Face Detection & Recognition

이영기
서울대학교 컴퓨터공학부

서 울 대 학 교
SEOUL NATIONAL UNIVERSITY

References

- Many papers and blogs in the Internet…
- Our research experience

Content Overview

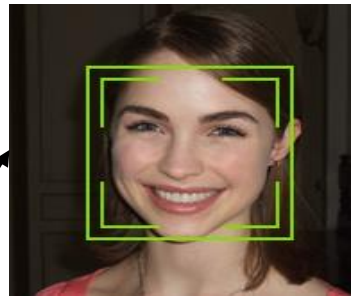
- Task overview
- Face detection
  - ✓MTCNN
  - ✓TinyFace
  - ✓SSH & PyramidBox
  - ✓RetinaFace
- Face recognition
  - ✓Metric learning
  - ✓Angular classification (SphereFace/CosFace/ArcFace)
  - ✓Target-specific face recognition

Content Overview

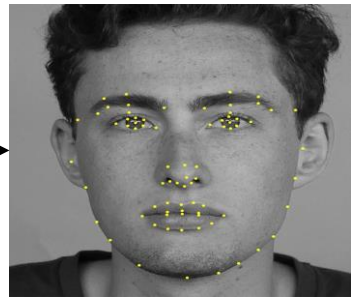
- **Task overview**
- Face detection
  - ✓MTCNN
  - ✓TinyFace
  - ✓SSH & PyramidBox
  - ✓RetinaFace
- Face recognition
  - ✓Metric learning
  - ✓Angular classification (SphereFace/CosFace/ArcFace)
  - ✓Target-specific face recognition

Task Overview

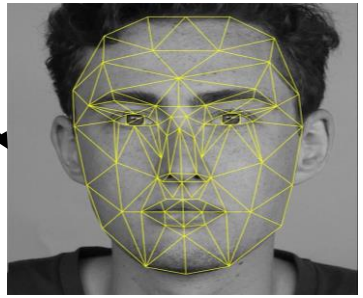
- Face detection



Bounding box



Facial landmarks



Face meshes

...

Task Overview

- Face recognition
  - ✓Identification

Who is this?
a) John
b) Alice
c) Jane
d) Bob
…

Evaluation metric: Top-K accuracy

  - ✓Verification

Does the two faces match?

vs.

vs.

Evaluation metric: ROC curve

Task Overview

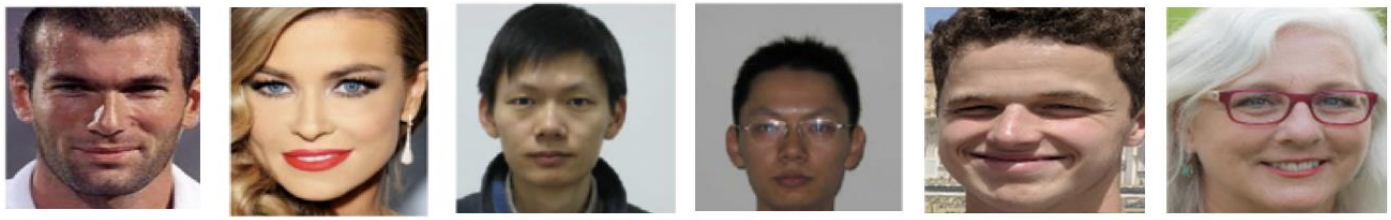
- Why is face detection difficult?

S. Yang et al., “WIDER FACE: A Face Detection Benchmark,” CVPR 2016.

Task Overview

- Why is face recognition difficult?

"Ideal" face recognition



"Unconstrained" face recognition

Pose

Occlusion

Illumination

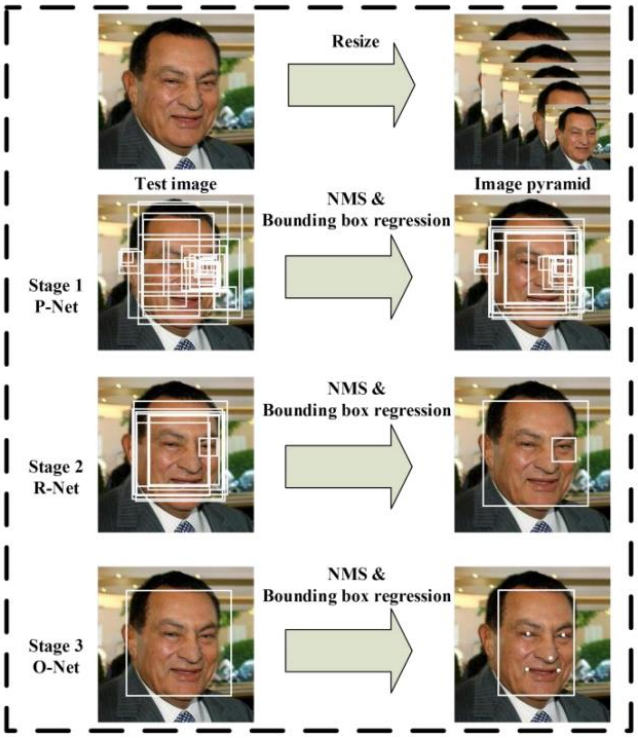
Resolution

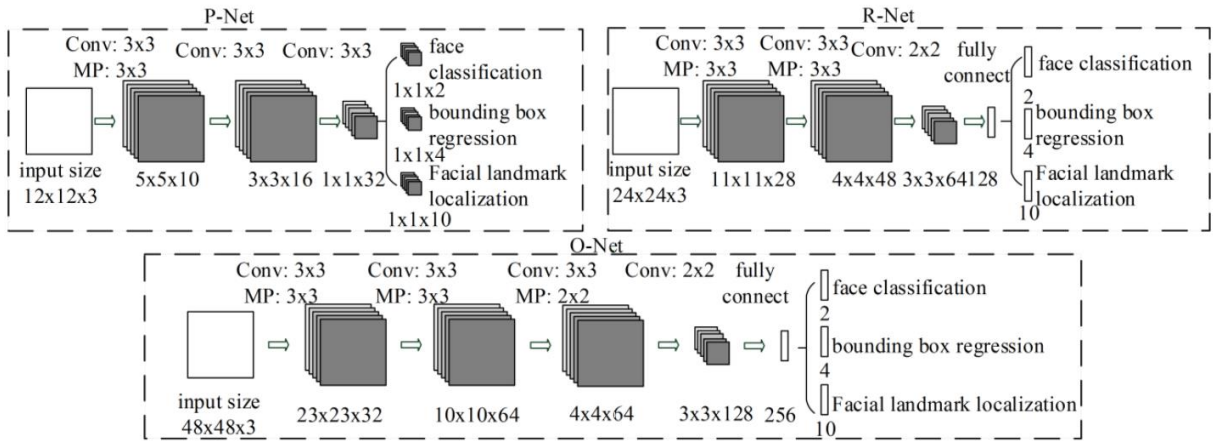
Content Overview

- Task overview
- **Face detection**
  - ✓MTCNN
  - ✓TinyFace
  - ✓SSH & PyramidBox
  - ✓RetinaFace
- Face recognition
  - ✓Metric learning
  - ✓Angular classification (SphereFace/CosFace/ArcFace)
  - ✓Target-specific face recognition

MTCNN

- Output: bounding box + 5 facial landmarks
- De-facto standard for face cropping

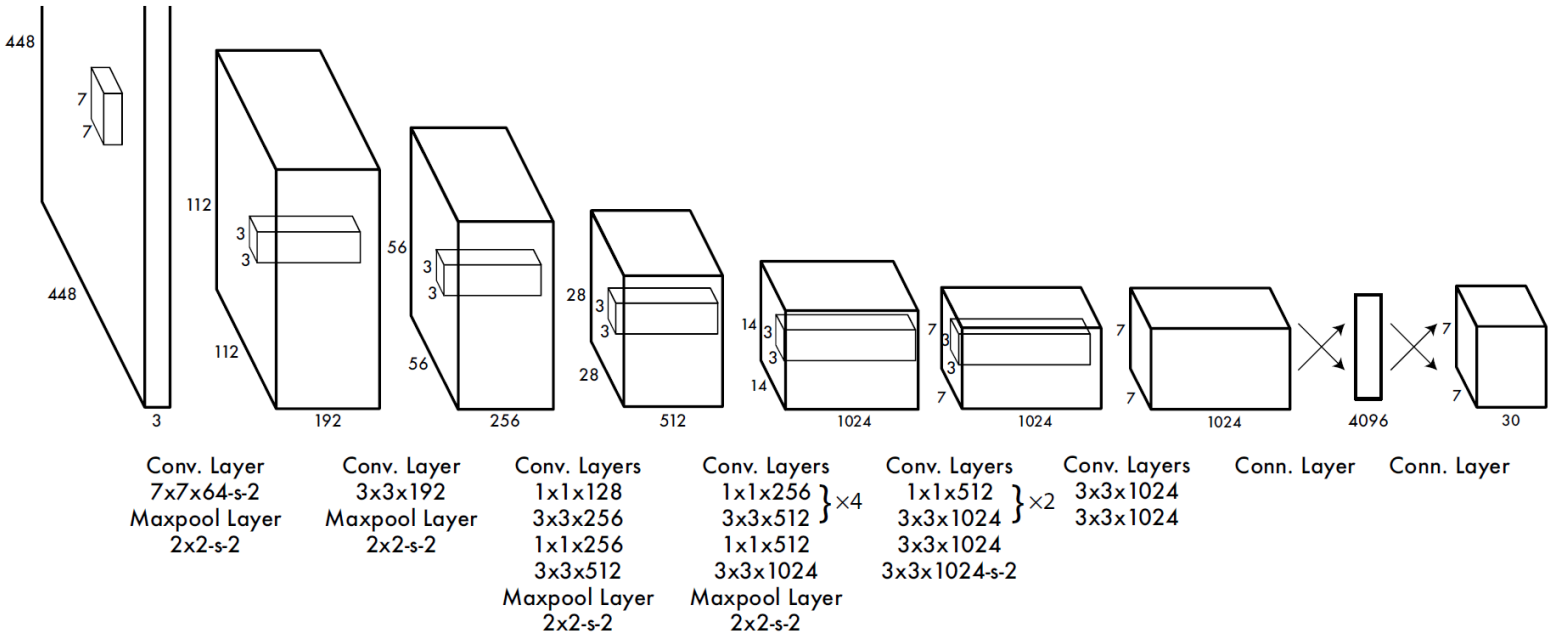




[1] K. Zhang et al., "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks," IEEE Signal Processing Letters 2016.

Recall from Last Session…

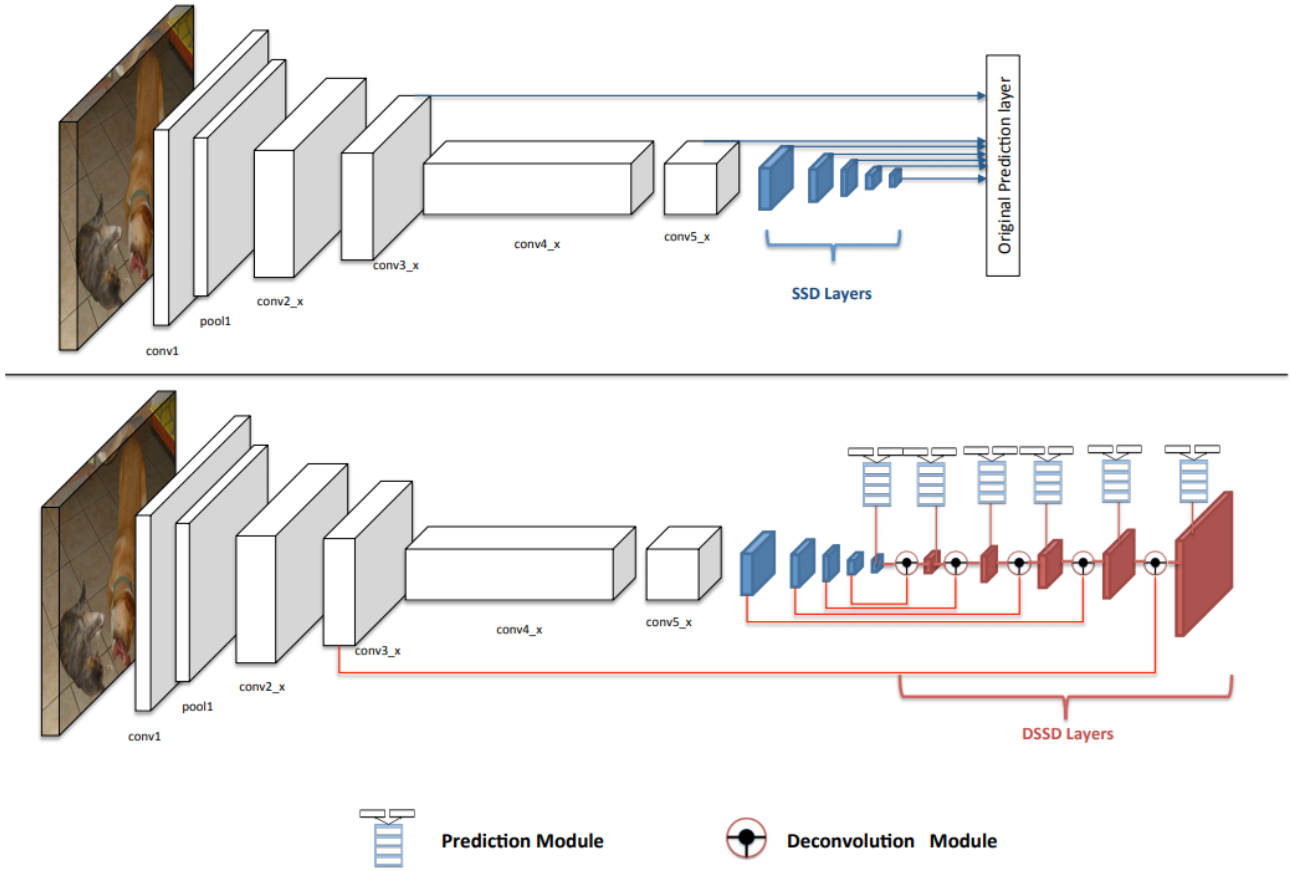
- Conventional CNNs “summarize” image through layers
- There are no features left for small objects!



J. Redmon et al. “YOLOv3: An Incremental Improvement,” arXiv 2018.

Recall from Last Session…

- Deconvolutional SSD (DSSD)

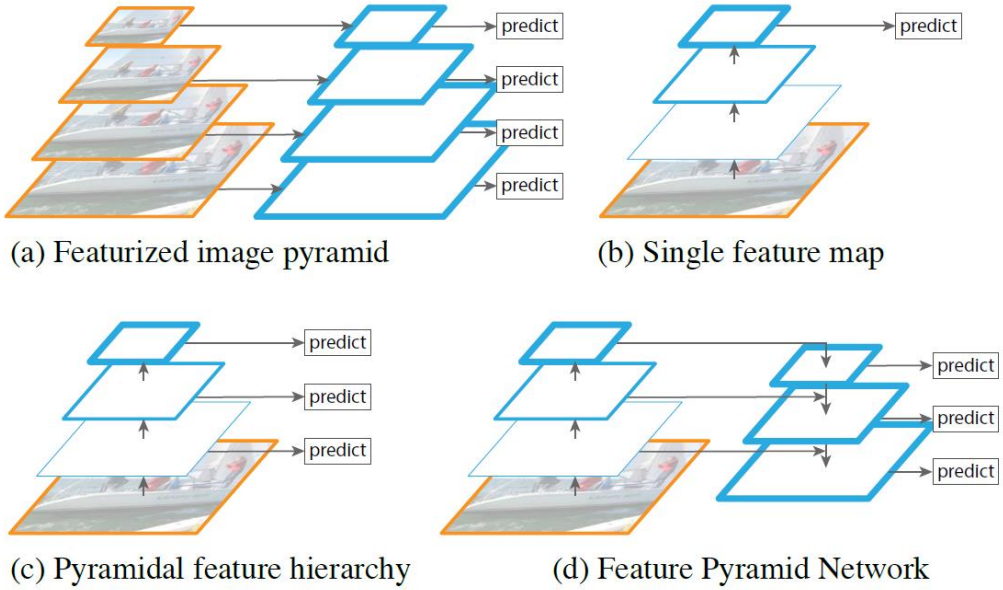


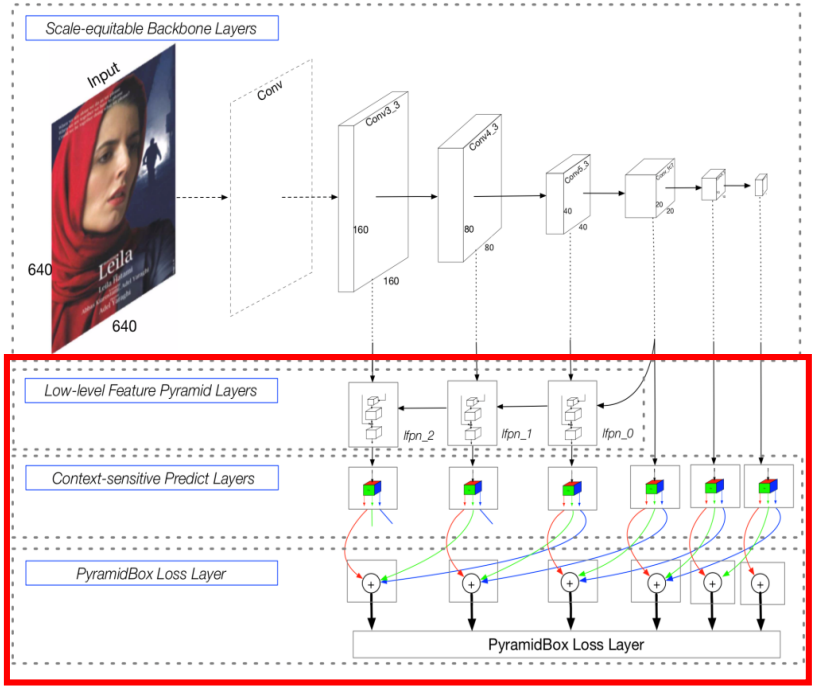
C. Fu et al. "DSSD : Deconvolutional Single Shot Detector," arXiv 2017.

Trend: Small Object Detection

- Feature pyramid network

  → De-facto standard for SOTA face detectors





T. Lin et al. "Feature Pyramid Networks for Object Detection," CVPR 2017.
X. Tang et al. "PyramidBox: A Context-assisted Single Shot Face Detector." ECCV 2018.

WIDER Face Dataset - Leaderboard

CVPR ’19 [7]

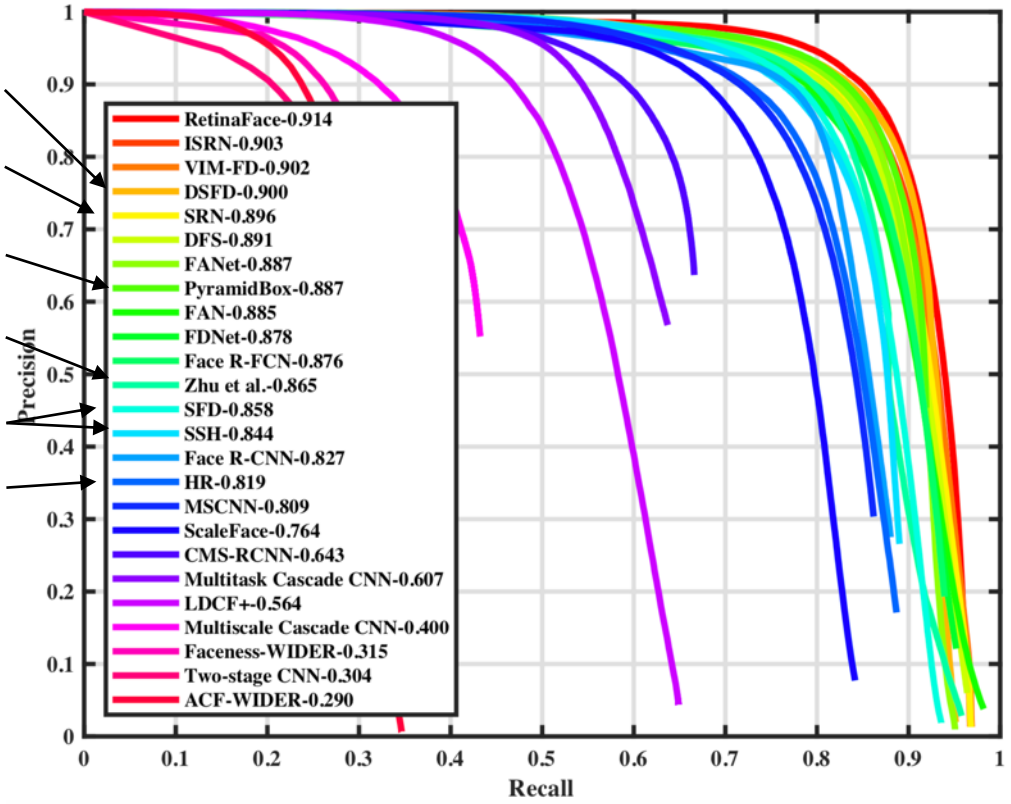
AAAI ’19 [6]

ECCV ’18 [5]

CVPR ’18 [4]

ICCV ’17 [2,3]

CVPR ’17 [1]



[1] P. Hu et al., “Finding Tiny Faces,” CVPR 2017.
[2] M. Hajibi et al., “SSH: Single Stage Headless Face Detector,” ICCV 2017.
[3] S. Zhang et al., “S3FD: Single Shot Scale-invariant Face Detector,” ICCV 2017.
[4] C. Zhu et al., “Seeing Small Faces from Robust Anchor’s Perspective,” CVPR 2018.
[5] X. Tang et al., ”PyramidBox: A Context-assisted Single Shot Face Detector,” ECCV 2018.
[6] C. Chi et al., “Selective Refinement Network for High Performance Face Detection,” AAAI 2019.
[7] J. Li et al., “DSFD: Dual Shot Face Detector,” CVPR 2019.

TinyFace

- Motivation
  - ✓ Existing networks: scale-invariant (trained/tested on a single resolution)
    → Can only detect objects with sizes observed in training dataset
  - ✓ Cues for recognizing a 3px tall face are fundamentally different than those for recognizing a 300px tall face.

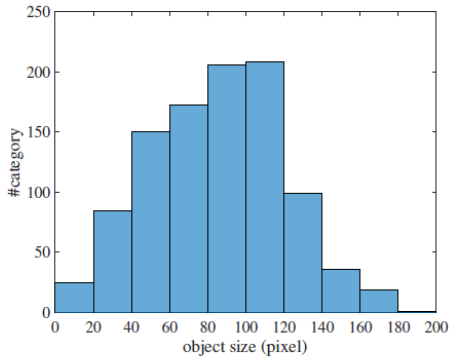
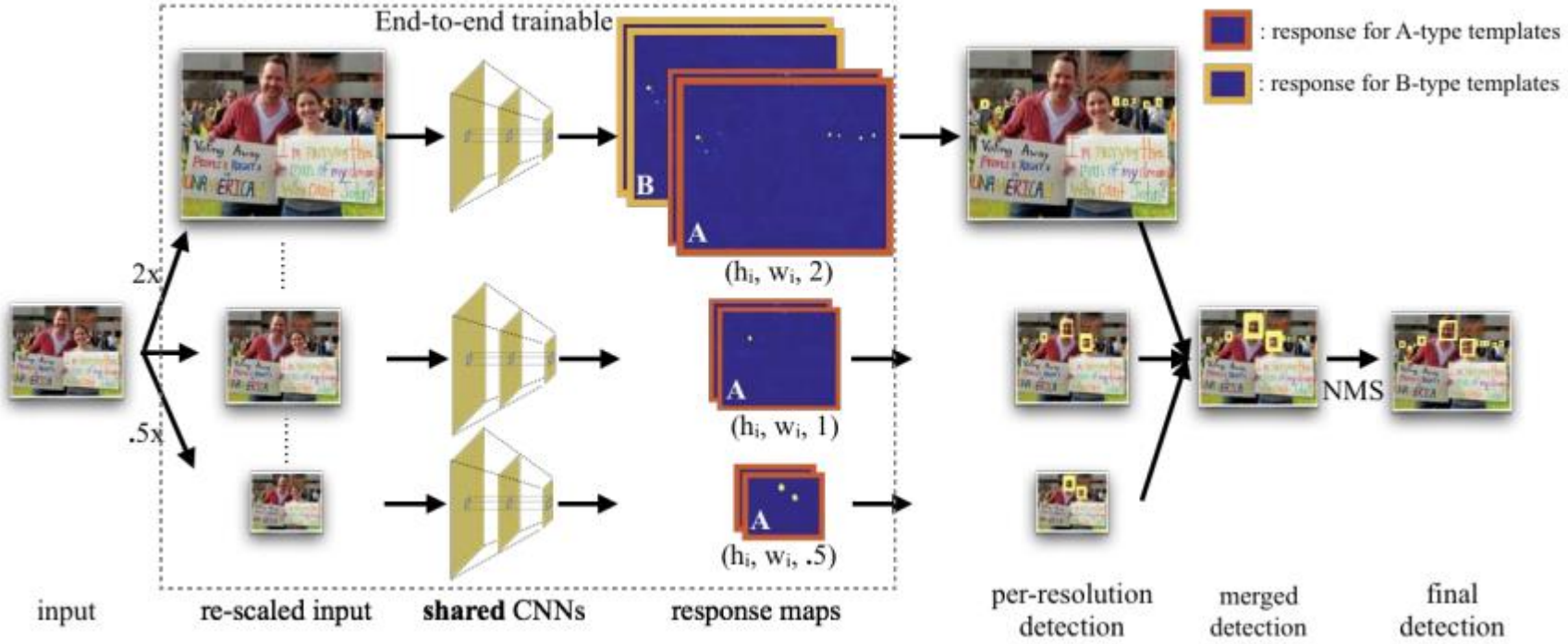


Figure 6: The distribution of average object scales in the ImageNet dataset (assuming images are normalized to 224x224). More than 80% categories have an average object size between 40 and 140 pixel. We hypothesize that models pre-trained on ImageNet are optimized for objects in that range.

P. Hu et al., "Finding Tiny Faces," CVPR 2017.

TinyFace

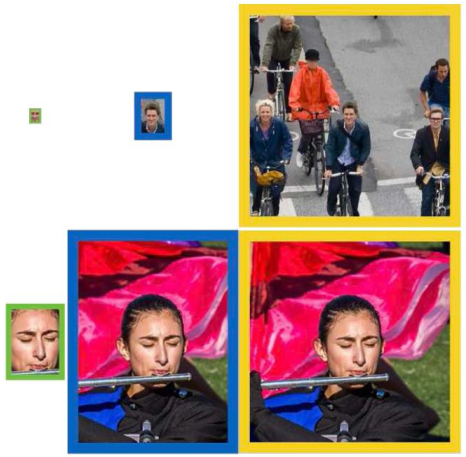
- Approach
  - ✓Train multiple scale-specific detectors

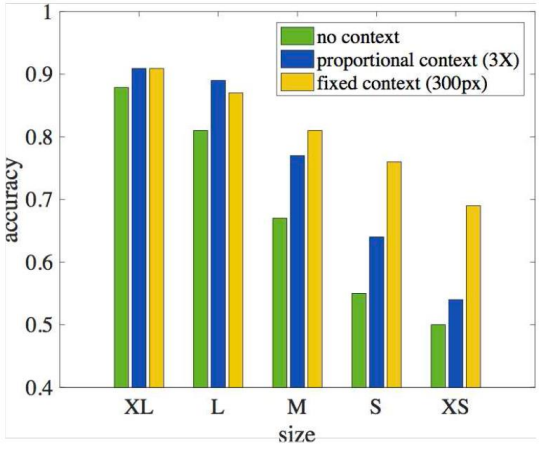


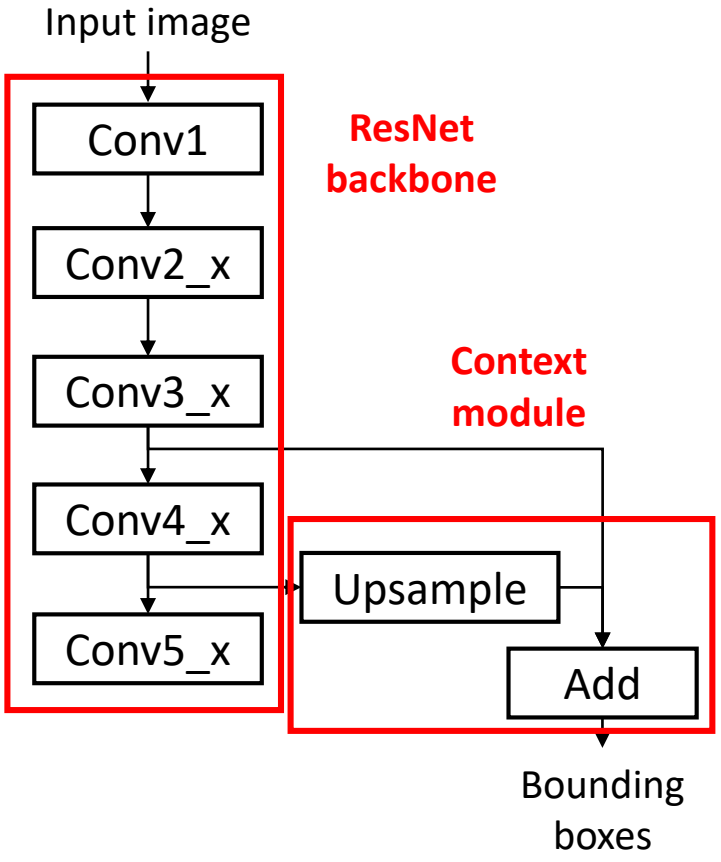
P. Hu et al., "Finding Tiny Faces," CVPR 2017.

TinyFace

- Approach
  - ✓Employ context module to enhance small face detection accuracy



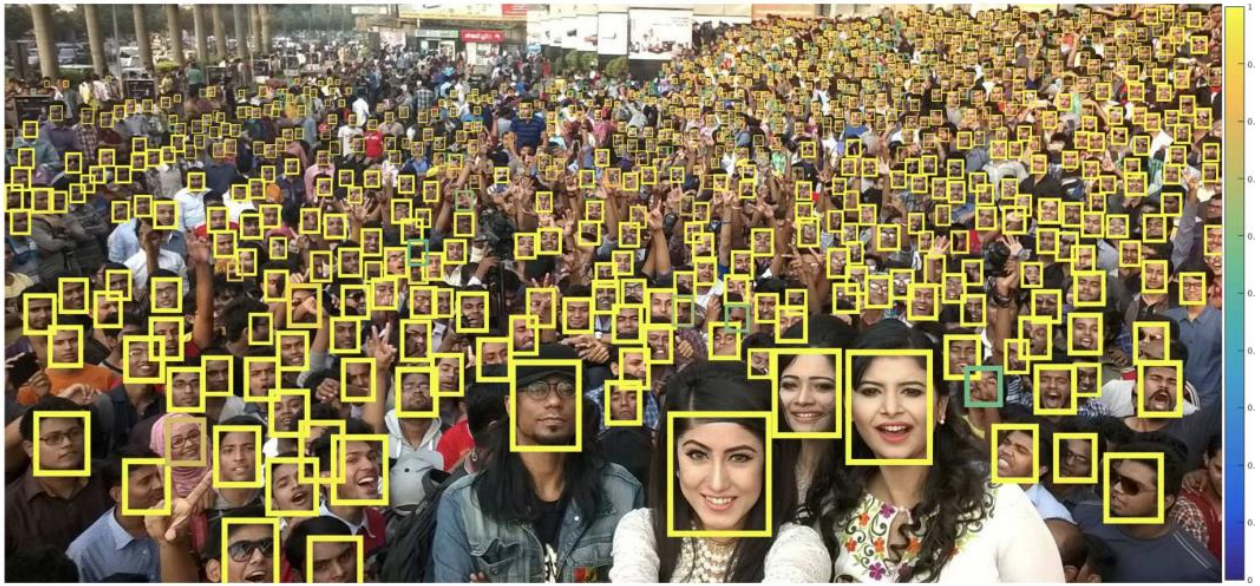




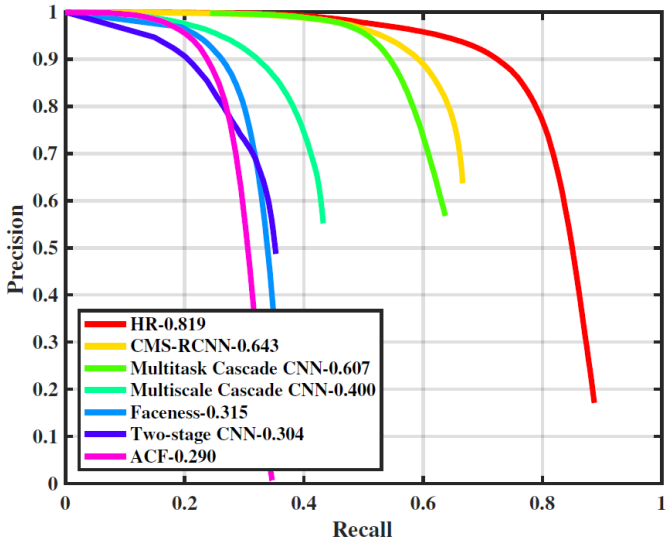
P. Hu et al., "Finding Tiny Faces," CVPR 2017.

TinyFace

- Results



Method	Easy	Medium	Hard
ACF[23]	0.659	0.541	0.273
Two-stage CNN[25]	0.681	0.618	0.323
Multiscale Cascade CNN[24]	0.691	0.634	0.345
Faceness[24]	0.713	0.664	0.424
Multitask Cascade CNN[26]	0.848	0.825	0.598
CMS-RCNN[27]	0.899	0.874	0.624
HR-VGG16	0.862	0.844	0.749
HR-ResNet50	0.907	0.890	0.802
HR-ResNet101	**0.919**	**0.908**	**0.823**



P. Hu et al., "Finding Tiny Faces," CVPR 2017.

TinyFace

- Issue: latency
  - ✓We need at least 1,280x720 resolution for robust detection
  - ✓X2 upsampling is crucial for detecting small faces
    - → Need to process a 2,560x1,440 frame

360p **720p** **1080p**

P. Hu et al., “Finding Tiny Faces,” CVPR 2017.

TinyFace

- Issue: latency
  - ✓We need at least 1,280x720 resolution for robust detection
  - ✓X2 upsampling is crucial for detecting small faces
    - → Need to process a 2,560x1,440 frame

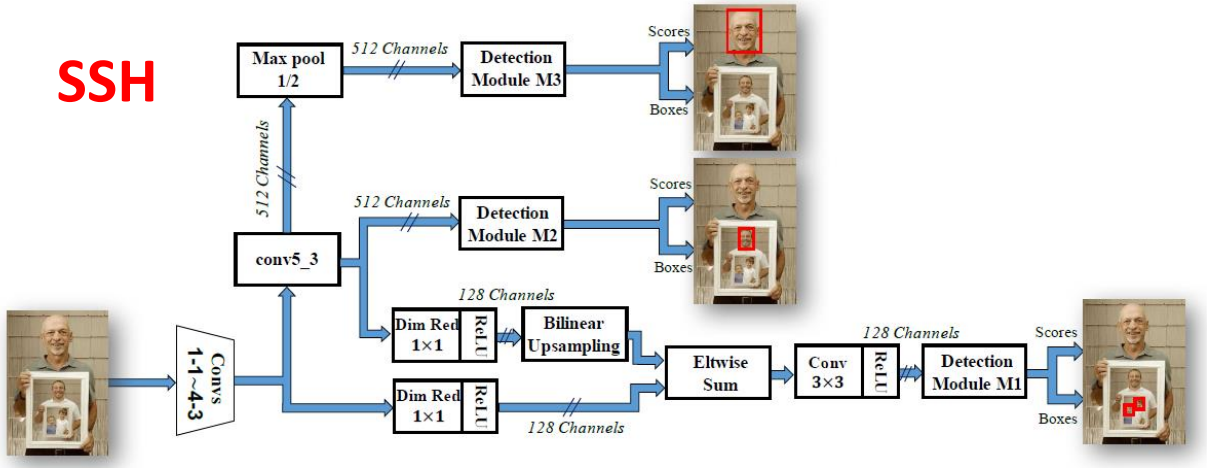
Architecture	Input resolution	GTX 2080 Ti (TensorFlow)	Google Pixel2 (TensorFlow-Lite)
ResNet-101	1440x2560	190 ms	15 s
	720x1280	57 ms	4 s
ResNet-50	1440x2560	-	6.4 s
	720x1280	-	1.6 s

P. Hu et al., "Finding Tiny Faces," CVPR 2017.

SSH & PyramidBox

- Acknowledging the latency issue of TinyFace, following work focused on designing a single shot detector
- Approaches
  - ✓ Detect faces at multiple layers (using feature pyramid)
  - ✓ Employ context module to enhance small face detection accuracy

SSH



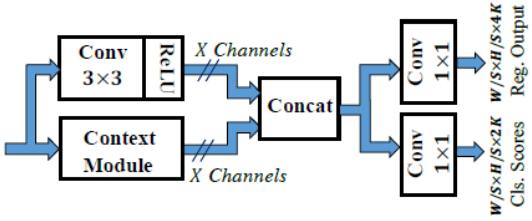


Figure 3: *SSH* detection module.

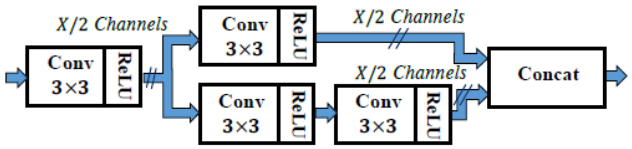


Figure 4: *SSH* context module.

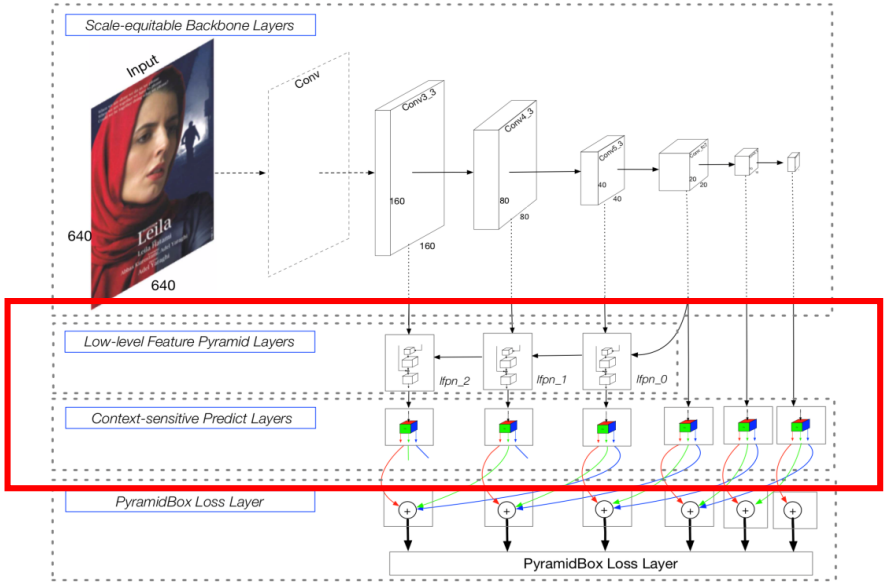
M. Najibi et al., "SSH: Single Stage Headless Face Detector," ICCV 2017.

X. Tang et al., "PyramidBox: A Context-assisted Single Shot Face Detector," ECCV 2018.

SSH & PyramidBox

- Acknowledging the latency issue of TinyFace, following work focused on designing a single shot detector
- Approaches
  - ✓ Detect faces at multiple layers (using feature pyramid)
  - ✓ Employ context module to enhance small face detection accuracy

PyramidBox



M. Najibi et al., “SSH: Single Stage Headless Face Detector,” ICCV 2017.
X. Tang et al., ”PyramidBox: A Context-assisted Single Shot Face Detector,” ECCV 2018.

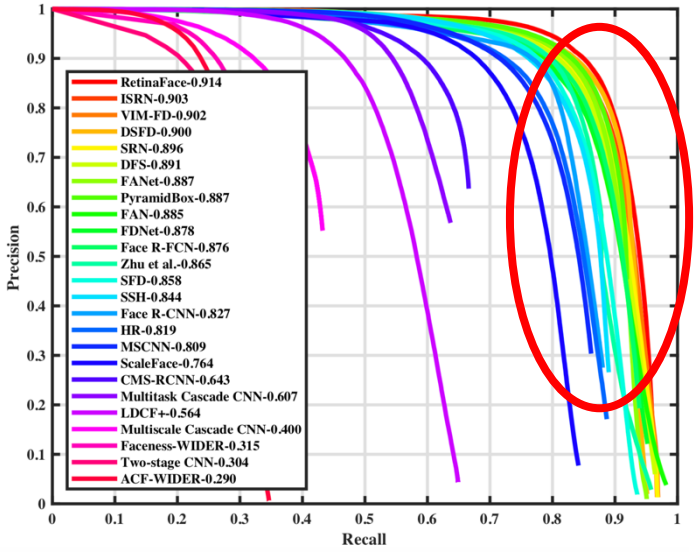
SSH & PyramidBox

- Result
  - ✓ Faster inference time

Table 2: *SSH* inference time with respect to different input sizes.

Max Size	400 × 800	600 × 1000	800 × 1200	1200 × 1600
Time	48 ms	74 ms	107 ms	182 ms

  - ✓ Higher accuracy (marginal gain)



M. Najibi et al., “SSH: Single Stage Headless Face Detector,” ICCV 2017.
X. Tang et al., ”PyramidBox: A Context-assisted Single Shot Face Detector,” ECCV 2018.

RetinaFace

- State-of-the-art as of early 2019
- Key motivation: learning to regress facial landmarks and mesh helps improve face detection accuracy
  - ✓ e.g., Mask R-CNN improves detection accuracy by pixel-level segmentation module

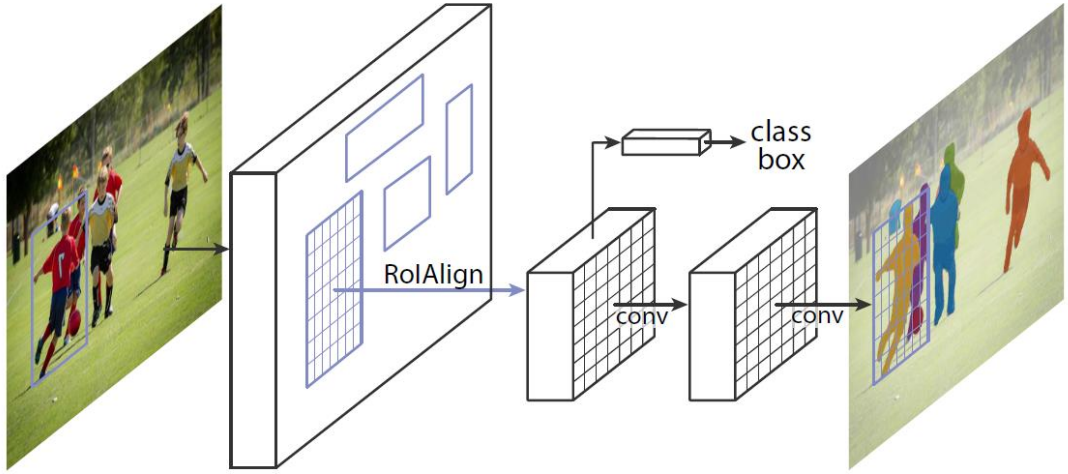
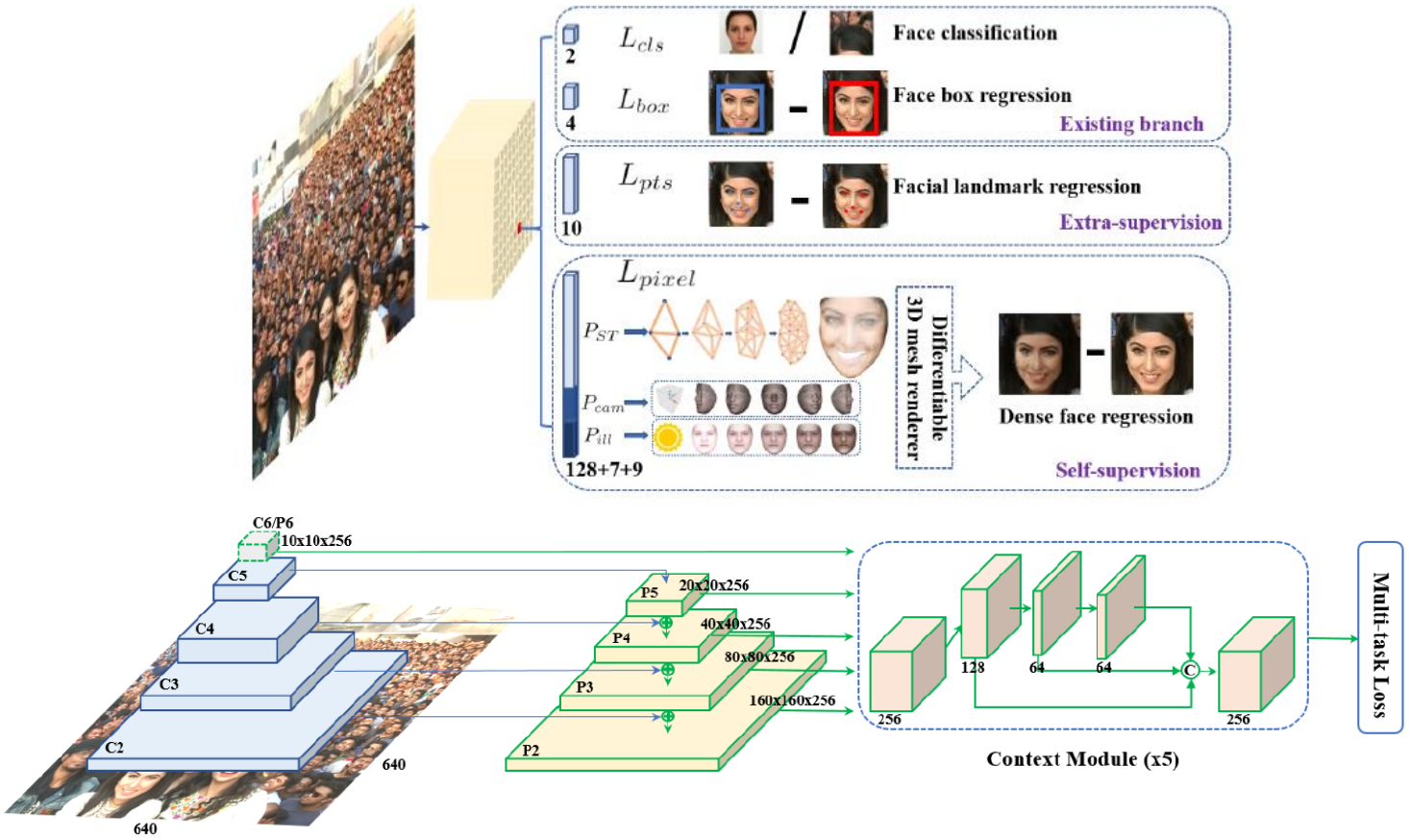


Figure 1. The **Mask R-CNN** framework for instance segmentation.

J. Deng et al., " RetinaFace: Single-stage Dense Face Localisation in the Wild," arXiv 2019.

RetinaFace

- Network architecture & training is similar to existing work
- Face mesh regression is not released in the official Git repo



J. Deng et al., ” RetinaFace: Single-stage Dense Face Localisation in the Wild,” arXiv 2019.

RetinaFace

- Important point to note: inference latency

RetinaFace

4.8. Inference Efficiency

During testing, RetinaFace performs face localisation in a single stage, which is flexible and efficient. Besides the above-explored heavy-weight model (ResNet-152, size of 262MB, and AP 91.8% on the WIDER FACE hard set), we also resort to a light-weight model (MobileNet-0.25 [22], size of 1MB, and AP 78.2% on the WIDER FACE hard set) to accelerate the inference.

Backbones	VGA	HD	4K
ResNet-152 (GPU)	75.1	443.2	1742
MobileNet-0.25 (GPU)	1.4	6.1	25.6
MobileNet-0.25 (CPU-m)	5.5	50.3	-
MobileNet-0.25 (CPU-1)	17.2	130.4	-
MobileNet-0.25 (ARM)	61.2	434.3	-

TinyFace

Method	Easy	Medium	Hard
ACF[23]	0.659	0.541	0.273
Two-stage CNN[25]	0.681	0.618	0.323
Multiscale Cascade CNN[24]	0.691	0.634	0.345
Faceness[24]	0.713	0.664	0.424
Multitask Cascade CNN[26]	0.848	0.825	0.598
CMS-RCNN[27]	0.899	0.874	0.624
HR-VGG16	0.862	0.844	0.749
HR-ResNet50	0.907	0.890	0.802
HR-ResNet101	**0.919**	**0.908**	**0.823**

J. Deng et al., ” RetinaFace: Single-stage Dense Face Localisation in the Wild,” arXiv 2019.

RetinaFace

- Important point to note: inference latency

Inference time on Google Pixel 3 XL GPU

Model	Backbone	Input size	Inference time
RetinaFace	ResNet-50	1,920x1,080	11,270 ms
	MobileNet-v1	1,920x1,080	918 ms
TinyFace	ResNet-50	1,280x720	1,623 ms
		2,560x1,440	6,469 ms

J. Deng et al., ” RetinaFace: Single-stage Dense Face Localisation in the Wild,” arXiv 2019.

RetinaFace

- Detection examples (ResNet backbone)

RetinaFace

- Detection examples (ResNet backbone)

RetinaFace

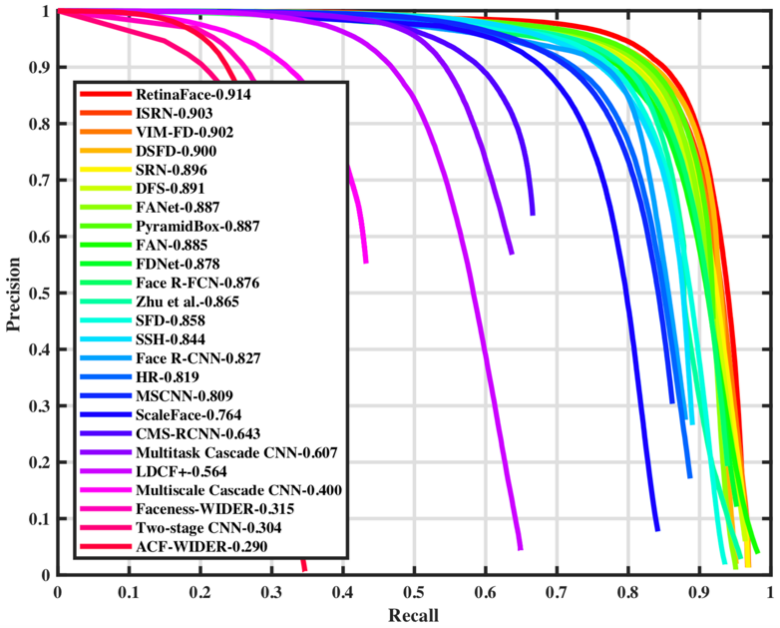
- Detection examples (MobileNet backbone)

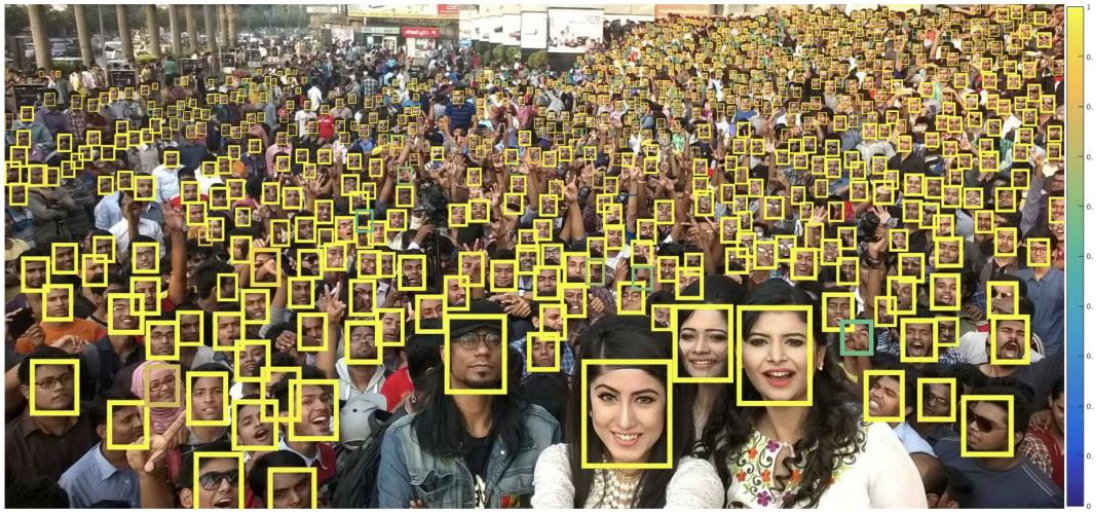
RetinaFace

- Detection examples (MobileNet backbone)

Our Experience

- In terms of detection accuracy, anything better than TinyFace is sufficient in practice
- What matters is the complexity & availability of the model for your choice of platform





Content Overview

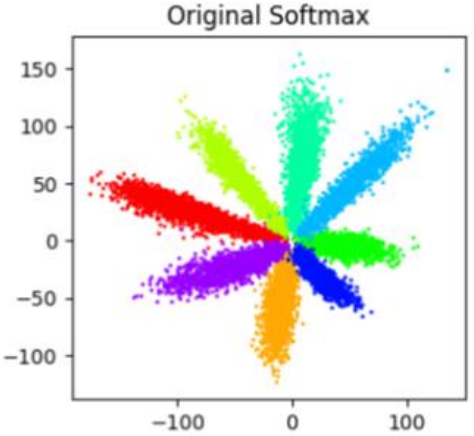
- Task overview
- Face detection
  - ✓MTCNN
  - ✓TinyFace
  - ✓SSH & PyramidBox
  - ✓RetinaFace
- **Face recognition**
  - ✓Metric learning
  - ✓Angular classification (SphereFace/CosFace/ArcFace)
  - ✓Target-specific face recognition

Overview

- Face recognition is fundamentally an image classification problem
- Many optimized networks have been proposed (e.g., ResNet)
- Problem: too many identities to classify
  - ✓ e.g., 85K (MS-Celeb-1M dataset), 672K (MegaFace)

→ Traditional softmax loss lacks discriminative power

- How can we find efficient loss function for accurate classification?



https://www.microsoft.com/en-us/research/project/ms-celeb-1m-challenge-recognizing-one-million-celebrities-real-world/
http://megaface.cs.washington.edu/

Metric Learning

- Task of learning a distance function over objects.
- Early stage research trend in face recognition

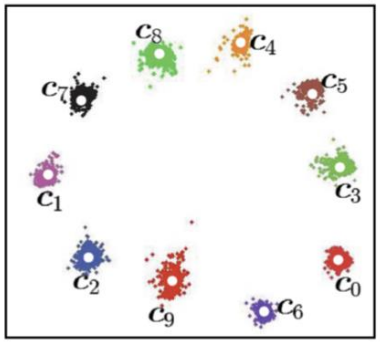
[NIPS '14] Contrastive loss

$$\text{Verif}(f_i, f_j, y_{ij}, \theta_{ve}) = \begin{cases} \frac{1}{2}\|f_i - f_j\|_2^2 & \text{if } y_{ij} = 1 \\ \frac{1}{2}\max\left(0, m - \|f_i - f_j\|_2\right)^2 & \text{if } y_{ij} = -1 \end{cases}$$

[CVPR '15] Triplet loss

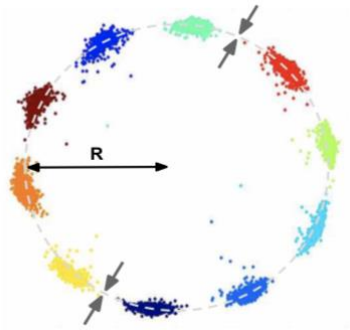
$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]$$

[ECCV '16] Center loss



$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m}\|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

[CVPR '18] Ring loss



$$L_R = \frac{\lambda}{2m}\sum_{i=1}^{m}(\|\mathcal{F}(\mathbf{x}_i)\|_2 - R)^2$$

Metric Learning

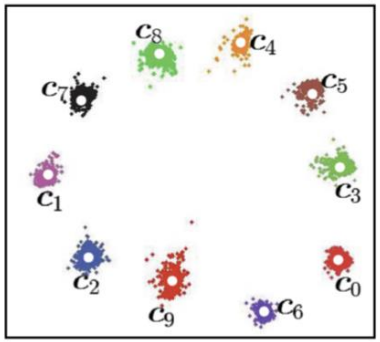
- Task of learning a distance function over objects.
- Early stage research trend in face recognition
- **Problem: requires a lot of labor & computation overhead**

[CVPR '15] Triplet loss

$$\sum_{i}^{N}\left[\|f(x_i^a)-f(x_i^p)\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\alpha\right]$$

→ How can we choose the right positive/negative pairs?

[ECCV '16] Center loss

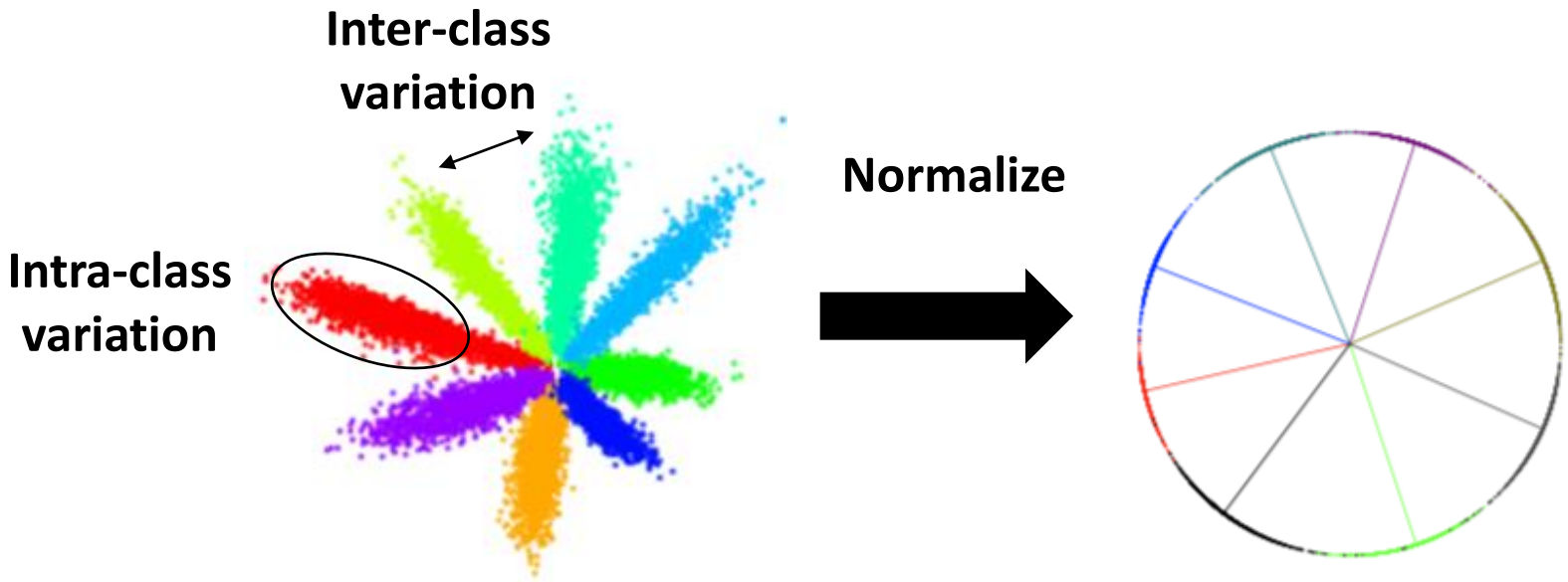


$$\mathcal{L}_C=\frac{1}{2}\sum_{i=1}^{m}\|\boldsymbol{x}_i-\boldsymbol{c}_{y_i}\|_2^2$$

→ Need to compute center every epoch

Angular Classification

- Alternative: revamp the softmax function itself
- Recent trend: face recognition as angular classification (i.e., ignore intra-class variation in radial direction)



Variants of Softmax

[CVPR ‘17] SphereFace

$$L_{\text{ang}} = \frac{1}{N}\sum_i -\log\left(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})} + \sum_{i\neq y_i} e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\right)$$

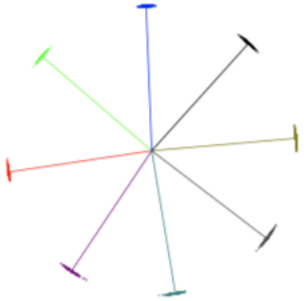
[CVPR ‘18] CosFace

$$L_{lmc} = \frac{1}{N}\sum_i -\log\frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_{j,i})}}$$

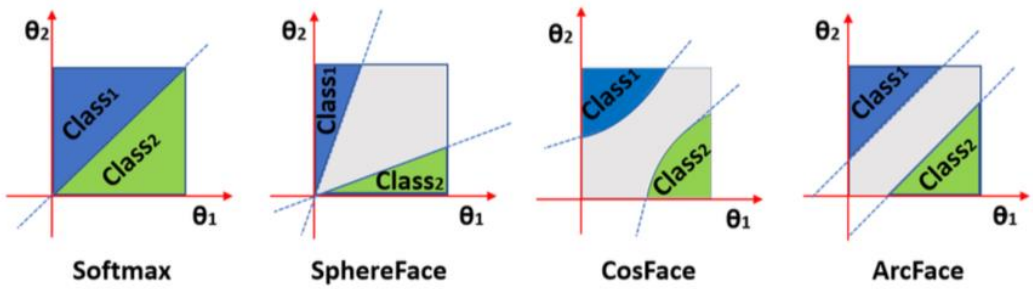
[CVPR ‘19] ArcFace

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}$$

(a) Softmax



(b) ArcFace



W. Liu et al., “SphereFace: Deep Hypersphere Embedding for Face Recognition,” CVPR 2017.
H. Wang et al., “CosFace: Large Margin Cosine Loss for Deep Face Recognition,” CVPR 2018.
J. Deng et al., “ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” CVPR 2019.

Variants of Softmax

SphereFace, CosFace, and ArcFace can be combined as

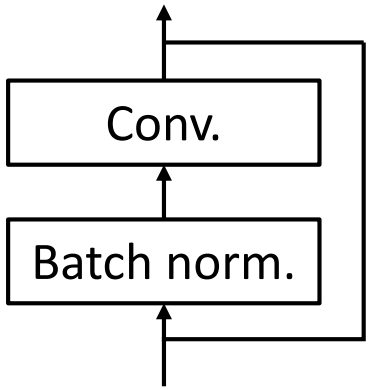
$$-\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

W. Liu et al., "SphereFace: Deep Hypersphere Embedding for Face Recognition," CVPR 2017.
H. Wang et al., "CosFace: Large Margin Cosine Loss for Deep Face Recognition," CVPR 2018.
J. Deng et al., "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," CVPR 2019.

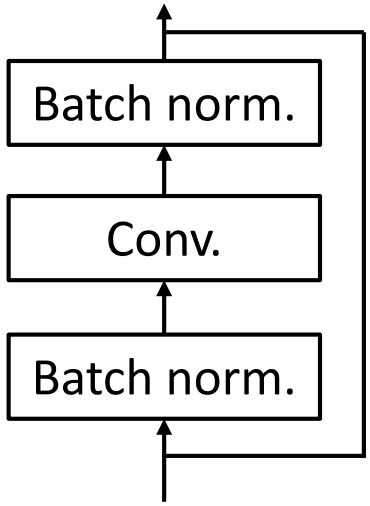
Backbone Network

- Most of the papers employ ResNet-like network backbones with lots of batch normalization layers

Plain residual block



[CVPR '19] ArcFace



MobileFaceNet

Input	Operator	t	c	n	s
$112^2 \times 3$	conv3x3	-	64	1	2
$56^2 \times 64$	depthwise conv3x3	-	64	1	1
$56^2 \times 64$	bottleneck	2	64	5	2
$28^2 \times 64$	bottleneck	4	128	1	2
$14^2 \times 128$	bottleneck	2	128	6	1
$14^2 \times 128$	bottleneck	4	128	1	2
$7^2 \times 128$	bottleneck	2	128	2	1
$7^2 \times 128$	conv1x1	-	512	1	1
$7^2 \times 512$	linear GDConv7x7	-	512	1	1
$1^2 \times 512$	linear conv1x1	-	128	1	1

J. Deng et al., " ArcFace: Additive Angular Margin Loss for Deep Face Recognition," CVPR 2019.
S. Chen et al., "MobileFaceNets: Efficient CNNs for Accurate Real-time Face Verification on Mobile Devices," arXiv 2018.

Backbone Network

- Performance

ArcFace with varying backbones

Model	Accuracy			Inference time	
	LFW	VGG2_FP	CFP_FP	Pixel 3 XL	GTX 2080 Ti
ResNet100	99.46 (99.77)	94.14 (95.24)	96.40 (98.27)	395 ms	19.3 ms
ResNet50	99.31 (99.80)	92.50 (93.04)	91.80 (92.74)	193 ms	7.1 ms
MobileNet	99.00 (99.45)	86.50 (90.64)	82.76 (89.77)	18 ms	2.2 ms

Target-Specific Face Recognition

- Datasets

Celebrities in Frontal-Profile (CFP)

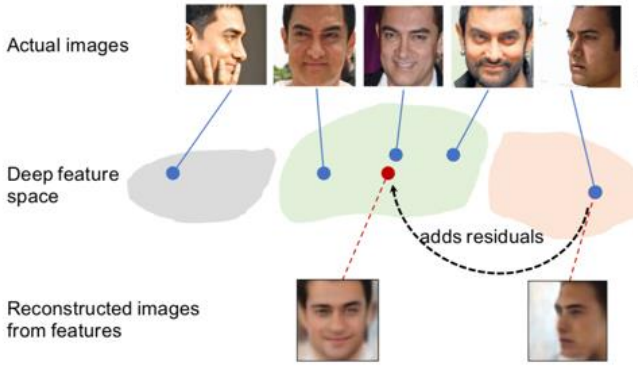
CASIA Near Infra-Red vs Visible light 2.0 (NIR-VIS)

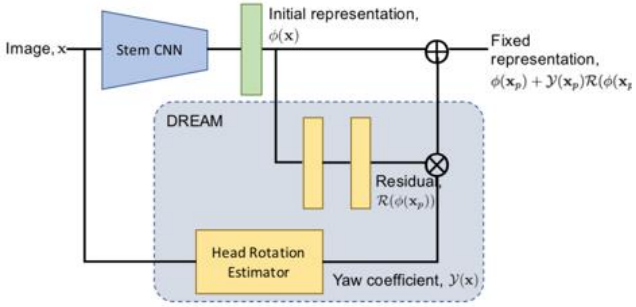
Cross-Pose LFW (CPLFW)

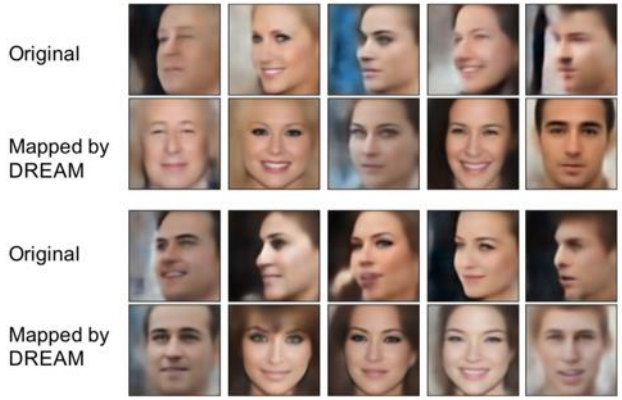
Cross-Age LFW (CALFW)

Target-Specific Face Recognition

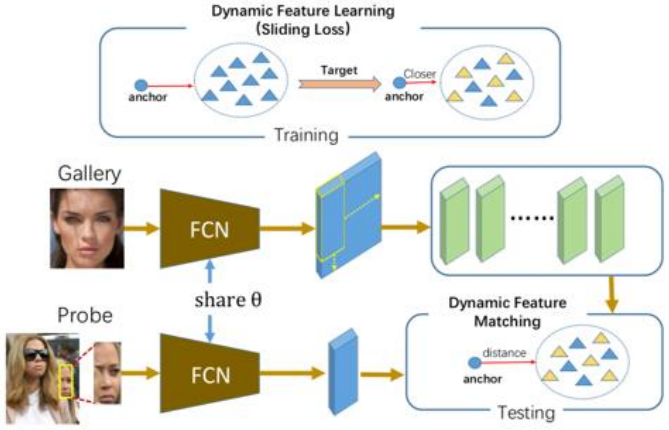
- Profile face recognition







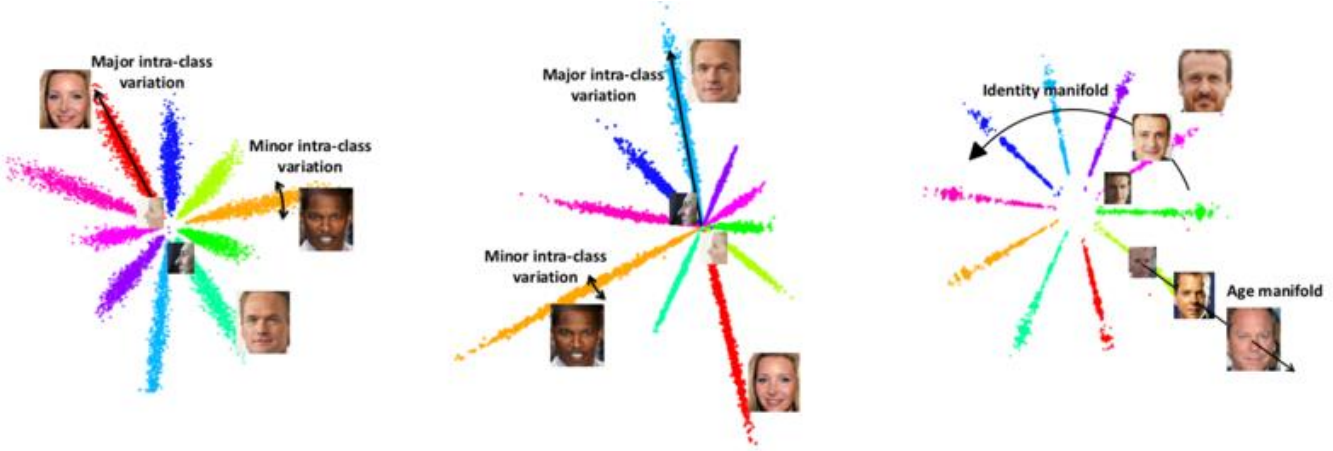
- Partial face recognition



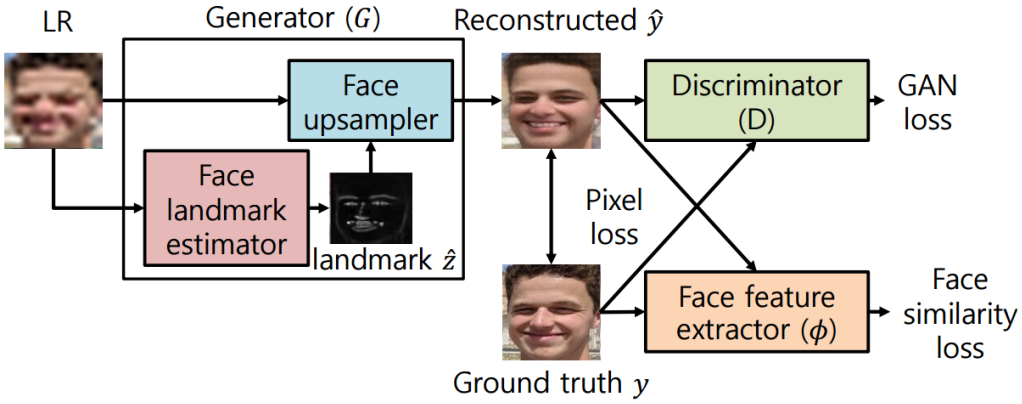
J. Zhao et al., "Towards Pose Invariant Face Recognition in the Wild," CVPR 2018.
K. Cao et al., "Pose-Robust Face Recognition via Deep Residual Equivariant Mapping," CVPR 2018.
L. He et al., "Dynamic Feature Learning for Partial Face Recognition," CVPR 2018.

Target-Specific Face Recognition

- Age-invariant face recognition



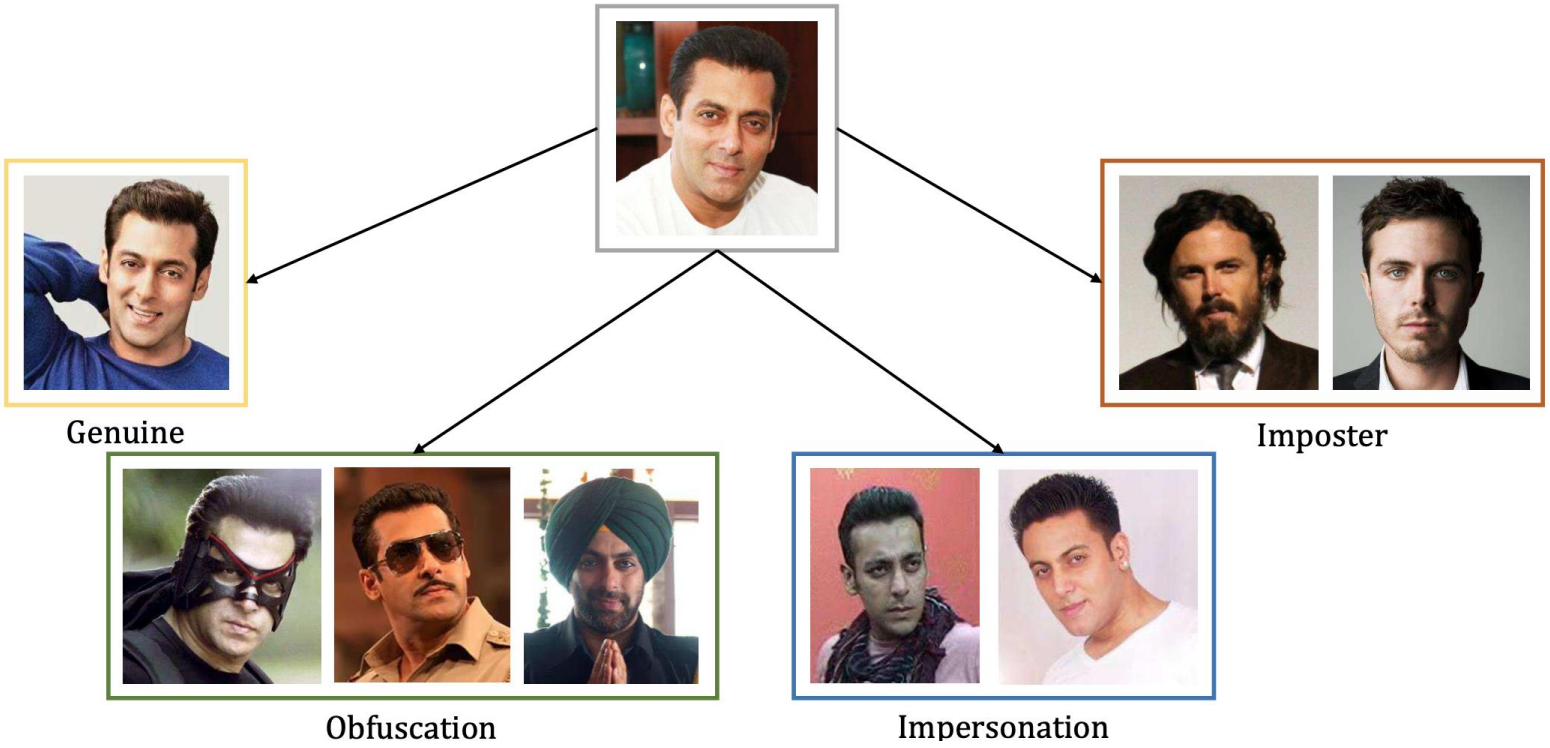
- Low-resolution face recognition



Y. Wang et al., "Orthogonal Deep Features Decomposition for Age-Invariant Face Recognition," ECCV 2018.
J. Yi et al., "AR-based Person Identification in Crowded Urban Spaces," ACM MobiCom 2020.

Trend: Disguised Faces in the Wild

- Challenge held on CVPRW 2018, ICCVW 2019
- Mostly focused on existing approach + training on DFW dataset



V. Kushuwaha et al., ” Disguised Faces in the Wild,” CVPRW 2018.

Trend: Disguised Faces in the Wild

- Challenge held on CVPRW 2018, ICCVW 2019
- Mostly focused on existing approach + training on DFW dataset

Table 1: Review of existing disguise face datasets.

Name	Year	Controlled	Number of		Publicly Available
			Images	Subjects	
AR Dataset [12]	1998	Yes	3,200	126	Yes
National Geographic Dataset [15]	2004	Yes	46	1	No
Synthetic Disguise Dataset [19]	2009	Yes	4,000	100	No
IIIT-Delhi Disguise V1 Dataset [5]	2014	Yes	684	75	Yes
Disguised and Makeup Faces Dataset [22]	2016	No	2,460	410	Yes
Proposed DFW Dataset	**2018**	**No**	**11,157**	**1,000**	**Yes**

Table 2: Number of images in the training and testing partition of the proposed DFW dataset.

Number of	Training Set	Testing Set
Subjects	400	600
Images	3,386	7,771
Normal Images	400	600
Validation Images	308	595
Disguised Images	1,756	3,058
Impersonator Images	922	3,518

Trend: Disguised Faces in the Wild

- Challenge held on CVPRW 2018, ICCVW 2019
- Mostly focused on existing approach + training on DFW dataset
- Winner of ICCVW challenge: ArcFace

Algorithm	GAR	
	@1%FAR	@0.1%FAR
VGG-Face	52.77	27.05
ByteFace	75.53	55.11
DDRNET	84.20	51.26
DenseNet + COST	92.1	62.2
DR-GAN	65.21	11.93
LearnedSiamese	57.64	27.73
MEDC	91.26	55.46
OcclusionFace	93.44	46.21
UMDNets	94.28	53.27
WVU_CL	81.34	40.00
AEFRL	96.80	57.64
MiRA-Face	95.46	51.09
ArcFace	**98.66**	**60.84**

Table 8. Verification accuracy (%) of ArcFace and the baselines on protocol-1 (impersonation) of the validation dataset. Results of other methods are from [24].

Algorithm	GAR	
	@1%FAR	@0.1%FAR
VGG-Face	31.52	15.72
ByteFace	76.97	21.51
DenseNet + COST	87.1	72.1
DDRNET	71.04	49.28
DisguiseNet	66.32	28.99
DR-GAN	74.56	58.31
LearnedSiamese	37.81	16.95
MEDC	81.25	65.14
OcclusionFace	80.45	66.05
Tessellation	1.23	0.18
UMDNets	86.62	74.69
WVU_CL	78.77	61.82
AEFRL	87.82	77.06
MiRA-Face	90.65	80.56
ArcFace	**95.08**	**92.20**

Table 9. Verification accuracy (%) of ArcFace and the baselines on protocol-2 (obfuscation) of the validation dataset. Results of other methods are from [24].

Algorithm	GAR	
	@1%FAR	@0.1%FAR
VGG-Face	33.76	17.73
ByteFace	75.53	54.16
DenseNet + COST	87.6	71.5
DDRNET	71.43	49.08
DisguiseNet	60.89	23.25
DR-GAN	74.89	57.30
LearnedSiamese	39.73	18.79
MEDC	81.31	63.22
OcclusionFace	80.80	65.34
Tessellation	1.23	0.17
WVU_CL	79.04	60.13
UMDNets	86.75	72.90
AEFRL	87.90	75.54
MiRA-Face	90.62	79.26
ArcFace	**95.11**	**91.76**

Table 10. Verification accuracy (%) of ArcFace and the baselines on protocol-3 (overall) of the validation dataset. Results of other methods are from [24].

J. Deng et al., " ArcFace for Disguised Face Recognition," ICCVW 2019.

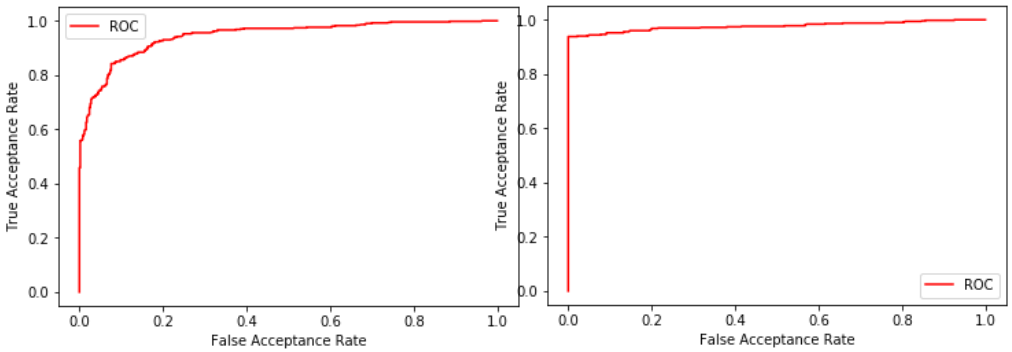
Our Experience

- Face cropping & alignment affects accuracy

Bounding box-based cropping (a)

Landmark-based cropping (b)

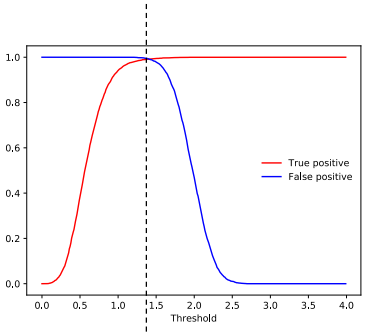
Methods	LFW	CFP-FP	AgeDB-30
MTCNN+ArcFace [11]	99.83	98.37	98.15
RetinaFace+ArcFace	**99.86**	**99.49**	**98.60**

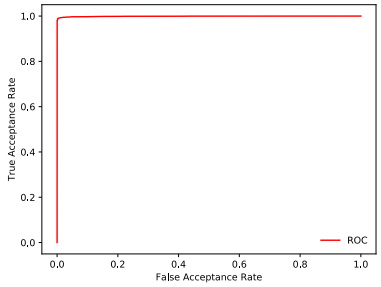


Our Experience

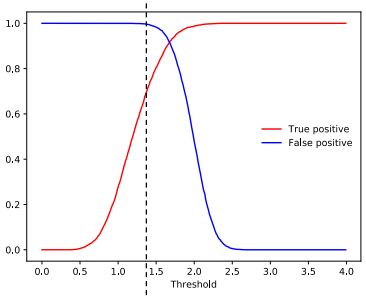
- The devil of face recognition lies in determining the threshold

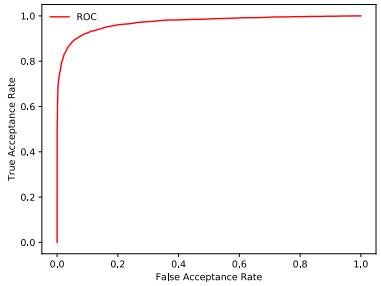
LFW



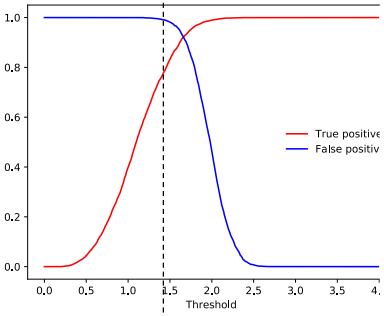


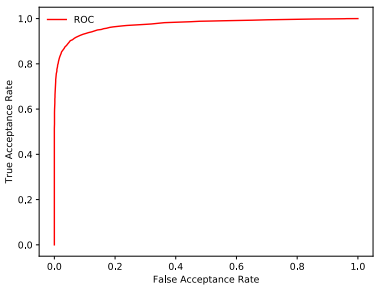
CFP





VGGFace2



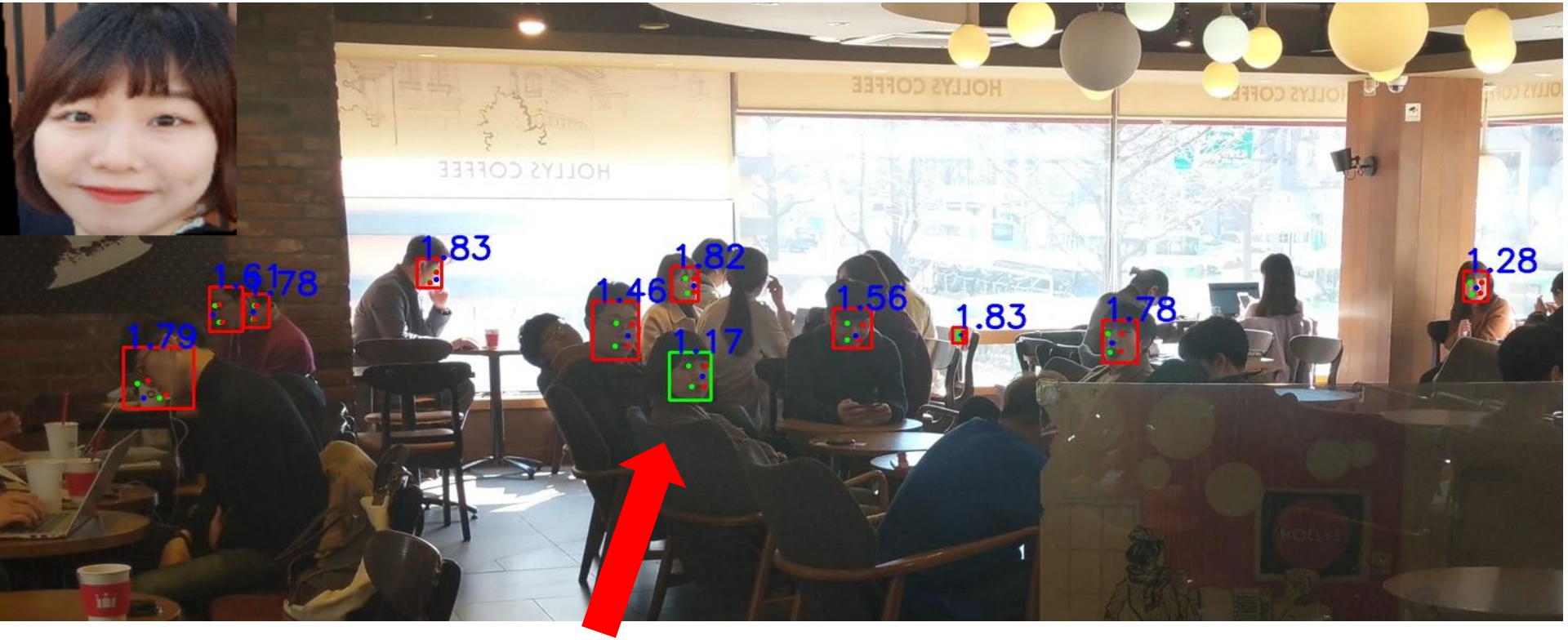


Accuracy = (TP+TN)/N

Dataset	Optimal	Fixed (1.63)
LFW	99.3	96.5
CFP	91.8	91.8
VGGFace2	92.5	92.3

Our Experience

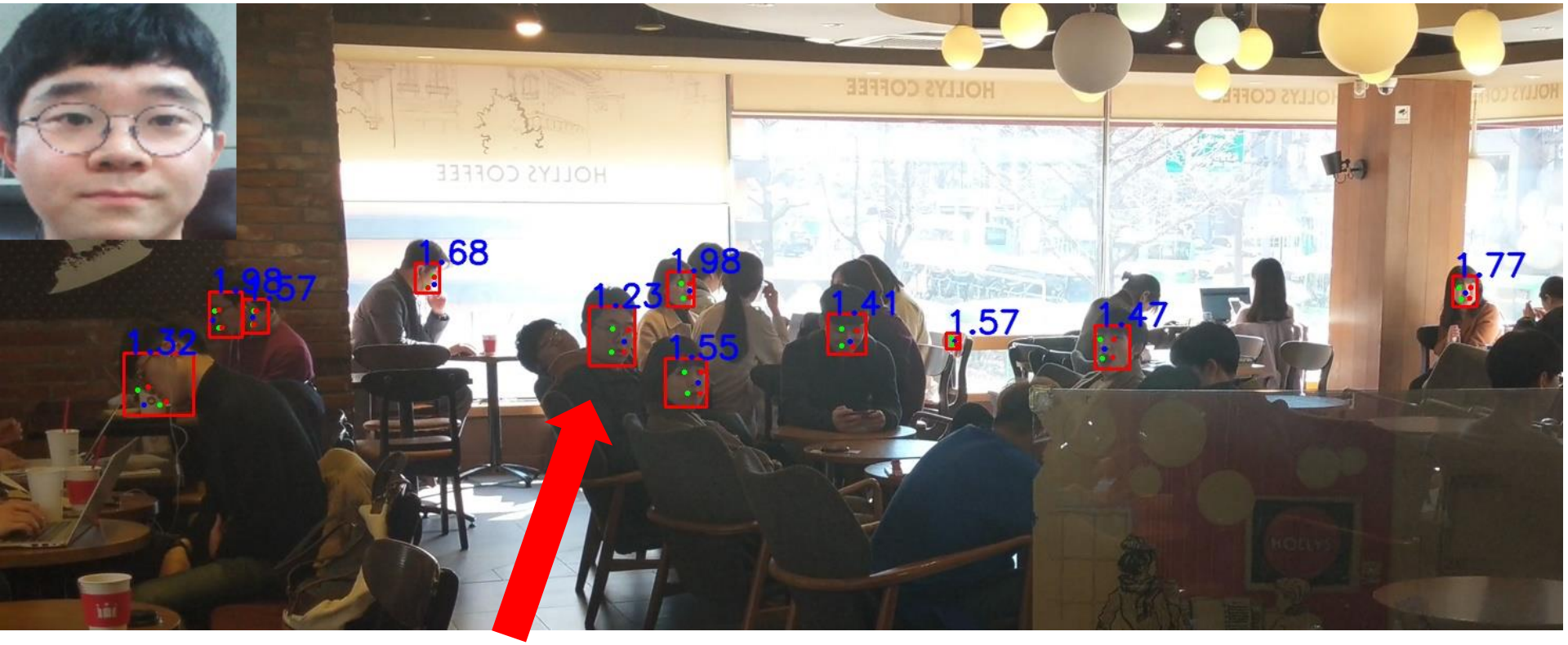
- Let's say we set the threshold as 1.2



Closes distance among the faces in the scene & below threshold
→ desired result!

Our Experience

- Let's say we set the threshold as 1.2



Closes distance among the faces in the scene & but above threshold → undesired result!

EagleEye: Wearable Camera-based Person Identification in Crowded Urban Spaces

Juheon Yi[1], Sunghyun Choi[2], Youngki Lee[1]

[1]Seoul National University, [2]Samsung Research Korea

ACM MobiCom 2020

SEOUL
NATIONAL
UNIVERSITY

Needs for Robust Person Identification

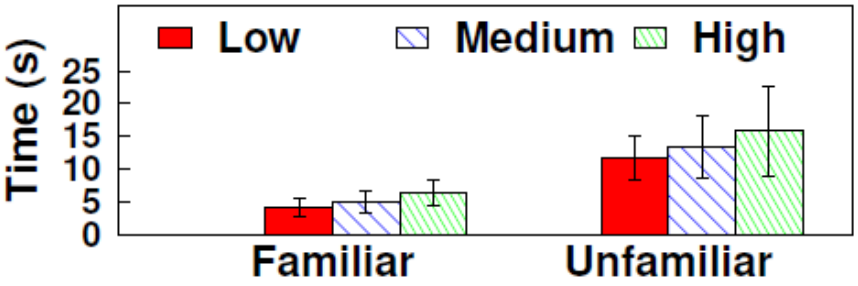
Finding a crime suspect in crowded streets

Children counting in kindergarten field trips

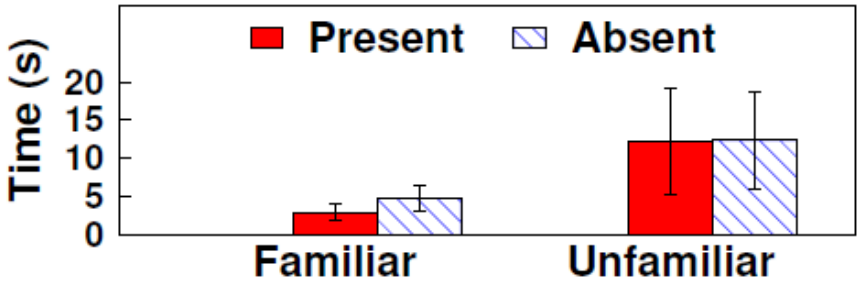
Humans: How Fast and Accurate?

Human Vulnerability to Cognitive Overload

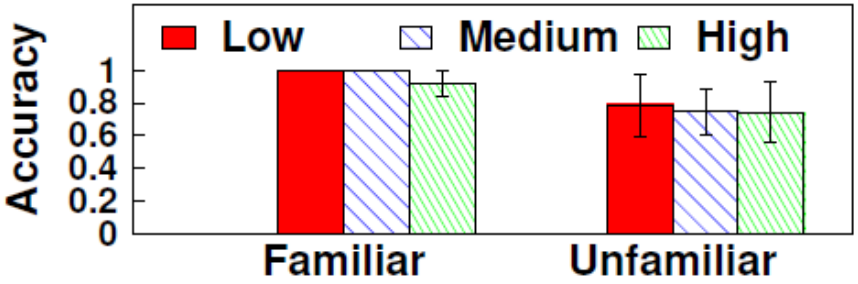
- **Latency/accuracy of humans finding a target person on images displayed on laptop screen**



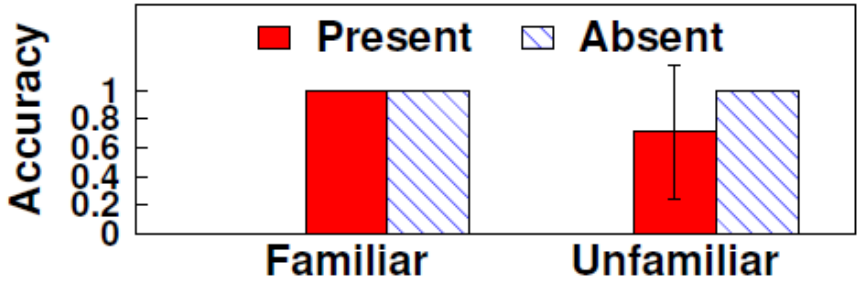
(a) Crowdedness (latency).



(b) Presence vs. absence (latency).



(d) Crowdedness (accuracy).

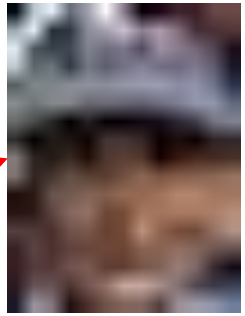


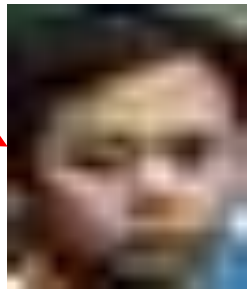
(e) Presence vs. absence (accuracy).

EagleEye: AR Person Identification System

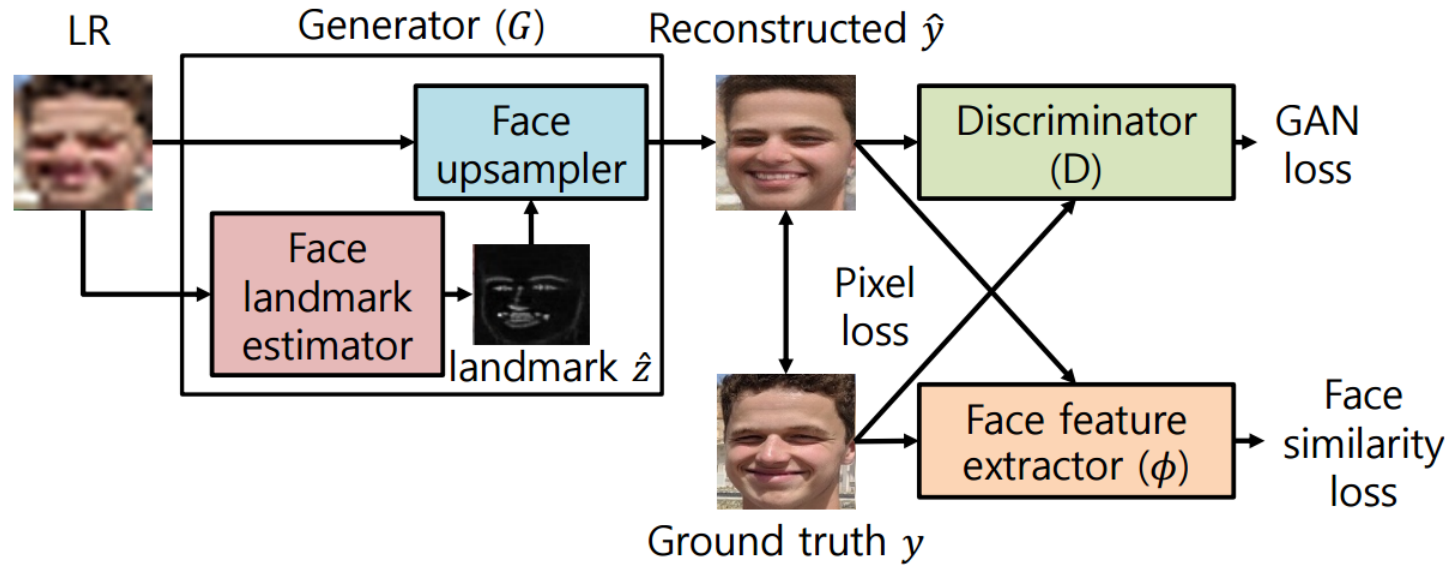
Challenge #1: Identification Accuracy

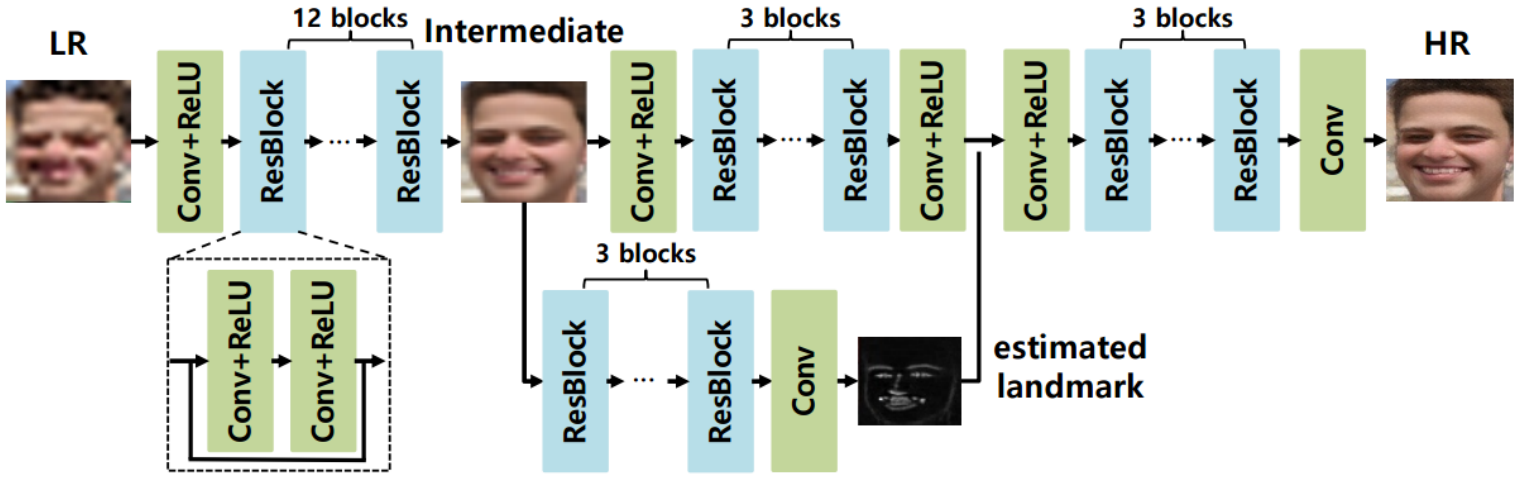
- **Faces captured from distance lacks facial details**





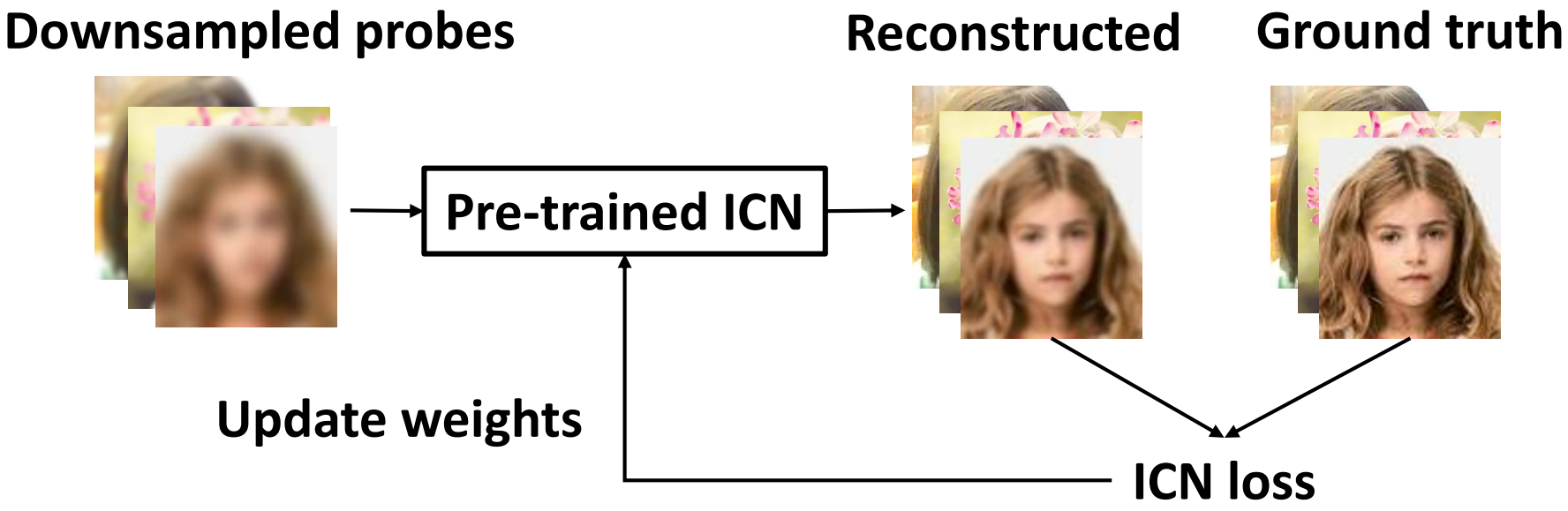
Solution: Identity Clarification Network (ICN)





Identity-Specific Fine-Tuning

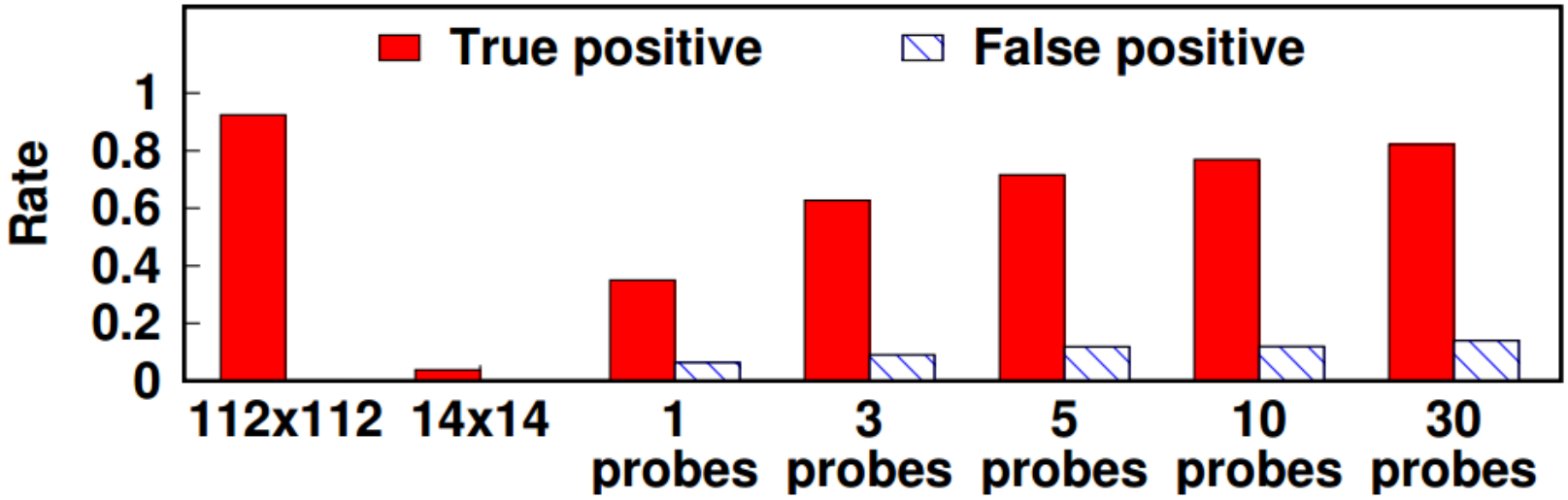
- Observation: **we know who we are looking for!**
- Identity-Specific Fine-Tuning: **fine-tune** the baseline ICN **with reference photos (probes)**



Key Evaluation Results

- **ICN enhances TP by 78% at the cost of 14% FP**

(a) 112×112. (b) 14×14. (c) Baseline. (d) Fine-tuned



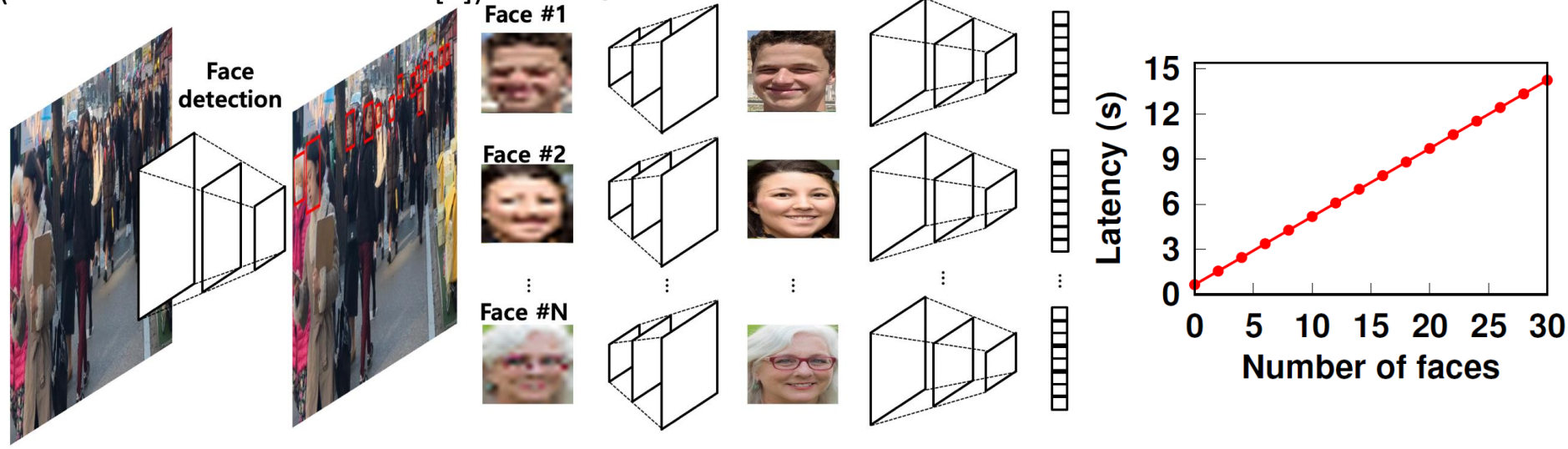
Challenge #2: Computational Complexity

- Face identification: **sequential execution of multiple DNNs**

648 ms per 1080p frame
(MobileNet-based RetinaFace [1])

166 ms per face
(ICN)

287 ms per face
(ResNet50-based ArcFace [2])

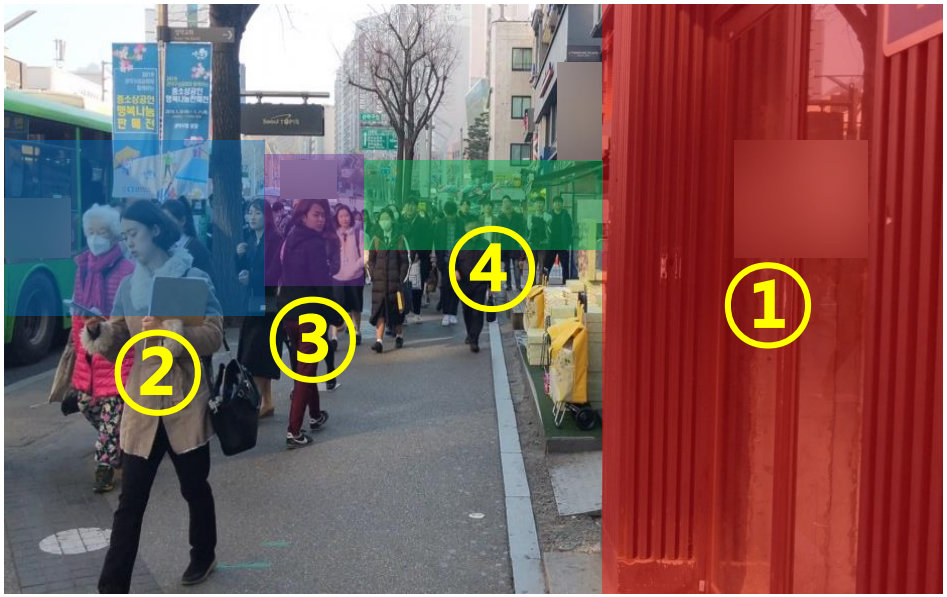


* measured on LG V50 (Qualcomm Adreno 640 GPU) with TensorFlow-Lite

[1] J. Deng *et al*, "RetinaFace: Single-stage Dense Face Localisation in the wild," arXiv 2019.
[2] J. Deng *et al*, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," CVPR 2019.

Solution: Content-Adaptive Parallel Execution

- Operational flow in a nutshell



① Background
→ Excluded from processing

② Large, frontal faces
→ Detection + lightweight recognition

③ Large, profile faces
→ Detection + heavy recognition

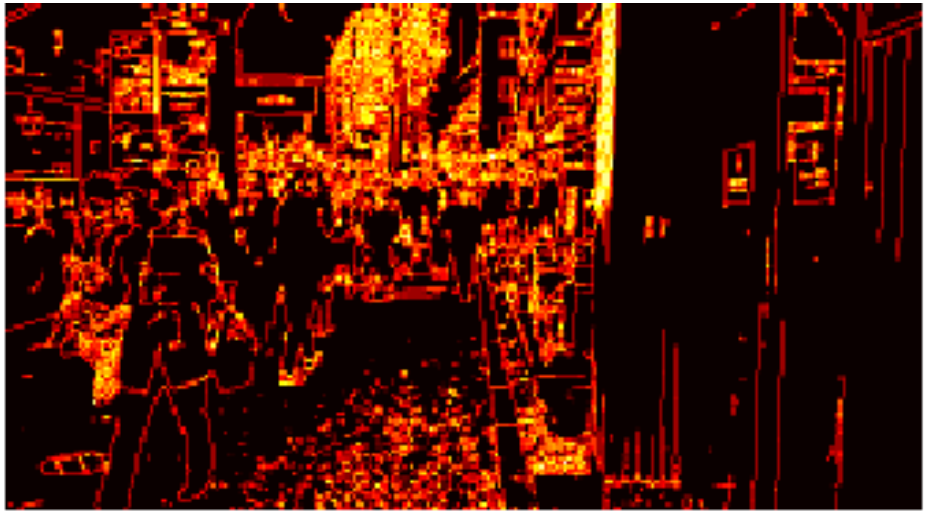
④ Small faces
→ Detection + ICN + heavy recognition

Edge-Based Background Filtering

- **Edges** give a loose hint on where faces would be

  → **Save unnecessary computation**

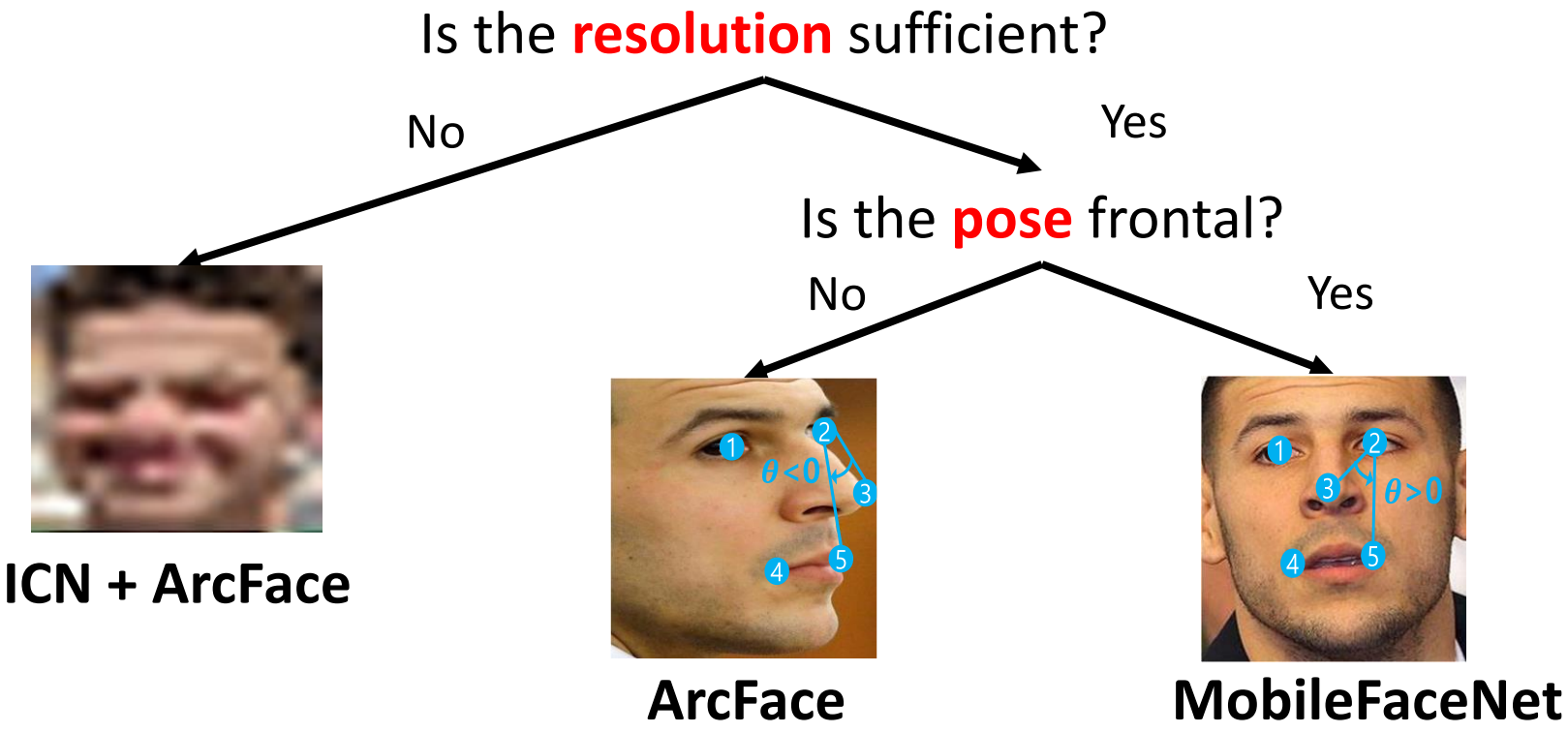
- Edge detectors run in real-time (e.g., 2 ms)



Save 64% of computation!

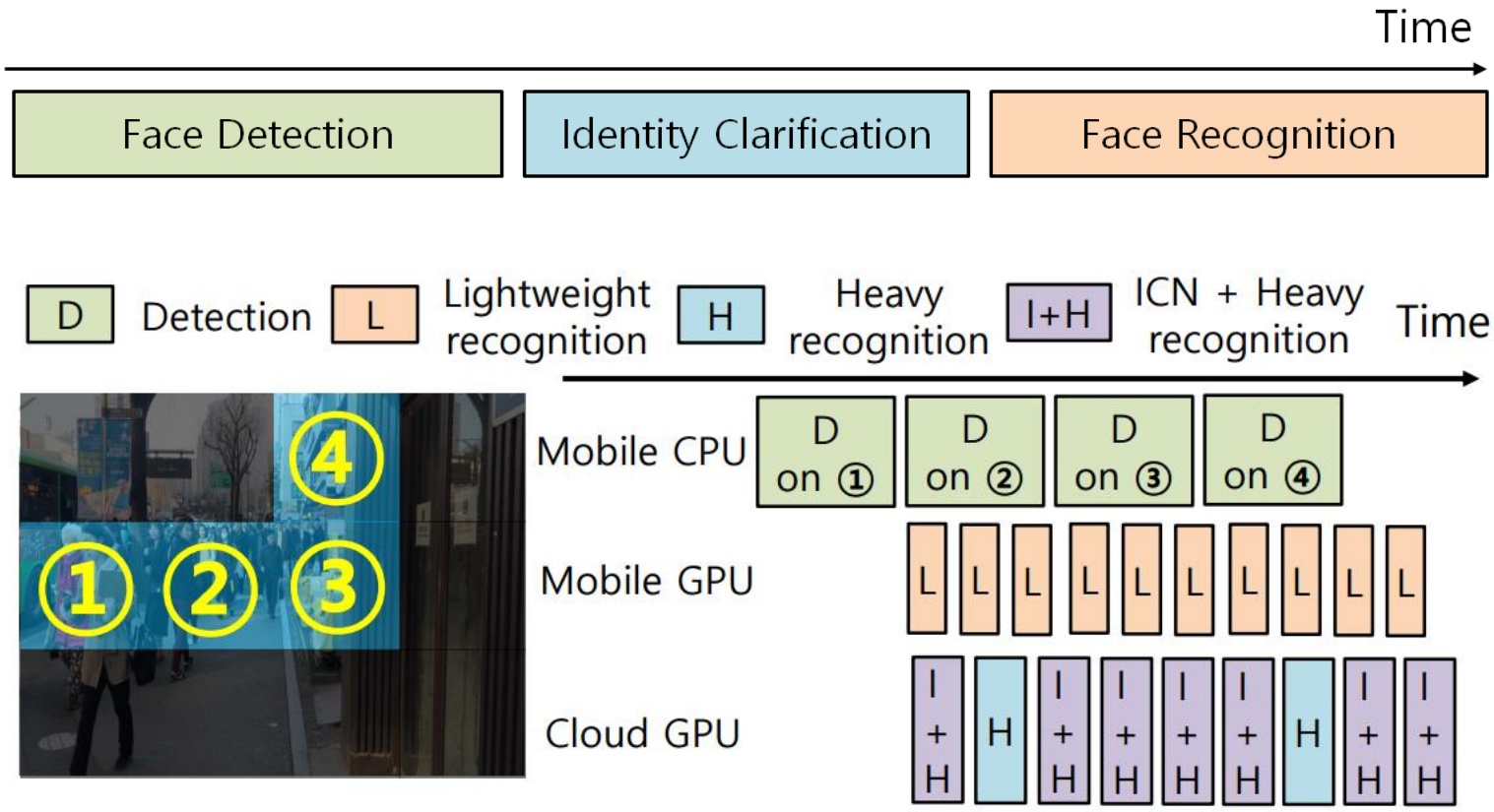
Variation-Adaptive Face Recognition

- **Select pipeline depending on recognition difficulty**



Spatial Pipelining

- **Divide image into blocks and pipeline the execution on mobile CPU & GPU, and cloud GPU**



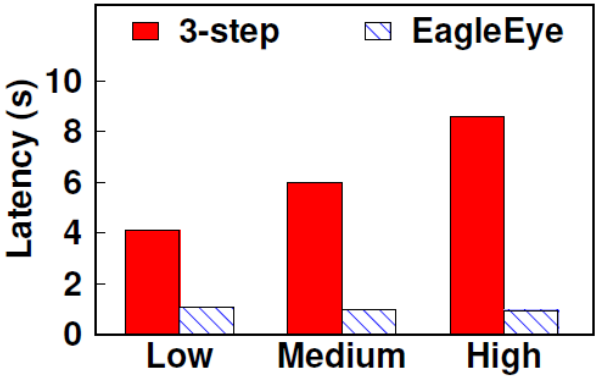
Key Evaluation Results

- **Up to 9.07x latency gain in crowded scenes**

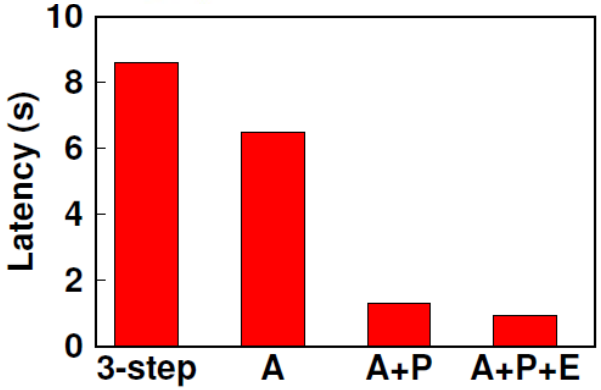
(a) Low.

(b) Medium.

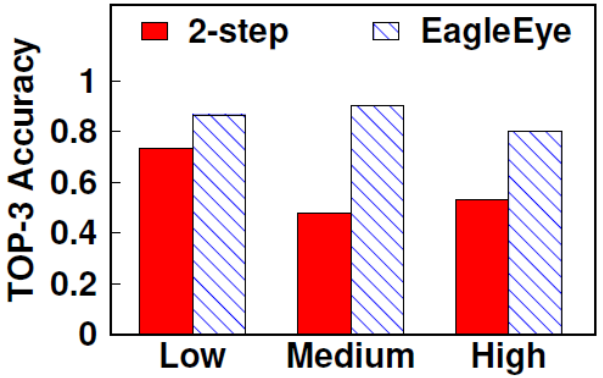
(c) High.



(a) Latency.



(b) Latency breakdown.



(c) Top-3 Accuracy.

(**A**: Adaptive recognition, **P**: Spatial pipelining,
E: Edge-based background filtering)

Summary

- We present EagleEye, an AR-based person identification system on mobile for crowded urban spaces
- Identity-Clarification Network for LR face recognition
  - Utilize reference images to fine-tune the network
  - 78% TP increase at the cost of 14% FP
- Content-Adaptive Parallel Execution for multi-DNN execution
  - 946 ms latency (9.07x gain over the baseline in crowded scenes)
  - Generalizable to various multi-DNN-enabled applications

Thank You!