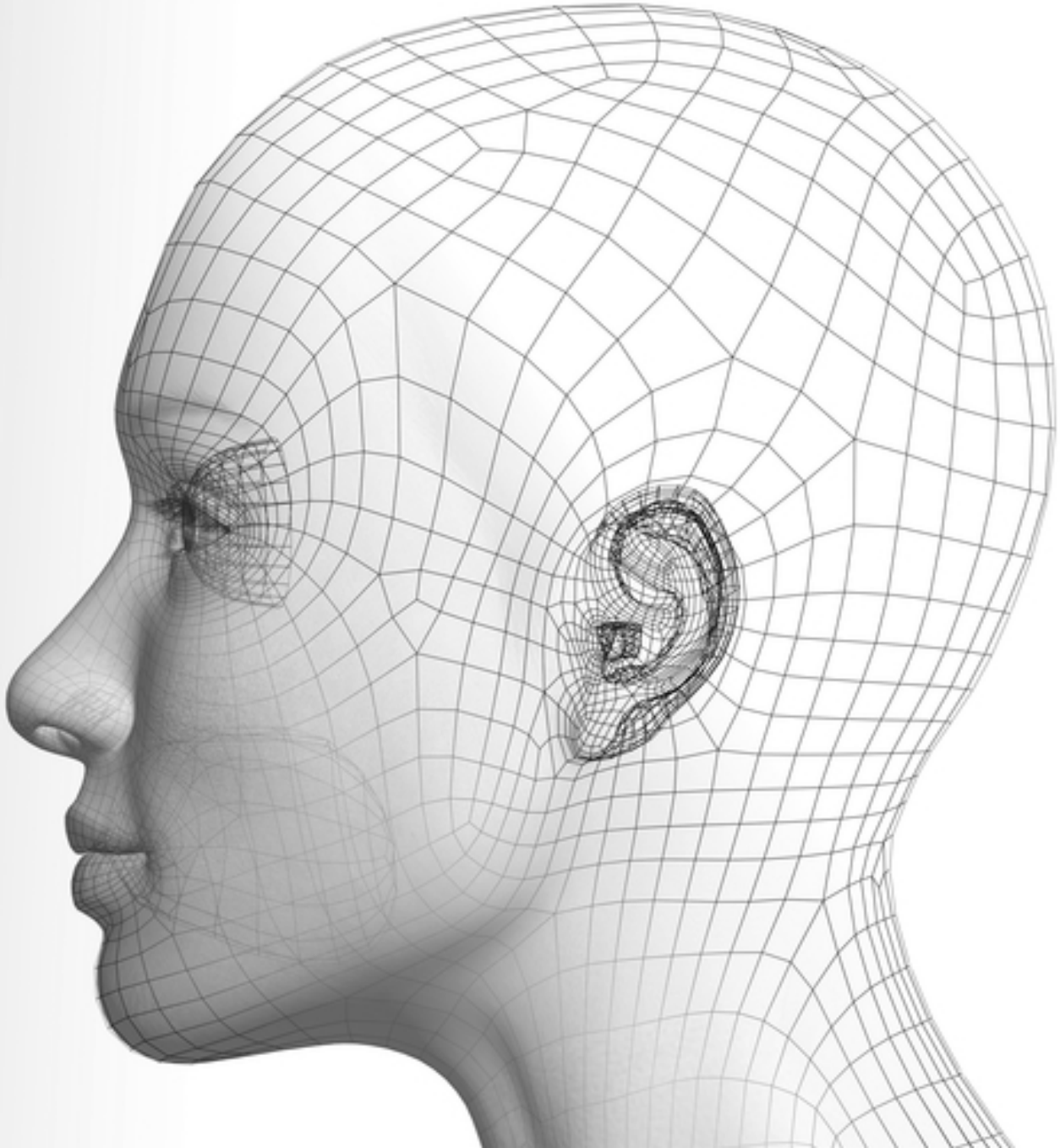
Deep Learning for Face Analysis

Chen-Change LOY

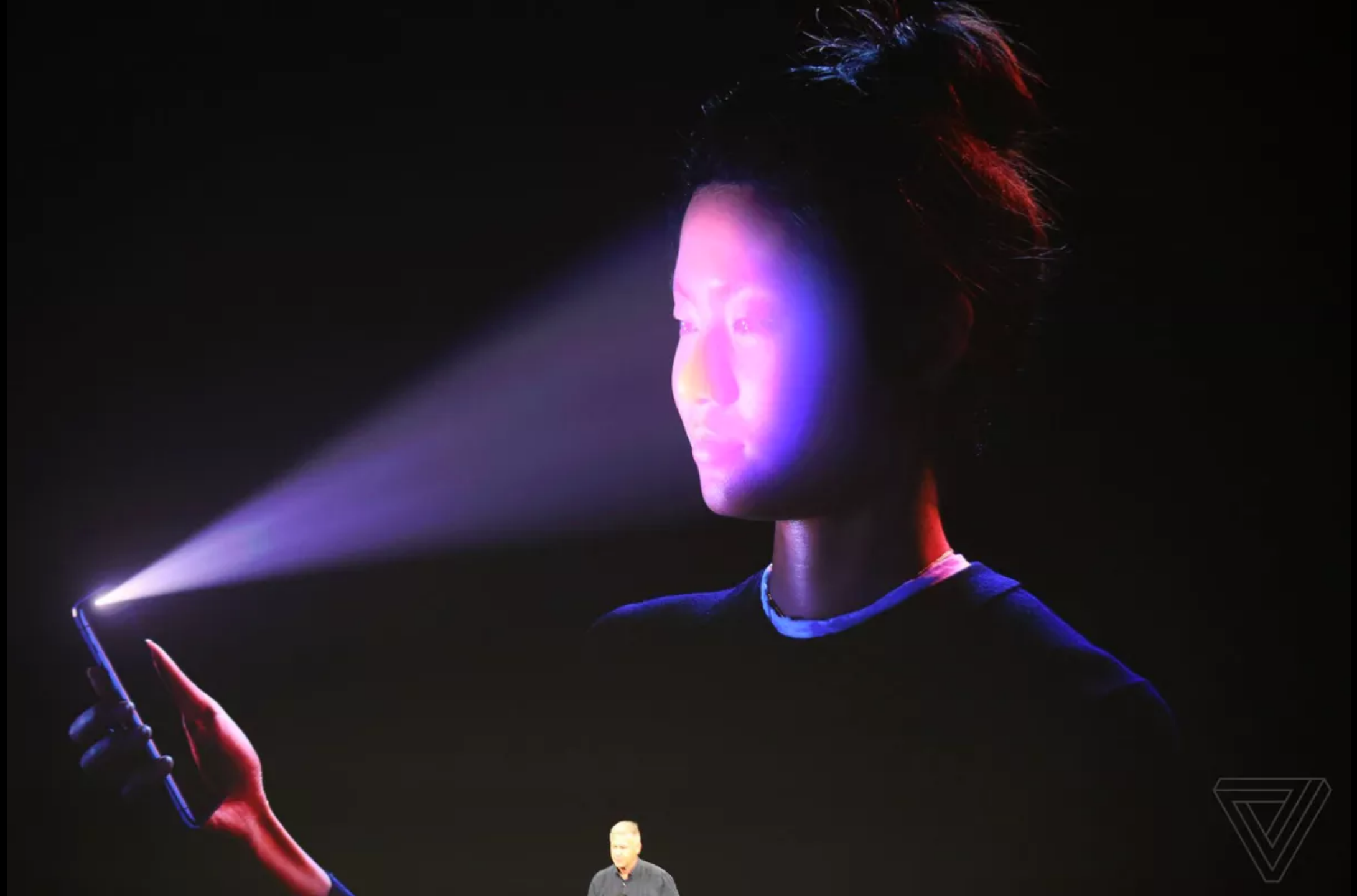
MMLAB

The Chinese University of Hong Kong

Homepage: http://personal.ie.cuhk.edu.hk/~ccloy/



https://www.youtube.com/watch?v=k3T2WbRkgvg&index=4&list=PLkNuzPSJx0mO0_mLUjDQFXFgngTV7QwHZ



Vivo X20 Face Wake: unlock your mobile phone in 0.1 seconds


请您摘掉眼镜、帽子看这里3秒
Look here 3 seconds , automatic paper out
让科技·改变行为
文明游园
节约用纸
北京知道

Papers

DeepID3 99.55%

DeepID2 99.15%

GaussianFace 98.52%

C. Lu, X. Tang, "Surpassing Human-Level Face Verification Performance on LFW with GaussianFace", *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI),*, January 2015. **Best student paper of AAAI 2015**

Human accuracy 97.45%

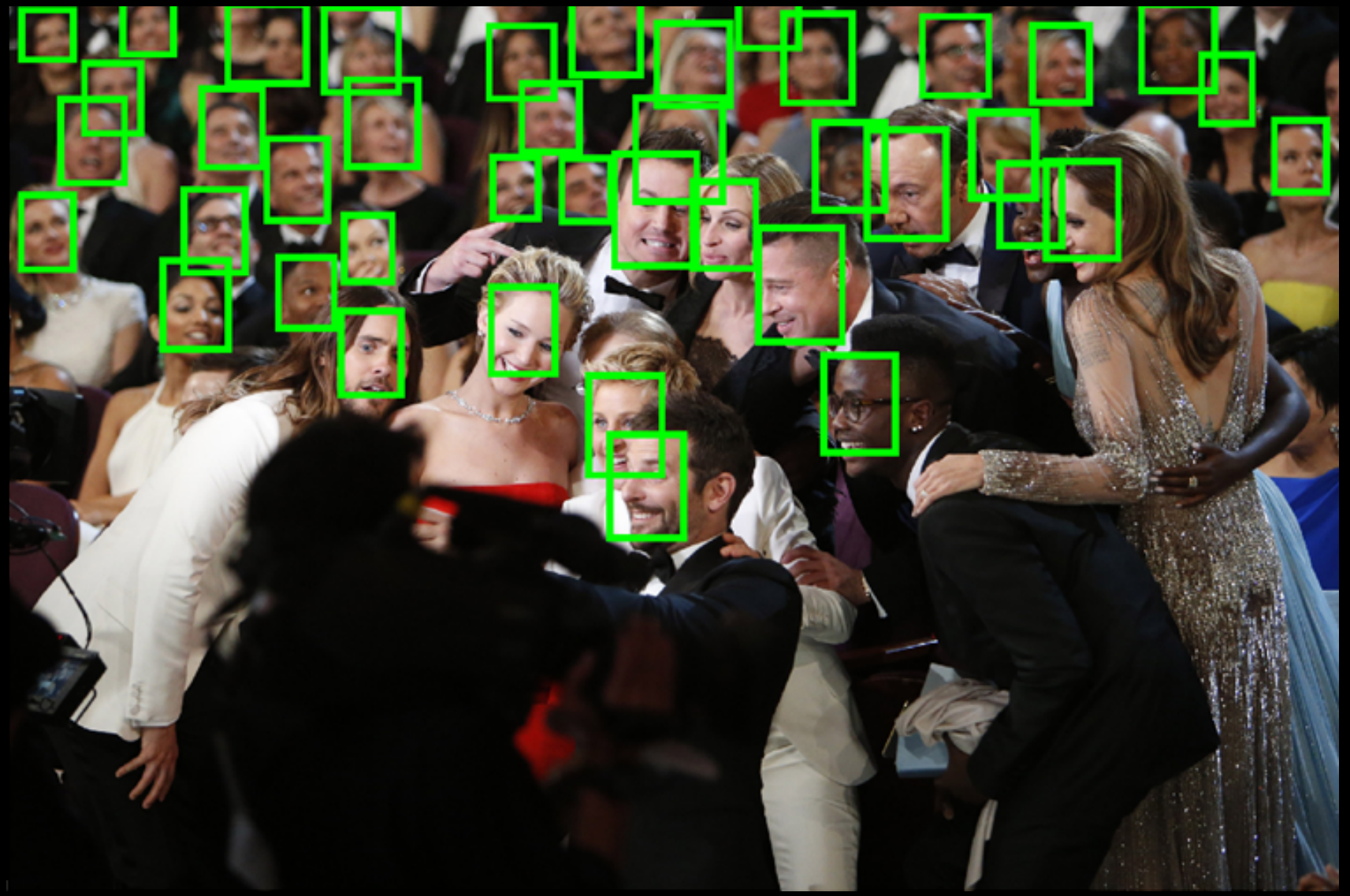
Industry Breakthrough

Training set
DeepID2: 200K images
Now: 2 billion images in total, 200M individuals' faces

1:1 result
DeepID2 (2014): 99.5% accuracy @ 0.5% FAR
6 digit password (2015): >90% accuracy @10^-6 FAR
8 digit password (2017): >97% accuracy @10^-8 FAR

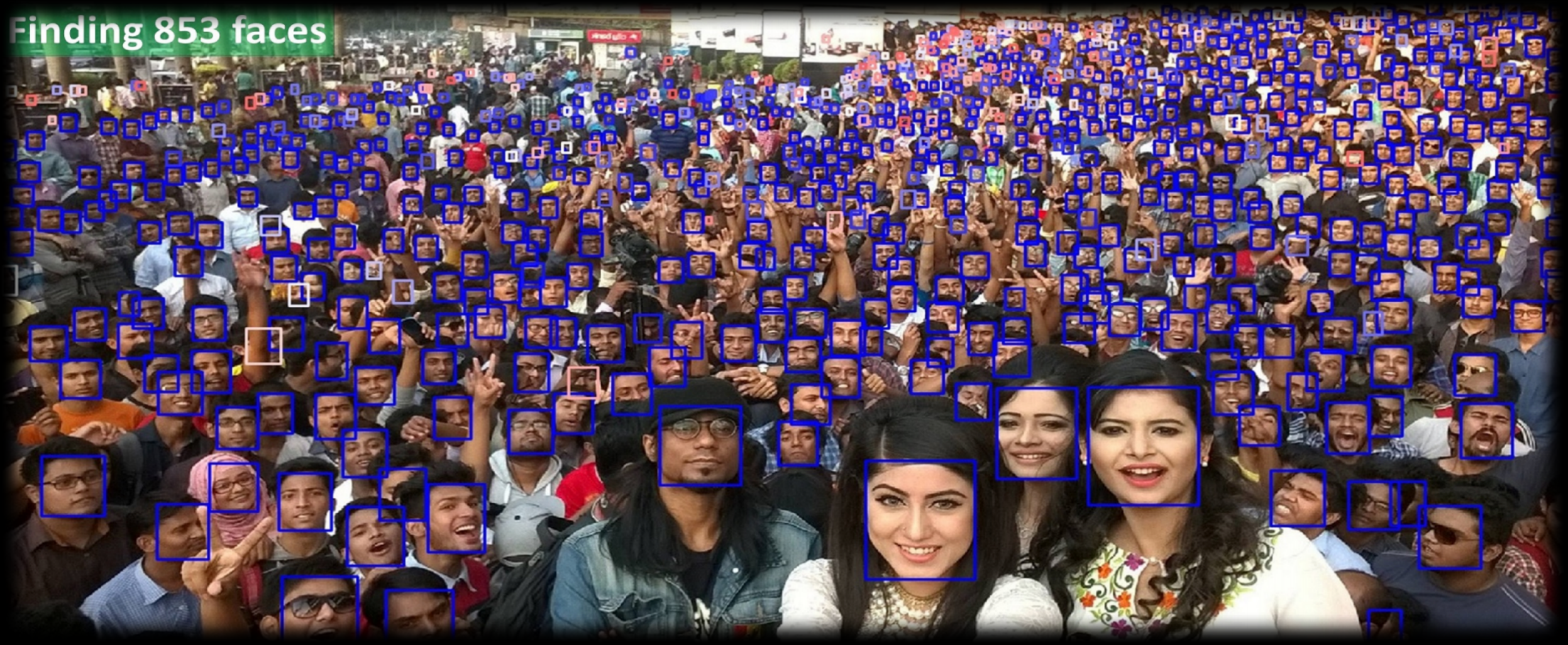
1:N result
DeepID2: top 30 < 40% for N = 100M
Now: top 30 > 90% for N = 100M

2015



Yang et al., From Facial Part Responses to Face Detection: A Deep Learning Approach, ICCV 2015

2017



Zhang et al., S 3FD: Single Shot Scale-invariant Face Detector, ICCV 2017

Is there anything else
I can solve?

Is there anything else I can solve?

- Learning in small data regime
- The use of unannotated data
- Challenging scenarios
- Generalization and transferability
- Imbalance problem
- …

Face Recognition

Pose-Robust Face Recognition via Deep Residual Equivariant Mapping

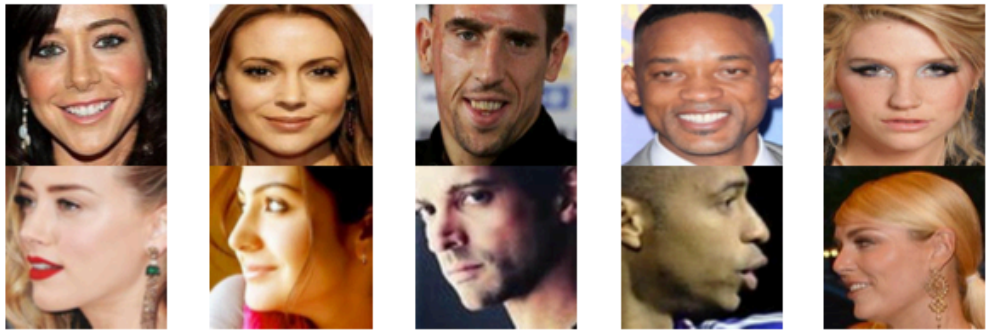
K. Cao, Y. Rong, C. Li, C. C. Loy

A submission to CVPR 2018

Profile and Frontal Face Recognition

- Large pose discrepancy between two face images is one of the key challenges in face recognition
- The number of frontal and profile training faces are highly imbalanced

False Positives



False Negatives

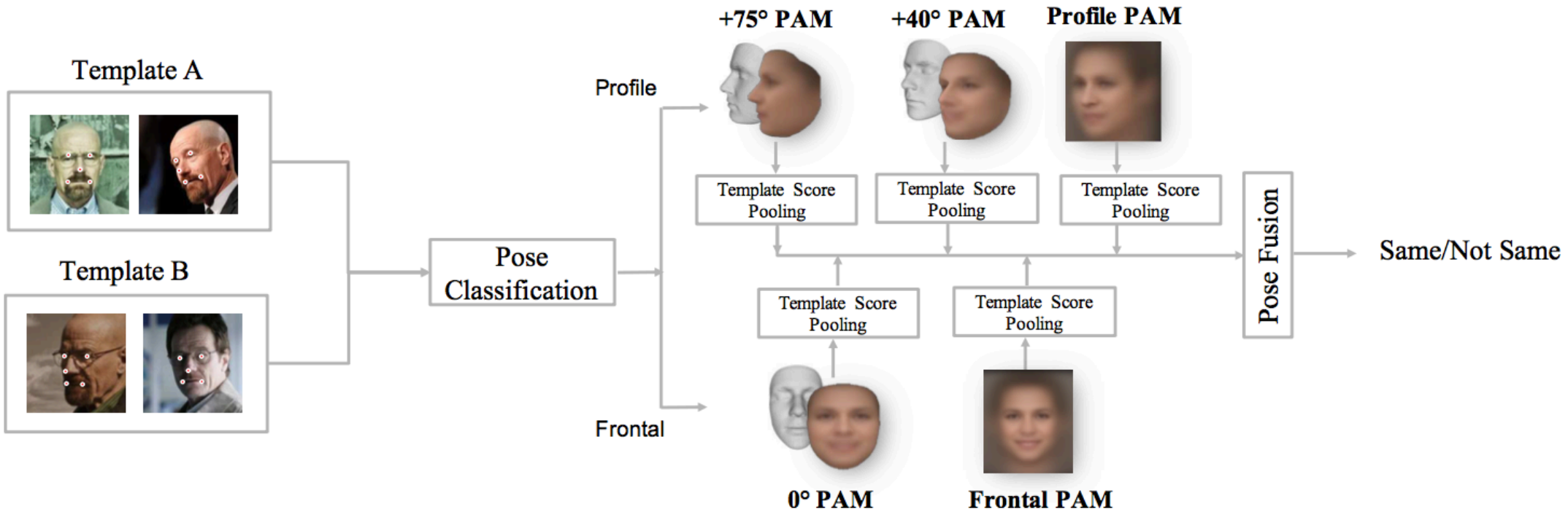
Profile faces of different persons are easily to be mismatched (false positives), and profile and frontal faces of the same identity may not trigger a match leading to false negatives

Why does not face recognition work well on profile faces?

- The generalization power of deep models is usually proportional to the training data size

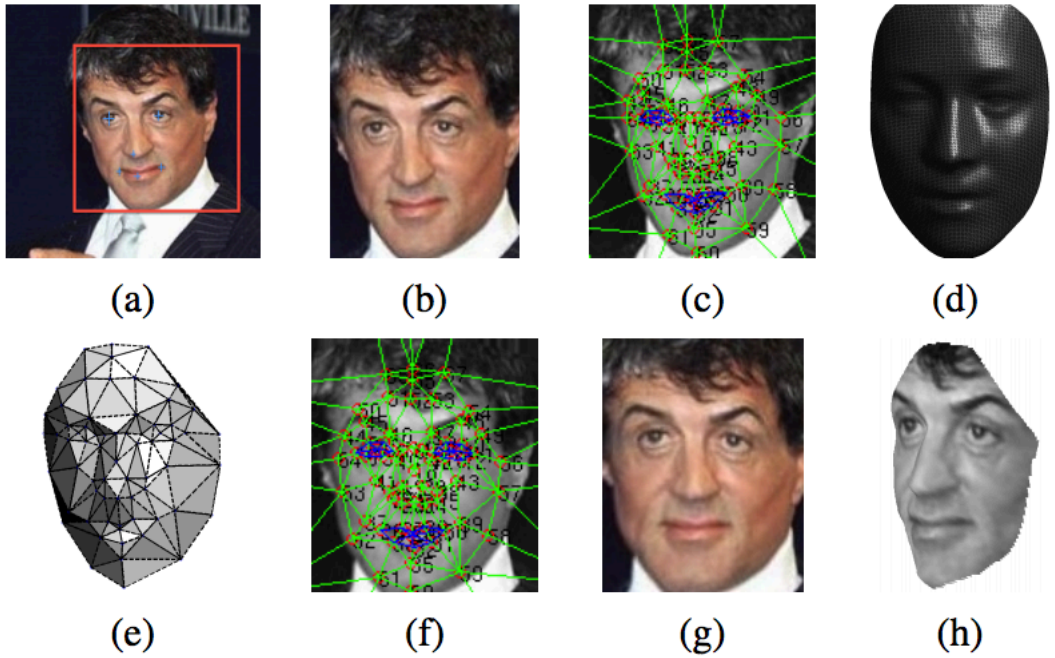
- Given an uneven distribution of profile and frontal faces in the dataset, deeply learned features tend to bias on distinguishing frontal faces rather than profile faces.

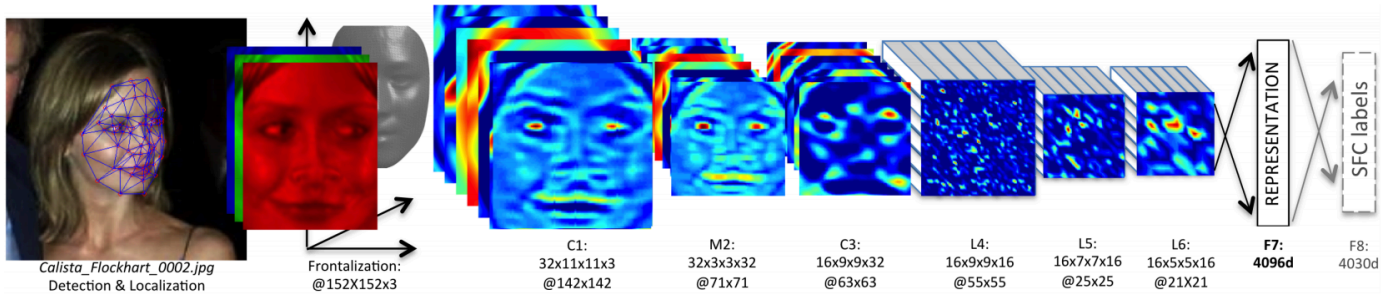
Existing solutions



I. Masi, S. Rawls, G. Medioni, and P. Natarajan. Pose-aware face recognition in the wild. In CVPR, 2016

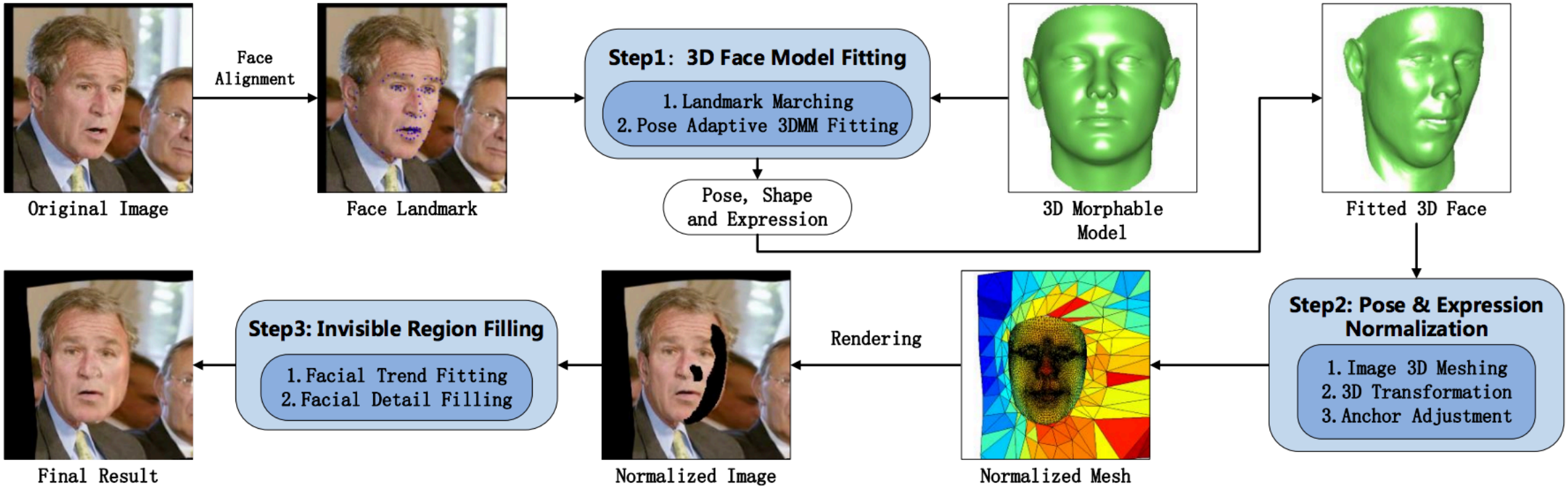
Existing solutions





Y. Taigman et al. Deepface: Closing the gap to human-level performance in face verification. In CVPR, 2014

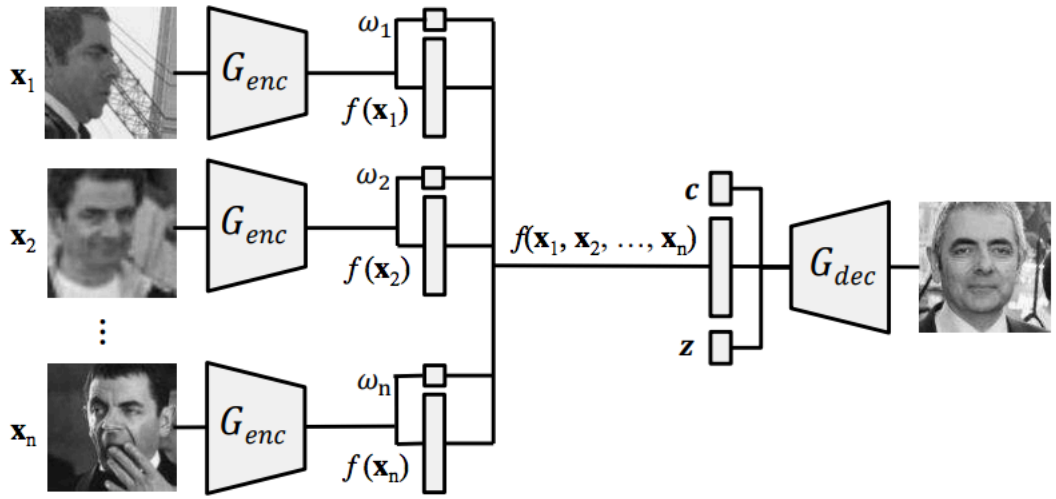
Existing solutions



Zhu et al. High-Fidelity Pose and Expression Normalization for Face Recognition in the Wild, CVPR 2015

Existing solutions

Model



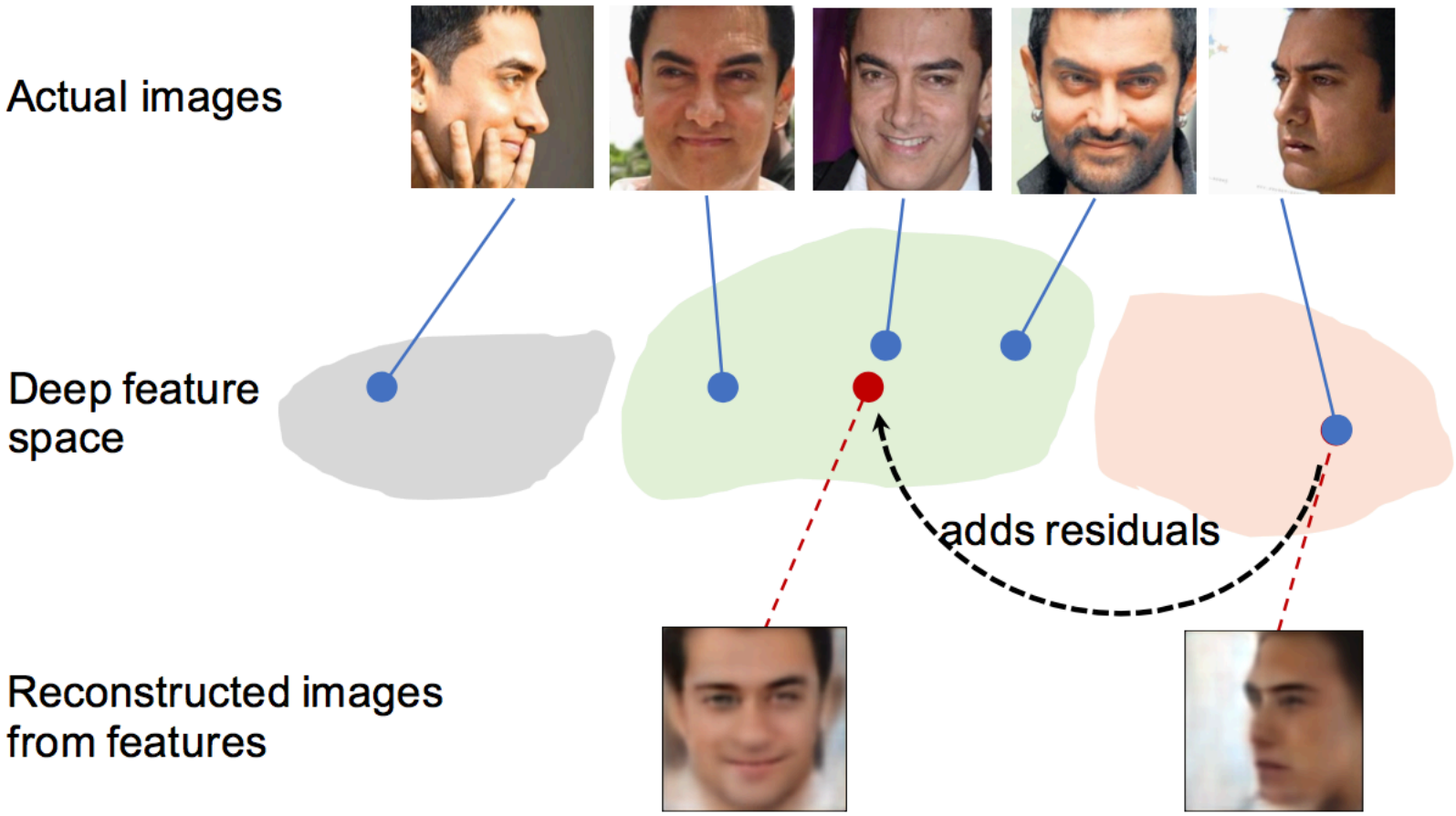
Input

Generated

Real

L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In CVPR, 2017

Motivation



We can map profile face feature to the frontal space through a mapping function that adds residual.

Feature equivariance

- The representation of many deep layers depends upon transformations of the input image
- Such transformations can be learned by a mapping function from data
- The function can be subsequently applied to manipulate the representation of an input image to achieve the desired transformation

K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In CVPR, 2015

Feature equivariance

- A convolutional neural network (CNN) can be regarded as a function φ that maps an image $x \in X$ to a vector $\varphi(x) \in R^d$

- The representation φ is said equivariant with a transformation g of the input image if the transformation can be transferred to the representation output

$$\forall x \in X: \ \varphi(gx) \approx M_g \varphi(x)$$

K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In CVPR, 2015

Problem formulation

- For simplicity, let's assume we have: frontal face image $\boldsymbol{x}_f$ and profile face image $\boldsymbol{x}_p$
-
- We wish to obtain a transformed representation of a profile image $\boldsymbol{x}_p$ through a mapping function M_g , so that $M_g\ \varphi(\boldsymbol{x}_p) \approx \varphi(\boldsymbol{x}_f)$

$$M_g\ \varphi(\boldsymbol{x}_p) = \varphi(\boldsymbol{x}_p) + \mathcal{Y}(\boldsymbol{x}_p)\mathcal{R}(\boldsymbol{x}_p) \approx \varphi(\boldsymbol{x}_f)$$

residual function

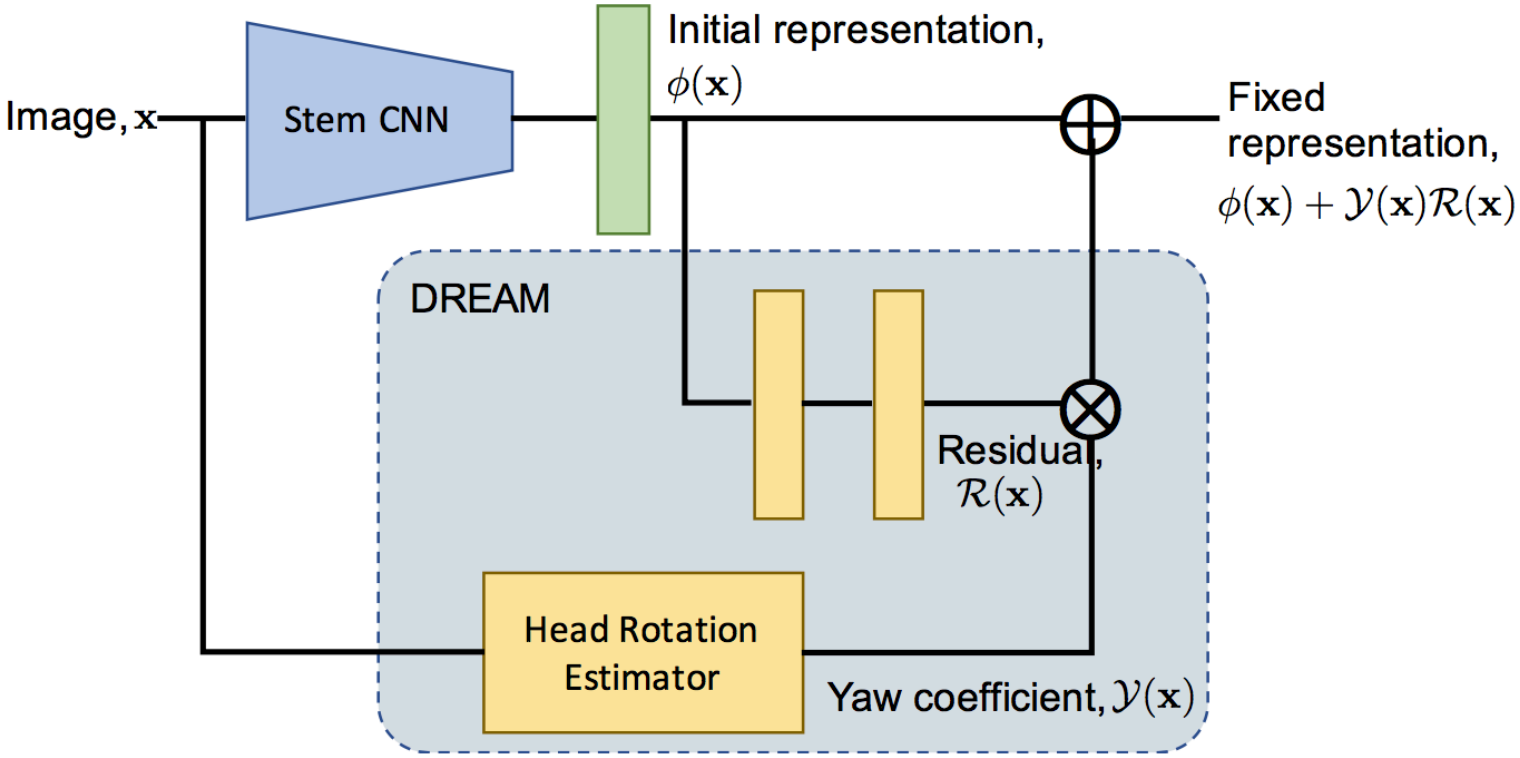
yaw coefficient, [0 1], a *soft gate* of the residuals

Problem formulation

- Yaw coefficient
  - provide a higher magnitude of residuals (thus a heavier fix) to a face that deviates more from the frontal pose
  - $\mathcal{Y}(\boldsymbol{x}) = 0$ for frontal face and gradually changes from 0 to 1 when the face pose shifts from frontal to a complete profile

- The soft gate can be viewed as a correction mechanism that adopts top-down information (the yaw in our case) to influence the feed-forward process

Network structure – the DREAM block

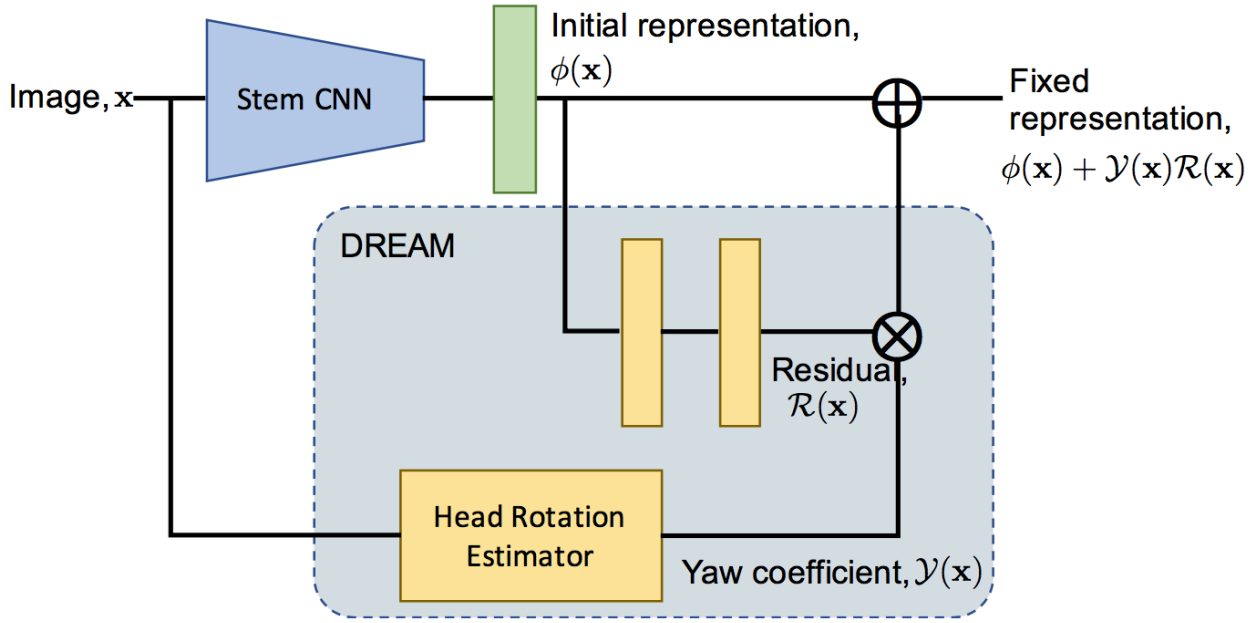


The *Deep Residual EquivAriant Mapping* (DREAM) block

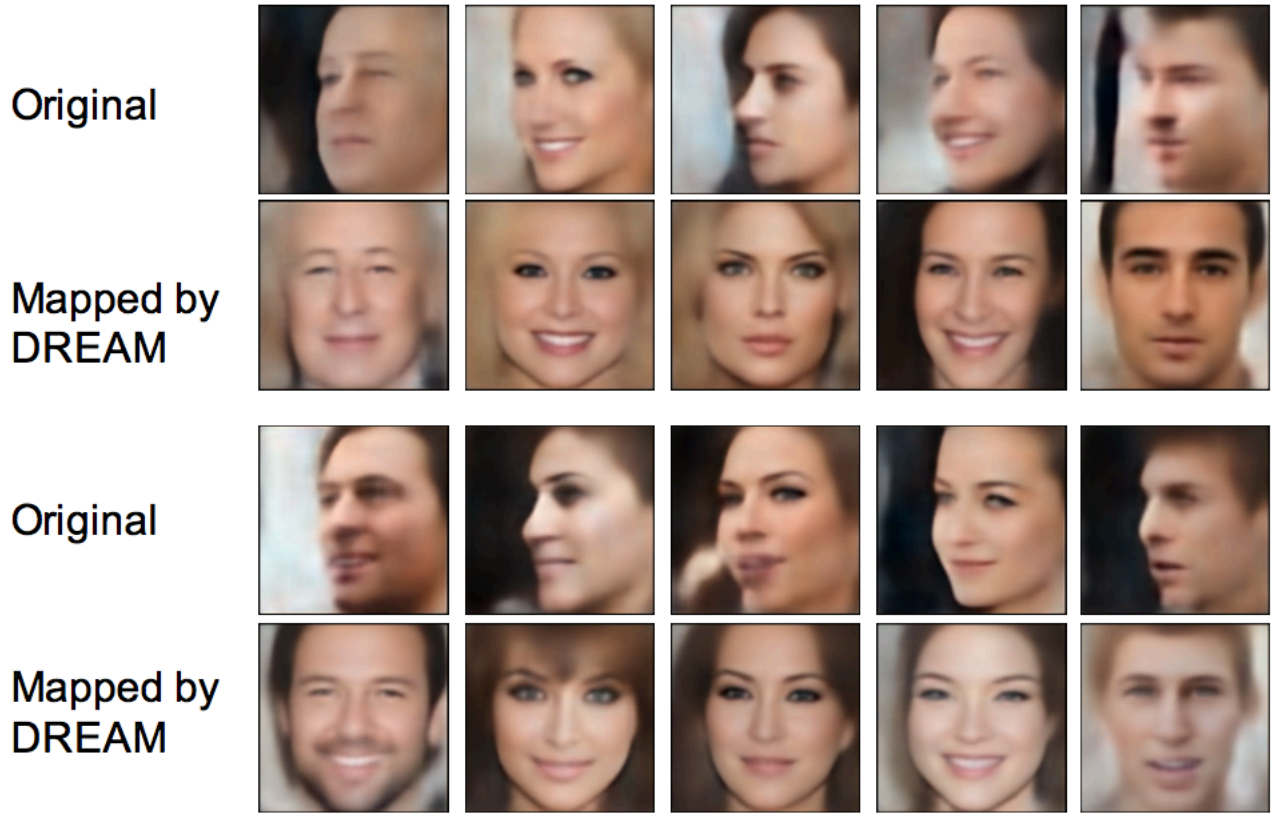
Usage of DREAM

- Stitching
  - Stitch the DREAM block to an existing stem CNN
- End-to-end + Stitching
  - First end-to-end training
  - Followed by DREAM block fine-tuning
- DREAM block training

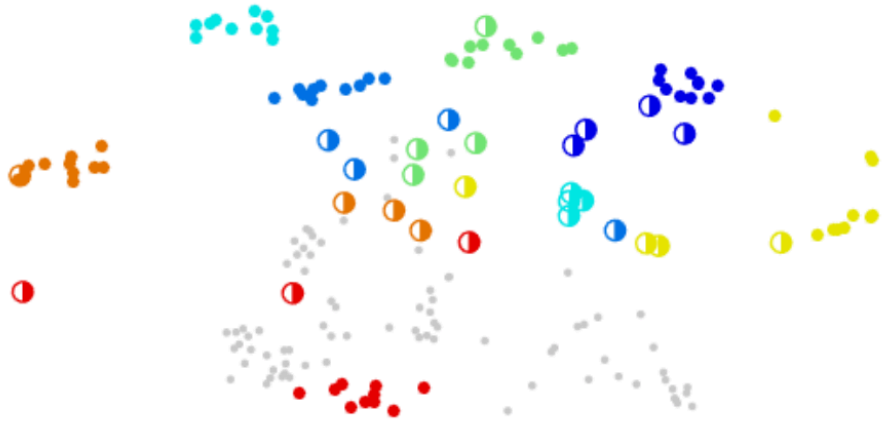
$$\min_{\Theta_R} \mathrm{E}\|\phi(\mathbf{x}) + \mathcal{Y}(\mathbf{x})\mathcal{R}(\phi(\mathbf{x}); \Theta_{\mathcal{R}}) - \phi(\mathbf{x}_f)\|_2^2$$



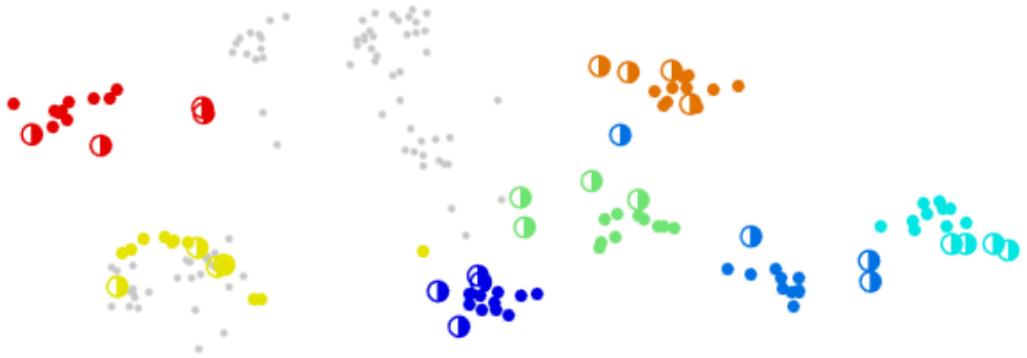
Visualization



Visualization



a) Feature Space of ResNet-18



b) Feature Space of ResNet-18 + DREAM block

Results on Celebrities in Frontal-Profile (CFP)

- Equal error rate (EER).
- Baselines
  - CDFE - Two transforms are simultaneously learned to map the samples in two modalities respectively to the common feature space.
  - JB – Joint Bayesian approach for face verification
  - FF - Face Frontalization morphs faces from profile to frontal with a generative adversarial network

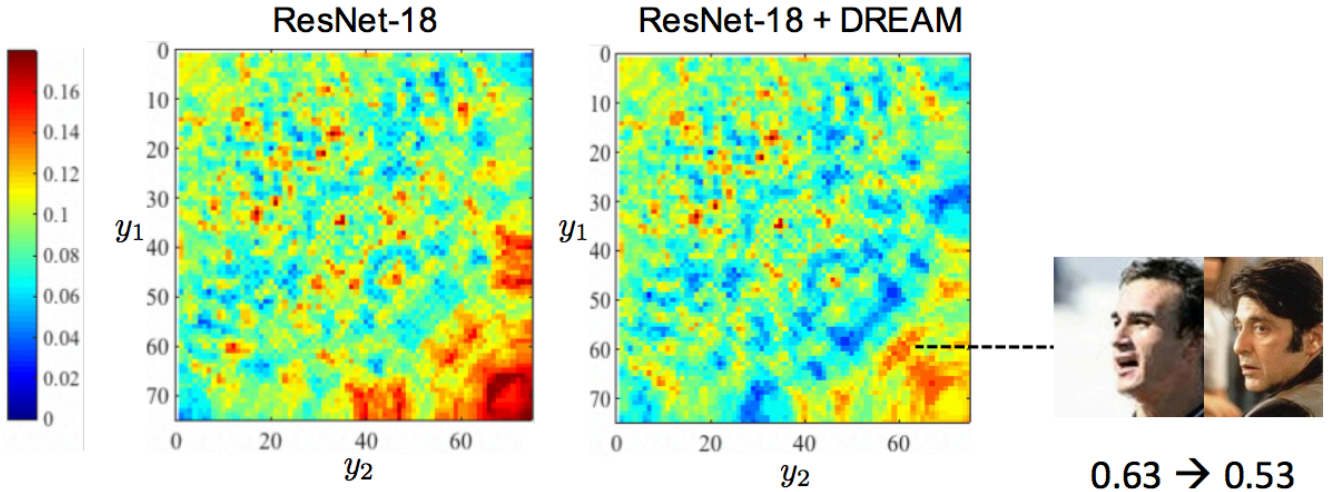
Model	Training Data	Naïve	Other Strategies			Variants	
			CDFE [17]	JB [3]	FF [31]	stitching	end2end+retrain
ResNet-18	MS-Celeb-1M	8.40	8.30	8.37	14.40	7.71	**7.03**
ResNet-50	MS-Celeb-1M	7.89	7.71	8.49	14.26	7.29	**6.02**
Center-Loss	MS-Celeb-1M	8.54	8.49	8.29	14.53	7.82	**7.26**

S. Sengupta et al. Frontal to profile face verification in the wild. In WACV, 2016

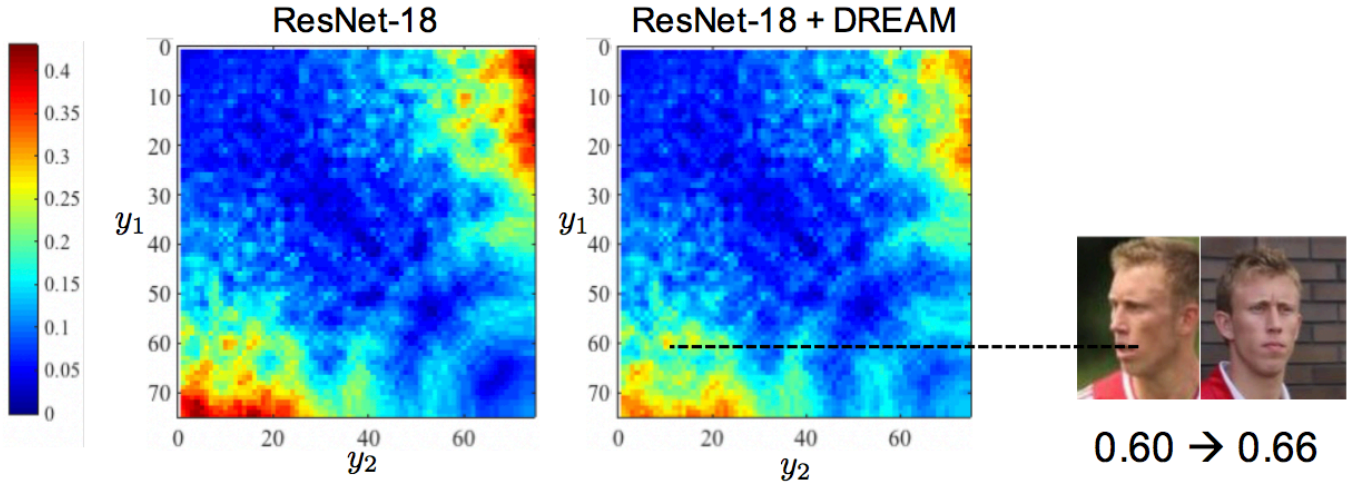
Results on IJB-A

Methods ↓	Verification		Identification	
Metrics →	TAR @ FAR=0.01	TAR @ FAR=0.001	Rec. Rate @ Rank-1	Rec. Rate @ Rank-5
Our Approach with MS-Celeb-1M subset:				
ResNet-18 (naïve)	0.840±0.026	0.656±0.040	0.897±0.016	0.951±0.011
ResNet-18 (end2end+retrain)	0.872±0.018	0.712±0.035	0.915±0.012	0.962±0.008
ResNet-50 (naïve)	0.881±0.018	0.714±0.034	0.913±0.013	0.957±0.010
ResNet-50 (end2end+retrain)	0.891±0.016	0.764±0.031	0.924±0.016	0.962±0.010
Our Approach with full MS-Celeb-1M:				
ResNet-18 (naïve)	0.934±0.009	0.836±0.016	0.939±0.012	0.960±0.010
ResNet-18 (end2end+retrain)	**0.944±0.009**	0.868±0.015	**0.946±0.011**	**0.968±0.010**
Existing Methods:				
Wang *et al.* [32]	0.729±0.035	0.510±0.061	0.822±0.023	0.931±0.014
Pooling Faces [8]	0.819± ——	0.631± ——	0.846± ——	0.933± ——
Deep Multi-Pose [1]	0.787± ——	–	0.846± ——	0.927± ——
PAMs [19]	0.826±0.018	0.652±0.037	0.840±0.012	0.925±0.008
$\text{DCNN}_{\text{fusion}}$ (f.t.) [4]	0.838±0.042	–	0.903±0.012	0.965±0.008
Augmentation+Video Pooling+Rendered Test [20]	0.886± 0.017	0.725± 0.044	0.906± 0.013	0.962± 0.007
$\text{CNN}_{\text{media}}$+TPE (f.t.) [24]	0.900±0.010	0.813±0.020	0.932±0.010	–
Template Adaptation (f.t.) [6]	0.939±0.013	–	0.928±0.010	–
Quality Aware Network (f.t.) [18]	0.942±0.015	0.893±0.039	–	–

Further analysis



(a) Heatmap of false positive rate



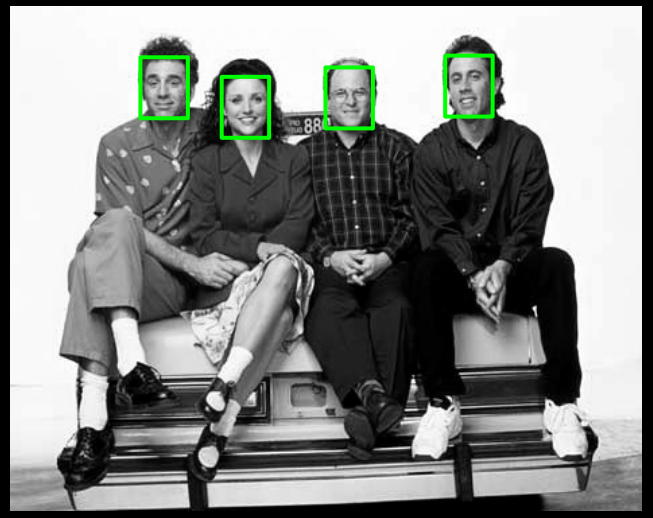
(b) Heatmap of false negative rate

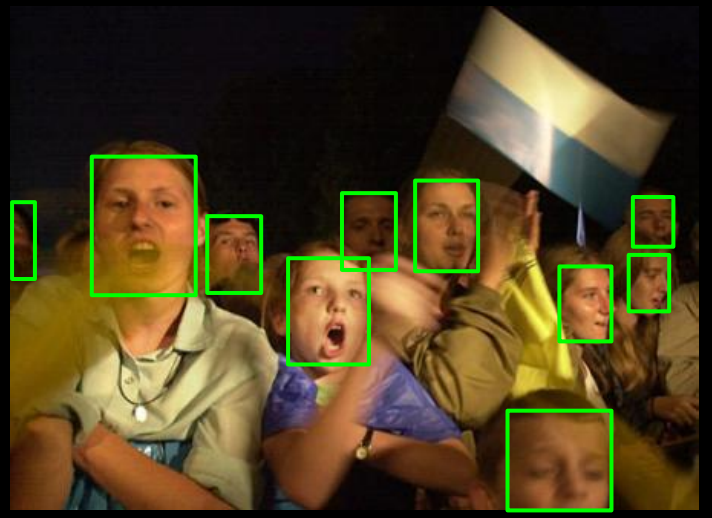
Summary

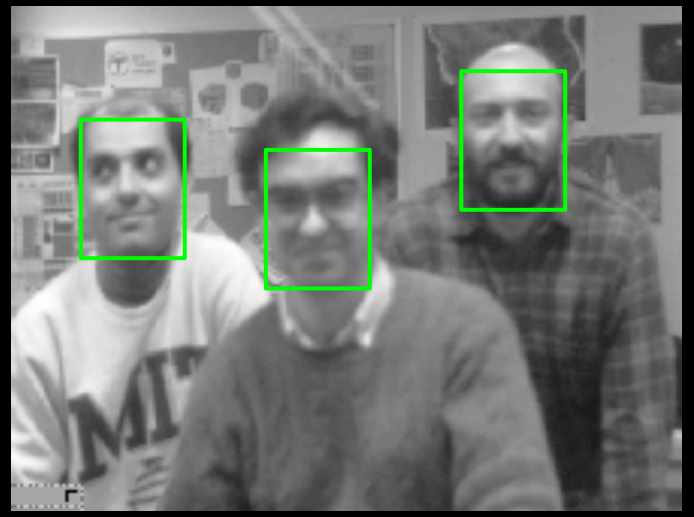
- Equivariant mapping in the deep feature space
- Performing frontalization in the feature space is more fruitful than the image space
- Easy to use, light-weight, and can be implemented with a negligible computational overhead.

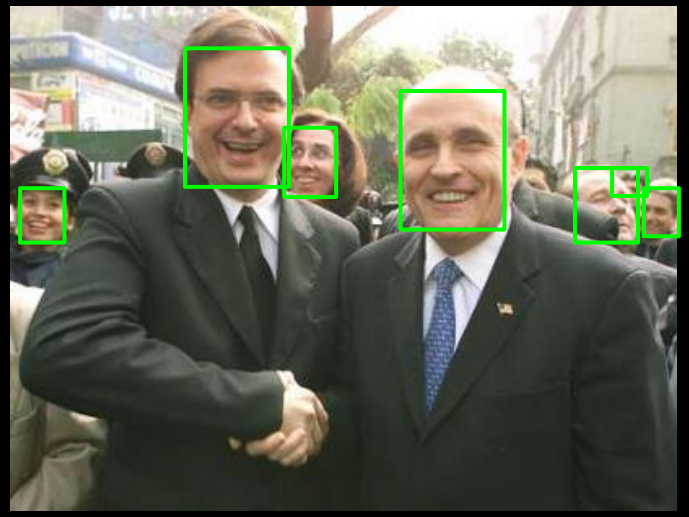

WIDER FACE

Diversity







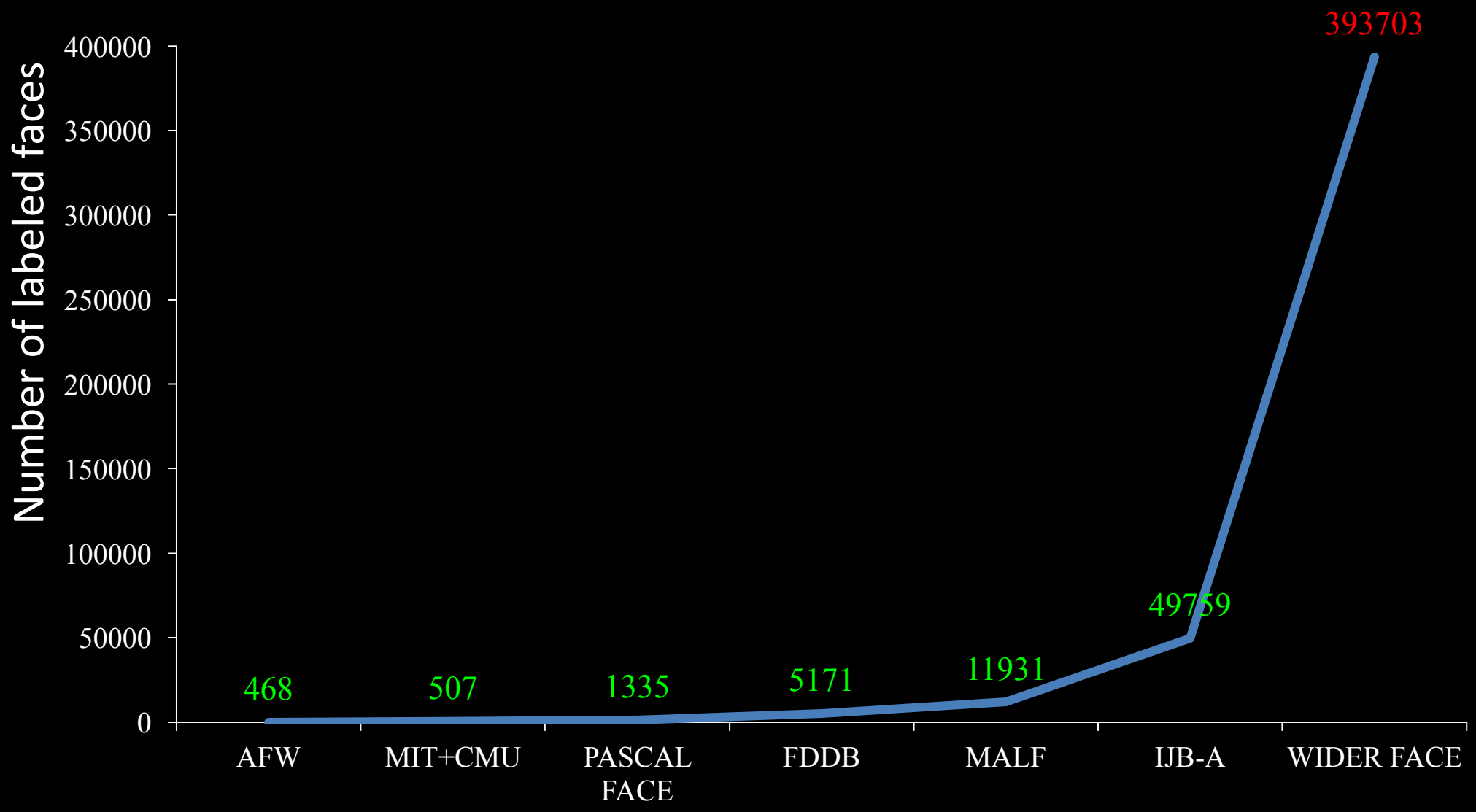


MIT+CMU

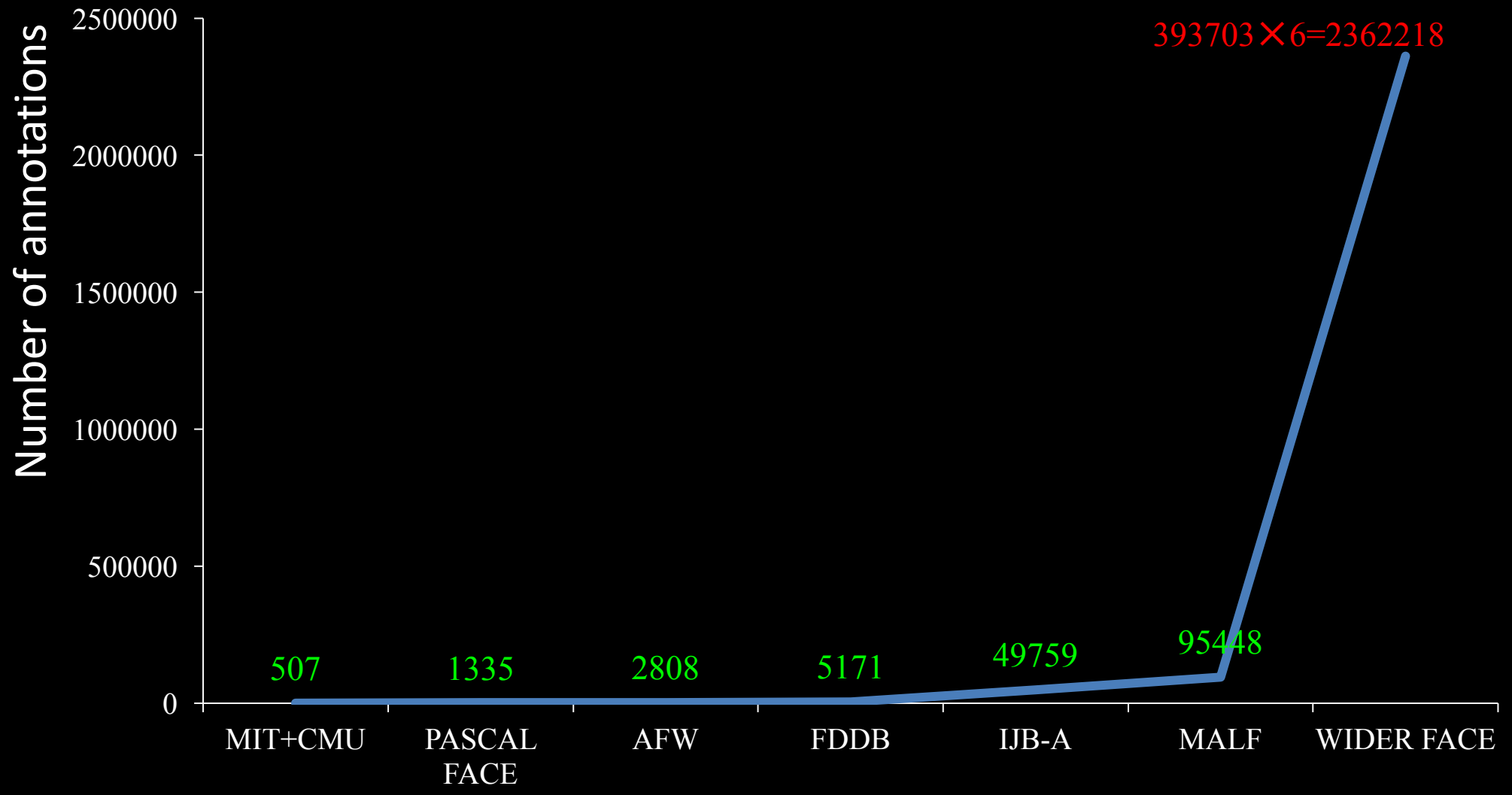
FDDB

WIDER FACE

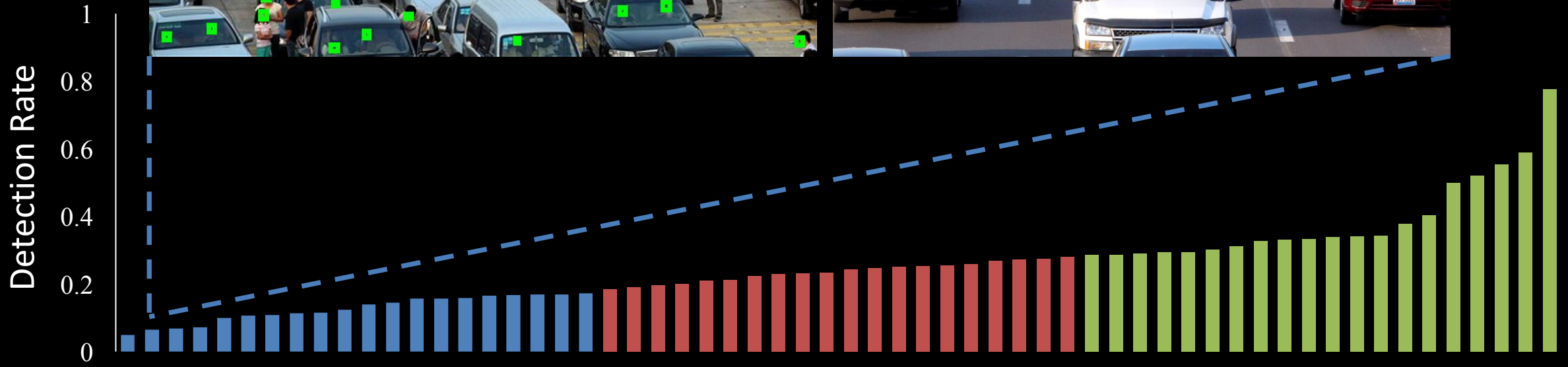
Data scale


Number of labeled faces
400000
350000
300000
250000
200000
150000
100000
50000
0
468
507
1335
5171
11931
49759
393703
AFW
MIT+CMU
PASCAL FACE
FDDB
MALF
IJB-A
WIDER FACE

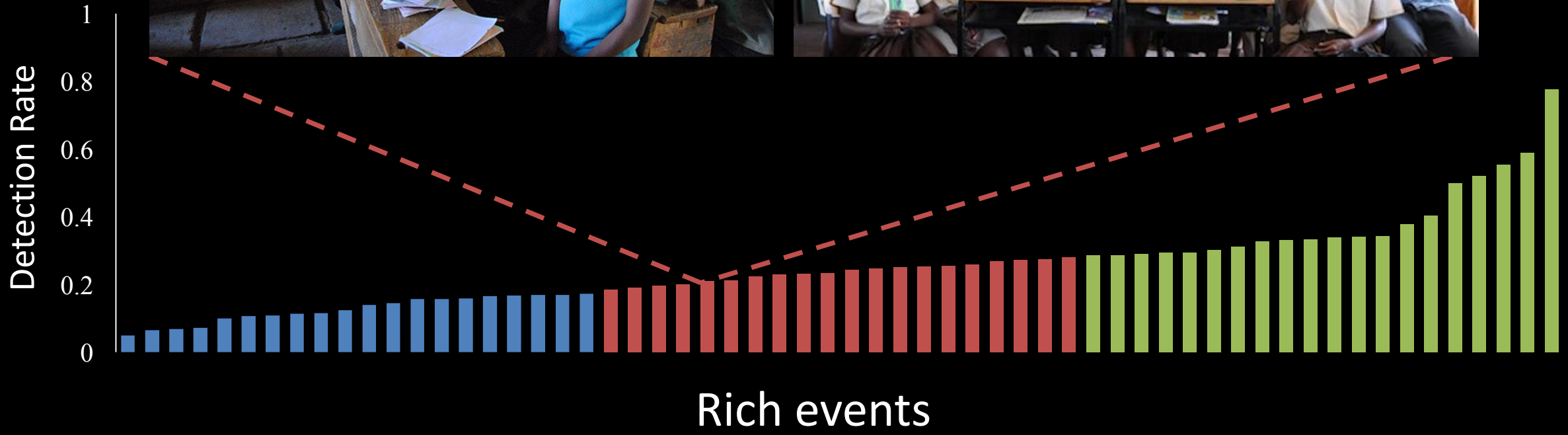
Richer annotations



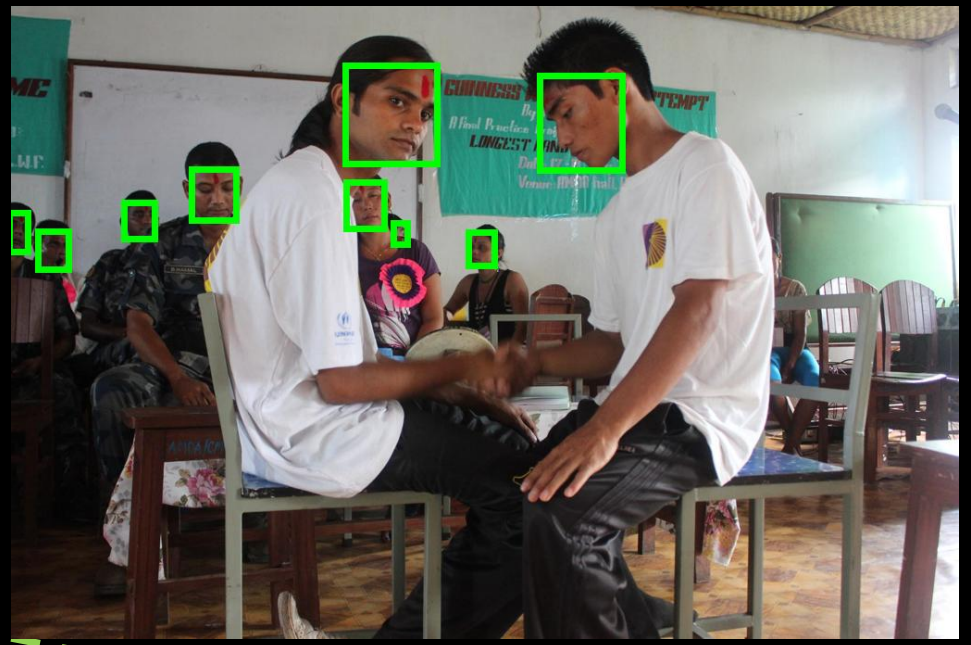
Traffic

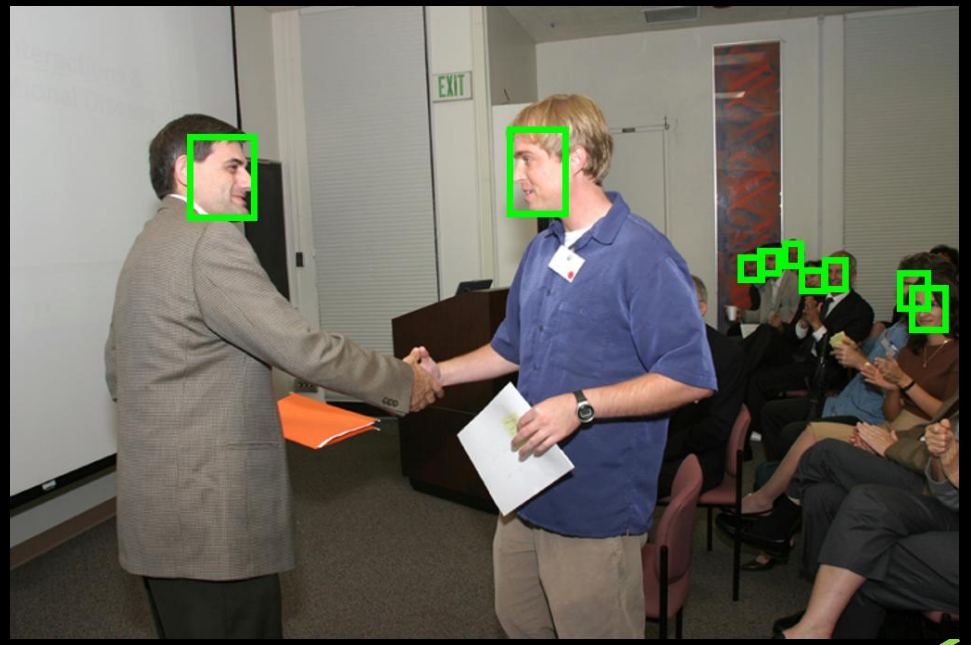


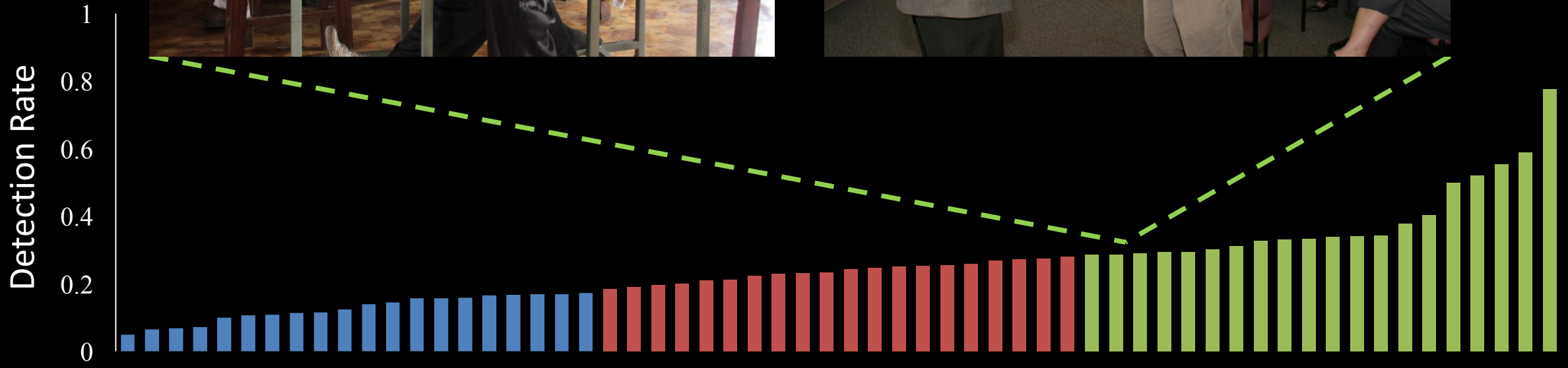
Students Schoolkids



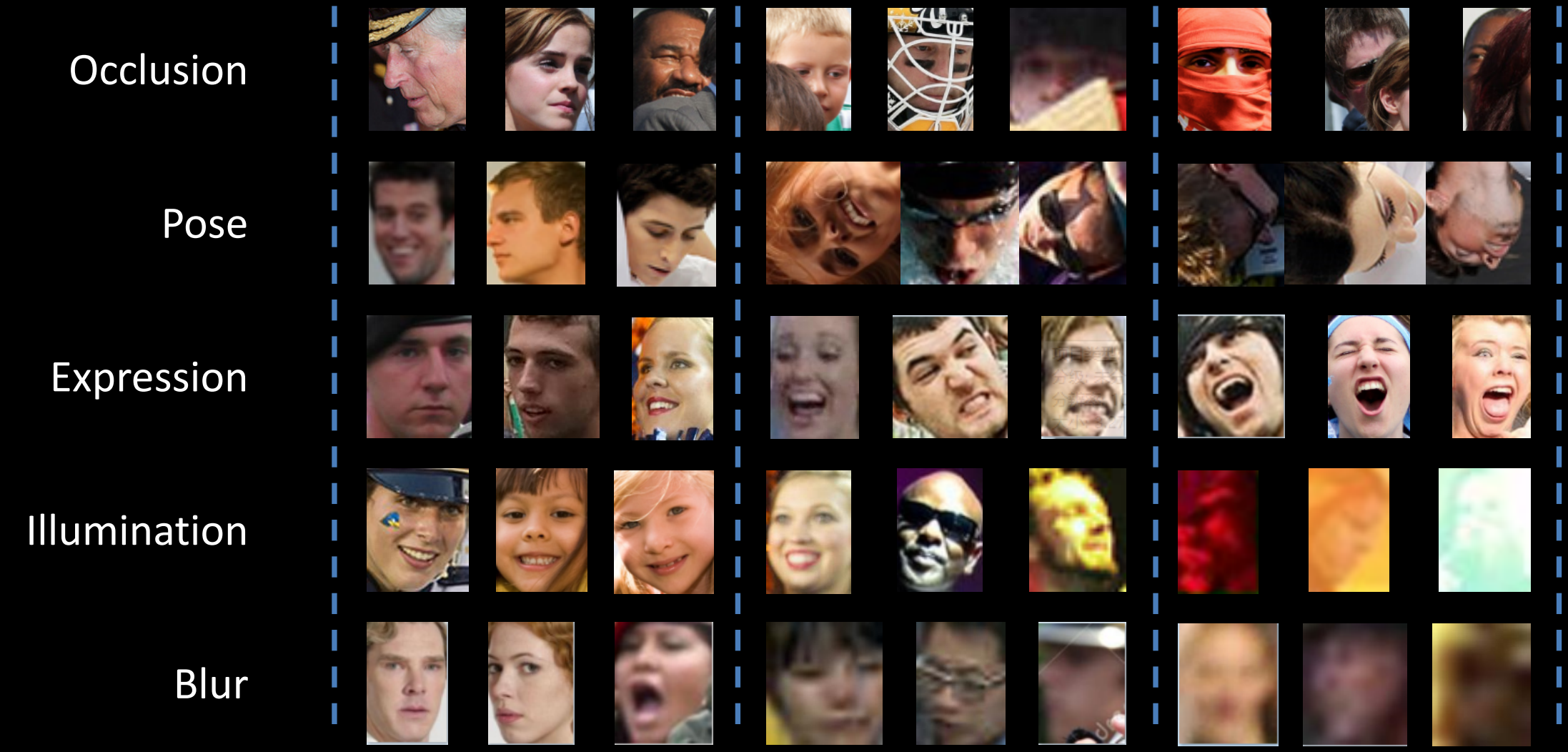
Handshaking



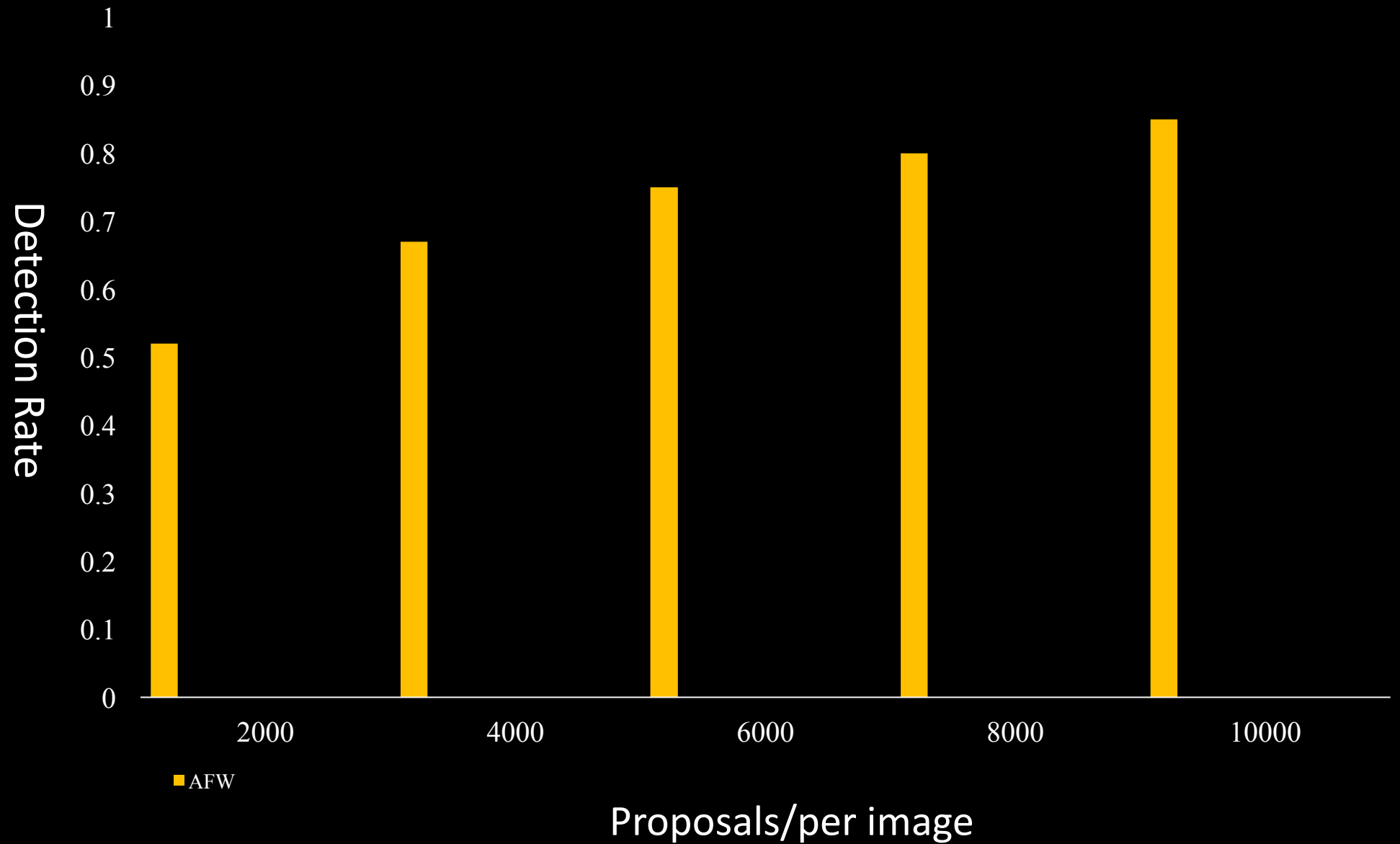




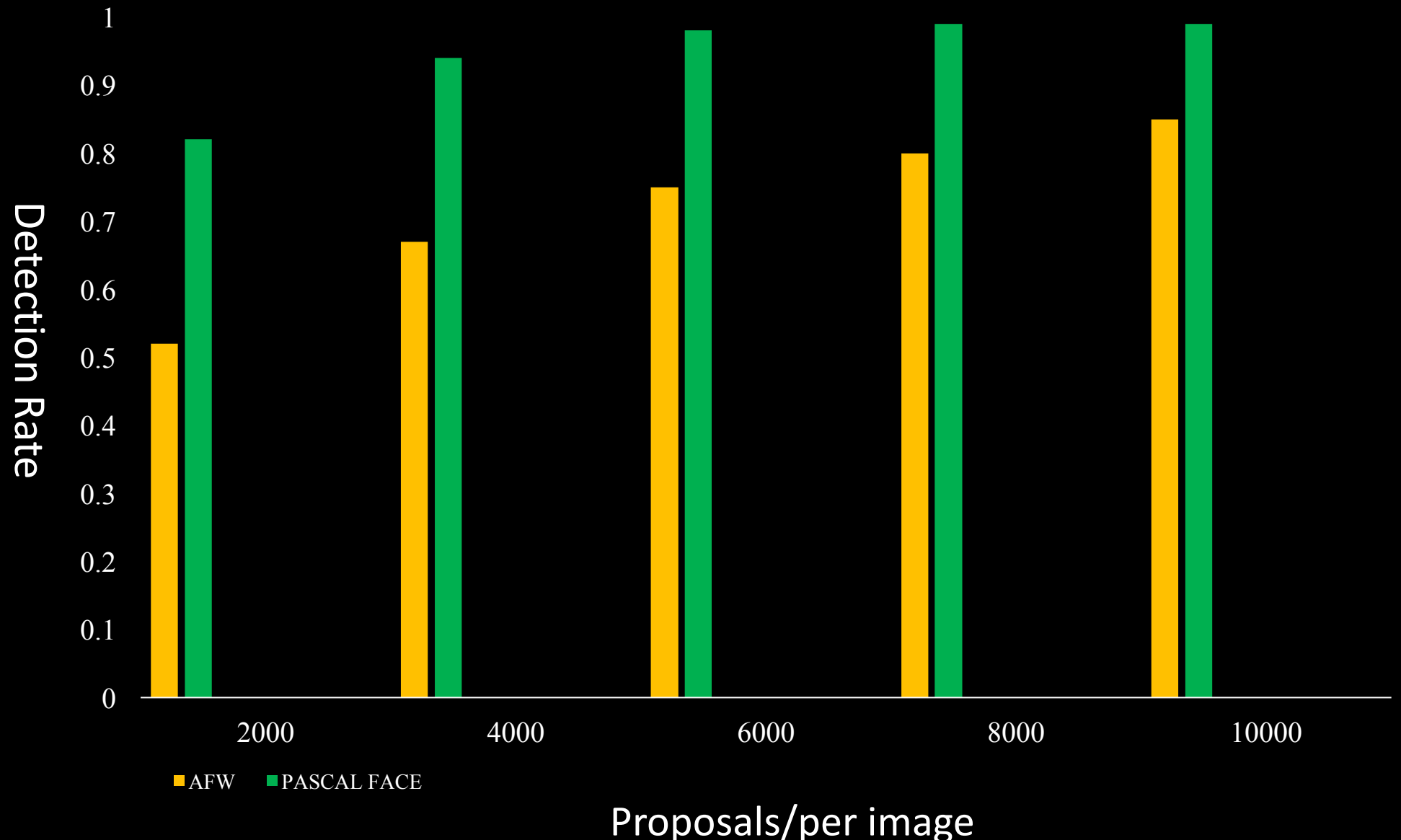
Rich label annotations



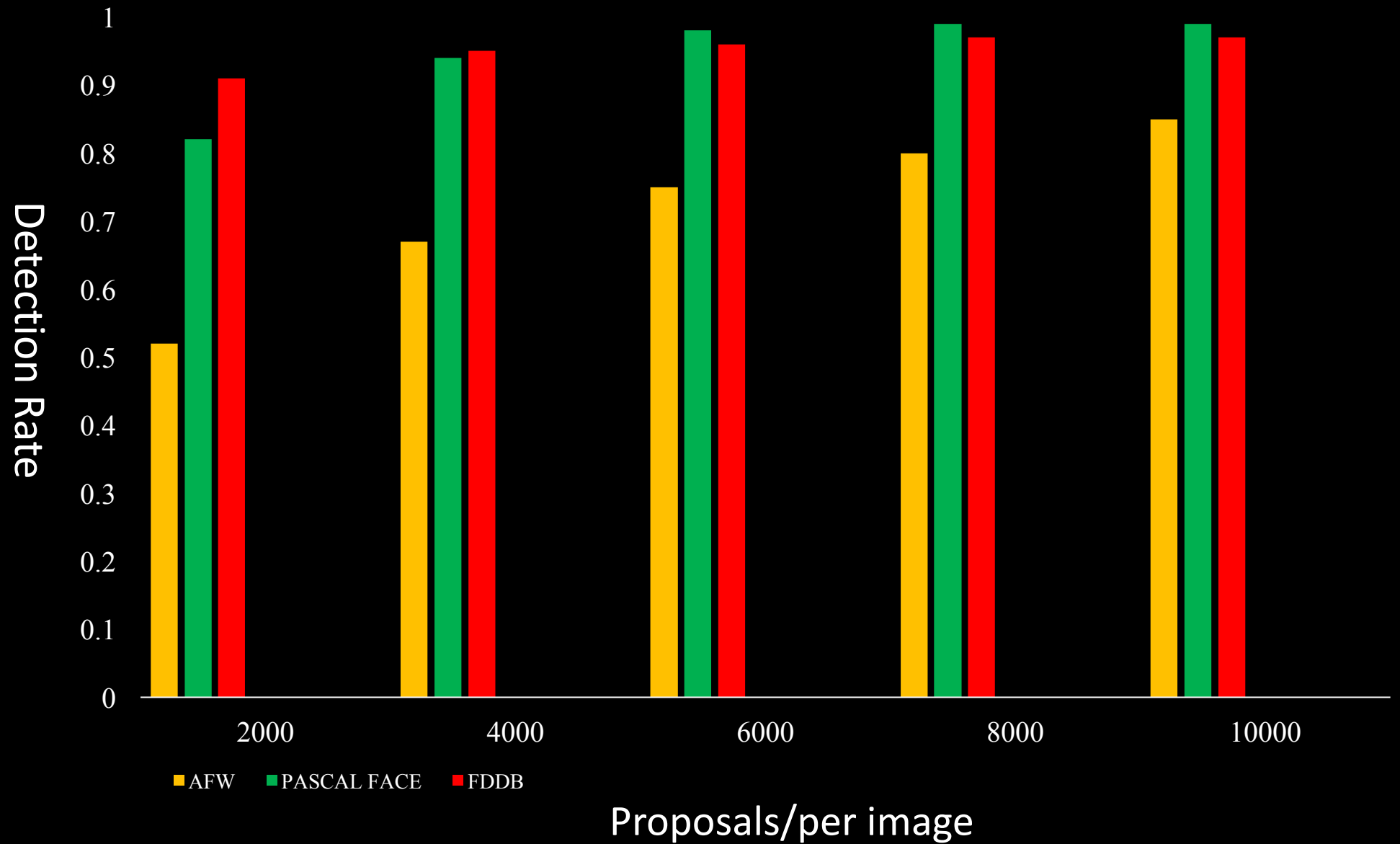
WIDER FACE is more challenging



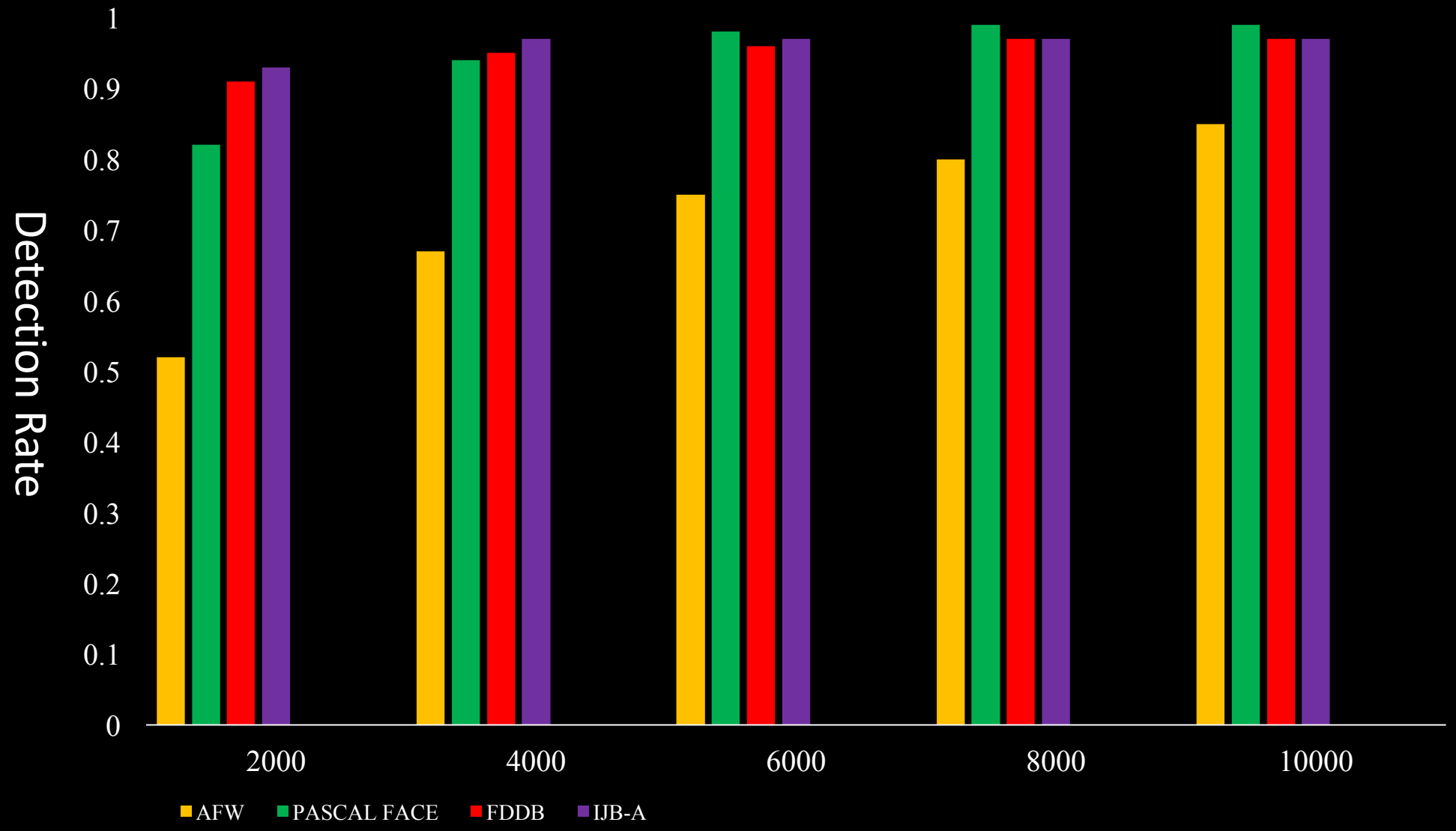
WIDER FACE is more challenging



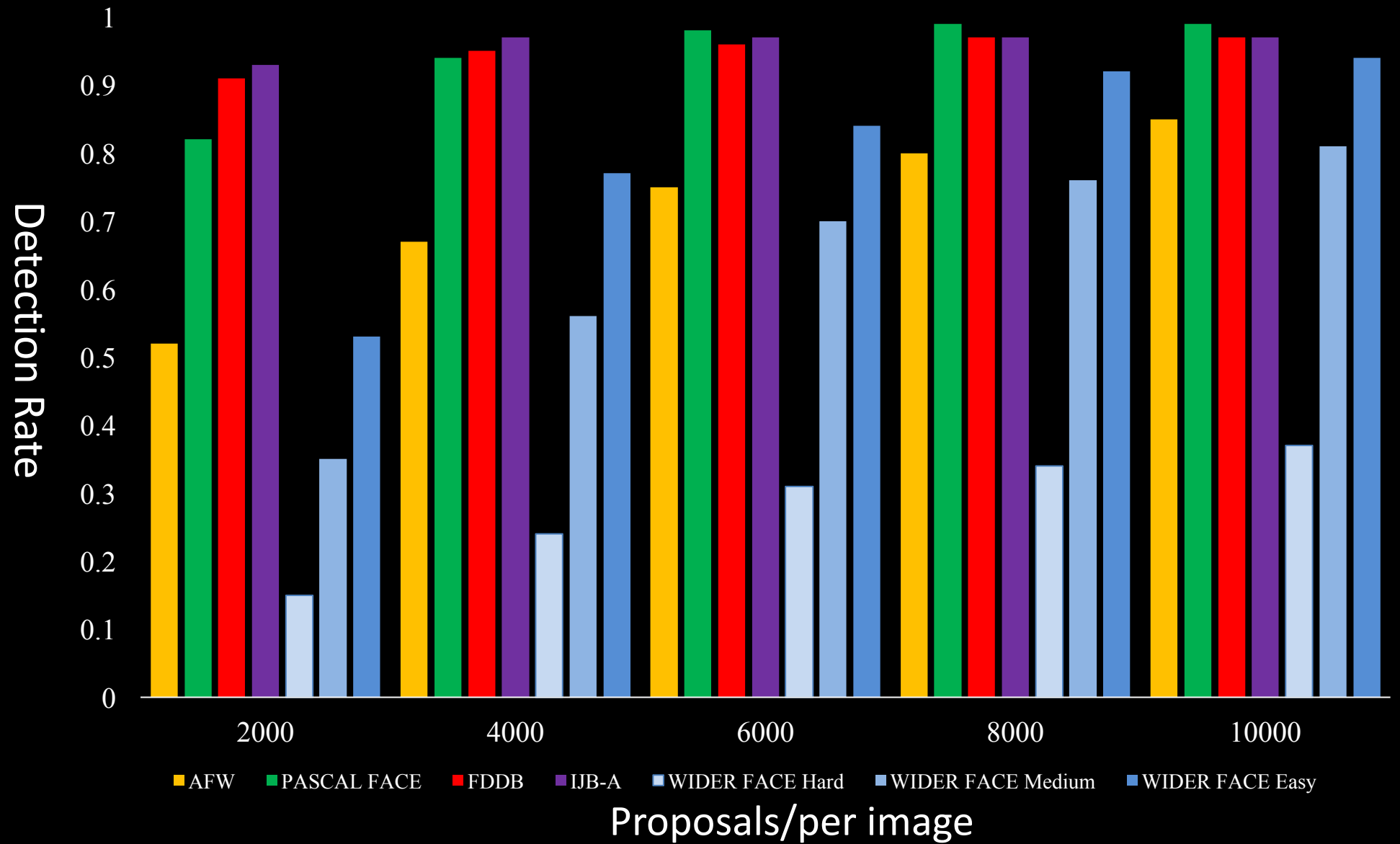
WIDER FACE is more challenging



WIDER FACE is more challenging



WIDER FACE is more challenging



WIDER FACE: A Face Detection Benchmark

Multimedia Laboratory, Department of Information Engineering, The Chinese University of Hong Kong

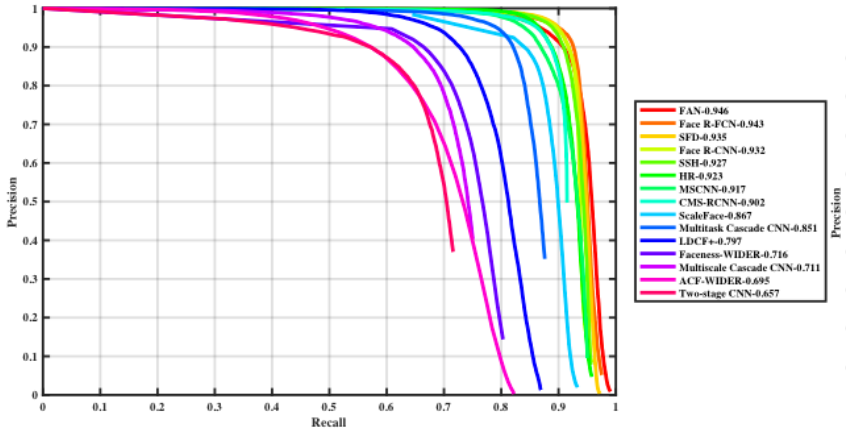
HOME RESULTS

News

- **2016-04-17** The face attribute labels i.e. pose and occlusion are available. NEW!
- **2015-11-19** Results of four baseline methods: ACF, Faceness, Multiscale Cascade CNN, and Two-stage CNN are released.
- **2015-11-19** WIDER FACE v1.0 is released with images, face bounding box annotations, and event category annotations.

Webpage: http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/

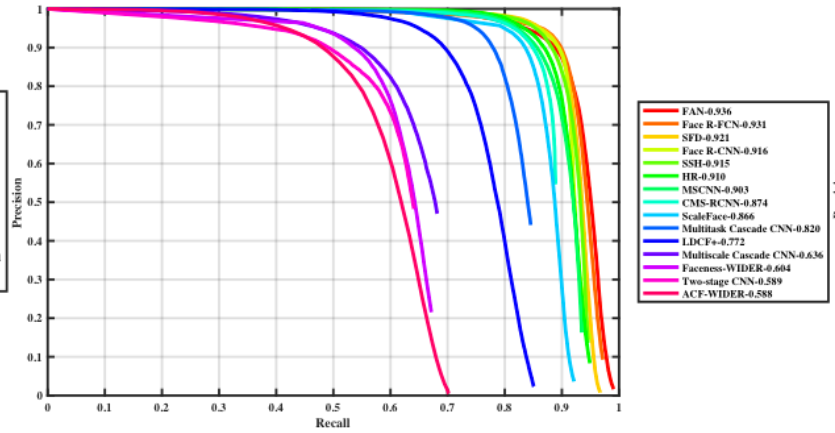
WIDER FACE Benchmark



Easy

Average precision

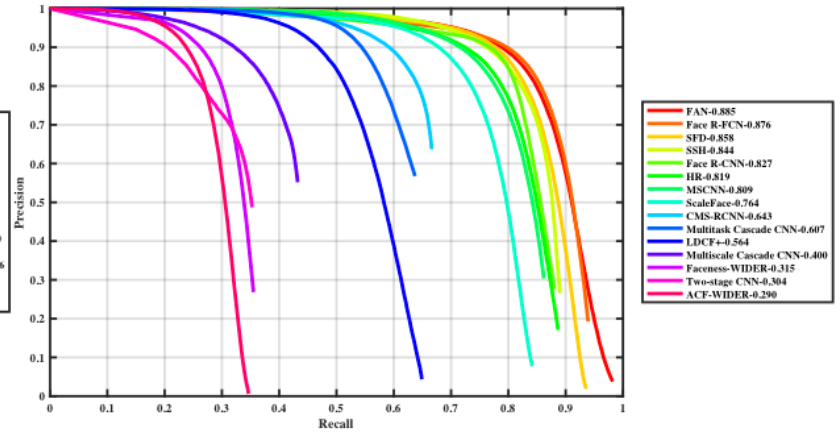
FAN – 0.946
Face R-FCN – 0.943
SFD - 0.935
…
2015 method - 0.711



Medium

Average precision

FAN – 0.936
Face R-FCN – 0.931
SFD - 0.921
…
2015 method - 0.636



Hard

Average precision

FAN – 0.885
Face R-FCN – 0.876
SFD - 0.858
…
2015 method - 0.400

Is there anything else I can solve?

- While maintaining good detection performance
  - Light-weight architecture and speed
  - Training with fewer annotated data
  - Coping with noisy annotations
  - …

Face Detection

Face Detection through Scale-Friendly Deep Convolutional Networks

S. Yang, Y. Xiong, C. C. Loy, X. Tang

https://arxiv.org/pdf/1706.02863.pdf, 2017

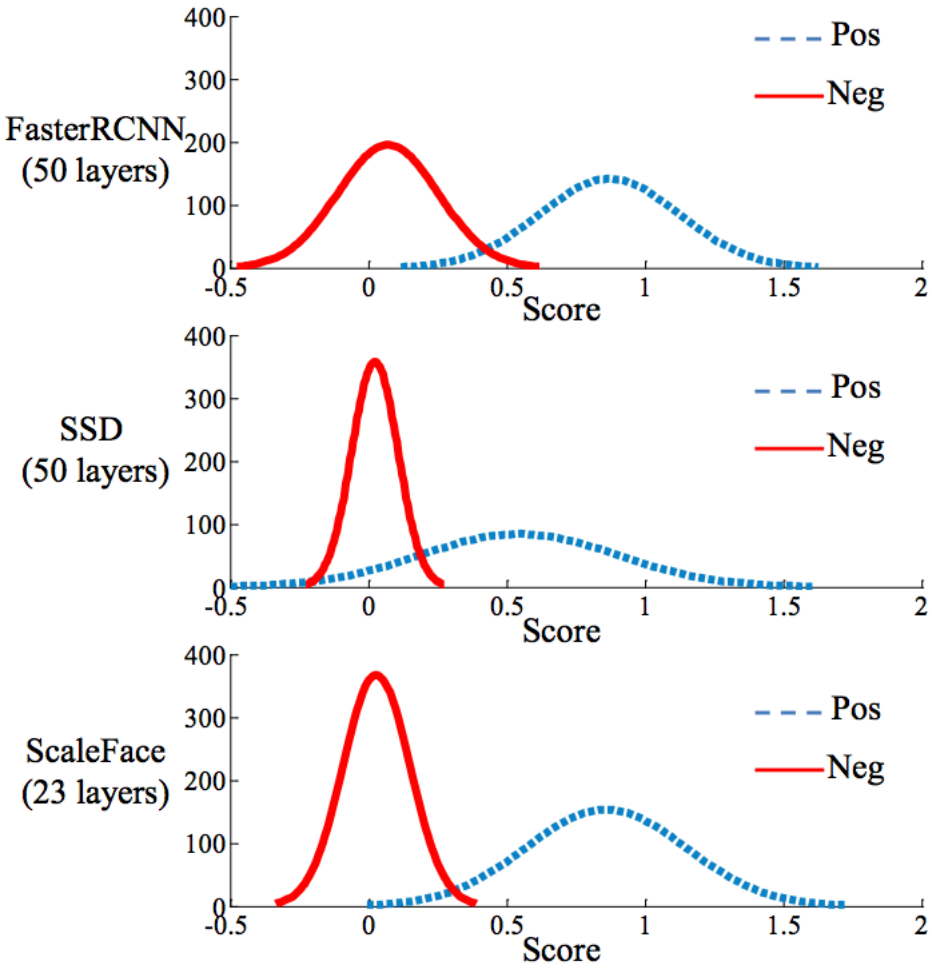
Problem

- The clues to be gleaned for recognizing a 300-pixels tall face are qualitatively different than those for recognizing a 10-pixels tall face
- More convolution layers are required to learn highly representative features that can distinguish faces with large appearance variations
- By going deeper, the spatial information will lose through pooling or convolution operations
- Dilated convolution? Remove pooling?

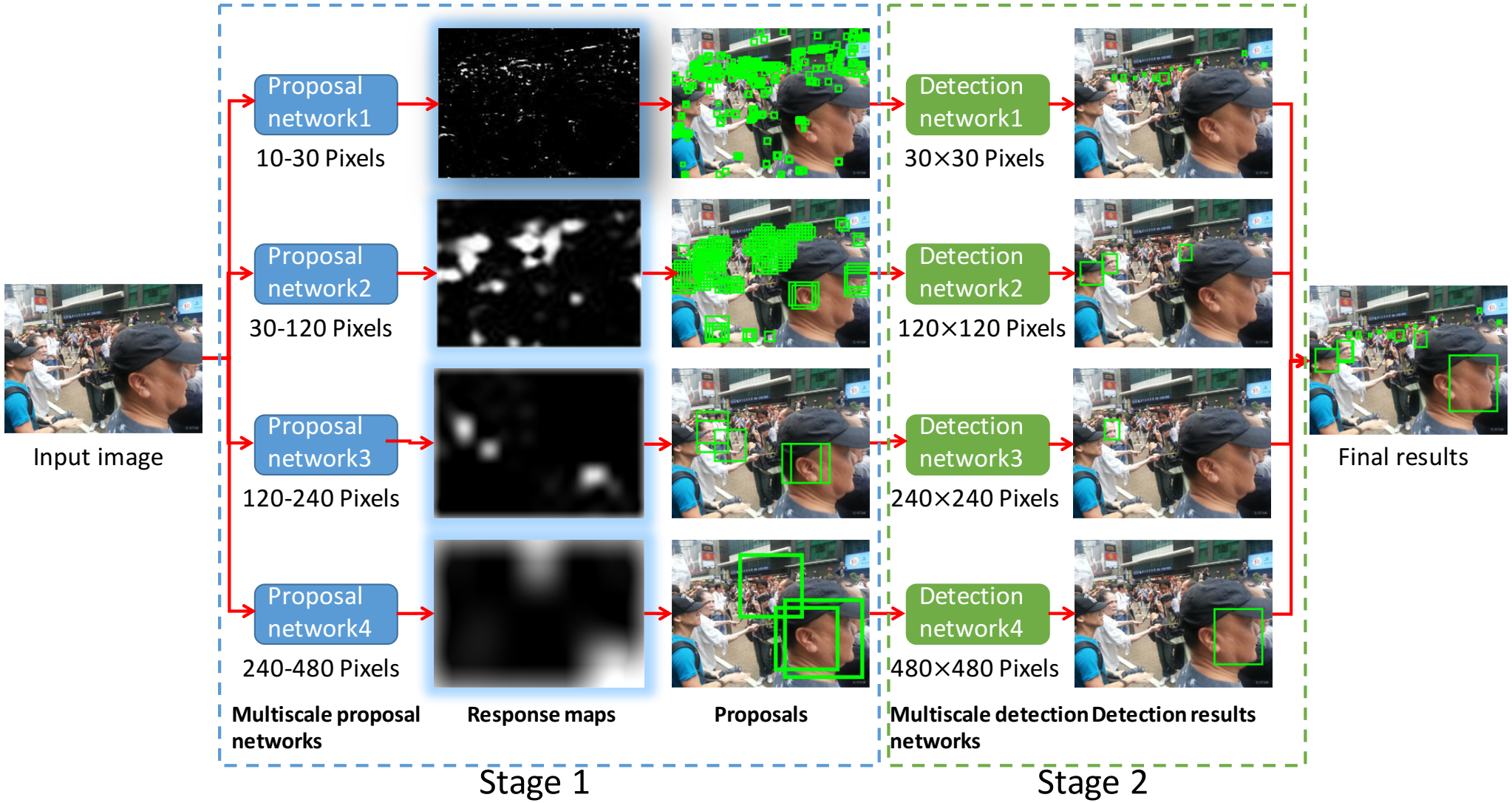
Motivation

- Faces with different scales possess different inherent visual cues and thus lead to disparate detection difficulties

- Use different specialized network structures



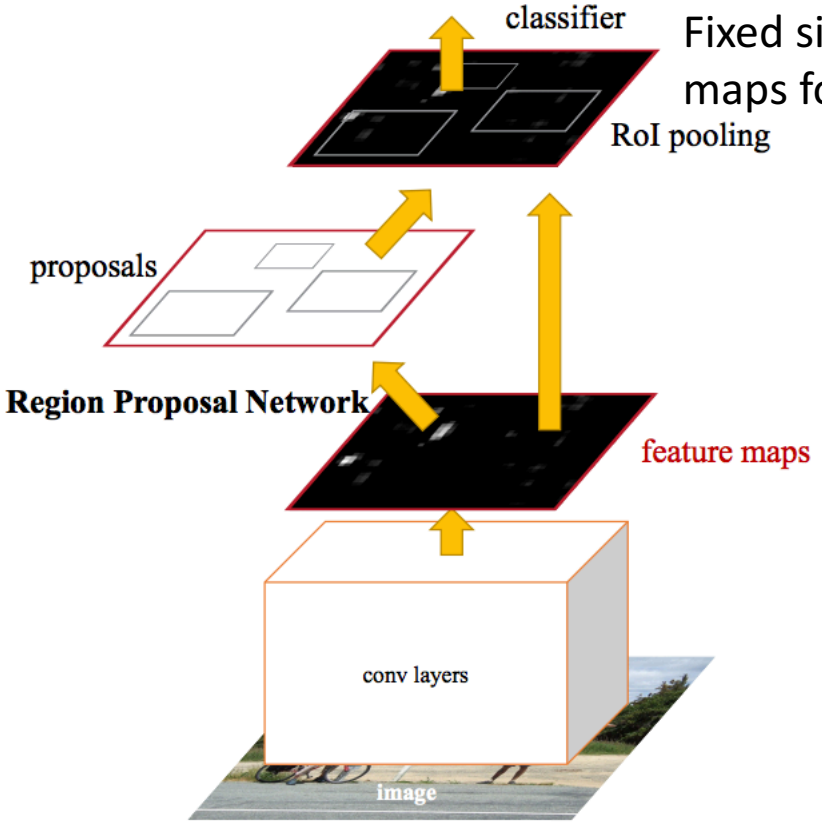
A naïve solution



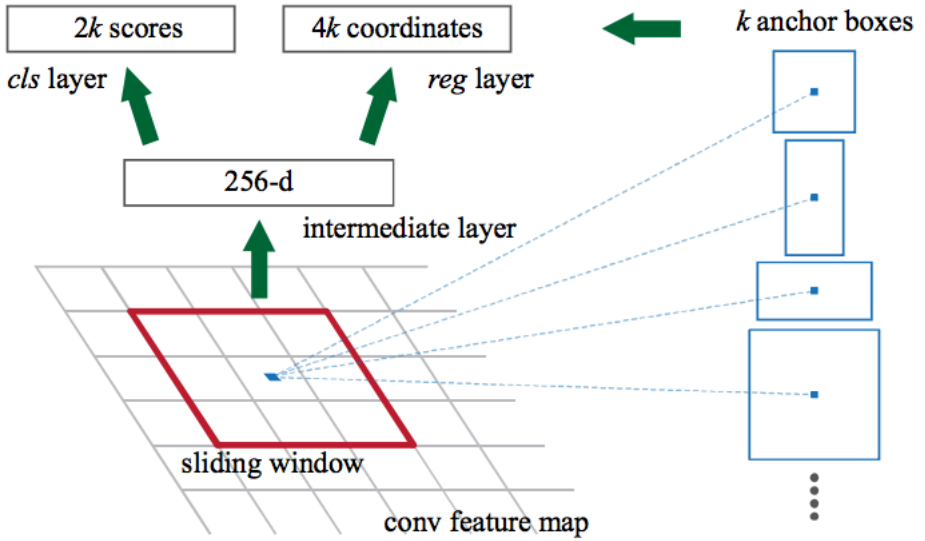
Solution

- Splits a large range of target scales into a set of sub-ranges
- Each subrange is modeled by a specialized network with carefully designed depth and spatial pooling to optimize the receptive field for the particular range
- Combine sub-nets into a single network and optimize them end-to-end
- Previous state-of-the-art = average precision of 81%, and runs at 0.6 fps.
- Ours = average precision of **76.4%** with just **7** fps

Faster R-CNN

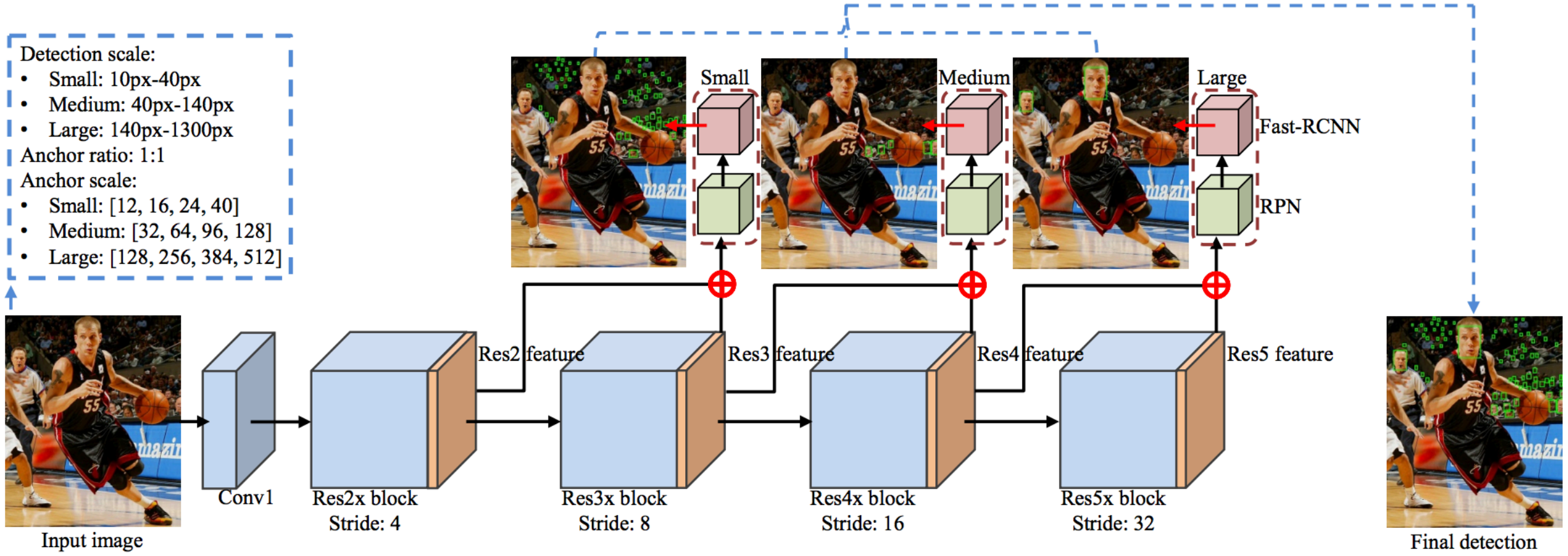


Fixed size feature maps for each ROI



Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

ScaleFace

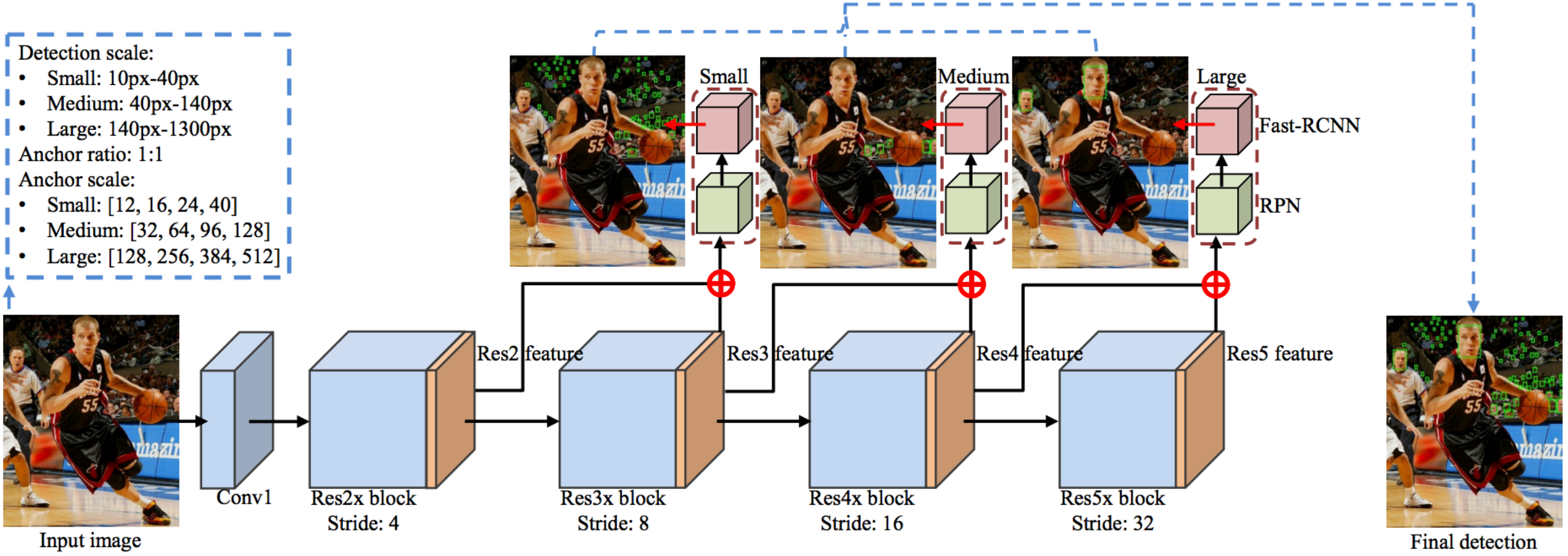


Contains three scale-variant detectors with different size of spatial pooling stride and depth

Scale-variant detectors are integrated into a single backbone network by sharing representation (ResNet-50)

Single-scale inference -- using a single input image without an image pyramid.

ScaleFace



Given a test image, a forward pass is performed and each scale-variant face detector will generate detection windows independently

Finding a network for specific scale range

- Faces with different scales can be better modeled by networks with different spatial pooling structure

Experiment

- Group faces into three classes according to the image height:
  - small(10px – 40px), medium (40px – 140px), and large(140px or more).
- For each face group, we train four deep networks with different spatial pooling structure.

Finding a network for specific scale range

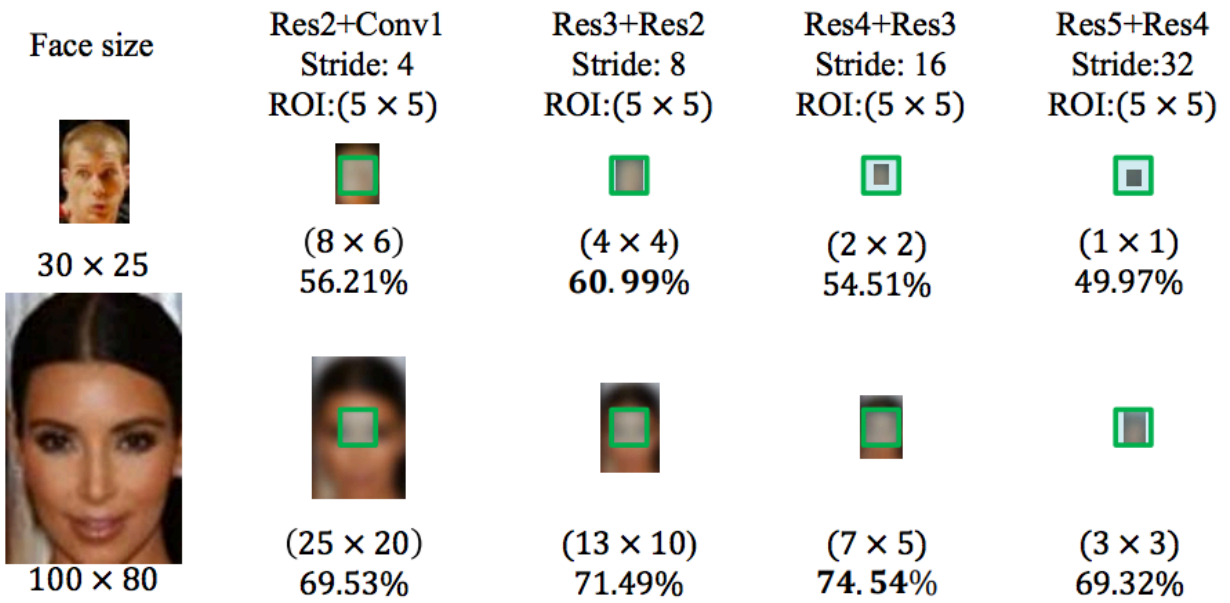
Face scale	Conv1+Res2 stride=4	Res2+Res3 stride=8	Res3+Res4 stride=16	Res4+Res5 stride=32
Small	56.21%	**60.99%**	54.51%	49.97%
Medium	69.54%	71.49%	**74.54%**	69.33%
Large	58.43%	72.19%	81.89%	**84.68%**

- The best performance of certain scales when the projected face scale on the feature map is close to the ROI template

Finding a network for specific scale range

- Convolutional features at higher layers tend to have smaller projected ROI size
- The detection performance of a target scale consistently decreases when the ROI on the target layer is smaller than ROI pooling size
- Even if we increase the depth of the network which will generally improve the discriminative power of the feature representation, the detection performance still drops

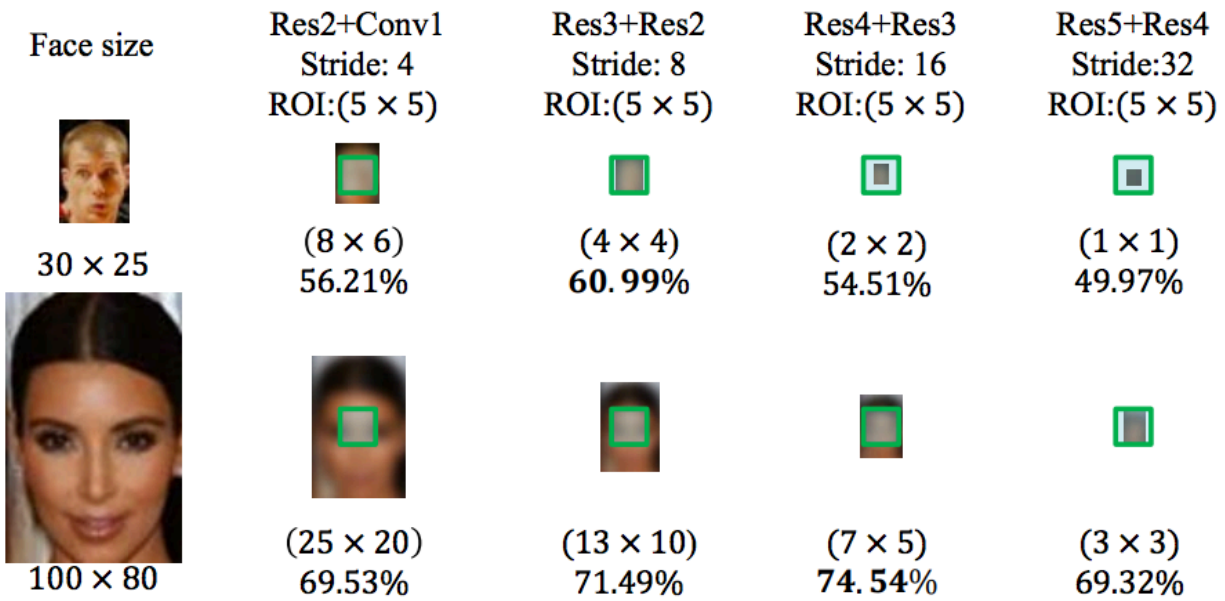
Face scale	Conv1+Res2 stride=4	Res2+Res3 stride=8	Res3+Res4 stride=16	Res4+Res5 stride=32
Small	56.21%	**60.99%**	54.51%	49.97%
Medium	69.54%	71.49%	**74.54%**	69.33%
Large	58.43%	72.19%	81.89%	**84.68%**



The green box represents the ROI template

Finding a network for specific scale range

- Remapped features with a similar size of ROI template will yield the best detection performance.
- If the projected region is much larger than the ROI template discriminative information will loss during pooling procedure.
- On the other hand, if the projected region is much smaller than the ROI template, the insufficient information and overlapping between features will cause a performance drop.



How many scale-variant detectors

Small

Medium

Large

- **Small faces** (less than 40 pixel height)
  - lose most appearance information and can be characterized by rigid structures and context.
- **Medium faces** (40px − 140px)
  - have high variance since persons in these images are usually not the main subjects of the photographer, and therefore they can be of various poses looking at different directions.
- **Large faces** (140px or more)
  - usually have low variance as they are the main subjects when a photo is captured. These large faces are usually in a frontal or profile pose.

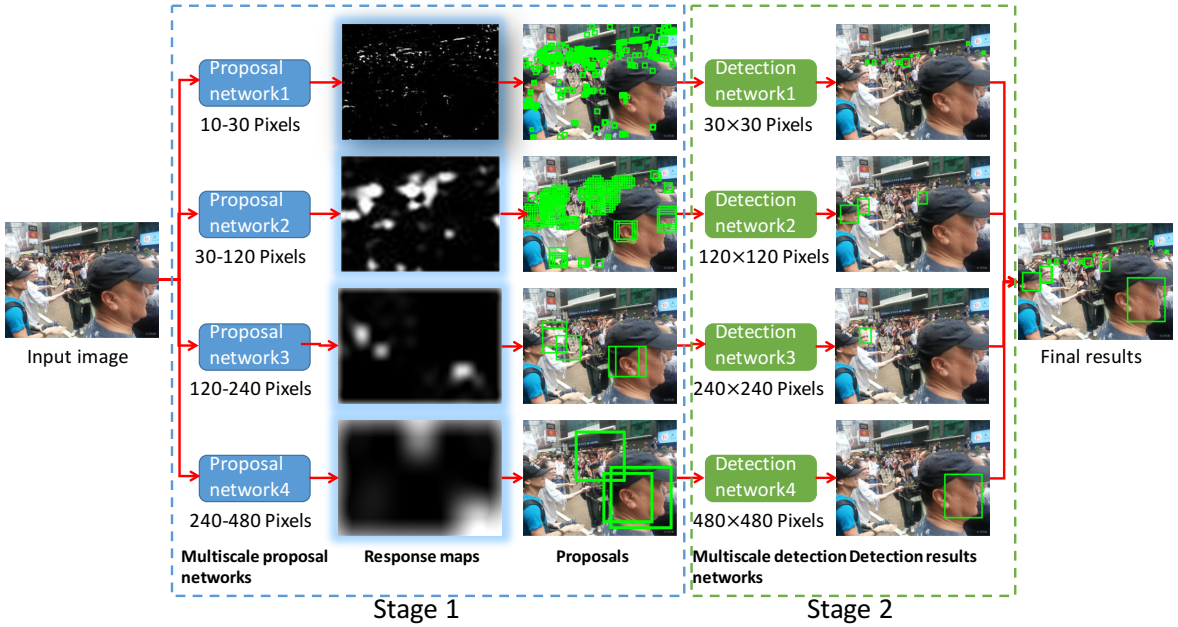
How many scale-variant detectors

Method	Split ranges
One split	[10, 1300]
Two splits	[10, 140], [140, 1300]
Two evenly splits	[10, 650], [650, 1300]
Three splits	[10, 40], [40, 140], [140, 1300],
Three evenly splits	[10, 450], [450, 900], [900, 1300],
Four splits	[10, 25], [25, 60], [60, 140], [140, 1300]
Four evenly splits	[10, 300], [300, 600], [600, 900], [900, 1300]

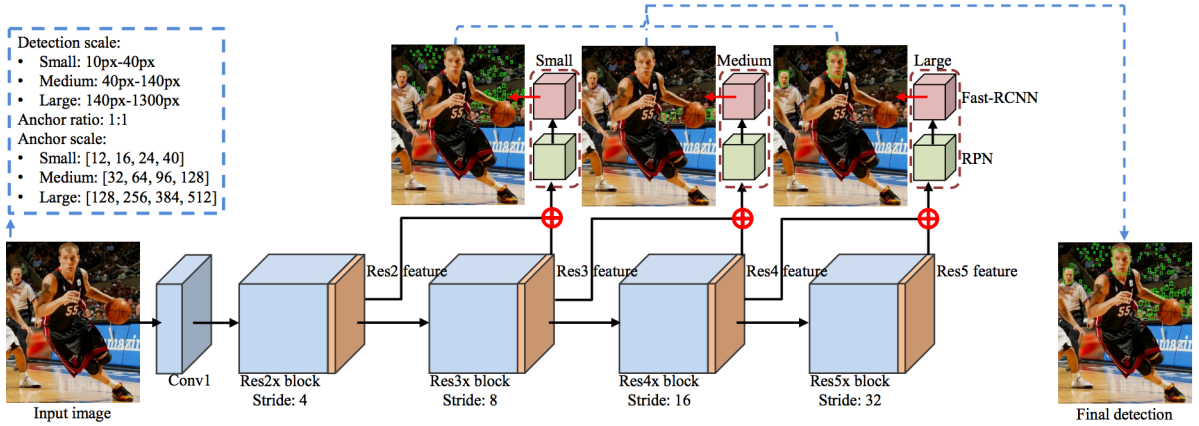
	Conv1+Res2 stride=4	Res2+Res3 stride=8	Res3+Res4 stride=16	Res4+Res5 stride=32
One split				✓
Two splits			✓	✓
Three splits		✓	✓	✓
Four splits	✓	✓	✓	✓

Method	Easy	Medium	Hard
One split	82.4%	79.3%	62.4%
Two splits	83.0%	83.5%	74.7%
Three splits	**86.8%**	**86.7%**	**77.2%**
Four splits	84.2%	85.1%	72.1%
Two evenly splits	78.4%	79.5%	62.8%
Three evenly splits	72.2%	73.6%	57.1%
Four evenly splits	68.4%	69.0%	53.1%

How to combine the scale-variant detectors



Method	Easy	Medium	Hard
Navies ensemble	73.2%	74.1%	60.8%
Joint optimize	**86.8%**	**86.7%**	**77.2%**

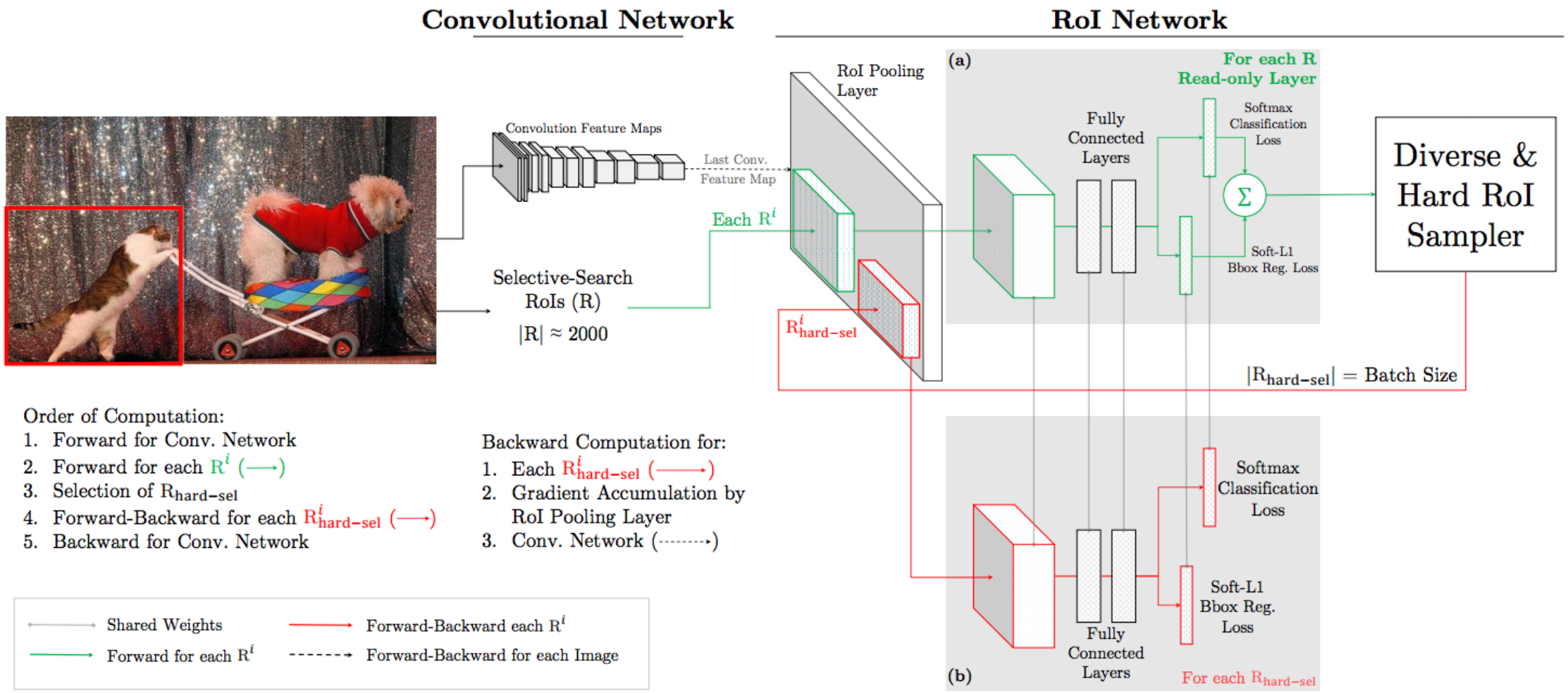


Online hard negative mining

- Detection datasets contain an overwhelming number of easy examples and a small number of hard examples.
- Automatic selection of these hard examples can make training more effective and efficient
- Training examples are sampled according to a non-uniform, non-stationary distribution that depends on the current loss of each example under consideration

Shrivastava et al., Training Region-based Object Detectors with Online Hard Example Mining, CVPR 2016

Online hard negative mining



Shrivastava et al., Training Region-based Object Detectors with Online Hard Example Mining, CVPR 2016

Online hard negative mining

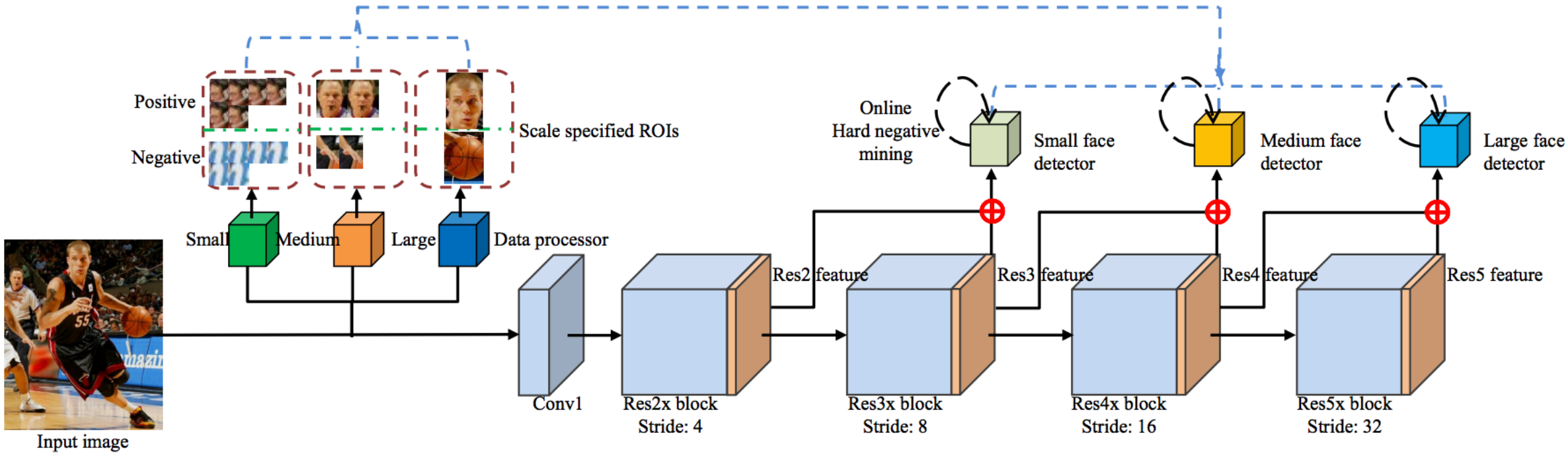
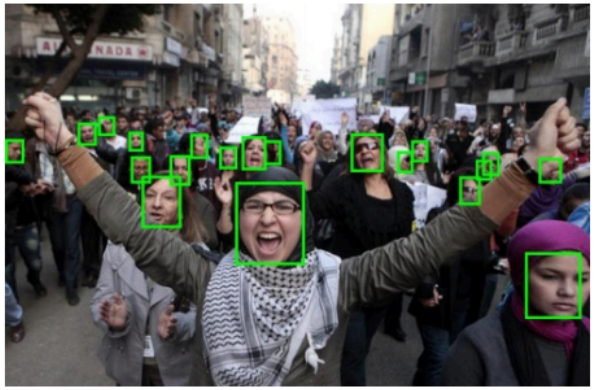


Figure 4. The data processing procedure during the training stage of ScaleFace.

Results

Small



Medium

Large

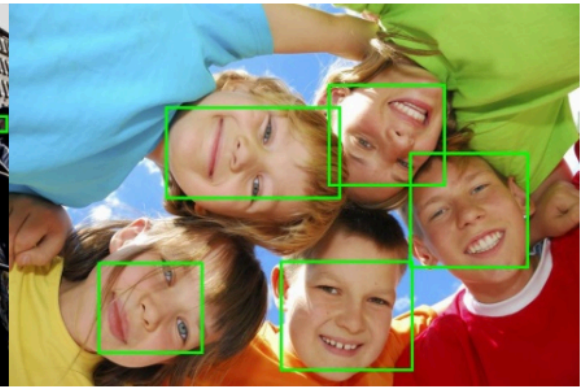
Final

Blur

Tiny

Illumination

Occlusion



Pose

Results

Results

Evaluation of different range partitioning schemes across three difficulty settings of WIDER FACE (Easy, Medium, Hard)

Method	Easy	Medium	Hard
Faceness-Net [23]	71.6%	60.4%	31.5%
LDCF+ [16]	79.7%	77.2%	56.4%
MTCNN [27]	85.1%	82.0%	60.7%
CMS-RCNN [28]	90.2%	87.4%	64.3%
HR [7]	92.3%	91.0%	81.9%
SSD [13]	89.9%	85.4%	62.5%
FasterRCNN [20]	89.5%	87.1%	71.6%
ScaleFace	86.7%	86.6%	76.4%

Tested using NVIDIA Titan X GPU by averaging the runtime of 1, 000 images randomly sampled from the WIDER FACE dataset

Method	AP	Runtime (ms)
FastRCNN	71.2%	140
SSD	62.4%	110
HR	81.9%	1,600
ScaleFace	76.4%	270
ScaleFace-Fast	75.5%	160

Face Attribute Recognition

Learning Deep Representation for Imbalanced Classification

C. Huang, Y. Li, C. C. Loy, X. Tang

in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016

Code available: http://mmlab.ie.cuhk.edu.hk/projects/LMLE.html

CelebA face attributes dataset

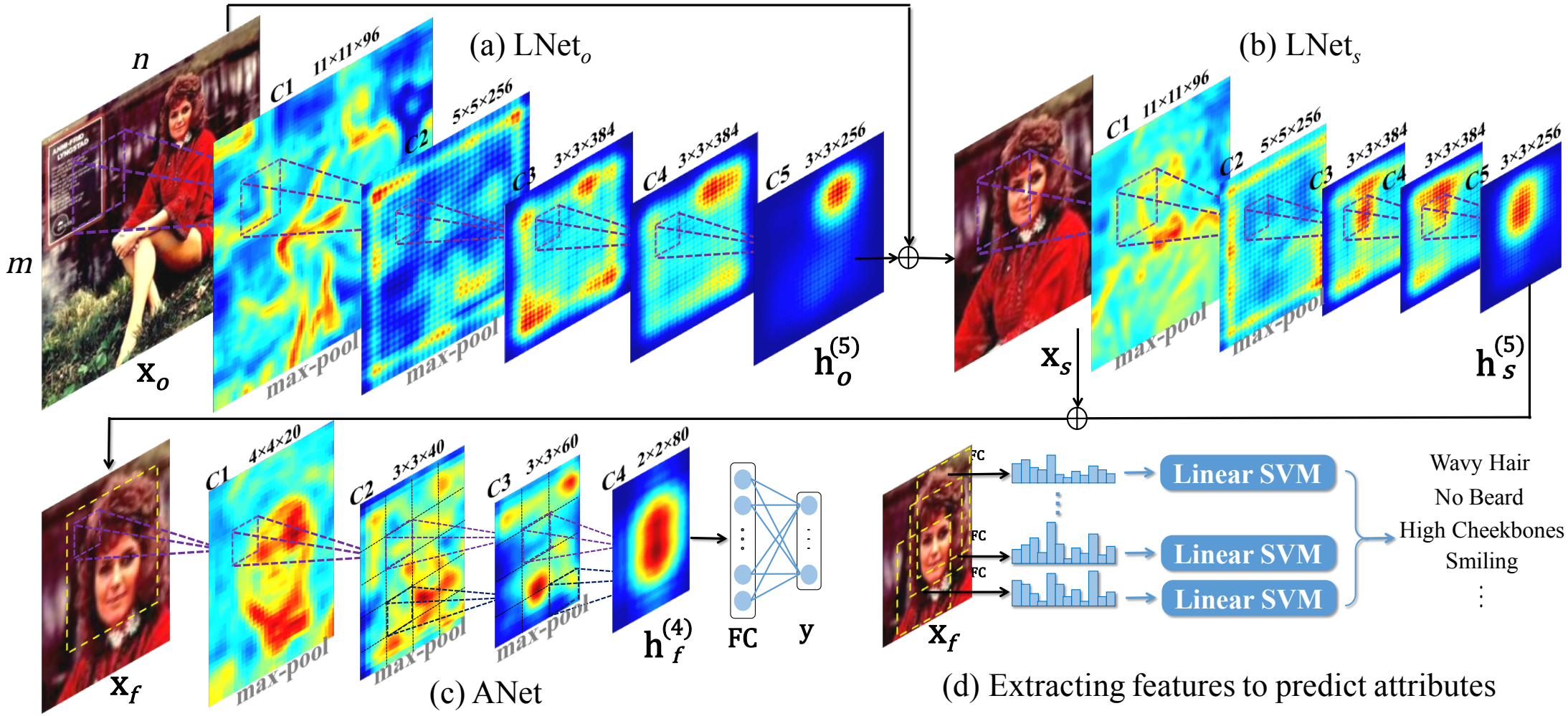
200K celebrity images, each with **40** attribute

Liu et al. “Deep Learning Face Attributes in the Wild”, ICCV 2015

http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

CelebA face attributes dataset

Previous work



Liu et al. “Deep Learning Face Attributes in the Wild”, ICCV 2015

Previous work

- *Classification accuracy* biased to the majority class
  - $accuracy = \left(\frac{tp + tn}{Np + Nn}\right)$

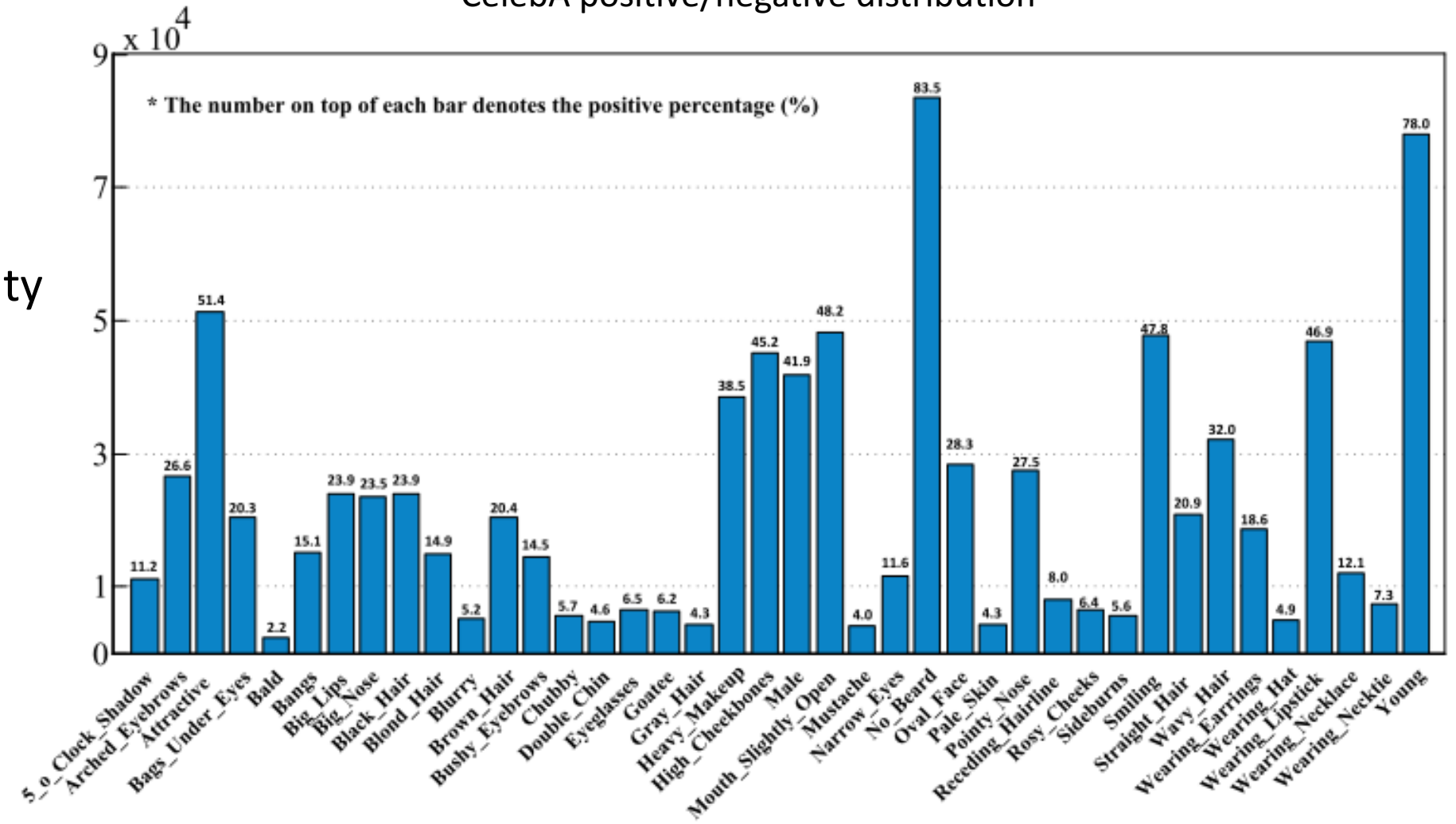
- We adopt a *balance accuracy*
  - $accuracy = \frac{1}{2}\left(\frac{tp}{Np} + \frac{tn}{Nn}\right)$

  Np and Nn are the numbers of positive and negative samples, while tp and tn are the numbers of true positive and true negative.

A more fundamental problem

- Without handling imbalanced class issue
  - Prediction biases toward the majority class
  - Poor accuracy for the minority class

CelebA positive/negative distribution



Existing solutions

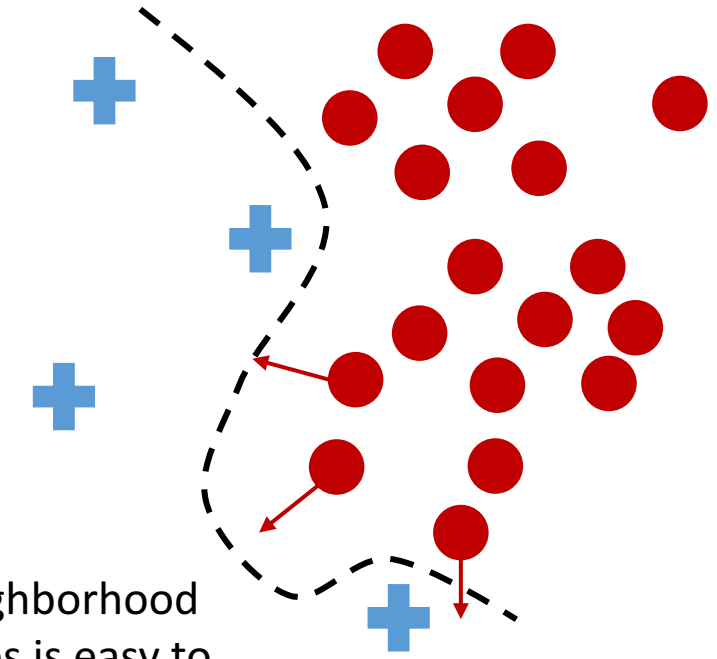
- Class re-sampling [Drummond & Holte, ICML'03]
  - Random under-sampling of majority class
    Remove valuable information
  - Random over-sampling of minority class
    Introduce artificial noise
- Cost-sensitive learning [Zadrozny et al., ICDM'03]
  - Assigns higher misclassification costs to the minority class
    How to design costs?

Motivation

- Is there a better way apart from sampling and cost learning?

Can we introduce tighter constrains to ameliorate such invasion?

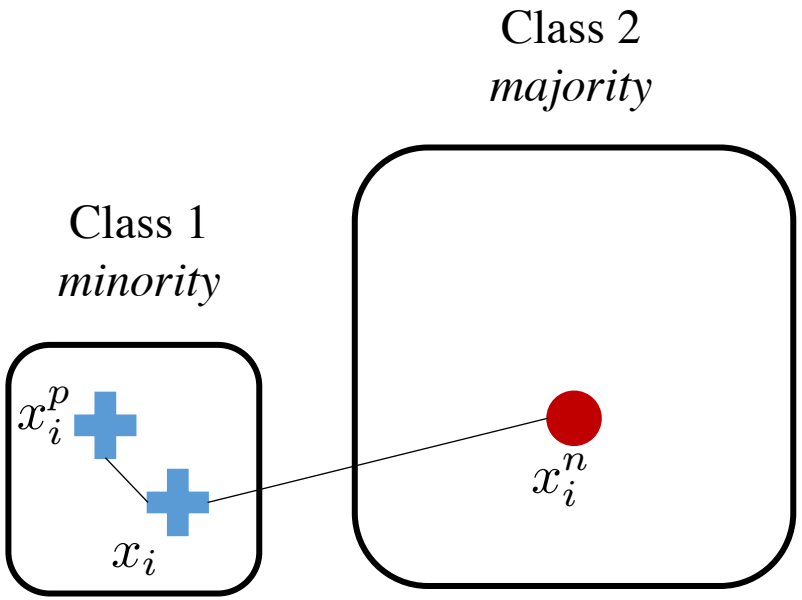
Minority class: very few instances with high degree of visual variability



The genuine neighborhood of these instances is easy to be invaded by other imposter nearest neighbors

Triplet loss helps to a certain extent

- Class-level constraint
  - x_i – an anchor
  - x_i^p – a positive instance (of the same class)
  - x_i^n – a negative instance (different class)

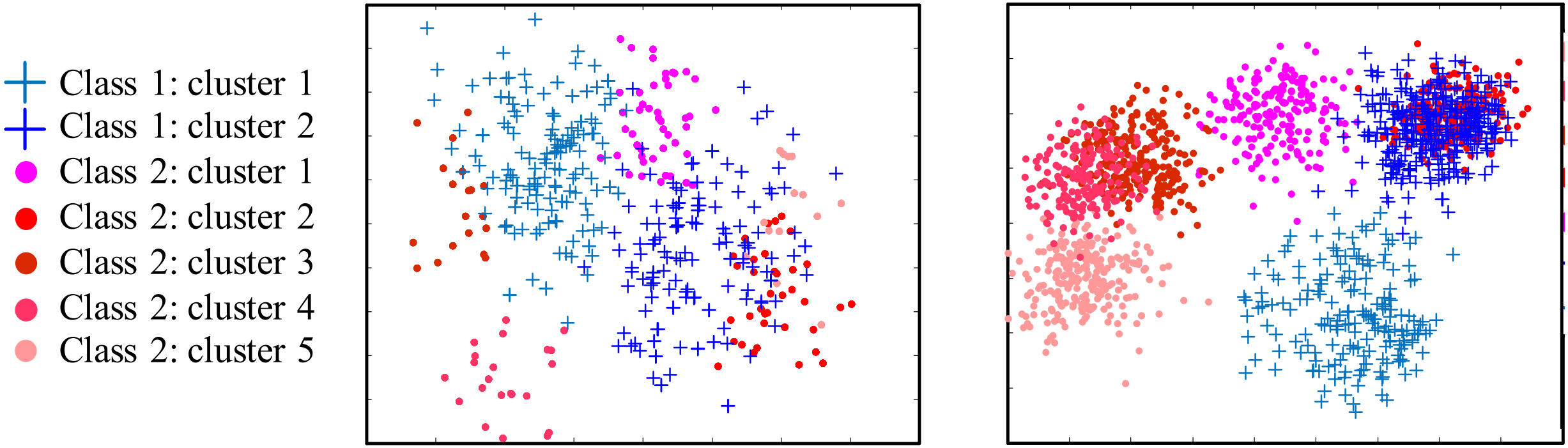


Wearing hat

Not wearing hat

Triplet loss helps to a certain extent

2D feature embedding of one imbalanced binary face attribute



Features extracted from DeepID2 model

Triplet embedding

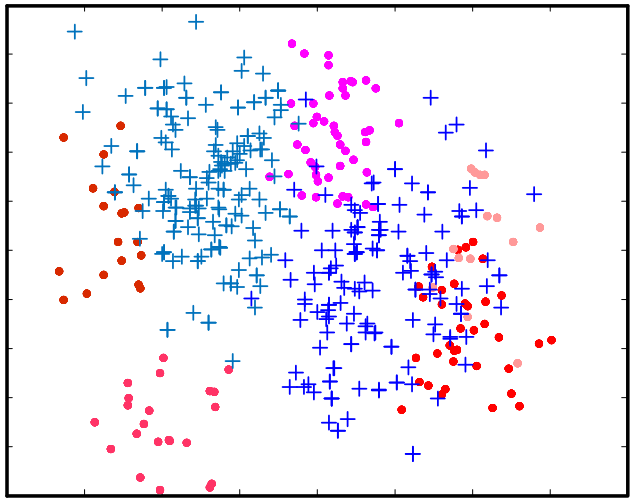
Contributions

- Learning deep feature embedding for imbalanced data classification
- A new method that preserves locality across clusters and discrimination between classes
- Large margin classification via fast cluster-wise kNN search

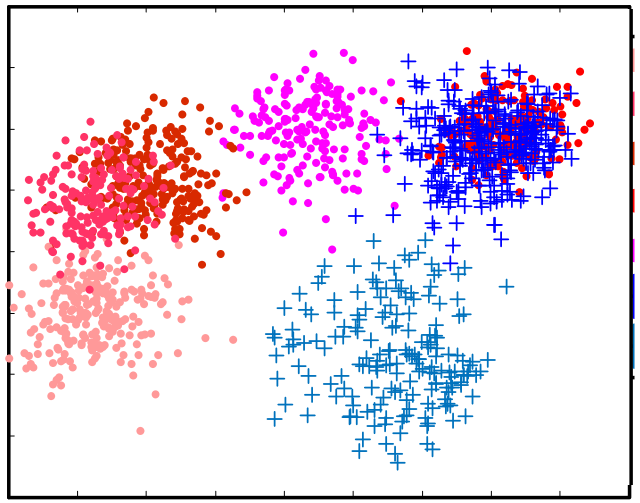
Our solution compared to triplet loss

2D feature embedding of one imbalanced binary face attribute

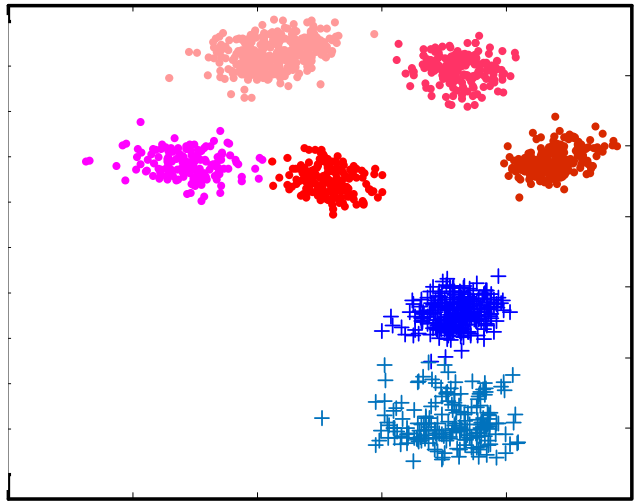
Class 1: cluster 1
Class 1: cluster 2
Class 2: cluster 1
Class 2: cluster 2
Class 2: cluster 3
Class 2: cluster 4
Class 2: cluster 5



Features extracted from DeepID2 model



Triplet embedding



Our solution

Large Margin Local Embedding

- Our goal:

  Learn a Euclidean embedding $f(x)$ from an image x into a feature space $\mathbb{R}^d$, such that the embedded features are discriminative with minimal possible local class imbalance.

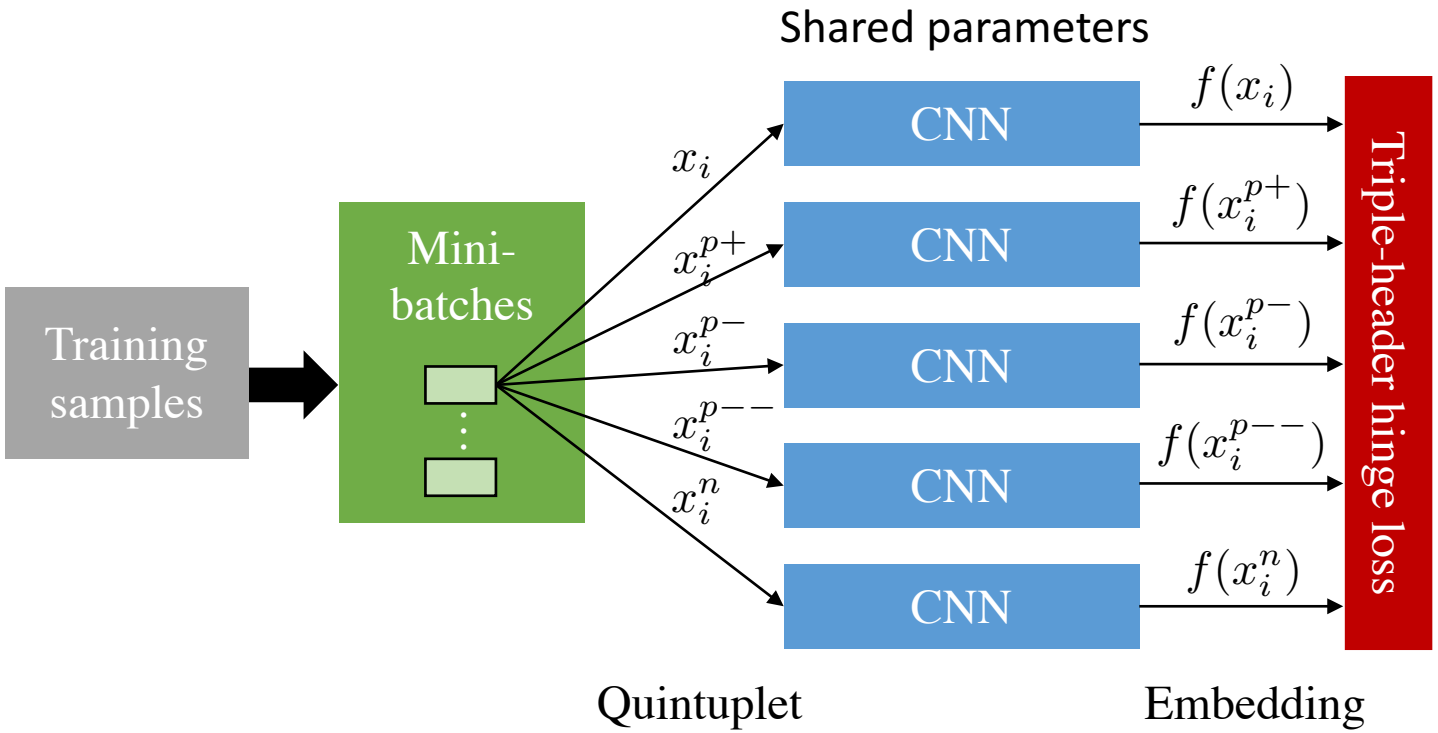
- Main idea:

  1. Find patterns (clusters) in each class

  2. Draw classification boundary locally only between marginal clusters, so not depends on class size

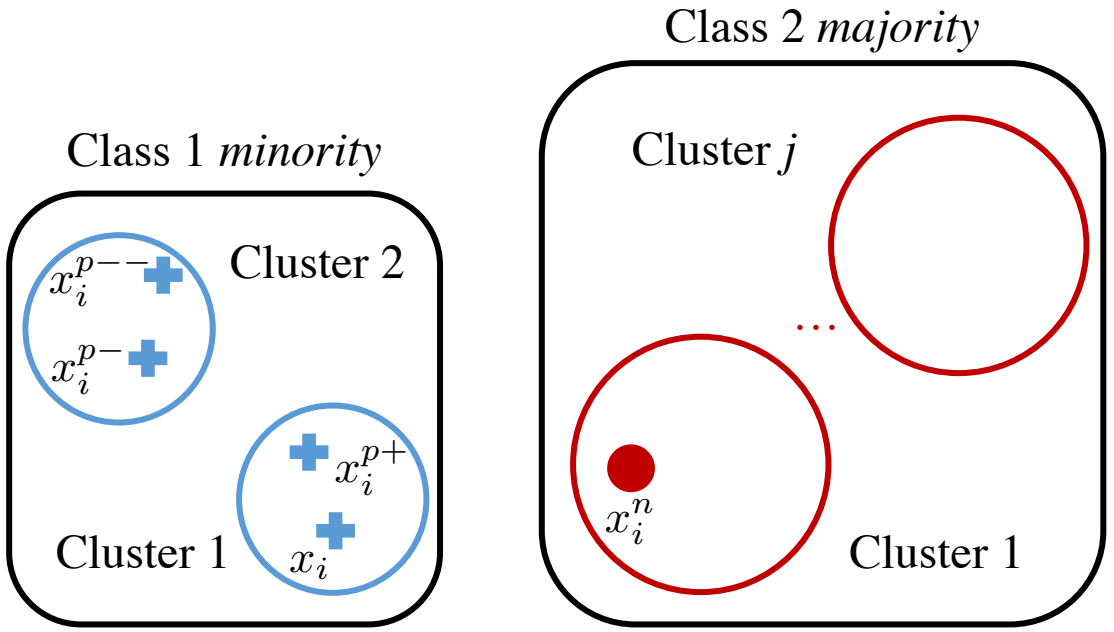
  3. Learn deep features to reduce class imbalance in any local neighborhood

Large Margin Local Embedding



Quintuplet sampling

- Cluster- and class-level
  - x_i – an anchor
  - x_i^{p+} – the anchor's most distant within-cluster neighbor
  - x_i^{p-} – the nearest within-class neighbor of the anchor, but from a different cluster
  - x_i^{p--} – the most distant within-class neighbor of the anchor
  - x_i^{n} – the nearest between-class neighbor of the anchor



Quintuplet sampling

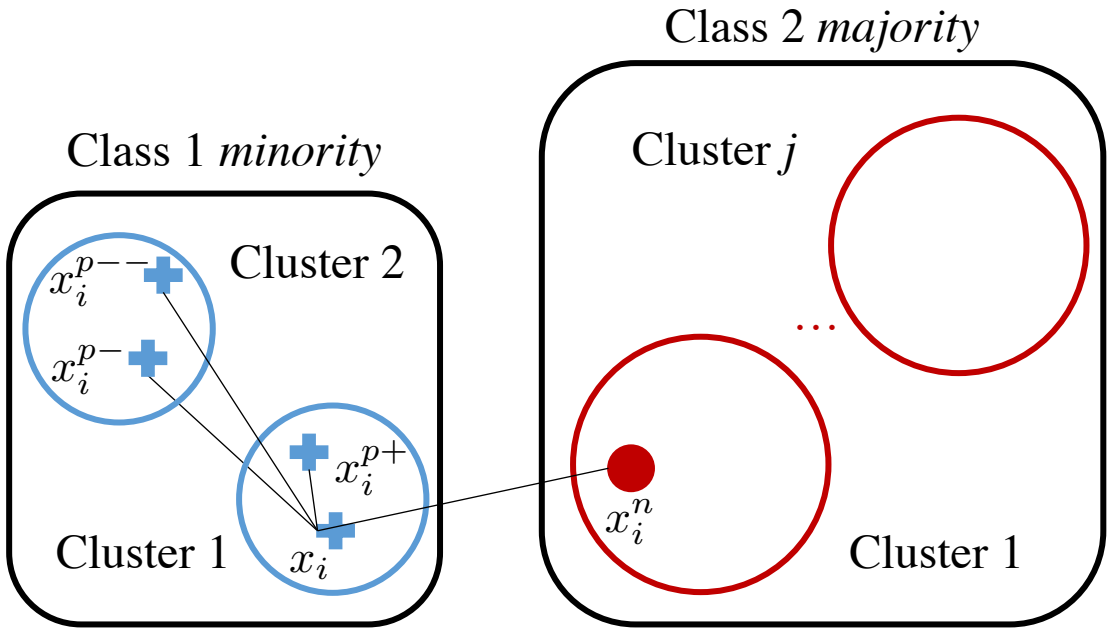
- Ensure the following relationship

$$D(f(x_i), f(x_i^n)) \quad >$$

$$D(f(x_i), f(x_i^{p--})) >$$

$$D(f(x_i), f(x_i^{p-})) \quad >$$

$$D(f(x_i), f(x_i^{p+}))$$



$D(f(x_i), f(x_j)) = \|f(x_i) - f(x_j)\|_2^2$ is the Euclidean distance

Advantages

- Richer information and a stronger constraint than the conventional class-level image similarity
- No information loss unlike under-sampling
- No artificial noise unlike over-sampling

How to obtain the clusters?

- Obtain the initial clusters for each class by applying *k*-means on some prior features
- Face attribute recognition, we use pre-trained DeepID2 features
- Alternating scheme
  - Refine the clusters using features extracted from the proposed model itself every n iterations

Triple-header hinge loss

- To constrain three margins between the four distances

$$\min \sum_i (\varepsilon_i + \tau_i + \sigma_i) + \lambda \|\mathbf{W}\|_2^2$$

s.t.:

$$\max\left(0, g_1 + D(f(x_i), f(x_i^{p+})) - D(f(x_i), f(x_i^{p-}))\right) \leq \varepsilon_i$$

$$\max\left(0, g_2 + D(f(x_i), f(x_i^{p-})) - D(f(x_i), f(x_i^{p--}))\right) \leq \tau_i$$

$$\max\left(0, g_3 + D(f(x_i), f(x_i^{p--})) - D(f(x_i), f(x_i^{n}))\right) \leq \sigma_i$$

$$\forall i,\ \varepsilon_i \geq 0,\ \tau_i \geq 0,\ \sigma_i \geq 0$$

Triple-header hinge loss

- To constrain three margins between the four distances

$$D(f(x_i), f(x_i^n)) \quad >$$

$$D(f(x_i), f(x_i^{p--})) \quad >$$

$$D(f(x_i), f(x_i^{p-})) \quad >$$

$$D(f(x_i), f(x_i^{p+}))$$

$$\min \sum_i (\varepsilon_i + \tau_i + \sigma_i) + \lambda \|\mathbf{W}\|_2^2$$

s.t.:

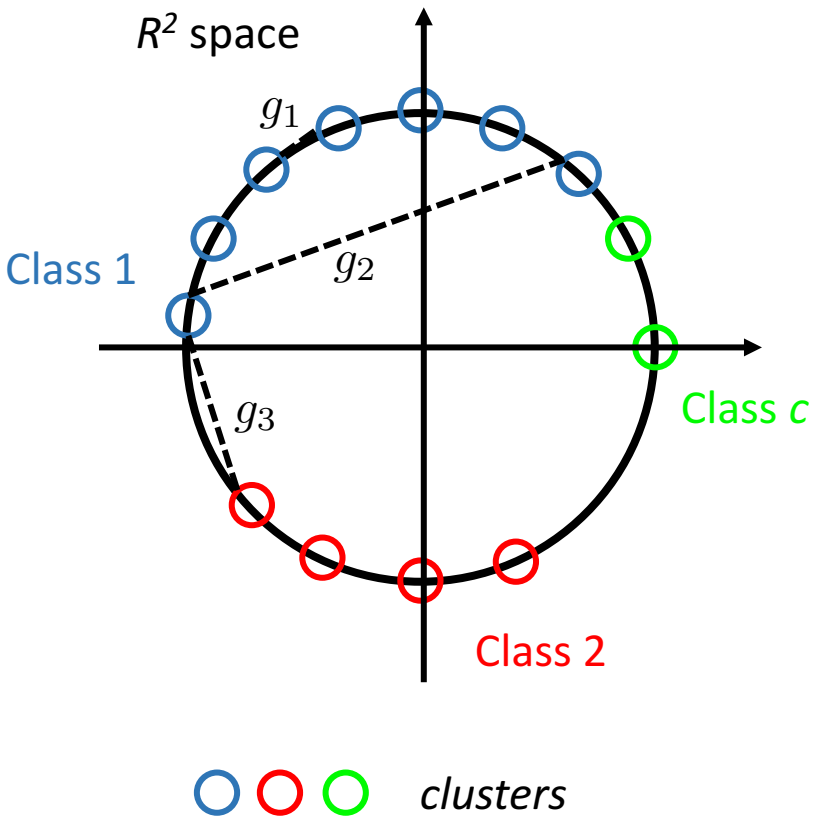
$$\max\left(0, g_1 + D(f(x_i), f(x_i^{p+})) - D(f(x_i), f(x_i^{p-}))\right) \leq \varepsilon_i$$

$$\max\left(0, g_2 + D(f(x_i), f(x_i^{p-})) - D(f(x_i), f(x_i^{p--}))\right) \leq \tau_i$$

$$\max\left(0, g_3 + D(f(x_i), f(x_i^{p--})) - D(f(x_i), f(x_i^{n}))\right) \leq \sigma_i$$

$$\forall i,\ \varepsilon_i \geq 0,\ \tau_i \geq 0,\ \sigma_i \geq 0$$

Triple-header hinge loss



$$\min \sum_i (\varepsilon_i + \tau_i + \sigma_i) + \lambda \|\mathbf{W}\|_2^2$$

s.t.:

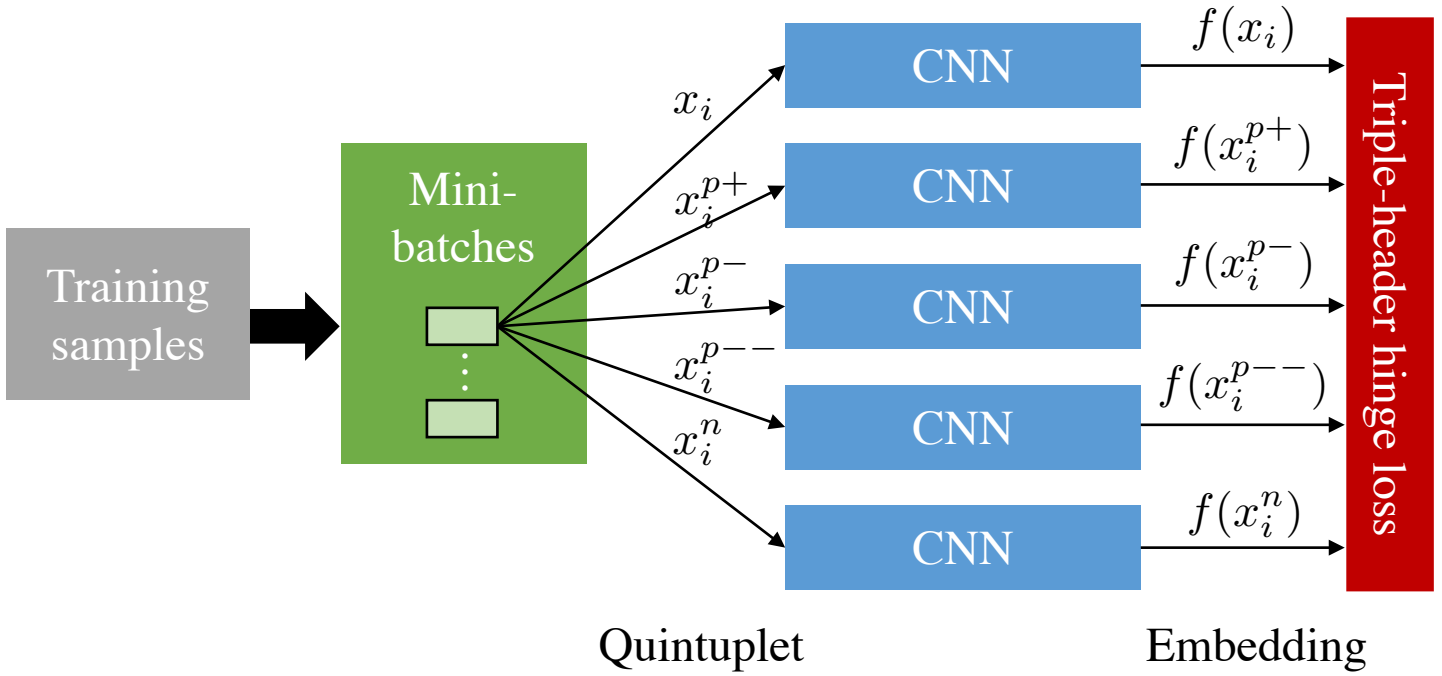
$$\max\left(0, g_1 + D(f(x_i), f(x_i^{p+})) - D(f(x_i), f(x_i^{p-}))\right) \leq \varepsilon_i$$

$$\max\left(0, g_2 + D(f(x_i), f(x_i^{p-})) - D(f(x_i), f(x_i^{p--}))\right) \leq \tau_i$$

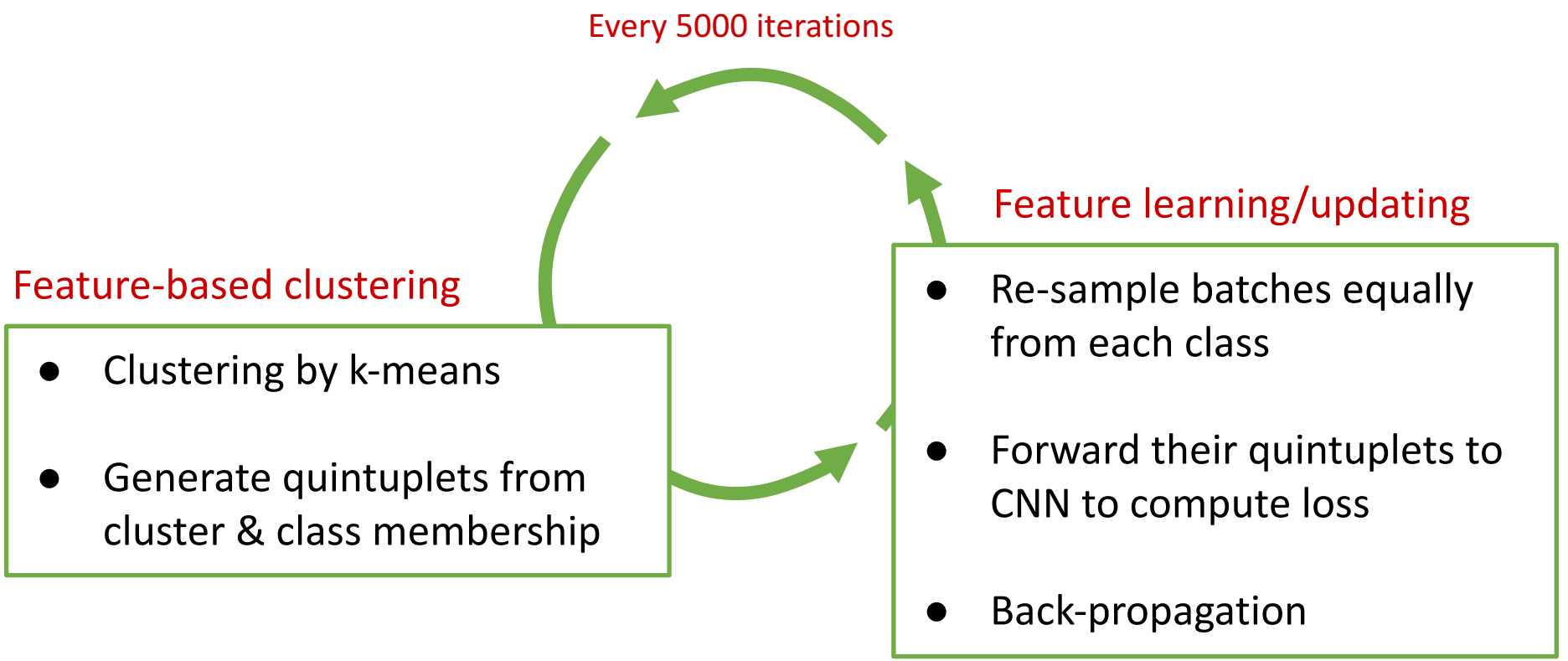
$$\max\left(0, g_3 + D(f(x_i), f(x_i^{p--})) - D(f(x_i), f(x_i^{n}))\right) \leq \sigma_i$$

$$\forall i,\ \varepsilon_i \geq 0,\ \tau_i \geq 0,\ \sigma_i \geq 0$$

Network architecture (learning)



Summary of steps



Why is it effective?

- Triplet loss
  - The similarity information is only extracted at the *class-level*
  - Homogeneously collapse each class irrespective of their different degrees of variation
  - When a class has high data variability, it is also hard to maintain the class-wise margin

- Triple-header hinge loss
  - Generates diverse quintuplets that differ in the membership of *both clusters and classes*
  - Captures the considerable data variability within each class
  - Can easily enforce the local margin

Nearest neighbor imbalanced classification

- We modified kNN in two ways:
  1. In the well-clustered embedding space LMLE, we treat each cluster as a class-specific exemplar, and perform a fast cluster-wise kNN search.
  2. Use a large margin decision

     Let $\phi(q)$ be query q's local neighborhood defined by its kNN cluster centroids $\{m_i\}_{i-1}^{k}$

$$y_q = \arg\max_{c=1,\ldots,C} \left(\min_{\substack{m_j \in \phi(q) \\ y_j \neq c}} D(f(q), f(m_j)) - \max_{\substack{m_i \in \phi(q) \\ y_i = c}} D(f(q), f(m_i)) \right)$$

CelebA dataset (100k train,10k test)

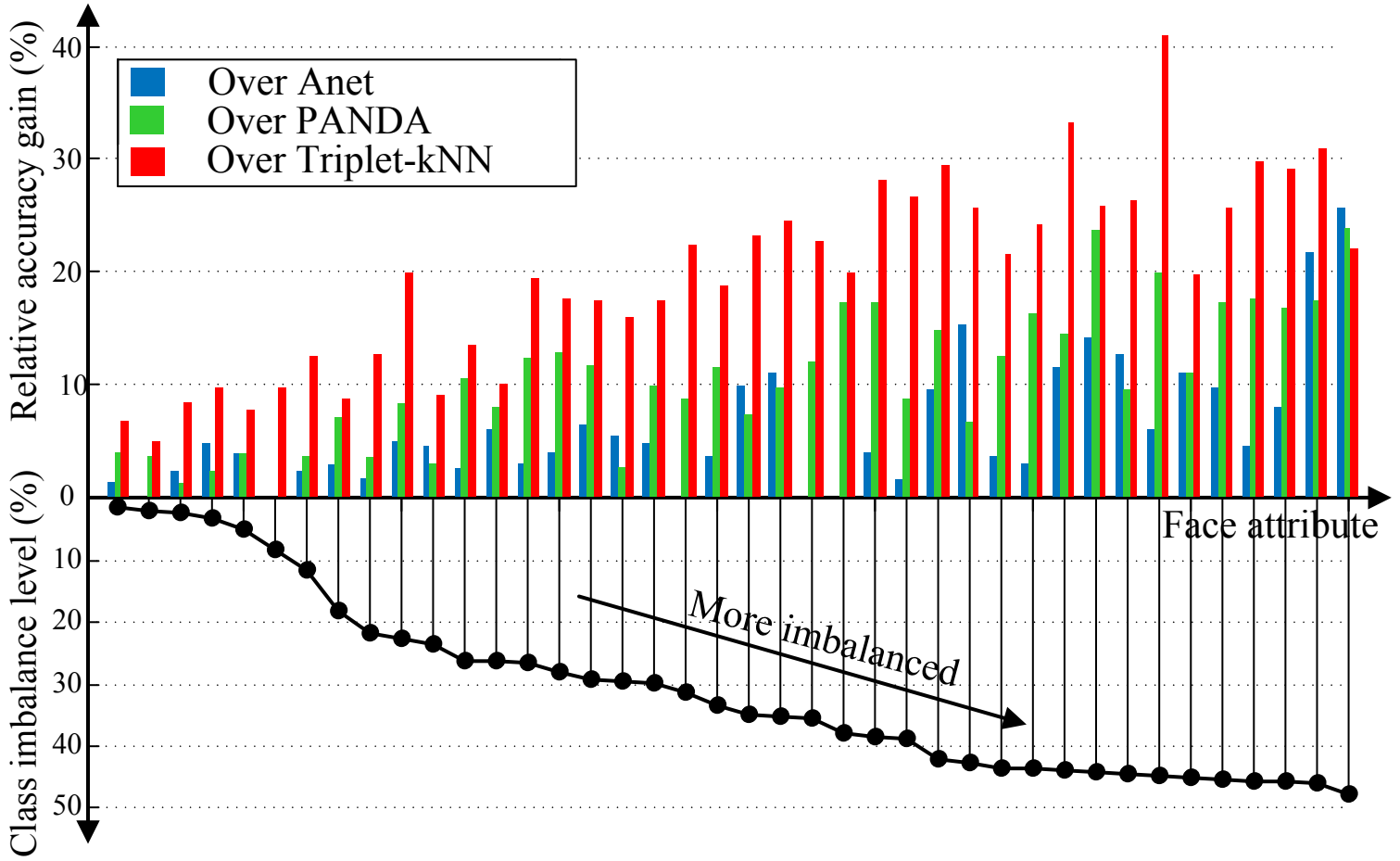
	Attractive	Mouth Open	Smiling	Wear Lipstick	High Cheekbones	Male	Heavy Makeup	Wavy Hair	Oval Face	Pointy Nose	Arched Eyebrows	Black Hair	Big Lips	Big Nose	Young	Straight Hair	Brown Hair	Bags Under Eyes	Wear Earrings	No Beard	Bangs
Imbalance level	1	2	2	3	5	8	11	18	22	22	23	26	26	27	28	29	30	30	31	33	35
Triplet-kNN [34]	83	92	92	91	86	91	88	77	61	61	73	82	55	68	75	63	76	63	69	82	81
PANDA [47]	85	93	98	97	89	99	95	78	66	67	77	84	56	72	78	66	85	67	77	87	92
ANet [29]	87	96	97	95	89	99	96	81	67	69	76	90	57	78	84	69	83	70	83	93	90
LMLE-kNN	88	96	99	99	92	99	98	83	68	72	79	92	60	80	87	73	87	73	83	96	98
	Blond Hair	Bushy Eyebrows	Wear Necklace	Narrow Eyes	5 o'clock Shadow	Receding Hairline	Wear Necktie	Eyeglasses	Rosy Cheeks	Goatee	Chubby	Sideburns	Blurry	Wear Hat	Double Chin	Pale Skin	Gray Hair	Mustache	Bald		**Average**
Imbalance level	35	36	38	38	39	42	43	44	44	44	44	44	45	45	45	46	46	46	48		
Triplet-kNN [34]	81	68	50	47	66	60	73	82	64	73	64	71	43	84	60	63	72	57	75		**72**
PANDA [47]	91	74	51	51	76	67	85	88	68	84	65	81	50	90	64	69	79	63	74		**77**
ANet [29]	90	82	59	57	81	70	79	95	76	86	70	79	56	90	68	77	85	61	73		**80**
LMLE-kNN	99	82	59	59	82	76	90	98	78	95	79	88	59	99	74	80	91	73	90		**84**

Anet
classification accuracy = 87.24%,
balance accuracy = 80.02%

Ours
classification accuracy = **90.35**%,
balance accuracy = **84.25**%

Class imbalance level (= |positive class rate-50|%)

CelebA dataset (100k train,10k test)



- Code available
- http://mmlab.ie.cuhk.edu.hk/projects/LMLE.html

Thanks