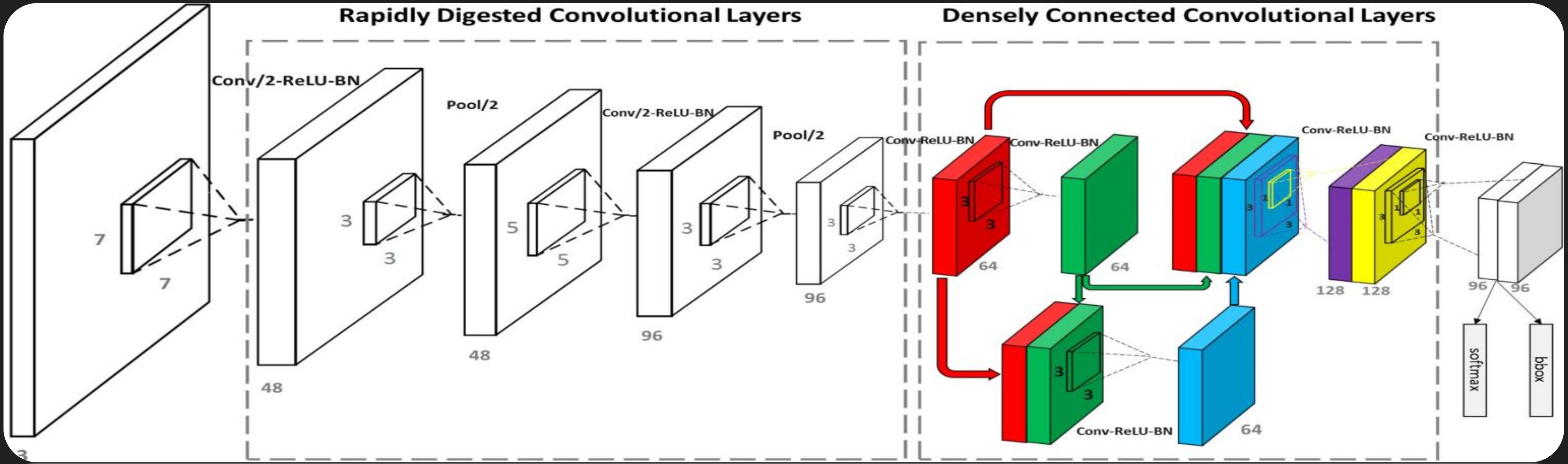
Contributions

- Developing a novel face detector (DCFPN) with high performance as well as CPU real-time speed
- Designing a lightweight-but-powerful network with the consideration of efficiency and accuracy
- Proposing a fair L1 loss and using dense anchor strategy to handle small faces well
- Achieving state-of-the-art performance on common benchmark datasets at the speed of 30 FPS on CPU and 250 FPS on GPU for VGA images

Content

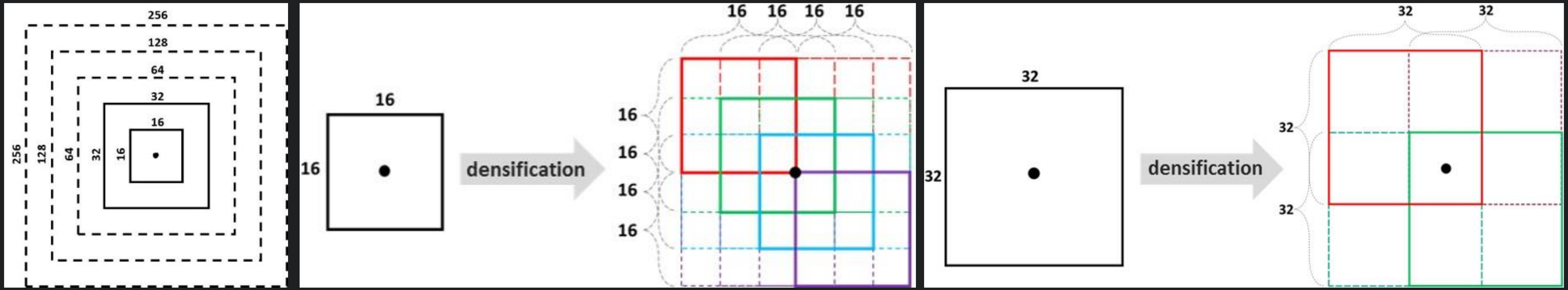
- Introduction
- **Proposed method**
- Experiments
- Summary

Architecture



- ***Rapidly Digested Convolutional Layers (RDCL)*:** quickly reducing spatial size by 16 times with narrow but large convolution kernels to achieve CPU real-time speed
- ***Densely Connected Convolutional Layers (DCCL)*:** enriching the receptive field to learn visual patterns for different scales of faces, combining coarse-to-fine information to improve the recall rate and precision of detection.

Dense anchor strategy



- ***Problem***: the last conv layer has 5 default anchors whose tiling interval are 16 pixels. Comparing with large anchors (64, 128, 256), small anchors (16, 32) are too sparse, which results in low recall rate of small faces.
- ***Solution***: dense anchor strategy is proposed by [1] to solve this tiling density imbalance problem. As illustrated in above figure, it uniformly tiles several anchors around the center of one receptive field instead of only tiling one.

[2] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z. Faceboxes: A CPU Real-time Face Detect or with High Accuracy. IJCB (2017)

Fair L1 loss

- The regression target of Fair L1 loss is as follows:

$$\boldsymbol{t_x = x - x^a,\ t_y = y - y^a,\ t_w = w,\ t_h = h;}$$
$$\boldsymbol{t_x^* = x^* - x^a,\ t_y^* = y^* - y^a,\ t_w^* = w^*, t_h^* = h^*}$$

- where x, y, w, h denote center coordinates and width and height, x, xa, x* are for predicted box, anchor box, and GT box (likewise for y, w, h). The scale normalization is implemented to have scale-invariance loss value as follows:

$$\boldsymbol{L_{reg}(t, t^*) = \sum_{j\in\{x,y,w,h\}} fair_{L_1}(t_j - t_j^*),\quad where\ \ fair_{L_1}(z_j) = \begin{cases} |z_j|/w^*, & if\ j\ \in\ \{x, w\} \\ |z_j|/h^*, & otherwiese \end{cases}}$$

- It equally treats small and big face by directly regressing box's relative center coordinate and width and height.

Content

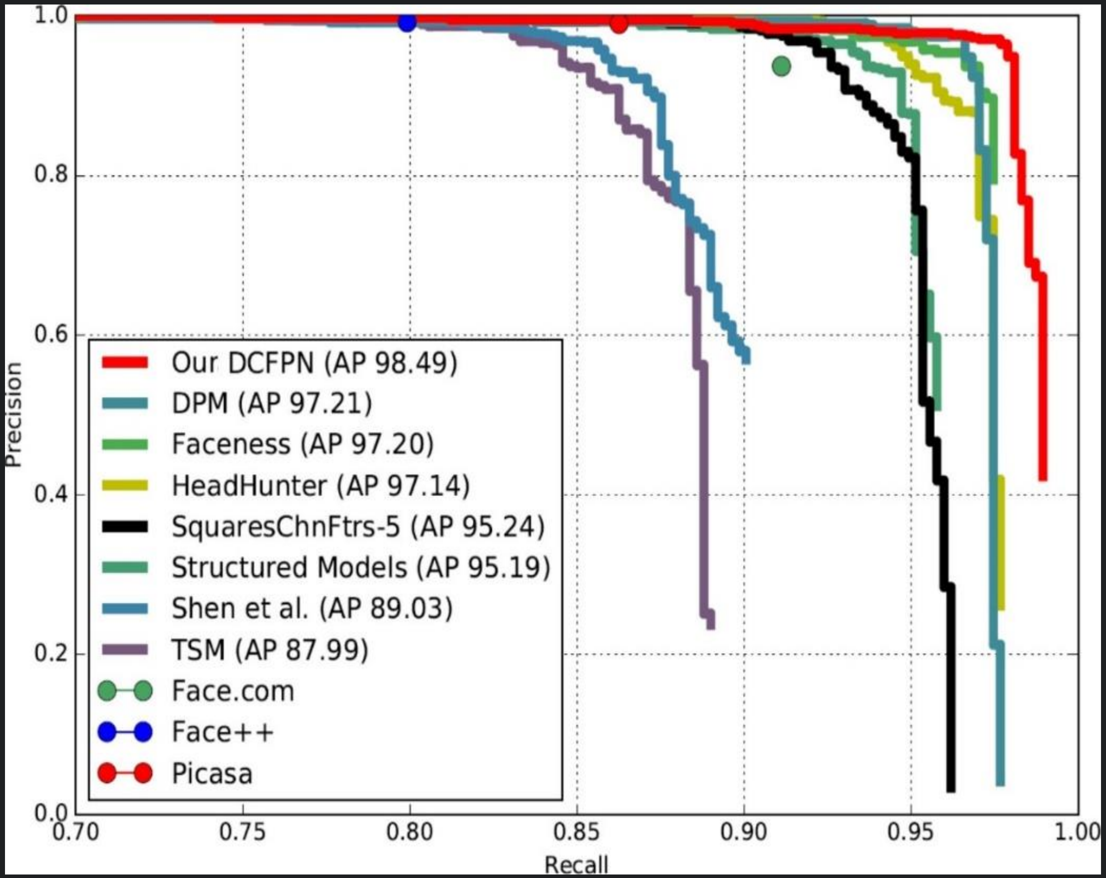
Model analysis

Component	DCFPN			
Designed architecture?	✓	✓	✓	
Dense anchor strategy [2]?	✓	✓		
Fair L1 loss?	✓			
Accuracy (mAP)	95.2	94.5	93.7	93.2

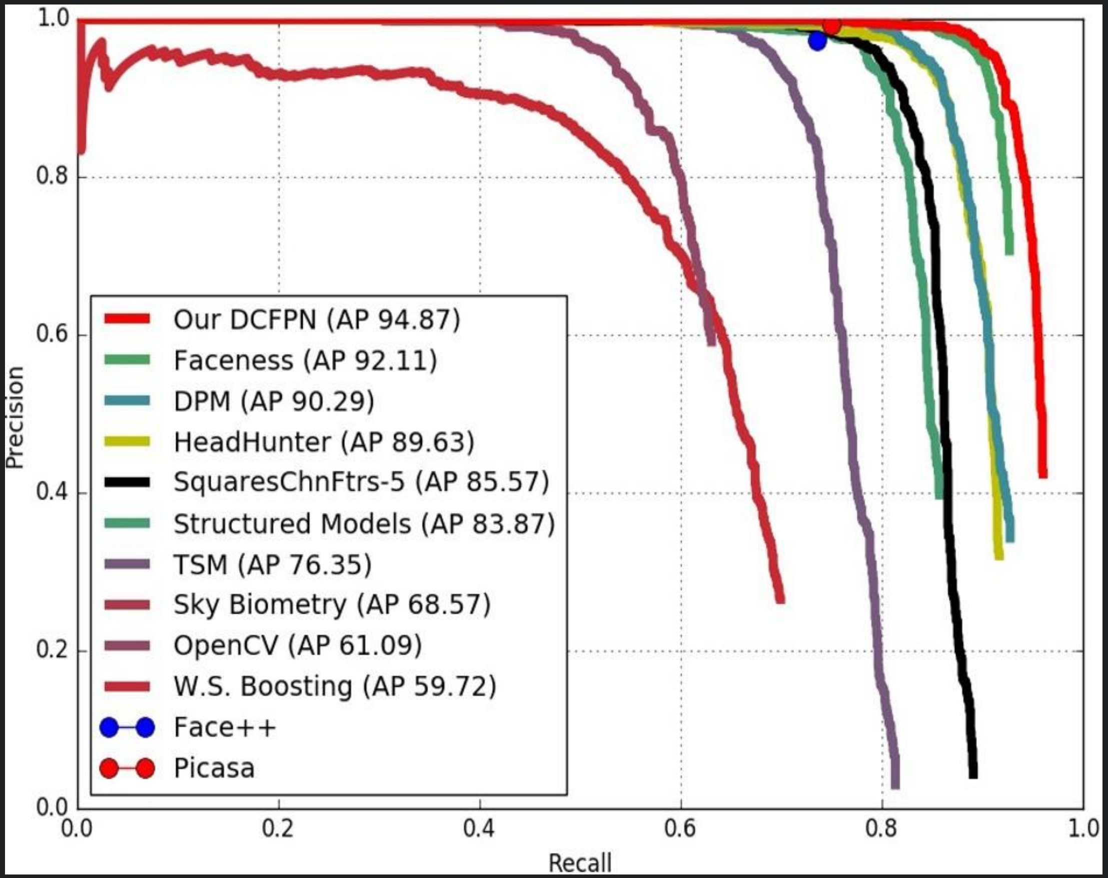
- ***Fair L1 loss is promising***: **+**0.7% owns to locating small faces well
- ***Dense anchor strategy is effective***: **+**0.8% shows the importance of this strategy
- ***Designed architecture is crucial***: **+**0.5% demonstrates the effectiveness of enriching the receptive fields and combining coarse-to-fine information across different layers

[2] Zhang, S., Zhu, X., Lei, Z., Shi, H., Wang, X., Li, S.Z. Faceboxes: A CPU Real-time Face Detect or with High Accuracy. IJCB (2017)

Evaluation on benchmark

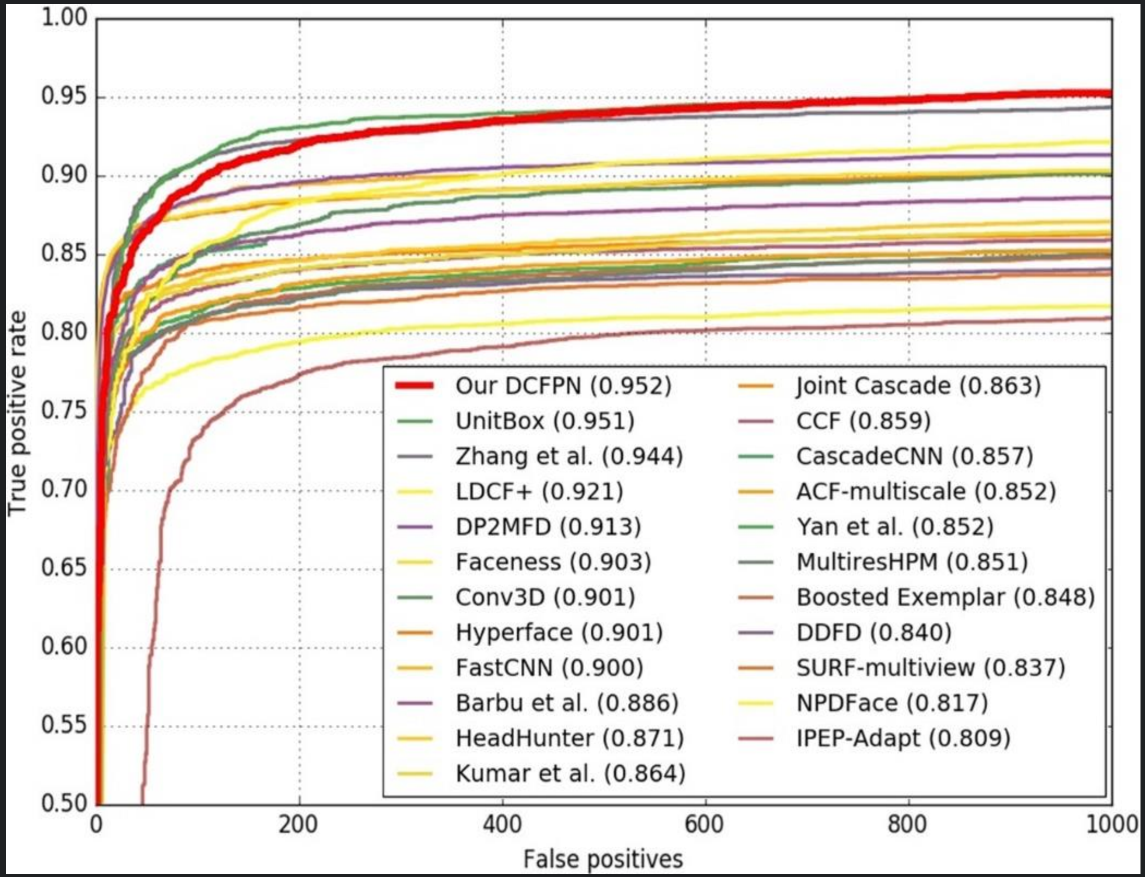


AFW dataset

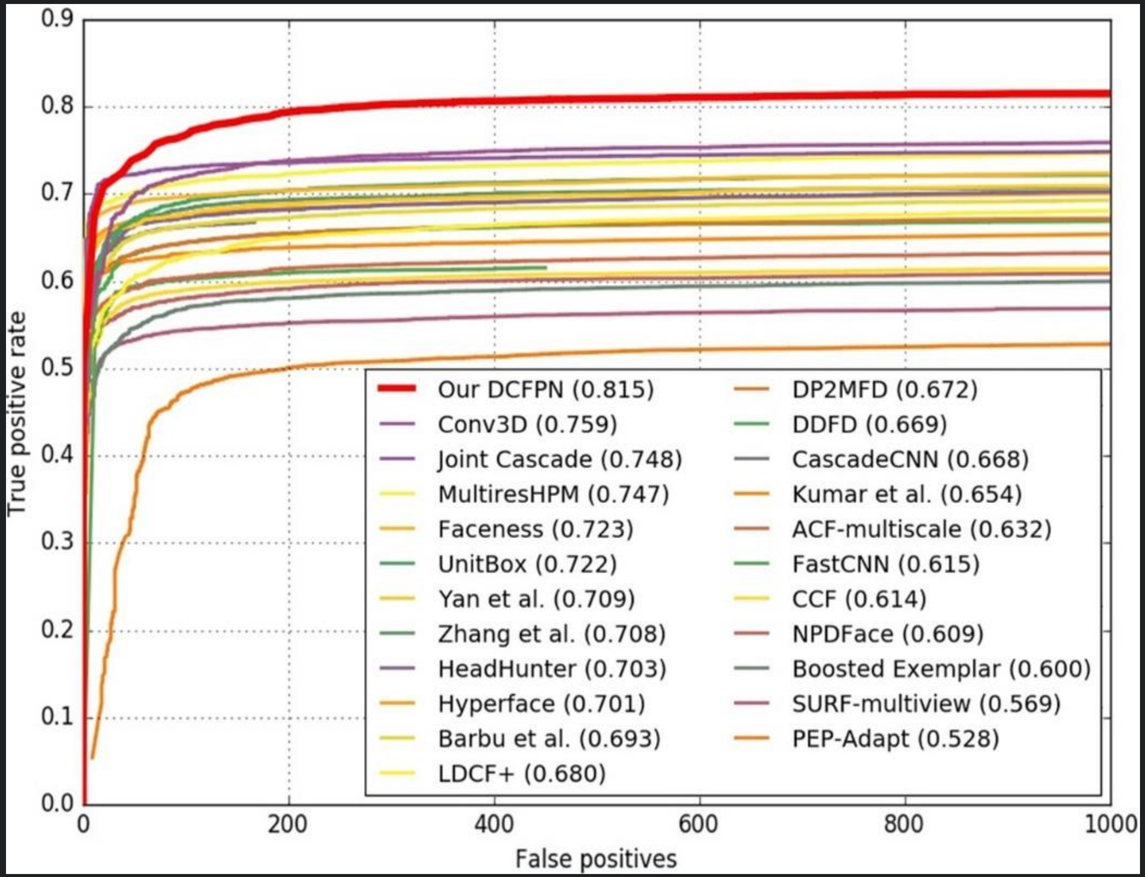


PASCAL face dataset

Evaluation on benchmark



Discontinuous score on FDDB dataset



Continuous score on FDDB dataset

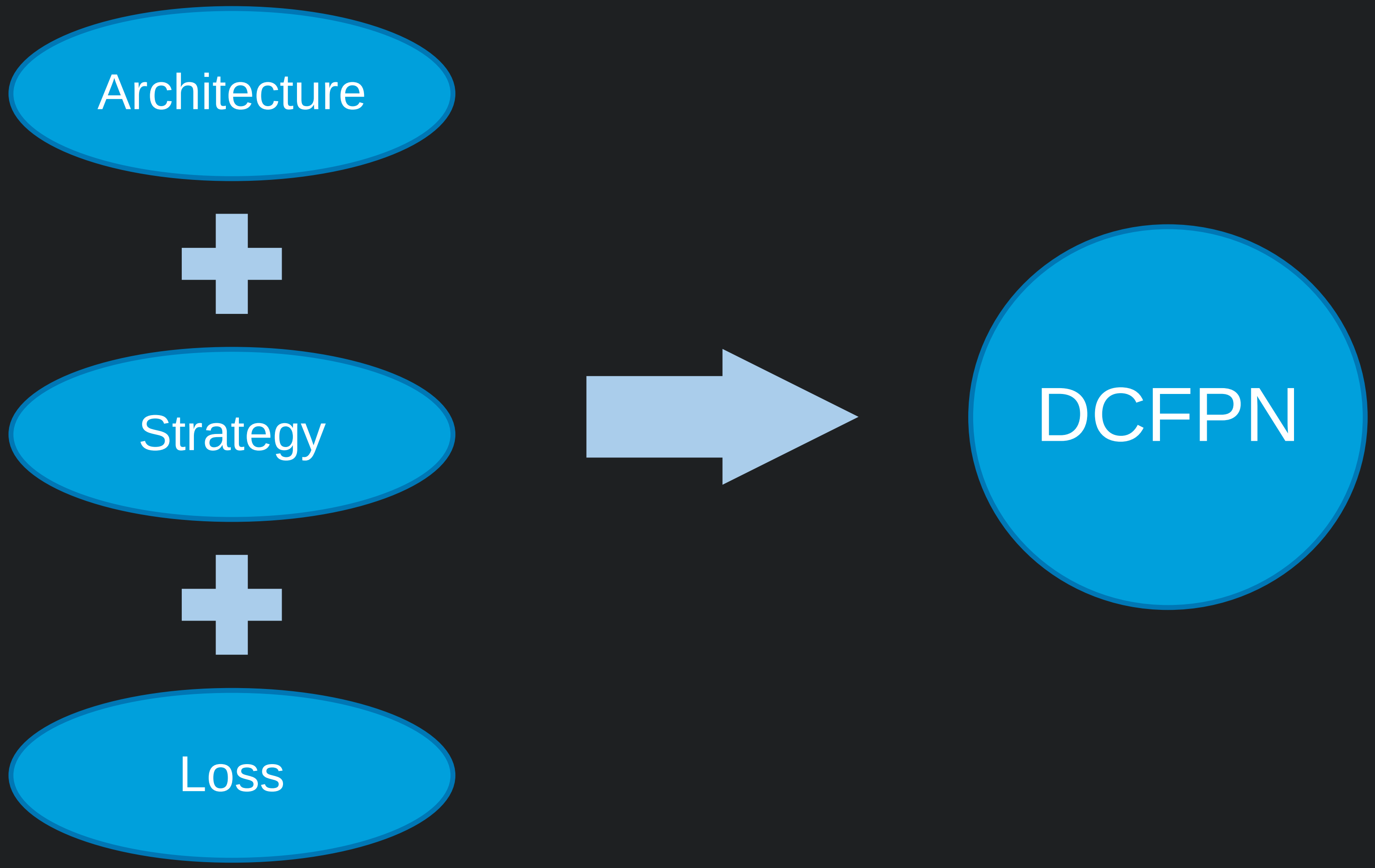
Runtime efficiency

For VGA-resolution images to detect faces ≥ 40 pixels:

CPU	30 FPS on a 2.60GHz CPU
GPU	250 FPS using a GPU card
Size	3.2 MB in model size

Content

- Introduction
- Proposed method
- Experiments
- **Summary**


Architecture
Strategy
Loss
DCFPN


Thanks