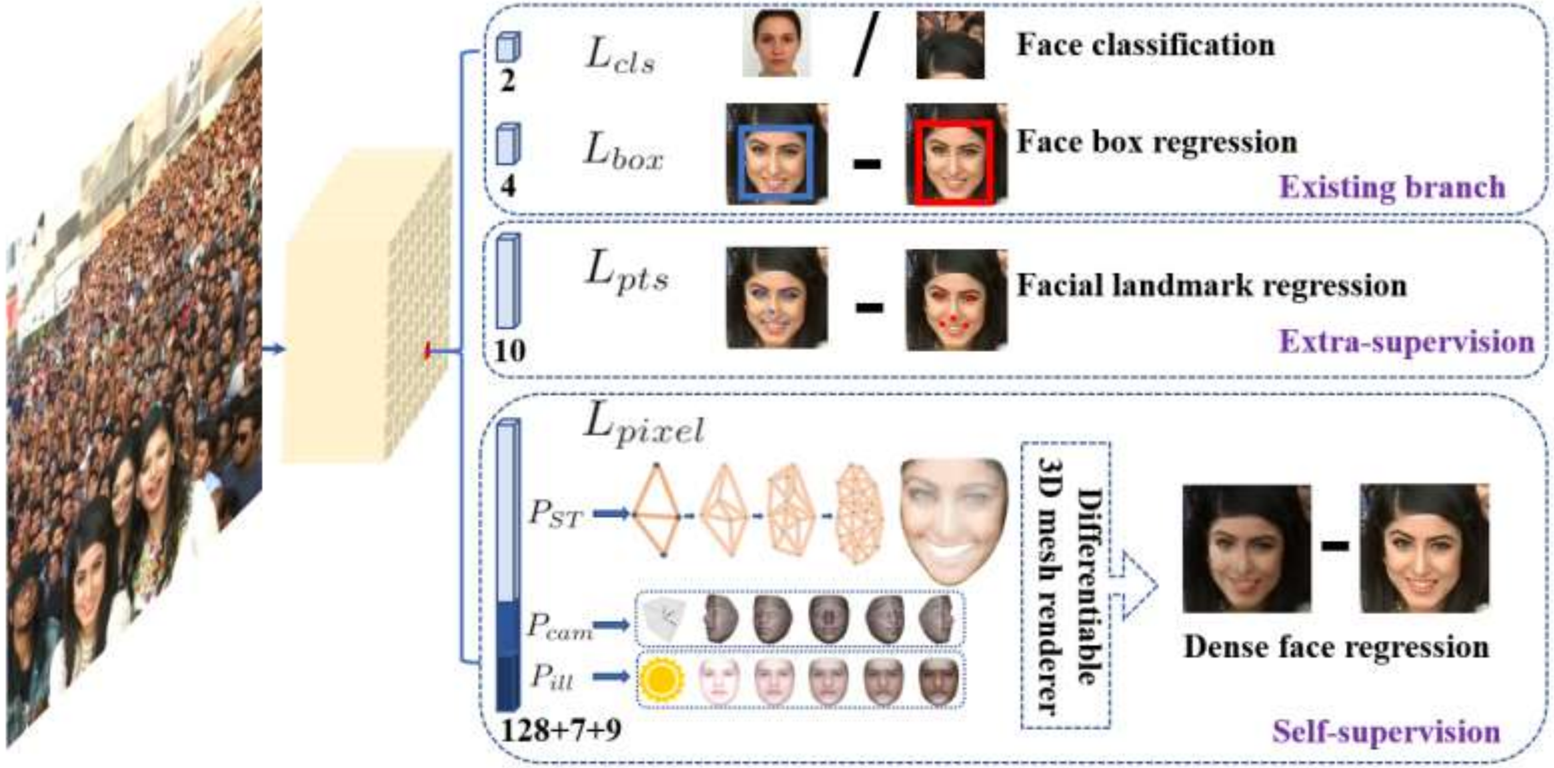
1-1. Face Detection

주제	
0. Introduction	강의 커리큘럼 소개
1. Face Recognition	1-1. Face Recognition 이론 소개
	1-2. Face Detection - 대표 모델 및 코드 소개
	1-3. [실습1] Dlib 및 Retina Face 코드 구현
	1-4. Face Alignment - 대표 모델 및 코드 소개
	1-5. [실습2] 황금비율 계산
	1-6. Face Recognition - 대표 모델 및 코드 소개
	1-7. [실습3] 그룹 가수 사진에서 각각 멤버 인식하기
2. Object Detection	2-1. Object Detection 이론 소개
	2-2. 대표 모델 – Yolov8 소개
	2-3. [실습1] 마스크 착용 유무 프로젝트
	2-4. [실습2] Tensor-RT 기반의 Yolov8, 표지판 신호등 검출
	2-5. 대표 모델 - Complex-Yolov4
	2-6. [실습3] Lidar Data 기반의 차량 Detection

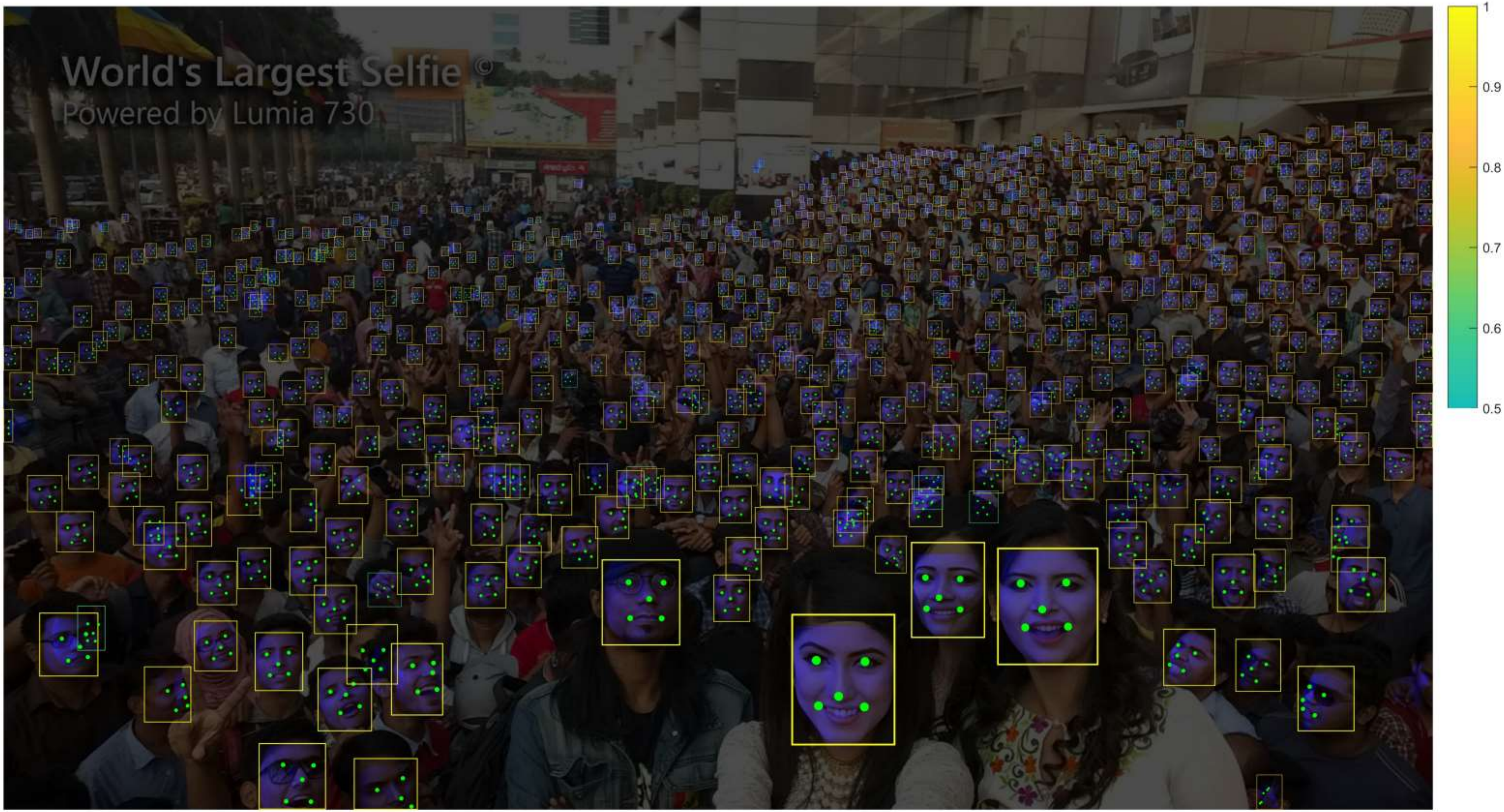
RetinaFace

RetinaFace: Single-stage Dense Face Localisation in the Wild

Jiankang Deng *[1,2,4] Jia Guo *[2] Yuxiang Zhou [1]
Jinke Yu [2] Irene Kotsia [3] Stefanos Zafeiriou[1,4]
[1]Imperial College London [2]InsightFace [3]Middlesex University London [4]FaceSoft



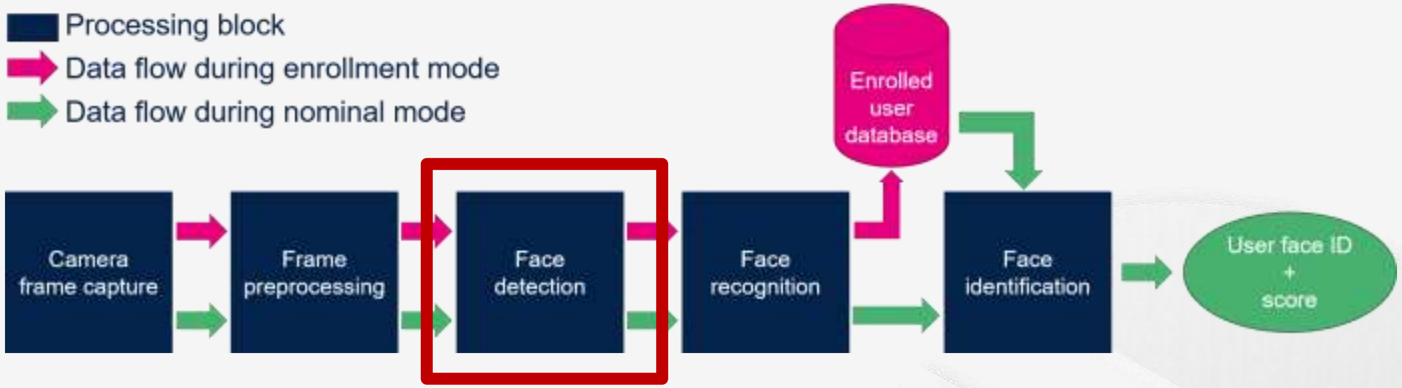
References
https://arxiv.org/abs/1905.00641



World's Largest Selfie ©
Powered by Lumia 730
1
0.9
0.8
0.7
0.6
0.5

Face Detection이란?

The most basic task on Face Recognition is of course, “Face Detecting”. Before anything, you must “capture” a face in order to recognize it, when compared with a new face captured on future.



References
https://wiki.st.com/stm32mpu/wiki/TFLite_Cpp_face_recognition

CONTENT

01

Introduction

02

Related Works

03

Proposed

04

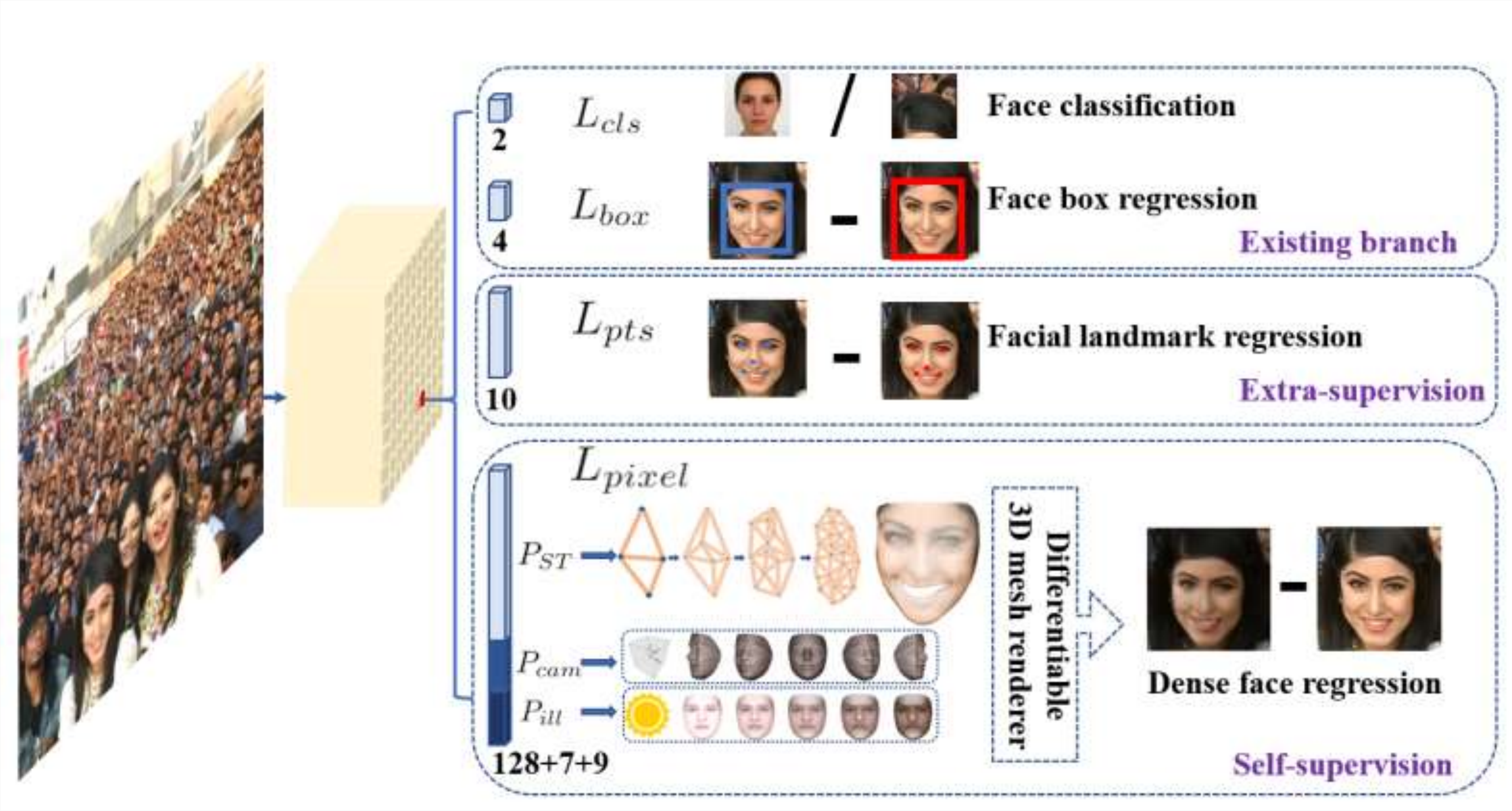
Experimental Results

05

Conclusion

Introduction

RetinaFace



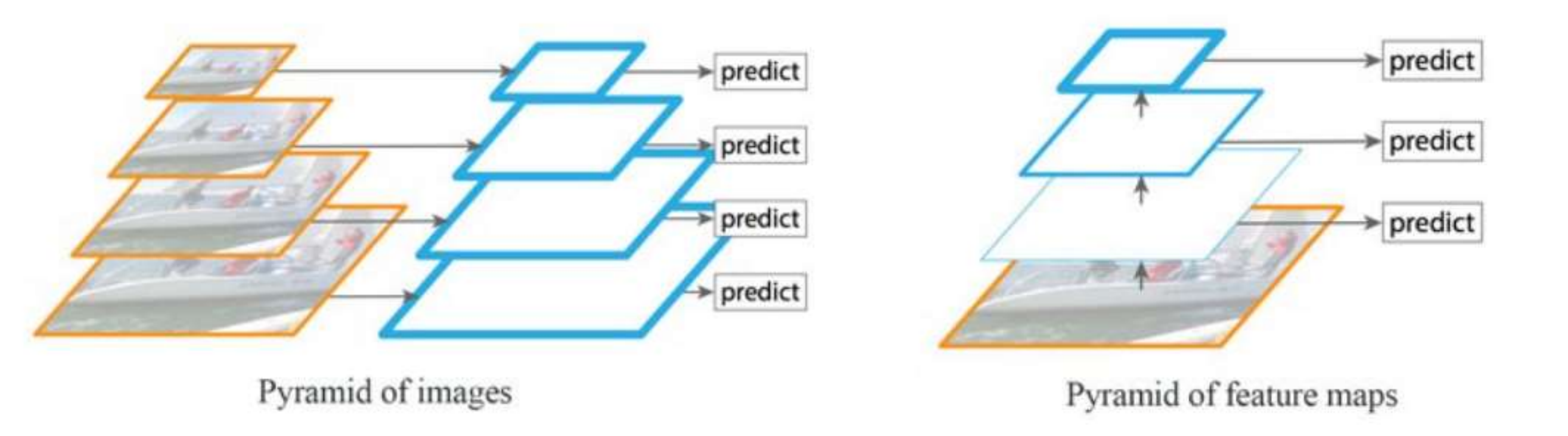
Multi-task learing

Main Contributions

- *Based on a **single-stage** design, we propose a novel pixel-wise face localisation method named RetinaFace, which employs a **multi-task learning** strategy to simultaneously predict face score, face box, five facial landmarks, and 3D position and correspondence of each facial pixel.*
- *On the WIDER FACE hard subset, RetinaFace outperforms the AP of **the state of the art two-stage method** (ISRN [67]) by 1.1% (AP equal to 91.4%).*
- *On the IJB-C dataset, RetinaFace helps to **improve ArcFace's verification accuracy** (with TAR equal to 89.59% when FAR=1e-6). This indicates that better face localisation can significantly improve face recognition.*
- *By employing light-weight backbone networks, RetinaFace can run real-time on a **single CPU core for a VGA-resolution image**.*
- ***Extra annotations** and code have been released to facilitate future research.*

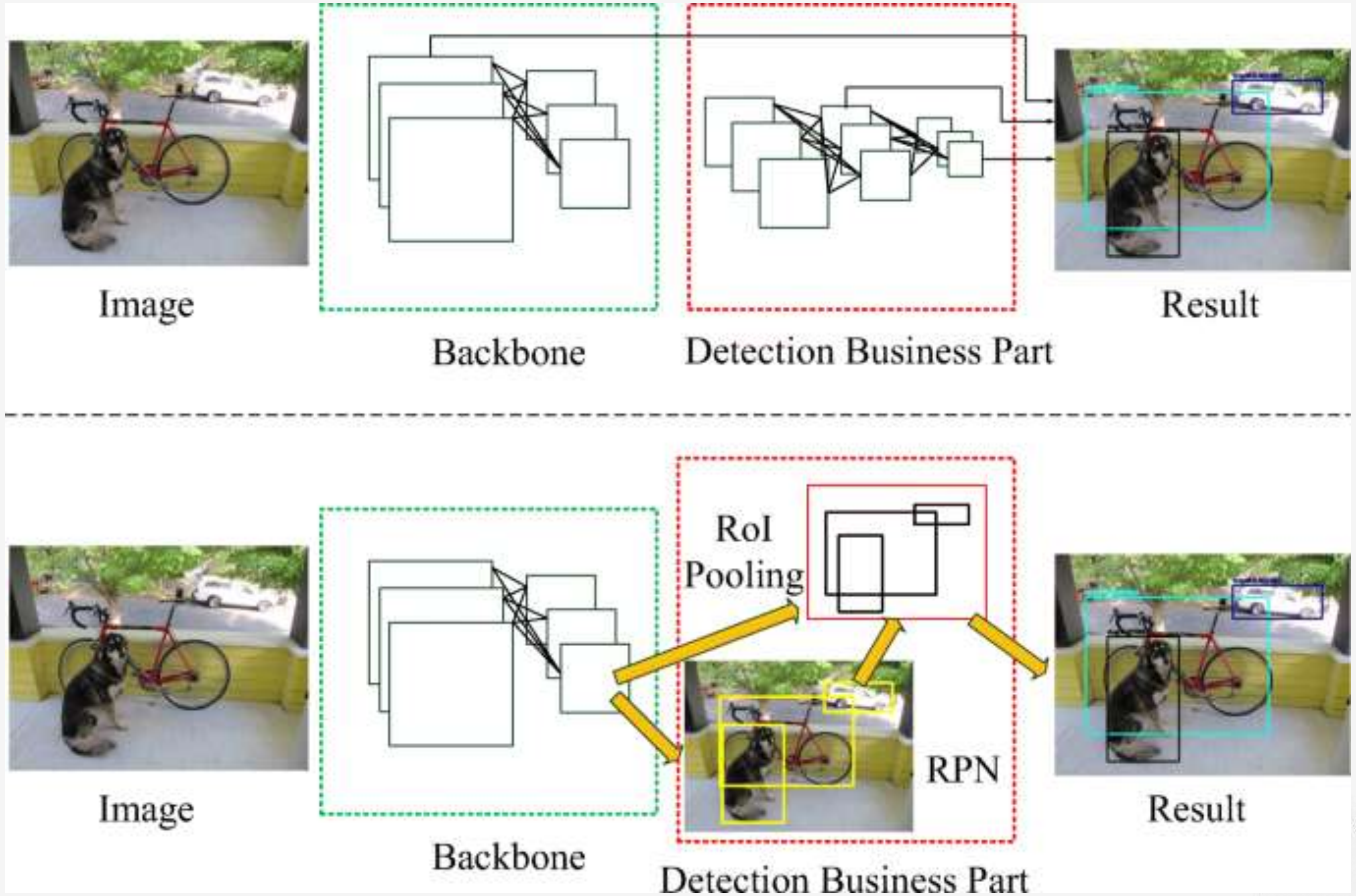
Related Works

Image Pyramid vs Feature Pyramid



References
https://wikidocs.net/162976

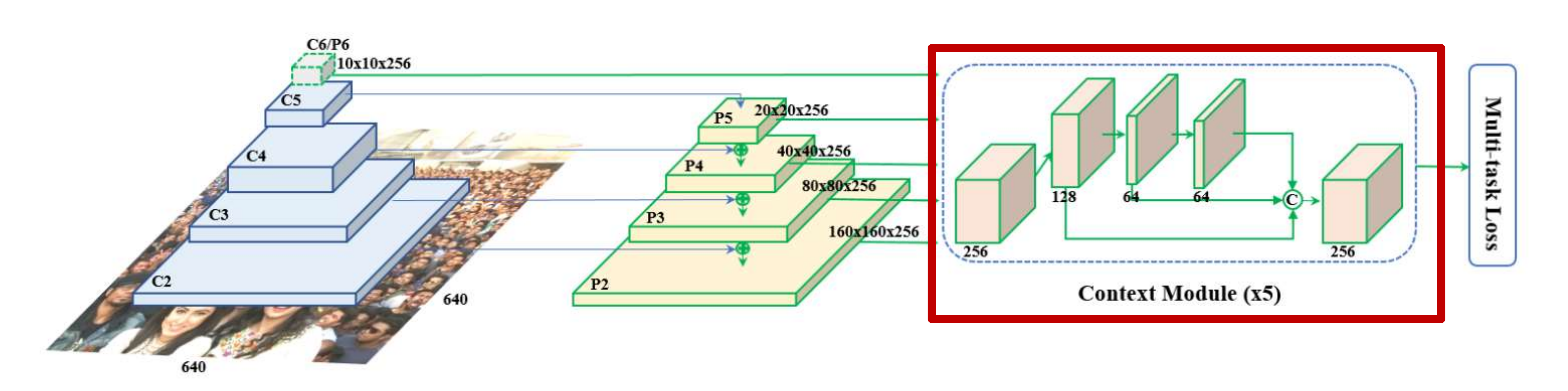
Two-stage vs Single-stage



References
https://link.springer.com/article/10.1007/s11042-019-07898-2

Context Modeling

To enhance the model's contextual reasoning power for capturing tiny faces



Context Modeling

To enhance the model's contextual reasoning power for capturing tiny faces

Deformable convolutions

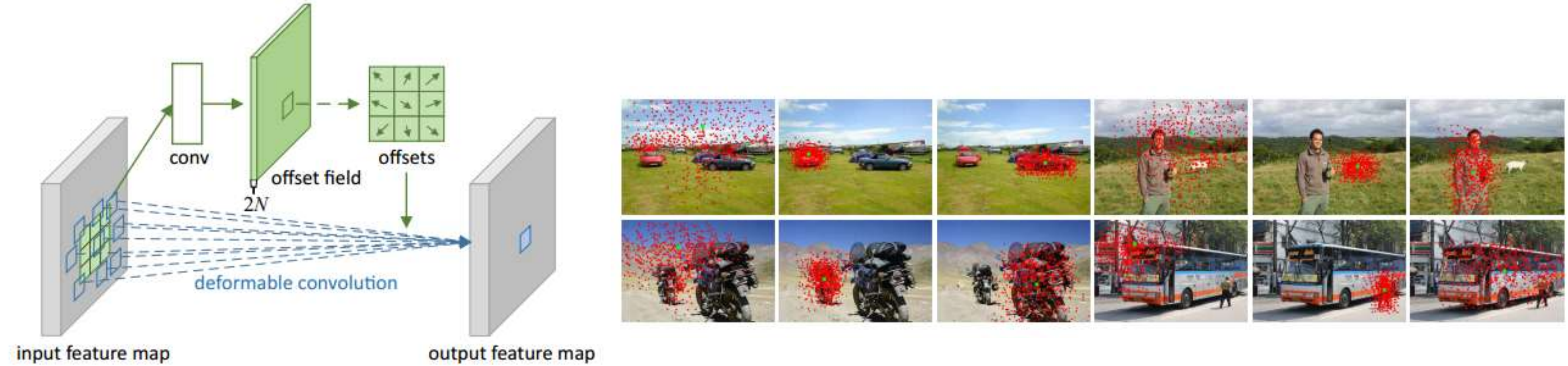
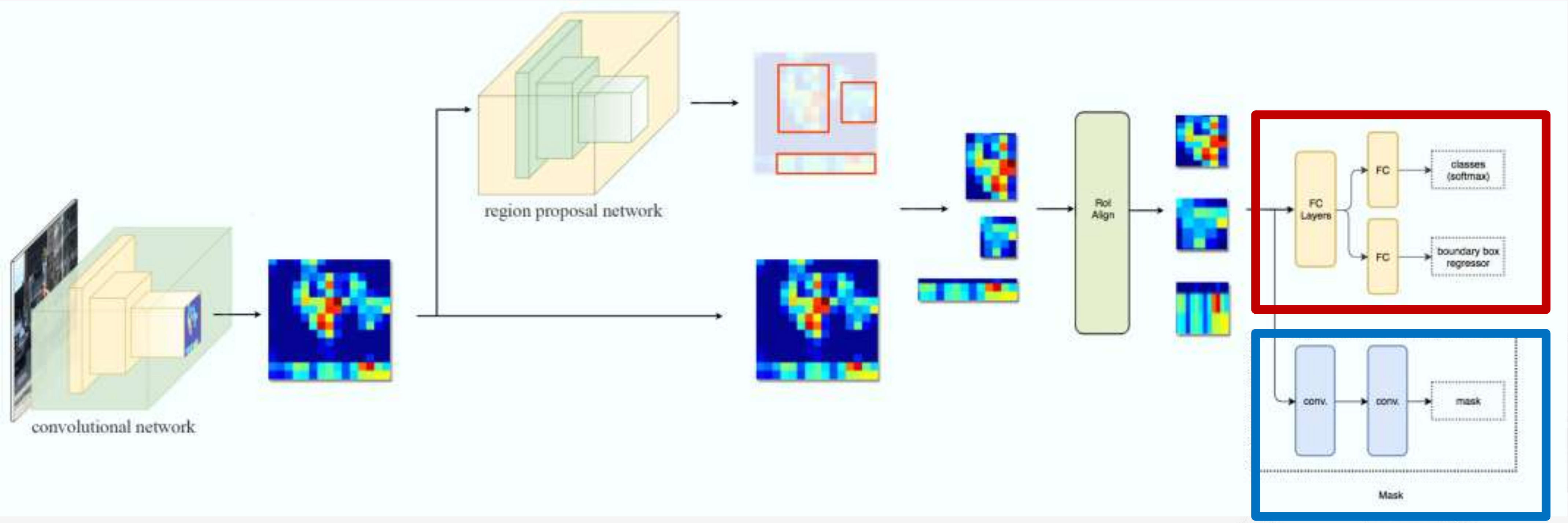


Figure 2: Illustration of 3 × 3 deformable convolution.

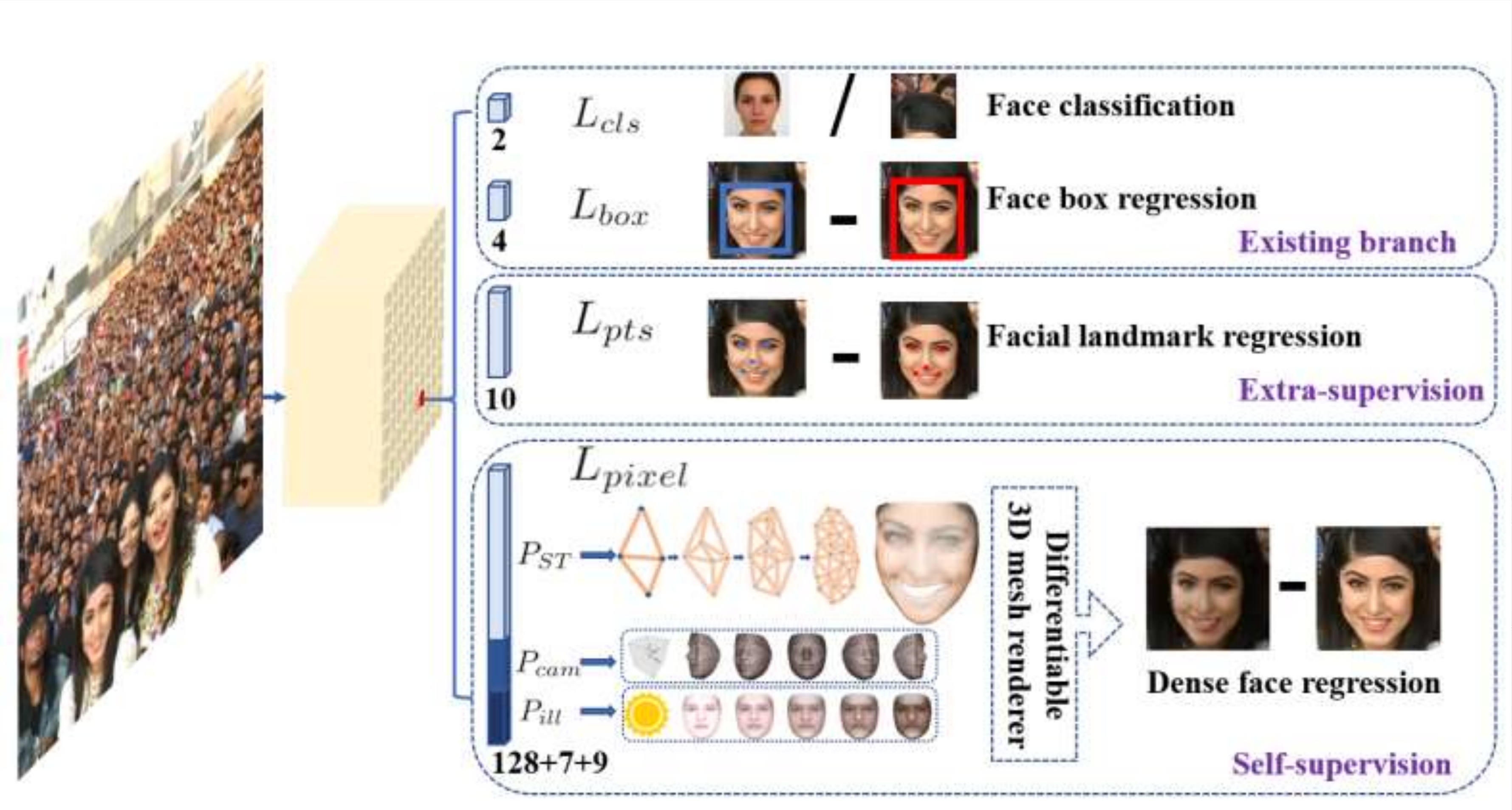
References
https://paperswithcode.com/method/deformable-convolution

Multi-task Learning



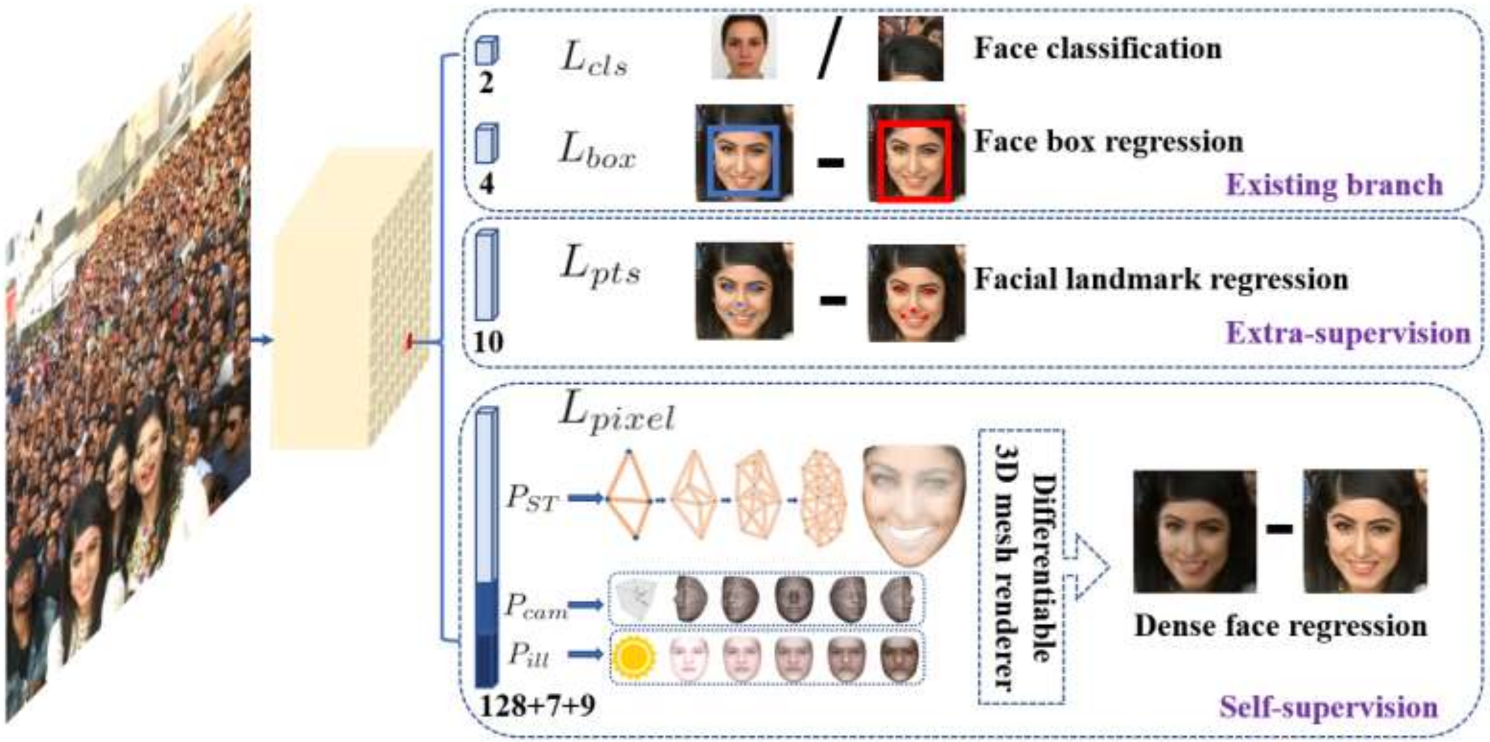
References
https://herbwood.tistory.com/20

Multi-task Learning



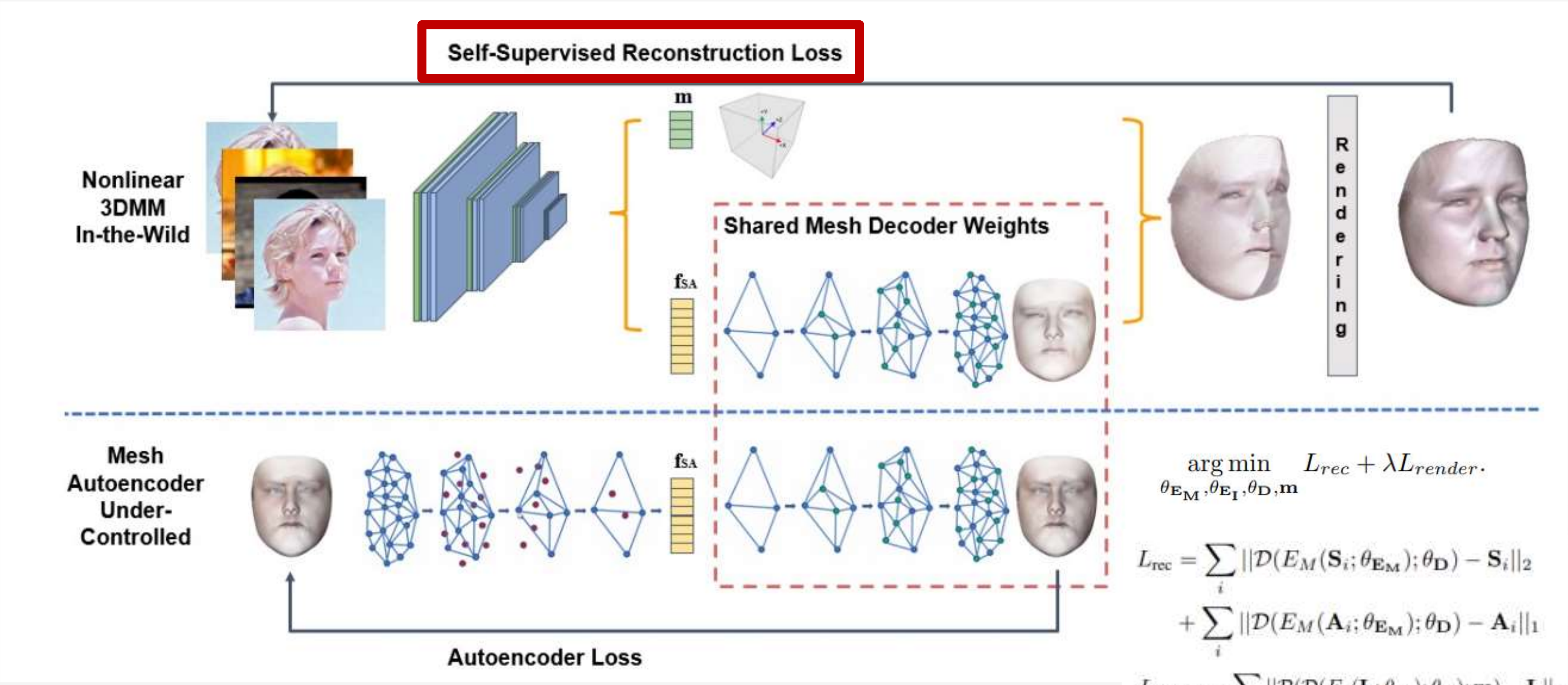
Proposed

Multi-task Loss



$$L = L_{cls}(p_i, p_i^*) + \lambda_1 p_i^* L_{box}(t_i, t_i^*) + \lambda_2 p_i^* L_{pts}(l_i, l_i^*) + \lambda_3 p_i^* L_{pixel}.$$

Dense Regression Branch



$$\underset{\theta_{E_M},\theta_{E_I},\theta_D,\mathbf{m}}{\arg\min} \quad L_{rec} + \lambda L_{render}.$$

$$L_{rec} = \sum_i ||\mathcal{D}(E_M(\mathbf{S}_i;\theta_{E_M});\theta_D) - \mathbf{S}_i||_2 + \sum_i ||\mathcal{D}(E_M(\mathbf{A}_i;\theta_{E_M});\theta_D) - \mathbf{A}_i||_1$$

$$L_{render} = \sum_i ||\mathcal{P}(\mathcal{D}(E_I(\mathbf{I}_i;\theta_{E_I});\theta_D);\mathbf{m}) - \mathbf{I}_i||_1$$

References
https://arxiv.org/pdf/1904.03525.pdf

Experimental Results

Dataset

32,203 images, 393,703 face boxes

References
http://shuoyang1213.me/WIDERFACE/WiderFace_Results.html

Dataset

Based on the detection rate of **EdgeBox**, three levels of difficulty (i.e. Easy, Medium and Hard)

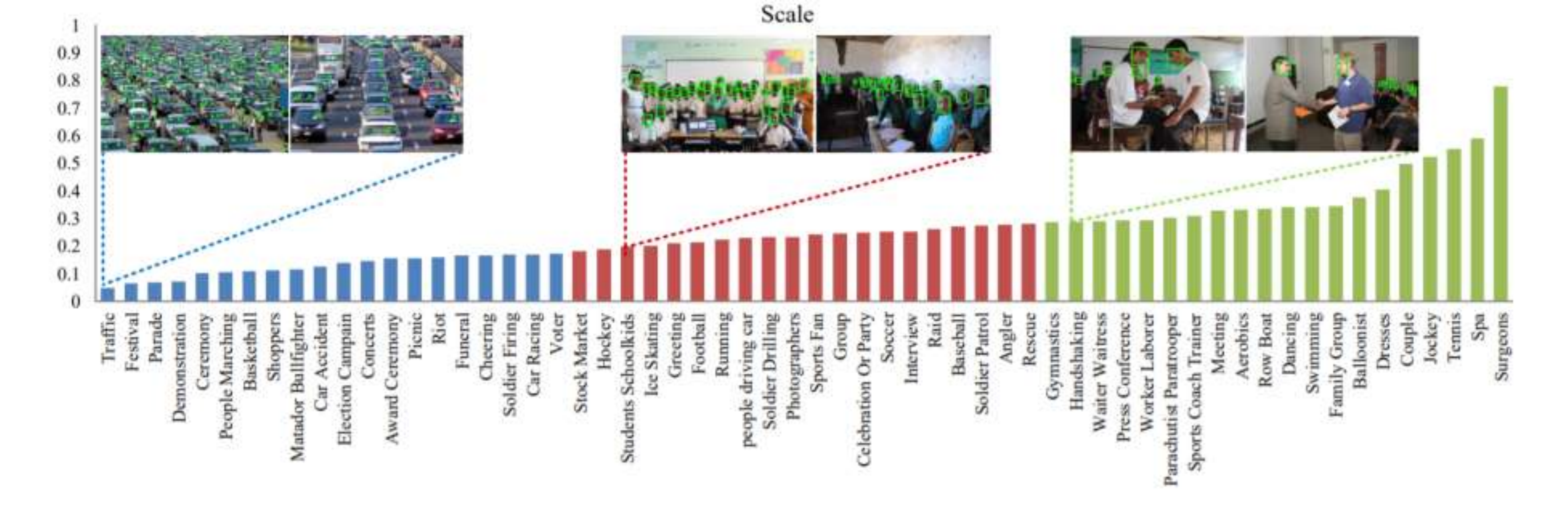


Figure 4. Histogram of detection rate for different event categories. Event categories are ranked in an ascending order based on the detection rate when the number of proposal is fixed at 10,000. Top 1 − 20, 21 − 40, 41 − 60 event categories are denoted in blue, red, and green, respectively. Example images for specific event classes are shown. Y-axis denotes for detection rate. X-axis denotes for event class name.

References
https://arxiv.org/pdf/1511.06523.pdf

Dataset – Extra annotation

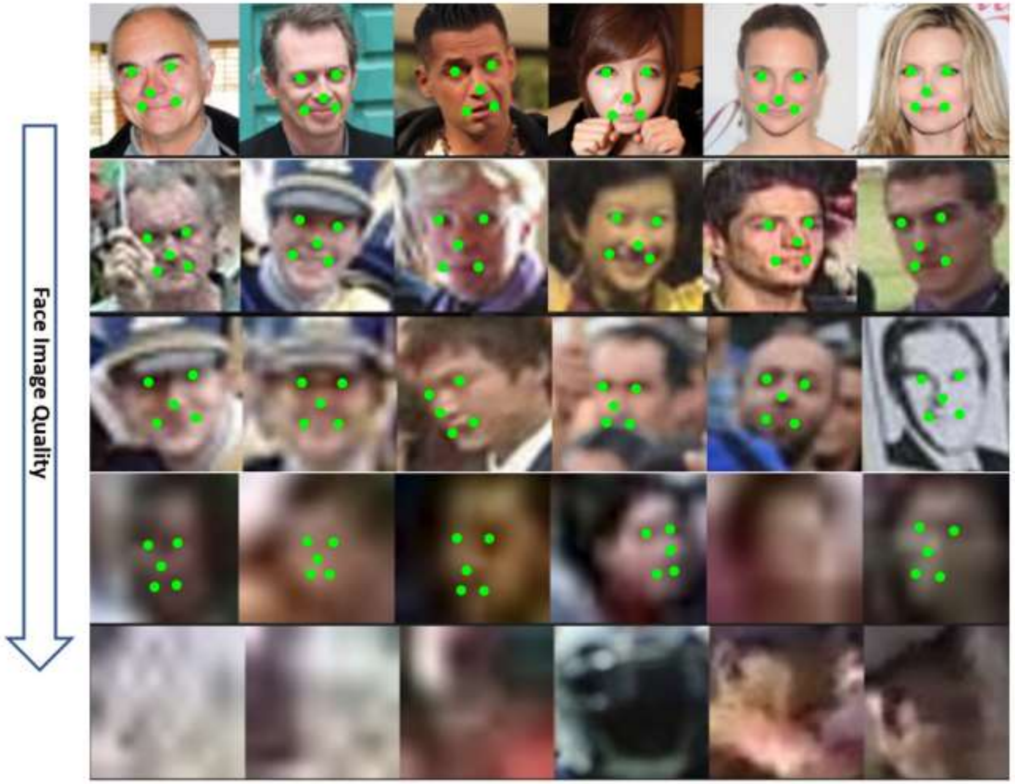


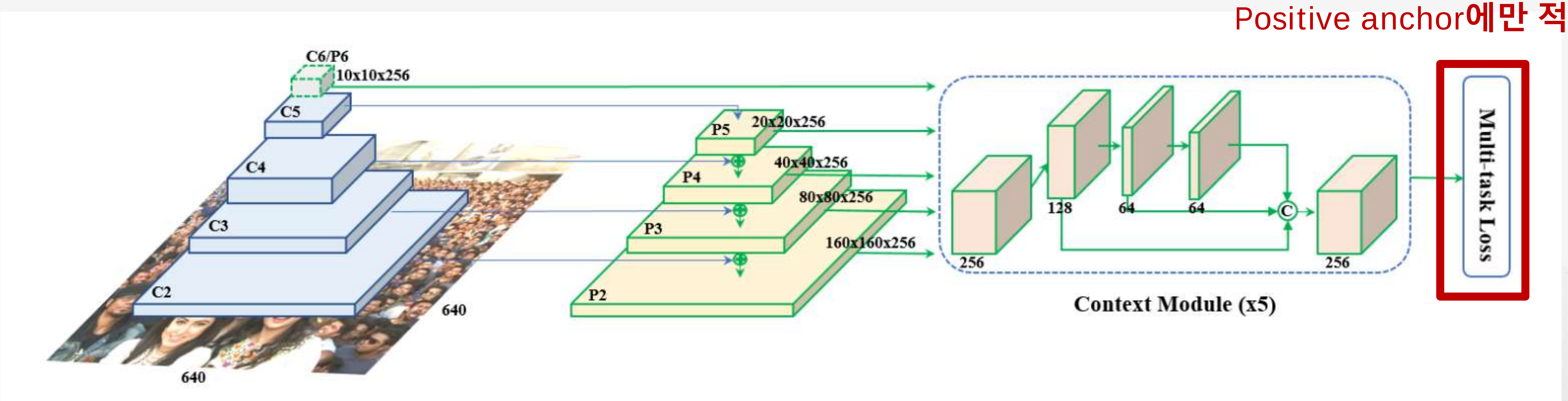
Figure 4. We add extra annotations of five facial landmarks on faces that can be annotated (we call them "annotatable") from the WIDER FACE training and validation sets.

Level	Face Number	Criterion
1	4,127	indisputable 68 landmarks [44]
2	12,636	annotatable 68 landmarks [44]
3	38,140	indisputable 5 landmarks
4	50,024	annotatable 5 landmarks
5	94,095	distinguish by context

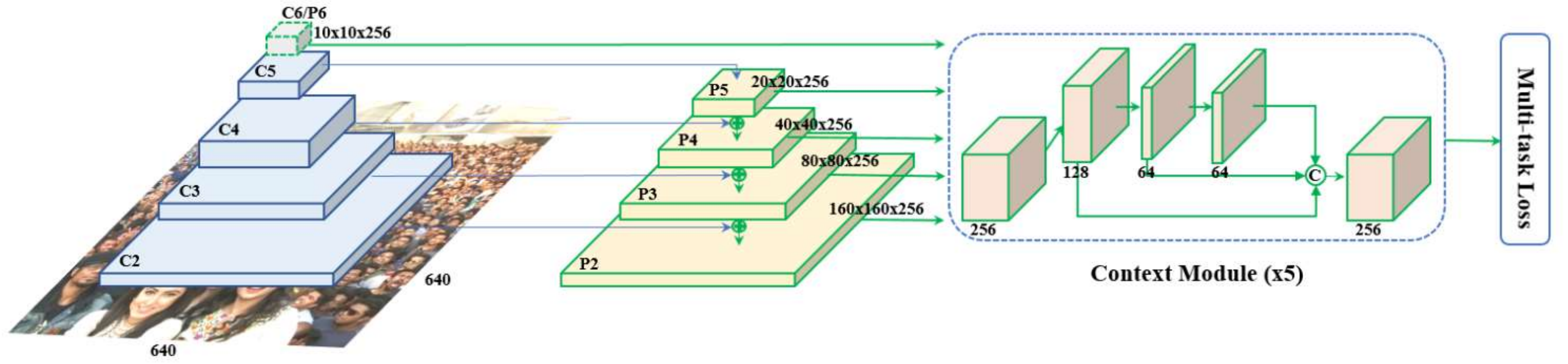
References
http://shuoyang1213.me/WIDERFACE/WiderFace_Results.html

Implementation Details

Positive anchor에만 적용



Implementation Details



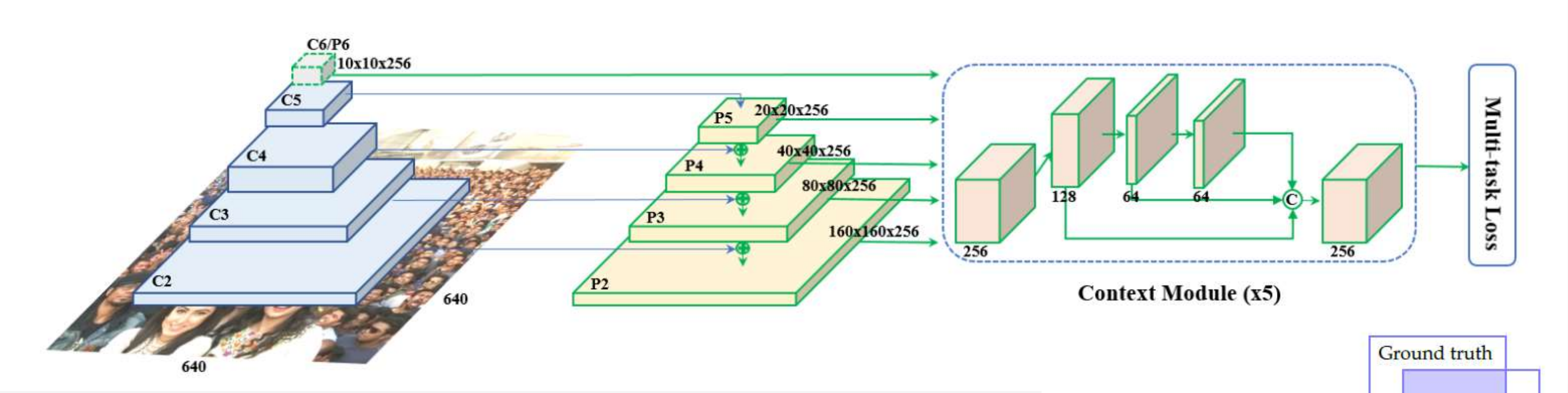
Anchor Setting

Scale step $2^{\frac{1}{3}}$ & aspect ratio 1

Ex. image size at 640 × 640,
the anchors can cover scales from 16 × 16 to 406 × 406
102,300 anchors, and **75%** of these anchors are from **P2**

Feature Pyramid	Stride	Anchor
P_2 ($160 \times 160 \times 256$)	4	16, 20.16, 25.40
P_3 ($80 \times 80 \times 256$)	8	32, 40.32, 50.80
P_4 ($40 \times 40 \times 256$)	16	64, 80.63, 101.59
P_5 ($20 \times 20 \times 256$)	32	128, 161.26, 203.19
P_6 ($10 \times 10 \times 256$)	64	256, 322.54, 406.37

Implementation Details



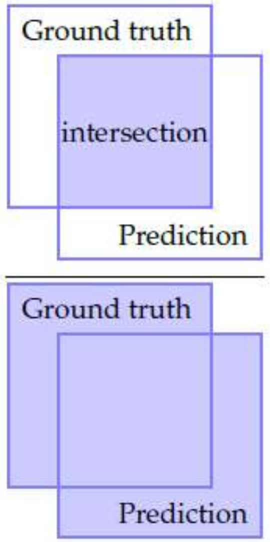
Anchor Setting

IoU (Intersection of Union)

IoU > 0.5 : anchors are matched

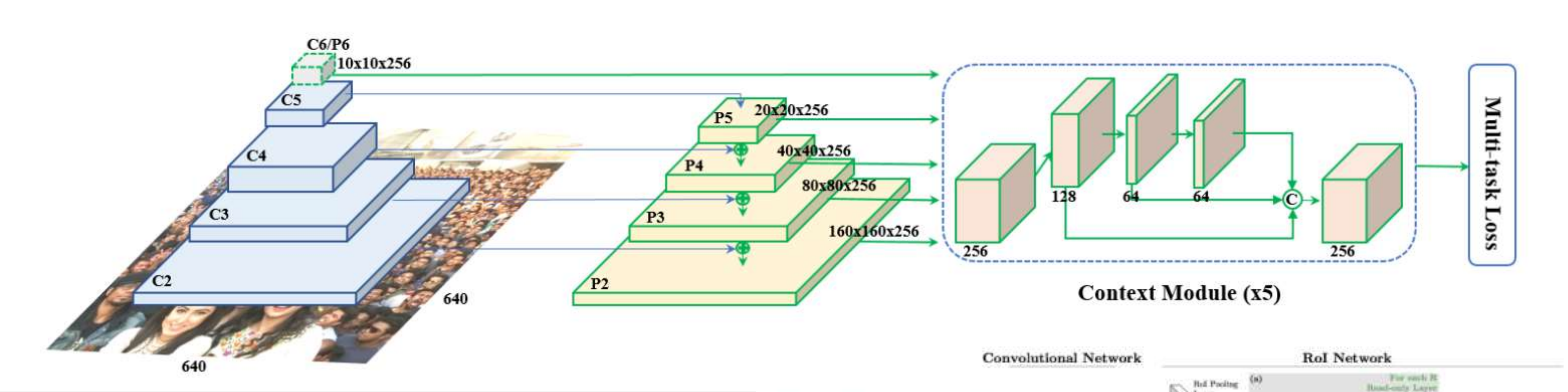
IoU < 0.3 : background (Not used training)

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} =$$

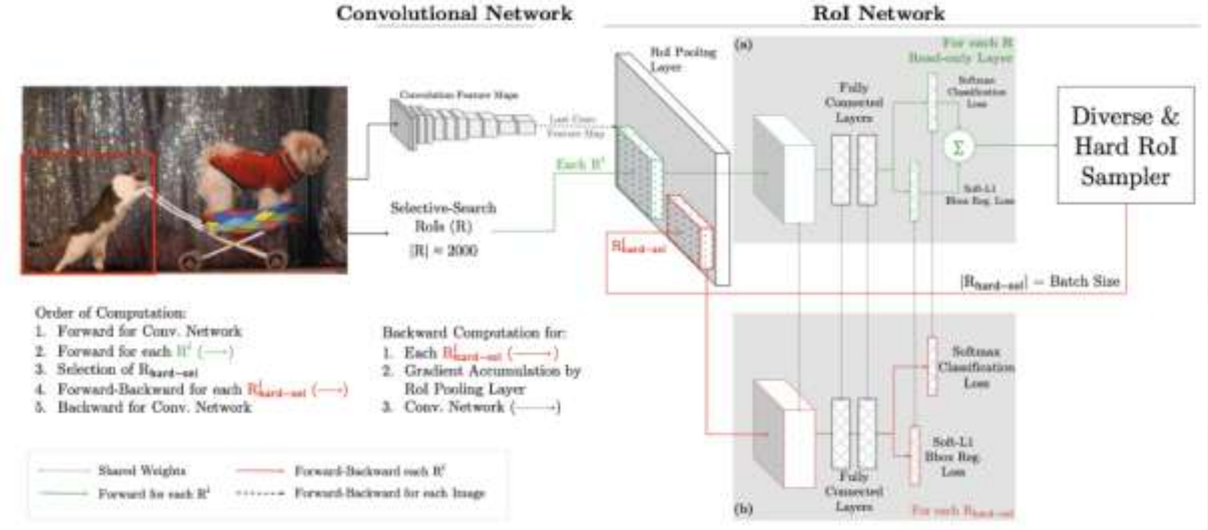


References
https://tex.stackexchange.com/questions/637812/drawing-intersection-over-union-in-equation

Implementation Details



Anchor Setting
OHEM (Online Hard Example Mining)
Positive : Negative = 3:1



References
https://arxiv.org/pdf/1604.03540.pdf

Implementation Details

Data augmentation

* Random crop
* Horizontal flip
* Photo-metric color distortion

Training

- Optimizer : SGD
- Momentum : 0.9
- Weight decay :5e-4

- Batch size : 8 x 4
- Learning rate : 0.001
- Epochs : 80

- GPU : NVIDIA Tesla P40 (24GB) * 4

Testing

- Flip
- Multi-scale :
  [500, 800, 1100, 1400, 1700]
- IoU : 0.4

Ablation Study

Method	Easy	Medium	Hard	mAP [33]
FPN+Context	95.532	95.134	90.714	50.842
+DCN	96.349	95.833	91.286	51.522
$+L_{pts}$	96.467	96.075	91.694	52.297
$+L_{pixel}$	96.413	95.864	91.276	51.492
$+L_{pts} + L_{pixel}$	**96.942**	**96.175**	**91.857**	**52.318**

Verification Performance (%)

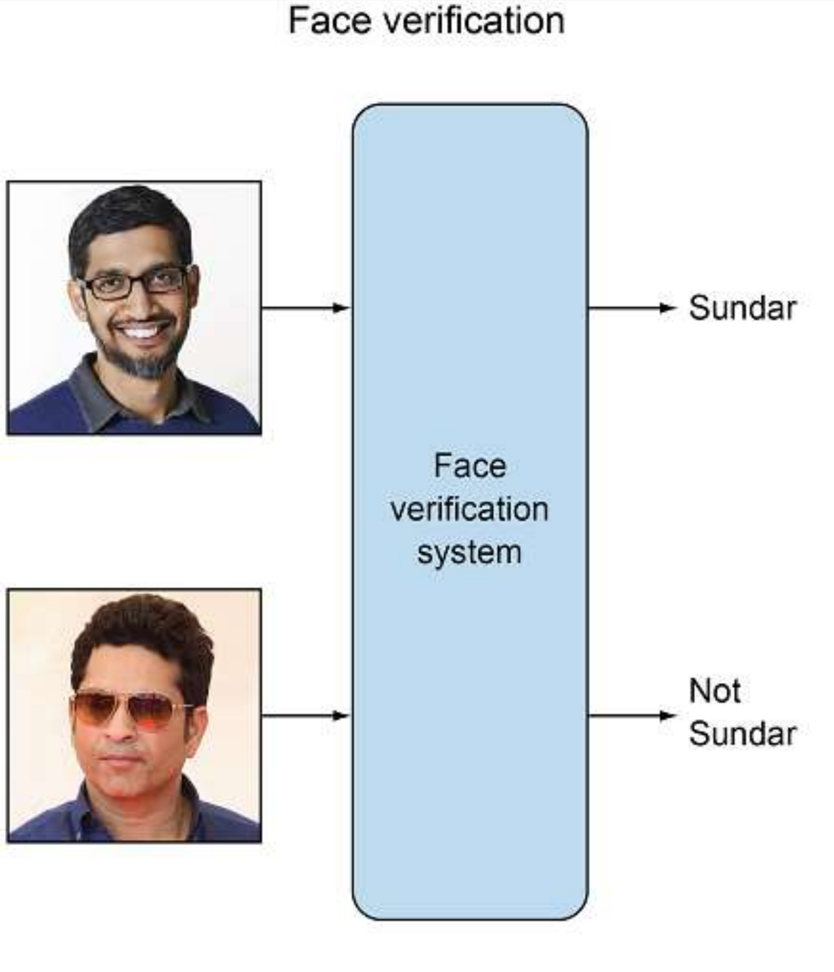
Methods	LFW	CFP-FP	AgeDB-30
MTCNN+ArcFace [11]	99.83	98.37	98.15
RetinaFace+ArcFace	**99.86**	**99.49**	**98.60**

Table 4. Verification performance (%) of different methods on LFW, CFP-FP and AgeDB-30.

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	-

Table 5. Inference time (ms) of RetinaFace with different backbones (ResNet-152 and MobileNet-0.25) on different input sizes (VGA@640x480, HD@1920x1080 and 4K@4096x2160). "CPU-1" and "CPU-m" denote single-thread and multi-thread test on the Intel i7-6700K CPU, respectively. "GPU" refers to the NVIDIA Tesla P40 GPU and "ARM" platform is RK3399(A72x2).

Face Verification



References
http://lacienciadelcafe.com.ar/kids-jbl-headphones/parka-arm%C3%A9e-de-l//iproov-on-twitter-what-s-the-difference-between-face-pp-24027720

Verification Performance (%)

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

Table 4. Verification performance (%) of different methods on LFW, CFP-FP and AgeDB-30.

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	-

Table 5. Inference time (ms) of RetinaFace with different backbones (ResNet-152 and MobileNet-0.25) on different input sizes (VGA@640x480, HD@1920x1080 and 4K@4096x2160). “CPU-1” and “CPU-m” denote single-thread and multi-thread test on the Intel i7-6700K CPU, respectively. “GPU” refers to the NVIDIA Tesla P40 GPU and “ARM” platform is RK3399(A72x2).

Conclusion

Conclusion

- *We manually annotate **five facial landmarks** on the WIDER FACE dataset and observe significant improvement in hard face detection with the assistance of this extra supervision signal.*
- *We further add a **self-supervised** mesh decoder branch for predicting a pixel-wise 3D shape face information in parallel with the existing supervised branches.*
- *On the WIDER FACE hard test set, RetinaFace outperforms the **state of the art** average precision (AP) by 1.1% (achieving AP equal to 91.4%).*
- *On the IJB-C test set, RetinaFace enables state of the art methods (ArcFace) **to improve** their results in **face verification** (TAR=89.59% for FAR=1e-6).*
- *By employing light-weight backbone networks, RetinaFace can run real-time on a single CPU core for a VGA-resolution image.*

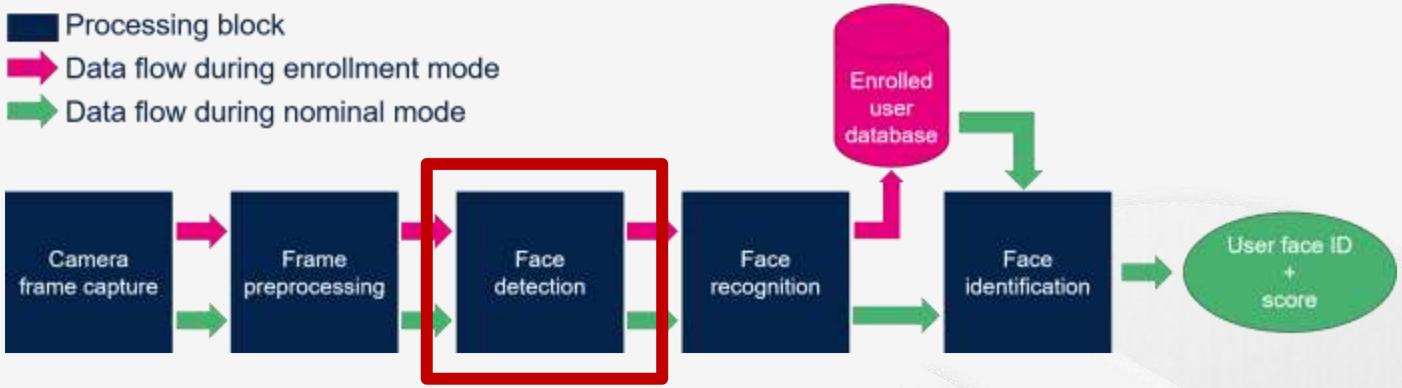
[Practice 1] Face Detection

CONTENT

실습 소개

Face Detection이란?

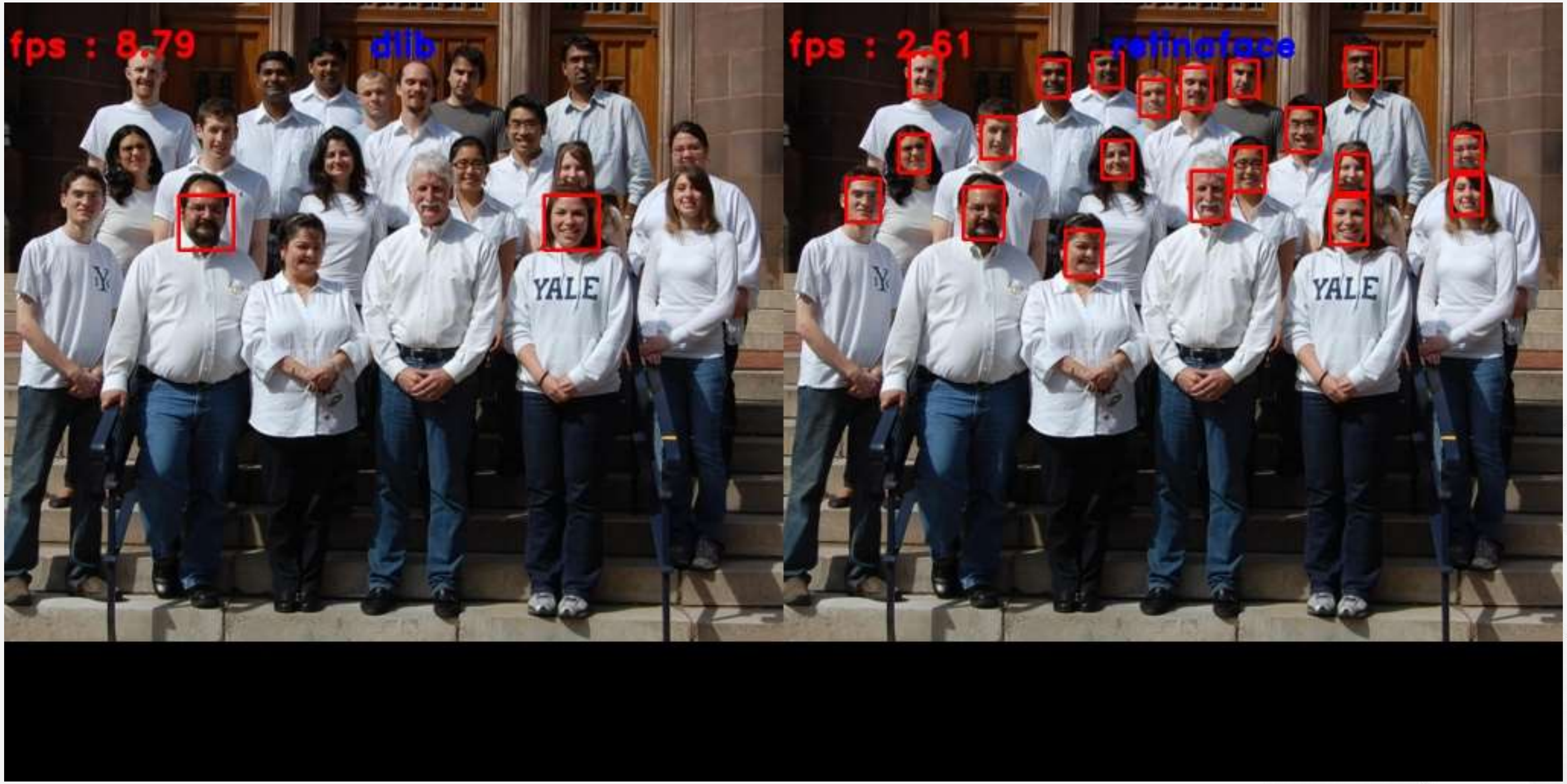
The most basic task on Face Recognition is of course, “Face Detecting”. Before anything, you must “capture” a face in order to recognize it, when compared with a new face captured on future.



References
https://wiki.st.com/stm32mpu/wiki/TFLite_Cpp_face_recognition

[실습1] Face Detection

1) Dlib 라이브러리 2) RetinaFace 모델 이용하여 얼굴 검출하기



데이터셋

데이터셋 소개

References
http://shuoyang1213.me/WIDERFACE/WiderFace_Results.html

데이터셋 소개

Paper : http://shuoyang1213.me/WIDERFACE/support/paper.pdf'

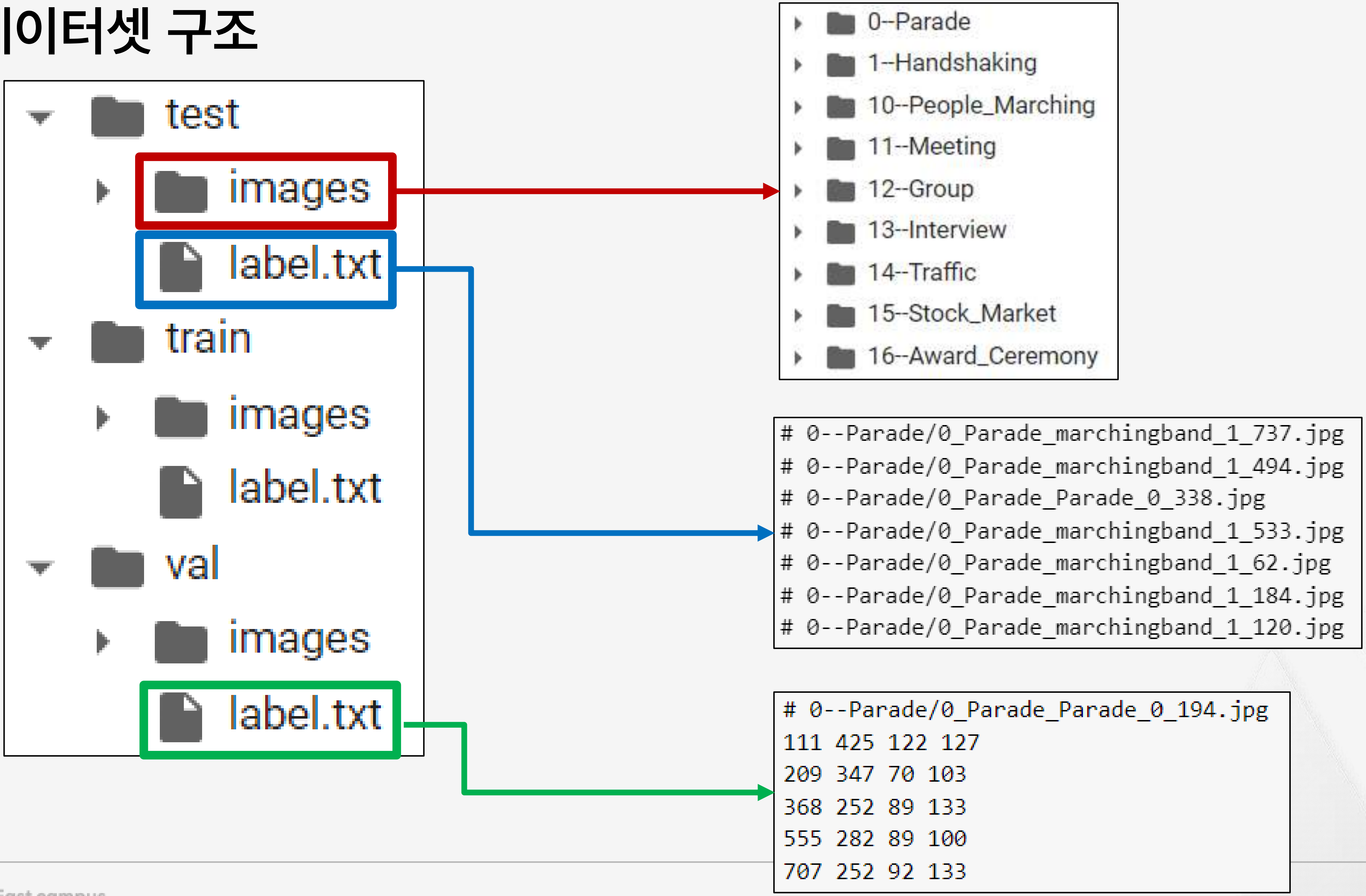
얼굴 검출 벤치마크 데이터셋

32,203개의 이미지, 393,703개의 얼굴

항목	이미지 수
Train	12880
Validation	3226
test	16097

References
https://www.tensorflow.org/datasets/catalog/wider_face?hl=ko

데이터셋 구조


test
images
label.txt
train
images
label.txt
val
images
label.txt
0--Parade
1--Handshaking
10--People_Marching
11--Meeting
12--Group
13--Interview
14--Traffic
15--Stock_Market
16--Award_Ceremony
0--Parade/0_Parade_marchingband_1_737.jpg
0--Parade/0_Parade_marchingband_1_494.jpg
0--Parade/0_Parade_Parade_0_338.jpg
0--Parade/0_Parade_marchingband_1_533.jpg
0--Parade/0_Parade_marchingband_1_62.jpg
0--Parade/0_Parade_marchingband_1_184.jpg
0--Parade/0_Parade_marchingband_1_120.jpg
0--Parade/0_Parade_Parade_0_194.jpg
111 425 122 127
209 347 70 103
368 252 89 133
555 282 89 100
707 252 92 133

실습 환경 설정

Dlib 실습 준비

Dlib 이란

- 오픈 소스 라이브러리
- Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems.
- Python 3.8부터는 pip가 아닌 다른 방법으로 설치 필요

Dlib 설치

- ~ python 3.7
  pip install dlib
- Python 3.8 ~
  git install

References
http://dlib.net/

RetinaFace

Git clone https://github.com/biubug6/Pytorch_Retinaface.git

pip install torch==1.7.1+cu110 torchvision==0.8.2+cu110 -f https://download.pytorch.org/whl/torch_stable.html

실습 튜토리얼

Dlib을 이용한 Face Detection 코드

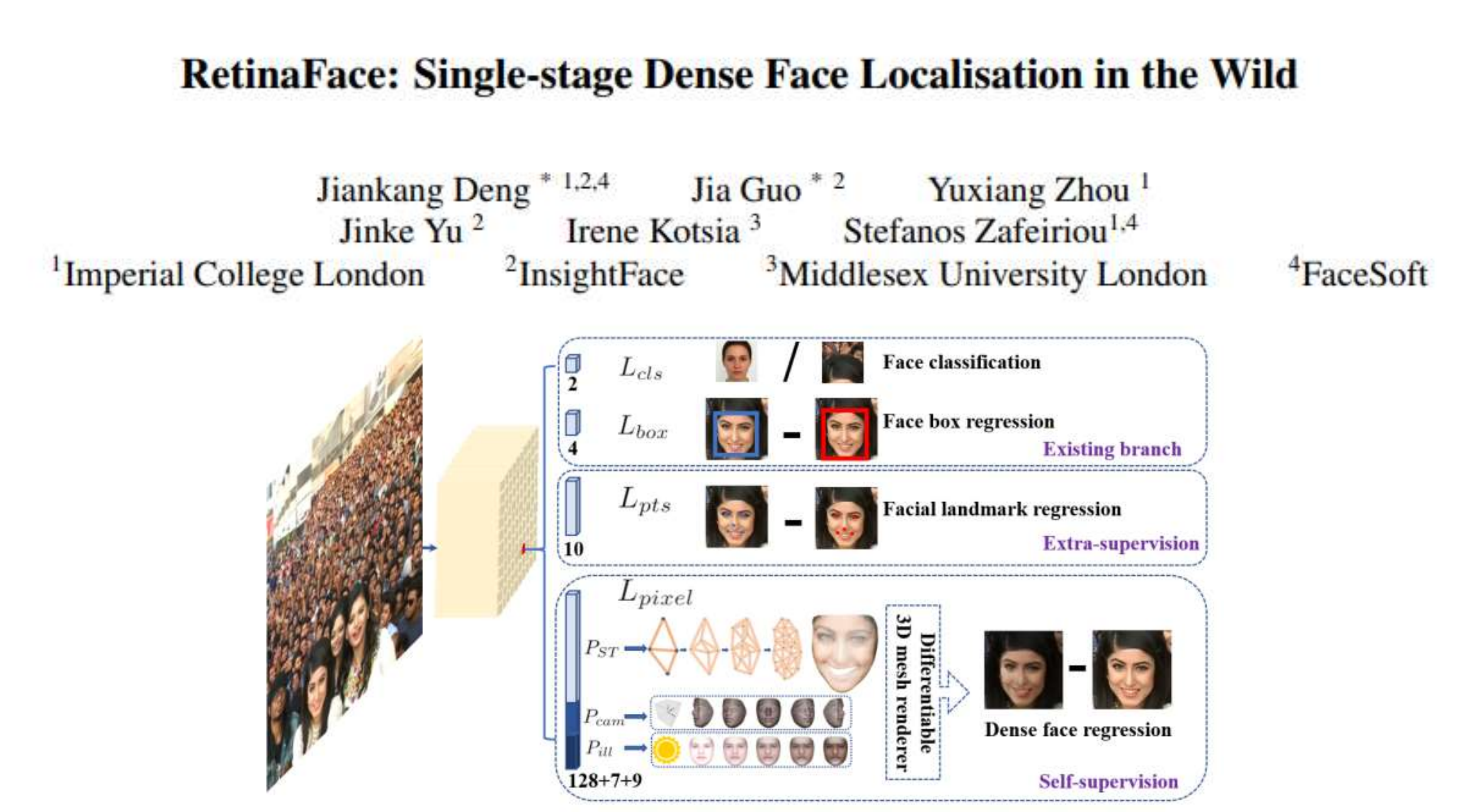
```
import dlib
face_detector = dlib.get_frontal_face_detector()

test_img = cv2.imread(test_path)
img = np.float32(test_img)
face_detection = face_detector(test_img)

for f in face_detection:
    cv2.rectangle(test_img, (f.left(), f.top()), (f.right(), f.bottom()), (255,0,0), 2)
```

RetinaFace

논문 : https://arxiv.org/abs/1905.00641

공식 Github : https://github.com/deepinsight/insightface/tree/master/detection/retinaface

Pytorch Github : https://github.com/biubug6/Pytorch_Retinaface



RetinaFace: Single-stage Dense Face Localisation in the Wild

Jiankang Deng [*1,2,4] Jia Guo [*2] Yuxiang Zhou [1]
Jinke Yu [2] Irene Kotsia [3] Stefanos Zafeiriou [1,4]
[1]Imperial College London [2]InsightFace [3]Middlesex University London [4]FaceSoft

RetinaFace

Training

!CUDA_VISIBLE_DEVICES=0 python train.py --network mobile0.25 --training_dataset /content/drive/MyDrive/dataset/face_detection/train/label.txt

Inference

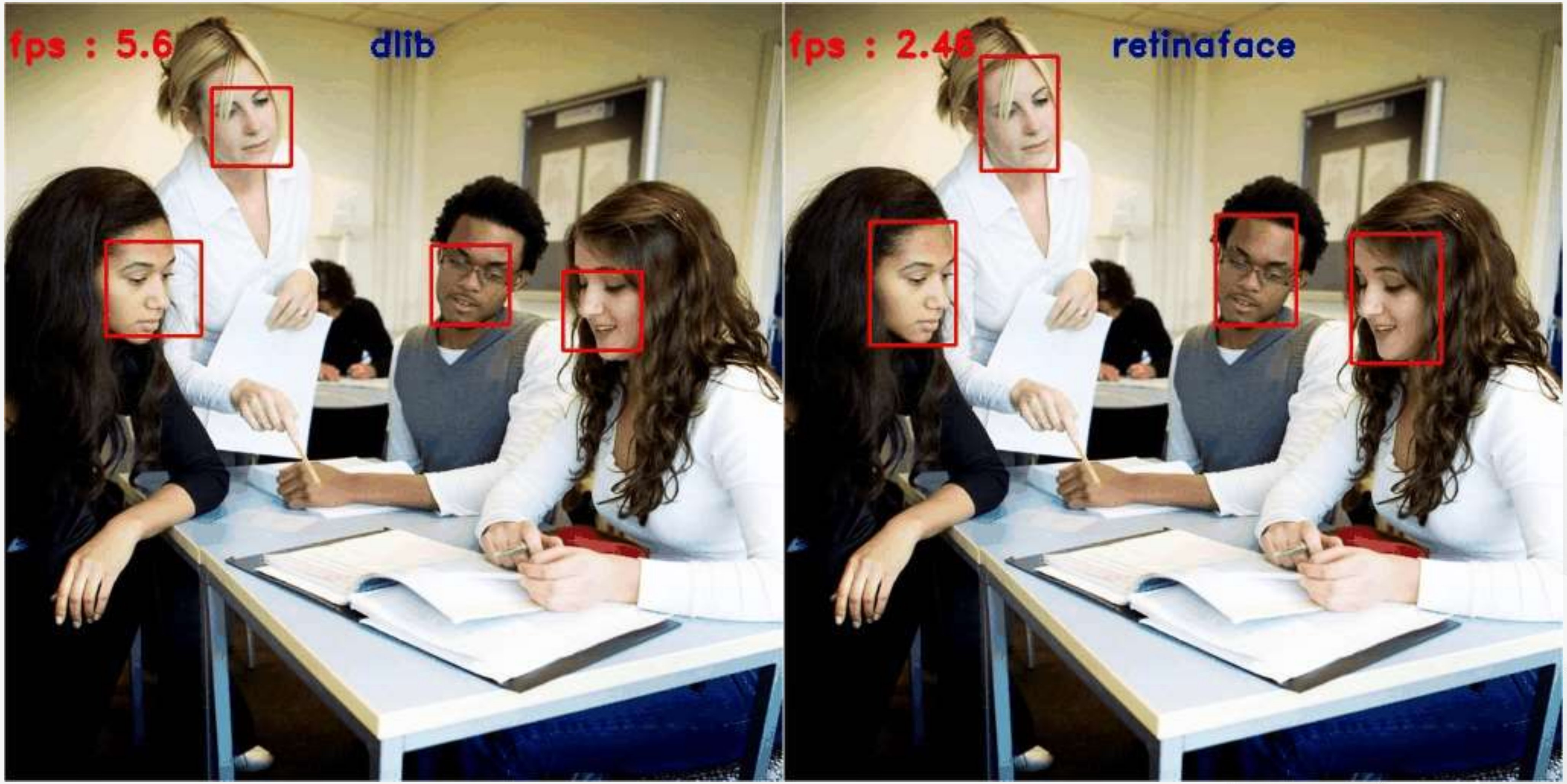
!python detect.py –m modelPath –cpu –s

87번째 줄 image_path 변경 필요

References

실습 결과

실행 결과



Thank You.