OpenForensics: Large-Scale Challenging Dataset For Multi-Face Forgery Detection And Segmentation In-The-Wild

Supplementary Material

https://sites.google.com/view/ltnghia/research/openforensics/

Trung-Nghia Le*[1], Huy H. Nguyen[2], Junichi Yamagishi[1,2], and Isao Echizen[1,2,3]

[1]National Institute of Informatics, [2]The Graduate University for Advanced Studies (SOKENDAI), [3]University of Tokyo

In this supplemental document, we provide the following contents, which were not presented in the main paper due to space limitations:

- Detailed implementation of proposed forgery synthesis workflow.
- Answers for the quiz in the main paper.
- Additional visualization of OpenForensics dataset.
- Additional dataset analysis.
- Additional user study results.
- Additional benchmark results.

1. Implementation of Forgery Synthesis Workflow

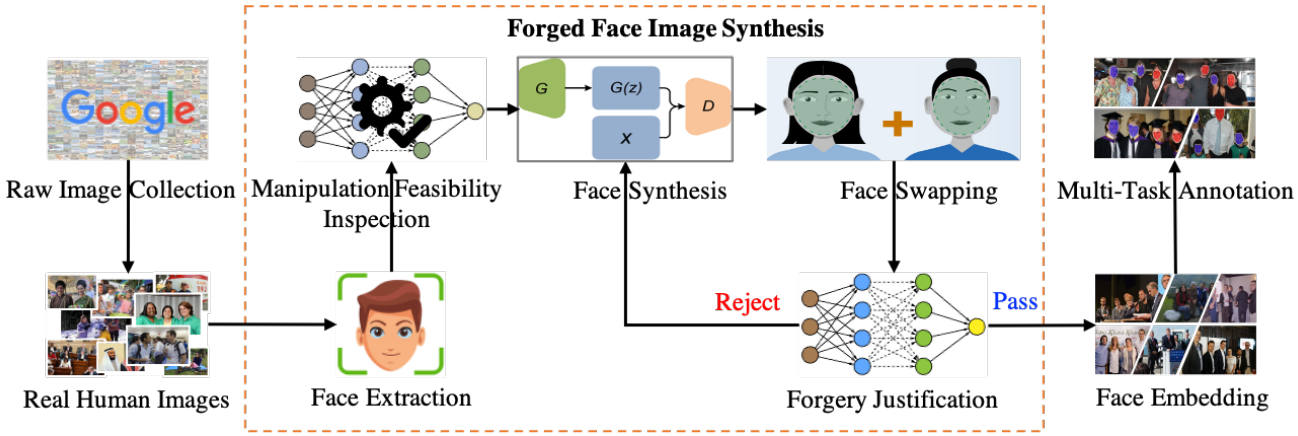


Figure 1. Dataset construction workflow: 1) collect raw images and manually select real face images; 2) synthesize forged face images (for each original extracted face, new identities are repeatedly generated until swapped faces can spoof our simple classifier); 3) perform face-wise multi-task annotation.

Figure 1 shows an overview of the process used to synthesize forged face images. First, all faces in the images are extracted using pre-trained FaceBoxes [32]. Next, for each feasibly manipulated face, we extract its identity latent vector. The facial identity is then modified and fed into fake identity generators to generate a new face. The synthesized face is subsequently transformed into the original pose via Homography transformation[1] and blended into the original face using

*Corresponding author. Email:ltnghia@nii.ac.jp. This work was partially supported by JSPS KAKENHI Grants (JP16H06302, JP18H04120, JP21H04907, JP20K23355, JP21K18023), and by JST CREST Grants (JPMJCR18A6, JPMJCR20D3), Japan.

[1]https://en.wikipedia.org/wiki/Homography_(computer_vision)

Poisson blending [19] and a color adaptation algorithm, with the final result being a new identity. After that, if the new identity image successfully spoofs the forgery justification network, it is overlaid onto the original image; otherwise, it is discarded.

1.1. Manipulable Area Identification

We introduce two types of manipulable areas: regions inside facial landmarks or entire face.

Facial Landmark Detection: To identify feasible manipulation regions, we extract 68 facial landmark points, resulting in fine boundaries and complete facial coverage. 3D facial landmarks are useful for presenting the face under different poses and occlusions, but they do not accurately present face boundaries. Hence, we propose to combine both 3D and 2D facial landmarks to present human poses. In particular, we employed 3DDFA-V2 [9], which was trained on 300W-LP dataset [35], to extract inside 3D facial landmarks, and Dlib [14], which was trained on iBUG300-W dataset [23], to extract 2D landmarks across face boundary. We used pre-trained models provided by the authors.

Face Segmentation: We adopted Bilateral Segmentation Network (BiSeNet) [30] to segmentation face regions[2]. The BiSeNet model was trained on CelebAMask-HQ dataset [15] for 80,000 iterations and warmup of 1000 steps with batch size of 16. Stochastic Gradient Descent (SGD) optimization was used with a weight decay of 0.0005 and momentum of 0.9. The base learning rate was initialized as 0.01 and reduced via the poly learning rate strategy with power 0.9 [15]. We remark that we combined all facial elements such as eyes, nose, mouth, skin to obtain the face region.

1.2. Fake-Identity Generators

To synthesize faces with high resolution (*i.e.*, 512×512 or 1024×1024 pixels) and visual quality, we employed InterFaceGAN [24] (*i.e.*, GAN-based generator) and ALAE [20] (*i.e.*, Autoencoder-based generator). We used pre-trained models provided by the authors, in which InterFaceGAN was trained on CelebA-HQ dataset [12], and ALAE was trained on FFHQ dataset [13]. Given a face image, we modified and reconstructed it with a new identity using these generators randomly. In particular, we first extracted the facial identity latent vector via the encoder of the network. Small random factors were adaptively multiplied with the latent vector to edit its attributes (*i.e.*, age, gender, and smile). The manipulated latent vector was then fed into the decoder to reconstruct the face with a new fake identity.

1.3. Forgery Justification Network

To control the quality of new identity images, we trained a simple forgery classifier (*i.e.*, XceptionNet [6]) to reject low-quality fake images. We remark that we do not train a strong network to avoid rejecting all generated images. We build a pseudo database with 50,000 images by pasting synthesized faces into the original images without any visual improvement (*i.e.*, Poisson blending [19] and color adaptation) as the fake label. We used original faces as the real label. XceptionNet was trained from scratch with a base learning rate of 0.0001 on DFDC dataset [7] for 10 epochs and then finetuned on our pseudo database for only 2 epochs. We set the size of each mini-batch to 64 and employed binary cross entropy loss. Adam optimization was used with moments $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also applied simple augmentations such as resizing, cropping, translation, rotation, flipping.

2. OpenForensics Dataset

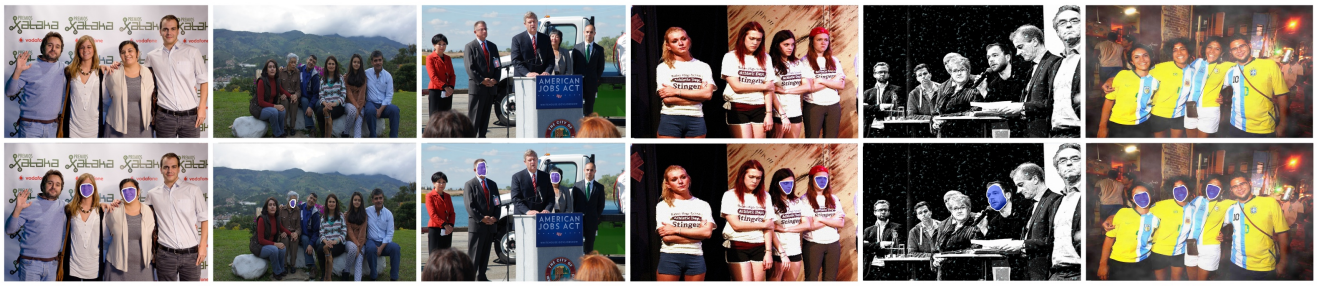


Figure 2. Answers to question posed in Fig. 1 in main paper showing overlaid manipulated areas (best viewed online in color with zoom-in).

[2] https://github.com/zllrunning/face-parsing.PyTorch

Figure 3. Additional examples of standard images in OpenForensics dataset with overlaid ground truths.

2.1. Face Blending

Our use of Poisson blending [19] and a color adaptation algorithm to reduce the color mismatch between the synthesized and the original face enhances the naturalness of the forged faces (Fig. 4).

Figure 4. From top to bottom: Original faces (top), forged faces without (middle) and with (bottom) Poison blending and color adaptation. Note the reduced color mismatch between the synthesized and non-synthesized face regions. This blending method with color correction is used for all forged faces in the OpenForensics dataset.

We also improve the smoothness of the blending mask by extracting 68 facial landmark points and training face segmentation models, resulting in fine boundaries and complete facial coverage (see Fig. 5 for different blending masks).

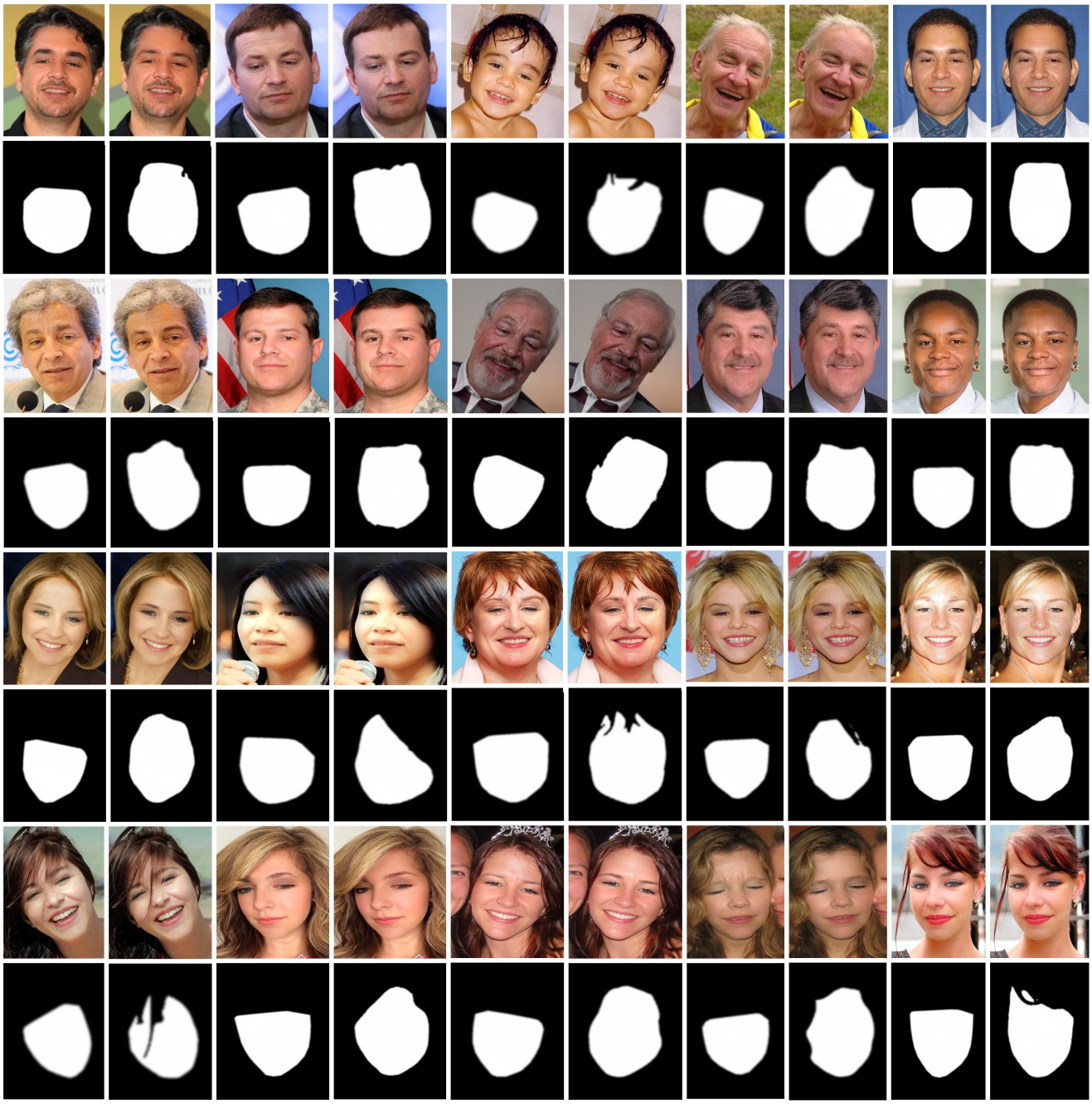


Figure 5. From left to right: blending masks in OpenForensics smoothly cover important facial parts inside facial landmarks with soft boundary (left) or completely cover the entire face (right).

2.2. Face-Wise Rich Annotation

We aim to exploit the face-wise ground truth, which requires much more annotation effort, to advance further forgery analysis. Each face was labeled with various ground truths such as forgery category (real/fake), bounding box, segmentation mask, forgery boundary, and facial landmarks (cf. Fig. 6). Our rich annotation can be utilized for various tasks and even multi-task learning.

Figure 6. Face-wise multi-task ground truth in OpenForensics dataset (best viewed online in color with zoom-in). From left to right, image is followed by overlaid ground truth bounding box and segmentation mask, forgery boundary, and general facial landmarks.

2.3. Scenario Augmentation

To enhance the challenges posed by our OpenForensics dataset for real-world face forgery detection and segmentation, we applied various perturbations to better simulate contexts in natural scenes, resulting in a test-challenge subset. Various augmented operators are divided into overarching groups.

- Color manipulation: Hue change, saturation change, brightness change, histogram adjustment, contrast addition, grayscale conversion.
- Edge manipulation: edge detection and alteration.
- Block-wise distortion: color grouping, color pooling, color quantization, and pixelation.
- Image corruption: elastic deformation, jigsaw distortion, JPEG compression, noise addition, and dropout.
- Convolution mask transformation: Gaussian blurring, motion blurring, sharpening, and embossing.
- External effect: fog, cloud, sun, frost, snow, rain, and spatter.

These augmentations are divided into three intensity levels (*i.e.*, easy, medium, and hard) to ensure diverse scenarios. For each level, random-type augmentation is applied separately or as a mixture (cf. Fig. 7). Example images in test-challenge set are shown in Fig. 8.

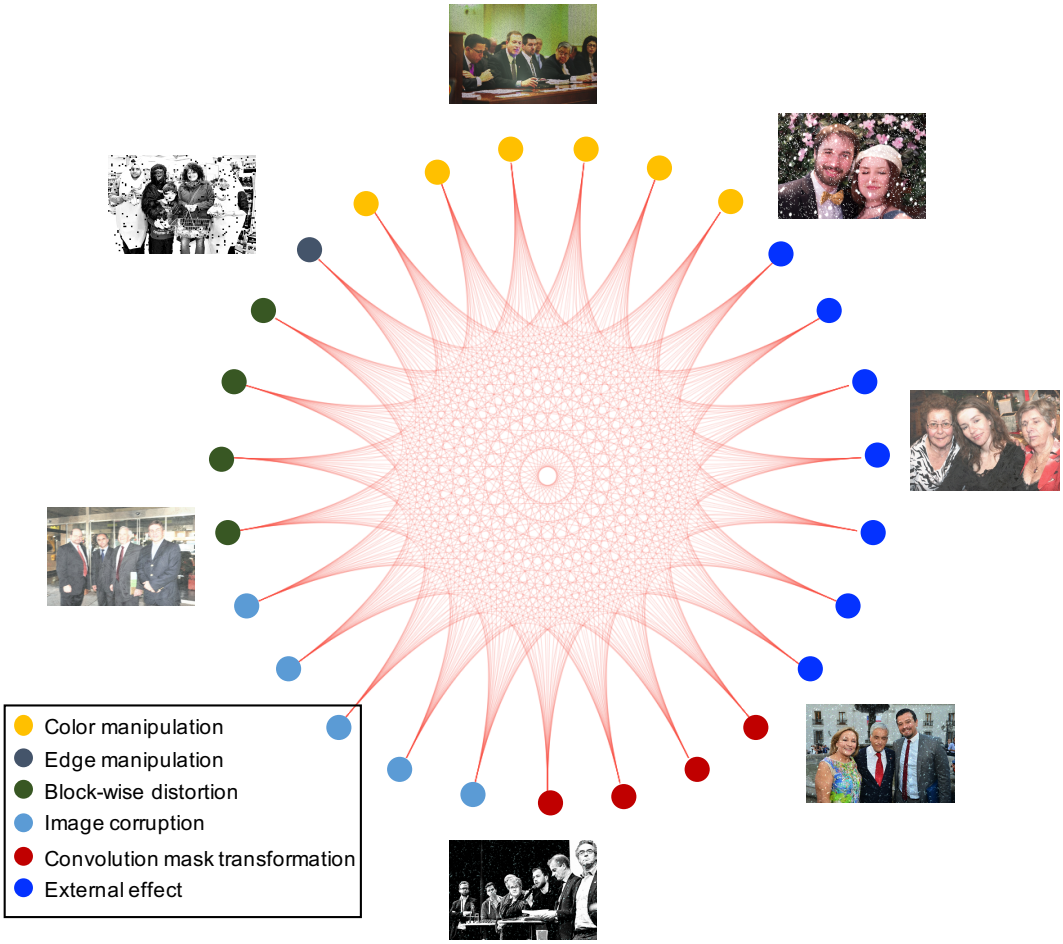


Figure 7. Six-group augmentations used to generate test-challenge subset. Different augmented operators are mixed randomly with various parameters to increase diversity of scenarios.

Figure 8. Examples of test-challenge set with overlaid manipulated areas.

3. Additional Dataset Analysis

3.1. Image Scene

Images and videos in existing datasets [22, 17] have been collected from a limited number of scenes, such as indoor scenes and TV shows. In contrast, the OpenForensics dataset contains images from a wide variety of scenes. We computed image scenes using a model pre-trained on the large-scale Places2 dataset [33]. Figure 9 shows the distribution of image scenes in the OpenForensics dataset, of which 36.3% are outdoor scenes.

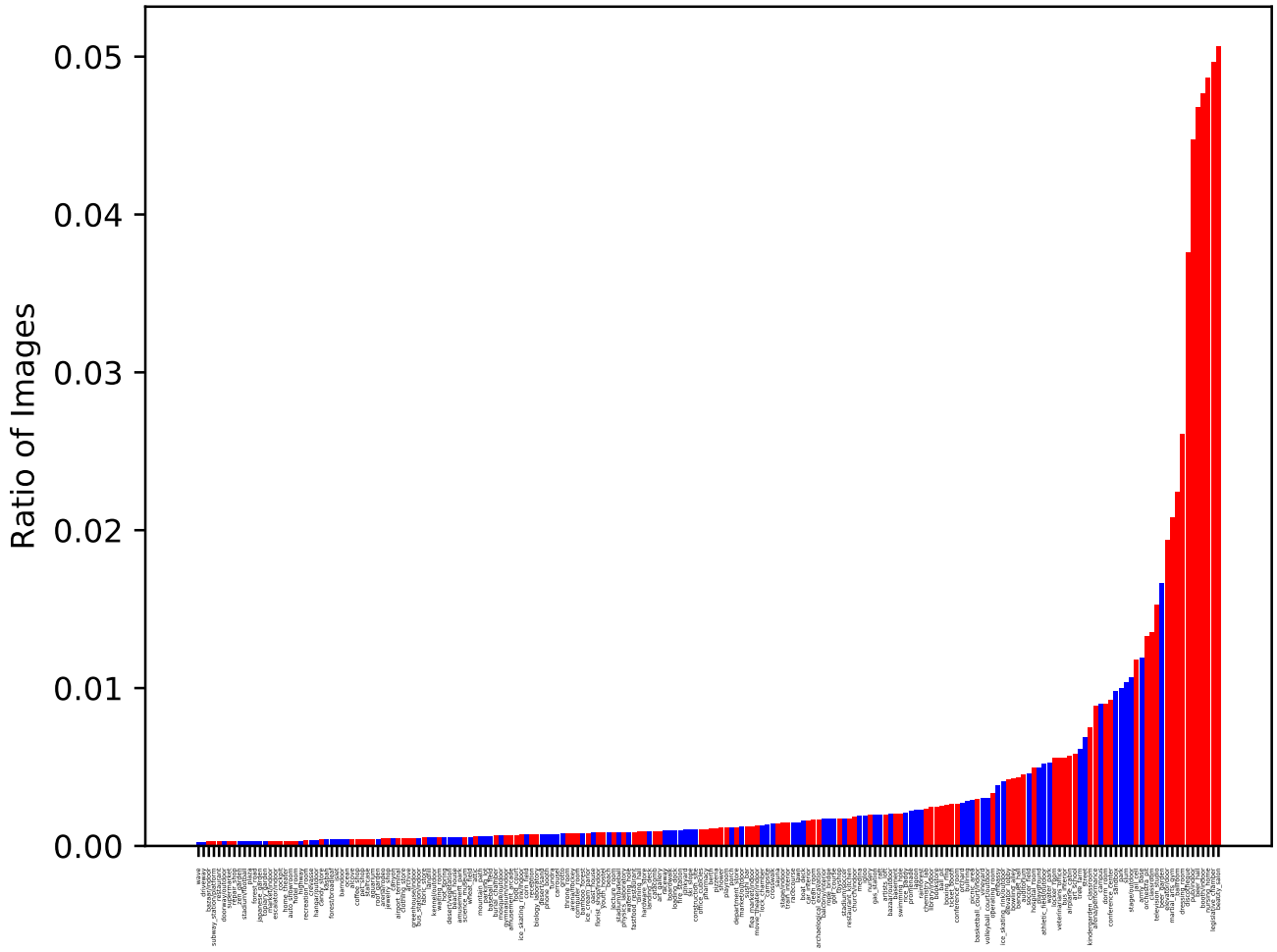
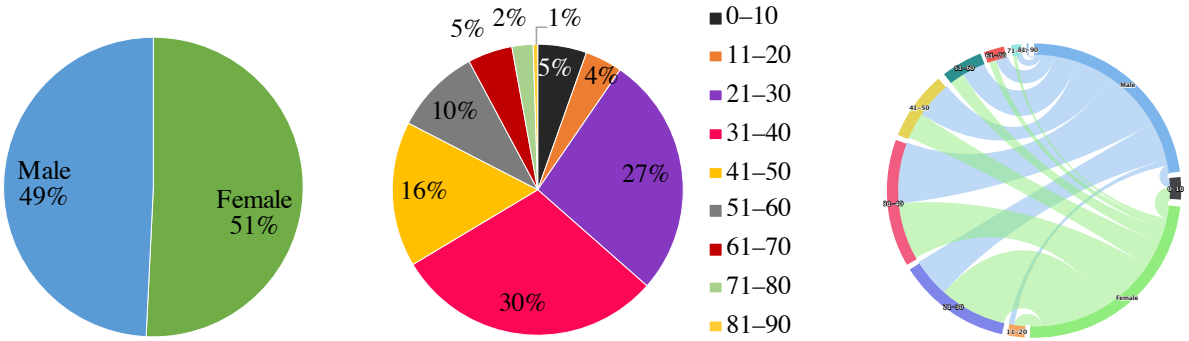


Figure 9. Scene distribution in OpenForensics dataset. Red represents indoor scenes (63.7%), and blue represents outdoor scenes (36.3%). Best viewed online in color with zoom-in.

3.2. Gender and Age



a) Gender b) Age c) Mutual dependencies

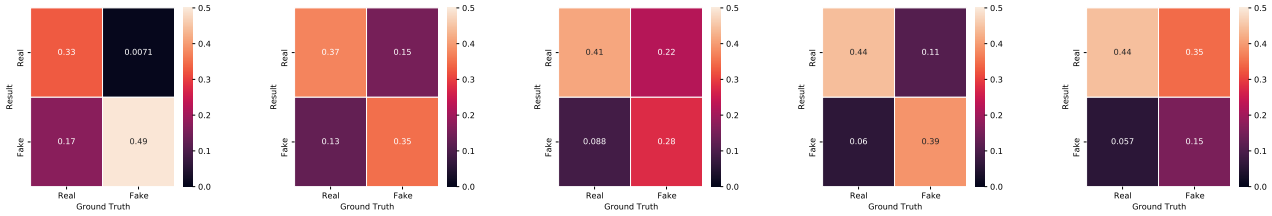
Figure 10. Mutual dependencies of age and gender in OpenForensics dataset. A thicker arc indicates a higher probability of one attribute correlating to another. Best viewed online in color with zoom-in.

The gender and age of persons in the OpenForensics dataset are exploited using pre-trained convolutional neural networks (CNNs) [1, 2]. Figure 10 shows the ratios of age and gender and their mutual dependencies in the OpenForensics dataset. The ratios of male and female are mostly equal, 49% and 51%, respectively; 5% are children (under 10), 4% are teenagers (10–20), 83% are adults (21–60), and 8% are seniors (60+).

4. Additional User Study Results

4.1. Additional Result of Face Forgery Classification

Figure 11 shows confusion matrices of the user study of face forgery classification tasks on different datasets.



a) FaceForensics++ Dataset b) DFDC Dataset c) Celeb-DF Dataset d) DeeperForensics Dataset e) OpenForensics Dataset

Figure 11. Human performance in face forgery classification on deepfake datasets. Images in OpenForensics dataset were most effective in spoofing all participants.

5. Benchmark Suite Details

5.1. Baseline Methods

We trained and evaluated existing instance detection and segmentation methods in various scenarios: MaskRCNN [10], MSRCNN [11], RetinaMask [8], YOLACT [3], YOLACT++ [4], CenterMask [16], BlendMask [5], PolarMask [29], MEInst [31], CondInst [25], SOLO [27], and SOLO2 [28]. MaskRCNN [10] and MSRCNN [11] are well-known two-stage models that perform detect-then-segment. The YOLACT family [3, 4] includes early single-stage methods, which are based on anchor-free object detection [34, 26]that are aimed at real-time performance. The remaining methods are widely used modern single-stage methods aimed at solving accuracy and processing time problems.

MaskRCNN [10], the first end-to-end instance segmentation model, was extended from Faster R-CNN [21] by adding a branch for predicting an object mask that is parallel with the existing branch for detecting a bounding box.

MSRCNN [11] was extended from Mask R-CNN by integrating two segmentation head-networks to improve the quality of segmented instances.

RetinaMask [8] was extended from RetinaNet [18] by integrating an instance mask prediction head network.

YOLACT [3] and **YOLACT++** [4] are aimed at real-time performance by breaking the segmentation process into two parallel subtasks (*i.e.*, generating a set of prototype masks and predicting per-instance mask coefficients) and then linearly combining the masks with the coefficients.

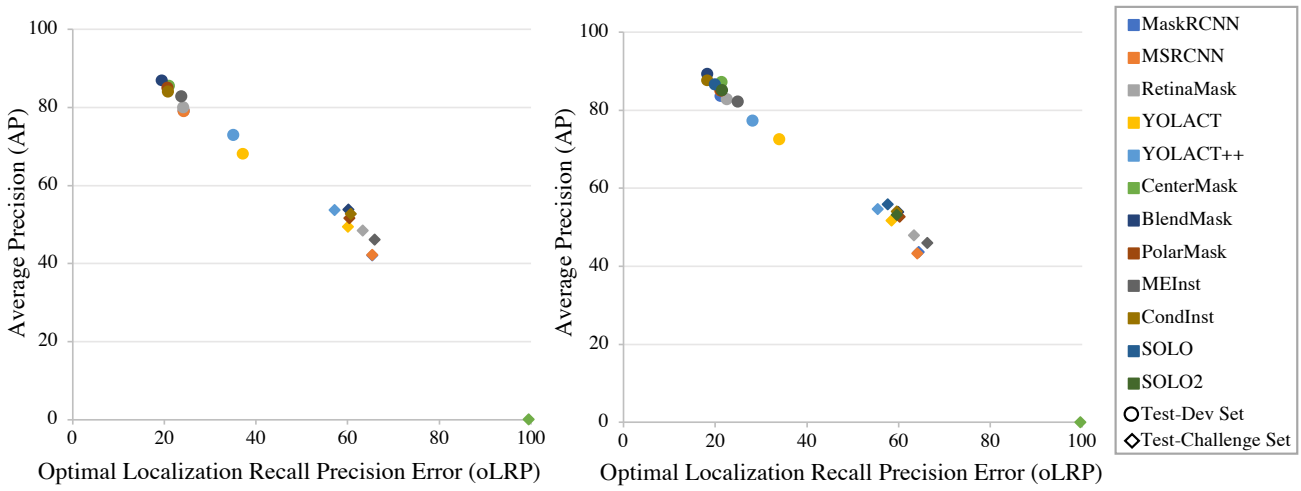
BlendMask [5] and **CenterMask** [16] were extended from YOLACT by blending cropped prototype masks with a finer-grained mask within each bounding box.

PolarMask [29] formulates the instance segmentation problem as instance center classification and dense distance regression in a polar coordinate.

MEInst [31] and **CondInst** [25] utilize fully convolutional networks to produce masks.

SOLO [27] and **SOLO2** [28] reformulate the instance segmentation as category prediction and mask generation to directly output masks without computing bounding boxes.

5.2. Overall Evaluation



a) Forgery Detection b) Forgery Segmentation

Figure 12. Benchmark results for multi-face forgery multi-task on OpenForensics dataset. Test-dev set results reflect benchmark performance for standard images while test-challenge set results reflect robustness for unseen images. Lower oLRP is better while higher AP is better. BlendMask was best method and YOLACT++ was most robust method.

5.3. Results on Val Set

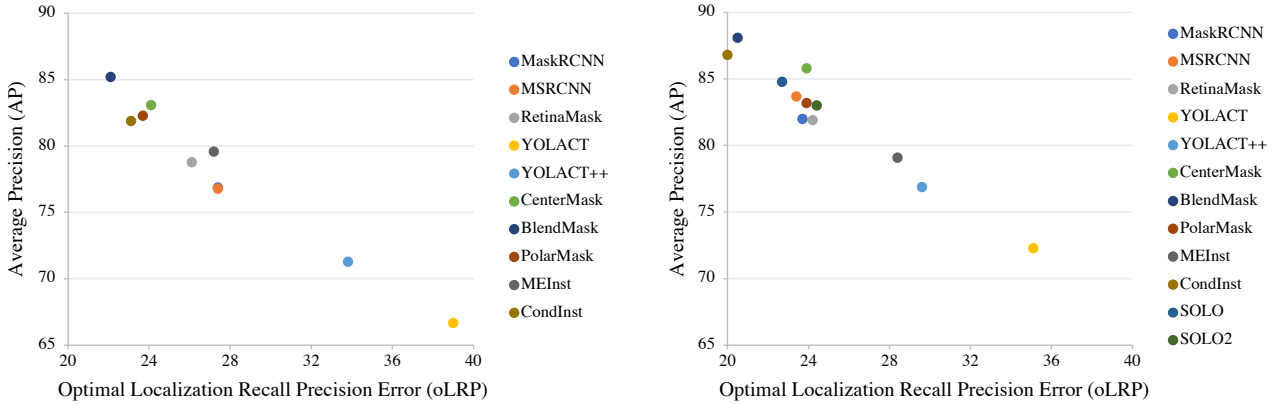


Figure 13. Benchmark results for multi-face forgery detection and segmentation on Val set. From left to right, multi-face forgery detection and multi-face forgery segmentation. Lower oLRP is better while higher AP is better. BlendMask was best method on both metrics for all subsets. Best viewed online in color with zoom-in.

Table 1. Benchmark results for multi-face forgery detection and segmentation on Val set using AP and oLRP. Higher AP is better while lower oLRP is better. Best and second-best results are shown in blue and red, respectively.

Method	Year	Multi-Face Forgery Detection								Multi-Face Forgery Segmentation							
		AP↑	AP_S↑	AP_M↑	AP_L↑	oLRP↓	$oLRP_{Loc}$↓	$oLRP_{FP}$↓	$oLRP_{FN}$↓	AP↑	AP_S↑	AP_M↑	AP_L↑	oLRP↓	$oLRP_{Loc}$↓	$oLRP_{FP}$↓	$oLRP_{FN}$↓
MaskRCNN [10]	ICCV 2017	76.9	33.9	73.4	78.0	27.4	10.1	4.2	5.6	82.0	20.1	74.4	84.1	23.7	7.8	3.8	6.4
MSRCNN [11]	CVPR 2019	76.8	34.0	73.7	77.8	27.4	10.2	3.9	5.3	83.7	21.2	77.2	85.1	23.4	8.0	3.3	5.9
RetinaMask [8]	arXiv 2019	78.8	34.1	74.6	79.9	26.1	9.3	3.9	5.8	81.9	22.1	74.1	83.7	24.2	8.1	4.1	5.9
YOLACT [3]	ICCV 2019	66.7	21.5	59.0	67.7	39.0	13.7	7.6	9.6	72.3	4.9	58.1	74.4	35.1	11.2	7.7	9.8
YOLACT++ [4]	TPAMI 2020	71.3	27.7	66.7	72.3	33.8	12.4	5.5	7.0	76.9	9.4	65.8	78.9	29.6	9.9	5.6	7.2
CenterMask [16]	CVPR 2020	83.1	35.9	77.8	84.2	24.1	7.2	4.4	8.0	85.8	21.0	77.7	87.5	23.9	6.4	5.5	8.6
BlendMask [5]	CVPR 2020	85.2	38.8	79.3	86.1	22.1	6.5	3.0	8.6	88.1	25.7	79.6	89.6	20.5	5.4	3.3	8.5
PolarMask [29]	CVPR 2020	82.3	32.7	77.5	83.2	23.7	7.1	3.2	8.8	83.2	19.2	75.8	84.6	23.9	7.2	3.2	8.8
MEInst [31]	CVPR 2020	79.6	32.9	75.2	80.6	27.2	8.2	5.1	9.5	79.1	20.4	73.4	80.4	28.4	8.7	5.2	9.7
CondInst [25]	ECCV 2020	81.9	34.7	77.4	82.8	23.1	7.9	3.0	6.5	86.8	22.6	77.5	88.4	19.8	6.0	3.0	6.5
SOLO [27]	ECCV 2020	-	-	-	-	-	-	-	-	84.8	19.8	77.5	86.4	22.7	7.0	3.4	7.7
SOLO2 [28]	NeurIPS 2020	-	-	-	-	-	-	-	-	83.0	17.1	75.2	84.7	24.4	7.4	4.9	7.4

5.4. Class-Wise Evaluation

We report the results for each method in two categories, 'Real' and 'Fake': AP_{Real}, AP_{Fake}, $oLRP_{Real}$, and $oLRP_{Fake}$. Notably, we observed high detection performance by BlendMask for the 'Fake' category, as shown in Table 2. For the 'Real' category, the modern single-stage methods (*i.e.*, BlendMask and CenterMask) achieved the highest AP while the two-stage methods (*i.e.*, RetinaMask and MSRCNN) tended to have a lower oLRP error. In addition, BlendMask had the best segmentation performance for forged faces along with the best results for both AP and oLRP error (cf. Table 2). BlendMask and MSRCNN had the best segmentation performance for real faces.

Table 2. Class-wise performance on test-dev set. Higher AP is better while lower oLRP error is better. Best and second-best results are shown in blue and red, respectively.

Method	Year	Multi-Face Forgery Detection				Multi-Face Forgery Segmentation			
		$\mathbf{AP}_{Real}\uparrow$	$\mathbf{AP}_{Fake}\uparrow$	$\mathbf{oLRP}_{Real}\downarrow$	$\mathbf{oLRP}_{Fake}\downarrow$	$\mathbf{AP}_{Real}\uparrow$	$\mathbf{AP}_{Fake}\uparrow$	$\mathbf{oLRP}_{Real}\downarrow$	$\mathbf{oLRP}_{Fake}\downarrow$
MaskRCNN [10]	ICCV 2017	81.0	77.3	25.1	23.4	85.4	81.8	22.7	19.6
MSRCNN [11]	CVPR 2019	81.2	76.9	24.9	23.8	87.6	82.7	22.0	20.1
RetinaMask [8]	arXiv 2019	83.2	76.8	24.2	24.1	86.4	79.3	22.8	22.3
YOLACT [3]	ICCV 2019	72.1	64.1	36.9	37.5	78.8	66.2	32.3	35.6
YOLACT++ [4]	TPAMI 2020	76.9	68.9	30.6	32.4	83.2	71.4	26.9	29.4
CenterMask [16]	CVPR 2020	83.2	87.8	26.6	15.6	86.3	88.1	26.7	16.0
BlendMask [5]	CVPR 2020	84.3	89.6	25.2	13.8	88.4	90.1	22.7	13.8
PolarMask [29]	CVPR 2020	82.0	87.9	26.3	15.1	83.7	86.2	26.0	16.7
MEInst [31]	CVPR 2020	80.8	84.7	29.4	18.2	81.9	82.4	29.2	21.0
CondInst [25]	ECCV 2020	81.6	86.4	26.2	15.4	86.9	88.6	22.7	13.9
SOLO [27]	ECCV 2020	-	-	-	-	85.6	87.5	23.9	16.2
SOLO2 [28]	NeurIPS 2020	-	-	-	-	85.3	84.9	25.0	18.1

Table 3. Class-wise performance on test-challenge set. Higher AP is better while lower oLRP error is better. Best and second-best results are shown in blue and red, respectively.

Method	Year	Multi-Face Forgery Detection				Multi-Face Forgery Segmentation			
		$\mathbf{AP}_{Real}\uparrow$	$\mathbf{AP}_{Fake}\uparrow$	$\mathbf{oLRP}_{Real}\downarrow$	$\mathbf{oLRP}_{Fake}\downarrow$	$\mathbf{AP}_{Real}\uparrow$	$\mathbf{AP}_{Fake}\uparrow$	$\mathbf{oLRP}_{Real}\downarrow$	$\mathbf{oLRP}_{Fake}\downarrow$
MaskRCNN [10]	ICCV 2017	45.3	38.8	64.1	66.7	47.9	39.5	62.6	66.2
MSRCNN [11]	CVPR 2019	45.7	38.8	64.0	66.7	46.3	40.2	62.4	65.7
RetinaMask [8]	arXiv 2019	49.0	48.1	63.0	63.7	50.5	45.6	61.9	64.8
YOLACT [3]	ICCV 2019	50.8	47.9	60.8	59.5	55.5	48.0	57.8	59.0
YOLACT++ [4]	TPAMI 2020	55.6	51.9	57.5	56.6	56.8	52.5	55.2	55.7
CenterMask [16]	CVPR 2020	0.02	0.05	99.7	99.4	0.01	0.04	99.7	99.5
BlendMask [5]	CVPR 2020	53.1	54.7	60.9	59.4	54.6	53.4	59.7	60.1
PolarMask [29]	CVPR 2020	49.9	53.6	62.8	57.9	51.6	53.8	62.2	58.2
MEInst [31]	CVPR 2020	44.6	47.5	66.6	66.1	45.2	46.8	66.4	66.1
CondInst [25]	ECCV 2020	51.5	53.9	62.5	58.9	54.2	54.0	60.6	58.6
SOLO [27]	ECCV 2020	-	-	-	-	55.9	55.8	59.6	55.7
SOLO2 [28]	NeurIPS 2020	-	-	-	-	53.1	53.4	62.0	57.1

Table 4. Class-wise performance on Val set using AP and oLRP. Higher AP is better while lower oLRP is better. Best and second-best results are shown in blue and red, respectively.

Method	Year	Multi-Face Forgery Detection				Multi-Face Forgery Segmentation			
		$\mathbf{AP}_{Real}\uparrow$	$\mathbf{AP}_{Fake}\uparrow$	$\mathbf{oLRP}_{Real}\downarrow$	$\mathbf{oLRP}_{Fake}\downarrow$	$\mathbf{AP}_{Real}\uparrow$	$\mathbf{AP}_{Fake}\uparrow$	$\mathbf{oLRP}_{Real}\downarrow$	$\mathbf{oLRP}_{Fake}\downarrow$
MaskRCNN [10]	ICCV 2017	75.9	78.0	31.6	23.2	81.0	83.0	28.5	18.9
MSRCNN [11]	CVPR 2019	76.3	77.2	30.9	23.9	83.9	83.4	27.1	19.7
RetinaMask [8]	arXiv 2019	79.0	78.6	29.3	22.8	82.7	81.1	27.5	20.9
YOLACT [3]	ICCV 2019	67.3	66.0	42.2	35.8	76.1	68.5	36.6	33.5
YOLACT++ [4]	TPAMI 2020	72.8	69.9	35.8	31.8	81.1	72.8	31.0	28.3
CenterMask [16]	CVPR 2020	77.7	88.5	33.3	14.9	82.0	89.4	32.6	15.2
BlendMask [5]	CVPR 2020	79.5	90.9	31.6	12.7	84.3	91.8	28.6	12.5
PolarMask [29]	CVPR 2020	75.6	89.0	33.4	14.0	78.6	87.8	18.4	29.7
MEInst [31]	CVPR 2020	73.8	85.5	37.0	17.4	75.2	83.1	37.0	20.0
CondInst [25]	ECCV 2020	76.9	87.0	31.1	15.0	82.9	90.6	27.2	12.4
SOLO [27]	ECCV 2020	-	-	-	-	80.2	89.4	30.6	14.9
SOLO2 [28]	NeurIPS 2020	-	-	-	-	79.8	86.1	31.4	17.5

References

[1] Age and gender estimation. https://pypi.org/project/py-agender/, 2018. [Online; accessed 18-Feb-2021]. 10

[2] Insightface: a deep learning toolkit for face analysis. http://insightface.ai/, 2018. [Online; accessed 18-Feb-2021]. 10

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *International Conference on Computer Vision*, 2019. 11, 12, 13

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++: Better real-time instance segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 2020. 11, 12, 13

[5] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 11, 12, 13

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Conference on Computer Vision and Pattern Recognition*, pages 1800–1807, 2017. 2

[7] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020. 2

[8] Cheng-Yang Fu, Mykhailo Shvets, and Alexander C. Berg. RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. In *arXiv preprint arXiv:1901.03353*, 2019. 11, 12, 13

[9] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *European Conference on Computer Vision*, 2020. 2

[10] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision*, pages 2980–2988, 2017. 11, 12, 13

[11] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask Scoring R-CNN. In *Conference on Computer Vision and Pattern Recognition*, 2019. 11, 12, 13

[12] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[14] Davis E. King. A toolkit for making real world machine learning and data analysis applications in c++. http://dlib.net/. 2

[15] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[16] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 11, 12, 13

[17] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A large-scale challenging dataset for deepfake forensics. In *Conference on Computer Vision and Pattern Recognition*, June 2020. 9

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, pages 2980–2988, 2017. 11

[19] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *SIGGRAPH*, pages 313–318, 2003. 2, 4

[20] Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 11

[22] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. Faceforensics++: Learning to detect manipulated facial images. In *International Conference on Computer Vision*, pages 1–11, Oct 2019. 9

[23] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: Database and results. *Image and vision computing*, 47:3–18, 2016. 2

[24] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Conference on Computer Vision and Pattern Recognition*, 2020. 2

[25] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *European Conference on Computer Vision*, 2020. 11, 12, 13

[26] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *International Conference on Computer Vision*, October 2019. 11

[27] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. In *European Conference on Computer Vision*, 2020. 11, 12, 13

[28] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic, faster and stronger. In *Conference on Neural Information Processing Systems*, 2020. 11, 12, 13

[29] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 11, 12, 13

[30] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *European Conference on Computer Vision*, pages 325–341, 2018. 2

[31] Rufeng Zhang, Zhi Tian, Chunhua Shen, Mingyu You, and Youliang Yan. Mask encoding for single shot instance segmentation. In *Conference on Computer Vision and Pattern Recognition*, 2020. 11, 12, 13

[32] Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. Faceboxes: A cpu real-time face detector with high accuracy. In *International Joint Conference on Biometrics*, pages 1–9, 2017. 1

[33] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 2017. 9

[34] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 11

[35] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Conference on Computer Vision and Pattern Recognition*, pages 146–155, 2016. 2