Surveillance Face Presentation Attack Detection Challenge

Hao Fang¹, Ajian Liu², Jun Wan¹,²*, Sergio Escalera³, Hugo Jair Escalante⁴, Zhen Lei¹,²,⁵
¹ School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
² MAIS, Institute of Automation, Chinese Academy of Sciences, Beijing, China
³ Universitat de Barcelona and Computer Vision Center, Barcelona, Catalonia, Spain
⁴ Instituto Nacional de Astrofísica, Óptica y Electrónica, Puebla, Mexico
⁵ CAIR, HKISI, CAS
fanghao21@mails.ucas.ac.cn, ajianliu92@gmail.com, jun.wan@ia.ac.cn

Abstract

Face Anti-spoofing (FAS) is essential to secure face recognition systems from various physical attacks. However, most of the studies lacked consideration of long-distance scenarios. Specifically, compared with FAS in traditional scenes such as phone unlocking, face payment, and self-service security inspection, FAS in long-distance such as station squares, parks, and self-service supermarkets are equally important, but it has not been sufficiently explored yet. In order to fill this gap in the FAS community, we collect a large-scale Surveillance High-Fidelity Mask (SuHiFiMask). SuHiFiMask contains 10,195 videos from 101 subjects of different age groups, which are collected by 7 mainstream surveillance cameras. Based on this dataset and protocol-3 for evaluating the robustness of the algorithm under quality changes, we organized a face presentation attack detection challenge in surveillance scenarios. It attracted 180 teams for the development phase with a total of 37 teams qualifying for the final round. The organization team re-verified and re-ran the submitted code and used the results as the final ranking. In this paper, we present an overview of the challenge, including an introduction to the dataset used, the definition of the protocol, the evaluation metrics, and the announcement of the competition results. Finally, we present the top-ranked algorithms and the research ideas provided by the competition for attack detection in long-range surveillance scenarios.

1. Introduction

In recent years, as face recognition systems are widely used, the security of face recognition systems has been increasingly threatened. Therefore, face Presentation Attack Detection (PAD) techniques are crucial in defending against malicious attacks and securing face recognition systems. With the release of some high-quality 2D attack datasets [2, 6, 27, 36, 54–56] and 3D mask datasets [10, 14, 32, 33, 40, 41, 43], existing work [5, 13, 21, 26, 28, 30, 37, 42, 49, 51, 53] has achieved satisfactory performance in short-distance applications. However, these algorithms rely heavily on the quality of face images. As a result, their performance degrades significantly in long-distance scenarios, hindering the expansion of FAS work to surveillance contexts. With the popularity of remote cameras and the deployment of surveillance networks, face recognition technology [19, 20, 57] is gradually expanding to long-distance applications, such as real-time capture, identity verification in security channels, and payment in self-service supermarkets. Therefore, in order to safeguard society and citizens, we require vision technology that can effectively operate in long-distance surveillance scenarios.

However, the FAS community has conducted little research on deceptive face detection in long-distance surveillance environments. We identify two reasons hindering the development of PAD technology: (1) Lack of surveillance scenario-based datasets. Due to the difficulty of building surveillance scenes and expensive acquisition costs, the existing FAS datasets require subjects to face the acquisition device at a certain distance. However, diverse surveillance scenes, realistic weather and lighting, and natural human behavior are important data features of the remote surveillance PAD dataset. (2) Lack of a common benchmark based on remote surveillance dataset to compare the performance of different algorithms. Current research in the FAS community, both algorithms based on color texture feature learning [14, 17, 18, 50] and remote photoplethysmography (rPPG)-based detection [23, 34, 35] require high-quality face images to extract fine-grained features. However, data based on long-distance surveillance scenarios bring new challenges to FAS algorithms such as low res-
olution, motion blur, and occlusion. Therefore, a common benchmark based on remote surveillance datasets to drive algorithm advancement is critical.

In order to facilitate the FAS community’s research on attack detection in long-distance surveillance scenarios, we targeted the following three aspects to address the current difficulties based on the above analysis: (1) We collected a large-scale PAD dataset based on surveillance scenarios, namely SuHiFiMask [11]. Compared to existing short-distance datasets, it has several advantages, such as diverse remote surveillance scenarios, realistic distribution of human faces, a rich pool of spoofing attacks, realistic illumination and diverse weather. (2) We define a more general test protocol for real-world FAS algorithm deployment and publish a public benchmark based on remote surveillance scenarios. We design protocols to evaluate the generality of the algorithms for ID changes, mask type changes, and picture quality changes. (3) Based on the dataset and protocol, we successfully held the Surveillance Face Presentation Attack Detection Challenge at CVPR2023, which attracted 180 teams from all over the world and significantly boosted the community’s research on FAS tasks in surveillance scenarios. A summary of the team names and affiliations that made it to the final stage is shown in Tab 1.

Table 1. Team and affiliations are listed in the final ranking of this challenge.

<table>
<thead>
<tr>
<th>Ranking</th>
<th>Team Name</th>
<th>Leader Name, Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MateoH</td>
<td>Keyao Wang, Baidu Inc</td>
</tr>
<tr>
<td>2</td>
<td>horseGo</td>
<td>Xianhua He, Meituan</td>
</tr>
<tr>
<td>3</td>
<td>OPDAI</td>
<td>Heng Cong, NetEase Inc</td>
</tr>
<tr>
<td>4</td>
<td>SeaRecluse</td>
<td>Minzhe Huang, akivox</td>
</tr>
<tr>
<td>5</td>
<td>ChenyiFan</td>
<td>Yingyu Chen, ioNetworks</td>
</tr>
<tr>
<td>6</td>
<td>XiangR</td>
<td>Xiang Rao, Xiamen University</td>
</tr>
<tr>
<td>7</td>
<td>Dingheng Zeng, MaShang-Consumer Finance Co., Ltd</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ChenyiFan</td>
<td>Xiang Rao, Xiamen University</td>
</tr>
<tr>
<td>9</td>
<td>Keyao Wang, Baidu Inc</td>
<td></td>
</tr>
</tbody>
</table>

2. Challenge Overview

In this section, we review the organized challenge, including a brief introduction to the surveillance-based dataset SuHiFiMask, an introduction to the protocol produced for this challenge, the process and timeline of the challenge, and the evaluation metrics.

2.1. SuHiFiMask Dataset

To the best of our knowledge, SuHiFiMask is the first dataset to extend FAS to real surveillance scenes rather than mimicking low-resolution images and surveillance environments. It contains 10,195 videos, which were taken from 101 subjects of different age groups. For the attack pool, it contains 232 3D high-fidelity masks (plaster, resin, silicone, and head mold), 200 2D attacks (posters, portraits, and screens), and 2 adversarial attacks (adversarial hats, adversarial masks). During the collection process, it considers 40 surveillance scenes for video recording, including daily life scenes (e.g., cafes, cinemas, and theaters) and security check scenes (e.g., security check lanes and parking lots) for deploying face recognition systems. In addition, SuHiFiMask takes into account four different types of weather (Sunny, Windy, Cloudy, and Snowy days) and natural lighting (Day and Night light) to imitate the complex and changing real environment. Finally, seven mainstream surveillance cameras were used to record video simultaneously to ensure complete capture of the entire scene.

In order to facilitate the use of the dataset by the participating teams, we performed the following pre-processing: (1) We used RetinaFace [8] to detect the face in each frame of the original video. (2) We use face similarity to track the position of faces in consecutive frames. (3) We sample each video in 10-frame intervals and store the cropped face image in the corresponding face tracking box folder. (4) We named the folder of this video according to the following rule: Group_Scene_Camera_Epoch_Time.

2.2. Challenge Protocol and Data Statistics

In order to promote the extension of the FAS task to long-range scenarios and to meet the deployment requirements of realistic surveillance networks, we designed a protocol that can evaluate the generalization ability of the algorithm under image quality variations. The inference phase of the algorithm degrades in performance due to the input of low-quality images containing noise, which indicates the inadequate generalization ability of the algorithm. Therefore, to better evaluate the generalization ability of the algorithm under quality variations, we constitute the training set, development set and testing set with images of different quality distributions. In detail, we use SER-FIQ [45] algorithm to calculate the image quality score which ranges from 0 to 1. As shown in the fifth column of Tab. 2, we assigned images with scores [0.4, 1] as the training set, scores [0.3, 0.4] as the development set, and scores [0, 0.3) as the testing set. In addition, to accurately evaluate the robustness of the algorithm to quality changes without being affected by other factors, all mask types (referred to as 1,2,3,4), subjects, scenes, weather (Sunny, Windy, Cloudy and Snowy...
days), and lighting (Day and Night light) were included in the training set, the development set, and the testing set.

Table 2. Statistical information for each protocol of the proposed SuHiFiMask dataset. Note that 1, 2, and 3 in the fourth column mean resin, silicone, and plaster. 4 represents headgear and head mold.

<table>
<thead>
<tr>
<th>Prot</th>
<th>SubSet</th>
<th>#Subj.</th>
<th>Mask</th>
<th>Score</th>
<th>#Live</th>
<th>#Mask</th>
<th>#Others</th>
<th>#All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>101</td>
<td>1~4</td>
<td>[0.4,1]</td>
<td>64,276</td>
<td>35,898</td>
<td>58,889</td>
<td>159,063</td>
<td></td>
</tr>
<tr>
<td>Dev</td>
<td>101</td>
<td>1~4</td>
<td>[0.3,0.4]</td>
<td>37,990</td>
<td>24,031</td>
<td>27,255</td>
<td>89,276</td>
<td></td>
</tr>
<tr>
<td>Test</td>
<td>101</td>
<td>1~4</td>
<td>[0.0,0.3]</td>
<td>83,181</td>
<td>43,087</td>
<td>35,614</td>
<td>161,882</td>
<td></td>
</tr>
</tbody>
</table>

2.3. Challenge Process and Timeline

The challenge was run in the CodaLab\(^2\) platform, and this section will introduce the two stages of this challenge.

2.3.1 Development Phase

*(Started: February. 1, 2023 - Ended: February 24, 2023)*

During this phase, teams have access to labeled training data and unlabeled development data. Participants could use the training data to train their models and submit predictions on the development data. The training data and development data contain all subjects, attack types, surveillance scenarios, weather, lights, and sensors, the only difference is the quality of the images in the training and development sets. In addition, labels for the development set are not provided to participants, who could submit predictions for development data and receive real-time ranking feedback via the leaderboard.

2.3.2 Final Phase

*(Started: February. 24, 2023 - Ended: March 5, 2023)*

In the final phase, the labels of the development set are provided to the teams so that the teams incorporate the development data into the training phase of the model. At the same time, testing sets without labels were published and participants had to make predictions on the test sets and upload the results of their algorithms to the challenge platform. During the final phase, teams will have two opportunities per day to submit their solutions to the challenge platform for the purpose of assessing the stability of their algorithms. Note that the CodaLab platform will display the results of the last submission by default on the challenge website. In addition, to ensure that the final ranking is authentic, teams must publicly release their code under the license of their choice and provide a fact sheet describing their solution.

2.4. Evaluation Metrics

For the performance evaluation, we selected the recently standardized ISO/IEC 30107-3 metrics: Attack Presentation Classification Error Rate (APCER), Normal/ Bona Fide Presentation Classification Error Rate (NPCER/ BPCER) and Average Classification Error Rate (ACER) as the evaluation metric, in which APCER and BPCER/ NPCER are used to measure the error rate of fake or live samples, respectively. The ACER on the testing set is determined by the Equal Error Rate (EER) thresholds on the development set. In addition, the Area Under Curve (AUC) is defined as the area under the ROC curve and its value generally ranges from 0.5 to 1. In this competition, AUC was used as an additional evaluation criterion. Finally, the final ranking of this challenge was determined by the value ACER.

3. Description of Solutions

3.1. MateoH

Considering that there are many indistinguishable samples in the protocol, the team of MateoH proposes a progressive training strategy (PTS) to solve the catastrophic forgetting problem brought by non-stationary data to the deep network. As shown in Fig. 1, the process of PTS is as follows:

1. Sample the initial set of samples. PTS will randomly select a balanced number of positive and negative samples from the training set as the initial training sample set based on the initial sampling rate. The remaining samples are put into the pending set for validation.
2. Hard sample mining. After each training round, the samples in the pending set are predicted with the model and ranked according to their prediction scores. Positive samples with a low score and negative samples with a high score will be added to the training set according to the sampling rate.
3. Reduced sampling rate. A decay factor is used to reduce the sampling rate after each round as a way to gradually increase the focus on hard samples. Overall, the PTS training strategy provides a framework paradigm for progressive hard-sample mining. The strategy allows the model to progressively strengthen its focus on hard samples while playing back previous simple samples to avoid catastrophic forgetting.

To address the issue of limited defense capability of FAS systems against unknown attack types in surveillance scenarios, the team of MateoH proposes a Dynamic Feature Queue (DFQ) module. This module views the negative sample set as a cluster of multiple unknown categories and the positive sample set as a closed sample set. Thus, the binary classification task in FAS is modeled as a multiclassification task to improve the model’s detection of multiple attack types. The training flow of this module is shown in Fig. 1: (1) ViT-Large is used as a backbone network to extract image features. (2) Calculate the cosine similarity log0 of the feature to the negative sample center and
the highest cosine similarity $\log{1}$ of the feature to the negative sample center queue. (3) Concatenate $\log{0}$ and $\log{1}$ as logit and calculate the cross entropy loss of logit and label. (4) Judge the size of $\log{0}$ and threshold $\alpha$, and add the negative samples with $\log{0}$ less than the threshold to the negative sample queue. (5) Remove the queue head element from the queue and update the network parameters.

3.2. CTEL.AI

Since the training, development, and testing sets for this challenge have different image quality distributions, models with insufficient generalization capabilities will experience degraded performance under the influence of quality differences. To address this problem, the CTEL.AI team proposes an adversarial domain generalization method that learns a consistent distribution of deceptive features to avoid models that rely too much on detailed features of high-quality images. Specifically, they delineate the domain using the quality score as the boundary, and consider the training set with the picture score interval of $[0.4, 1]$ as domain one and the development set with the picture quality score interval of $[0.3, 0.4)$ as domain two. By optimizing the adversarial domain loss, the encoder learns to extract consistent quality-independent deceptive features. As shown in Fig. 2, the CTEL.AI team uses the ViT-Large model [9] as the encoder of the solution and adds the gradient reversal layer GRL [12] after the encoder to invert the gradient, thus allowing the parameters of the domain classifier to be optimized in the opposite direction.

In terms of training strategies, SGD was used for gradient descent with a momentum of 0.9 and an initial learning rate (lr) of 0.01. Cosine learning rate decay was employed and the lowest lr is the 1% of initial lr. An epoch was warm-up with a learning rate, and a total of 100 epochs were trained. It is worth mentioning that based on the above training strategy, they fine-tune the model for a total of 30 epochs after completing 100 epochs of training. In the fine-tuning phase, they used different data augmentation methods to expand the amount of data and gradually added it to the training set. The data augmentation methods include Coarse dropout on the image, Random flip, Random rotate, Random crop, Split patch and shuffle, Gray-scale spoof image, Motion blur, Gaussian blur, Sharpen, Down-sampling, Randomly erase and repeat patches, Random fog, Posterize. The data enhanced by different methods are gradually added to the training set to avoid conflicts caused by using too many enhancement methods at the same time, and also to verify the effectiveness of each method.

Finally, the loss of the whole scheme consists of the cross-entropy loss of the main classifier and the adversarial loss of the domain classifier, as follows:

$$L_{total} = L_{cls} + L_{adv}$$ (1)

3.3. horsego

Compared to high-quality images, the training set images of this competition have low resolution, and thus there is little discriminative information in the images for the network to learn. To solve this problem, the team of horsego filtered the original images through a band-pass filter to obtain additional frequency domain information. In addition, they concatenate the spatial domain feature and frequency domain feature of the images before the fully connected layer to fuse them across channels, which makes the binary classification results more accurate. As shown in Fig. 3, they use a dual-stream structure, where one branch performs data enhancement and transformation on the original image to obtain a three-channel image of size 224x224. The other branch uses a band-pass filter to mask some high-frequency information and low-frequency information of the original image, and then converts the image into a filtered map of size 224x224 by inverse fast Fourier transform. Then, two
EfficientFormerV2 [22] are used as encoders to extract frequency domain features and spatial domain features of the processed images, respectively. Finally, these two features are fused across channels and fed to the main classifier for classification, and the cross-entropy loss is used to supervise the main classifier.

![Image](https://via.placeholder.com/150)

Figure 3. The framework of the team horsego. The AI and BI are obtained by data enhancement and band-pass filtering of the original image I. The two encoders then extracted the features of AI and BI separately and concatenated the features. Finally, the fused features are sent to the main classifier and the cross-entropy loss is used for the optimization of the network.

In detail, the steps for data pre-processing through band-pass filters are as follows: (a) The original image undergoes Fast Fourier Transform (FFT) transformation and translation to obtain a frequency spectrum F with energy centralized. (b) Two Gaussian kernels with different variances are subtracted to obtain the band-pass kernel K. (c) The point product of F and K is calculated to obtain the frequency domain band-pass filtered frequency spectrum BF. (d) BF undergoes translation, IFFT (Inverse Fast Fourier Transform) transformation, and is taken as a real value to obtain the band-pass filtering image (BI). In addition, Google’s newly proposed LION optimizer [4] is used for network training, which has faster convergence and higher accuracy compared to other optimizers.

### 3.4. hexianhua

In the acquisition process of the challenge dataset, multiple surveillance cameras capture the same face simultaneously from different distances and angles. The team of hexianhua refers to such images with different perspectives of the same identity as homologous images. They propose a new data pre-processing strategy to effectively simulate homologous images of different quality to expand the training set, so that the model learns features of different quality distributions. As shown in Fig. 4, in the RandomZoomInOut module, the original images are randomly reduced to between 0.20x and 0.75x, and the interpolation algorithms used for the reduction (Nearest neighbor interpolation, Linear interpolation, Area interpolation, Lanczos interpolation) are also chosen randomly. Then, the interpolation algorithms are again randomly selected to resize these low-resolution images to 224×224 to fit the input size of the encoder.

Moreover, considering that the number of live samples and the number of attack samples in the training samples are not balanced, the team of hexianhua uses the data balancing strategy of Max Upsampling to mitigate the performance degradation caused by the long tail problem. As shown in the right side of Fig. 4, they use the total number of categories with more samples as the number of Up-sampling. Then they repeatedly selected images from the set of images in the fewer categories to add to the training set until the number of images in the two categories in the training set is equal. In addition, they use the training method of EMA (Exponential Moving Average) and the data post-processing method of TTA (Test Time Augmentation) to further improve the performance of the model.

Finally, SwinV2-Huge [38] is used as the backbone network of the method, and both cross-entropy loss and Focal loss [24] are used to constrain the encoder, making it give more attention to some hard samples while learning discriminative features. The total loss function formulates as follows:

$$ L_{total} = L_{cls} + 0.5 \cdot L_{focal} $$

(2)

### 3.5. OPDAI

The team of OPDAI proposed a dual-stream network, with ConvNextv2_base and ConvNeXtV2_large [48] as the backbone of the two branches. They also added an MLP for each branch to improve the representativeness of the features extracted from different branches by optimizing the respective classifier parameters. Finally, they concatenate the features extracted by the two extractors and classify the concatenated features with an independent MLP, which makes the features learned by the two encoders more effective after fusion. As shown in Fig. 5, three MLP layers are supervised with focal loss to enhance the model’s focus on difficult samples in surveillance scenarios. Where loss1 is the focal loss of MLP1 for the classification of fused features, and loss2 and loss3 are the focal losses of MLP2 and MLP3 for the classification of branching features, respectively. The total loss function is expressed as follows:

$$ L_{total} = L_{focal1} + 0.5 \cdot L_{focal2} + 0.5 \cdot L_{focal3} $$

(3)

In terms of training strategy, they contribute a scheme to optimize the overall network framework. First, they fixed the parameters of the two feature extractors and pre-trained the three MLPs with a batch size of 512. Then they fine-tuned the whole dual-stream network with a batch size of 200, and the training process continued until the sum of the losses dropped to 0.0001. This has the following two advantages: (1) Allowing the fully connected layer in the MLP
Figure 4. Pre-processing strategies used by the team of hexianhua. RandomZoomInOut data enhancement method randomly picks an interpolation method to shrink the image. The Max Upsampling method mitigates the long tail problem by complementing fewer categories. The AUG represents the various methods.

Figure 5. The framework of team OPDAI. The two branches extract features with different variants of ConvNextV2 respectively, and subsequently, the two features are concatenated as fused features. Finally, the three features are sent to separate MLP modules and supervised with three focal losses.

tical process, they found that the model trained on data of size 224×224 had the best generalization performance for the images in the surveillance scenario.

In general, they used ConvNeXt_base [39] as the backbone network of their method and used the techniques of mix up [52] and label smoothing [44] to improve the generalization of the network to complex data distributions. For the training strategy, they choose SoftTargetCrossEntropy as the loss function and use the CyclicLRd scheduler to let the learning rate cycle between 1e-5 and 0.002. It is worth mentioning the data enhancement scheme considering that image flipping was used in the training phase. During the model inference, they weighted the prediction scores of the original and flipped images as the final prediction result.

3.7. XiangR

In order to explore the association between model size and algorithm performance, the XiangR team at Xiamen University trained ConvNext variants with different parameter scales (ConvNext-T, ConvNext-B, ConvNext-L, ConvNext-XL) under the same experimental setup. They found that the larger model performed better on the dataset of this challenge, but constrained by hardware resources and time constraints, they chose ConvNext-B as the solution backbone network.

Before starting the training, they used various preprocessing methods such as Random Resized Crop, Horizontal Flip, and Random Affine to increase the diversity of the data. Then, he used Mix-up, Transfer Learning, Label Smoothing, and other tricks to improve the model’s ability to generalize data with the complex distribution. For the training strategy, they chose the AdamW optimizer, which is initialized with a learning rate of 0.01 and a weight decay factor of 1E-2. In addition, a cosine annealing learning rate scheduler is used, which gradually decreases the learning rate in a cosine pattern. This scheduler has a maximum number of epochs of 100 and a minimum learning rate of 0. In other words, the AdamW optimizer with weight decay

3.6. SeaRecluse

Since the challenge datasets contain multiple sizes of images, the team of SeaRecluse tries to find a uniform size of the input images that can maximize the performance of the algorithm. Specifically, they trained multiple models simultaneously, and during the training of each different model, they scaled the data in advance to a fixed size such as 112×112, 128×128, 168×168, 192×192, 224×224. Then, they tested the trained model on the validation set and recorded the score for each test image. During the statistical process, they found that the model trained on data of size 224×224 had the best generalization performance for the images in the surveillance scenario.

In general, they used ConvNeXt_base [39] as the backbone network of their method and used the techniques of mix up [52] and label smoothing [44] to improve the generalization of the network to complex data distributions. For the training strategy, they choose SoftTargetCrossEntropy as the loss function and use the CyclicLRd scheduler to let the learning rate cycle between 1e-5 and 0.002. It is worth mentioning the data enhancement scheme considering that image flipping was used in the training phase. During the model inference, they weighted the prediction scores of the original and flipped images as the final prediction result.

3.7. XiangR

In order to explore the association between model size and algorithm performance, the XiangR team at Xiamen University trained ConvNext variants with different parameter scales (ConvNext-T, ConvNext-B, ConvNext-L, ConvNext-XL) under the same experimental setup. They found that the larger model performed better on the dataset of this challenge, but constrained by hardware resources and time constraints, they chose ConvNext-B as the solution backbone network.

Before starting the training, they used various preprocessing methods such as Random Resized Crop, Horizontal Flip, and Random Affine to increase the diversity of the data. Then, he used Mix-up, Transfer Learning, Label Smoothing, and other tricks to improve the model’s ability to generalize data with the complex distribution. For the training strategy, they chose the AdamW optimizer, which is initialized with a learning rate of 0.01 and a weight decay factor of 1E-2. In addition, a cosine annealing learning rate scheduler is used, which gradually decreases the learning rate in a cosine pattern. This scheduler has a maximum number of epochs of 100 and a minimum learning rate of 0. In other words, the AdamW optimizer with weight decay
helps prevent over-fitting, while the cosine annealing scheduler gradually decreases the learning rate to accelerate the model optimization. Finally, the cross-entropy loss is used to supervise the training process of the model.

3.8. chenyifan

Considering that various patches of the face contain deceptive features, the team of chenyifan proposes a high-resolution face patch pipeline based on the original image cropped out, as shown in Fig. 6. The networks on these different branches can focus on learning deceptive traces on a particular face patch, and these traces are used as additional information to assist in the classification of the whole model. In the ready state, the original image is resized to a size of 256 × 256. The bounding box used to crop individual patches is predefined to be no more than 256 × 256. The face boundaries are then used to crop four face patches: face, eyes, nose, and chin. It is worth stating that the original image is also used as a branch input. Then, each patch was resized to 224 × 224 and input to each network branch after mix up [52]. As shown in Fig. 6, the team of chenyifan builds a multi-branch network, including branches for the original image, face, eyes, nose, and chin, each branch has a separate encoder to learn specific features, and the five encoders are efficientnetv2_m, efficientnetv2_b2, efficientnetv2_b1, efficientnetv2_b0, efficientnetv2_b0. Next, they input these five features into the attention module and supervise them with binary cross-entropy (BCE) loss. Finally, these five features are concatenated to obtain the fused deception information and perform classification. The final losses are as follows:

\[ L_{total} = 3 \times L_{ori} + L_{face} + 0.5 \times L_{eyes} \\
+ 0.5 \times L_{nose} + 0.5 \times L_{chain} + 3 \times L_{concat} \]  

(4)

where all the losses are BCE losses. Since patches such as the nose, eyes, and chin contain little deceptive information, they reduce the weights of these loss terms to 0.5 times to reduce the confidence of the whole network in this auxiliary information. For the training strategy, the cosine annealing schedule is used to decay the learning rate, where the initial learning rate is set to 0.0001 and the learning cycle decay is set to 0.5.

3.9. ioNetworks

The team of ioNetworks tries to use feature maps from multiple scales to spoof traces. They use Deep-PixBis [13] as the basic framework, DenseNet as the backbone to reuse features from the previous layers, and fuse feature maps from multiple scales. They utilize BCE to supervise the output feature maps and encourage the network to learn local information about the image patches. As shown in Fig. 7, to improve the robustness of the model to noise in the challenge dataset, inspired by advanced face recognition research [7], they also employ sub-center angular margin loss [16] to supervise the binary output. This loss encourages sub-classes containing clean samples to dominate the optimization of the model thus improving the robustness of the model. The overall loss function is as follows:

\[ L_{total} = 0.5 \times L_{BCE} + 0.5 \times L_{Ang} \]  

(5)

where the second term of this formula is actually a variant of binary cross-entropy loss, which is calculated as:

\[ L_{Ang} = -\frac{1}{n} \sum_{i} (t[i] \times \cos(\theta + 0.5) \\
+ (1 - t[i]) \times \log(1 - \cos(\theta))) \]  

(6)

where \( t[i] \) represents the label of the i-th sample and \( \theta \) represents the angle between the features learned by the backbone network and the weights.

For the training strategy, they used AdamW as an optimizer, with a learning rate of 0.0001 and a weight decay factor of 0.0005. To adjust the learning rate during training, they utilized a scheduler that decreased the learning rate for each parameter group by a factor of 0.8 every 20 epochs. Finally, they implemented early stopping with patience of 50 epochs to halt model training at a better convergence time.

4. Challenge Results

4.1. Challenge Results Report

We adopted four metrics to evaluate the performance of the solutions, which are APCER, NPCER, ACER, and AUC respectively. Please note that although we report performance on various evaluation metrics, ACER is the leading metric that determines the final ranking of the participating
teams. As shown in Tab. 3, which lists the results and ranking of the top 9 teams. We can draw the following three conclusions: (1) The first-place team got the best results on indicators ACER, APCER, AUC, and the second-place team got the best results on BPACER. (2) Eight of the top nine teams are from the industry, indicating that the task of attack detection in surveillance scenarios is important in real-world applications. (3) The top nine teams’ ACER scores were distributed between 4% and 12% and there was some disparity in the results between teams, which shows that face attack detection under surveillance is challenging and proves the value of SuHiFiMask’s in-depth research.

Table 3. Team and results are listed in the final ranking of this challenge.

<table>
<thead>
<tr>
<th>R.</th>
<th>Team</th>
<th>ACER(%)</th>
<th>APCER(%)</th>
<th>BPACER(%)</th>
<th>AUC(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MateoH</td>
<td>4.73</td>
<td>5.07</td>
<td>4.38</td>
<td>98.38</td>
</tr>
<tr>
<td>2</td>
<td>CTEL_AI</td>
<td>5.56</td>
<td>9.20</td>
<td>1.91</td>
<td>98.21</td>
</tr>
<tr>
<td>3</td>
<td>horsego</td>
<td>6.22</td>
<td>8.17</td>
<td>4.26</td>
<td>96.97</td>
</tr>
<tr>
<td>4</td>
<td>hexianhua</td>
<td>7.08</td>
<td>11.21</td>
<td>2.94</td>
<td>97.83</td>
</tr>
<tr>
<td>5</td>
<td>OPDAI</td>
<td>7.16</td>
<td>9.18</td>
<td>5.13</td>
<td>97.38</td>
</tr>
<tr>
<td>6</td>
<td>SeaRecluse</td>
<td>9.89</td>
<td>15.92</td>
<td>3.86</td>
<td>96.02</td>
</tr>
<tr>
<td>7</td>
<td>XiangR</td>
<td>9.96</td>
<td>11.35</td>
<td>8.57</td>
<td>95.80</td>
</tr>
<tr>
<td>8</td>
<td>Chenyifan</td>
<td>11.01</td>
<td>14.12</td>
<td>7.89</td>
<td>94.39</td>
</tr>
<tr>
<td>9</td>
<td>ioNetworks</td>
<td>12.00</td>
<td>15.47</td>
<td>8.53</td>
<td>95.13</td>
</tr>
</tbody>
</table>

4.2. Competition summary and Future Work

Through the challenge, we summarize the effective ideas for surveillance face presentation attack detection: (1) For backbone networks, the performance is superior for networks with a large number of parameters, which proves that large models play an important role in complex applications in real life. (2) At the data level, using pre-processing methods to increase the diversity of data can effectively prevent model over-fitting, and the method of balancing the number of categories can improve the stability of the algorithm. (3) Multi-branching based feature learning method has been shown by several teams to be an effective approach for FAS tasks. In the following work, we further improve the performance from the following aspects: (1) We will explore more efficient super-resolution [3, 46, 47] methods to recover detailed features of low-quality face images. (2) We will generate [15] multiple images with similar data distribution as SuHiFiMask to guide the model to learn more general features. (3) We will explore the interpretability and validity of the large model for FAS tasks.

5. Conclusion

We organized the Surveillance Face Presentation Attack Detection Challenge at CVPR2023 based on the SuHiFiMask dataset and running on the CodaLab platform. 180 teams registered for the competition and 37 teams made it to the final stage. In the final stage of the competition, a total of nine teams submitted their code and participated in the ranking of results after verification by the organizers. We first introduce the associated dataset, the competition protocol, and the evaluation metrics. Then, we review the solutions of the participating ranked teams and report the results of the final phase. Finally, we summarize the conclusions related to the challenges and point out effective methods for attack detection for remote monitoring scenarios.

Acknowledgments

This work was supported by the National Key Research and Development Plan under Grant 2021YFF0602103, the External cooperation key project of Chinese Academy Sciences 173211KYSB20200002, the Chinese National Natural Science Foundation Projects 61876179, 6196160704, and 62276254, the Science and Technology Development Fund of Macau Project (No.~0010/2019/AFJ, 0008/2019/A1, 0025/2019/AKP, 0019/2018/ASC), and the InnoHK program.
References


[25] Ajian Liu, Xuan Li, Jun Wan, Yanyan Liang, Sergio Escalera, Hugo Jair Escalante, Meysam Madadi, Yi Jin,


