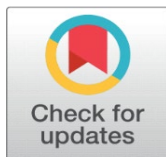


# ADVANCED AI FACE RECOGNITION AND ACCURATE HUMAN TEMPERATURE MONITORING SYSTEM THROUGH INFRARED AND VISIBLE IMAGE FUSION

Hsin-Chun Tsai  , Jhing-Fa Wang 

<sup>1</sup> Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan



Received 20 May 2024  
Accepted 25 June 2024  
Published 09 July 2024

## Corresponding Author

Hsin-Chun Tsai, [tsaihcm@gmail.com](mailto:tsaihcm@gmail.com)

## DOI

[10.29121/ijetmr.v11.i7.2024.1477](https://doi.org/10.29121/ijetmr.v11.i7.2024.1477)

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Copyright:** © 2024 The Author(s). This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

With the license CC-BY, authors retain the copyright, allowing anyone to download, reuse, re-print, modify, distribute, and/or copy their contribution. The work must be properly attributed to its author.



## ABSTRACT

Currently, facial recognition systems need to reveal a person's face to perform recognition. However, some people currently wear masks to prevent respiratory infections. Therefore, this will affect the efficiency of some face recognition systems. Therefore, in this study, we propose a face recognition and body temperature measurement system, and propose an integrated system for masked face recognition, mask detection, and body temperature detection. The proposed system can be used to recognize unoccluded facial images, and we use methods capable of simulating occluded images. The resulting images are used to train the neural network. Experimental results show that the accuracy of naked face recognition reaches 99.79%, and the accuracy of masked face recognition reaches 99.4%. In addition, the mask detection accuracy rate reaches 99.6%. Therefore, the system can improve the accuracy of face recognition and provide efficient security inspection results.

**Keywords:** Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), Neural Network (NN), Support Vector Machine (SVM)

## 1. INTRODUCTION

The current face recognition system has difficulties in identifying faces that are forged or with objects covering the face. The COVID-19 epidemic in recent years has made some people accustomed to wearing masks, or medical staff and patients in some medical places also wear masks. The use of facial recognition in these situations can prevent contact and avoid contagious infections. In addition, since forged faces may result in poor recognition rates, in order to solve the problem of identifying living faces, human body temperature can not only become personal characteristics, but also help the system solve the problem of face misjudgment.

Although facial recognition systems have been used in many places, some systems can only operate under specific conditions and will not be able to correctly identify masked faces in the above-mentioned situations and occasions.

Consequently, we have proposed a face recognition and body temperature measurement system which has four objectives. First, it addresses the situations where facial recognition may be invalid, i.e., when a user is wearing a mask. Second, it detects the presence of a face mask. Third, it measures the user's body temperature using a thermal imager to determine whether the user has a fever. Lastly, it performs liveness detection by integrating visible and thermal images. These objectives are expected to increase convenience, reduce personnel workload, and reduce the risk of infection.

Therefore, this study made two major contributions. First, we propose a masked face image generator that augments training samples and improves face recognition accuracy. Second, we propose a face recognition combined with body temperature monitoring system. This innovative method can solve the problem of not being able to recognize faces when wearing a mask and use a thermal imager to measure body temperature to determine whether the detected object is a living object before performing face recognition, ultimately improving system security and accuracy.

To implement the proposed system, we have analysed the methods proposed in relevant literature and performed a comparison. The remainder of this section is organized as follows: (1) review of masked face dataset, (2) review of occluded-face recognition, and (3) review of liveness detection.

### **1.1. REVIEW OF MASKED FACE DATASET**

There are several publicly available large-scale face datasets, such as MS-Celeb-1M [Guo et al. \(2016\)](#), labelled faces in the wild (LFW) [Huang et al. \(2008\)](#), and CASIA-Web Face [Yi et al. \(2014\)](#). Most images in these datasets originate from the internet. Although these datasets have many face images under unconstrained conditions, they often contain numerous noisy signals. [Ge et al. \(2020\)](#) proposed an image dataset containing occluded faces. It comprises 30,811 images from the internet, including 35,806 labelled occluded faces. The masks in the dataset are not limited to face masks, and include scarves, glasses, and even hands. They used this dataset to enhance facial detection under unconstrained conditions. [Wang et al. \(2023\)](#) proposed a real-world masked face dataset containing unmasked and real masked face images of 525 people and a simulated masked face-recognition dataset generated using the LFW [Huang et al. \(2008\)](#) and Web Face [Yi et al. \(2014\)](#) datasets. However, the number of masked images in the real-world masked face dataset required to train the neural network is extremely small. Moreover, only one mask image was used to generate the simulated masked face-recognition dataset, resulting in insufficient data diversity. Thus far, very few datasets have collected masked facial images to identify individuals. For face recognition under a face mask, a dataset of masked faces is built for individual identities that can be used to train face-recognition models. We used a self-collected face dataset and public face dataset with image synthesis for this purpose. The details are presented in Section 2.

## 1.2. REVIEW OF OCCLUDED-FACE RECOGNITION

Occluded-face recognition is one of the primary challenges in facial recognition. Face recognition is realized through the extraction and comparison of key features of the face. The completeness of facial features is a critical factor that determines success or failure. Occlusion causes the disappearance of certain facial features, presenting in-complete facial features. Under this condition, the facial features cannot be compared with the face information in the database, resulting in recognition failure. [Zeng et al. \(2021\)](#) divided face occlusion into five categories: (1) real-world occlusion, such as those caused by scarves, sunglasses, and other accessories; (2) partial face occlusion, containing only a part of the face; (3) synthetic occlusion, which simulates the occlusion that may occur in the real world through image synthesis; (4) rectangular occlusion, such as using white or black rectangular blocks to overlay the original image; and (5) occlusion caused by irrelevant images, such as overlaying a non-face-related image over the face image. [Hariri \(2015\)](#) divided the approaches to occluded-face recognition into three categories:

- 1) Match-based approach
- 2) Occlusion removal-based approach
- 3) Occlusion-recovery-based approach

**1) Match-based approach:** This approach involves matching the similarities between images. In general, a facial image is segmented into small patches and feature extraction is performed on each patch. Thus, the occluded area does not affect other intact facial regions. [Cheheb et al. \(2017\)](#) first divided the facial images into multiple patches. Subsequently, they used a local binary pattern to obtain the local descriptor of each patch. They then trained multiple sub-support vector machine (SVM) classifiers by randomly selecting patches. Finally, the results of several subclassifiers were combined to improve the recognition performance. [Trigueros et al. \(2018\)](#) used visualization techniques to identify areas of the face that were more effective for identification when using convolutional neural networks (CNNs). They then occluded these areas during training to increase the robustness of the model-to-face occlusions.

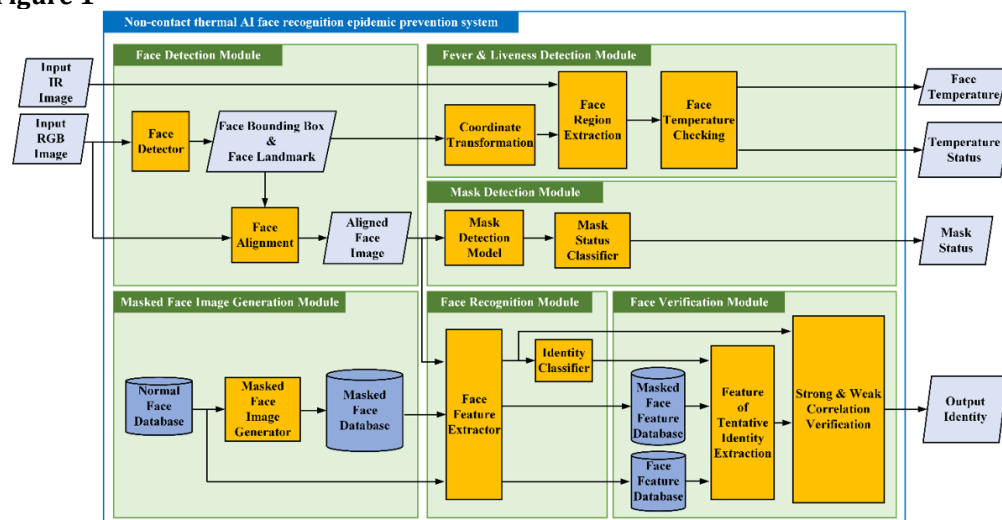
**2) Occlusion-removal-based approach:** This approach removes the occluded portion from the original image. Only facial areas can be used for recognition. For masked face recognition, [Hariri \(2015\)](#) first aligned the facial image to ensure that the eyes were horizontal. Following adjustments of the image to a specific size, the bottom half of the image containing the face mask was removed. Only the eye and forehead areas were included in the subsequent identification tasks. [Qiu et al. \(2022\)](#) reported that the human visual system subconsciously ignored occluded parts to focus on other unoccluded parts. Therefore, they proposed a Face Recognition with Occlusion Masks (FROM). The FROM learns to discover the corrupted features from the deep convolutional neural networks and clean them by the dynamically learned masks.

**3) Occlusion recovery-based approach:** This approach performs general face recognition by restoring an occluded area to its unoccluded appearance. [Zhao et al. \(2017\)](#) proposed a robust long short-term memory (LSTM)-autoencoder (RLA) model to recognize occluded faces. It

comprised two LSTM components. The multiscale spatial LSTM encoder generated feature representations with occlusion robustness. The other two-channel LSTM decoder iteratively restored the occluded face and performed occluded area detection. Following the removal of occlusion, the occluded-face recognition performance was effectively improved.

There are several excellent face-recognition methods, such as SphereFace [Liu et al. \(2017\)](#), CosFace [Wang et al. \(2018\)](#), and ArcFace [Deng et al. \(2019\)](#). They improve the inter-class separability and intra-class compactness by adding angular intervals to the loss function. The loss functions of these three studies are similar apart from the placement of the angle interval. They achieved good results in general face recognition but did not address masked faces. Based on the findings of [Trigueros et al. \(2018\)](#), we used the simulated masked face database mentioned in Section 3.2 to train our deep-learning model and perform face verification. In addition, a face-verification approach was proposed to improve the facial recognition performance of the system. Moreover, we addressed masked faces through by integrating these two technologies into a general face-recognition system.

**Figure 1**



**Figure 1** Proposed System Architecture

### 1.3. REVIEW OF LIVENESS DETECTION

Spoofing attacks have emerged with the widespread use of biometric authentication technologies. This has resulted in security and privacy concerns regarding the system. A secure system requires anti-spoofing techniques such as liveness detection to prevent improper usage. Different biometric authentication technologies incorporate corresponding liveness detection approaches, such as fingerprint liveness [Sharma & Selwal \(2022\)](#) and face liveness detection. This approach can be implemented using the motion, texture, and deep-learning-based approaches.

A motion-based approach extracts the motion information from a sequence containing numerous frames, such as blinking and head rotation. These behaviours can be used to distinguish whether a face is a spoof [Birla & Gupta \(2022\)](#), [Kamanga & Lyimo \(2022\)](#). Texture-based approaches distinguish the spoofs through the extraction of handmade texture features. [Määttä et al. \(2011\)](#) asserted that face prints usually contained defects that can be detected by texture features. Therefore,

they used a local binary pattern to analyse facial textures and an SVM classifier to classify real and fake faces. The deep-learning- based approach uses anti-spoofing features learned by deep neural networks to distinguish spoofs. [Atoum et al. \(2017\)](#) combined two CNN models for liveness detection. One model distinguished real faces from spoof images using local features. They assumed that face prints would have a flat depth map. Therefore, another model was used to estimate the depth of the face images. Finally, the results of the two models were combined to determine if they were spoofs.

Although these methods have achieved good results, they face certain limitations. Among the motion-based approaches, [Teh et al. \(2022\)](#) believed that using blinking to determine whether a person is alive has a relatively limited effect on the video and is heavily influenced by the

reflection of glasses. [Yang et al. \(2014\)](#) believed that a texture-al-feature-based approach may not be adequate to learn the most discriminative information for distinguishing a spoof. Deep- learning-based approaches consume considerable computing resources. Since we perform human temperature measurements in our system, we believe that the measured temperature can also be used to perform liveness detection. Although an additional hardware requirement is necessary when compared to previous approaches, a thermal camera is necessary for temperature measurement. Therefore, the cost of liveness detection can be ignored.

## 2. PROPOSED METHOD

### 2.1. SYSTEM OVERVIEW

[Figure 1](#) depicts the architecture of the proposed system. It comprises six parts: (1) masked face image generation, (2) face detection, (3) face recognition, (4) face verification, (5) mask detection, and (6) fever and liveness detection modules. The inputs of the system were visible and thermal images, and the outputs were temperature, temperature state, mask state, and identity.

**Table 1**

Table 1 Presents the Pseudo Code of our Proposed System Algorithm
<b>The pseudo code of our proposed system algorithm.</b>
<b>Input:</b> input IR images $\{ir_i\}$ and RGB images $\{x_i\}$
<b>Output:</b> The face temperature $\{T_i\}$ , liveness status $\{S_i\}$ , mask status $\{M_i\}$ and Identity results $\{O_i\}$ .
1: While not end do
2: $i \leftarrow i + 1$
3: Run Face Detection Module for each $\{x_i\}$ 4: Output face region $\{R_i\}$
5: Run Fever & Liveness Detection Module for each $\{ir_i\}$
6: Combine face region $\{R_i\}$ and IR face region $\{ir_i\}$ to get face temperature $\{T_i\}$ 7: Output face temperature $\{T_i\}$ and liveness status $\{S_i\}$
8: Run Mask Detection Module for each face region $\{R_i\}$ 9: Output mask status $\{M_i\}$
10: Run Face Recognition Module for each $\{R_i\}$ 11: Output recognition results $\{C_i\}$
12: Input $\{C_i\}$ to run Face Verification Module
13: Output Identity results $\{O_i\}$
14: end while

**1) Masked Face Image Generation Module:** In this module, the input was an unmasked face image. Facial landmarks were obtained after passing through the face detector. Subsequently, the facial landmarks were analysed to

determine the scale and rotation angle of the face mask image. Finally, the face mask image was pasted onto the input image through image processing to complete masked face generation.

- 2) **Face Detection Module:** In this part, the input was an image captured by the webcam. Facial landmarks were obtained after passing through the face detection network. The input image was then cropped and aligned with facial landmarks. Finally, the module outputs the facial landmarks and the aligned facial image for use in subsequent tasks.

Figure 2

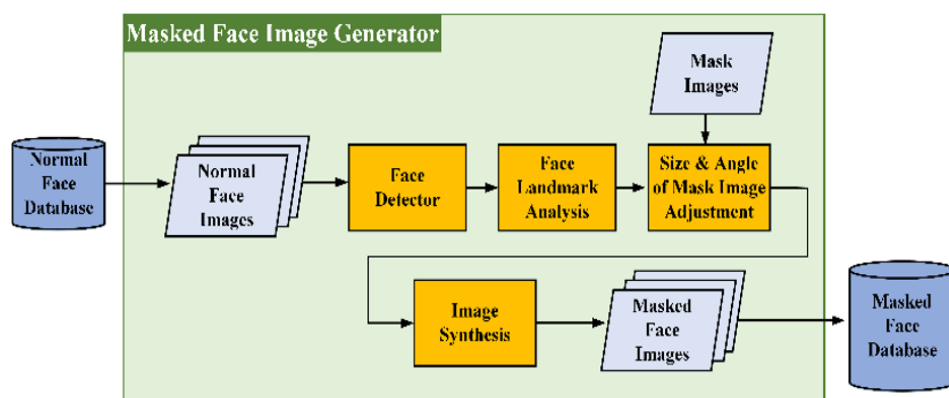


Figure 2 Flowchart of Masked Face Image Generator

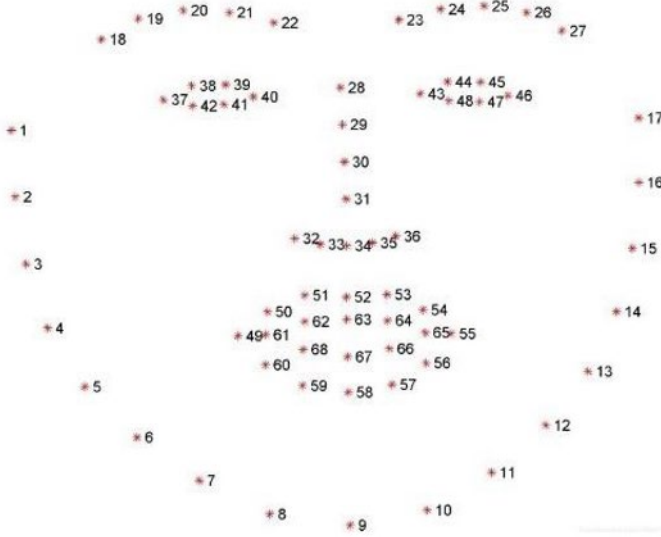
- 3) **Face-Recognition Module:** The input is a facial image that is aligned by the face detection module. The facial features were obtained after the face image was passed through the facial feature extraction network. The facial image was then identified using an identity classifier. Finally, the module outputs the tentative identity and the facial feature for use by the Face Verification Module.
- 4) **Face-Verification Module:** This module used the system's facial feature database and a masked facial feature database to verify the output of the tentative identity from the face- recognition module. If the tentative identity passes the verification, the module outputs it as the system's identity recognition result. Otherwise, the unknown identity is generated as the output.
- 5) **Mask-Detection Module:** The input of this module is the aligned facial image output from the face detection module. The presence of a mask can be determined after passing the input image through the mask-detection network and mask status classifier.
- 6) **Fever & Liveness Detection Module:** This module converted facial landmarks into thermal image coordinates through a coordinate transformation. The coordinates were then used to extract the facial region from the entire thermal image. Finally, the facial temperature was examined and the temperature value and status of the user were generated as output.

## 2.2. MASKED FACE IMAGE GENERATION MODULE

We implemented four primary processes in this module: (1) face detection, (2) face landmark analysis, (3) size and angle of mask image adjustment, and (4) image synthesis. Figure 2 presents a flowchart of this process. We used a face detector from

the DLib machine learning library to detect faces. We then used a 68-point facial landmark detection model to obtain the facial landmarks. By analyzing the predicted facial landmarks, we obtained the scaling and rotation angles of the mask image. Finally, we performed image synthesis to determine the effect of wearing a mask on a bare face. We masked the unmasked face database with this module to generate a simulated masked face database, which was then used in the face-verification module and feature extraction model training.

**Figure 3**



**Figure 3** 68 Facial Landmarks

- 1) Face Landmark Analysis:** We used a 68-point facial landmark extractor provided by Dlib for facial landmark analysis. [Figure 3](#) depicts the 68-point facial landmarks. The landmarks of both eyes (points 37 and 46) were used to determine facial rotation. Assuming that the coordinates of the left eye are ( $X_{left}$ ,  $Y_{left}$ ) and those of the right eye are ( $X_{right}$ ,  $Y_{right}$ ), the angle between the two eyes can be expressed as follows:

$$\theta = \tan^{-1} \left( \frac{Y_{left} - Y_{right}}{X_{left} - X_{right}} \right) \quad (1)$$

The face mask image was rotated using this angle. Points 4 and 14 determine the width of the face mask image, and Points 9 and 30 determine the height of the face mask image. Furthermore, Points 30, 9, 49, and 55 determine the center of the mouth.

- 2) Image Synthesis:** We adjusted the angle and size of the face mask image using the settings explained in Section 2.2.1. The center of the mask was then overlaid on the center of the mouth using image synthesis to create a simulated masked face image. [Figure 4](#) presents the samples of mask images. [Figure 5](#) presents some unmasked and simulated masked facial images generated using this module. The unmasked facial image samples were obtained from the VGGFace2 [Cao et al. \(2018\)](#) dataset.

**Figure 4****Figure 4** Samples of Face Mask Image

### 2.3. FACE DETECTION MODULE

**This module comprised two primary processes:** (1) face detection and (2) face alignment. [Figure 6](#) presents a flowchart of this process. Firstly, we used a single-stage headless face detector [Najibi et al. \(2017\)](#) to detect faces during the face detection stage. The input image was then passed through the face detection model to obtain the facial bounding boxes. In the face alignment stage, we used facial bounding boxes and landmarks to crop the facial region out of the original input image and calculate the rotation angle of the face. We then rotated the cropped facial image using the rotation angle. Finally, the module outputs the aligned facial images and landmarks for subsequent tasks.

**Figure 5****Figure 5** Samples of Normal Face and Simulated Masked

- 1) Face Alignment:** Facial rotation has always been a significant challenge in facial recognition. The accuracy of the other tasks may be affected if it is not addressed appropriately. The rotational changes are classified into yaw, roll, and pitch. Before outputting the extracted facial images to the other modules, face alignment must be performed for subsequent tasks to proceed smoothly.

A complete facial image cannot be obtained if the face is affected by the self-occlusion of yaw and pitch changes. We can only train the model for subsequent tasks by increasing the database to avoid the influence of these two types of rotations to the maximum possible extent. Facial alignment can be achieved using facial landmarks to manage roll changes. We used the same approach as that in



Section 2.2.1 to determine the angle of the two eyes. Using this angle, we can rotate the original image by an angle of  $\theta$  through affine transformation to complete the face image alignment and send it to the subsequent module for other tasks.

Figure 6

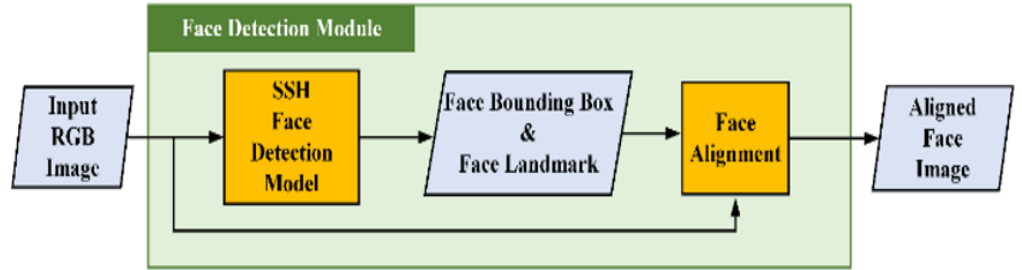


Figure 6 Flowchart of Face Detection Module

## 2.4. FACE RECOGNITION MODULE

This module comprised two primary modules: (1) facial feature extraction and (2) identity classification. Figure 7 presents a flowchart of this process. First, we used ResNet He et al. (2016) trained with an additive angular margin loss Deng et al. (2019) to extract the features of the input face. Table 2 lists the network architecture. Finally, the network outputs a 512-D feature, that is, the facial feature of the input image. We then calculate the similarities between the extracted features and the features of each identity in the face database.

Figure 7

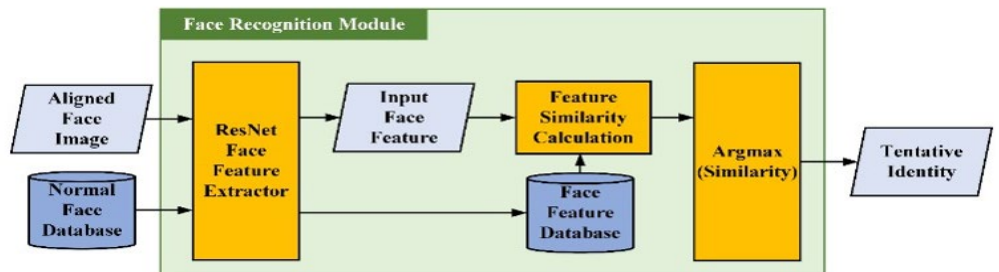


Figure 7 Flowchart of Face-Recognition Module

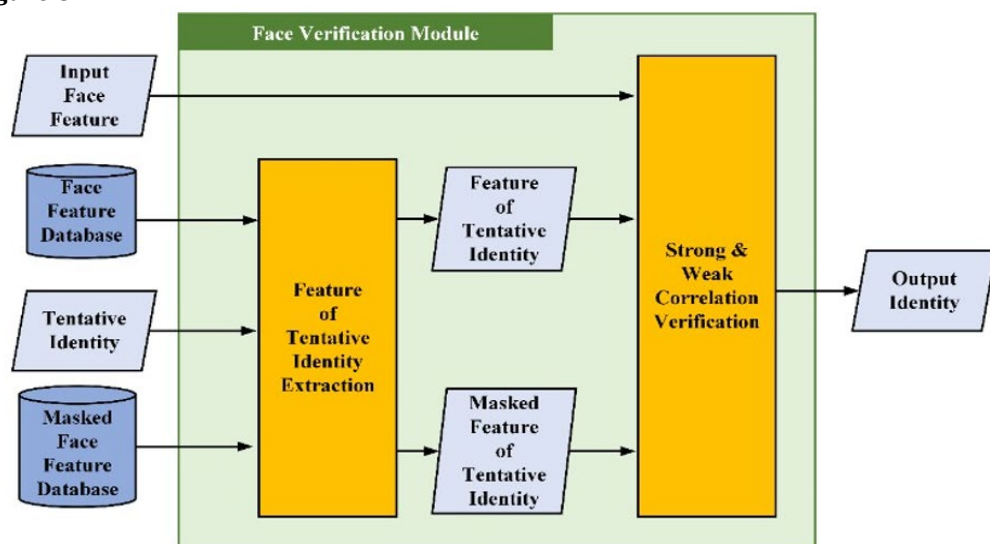
Table 2

Table 2 Architecture of Resnet 100		
Layer name	Output size	ResNet 100
Conv1	112 × 112	3 × 3, 64
Conv2_x	56 × 56	3 × 3, 64     × 3 3 × 3, 64
Conv3_x	28 × 28	3 × 3, 128     × 13 3 × 3, 128
Conv4_x	14 × 14	3 × 3, 256     × 30 3 × 3, 256
Conv5_x	7 × 7	3 × 3, 512     × 3 3 × 3, 512
FC	512	512

**1) Identity Classification:** Our system requires one unmasked and one simulated masked face image for each identity stored in the face image databases. We then extracted the facial features from the facial images of each identity in the database and stored them in the unmasked and simulated facial feature databases. We calculated the cosine similarities between the input facial features and the features in the unmasked facial feature database individually. The simulated facial feature database was used for face verification.

In the identity classification stage, we used the concept of a prototype network [Snell et al. \(2017\)](#), which uses relative distance as the basis for classification, to determine the identity with the highest similarity to the identity of the input image. Each class has a prototype representation; in our approach, it is the facial features of each identity in the database. The classification problem involves identifying the nearest neighbours in the feature space.

**Figure 8**



**Figure 8** Flowchart of Face-Verification Module

## 2.5. FACE VERIFICATION MODULE

This module verified the face-recognition results. [Figure 8](#) present a flowchart of this process. This module contains two primary processes: (1) tentative identity facial feature extraction and (2) strong and weak correlation verification. We use the facial features of the tentative identity stored in the database to verify the input facial features in the two stages. If the verification is validated, the module outputs the identity; otherwise, the output is an unknown identity.

**1) Strong & Weak Correlation Verification:** In Section 2.4.1, we explained that the identity classification of the face-recognition module is based on the relative distance between the input facial feature and the feature of each identity in the database. However, relying only on the relative distance can result in classification errors. Therefore, cooperating with verification can increase the accuracy. Facial verification verifies whether the absolute distance exceeds the set threshold value to determine whether two features belong to the same identity. Unlike setting a high threshold value, which renders it challenging for the masked face to pass the verification, setting a

low threshold value slightly improves the accuracy. Therefore, we propose strong and weak correlation verifications to address this issue. This process is illustrated in Figure 9.

Figure 9

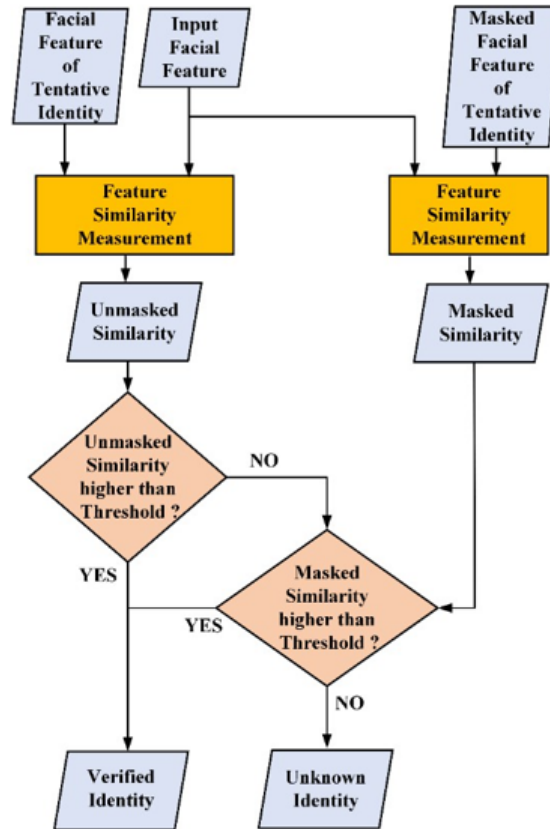


Figure 9 Flowchart of Strong and Weak Correlation Verification

Figure 10

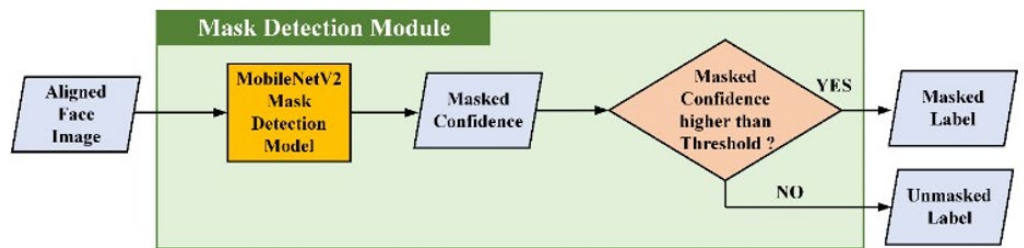


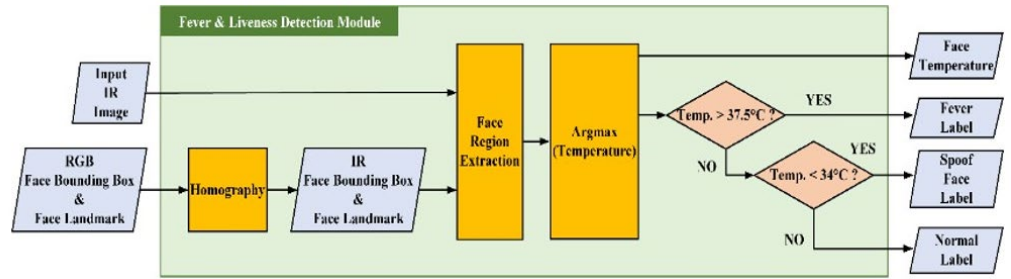
Figure 10 Flowchart of Mask-Detection Module

The first step in both the verifications was unmasked face matching. We calculated the similarity between the input and unmasked facial features of the tentative identity. Subsequently, we verify whether the similarity is higher than the threshold value. If this condition is satisfied, we believe that the input feature is strongly related to the tentative identity. If the input face feature fails the strong correlation verification, masked face matching is performed. We calculated the similarity between the input features and simulated masked facial features of the tentative identity. We then verified the masked face similarity. If it was higher than

another threshold, we inferred that the input feature is weakly correlated with the tentative identity. If the input face passed one of the verifications, we determined whether the input feature belongs to that identity. If the input face failed both verifications, we considered it as an unknown identity.

Since the similarity of unmasked face matching is calculated in the face-recognition module, the proposed face-verification approach only needs to calculate the similarity between the input feature and masked facial feature of the tentative identity. When compared to the calculation amount of face recognition, the increased calculation amount of the face-verification approach accounts for only a small part of the total calculation.

**Figure 11**



**Figure 11** Flowchart of Fever and Liveness Detection Module

## 2.6. MASK DETECTION MODULE

This module comprised two primary processes: (1) mask-wearing confidence calculation and (2) mask-status classification. Figure 10 presents a flowchart of this process. Since mask detection is relatively simpler than both facial detection and recognition, we choose a lightweight network to reduce the number of calculations in the system. In the mask-wearing confidence calculation stage, we used MobileNetV2 Sandler et al. (2018) to calculate the confidence of the input face wearing the mask. In the mask status classification stage, the module detected the presence of a mask based on whether the confidence exceeded the threshold.

After the input image is calculated using MobileNetV2, different output dimensions are obtained based on the number of categories to be classified. The output is two-dimensional since we determine whether the user is wearing a mask. We used the softmax activation function at the end of the neural network to calculate the confidence. The softmax activation function is expressed as follows Sandler et al. (2018):

$$\partial(Z)_j = \frac{e^{-z_j}}{\sum_{k=1}^K e^{-z_k}}, j = 1, \dots, K \tag{2}$$

The softmax activation function compresses a K-dimensional vector, Z, with arbitrary real numbers into another K-dimensional real vector,  $\partial(Z)$ . Each element of  $\partial(Z)$  ranges between 0 and 1, and all the elements sum to 1. This is the probability of each category; that is, the probability of wearing a mask versus the probability of not wearing a mask.

We use the binary cross entropy loss function defined as follows:

$$loss = - \sum n \hat{y}_i \log y_i + (1 - \hat{y}_i) \log(1 - y_i) \quad (3)$$

$$\frac{\partial loss}{\partial y} = \sum_{i=1}^n \frac{\hat{y}_i}{y} \frac{(1-\hat{y}_i)}{(1-y_i)} \quad (4)$$

where  $\hat{y}_i$  denotes the target label and  $y_i$  denotes the predicted probability value. This loss function evaluates the difference between the probabilities. Loss is zero only when  $y_i$  and  $\hat{y}_i$  are equal. Otherwise, it is positive. The larger the difference between  $y_i$  and  $\hat{y}_i$ , the larger the loss. For example, if the input image is a face with a mask,  $\hat{y}_i = 1$ .  $y_i$  is the probability that the model believes that the user is wearing a mask. The greater the difference between the predicted probability and 1, the greater the loss.

## 2.7. FEVER AND LIVENESS DETECTION MODULE

This module comprised four primary processes: (1) homography, (2) facial region extraction, (3) determination of the highest temperature, and (4) checking the temperature. Figure 11 presents a flowchart of this process. We obtained the facial bounding box on the thermal image after performing homography using the facial bounding box of the visible image. The facial bounding box in the thermal image was combined with the thermal image to extract the facial region from the image. The highest temperature value in the facial region is the user's forehead temperature since the pixel intensity of the thermal image corresponds to the temperature. Finally, the temperature status of the person was determined by verifying where the temperature falls within the specified range. If the temperature exceeds 37.5 °C, the person may have a fever. If it is lower than 34 °C, the person is considered to be a spoof, and if it is in between these values, the person has a normal temperature.

All the images used in the previous modules were visible images. In this module, we introduce thermal images captured using a thermal camera for temperature measurements. However, even if two cameras with different fields of view are placed close together, the images obtained by the two cameras capturing the same object cannot be directly superimposed. Therefore, we cannot share information regarding the thermal and visible images. Figure 12 depicts an example of directly superimposing visible and thermal images before processing. Notably, the visible and thermal images are not completely overlapped. Although we can obtain the face bounding box from a visible image, the bounding box cannot perfectly encompass the facial region in the thermal image. Therefore, homography must be performed to ensure that the two images overlap to the maximum possible extent.

**1) Homography:** In the field of computer vision, two images of the same planar surface in space are related using homography. The corresponding points of the two images can be mapped using a homography matrix ( $3 \times 3$  matrix). The homography matrix is expressed as follows:

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \quad (5)$$

The formula for mapping the corresponding point (x1, y1) in the left image to the corresponding point in the right image can be expressed as follows [28]:

$$\begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = H \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} \tag{6}$$

Homogeneous coordinates are used instead of Cartesian coordinates for matrix multiplication to express the spatial translation, scaling, and rotation. As the homogeneous coordinates can be scaled arbitrarily, the homography matrix can be expressed as follows Bradski & Kaehler (2008):

$$H = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & 1 \end{bmatrix} = \begin{bmatrix} h'_{00} & h'_{01} & h'_{02} \\ h'_{10} & h'_{11} & h'_{12} \\ h'_{20} & h'_{21} & 1 \end{bmatrix} \tag{7}$$

**Figure 12**



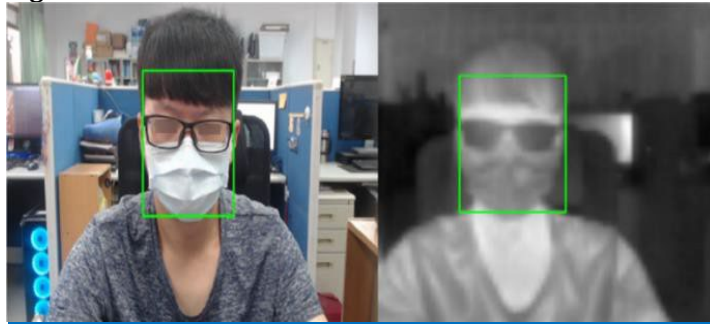
**Figure 12** Thermal Image and Visible Image Superimposed Before Homography

The revised homography matrix contains eight parameters. We can derive this matrix using four corresponding points in the two images related to homography. The coordinates of the same plane in the two images can be mapped arbitrarily by using the homography matrix.

**Figure 13**



**Figure 13** Thermal Image and Visible Image Superimposed after Homography

**Figure 14****Figure 14** Coordinate Transformation of Facial Bounding Box

To align the visible image with the thermal image using homography, we allowed the webcam and thermal camera to shoot an object with a temperature higher than the background temperature to manually mark the corresponding points. After determining the four corresponding points, we calculate the homography matrix by using the function provided by the OpenCV library [Bradski & Kaehler \(2008\)](#) and finally align the visible light image with the thermal image. [Figure 13](#) presents the result. Notably, the error is significantly smaller than the error depicted in [Figure 12](#).

After obtaining the homography matrix which converts the coordinates of the thermal and visible images, we use it to convert the coordinates of the face bounding box. We obtain the facial bounding box on the thermal image by using homography for the coordinate transformation of the facial bounding box of the visible image, as shown in [Figure 14](#). Thus, we determine the highest temperature in the facial region in the thermal image as the user's temperature and determine the user's temperature status by verifying the temperature.

### 3. EXPERIMENTAL RESULT

#### 3.1. UNMASKED FACE-RECOGNITION ACCURACY TEST

We train two models for testing: one with the simulated masked face database and the other without the simulated masked face database.

For the model trained without the simulated masked face database, we use the self-collected and VGGFace2 datasets. The training data contains 50 identities, of which 28 are obtained from the self-collected dataset, and the remaining are obtained from the VGGFace2 dataset. The self-collected dataset contains 12188 unmasked face images from different angles. The VGGFace2 dataset contains 6295 face images under unconstrained conditions. For the model trained with the simulated masked face database, we additionally use 4586 simulated masked face images generated from both datasets.

We used the Large-scale Celeb Faces Attributes (CelebA) [Liu et al. \(2015\)](#), VGGFace2, and self-collected datasets for testing. None of the test images were included in the training dataset. For the CelebA test set, we obtained 100 identities from the original database, totalling 2,274 images. For the VGGFace2 test set, we used 90 identities with 29077 images. The self-collected test set contained 9 identities with 13,433 images captured in the laboratory. The testers did not wear a mask. They slightly rotated and moved their faces during shooting to increase the reliability of the test.

One image of each identity to be recognized was used as the face-recognition database, that is, the output of the face recognition will only be an unknown identity or one of these identities. For the CelebA and VGGFace2 test sets, the face-recognition database contained only the identities in the test set. For the tests of the self-collected test set, the face-recognition database contained 81 identities of the VGGFace2 test set in addition to the 9 identities tested to increase the reliability of the test. If the classification result was determined to be of an unknown identity, it was excluded from the accuracy calculation.

$$Accuracy = \frac{\#Correct\ images}{\#Test\ images - \#Unknown\ Identity} \times 100\%$$

(8)

**Table 3**

Table 3 Accuracy on Unmasked Celeba Test Set					
Setting	#Test	#Unknown	#Correct	#Wrong	Accuracy
Origin	2274	3	2261	10	99.56%
Origin+SM	2274	3	2261	10	99.56%
Origin+Ver	2274	21	2252	1	99.96%
Origin+SM+Ver	2274	18	2255	1	99.96%

**Table 4**

Table 4 Accuracy on Unmasked Vggface2 Test Set					
Setting	#Test	#Unknown	#Correct	#Wrong	Accuracy
Origin	29077	432	27349	1296	95.48%
Origin+SM	29077	328	27349	1400	95.13%
Origin+Ver	29077	2944	26098	35	99.87%
Origin+SM+Ver	29077	2762	26261	54	99.79%

**Table 5**

Table 5 Accuracy on Unmasked Self-Collected Test Set					
Setting	#Test	#Unknown	#Correct	#Wrong	Accuracy
Origin	13433	0	13433	0	100%
Origin+SM	13433	0	13433	0	100%
Origin+Ver	13433	22	13411	0	100%
Origin+SM+Ver	13433	17	13416	0	100%

Table 6 presents the unmasked test results for the four experimental settings of the three test sets. The experimental results demonstrate that, even if the model is trained with the SMFD or not, adding face verification can improve the accuracy. The harder the test set, the greater the improvement in the accuracy. Adding facial verification improves the accuracy by 0.4% in the celebA test and at least 4.39% in the VGGFace2 test, whereas no effect is observed in the tests on the self-collected dataset.



**Table 6**

Table 6 Accuracy in Unmasked Face Test			
Accuracy	CelebA	VGGFace2	Self-collected dataset
Origin	99.56%	95.48%	100%
Origin+SM	99.56%	95.13%	100%
Origin+Ver	99.96%	99.87%	100%
Origin+SM+Ver	99.96%	99.79%	100%

Table 7 presents a comparison of the proposed system with other methods. The proposed system was implemented using ArcFace and the proposed face-verification approach. We trained all the test models using the SMFD. Although the proposed face-verification approach was designed for masked face recognition, the experimental results demonstrated that it could also improve the accuracy of unmasked face recognition.

**Table 7**

Table 7 Accuracy Compared to Other Methods in Unmasked Face Test		
Accuracy	CelebA	VGGFace2
SphereFace Liu et al. (2017) + SMFD	99.43%	94.92%
CosFace Wang et al. (2018) + SMFD	99.52%	95.32%
ArcFace Deng et al. (2019) + SMFD	99.56%	95.13%
(Origin+SM in above table)	99.96%	99.79%
Proposed system (Origin+SM+Ver in above table)		

### 3.2. MASKED FACE-RECOGNITION ACCURACY TEST

We used the masked face image generation module on the CelebA and VGGFace2 test sets and used the generated masked face images for testing. A total of 2210 and 26161 images were generated from the CelebA and VGGFace2 datasets, respectively. The self-collected test set contained 9 identities and 18026 images captured in the laboratory. The testers wore masks, slightly rotated, and moved their faces during shooting to increase the reliability of the test. The experimental settings and combinations were identical to those used in the unmasked face accuracy test.

**Table 8**

Table 8 Accuracy on Simulated Masked Celeba Test Set					
Setting	#Test	#Unknown	#Correct	#Wrong	Accuracy
Origin	2210	5	2172	33	98.50%
Origin+SM	2210	2	2187	21	99.05%
Origin+Ver	2210	91	2111	8	99.62%
Origin+SM+Ver	2210	47	2156	7	99.68%

**Table 9**

Table 9 Accuracy on Simulated Masked Vggface2 Test Set					
Setting	#Test	#Unknown	#Correct	#Wrong	Accuracy
Origin	26161	590	23839	1732	93.23%
Origin+SM	26161	333	24163	1665	93.55%

Origin+Ver	26161	4231	21794	136	99.38%
Origin+SM+Ver	26161	3311	22712	138	99.4%

**Table 10**

Table 10 Accuracy on Real Masked Self-Collected Test Set					
Setting	#Test	#Unknown	#Correct	#Wrong	Accuracy
Origin	18026	344	17482	200	98.87%
Origin+SM	18026	64	17883	79	99.56%
Origin+Ver	18026	604	17403	19	99.89%
Origin+SM+Ver	18026	158	17848	20	99.89%

Table 11 presents the masked face test results for the four experimental settings of the three test sets. Table 12 presents a comparison of the proposed system with other systems. The experimental results demonstrate that similar to the unmasked face test, adding facial verification to either model can improve the accuracy. Moreover, the accuracy improvement in the masked face test is also higher than that in the unmasked face test. Including facial verification improves the accuracies by at least 0.63% in the simulated masked celebA test, by 5.85% in the simulated masked VGGFace2 test, by and 0.33% in the test on a real masked self-collected dataset.

The model trained with the SMFD also improved the accuracy of the masked face tests. This improvement was more noticeable when no face verification was performed. If no face verification was conducted in the tests, using the SMFD to train the model could improve the accuracy by 0.55% in the simulated masked celebA, 0.32% in the simulated masked VGGFace2, and 0.69% in the real masked self-collected tests.

**Table 11**

Table 11 Accuracy in Masked Face Test			
Accuracy	CelebA	VGGFace2	Self-collected dataset
Origin	98.5%	93.23%	98.87%
Origin+SM	99.05%	93.55%	99.56%
Origin+Ver	99.62%	99.38%	99.89%
Origin+SM+Ver	99.68%	99.40%	99.89%

**Table 12**

Table 12 Accuracy when Compared to other Methods in Unmasked Face Test		
Accuracy	CelebA	VGGFace2
SphereFace Liu et al. (2017) + SMFD	98.41%	92.72%
CosFace Wang et al. (2018) + SMFD	99.05%	93.81%
ArcFace Deng et al. (2019) + SMFD	99.05%	93.55%
(Origin+SM in above table)	99.68%	99.4%
Proposed system (Origin+SM+Ver in above table)		

### 3.3. FALSE ACCEPTANCE RATE (FAR) AND FALSE REJECTION RATE (FRR) TEST

We used the VGGFace2 and self-collection test sets to perform the test. For the VGGFace2 test, 90 identities from the VGG2 dataset were used. We used nine identities for the tests on the self- collection test set.

One image of each identity to be recognized is used as the face-recognition database. That is, the output of the face recognition can only be an unknown identity or one of these identities. For the VGGFace2 test set, the face-recognition database contained only the identities in the test set. For the tests on the self-collected test set, the face-recognition database contained 81 identities of the VGGFace2 test set in addition to the 9 identities tested to increase the reliability of the test. The images used in the face-recognition database were removed from the test set.

$$FAR = \frac{\#Incorrectly\ Accept}{\#Correctly\ Reject + \#Incorrectly\ Accept} \times 100\% \tag{9}$$

$$FRR = \frac{\#Incorrectly\ Reject}{\#Incorrectly\ Reject + \#Correctly\ Accept} \times 100\% \tag{10}$$

Table 13 presents the test results of the unmasked VGGFace2 test set with models trained with and without the SMFDs.

**Table 13**

Table 13 FAR and FRR for the Unmasked Vggface2 Test Set				
Test Setting	FAR without the SMFD	FRR without the SMFD	FAR with the SMFD	FRR with the SMFD
VerH	0.00135%	8.54%	0.00209%	8.05%
VerM	0.0029%	8.45%	0.00456%	7.93%
VerL	0.00568%	8.3%	0.00866%	7.79%
NVer	0.05%	5.18%	0.054%	5.21%

Table 14 presents the test results of the simulated masked VGGFace2 test set with the models trained with and without the SMFDs.

**Table 14**

Table 14 FAR and FRR for Simulated Masked Vggface2 Test Set				
Test Setting	FAR without the SMFD	FRR without the SMFD	FAR with the SMFD	FRR with the SMFD
VerH	0.00584%	13.87%	0.00593%	11.02%
VerM	0.017%	11.19%	0.01659%	9.41%
VerL	0.027%	10.08%	0.027%	8.43%
NVer	0.074%	7.4%	0.071%	6.53%

Table 15 presents the test results of the unmasked self-collected test set with models trained with and without the simulated masked face databases.

**Table 15**

Table 15 FAR and FRR for Unmasked Self-Collected Test Sets				
Test Setting	FAR without the SMFD	FRR without the SMFD	FAR with the SMFD	FRR with the SMFD
VerH	0	0.16%	0	0.13%
VerM	0	0.16%	0	0.13%
VerL	0	0.16%	0	0.13%
NVer	0	0	0	0

Table 16 presents the test results of the real masked self-collected test set with models trained with and without the simulated masked face databases.

**Table 16**

Table 16 FAR and FRR on Real Masked Self-Collected Test Sets				
Test Setting	FAR without the SMFD	FRR without the SMFD	FAR with the SMFD	FRR with the SMFD
VerH	0.0124%	3.45%	0.0131%	0.99%
VerM	0.0444%	3.06%	0.034%	0.87%
VerL	0.0464%	3.03%	0.0379%	0.85%
NVer	0.0464%	3.02%	0.0386%	0.79%

According to the experimental results, the FAR with face verification was significantly lower than that without face verification. Moreover, the higher the threshold of face verification, the lower the FAR. The experimental results demonstrated that including face verification can effectively increase the security of the system. Notably, the FAR of the unmasked self-collection test set was zero because its shooting environment was relatively simpler than that of VGGFace2, with less interference from the environment and light. In addition, the average image quality was better, due to which it was relatively easy to recognize.

The average FRR of the model trained using the simulated masked face database was lower than that of the model trained without the database. The experimental results demonstrated that including a simulated masked face database during model training can improve the convenience. For the VGGFace2 test set, the average FRR was high because the photo-shooting environment, including different lighting and backgrounds, was complex and the photo quality was uneven. For the self-collected test set, the average FRR performance was improved further because the lighting and background of the photo-shooting environment were simpler and the photo quality was higher.

### 3.4. MASK-DETECTION TEST

The training dataset contained 2971 masked and 3323 unmasked face images captured using web crawlers. The test dataset contained 9 identities. A total of 18,035 masked and 13,442 unmasked facial images were captured in the laboratory. The testers rotated and moved their faces slightly during shooting to increase the reliability of the test.

Table 17 presents the confusion matrix for the mask-detection test. The experimental results demonstrated that the proposed system could identify a user without a face mask. Most misclassified cases were caused by excessive head-

turning angles. In our experimental results, the accuracy is 99.6%, precision is 100%, recall is 99.3% and F1 score is 99.65%.

**Table 17**

<b>Table 17 Confusion Matrix of Mask-Detection Test</b>			
<b>Confusion Matrix</b>		<b>Actual class</b>	
		<b>Masked</b>	<b>Unmasked</b>
<b>Predicted class</b>	<b>Masked</b>	17909	0
	<b>Unmasked</b>	126	13442

### 3.5. LIVENESS DETECTION TEST

Table 18 presents the experimental results. The proposed system could effectively distinguish between real people and photographs. The temperature of the surrounding environment, such as those from hot drinks and food, is not considered to be human temperature. Some people were misclassified as photos because the system could not obtain the correct temperature of users whose faces were blocked by glasses, face masks, or hair.

**Table 18**

<b>Table 18 Confusion Matrix of Liveness Detection Test</b>			
<b>Confusion Matrix</b>		<b>Actual class</b>	
		<b>Live face</b>	<b>Fake face</b>
<b>Predicted class</b>	<b>Live face</b>	48	0
	<b>Fake face</b>	2	50

## 4. CONCLUSIONS

This study proposes a face recognition integrated human body temperature measurement system. The proposed system was used to realize unmasked face images; we then developed a method that could simulate masked images, apply the generated images to train neural networks, and perform face verification such that the system would be able to address occluded faces. In addition, strong and weak correlation verifications were proposed in the face-verification module to improve the accuracy of masked face recognition. By combining the proposed face-verification method with a dataset of simulated masked faces, the accuracies of the unmasked and simulated masked tests on the VGGFace2 test set were increased by 4.31% and 6.17%, respectively. In the fever living body detection module, we effectively fused visible light and thermal images to achieve accurate temperature measurements and living body detection.

In our experimental results, the accuracy rate of naked face recognition reached 99.79%, whereas that of masked face recognition reached 99.4%. In the mask-detection experiment, the accuracy reached 99.6%. In addition, live detection experiments demonstrated that the system could effectively prevent face spoofing attacks on photos and videos and integrate temperature measurement functions to improve the performance of mask face recognition, making the system convenient and practical.

## CONFLICT OF INTERESTS

None.

## ACKNOWLEDGMENTS

None.

## REFERENCES

- Atoum, Y., Liu, Y., Jourabloo, A., & Liu, X. (2017). Face Anti-Spoofing using Patch and Depth-Based CNNs," in 2017 IEEE International Joint Conference on Biometrics (IJCB), IEEE, 319-328. <https://doi.org/10.1109/BTAS.2017.8272713>
- Birla, L., & Gupta, P. (2022). PATRON: Exploring Respiratory Signal Derived from Non- Contact Face Videos For Face Anti-Spoofing, Expert Systems with Applications, 187, 2022. <https://doi.org/10.1016/j.eswa.2021.115883>
- Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Inc.
- Cao, Q., Shen, L., Xie, W., Parkhi, O. M., & Zisserman, A. (2018). Vggface2: A Dataset for Recognising Faces Across Pose and Age, in 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), IEEE, 67-74. <https://doi.org/10.1109/FG.2018.00020>
- Cheheb, I., Al-Maadeed, N., Al-Madeed, S., Bouridane, A., & Jiang, R. (2017). Random Sampling for Patch- Based Face Recognition," in 2017 5th International Workshop on Biometrics and Forensics (IWBF), 1-5. <https://doi.org/10.1109/IWBF.2017.7935104>
- Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive Angular Margin Loss for Deep Face Recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4690-4699. <https://doi.org/10.1109/CVPR.2019.00482>
- Ge, S., Li, C., Zhao, S., & Zeng, D. (2020). Occluded Face Recognition in the Wild by Identity-Diversity Inpainting, in IEEE Transactions on Circuits and Systems for Video Technology, 30(10), 3387-3397. <https://doi.org/10.1109/TCSVT.2020.2967754>
- Guo, Y., Zhang, L., Hu, Y., He, X., Gao, J. (2016). MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, 9907. Springer, Cham. [https://doi.org/10.1007/978-3-319-46487-9\\_6](https://doi.org/10.1007/978-3-319-46487-9_6)
- Hariri, W. (2015). Efficient Masked Face Recognition Method During the Covid-19 Pandemic. <https://doi.org/10.21203/rs.3.rs-39289/v4>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- Huang, G. B., Mattar, M., Berg, T. & Learned-Miller, E. (2008). Labeled Faces in the Wild: A Database Forstudying Face Recognition in Unconstrained Environments. in Workshop on Faces in'Real- Life'Images: Detection, Alignment, and Recognition.
- Kamanga, I. A., & Lyimo, J. M. (2022). Anti-Spoofing Detection Based on Eyeblink Liveness Testing for Iris Recognition. International Journal of Science and Research Archive, 7(1), 53-67. <https://doi.org/10.30574/ijrsra.2022.7.1.0186>
- Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., & Song, L. (2017). Sphereface: Deep Hypersphere Embedding for Face Recognition," in Proceedings of the IEEE

- conference on Computer Vision and Pattern Recognition, 212-220. <https://doi.org/10.1109/CVPR.2017.713>
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep Learning Face Attributes in the Wild," in Proceedings of the IEEE International Conference on Computer Vision, 3730-3738. <https://doi.org/10.1109/ICCV.2015.425>
- Määttä, J., Hadid, A., & Pietikäinen, M. (2011). Face Spoofing Detection from Single Images Using Micro-Texture Analysis," in 2011 International Joint Conference on Biometrics (IJCB), IEEE, 1-7. <https://doi.org/10.1109/IJCB.2011.6117510>
- Najibi, M., Samangouei, P., Chellappa, R., & Davis, L. S. (2017). Ssh: Single Stage Headless Face Detector, in Proceedings of the IEEE International Conference on Computer Vision, 4875-4884. <https://doi.org/10.1109/ICCV.2017.522>
- Qiu, H., Gong, D., Li, Z., Liu, W., & Tao, D. (2022, Oct. 1). End2End Occluded Face Recognition by Masking Corrupted Features, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(10), 6939-6952. <https://doi.org/10.1109/TPAMI.2021.3098962>
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted Residuals and Linear Bottlenecks, in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 4510-4520. <https://doi.org/10.1109/CVPR.2018.00474>
- Sharma, D., & Selwal, A. (2022). An Intelligent Approach for Fingerprint Presentation Attack Detection using Ensemble Learning with Improved Local Image Features, Multimedia Tools and Applications, 81, 22129-22161. <https://doi.org/10.1007/s11042-021-11254-8>
- Snell, J., Swersky, K. & Zemel, R. S. (2017). Prototypical Networks for Few-Shot Learning. <https://doi.org/10.48550/arXiv.1703.05175>
- Teh, W. Y., Tan, I. K. T., & Tan, H. K. (2022). Liveness Detection for Images and Audio," Recent Advances in Face Recognition, in Proceedings of the IEEE International Symposium on Intelligent Signal Processing and Communication Systems. <https://doi.org/10.1109/ISPACS57703.2022.10082821>
- Trigueros, D. S., Meng, L., & Hartnett, M. (2018). Enhancing Convolutional Neural Networks for Face Recognition with Occlusion Maps and Batch Triplet Loss, Image and Vision Computing, 79, 99-108. <https://doi.org/10.1016/j.imavis.2018.09.011>
- Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., & Liu, W. (2018). Cosface: Large Margin Cosine Loss for Deep Face Recognition," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5265-5274. <https://doi.org/10.1109/CVPR.2018.00552>
- Wang, Z., Huang, B., Wang, G., Yi, P., & Jiang, K. (2023). Masked Face Recognition Dataset and Application, in IEEE Transactions on Biometrics, Behavior, and Identity Science, 5(2), 298-304. <https://doi.org/10.1109/TBIOM.2023.3242085>
- Yang, J., Lei, Z., & Li, S. Z. (2014). Learn Convolutional Neural Network For Face Anti-Spoofing. <https://doi.org/10.48550/arXiv.1408.5601>
- Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning Face Representation from Scratch. <https://doi.org/10.48550/arXiv.1411.7923>
- Zeng, D., Veldhuis, R., & Spreeuwers, L. (2021). A Survey of Face Recognition Techniques Under Occlusion. <https://doi.org/10.1049/bme2.12029>
- Zhao, F., Feng, J., Zhao, J., Yang, W., & Yan, S. (2017). Robust LSTM-Autoencoders for Face De-Occlusion in the Wild, IEEE Transactions on Image Processing, 27(2), 778-790. <https://doi.org/10.1109/TIP.2017.2771408>