VICTORIA UNIVERSITY
MELBOURNE AUSTRALIA

Advancing face detection efficiency: Utilizing classification networks for lowering false positive incidences

This is the Published version of the following publication

Zhang, Jianlin, Hou, Chen, Yang, Xu, Yang, Xuechao, Yang, Wencheng and Cui, Hui (2024) Advancing face detection efficiency: Utilizing classification networks for lowering false positive incidences. Array, 22. ISSN 2590-0056

The publisher’s official version can be found at
https://www.sciencedirect.com/science/article/pii/S2590005624000134?via%3Dihub
Note that access to this version may require subscription.

Downloaded from VU Research Repository https://vuir.vu.edu.au/48801/

Array 22 (2024) 100347

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/locate/array

Advancing face detection efficiency: Utilizing classification networks for lowering false positive incidences

Jianlin Zhang [a], Chen Hou [b], Xu Yang [b,*], Xuechao Yang [c], Wencheng Yang [d], Hui Cui [e]

[a] Fuzhou Institute of Oceanography & College of Computer and Cyber Security, Fujian Normal University, Fuzhou, China
[b] College of Computer and Data Science, Minjiang University, Fuzhou, China
[c] College of Art, Business, Law, Education & IT, Victoria University, Melbourne, Australia
[d] School of Mathematics, Physics and Computing, University of Southern Queensland, Australia
[e] Department of Software Systems & Cybersecurity, Monash University, Melbourne, Australia

ARTICLE INFO

Keywords:
Convolutional Neural Network (CNNs)
Face detection
Pseudo-face image
False positive rate
Object detection

ABSTRACT

The advancement of convolutional neural networks (CNNs) has markedly progressed in the field of face detection, significantly enhancing accuracy and recall metrics. Precision and recall remain pivotal for evaluating CNN-based detection models; however, there is a prevalent inclination to focus on improving true positive rates at the expense of addressing false positives. A critical issue contributing to this discrepancy is the lack of pseudo-face images within training and evaluation datasets. This deficiency impairs the regression capabilities of detection models, leading to numerous erroneous detections and inadequate localization. To address this gap, we introduce the WIDERFACE dataset, enriched with a considerable number of pseudo-face images created by amalgamating human and animal facial features. This dataset aims to bolster the detection of false positives during training phases. Furthermore, we propose a new face detection architecture that incorporates a classification model into the conventional face detection model to diminish the false positive rate and augment detection precision. Our comparative analysis on the WIDERFACE and other renowned datasets reveals that our architecture secures a lower false positive rate while preserving the true positive rate in comparison to existing top-tier face detection models.

1. Introduction

Face detection constitutes the initial and foundational step for numerous tasks and applications related to faces, including face alignment [1,2], facial attribute analysis [3–6], face recognition [7–10], and identity verification [11]. This makes it an exceptionally crucial task within the realm of computer vision. Consequently, over recent years, diverse approaches have been proposed to tackle this challenge from various angles. Some works [12–14] introduce annotated landmark information as supplementary supervisory signals, while others [15–21] prioritize network design aspects. Additionally, novel loss formulations [15,16,22] and data augmentation methodologies [17,18] have been put forward. Most notably, certain contributions [21,23] have initiated a rethinking of matching strategies and label assignment processes.

Another subset of research focuses on the architecture of face detectors, mainly encompassing single-stage face detectors [12,16,17,22,24,25], as well as two-stage and multi-stage face detectors [26–28]. Among these, the single-stage approach relies on domain and anchor-based face detection methods, employing tiling rules and dense anchors across all positions of multi-scale feature maps with various scales and aspect ratios. Typically, this framework consists of four key components: the backbone, feature module, head network, and multi-task loss. The feature module employs a Feature Pyramid Network (FPN) [29,30] to aggregate hierarchical feature maps between the backbone's high-level and low-level features. Additionally, modules for refining receptive fields [16,22,31], such as the Receptive Field Block (RFB), are introduced to provide abundant hard-surface contextual information. Furthermore, the multi-task loss encompasses binary classification and bounding box regression. The former classifies predefined anchors into faces and backgrounds, while the latter accurately regresses detected faces to their precise locations.

All these efforts have significantly elevated the performance of face detection by focusing on accurately identifying genuine faces. However, they often neglect another vital face detection metric: the false positive rate, which gauges the ability to exclude non-authentic images. One contributing factor to a high false positive rate is the lack of robust regression ability, leading to Localization (LOC) errors that manifest as

[*] Corresponding author.
E-mail address: xu.yang@mju.edu.cn (X. Yang).

https://doi.org/10.1016/j.array.2024.100347
Received 21 February 2024; Received in revised form 21 April 2024; Accepted 26 April 2024
Available online 29 April 2024
2590-0056/© 2024 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

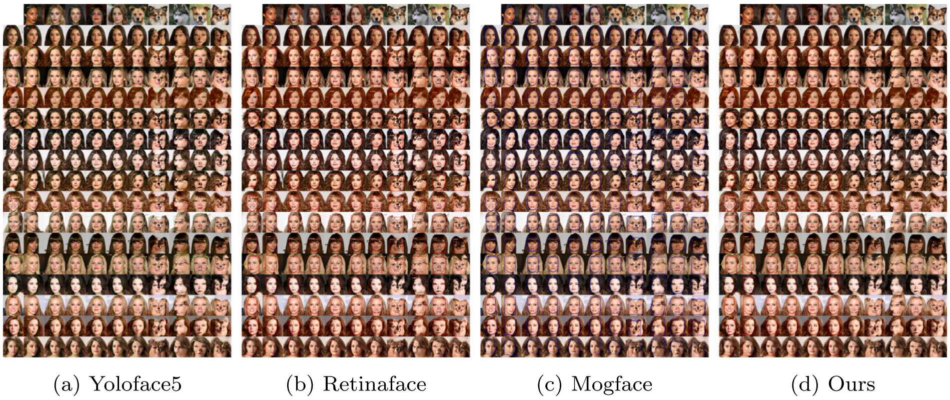

(a) Yoloface5
(b) Retinaface
(c) Mogface
(d) Ours

Fig. 1. Various face detection models exhibit different detection outcomes on both real and fake face images. Our model demonstrates a remarkable ability to accurately discriminate and exclude fake faces.

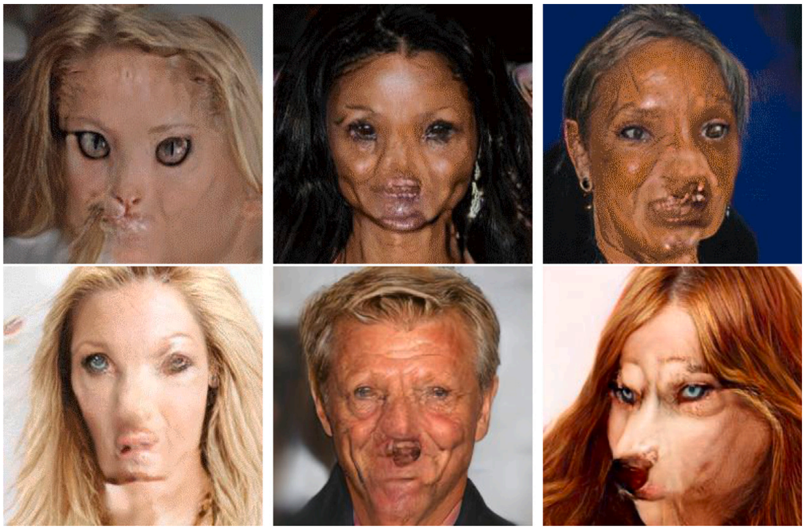


Fig. 2. Examples of synthetic face images in our dataset.

numerous erroneous detections and imprecise positioning. Enhancing the regression capability of facial detectors could mitigate these false positives stemming from regression errors. As highlighted in [16], with increasing IoU thresholds, the Average Precision (AP) drastically drops, indicating the need for enhanced accuracy in bounding box localization. Nonetheless, blindly introducing multi-step regression into face detection tasks can paradoxically yield counterproductive outcomes, and many multi-view learning [32]and multi-view clustering [33]can solve these problems, demanding further exploration.

Another factor contributing to a high false positive rate is attributed to deficiencies within the dataset itself, as facial datasets often include counterfeit face images. When considering genuine facial images versus counterfeits, the significance of face detection becomes apparent. The essence of face detection lies in identifying actual faces for subsequent recognition or authentication purposes, which generally pertain to authentic individuals. Ensuring that the detected faces correspond to real individuals rather than animated or sculpted representations is of paramount importance. Despite this, within some of the well-regarded facial detection datasets like FDDB [34] and the WIDERFACE[35] dataset, instances of non-authentic "faces" can still be found. We contend that this is less than ideal, as dataset flaws can lead to flaws in detection outcomes. For instance, training on datasets with a significant number of non-authentic "face" images may yield models with higher false positive rates. Such models could potentially compromise downstream tasks' security, such as making identity verification systems susceptible to attacks or raising safety concerns in autonomous driving scenarios.

In this paper, we begin by conducting a series of experiments that highlight the prevalent issue of high false positive rates in current face detection models (as depicted in Fig. 1). Addressing this issue is of utmost significance. Subsequently, we introduce a synthetic dataset named "FAKEFACE", comprising images that are not actual human faces but generated through image synthesis. Interestingly, all images in this dataset are identified as faces by the RetinaFace detector. Although the authenticity of images in these datasets was not detected on other face detection models, as this would require a significant amount of time and efficiency. But it should be noted that Retinaface has better facial detection capabilities than most detection models, so its results have enough ability to replace most facial detection models. Furthermore, we propose a false positive rate evaluation metric for this dataset, aimed at assessing different models' ability to exclude non-genuine images. Lastly, we present a methodology to mitigate the false positive rates in face detection models. In summary, our contributions are as follows:

- We conducted relevant experiments to assess the capability of various face detection models to exclude counterfeit face images.
- We introduced a dataset comprising synthetic facial images that are all detected as faces. Our dataset can serve as a means to evaluate the false positive rate of a face detection model.
- Lastly, we presented an approach to mitigate the false positive rates in face detection models.

2. Related work

With the advent of the deep learning era, general object detection has rapidly transitioned under the dominance of deep learning methodologies. Multi-task models amalgamate multiple single-task methods within a single framework. In MTCNN [36], the authors employed a cascade of image pyramids and shallow CNNs to predict facial bounding boxes and landmarks. In [29], the naturally occurring feature pyramid in CNNs was utilized. For landmark localization, additional regression heads were incorporated into popular detectors such as SSD [37] and RetinaNet [38], as demonstrated in [12,39]. In [12,40], branches for predicting facial 3D shapes were added. Mask R-CNN [41] offers a versatile and flexible architecture for multi-task handling. While most literature is dedicated to the application of OHEM in face detection, maskface [14] demonstrated the efficacy of focal loss in yielding advanced outcomes.

The two-stage approach originated from R-CNN [42] and Fast R-CNN [43]. Faster R-CNN [28] swiftly introduced the Region Proposal Network (RPN), replacing selective search with predefined anchor boxes, thus becoming the most renowned anchor-based general object detection method. The anchor-based method encounters significant class imbalance between positive and negative anchors. Class imbalance issues are typically addressed through techniques like Online Hard Example Mining (OHEM) [37] or dynamically scaled cross-entropy loss (focal loss) [37,38]. Building upon Faster R-CNN [28], numerous new methods emerged, such as FPN [29], Mask R-CNN [41], Cascade R-CNN [44], and more. Context can be modeled by explicitly enlarging the window surrounding candidate proposals [45]. In single-shot detection methods, context is integrated through additional convolutional layers that expand the receptive field [20,22].

To overcome the high latency of two-stage methods, several single-stage methods have been proposed. In single-stage face detection models, bounding boxes are predicted in a single forward pass [37,46]. To detect challenging objects, such as small or highly occluded facial features with complex poses, YOLO [47,48] introduced novel anchor matching strategies that involve feedback on proposals, associating ground truth with an anchor, and reweighting the regression of object width and height. Regression and segmentation techniques are utilized for facial landmark prediction. Regression methods often rely on L1 and L2 loss or their variations [49–51]. Furthermore, multiple stages for landmark refinement can be employed to enhance accuracy [49,52]. Non-Maximum Suppression (NMS) [53] is used to compute the final predicted boxes. Methods that utilize 3D facial reconstruction for dense landmark prediction also exist [54,55].

For deep convolutional networks, VGG [56] employed an architecture with very small 3 × 3 convolutional filters to increase depth. ResNet [57] demonstrated the importance of the information flow and introduced skip connections to address degradation in deeper networks. The PyramidBox series [17,18] recommended their own upsampling blocks to enhance the expressiveness of features for finer facial details. The Feature Pyramid Network (LFPN) and the Contextual Prediction Module (CPM) emphasized the significance of context and data anchor sampling enhancement, merging strong semantic features with low resolution and weak features from high-resolution layers. DSFD [22] introduced a dual-head detector using Improved Anchor Matching (IAM) and Progressive Anchor Loss (PAL). Subsequently, RetinaFace [12] manually annotated five facial landmarks as extra supervisory signals on facial regions. RefineFace [16] introduced five additional modules: Selective Two-step Regression (STR), Selective Two-step Classification (STC), Scale-aware Margin Loss (SML), Feature Supervision Module (FSM), and Receptive Field Enhancement (RFE).

HAMBox [23] underscored strong regression capability for some mismatched anchors and proposed an online high-quality anchor mining strategy. Moreover, ASFD [15] employed neural architecture search techniques to automatically discover architectures for efficient multi-scale feature fusion and context enhancement. Mogface [58] introduced an adaptive online incremental anchor mining strategy, selective scale enhancement strategy, and hierarchical context-aware module, achieving state-of-the-art results on WIDERFACE [35].

Despite the superior performance of these detection methods, as shown in Fig. 1, even the most powerful Mogface model has not solved the problem of false positives. We conducted a series of experiments to try to solve the problem of false positives.

3. FAKEFACE dataset

In this section, we introduce our proposed dataset called "FAKEFACE" a challenging synthetic dataset for face detection(see Fig. 2). Face detection forms the foundation for downstream tasks like face recognition and identity verification. The distinction between face detection and facial feature detection needs to be clarified: face detectors must learn to identify real human faces, rather than images containing facial features that may belong to animations, sculptures, or graffiti.

While many datasets contain faces with wild makeup or facial abnormalities, or even heavily blurred faces [59], the classification of whether these facial images qualify as authentic is a subject of debate. However, our dataset eliminates such ambiguities. The FAKEFACE dataset we propose is synthesized using starganv2 [60] and consists of images blending animal faces with human faces.

3.1. Overview

Our dataset comprises 20,307 trainable synthetic face images, 2303 validation images, and 1410 testing images. We refrained from annotating the dataset, as we do not consider these to be genuine human face images; however, they exhibit strong facial features, capable of deceiving many face detection models.

The envisaged scenarios for our dataset's utilization are as follows: Train facial detectors on external datasets and test on FAKEFACE. Alternatively, train/validate facial detectors on the training/validation partitions of FAKEFACE and test on FAKEFACE.

3.2. Data collection

Our dataset is created by synthesizing "human" faces through the fusion of human and animal faces using starganv2. We selected human face images from CelebA-HQ with varying genders, skin tones, and ages, and combined them with diverse animal face images from AFHQ. Subsequently, we employed RetinaFace to detect the images that were classified as faces, and extracted these as part of our dataset. The fusion of different human and animal features produces the synthesized "human" faces in our dataset.

The image synthesis model is shown in Fig. 3, The generator converts the input image into an output image, reflects the style code of a specific domain through instance normalization (IN) [61] downsampling, and outputs the image using adaptive instance normalization (AdaIN) [62]upsampling. Where the mapping network or style encoder provides a specific domain style code S, which is injected into the generator using ADAIN. We sample latent codes from a standard Gaussian distribution and input them into an MLP to generate the style code. The style encoder transforms the input style image into a style code using a CNN.

To generate face images, a generator G takes an image x and a style code $\tilde{s}$ as input, and learns to generate an output image G(x, $\tilde{s}$) through an adversarial loss:

$$\begin{aligned}\mathcal{L}_{adv} =&\mathbb{E}_{x,y}[\log D_y(X)]+\\ &\mathbb{E}_{x,\tilde{y},z}[\log(1 - D_{\tilde{y}}(G(x,\tilde{s})))]\end{aligned} \tag{1}$$

Among them, $\tilde{s} = F_{\tilde{y}}(z)$ is the style code generated by using a mapping network to input a random latent code z, and $D_y(.)$ and $D_{\tilde{y}}(.)$ are the outputs of the discriminator. In order to use the style code $\tilde{s}$ when generating images, we employ a style reconstruction loss:

$$\mathcal{L}_{sty} = \mathbb{E}_{x,\tilde{y},z}\left[\left\|\tilde{s} - E_{\tilde{y}}(G(x,\tilde{s}))\right\|_1\right] \tag{2}$$

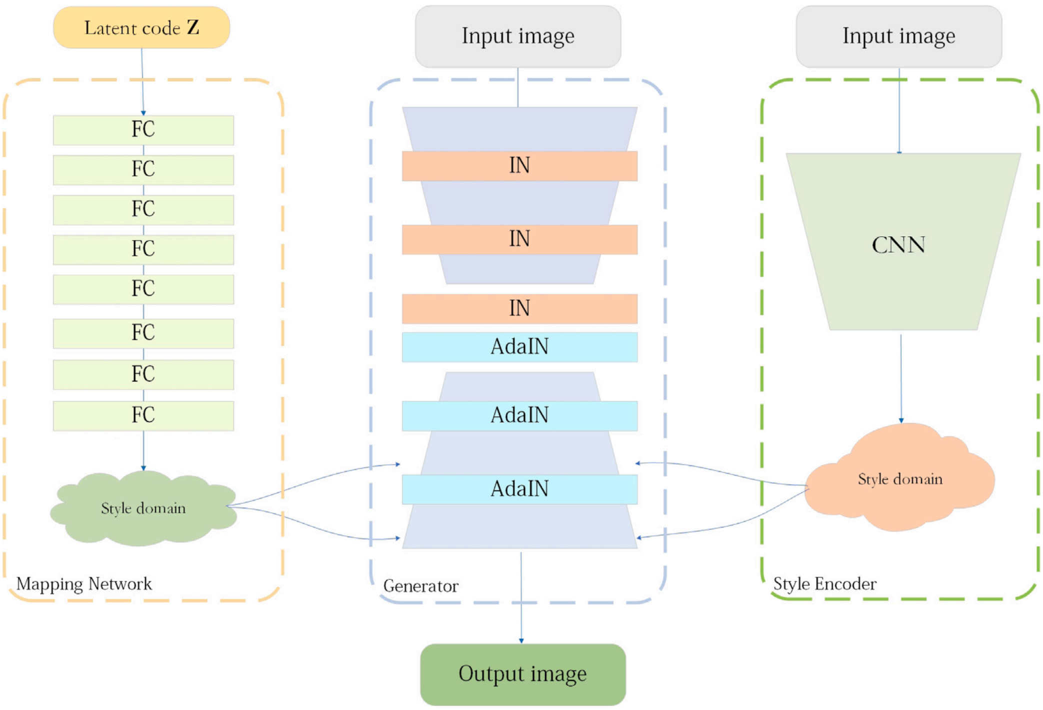


Fig. 3. Starganv2 [60] model structure.

And a style diversity loss to enhance the diversity of generated images:

$$\mathcal{L}_{ds} = \mathbb{E}_{x,\tilde{y},z_1,z_2}\left[\|G(x,\tilde{s_1}) - G(x,\tilde{s_2})\|_1\right] \tag{3}$$

Among them, the target style codes s_1 and s_2 are generated by a mapping network conditional on two random latent codes z_1 and z_2, respectively. At the same time, in order to ensure that the original features such as face pose can be maintained in the generated image, we use the cycle consistency loss:

$$\mathcal{L}_{cyc} = \mathbb{E}_{x,y,\tilde{y},z}[\|x - G(G(x,\tilde{s}),\hat{s})\|_1] \tag{4}$$

Among them, $\hat{s}$ is the style code of the original image. The overall objective function can be summarized as:

$$\min_{G,F,E}\max_{D} \mathcal{L}_{adv} + \lambda_{sty}\mathcal{L}_{sty} - \lambda_{ds}\mathcal{L}_{ds} + \lambda_{cyc}\mathcal{L}_{cyc} \tag{5}$$

Where λ_{sty}, λ_{ds}, λ_{cyc} are the hyperparameters of each item, and they are all set to 1 in our training model.The discriminator structure is consistent with the style encoder and includes multiple output branches.

We selected face images from CelebA-HQ with different genders, skin tones, and ages, and synthesized them with various animal face images from AFHQ. The resulting images were then fed into Retinaface for detection, and any images detected as faces were cropped and included as part of our dataset. This means that every image in our dataset is detected as a face by Retinaface.

4. Method

In this section, we will introduce our approach for optimizing and reducing false positives in face detection. Our method is based on Retinaface [12], an efficient and fast face detection model. The main structure of the model is illustrated in Fig. 4, which utilizes a feature pyramid to extract image features and employs the Context Module to enlarge the network's receptive field, enhancing the model's ability to capture small facial details. Subsequently, the multi-task loss assists in simultaneously predicting face scores, face bounding boxes, five facial landmarks, as well as the 3D positions and correspondences of each facial pixel, significantly improving the effectiveness of face detection. Finally, our Classification Module is employed to reduce false positives and can be applied to other detection models.

Classification Network. We believe that current face detection models have already achieved very high precision and recall rates on the WIDERFACE dataset, with only some room for improvement in terms of details. To address this issue, we adopted an early solution from object detection, which is image classification. Image classification models are not only easy to train but also exhibit good modular performance and robustness. They can be applied to other face detection models facing high false positive issues. The structure of our classification network is shown in Fig. 5. The dashed box represents the ResNet50 network, which is trained using cross-entropy loss for classification.

Backbone Network. As shown in Fig. 5,for our classifier, we employ ResNet50 [57] as the backbone network, Output parameters are computed using cross-entropy loss and updated through the Adam optimizer to minimize the loss. The depth of this model allows it to grasp more intricate features, thereby enhancing its accuracy. Additionally, ResNet50 employs residual learning. If the input and output of a layer are the same, that layer is an identity mapping; if they differ, it is a residual mapping. ResNet50 uses residual blocks to implement this concept. Each residual block consists of two convolutional layers and a skip connection. This skip connection directly passes the input to the output, mitigating the vanishing gradient problem. The model also employs global average pooling, computing the average value of all pixels in each feature map as its output. This technique reduces the model's parameter count, thereby curbing the risk of overfitting. Overall, ResNet50 is a powerful deep learning model widely applied in the field of computer vision.

Classification Loss. Cross-entropy loss is a commonly used loss function in deep learning, frequently employed for classification tasks. It quantifies the discrepancy between predicted and actual outcomes, serving as a key metric for optimizing model parameters.

$$L(x, y) = -\sum_{i=1}^{N} x_i log y_i \tag{6}$$

Given the input data x and N as the number of classes, where x_i denotes the ith element of the true labels and y_i signifies the probability of x

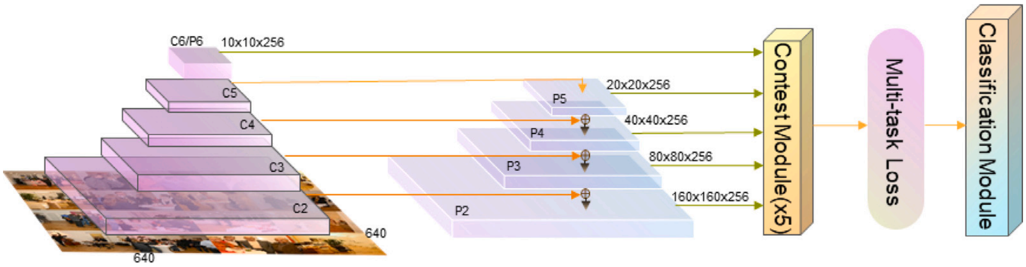


Fig. 4. The architecture of our solution.

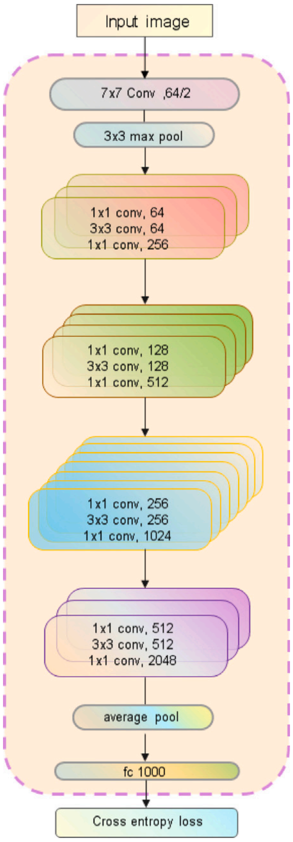


Fig. 5. The diagram of the classification network structure is shown with dashed lines.

belonging to the ith class, the essence of cross-entropy loss lies in measuring the distance between two probability distributions. When these distributions are closer, the cross-entropy loss is smaller, indicating more accurate predictive results from the model.

An added advantage of using cross-entropy as a loss function is that it sidesteps the issue of diminishing learning rates during gradient descent, which is commonly encountered with the mean squared error loss function. This is because the learning rate can be regulated by the error in the output when employing the sigmoid function.

Optimizer. Adam optimizer [63] combines the advantages of both AdaGrad and RMSProp optimization algorithms, making it suitable for sparse gradients and mitigating the issue of gradient oscillation. By considering both the first-order moment estimate (mean of gradients) and the second-order moment estimate (uncentered variance of gradients), Adam calculates the update step. The core concept is to compute moving averages of gradients and squared gradients at each time step, using them to update model parameters.

Specifically, the Adam optimizer defines two exponential moving averages: the first one for the exponential moving average of gradients and the second one for the exponential moving average of squared gradients. These two averages are utilized to adaptively adjust the learning rate for each parameter, achieving the effect of adaptive learning rates. The update rule for the Adam optimizer is as follows:

To compute the gradient at time step t: $g_t = \nabla_\theta f_t(\theta_{t-1})$ Compute the exponential moving average of the gradient, where m_0 is initialized to 0. β_1 is a coefficient that controls the weighting between momentum and the current gradient, typically set close to 1 (default: 0.9).$m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t$. Next, calculate the exponential moving average of the squared gradient, with v_0 initialized to 0.β_2 is a coefficient that controls the influence of previous squared gradients (default: 0.999). Weighted average is applied to the squared gradient.$v_t = \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot g_t^2$ Correct the bias of the gradient mean to reduce its impact during the initial training phase.$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$. Similarly, the bias correction is applied to $v_0$as well: $\hat{v}_t = \frac{v_t}{1-\beta_2^t}$. The final formula is as follows:

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \tag{7}$$

where g_t is the gradient of the parameters.β_1 and β_2 are decay coefficients for the two exponential moving averages.$\hat{m}_t$ and $\hat{v}_t$ are bias-corrected moving averages of the gradient.θ_t is the updated parameter. $\aleph = 0.001$ is the learning rate.$\epsilon = 10^{-8}$ is a small constant to avoid division by zero.

5. Experiment

In this section, we presented our approach of using an image classifier to address the issue of high false positive rates in object detection, as described in Section 5.1. Subsequently, the detection validation in Section 5.2 demonstrated that our method does not compromise the original effectiveness of object detection. Lastly, Section 5.3 employs false positive rate assessment to evaluate and validate the outcomes of our approach in reducing false positive rates.

5.1. Training classifiers

Dataset. For classifier training, we utilized both the CelebA-HQ dataset and the FAKEFACE dataset. The CelebA-HQ dataset consists of 26,579 training images and 2000 validation images, all with a resolution of 1024 × 1024. The FAKEFACE dataset contains 20,307 training images and 2303 validation images, with a resolution of 256 × 256.

Training. Firstly, we adjusted the image data from both datasets to a size of 224 × 224 pixels and normalized the pixel values to be within the range of 0 to 1 to meet the input requirements of ResNet-50. Simultaneously, we split the data into training and testing sets and employed data loaders for batch loading to efficiently handle the data.

We utilized ResNet-50 as the backbone network, initializing all its weights randomly, and trained all layers from scratch. In this process, we employed cross-entropy loss as the classifier's loss function, which is a commonly used loss function in multi-class classification problems. To optimize the model parameters, we used the Adam optimizer, primarily for adjusting the parameters of the fully connected layers. By iteratively training on the dataset, we continuously updated the parameters of the

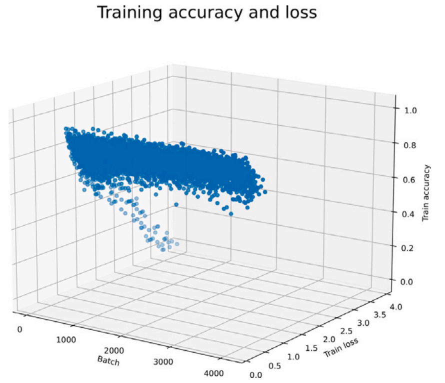


Fig. 6. Classifier training loss function and accuracy.

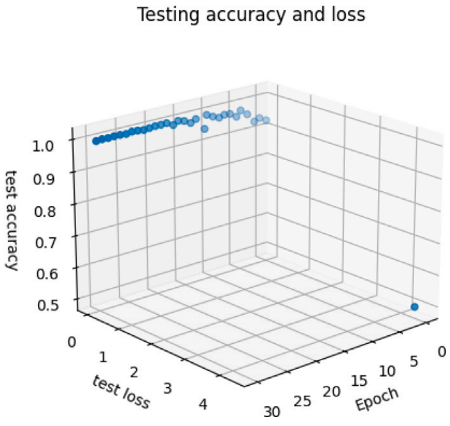


Fig. 7. Test loss and accuracy for each Epoch in the test set.

fully connected layers, allowing the model to gradually adapt to the dataset's features. Throughout the training, the model's objective was to minimize the cross-entropy loss as much as possible.

In summary, through this training process, our model gradually learned the features within the dataset and optimized the classifier by minimizing the cross-entropy loss to achieve improved classification performance.

We employed a learning rate scheduler to dynamically adjust the learning rate during training to optimize the model's convergence and performance. After training, we evaluated the model's performance using the test set. We trained for 30 epochs on a batch size of 16 on 3070 GPUs, reducing the learning rate by half every 5 epochs. Finally, we saved the trained network and incorporated the classifier at the end of Retinaface to assess the face detection performance.

Fig. 6 illustrates the results of our classifier training. The loss decreases as the batch size becomes smaller, and after 1000 batches, the training loss stabilizes, while the accuracy approaches 1. This suggests that our classifier performs well.

Fig. 7 presents the training results on our test set. After one epoch, the test loss approaches zero, and the test accuracy gradually stabilizes close to 1.

Classifier Operation. After Retinaface detects a face, we crop the detected face and feed it into our classifier to determine whether this detected face belongs to fake faces. If it is determined to be a fake face, we do not draw a bounding box around it. If it is not a fake face, we display its detection result in the image as usual and save it.

5.2. *Verification of facial detection effect*

We will add the trained classifier to the Retinaface network after the last step, and use Retinaface's pre trained network and our trained classifier to run the WIDERFACE validation function to obtain the facial label information of the validation set images. Use MATLAB to compare the evaluation indicators of the validation set on the WIDERFACE dataset, and draw a recall accuracy curve. As shown in Figs. 8 and 9, the left table shows the accuracy of each face detection model in each mode, and the left side shows the recall accuracy curve. Among them, our model name is Ours T.

As shown in Fig. 8, Although our enhanced Retinaface can reduce false positives on the FAKEFACE dataset, it also reduces validation results on WIDERFACE. The accuracy results of the original Retinaface in simple, medium, and difficult modes are 0.954, 0.940, and 0.844, respectively. After adding our classifier (Ours-T), the results became: 0.795, 0.788, 0.579. This greatly reduces the performance of the detector, which we believe is due to the classifier only learning facial information from the training images CelebA-HQ and FAKEFACE, and mistakenly identifying some facial information from WIDERFACE as fake facial images.

Therefore, we saved the images detected by Retinaface in the WIDERFACE training set as the positive training images for the classifier and retrained the classifier. On this basis, we re ran the evaluation of Retinaface on WIDERFACE, as shown in Figures 9, and the results met our expectations. Our model remained consistent with Retinaface in simple and moderate modes, with only 0.001 lower than Retinaface in difficult modes. Overall, our enhanced face detection model can ensure the original face detection performance while reducing false positives.

5.3. *False positive rate test*

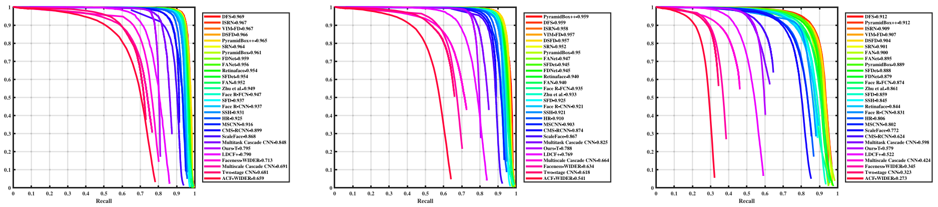
We employed the CelebA-HQ and FAKEFACE datasets as validation sets containing genuine and fake facial images. The CelebA-HQ dataset comprises 1421 validation images, while the FAKEFACE dataset contains 1410 validation images. We assessed the reduction in false positive rates (FPR) by evaluating the YOLO5-Face, RetinaFace, MogFace, and our enhanced RetinaFace models on these two datasets. The results are presented in Table 1 as follows:

As shown in Table 1, positive and negative represent the real face of CelebA-HQ and the "fake" face of FAKEFACE, respectively. "True" indicates that a certain model judges the face images in the dataset as true, and "fake" indicates that the model judges the face images in the "positive" or "negative" dataset as false (for example, 5 in the table means that the Yolo5 face model judges 5 images in the fake face dataset as true). Determine the false positive rate of a person's face detection model based on its recognition rate of positive and negative facial images.

Both Retinaface and Mogface have a false positive rate of 100%, classifying all fake face images as positive. Surprisingly, YOLO5, which performs slightly worse on the WIDERFACE validation set, can detect some fake face images. This suggests that when a model's detection performance is better, it becomes more challenging to detect fake face images, possibly due to extreme regression measures causing regression errors. In contrast, our Model has a false positive rate of only 0.56%, with only 8 out of 1410 fake face images being misclassified. On the other hand, our -T Model, which applies the classifier without incorporating WIDERFACE images in its training, successfully excludes all fake face images. This also confirms our hypothesis that the presence of fake face images in the dataset can affect the detector's robustness.

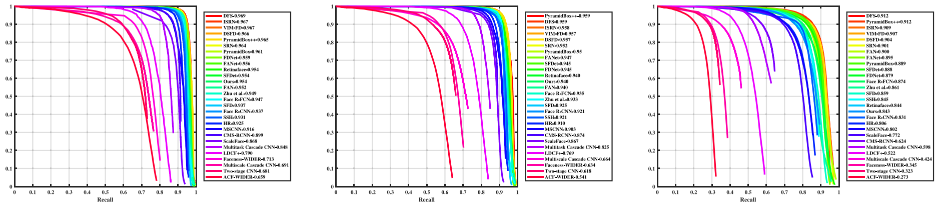
6. Conclusion

In this paper, we focus on another crucial aspect of object detection, the false positive rate. We have curated a non-real facial dataset called FAKEFACE, which provides the opportunity for any face detection model to train or evaluate its capability to discern and exclude fake images. Furthermore, we introduce an approach designed to mitigate the false positive rate of object detection models. Our experiments



(a) Easy (b) Medium (c) Hard

Fig. 8. Using the classifier trained on CelebA-HQ and our FAKEFACE dataset.



(a) Easy (b) Medium (c) Hard

Fig. 9. The classifier trained with the inclusion of face images from the WIDERFACE dataset yields validation results in the WIDERFACE dataset.

Table 1
Our detection model achieves the best performance in fake face validation, making it challenging for other detection models to evade the deception presented by our dataset.

	Yolo5-face		Retinaface		Mogface		Ours-T		Ours	
	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative	Positive	Negative
True	1416	5	**1421**	0	**1421**	0	**1421**	0	**1421**	0
False	1100	310	1410	0	1410	0	0	**1410**	8	**1402**

demonstrate that our method effectively reduces the false positive rate by 99.5% while preserving the inherent facial detection capabilities of the model(only a 1% decrease in difficult modes).

Our viewpoint emphasizes that detection models should not only maintain strong detection performance but also strive to reduce the false positive rate. Additionally, the accuracy of the dataset proves to be a significant factor influencing the model's detection effectiveness.

CRediT authorship contribution statement

Jianlin Zhang: Writing – original draft, Methodology. **Chen Hou:** Formal analysis. **Xu Yang:** Writing – review & editing, Supervision. **Xuechao Yang:** Software, Investigation. **Wencheng Yang:** Writing – review & editing. **Hui Cui:** Visualization, Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (62302203) and Science and Technology Project of Fuzhou Institute of Oceanography (2023F03).

References

[1] Tai Y, Liang Y, Liu X, Duan L, Li J, Wang C, et al. Towards highly accurate and stable face alignment for high-resolution videos. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, (01):2019, p. 8893–900.

[2] Bulat A, Tzimiropoulos G. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230 0003d facial landmarks). In: Proceedings of the IEEE international conference on computer vision. 2017, p. 1021–30.

[3] Pan H, Han H, Shan S, Chen X. Mean–variance loss for deep age estimation from a face. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 5285–94.

[4] Zhang F, Zhang T, Mao Q, Xu C. Joint pose and expression modeling for facial expression recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 3359–68.

[5] Shu Y, Yan Y, Chen S, Xue J-H, Shen C, Wang H. Learning spatial-semantic relationship for facial attribute recognition with limited labeled data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 11 916–25.

[6] Liu Z, Luo P, Wang X, Tang X. Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 3730–8.

[7] Yang J, Luo L, Qian J, Tai Y, Zhang F, Xu Y. Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes. IEEE Trans Pattern Anal Mach Intell 2016;39(1):156–71.

[8] Huang Y, Wang Y, Tai Y, Liu X, Shen P, Li S, et al. Curricularface: adaptive curriculum learning loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 5901–10.

[9] Deng J, Guo J, Xue N, Zafeiriou S. Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 4690–9.

[10] Wang K, Wang S, Zhang P, Zhou Z, Zhu Z, Wang X, et al. An efficient training approach for very large scale face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022, p. 4083–92.

[11] Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, et al. Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 5265–74.

[12] Deng J, Guo J, Zhou Y, Yu J, Kotsia I, Zafeiriou S. Retinaface: Single-stage dense face localisation in the wild. 2019, arXiv preprint arXiv:1905.00641.

[13] Earp SW, Noinongyao P, Cairns JA, Ganguly A. Face detection with feature pyramids and landmarks. 2019, arXiv preprint arXiv:1912.00596.

[14] Yashunin D, Baydasov T, Vlasov R. Maskface: multi-task face and landmark detector. 2020, arXiv preprint arXiv:2005.09412.

[15] Zhang B, Li J, Wang Y, Tai Y, Wang C, Li J, et al. Asfd: Automatic and scalable face detector. 2020, arXiv preprint arXiv:2003.11228.

[16] Zhang S, Chi C, Lei Z, Li SZ. Refineface: Refinement neural network for high performance face detection. IEEE Trans Pattern Anal Mach Intell 2020;43(11):4008–20.

[17] Tang X, Du DK, He Z, Liu J. Pyramidbox: A context-assisted single shot face detector. In: Proceedings of the European conference on computer vision. ECCV, 2018, p. 797–813.

[18] Li Z, Tang X, Han J, Liu J, He R. Pyramidbox++: high performance detector for finding tiny face. 2019, arXiv preprint arXiv:1904.00386.

[19] Najibi M, Singh B, Davis LS. Fa-rpn: Floating region proposals for face detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 7723–32.

[20] Najibi M, Samangouei P, Chellappa R, Davis LS. Ssh: Single stage headless face detector. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 4875–84.

[21] Zhang S, Zhu X, Lei Z, Shi H, Wang X, Li SZ. S3fd: Single shot scale-invariant face detector. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 192–201.

[22] Li J, Wang Y, Wang C, Tai Y, Qian J, Yang J, et al. Dsfd: dual shot face detector. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 5060–9.

[23] Liu Y, Tang X, Han J, Liu J, Rui D, Wu X. Hambox: Delving into mining high-quality anchors on face detection. In: 2020 IEEE/CVF conference on computer vision and pattern recognition. CVPR, IEEE; 2020, p. 13, 043–51.

[24] Chi C, Zhang S, Xing J, Lei Z, Li SZ, Zou X. Selective refinement network for high performance face detection. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, (01):2019, p. 8231–8.

[25] Zhu Y, Cai H, Zhang S, Wang C, Xiong Y. Tinaface: Strong but simple baseline for face detection. 2020, arXiv preprint arXiv:2011.13183.

[26] Wang H, Li Z, Ji X, Wang Y. Face r-cnn. 2017, arXiv preprint arXiv:1706.01061.

[27] Wang Y, Ji X, Zhou Z, Wang H, Li Z. Detecting faces using region-based fully convolutional networks. 2017, arXiv preprint arXiv:1709.05256.

[28] Zhang C, Xu X, Tu D. Face detection using improved faster rcnn. 2018, arXiv preprint arXiv:1802.02142.

[29] Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017, p. 2117–25.

[30] Li J, Qian J, Yang J. Object detection via feature fusion based single network. In: 2017 IEEE international conference on image processing. ICIP, IEEE; 2017, p. 3390–4.

[31] Liu S, Huang D, et al. Receptive field block net for accurate and fast object detection. In: Proceedings of the European conference on computer vision. ECCV, 2018, p. 385–400.

[32] Xu C, Si J, Guan Z, et al. Reliable conflictive multi-view learning. 2024, arXiv preprint arXiv:2402.16897.

[33] Fang U, Li M, Li J, et al. A comprehensive survey on multi-view clustering. IEEE Trans Knowl Data Eng 2023.

[34] Jain V, Learned-Miller E. Fddb: A benchmark for face detection in unconstrained settings. UMass Amherst technical report, Tech. Rep., 2010.

[35] Yang S, Luo P, Loy C-C, Tang X. Wider face: A face detection benchmark. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 5525–33.

[36] Zhang K, Zhang Z, Li Z, Qiao Y. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 2016;23(10):1499–503.

[37] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. Ssd: Single shot multibox detector. In: Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11–14, 2016, proceedings, part i 14. Springer; 2016, p. 21–37.

[38] Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 2980–8.

[39] Chen J-C, Lin W-A, Zheng J, Chellappa R. A real-time multi-task single shot face detector. In: 2018 25th IEEE international conference on image processing. ICIP, IEEE; 2018, p. 176–80.

[40] Chaudhuri B, Vesdapunt N, Wang B. Joint face detection and facial motion retargeting for multiple faces. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 9719–28.

[41] He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 2961–9.

[42] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014, p. 580–7.

[43] Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 1440–8.

[44] Cai Z, Vasconcelos N. Cascade r-cnn: Delving into high quality object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 6154–62.

[45] Wu Y, Tang S, Zhang S, Ogai H. An enhanced feature pyramid object detection network for autonomous driving. Appl Sci 2019;9(20):4363.

[46] Zhang S, Zhu X, Lei Z, Shi H, Wang X, Li SZ. Faceboxes: A cpu real-time face detector with high accuracy. In: 2017 IEEE international joint conference on biometrics. IJCB, IEEE; 2017, p. 1–9.

[47] Qi D, Tan W, Yao Q, Liu J. Yolo5face: why reinventing a face detector. In: European conference on computer vision. Springer; 2022, p. 228–44.

[48] Redmon J, Farhadi A. Yolov3: An incremental improvement. 2018, arXiv preprint arXiv:1804.02767.

[49] Feng Z-H, Kittler J, Awais M, Huber P, Wu X-J. Wing loss for robust facial landmark localisation with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 2235–45.

[50] Honari S, Molchanov P, Tyree S, Vincent P, Pal C, Kautz J. Improving landmark localization with semi-supervised learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 1546–55.

[51] Barron JT. A general and adaptive robust loss function. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 4331–9.

[52] Zhu S, Li C, Loy C-C, Tang X. Unconstrained face alignment via cascaded compositional learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 3409–17.

[53] Bodla N, Singh B, Chellappa R, Davis LS. Soft-nms–improving object detection with one line of code. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 5561–9.

[54] Feng Y, Wu F, Shao X, Wang Y, Zhou X. Joint 3d face reconstruction and dense alignment with position map regression network. In: Proceedings of the European conference on computer vision. ECCV, 2018, p. 534–51.

[55] Zhou Y, Deng J, Kotsia I, Zafeiriou S. Dense 3d face decoding over 2500fps: Joint texture & shape convolutional mesh decoders. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019, p. 1097–106.

[56] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014, arXiv preprint arXiv:1409.1556.

[57] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, p. 770–8.

[58] Liu Y, Wang F, Sun B, Li H. Mogface: rethinking scale augmentation on the face detector. 2021, arXiv preprint arXiv:2103.11139.

[59] Yucel MK, Bilge YC, Oguz O, Ikizler-Cinbis N, Duygulu P, Cinbis RG. Wildest faces: Face detection and recognition in violent settings. 2018, arXiv preprint arXiv:1805.07566.

[60] Choi Y, Uh Y, Yoo J, Ha J-W. Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020, p. 8188–97.

[61] Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: The missing ingredient for fast stylization. 2016, http://dx.doi.org/10.48550/arXiv.1607.08022.

[62] Huang X, Belongie S. Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 1501–10.

[63] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.