Multimedia Tools and Applications (2022) 81:1867–1891
https://doi.org/10.1007/s11042-021-11599-0

CFSM: a novel frame analyzing mechanism for real-time face recognition system on the embedded system

Slo-Li Chu[1] · **Chien-Fang Chen**[1] · **Yu-Chen Zheng**[1]

Received: 12 September 2020 / Revised: 14 July 2021 / Accepted: 22 September 2021 /
Published online: 14 October 2021

Abstract
The development of web cameras and smart phones is mature, and more and more facial recognition-related applications are implemented on embedded systems. The demand for real-time face recognition on embedded systems is also increasing. In order to improve the accuracy of face recognition, most of the modern face recognition systems consist of multiple deep neural network models for recognition. However, in an embedded system, integrating these complex neural network models and execute simultaneously is not easy to achieve the goal of real-time recognition of human faces and their identities. In view of this, this study proposes a new frame analysis mechanism, continuous frames skipping mechanism (CFSM), which can analyze the frame in real time to determine whether it is necessary to perform face recognition on the current frame. Through the analysis of CFSM, the frames that do not need to be re-recognized for face are omitted. In this way, the workload of the face recognition system will be greatly reduced to achieve the goal of real-time face recognition in the embedded system. The experimental results show that the proposed CFSM mechanism can greatly increase the speed of face recognition in the video on the embedded system, achieving the goal of real-time face recognition.

Keywords Deep learning · Face recognition · Embedded system · Real-time · Frame analysis

✉ Slo-Li Chu
slchu@cycu.edu.tw

Chien-Fang Chen
ticks0628@gmail.com

Yu-Chen Zheng
blacktea1031@gmail.com

[1] Department of Information and Computer Engineering, Chung Yuan Christian University, Chung Li District, Taoyuan, Taiwan

1 Introduction

The popularity of cameras and smartphones make more and more applications of real-time face recognition on mobile devices, such as using human faces as biometric identification [14, 21], or using human faces for access control systems. The corresponding management applications are becoming more and more popular. Previously, face recognition was mainly through image recognition related mechanisms [3, 4, 11, 20, 29]. At present, due to the rapid development of deep learning and neural networks, more and more face recognition mechanisms are implemented based on neural network models. Since the face recognition system usually requires multi-stage analysis, and different analysis stages need to use different neural network models. However, the mobile devices and embedded systems cannot provide the enough computation capabilities to process these complex neural network models at the same time, and then achieve the goal of real-time recognition of human faces. This will limit the development and popularization of possible face recognition systems.

The studies [2, 27] proposed the modification of the neural network models of recognition to reduce the computational complexity, to meet the device specifications of the embedded system. However, these modifications usually lead to a decrease in the recognizing accuracy. The studies [15, 23] propose the mechanism to offload the computation of face recognition to the remote GPU server via network, then returning the identification results to the local embedded system device. However, when a large amount of video frames to be recognized is transmitted through the network, and the recognition result is returned after the remote calculation is completed, the required transmission bandwidth and computation amount will also make it hard to achieve the goal of real-time face recognition. Meanwhile, the system used for face recognition requires a high-quality network connection for face recognition, which will severely limit the usage of the embedded face recognition system.

Therefore, studies [16, 21, 28, 35, 37] proposes to implement face recognition systems on embedded platforms. Although the recognition accuracies of these system can be improved, it can avoid the inconvenient of unstable network connections, which makes the remote-based face recognition system out of service. In view of this, how to provide robust real-time face recognition services for embedded and mobile devices, and improve the accuracy of recognition will be the main design challenge.

A basic face recognition system will include three analysis stages, which includes face detection, face extraction, and face matching, respectively. The first is the face detection stage. Its main function is to find the corresponding bounding box of the face object contained in the image and its location. The more faces contained in the image, or the more pixels in the image, the longer the detection time required. After this stage is completed, the human faces that are contained in the frame will be detected, and their corresponding cropped face images can be generated. Among the three stages, the face detection stage takes the most time, and is also the main goal to be improved, when we develop the embedded face recognition system. The second stage is face extraction, this stage mainly analyzes the input facial bounding box and its cropped face image, finds the facial features of the face, and generates the feature vector of the facial features. After finding the feature vector of the face, in the third stage, face matching, it will compare the feature vector of the face to identify the known face identity [22].

Since the video will contain 30 frames per second, if the aforementioned basic face recognition system performs three-stage analysis, it will need to perform 30 times of analysis and recognition per second, which will not be easy to achieve on embedded systems. However, when we analyze the adjacent frames of the video, the scenes and faces does not change drastically in each frame. The difference between the two adjacent frames is usually ignorable, as shown in Fig. 1. The video can be divided into three scenes, Scene A, Scene B, and Scene C, respectively. It evolves from Scene A to Scene B. After some frames with similar content, it changes to Scene C. Hence only a few frames need to be captured for face recognition in the same scene, such as the first frame of the scene. The subsequent frames are the continuous frames. Usually the faces included in these continuous frames will also keep the same result of the recognized face in the previous frame. When the number of faces changes or the scene changes drastically, it just perform face recognition for the frame. In this way, the number of frames that need to be face-recognized can be greatly reduced, and the computation requirements for performing three-stage face-recognition analysis on the embedded system can be reduced.

In view of this, this study is based on the realization of the video face recognition system on the embedded system, and the need for real-time recognition of the video, and proposes a new analysis mechanism, continuous frames skipping mechanism (CFSM), which includes three main analysis mechanisms, abrupt scene change detector, face recognition interval adjuster, and dark frame detector, to analyze each frame of the video, and determine whether to perform face recognition calculations for the current frame. If the scene difference between previous and current frames is too small, or the time interval from the last face-recognized frame is not too long, or the current image is not too dark to be analyzed, it can be regarded as a skipping frame, and ignore face recognition processing. In this way, the proposed CFSM mechanism can be used to detect the frames that do not need to be recognized. This will reduce the computational burden in the embedded face recognition system and achieve the goal of FPS greater than 30.

The contribution of this study is summarized as following. (1) The proposed continuous frame skipping mechanism (CFSM) mechanism consists of efficient scene change detection mechanism, abrupt scene change detector (ASCD), to detect abrupt cut and gradual scene without complex mechanism. (2) The dynamic face recognition interval adjuster (FRIA) mechanism, in addition to filtering out a large number of unnecessary face detection for subsequent frames and make up for the insufficient of ASCD

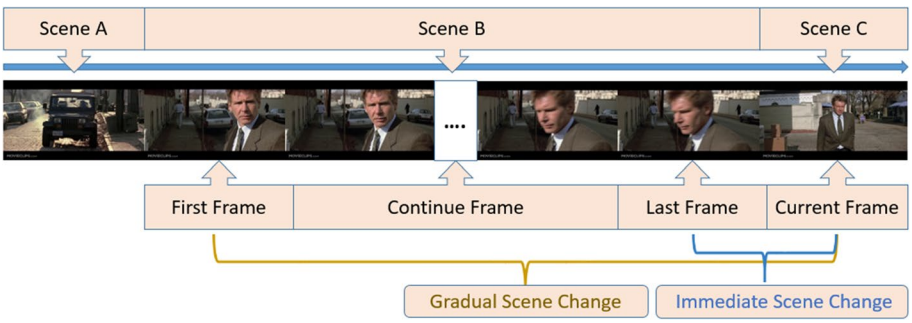


Fig. 1 The scenes and frames composition of the video

mechanism. (3) the dark frame detector (DFD) can reduce the error rate from the dark scenes. (4) The CFSM mechanism can help to achieve the real-time performance of the basic face recognition system that is targeted on the embedded system.

The rest of this paper is written as follows. Section 2 mentioned the previous researches on face recognition algorithms and embedded system technology. The details of the mechanisms we proposed are explained in Section 3. The experimental results will be discussed in Section 4. Finally, in the conclusion section, we will summarize our research.

2 Related works

2.1 The mechanisms of face recognition

Regarding the mechanisms of face detection, study [33] proposed real-time face detection technology, integrated facial feature analyzing algorithms, and after obtaining facial features, then selected the required feature points to quickly detect faces within the image. Recently, deep neural networks are mainly used to detect faces in the image, such as MTCNN [36], Faceboxes [38], and PCN [26], etc., which have good face detection performance. Among them, MTCNN adopts multi-task learning [36], uses three convolutional neural network models, and combines pyramid and non-maximum suppression technology to detect faces of various sizes in the image. However, detecting multiple faces in the frame requires a long processing time and computing resources, and it is not easy to achieve the goal of real-time computing when executed on embedded devices with limited computation capabilities. After detecting multiple faces in the screen, and according to the determined bounding boxes, cropped face images, then these cropped face images can be sent to the following stage, face extraction. In this stage, the facial features of the cropped face image are analyzed, and corresponding facial feature vectors are generated. The conventional methods such as PCA [3], LDA [20], or ORB [21] are mainly used to find the facial feature vector of a face image [30]. In recent years, deep neural network model analysis is mainly used, such as using DeepFace [31], ResNet [10] or FaceNet [25] to extract facial feature vectors. After the facial feature vector is obtained, the subsequent face matching stage can be performed to compare the known facial features, and determine the identification of the faces accordingly. Traditionally, SVM or clustering can be used to compare known face feature vectors. Recently, based on deep neural networks, the classification layers of the deep neural networks and the corresponding loss functions [6, 18, 25, 34] are modified to improve the classification accuracy of facial feature vectors.

2.2 The mechanisms for reducing computation of face recognition

Study [7] proposed an entropy-based method to select the continuous frames from the clip, then skip the subsequent processing steps of feature extraction and fact matching.

However, while the video clips are captured from the surveillance camera or dashboard camera, the regions of faces still need to be detected in the face detection step firstly. The face detection step is also the most time-consumed part of whole face recognition process.

In order to reduce the amount of computation required for face recognition analysis in an embedded system, study [24] proposed a face tracker mechanism to reduce the analyzing workload of face extraction and face matching. After detecting the faces within the frame in the face detection stage, the face tracker analyzes the features of cropped facial image. Then the face tracker applies visual tracking technology [5] to track the faces in the previous and current frames, and excludes the same faces with different positions to reduce the processing burden of subsequent face extraction and face matching. However, the requirement of detecting the faces within the frames makes the face tracker mechanism rely on the analysis results of the face detection stage.

Additionally, the studies [8, 9] propose the 3D convolutional networks [32] based shot boundary detection mechanisms. They can provide good performance to detect the abrupt cut frames in the video clip but not for determining the frames that are required to process the face recognition, such as the frames with the gradual transition, the dark scene, and moving people. Accordingly, the skippable frames detection can not only rely on abrupt cut detection mechanisms. Besides, the neural network based abrupt cut frame detection mechanism requires a lot of execution time. The advantage of continuous frame detection mechanism for saving the unnecessary face recognition processes will be reduced dramatically. The performance comparison of tracker and CNN mechanisms will be discussed in the Section 4.

3 Proposed methods

In embedded systems and mobile devices, facial recognition is already an important application. In order to reduce the computational burden of the face recognition system on the embedded devices and meet the design requirements of real-time face recognition in the video, this study proposes a continuous frame skipping mechanism (CFSM) to analyze the previous and current frames, to determine the abrupt scene changes of the frames, then decides whether the current frame will enter the following processing of face detection, face extraction, and face matching. CFSM is composed by three analysis mechanisms, abrupt scene change detector, face recognition interval adjuster, and dark frame detector, as shown in Fig. 2. After the current frame is analyzed by CFSM, you can decide whether to enter the face recognition system for face recognition, or skip directly, and use the face recognition result of the previous frame. The adopted mechanisms in the basic face recognition which include MTCNN [36] and FaceNet [25] neural network models, are used for the Face Detection and Face Extraction processing stages. These mechanisms will be discussed in Section 4.

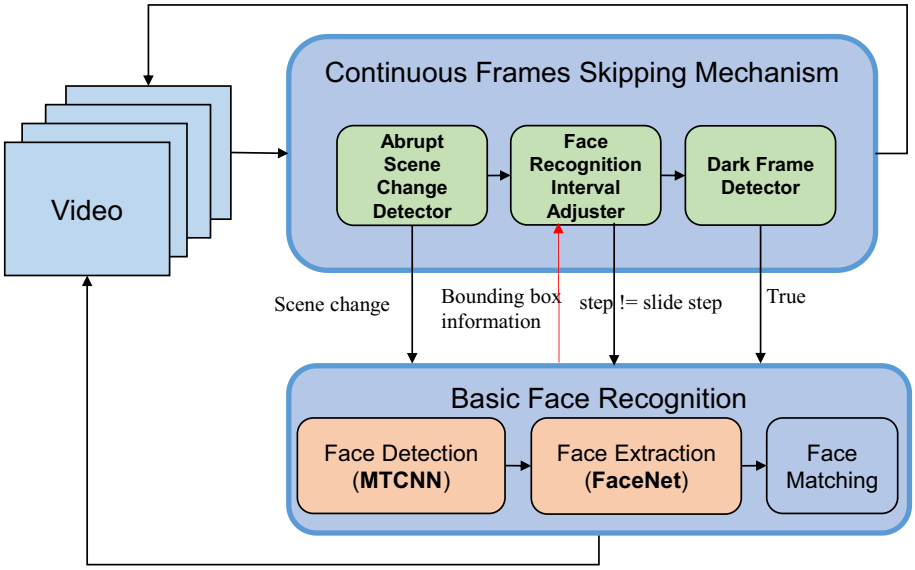


Fig. 2 The architecture of continuous frame skipping mechanism and the face recognition system

In the following, the functions of the three mechanisms of CFSM will be described in detail.

The algorithm of the proposed CFSM mechanism is as listed in Algorithm 1. The input video clip will be transcoded to the YUV color domain for the following processing. It contains three main functions, abrupt scene change detector (ASCD), face recognition interval adjuster (FRIA), and dark frame detector (DFD). The FRIA function will listed in Algorithm 2. In the algorithm, Td_{block} is used to determine the number of different blocks between the current frame and reference frame. Tf_{imm} acts in ASCD to determine whether abrupt scene change occurs or not. Tf_{grad} is used to determine the gradual scene change. Th_{Eff} is used to detect dark scene. These thresholds remain constant in this study. In CFSM mechanism, the ASCD is the first analysis stage. It contains two functions. One is to count the number of different blocks between the adjacent frames, $D^{c,r}$. If it is greater the predefined Tf_{imm}, then satisfied the condition of abrupt scene change. The other is to calculate the number of different blocks $D^{s,c}$, between the first frame of this scene and the current frame. If it is greater than Tf_{grad}, then satisfied the condition of gradual scene change. But some frames within the same scene still need to apply Face Recognition due to the people movement or hard to determine. Accordingly, the following analysis stages, FRIA and DFD, are proposed to improve the accuracy of CFSM mechanism. These two states will be discussed later.

Algorithm 1: The algorithm of CFSM mechanism.

```
Input: InputVideoFrame(V)
Output: FaceRecognition
skip_count: 0
procedure   CFSM(V)
        Input Feaures(V)
        Initial threshold: Td_block, Tf_imm, Tf_grad, Th_Eff
        D^{c,r} = ASCD( current_frame, previous_frame, Td_block)
        D^{s,c} = ASCD( current_frame, start_frame, Td_block)

        if( D^{c,r} > Tf_imm) then                          //Abrupt change detected
                FaceRecognition(current_frame )
        else if( D^{s,c} > Tf_grad)   then                  //Gradual change detected
                FaceRecognition(current_frame )
        end if
        if( skip_count >= Θ) then
                Θ=FRIA( current_frame )
                Reset skip_count
        else
                increase skip_count
        end if

        Amt_Eff = DFD( current_frame)
        if( Amt_Eff > Th_Eff) then                          //Dark frame detected
                        FaceRecognition(current_frame )
                        resetAmt_Eff
        end if

end procedure
function ASCD( current, reference, Td_block)                //ASCD as equation 1,2,3
        {B_{0,0}, B_{0,1}, B_{0,2}, … , B_{N−1,M−1}}=Tiling current, reference to NxM block
        for n = 0 to N-1
                for m = 0 to M-1
                        δ^{c,r}_{nm} = abs(B^c_{n,m} − B^r_{n,m})
                        Increase D^{c,r}, if δ^{c,r}_{nm} >Td_block
                end for
        end for
        return D^{c,r}
end function
function DFD( current_frame)
        Increase Amt_Eff, if Avg(Lum(current_frame)) > Th_dfd
        Return Amt_Eff
end function
function FaceRecognition(current_frame )
        [face_counter, bounding_box] = FaceDetection(current_frame )
        face_encoding = FaceExtraction(bounding_box )
        FaceMatching(face_encoding )
        face_counter_change = FaceDetection(current_frame) - FaceDetection(previous_frame)
        reset Amt_Eff, skip_count
end function
```

3.1 Abrupt scene change detector

The video to be face-recognized is input into the aforementioned face recognition system at a rate of 30 frames per second. Before the face-recognition, the CFSM system proposed by this study will be used to determine whether the current frame requires face recognition. The first analysis step is abrupt scene change detector (ASCD), which is used to detect whether the frame has drastic changes. The following will introduce the ASCD analysis mechanism.

The Frame F to be analyzed is composed of Width * Height pixels. Therefore, the pixel in the picture can be expressed as $p_{w,h}$. Here we use the $Lum(p_{w,h})$ function to get the Luminous value of $p_{w,h}$. In order to reduce the impact of a small number of pixels with large numerical differences on subsequent analysis, we use the block, with size of b*b pixels, to partition the entire frame into N*M blocks, where N = Width/b, M = Width/b. Therefore, a Frame F^a is composed of multiple blocks, $\{B^a_{0,0}, B^a_{0,1}, B^a_{0,2}, \ldots, B^a_{N-1,M-1}\}$. The Luminous values of b*b pixels in the block $B^a_{n,m}$ are averaged, $Avg(B^a_{n,m})$, and the calculation method is shown in Eq. 1.

$$Avg(B^a_{n,m}) = \frac{1}{b*b}\sum_{\mathrm{i}=n*b}^{(n+1)*b-1}\sum_{\mathrm{j}=\mathrm{m}*b}^{(m+1)*b-1} Lum(p_{i,j}) \tag{1}$$

In order to calculate the difference between the reference frame F^r and the current frame F^c, we first calculate the average measurement difference between the two blocks at the same location between F^r and F^c, as shown in Eq. 2:

$$\delta^{c,r}_{nm} = \left| B^c_{n,m} - B^{\mathrm{r}}_{n,m} \right| \tag{2}$$

In order to calculate the block difference $D^{c,r}$ between the reference frame and the current frame, as shown in Eq. 3. Where Td_{block} is the threshold of block difference.

$$D^{c,r} = \frac{1}{N*M}\sum_{i=0}^{N-1}\sum_{j=0}^{M-1}\begin{cases} 1, if\, \delta^{c,r}_{nm} > Td_{block} \\ 0, if\, \delta^{c,r}_{nm} \le Td_{block} \end{cases} \tag{3}$$

The proposed abrupt scene change detector will aim to real-time analysis for two kinds of video scene changes, as shown in Fig. 1. The first, (1) immediate scene change detection: this mechanism mainly detects sudden changes in the scene or change quickly. Therefore, the main purpose is to analyze the $D^{c-1,c}$, between the current frame F^c and the previous frame F^{c-1}, and whether the difference is greater than the immediate scene change threshold: Tf_{imm}. The second is (2) gradual scene change detection. This mechanism mainly detects gradual scene changes, such as the slow movement of people in the video, resulting in small cumulative frame changes. This change is hard to detect by using the first detection mechanism. Therefore, in the second mechanism, we will calculate the difference $D^{s,c}$ between the starting frame F^s of this scene and the current frame F^c, when the difference is greater than the progressive scene change threshold: Tf_{grad}, it means there is a gradual scene change. When an immediate or progressive scene change occurs, the current frame F^c is sent to the face recognition system for face recognition. And mark frame F^c as the starting point of the scene change.

3.2 Face recognition interval adjuster

The scenes in the video often have subtle changes that are not easily detectable. Through the aforementioned two scene detection mechanisms of ASCD, sometimes it is difficult to completely find the scenes with new faces. This will reduce the correct judgment of the CFSM proposed by this study and increase the occurrence of unrecognized faces. Therefore, this study proposes an analysis method, face recognition interval adjuster (FRIA), to cooperate with the ASCD, which determines the appropriate time interval for face recognition based on the previous recognizing history of face recognition. When the time interval from the previous face recognition is greater than the determined time interval of FRIA, regardless of whether it is a scene change, the frame is sent to the face recognition system for recognition. This mechanism uses a state machine to analyze the past face recognition history and dynamically adjust the recognition time interval, as shown in Fig. 3. The corresponding algorithm of FRIA is as listed in Algorithm 2. The Θ denotes the skip number, it will within two predefined limits, low_limit, and high_limit. This mechanism is mainly based on the number of faces C_{face} recognized by frame to determine. The mechanism decides to adjust the maximum recognition time interval according to the current state, which includes six time interval adjustment methods according to the states, namely entrance, major increase, minor increase, major decrease, minor decrease, and dense. The initial state is entrance. The adjustment mechanism will increase or decrease the maximum recognition time interval based on the C_{face} detected by the current frame after recognition

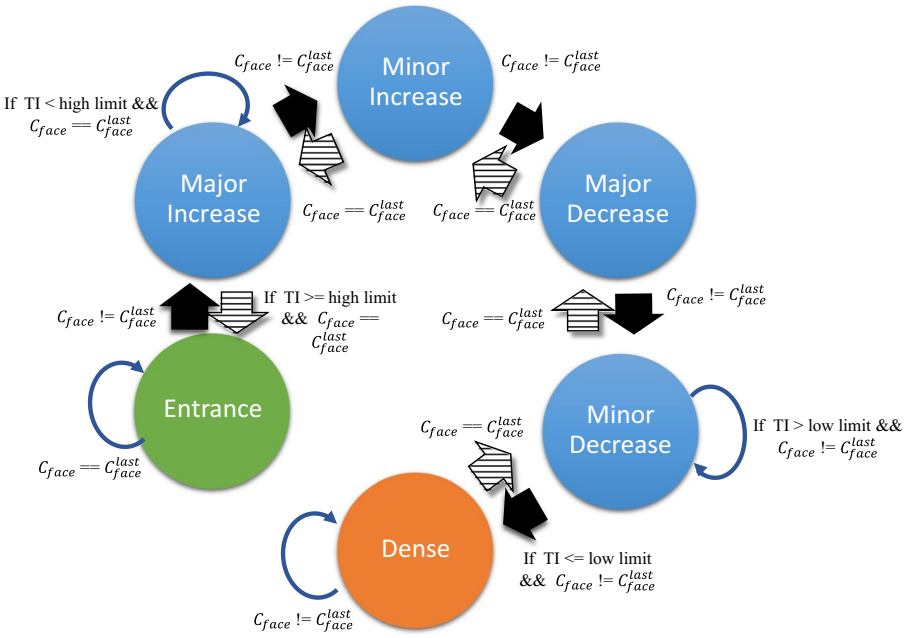


Fig. 3 The adjust states of face recognition interval adjuster

by the face recognition system and the last detected face number C_{face}^{last}. As long as a new scene is detected by ASCD, the state will restart to the entrance for the following analysis of FRIA.

Algorithm 2: The algorithm of FRIA mechanism.

```
skip_state: Entrance            // skip_state ∈{Entrance, Major Increase, Minor Increase,
                                //Major Decrease, Minor Decrease, Dense}
function FRIA( current_frame)
    face_counter_change = FaceRecognition( current_frame )

    if(C_face != C_face^last ) then        //different face count between conjunction frame
        if( skip_state ==Minor Decrease) or ( skip_state == Dense ) then
            if( skip_state > low_limit) then
                keep skip_state
            end if
        end if
        farward skip_state
    else
        if(skip_state==Major increase) or (skip_state==Entrance) then
            if( skip_state < high_limit) then
                keep skip_state
            endif
        end if
        rewind skip_state
    end if
    return Θ of skip_state
end function
```

3.3 Dark frame detector

Because the aforementioned face recognition system mainly performs face detection, feature extraction, and comparison based on the color changes between pixels of the screen. When the frame is too dark, it will greatly affect the stability of face recognition. The example of dark frame video and face detection results are shown in Fig. 4. Therefore, the mechanism, dark frame detector (DFD), proposes a detection mechanism for dark frames in the video. Based on the previously analyzed video, it determines the dark picture threshold Th_{DFD}, and records the analyzed frame, and its average luminous value of the current frame, $Avg(Lum(F^c))$. The Amt_{Eff} will keep the cumulate amount of the frame which the corresponding $Avg(Lum(F^c))$ is greater than the number of Th_{DFD}. If Amt_{Eff} exceeds the effective frame threshold, Th_{Eff}, send the current frame to the face recognition system, and reset Amt_{Eff} for further analysis.

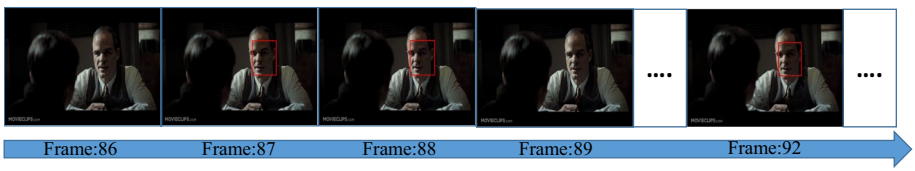


Fig. 4 The example of dark frame video with the corresponding face detection results

4 Experimental results

The proposed embedded face recognition system and CFSM mechanism for the embedded platform are targeted on NVIDIA Jetson TX2 and designed by using Python and TensorFlow framework. The test ten videos are adopted from Youtube-8 M [1] dataset. The attributes of these video clips are shown in Table 1. Among them, a total of ten video clips with the frame pixel dimensions of 1280×720. The following discussion will compare the frame rate and face recognition performances of the basic face recognition system, the face recognition system with the Face Tracking [24] mechanism, and the face recognition system with the proposed CFSM mechanism. The adopted thresholds of CFSM for the following experiments, are set as below:Td_{block}=30, Tf_{imm}=50%, Tf_{grad}=12.5%, Th_{Eff}=5, Th_{dfd} =40, low_limit = 5, and high_limit = 50.

The basic face recognition system includes three analyzing stages, face detection, face extraction, and face matching, for real-time video face recognition. Each of these three stages is processed by a set of neural network models. The face detection stage uses the MTCNN [36] model to figure out the candidates of the face in each frame in the video on-the-fly and provides the bounding boxes and the cropped face images. The face extraction stage uses the FaceNet model [25] to perform face feature extraction from the cropped face images respectively in the previous stage, and generates the feature embedding information of the cropped face image. Then in the third stage, face matching stage, the original triple loss function in the classification layer is replaced by ArcFace [6] loss function to achieve better result. The model is trained by LFW [13] dataset.

4.1 The design of embedded face recognition system

First of all, this study compares the aforementioned embedded face recognition system that ported on different computer environments. The different execution times of the proposed basic face recognition system on two computing devices are discussed. Take the "City Slickers" clip as an example, the video has a total of 3291 frames. Figure 5 compares the execution time spent on the NVIDIA Jetson TX2 embedded system and the computer equipped with NVIDIA GTX1080 GPU for face recognition system, and the breakdown of time spent in each of the three analysis stages. For the target platform of Jetson TX2, the face detection stage takes about 83.08% of the entire time of the face recognition system. The face extraction stage uses 15.38% of the total execution time, while the face matching stage takes 1.53% of the whole execution time. The execution time of the same face recognition system on a GTX1080 computer and the time spent in the three analysis stages are 84.2%, 14.7%, and 0.98% respectively. Therefore, the time proportions of the three face recognition phases on two systems with different computing capabilities are similar. In addition, it can be seen from Fig. 5 that Jetson TX2 takes almost three times the time of GTX1080. The result also illustrates the difficulty of designing a face recognition system on the embedded system.

In view of this, in order to reduce the execution time of the proposed face recognition system on the embedded platform, we focus on the most time-consuming stage, face detection to reduce the computation cost. The proposed face detection stage adopts MTCNN neural network to identify the candidate faces and crop the face images with to bounding box. In the MTCNN, it adopts pyramid resize to determine the suitable bounding box to crop the candidate's face image precisely. Then the multiple bounding boxes of the

Table 1 The attributes of the adopted ten video clips

Name of the video clip	Number of frame	Characteristics of the video clip
"Changeling"	2629	Same character in the same scene for a long time; few wipes; few abrupt cuts; dark scenes
"Before Sunset"	2916	The same character is in the long shot but the background is a continuous wipe; a few abrupt cuts
"The International"	3001	More abrupt cuts; some zoom in/out scenes
"Patriot Games"	3133	Multiple wipe; multiple cut; with fighting scene
"Say Anything"	3141	Many characters (>4); the same scene for a long time; a few wipes
"Instant Family"	3193	More characters, more cuts
"Funny People"	3201	Same character in the same scene for a long time; few wipes; few abrupt cuts
"City Slickers"	3291	Many characters (>4); many abrupt cuts; few wipes
"Uncle drew"	3848	More abrupt cut; more wipe
"Animal house"	4284	Complex scenes; wipe; abrupt cut; multiple characters

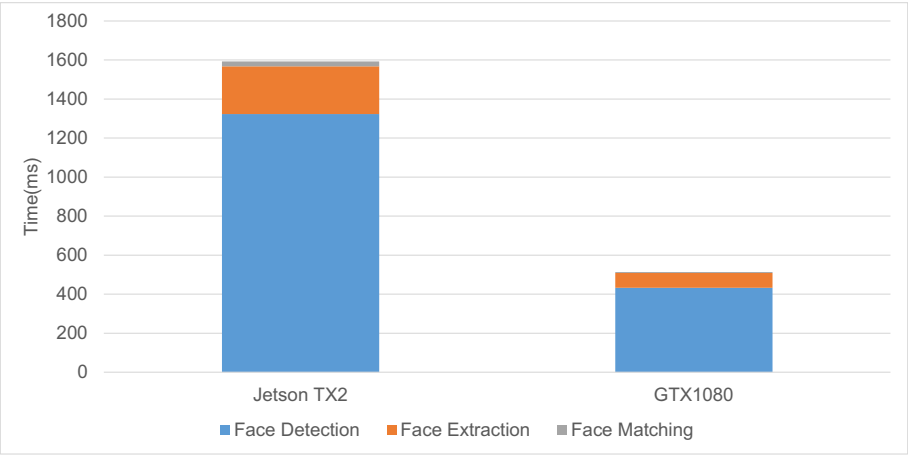


Fig. 5 The performance comparison of the basic face recognition systems on Jetson TX2 vs GTX1080

candidate face will be generated and processed by using the non-maximum suppression mechanism to find out the suitable bounding box. Therefore, as the number of faces in the picture increases, the analysis speed will decrease dramatically. In order to improve the speed of real-time analysis, this study considers to adjust the times of the pyramid resize. The corresponding execution time is as shown in Fig. 6. On the left are different resize times, and on the right are individual accuracy differences. It can be seen that if the number of times of MTCNN pyramid resize is reduced from 12 to 6, the execution time of MTCNN will be reduced from 80 to 60% of the total analysis time, but its accuracy rate will only decrease by 1%, although the resize number of times drops to 3, and the proportion of MTCNN execution time can be reduced to 52%, but its accuracy has a significant impact. Therefore, this study will set the number of MTCNN pyramid resize to 6 times.

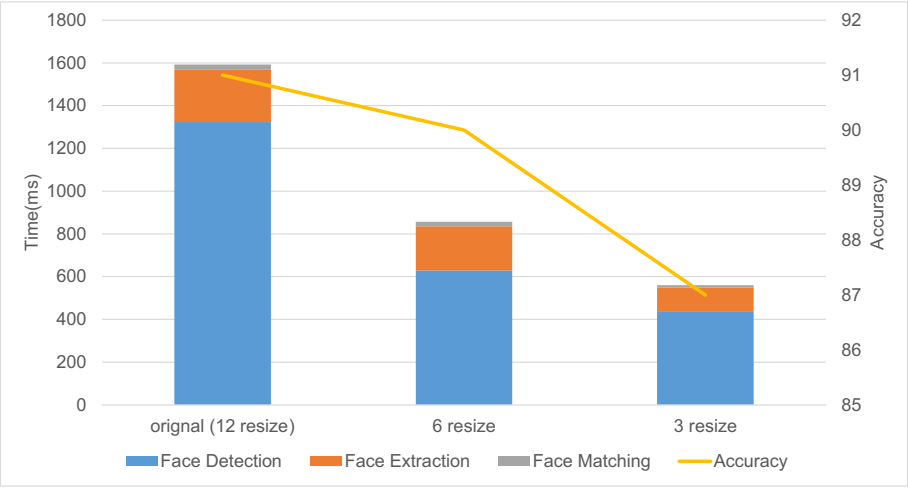


Fig. 6 The comparison of execution time and accuracy of three times of the pyramid resize

Besides, the original MTCNN combines two deep learning methods to detect the bonding boxes of the faces and alignment. In this study, we only adopt the face detection mechanism to reduce the execution time of the face detection stages.

4.2 The comparison of Basic FR, tracker, C3D_method, and CFSM mechanisms

The following will compare three methods, the basic face recognition system (Basic FR), the basic face recognition system with face tracking mechanism (Tracker), and the face recognition system with the proposed C3D_method mechanism, as shown in Figs. 7, 8, 9, respectively. The face recognition system with the proposed CFSM mechanism is mentioned in Fig. 2.

The following experiments adopt ten videos to evaluate the performance of three face recognition systems. The following experiments are targeted on NVIDIA Jetson TX2 platform. The numbers of the frame, execution times of the basic face recognition system and the corresponding frame rate per second (FPS) are listed in Table 2. The FPS of ten videos that are processed by the basic face recognition system on the NVIDIA Jetson TX2 range from 3.8 to 9.8, which does not meet the requirements of real-time face recognition.

The proposed face tracking (Tracker) mechanism used in this study is developed based on the studies [19, 24]. It uses the Hungarian algorithm [17] to detect the overlap

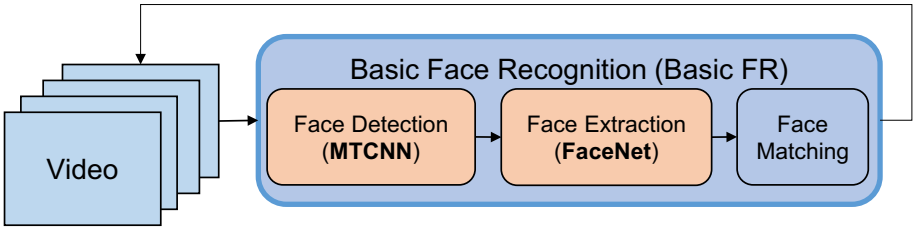


Fig. 7 The architecture of the basic face recognition system

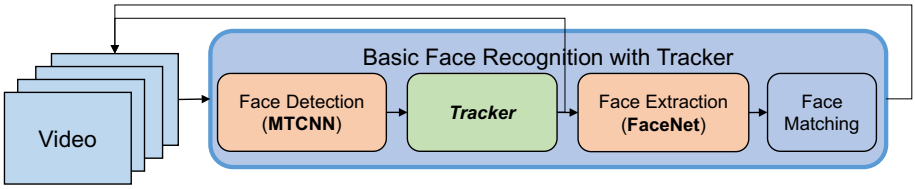


Fig. 8 The architecture of the basic face recognition system with tracker

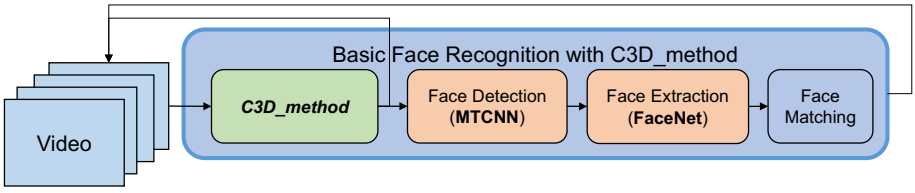


Fig. 9 The architecture of the Basic Face Recognition system with C3D_method

Table 2 The attributes and execution times of ten videos by basic face recognition system

	Number of frame	Basic FR Total execution time (s)	FPS
"Changeling"	2629	269.29	9.76
"Before Sunset"	2916	391.33	7.45
"The International"	3001	343.45	8.76
"Patriot Games"	3133	318.45	9.83
"Say Anything"	3141	505.85	6.2
"Instant Family"	3193	435.16	7.33
"Funny People"	3201	427.17	7.52
"City Slickers"	3291	856.57	3.84
"Uncle drew"	3848	898.58	4.28
"Animal house"	4284	706.14	6.06

bounding box of the current and previous frames. When the overlap range exceeds 30%, it is regarded as the same cropped face image. The Kalman filter method [12] is used to predict and analyze the bounding box, and compare the corresponding bounding box of the previous frame. Therefore, it is still necessary to perform the face detection stage for each frame to obtain the bounding boxes of the candidate's faces in the frame. The face tracking mechanism mainly omits the subsequent processing of face extraction and face matching stages, as shown in Fig. 10. Using the integrated basic face detection system and face tracking mechanism, analyze ten videos on NVIDIA Jetson TX2, and the times spent for face tracking mechanism and the basic face recognition system respectively is shown in Table 3. Compared with the basic face recognition system, the time reduction rate is about 19–36.3%, and the FPS can be increased to about 5–14. Although the performance has been improved, it cannot meet the requirements for real-time face recognition on the embedded system.

The performance of the proposed CFSM mechanism integrated with the basic face recognition system is discussed as following. The detailed execution time, FPS, and time-saving ratio of ten evaluated videos are listed in Table 4. According to the results, the proposed CFSM mechanism will greatly reduce the time required for the face recognition system since it can detect the continuous frame and omit these frames to prevent unnecessary processing of face recognition, which includes face detection, face extraction, and face matching stages. Hence the execution time can be reduced up to 89.5%, the corresponding FPS can reach 59. The objective of real-time face recognition on the embedded system can be achieved. Compared with the face tracking mechanism, the face tracking system still requires the face detection stage to determine the bounding boxes in each frame, the time-saving ratio achieves 36.3%.

The adopted C3D_method [8] mechanism in the following experiments are based on the 3D convolutional neural network, which can detect the shot boundary well. This model adopts the third dimension of the 3D convolution to process the continuous ten frames, to detect the changes among the neighboring frames. The model consists four 3D convolution layers to extract the required features of the ten frames. Finally it can determine whether there are scene transitions in these frames. The original input dimension is 64×64 pixels. While resize the input frame, from 1280×720 to the 64×64, the

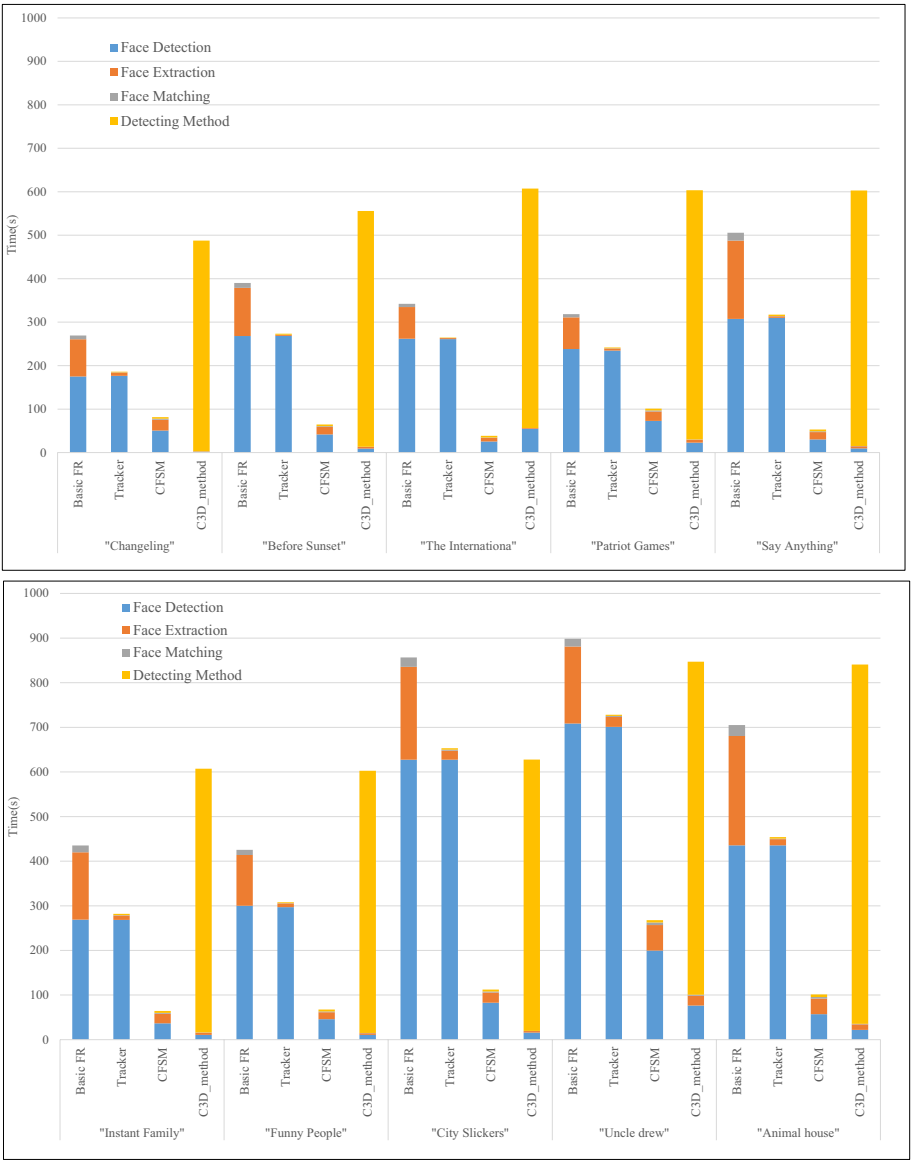


Fig. 10 The execution time of the three analyzing stage of face recognition in Basic FR, Tracker, CFSM, and C3D_method for ten videos

remained image features are too few to recognize. The detection accuracy is not enough for skipping continuous frames. Hence we partition the input frame into 5×3 tiles with dimension of 64×64 pixels. The experimental results are as shown in Tables 5 and 6. Although it can achieve acceptable accuracy but the execution time of C3D_method

Table 3 The attributes and execution times of 10 videos by basic face recognition system with tracker

	Execution time of Basic FR with Tracker (s)			Time reduction ratio (%)	FPS
	Total execution time (s)	Tracker	Basic FR		
"Changeling"	184.76	1.31	183.45	31.38	14.22
"Before Sunset"	273.34	2.38	270.96	30.15	10.67
"The International"	343.21	1.01	342.2	22.67	11.34
"Patriot Games"	241.82	1.88	239.94	24.06	12.95
"Say Anything"	317.36	3.4	313.96	36.27	9.89
"Instant Family"	281.97	2.85	279.11	35.2	11.32
"Funny People"	344.12	1.92	342.2	27.59	10.19
"City Slickers"	653.05	3.18	649.87	23.75	5.03
"Uncle drew"	728.27	2.8	725.46	18.95	5.28
"Animal house"	453.93	3.45	450.48	35.71	9.43

Table 4 The attributes and execution times of 10 videos by basic face recognition system with CFSM

	Execution time of basic FR with CFSM (s)			Time reduction ratio (%)	FPS
	Total execution time (s)	CFSM	Basic FR		
"Changeling"	81.61	3.61	78	69.69	32.21
"Before Sunset"	64.74	4.01	60.73	83.41	45.03
"The International"	267.71	4.12	263.59	88.7	77.62
"Patriot Games"	101.19	4.32	96.86	68.22	30.96
"Say Anything"	52.99	4.19	48.79	89.52	59.27
"Instant Family"	64.4	4.32	60.08	85.19	49.57
"Funny People"	67.45	4.39	63.06	84.14	47.45
"City Slickers"	112.38	4.53	107.86	86.87	29.28
"Uncle drew"	267.92	5.32	262.6	70.184	14.36
"Animal house"	101.34	5.88	95.46	85.64	42.27

mechanism is larger than original face recognition system (Basic FR). The advantage of frame skipping is limited accordingly.

4.3 The comparison of variant face recognition systems

The following compares three different face recognition systems, the basic face recognition system (Basic FR), the face recognition system with face tracking mechanism (Tracker), and the face recognition system with the proposed CFSM (CFSM). Ten videos are adopted to evaluate the execution time of three face recognition systems. The execution time breakdown of face detection, face extraction, face matching, and the detection methods (Basic FR, Tracker, C3D_method and CFSM) are illustrated in Fig. 10. In these videos, the execution time of the CFSM and Tracker mechanism only accounts for a small percentage of

Table 5 The attributes and execution times of ten videos by basic face recognition system with C3D_ method

	Execution time of basic FR with C3D_method(s)			Time reduction ratio (%)	FPS
	Total execution time	C3D_method	Basic FR		
"Changeling"	487.56	484.70	2.86	-81.05	5.39
"Before Sunset"	555.83	542.55	13.29	-42.04	5.25
"The International"	607.48	550.78	56.71	-76.88	4.94
"Patriot Games"	603.58	573.41	30.17	-89.54	5.19
"Say Anything"	602.82	587.80	15.02	-19.17	5.20
"Instant Family"	607.29	591.07	16.22	-39.56	5.26
"Funny People"	602.61	587.75	14.87	-41.07	5.31
"City Slickers"	627.77	608.23	19.55	-26.71	5.24
"Uncle drew"	846.97	746.34	100.64	-5.74	4.54
"Animal house"	840.80	805.31	35.49	-19.07	5.10

Table 6 The accuracy comparison of CFSM and C3D_method mechanisms

	CFSM		C3D_method	
	Accuracy	FD frame count	Accuracy	FD frame count
"Changeling"	0.92	727	0.68	38
"Before Sunset"	0.96	538	0.80	109
"The International"	0.97	320	0.96	276
"Patriot Games"	0.95	1011	0.87	350
"Say Anything"	0.96	273	0.85	118
"Instant Family"	0.86	577	0.71	171
"Funny People"	0.88	436	0.78	132
"City Slickers"	0.74	438	0.60	108
"Uncle drew"	0.85	1112	0.71	476
"Animal house"	0.87	594	0.70	287

the total execution time. On the contrary, the CNN based C3D_method mechanism spend a lot of time in the abrupt cut detection stage, even surpassing the total execution time of the corresponding Basic FR mechanism, as shown in Fig. 10. In the video, Basic FR and Tracker spend almost the same time in the face detection stage, since the face tracking mechanism of Tracker needs the bounding boxes of faces found in the face detection stage. After finding the face, it can be compared and tracking the current and previous frames, to save the time required for subsequent face extraction and face matching. However, face detection is the most time-consuming part of the three face recognition stages since the face detection mechanism need to scan whole frame and find multiple candidate face bounding boxes, but following face extraction and face matching only need to process the feature within the candidate bonding box. The performance improvement of Tracker is limited.

Unlike Tracker, CFSM can directly omit the time of face detection stage. Therefore, as long as it is a frame that can be determined to be skipped, the three-stage processing of face recognition can be omitted. The total execution time can be greatly reduced. Take the "Uncle Drew" video as an example. Using CFSM, in the face detection stage, it saves 71% of the execution time compared to Tracker and Basic FR. In another video, "Say Anything", the total execution time was saved by nearly 90% compared to Basic FR. The functionality of the CFSM mechanism can be seen. Since the Tracker requires to face detection stage to find the positions and sizes of candidate bonding boxes, the process of face detection stage cannot be omitted. It only can reduce the complexity of face extraction and face matching.

The accuracy comparison of CFSM and C3D_method mechanisms is shown in Table 6. The FD frame count shows the counts of the frame that the CFSM or C3D_method mechanisms decide to apply the FD mechanism. The accuracy is the ratio of frame count of correct FD detection with total frame count. Although C3D_method can detect the abrupt cut well, it cannot identify the minor difference among the consequent frames from the moving people, dark scene, and rotating faces… etc. Therefore the FD frame count of C3D_method is fewer than that of CFSM. The accuracy of C3D_method will be limited. Besides, the C3D_method consumes a lot of execution time to identify the abrupt cut frames, even larger than basic FD mechanism, as shown in Fig. 10. In the video clip of "Changeling", it contains many abrupt cuts, but the C3D_method cannot detect them due to the clip is dark tone. The detected FD frame count is only 38. But the proposed CFSM mechanism can deal with this situation.

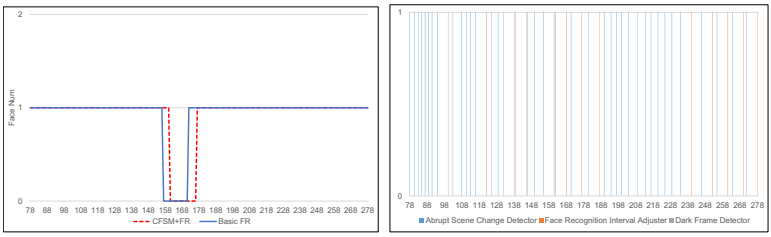
4.4 The performance of CFSM under four types of scene changes

The scene composition, transition methods, and movement of the screen during shooting of the video will bring different scene changes. The following will be divided into four types of scene changes, namely (1) generally scene change, (2) frequently scene change, (3) screen movement, and (4) screen fixation, to explore the detection function of the CFSM mechanism, and use four video clips, each with 200 frames, to discuss the difference in the number of detected faces between Basic FR and Basic FR with CFSM under these four scene change types. Here we use $\mathrm{CFSM}_{SkipRate}$ to indicate that after CFSM analysis, the amount of frames can be skipped, which accounts for the proportion of the total number of frames. $\mathrm{CFSM}_{MissRate}$ represents the amount of frames with different number of faces which is detected by Basic FR and Basic FR with CFSM, for the proportion of the total number of frames. The functionality of three analyzing mechanisms in the proposed CFSM, abrupt scene change detector, face recognition interval adjuster, and dark frame detector, are also illustrated. The following will discuss these four types of scene changes respectively.

4.4.1 Generally scene change

In the scenario of the general scene change, the foreground, background, and objects in the picture are not changed drastically, but the face in the picture may be different from the way the scene changes because of the shooting method. As shown in Fig. 11, this video clip is extracted from "Before Sunset" with a total of 200 frames. The analysis result of CFSM is shown in Fig. 12. Figure 12a compares the results of the number of the faces in the frame, which are detected by Basic FR (blue line) and Basic FR with CFSM (red dotted

Fig. 11 Part of the frames in the video of the *general scene change* type



(a) The number of faces detected by Basic FR and FR with CFSM. (b) Analysis results of the three mechanisms of CFSM.

Fig. 12 Analysis of CFSM mechanism on the video of *general scene change* type. **a** The number of faces detected by Basic FR and FR with CFSM. **b** Analysis results of the three mechanisms of CFSM

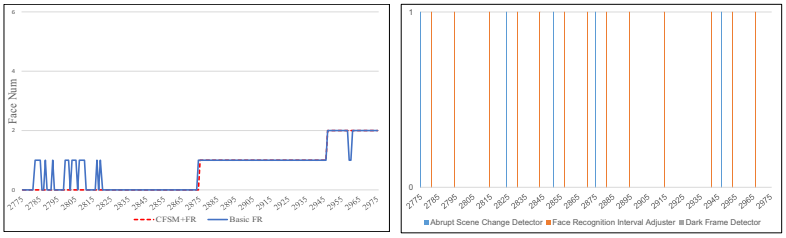
line), respectively. Figure 12b shows the three mechanisms of CFSM, abrupt scene change detector (blue line), face recognition interval adjuster (red line), and dark frame detector (gray line), that indicate the frame is determined to the face recognition system (Fig. 12).

According to the experimental results, $\text{CFSM}_{SkipRate}$ is about 74%. Comparing the results of Basic FR (blue line) and Basic FR with CFSM (red dotted line), only about 4% of $\text{CFSM}_{MissRate}$. The effectiveness of CFSM can be seen. Because the human face occupies a larger proportion of the frame, and it turns left and right. Although the background has not changed, a lot of abrupt scene changes can still be detected. Therefore, fewer frames can be omitted.

4.4.2 Frequently scene change

Due to the effects of shooting or video editing, there may be frequently scene changes in the video, as shown in Fig. 13, which is extracted from the video "Instant Family", and the result after CFSM analysis is shown in Fig. 14. $\text{CFSM}_{SkipRate}\text{CFSM}_{SkipRate}$ is about 90%, and $\text{CFSM}_{MissRate}$ is about 10%. In the 200 frame segments, four scene changes occurred, which are frame 2824, 2851, 2875 and 2947. It can be seen from Fig. 14b that when these four scene changes occur, the abrupt scene change detector (blue line) is correctly detected. At the same time, the face recognition interval adjuster (red line) is also correctly determined the time interval of face recognition (Figs. 14).

Fig. 13 Part of the frames in the video of the *frequently scene change* type



(a) The number of faces detected by Basic FR and FR with CFSM. (b) Analysis results of the three mechanisms of CFSM.

Fig. 14 Analysis of CFSM mechanism on the video of *frequently scene change* type. **a** The number of faces detected by Basic FR and FR with CFSM. **b** Analysis results of the three mechanisms of CFSM

4.4.3 Screen movement

Due to the needs of shooting, the screens of many movies are constantly moving, and scene changes will continue to occur. As shown in Fig. 15, this video clip is extracted from "Patriot Games", with a total of 200 frames. Due to the horizontal movement of the screen, scene changes continue to occur, so the $\text{CFSM}_{SkipRate}$ of the video clip is only 44%, but the $\text{CFSM}_{MissRate}$ is therefore reduced to 2%. It can be found from Fig. 16b that because the screen continues to move, the reason that triggers the face recognition system is mostly abrupt scene change detector (blue line), and only a few face recognition interval adjuster (red line) triggers (Fig. 16).

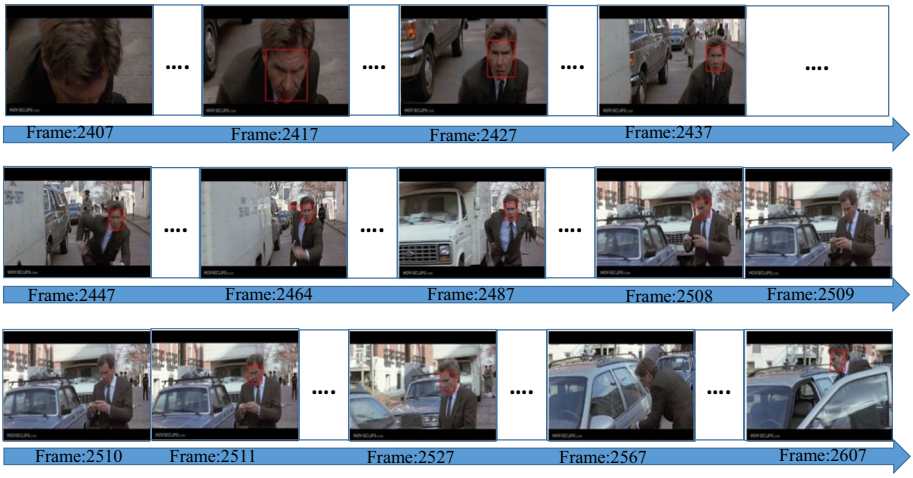
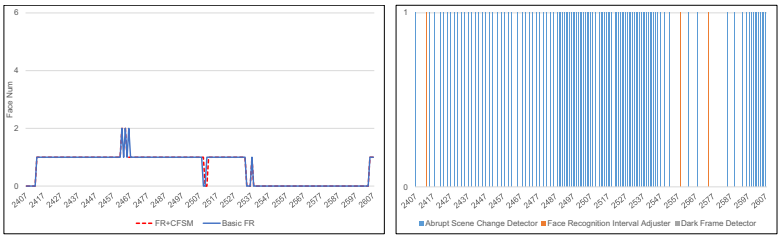


Fig. 15 Part of the frames in the video of the *screen movement* type



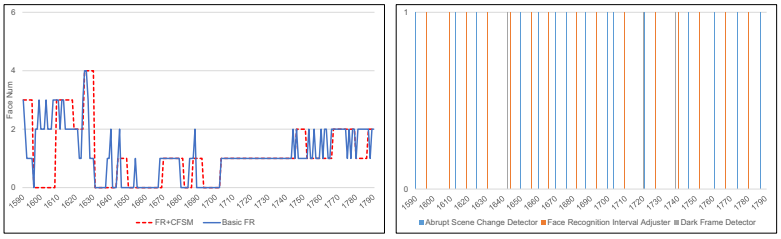
(a) The number of faces detected by Basic FR and FR with CFSM. (b) Analysis results of the three mechanisms of CFSM.

Fig. 16 Analysis of CFSM mechanism on the video of *screen movement* type. **a** The number of faces detected by Basic FR and FR with CFSM. **b** Analysis results of the three mechanisms of CFSM

4.4.4 Screen fixation

In the case of movie scenes or screens, if people continue to enter and exit the screen range, it will affect the accuracy of CFSM's judgment. As shown in Fig. 17, this video clip is also 2000 frames extracted from "Instant Family". In addition, when the proportion of people or faces in the picture is too small, or the face is constantly moving or rotating, the analysis accuracy of the CFSM mechanism will also be reduced. As shown in Fig. 18b, the $\text{CFSM}_{SkipRate}$ of this video clip is about 84%, but the $\text{CFSM}_{MissRate}$ is 35% (Fig. 18).

Fig. 17 Part of the frames in the video of the *screen fixation* type



(a) The number of faces detected by Basic FR and FR with CFSM. (b) Analysis results of the three mechanisms of CFSM.

Fig. 18 Analysis of CFSM mechanism on the video of *screen fixation* type. **a** The number of faces detected by Basic FR and FR with CFSM. **b** Analysis results of the three mechanisms of CFSM

5 Conclusions

In view of the increasing demand for real-time face recognition on embedded systems, this research is based on a three-stage neural network face recognition system and designed a new frame analysis mechanism, continuous frames skipping mechanism (CFSM), to analyze the video frame in real time. The CFSM system consists of three stages: the abrupt scene change detector (ASCD) to detect abrupt cut and gradual scene; the face recognition interval adjuster (FRIA) to skip a large number of unnecessary face detection frame; and the dark frame detector (DFD) to deal with the dark scenes. The proposed CFSM mechanism can decide whether to perform face recognition and effectively utilize the limited computing resources of the embedded system. The experimental results show that the CFSM mechanism proposed in this study can save up to 90% of the time under the NVIDIA Jetson TX2 system compared to the basic face recognition system. Compare to the existed face tracking technique and CNN based

abrupt cut detection mechanism, the proposed CFSM mechanism can help the basic face recognition system on the embedded system to achieve the real-time performance. Although CFSM mechanism can help to reduce the continuous frames in the streaming video that are not needed to redundantly detect human faces within the frames, the time-consumed MTCNN model for the face detection stage still limits the performance of whole face recognition system. A comprehensive neural network model that are included the functionalities of the continuous frame detection mechanism and efficient human face detection mechanism will help to improve the popularity and availability of the light-weight face recognition system for the embedded devices.

Acknowledgements This work is supported in part by the Ministry of Science and Technology of Republic of China, Taiwan under Grant MOST 105-2221-E-033-047.

References

1. Abu-El-Haija S, et al (2016) Youtube-8m: a large-scale video classification benchmark. arXiv, abs:1609.08675
2. Chen S et al (2018) MobileFaceNets: efficient CNNs for accurate real-time face verification on mobile devices. Chinese conference on biometric recognition 2018. Lecture notes in computer science, 10996, pp 428–438
3. Dabhade SB, et al (2017) Double layer PCA based hyper spectral face recognition using KNN classifier. International conference on current trends in computer, electrical, electronics and communication (CTCEEC), pp 289–293
4. Dadi HS, Pillutla GM (2016) Improved face recognition rate using HOG features and SVM classifier. IOSR J Electron Commun Eng IOSR-JECE 11:34–44
5. Danelljan M et al (2014) Accurate scale estimation for robust visual tracking. Proceedings of the British machine vision conference, pp 1–5
6. Deng J et al (2019) ArcFace: additive angular margin loss for deep face recognition. IEEE conference on computer vision and pattern recognition (CVPR), pp 4685–4694
7. Dhamecha TI (2016) On frame selection for video face recognition. In: Kawulok M (ed) Proceedings of advances in face detection and facial image analysis. Springer, Cham, pp 279–297
8. Gygli M (2018) Ridiculously fast shot boundary detection with fully convolutional neural networks. International conference on content-based multimedia indexing (CBMI), pp 1–4
9. Hassanien A et al (2017) Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. arXiv, abs:1705.03281
10. He K et al (2016) Deep residual learning for image recognition. IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778
11. He Q, He B, Zhang Y et al (2019) Multimedia based fast face recognition algorithm of speed up robust features. Multimed Tools Appl 78:24035–24045
12. Holmes SA, Klein G, Murray DW (2008) An $O(N^2)$ square root unscented Kalman filter for visual simultaneous localization and mapping. IEEE Trans Pattern Anal Mach Intell 31(7):1251–1263
13. Huang GB et al (2007) Labeled Faces in the wild: a database for studying face recognition in unconstrained environments. University of Massachusetts, pp 7–49
14. Jain AK, Ross AA, Nandakumar K (2011) Introduction to biometrics. Springer Science & Business Media, Berlin, pp 111–117
15. Jin X et al (2020) Efficient blind face recognition in the cloud. Multimed Tools Appl 79:12533–12550
16. Jose E et al (2019) Face recognition based surveillance system using FaceNet and MTCNN on Jetson TX2. 5th International conference on advanced computing & communication systems (ICACCS), pp 608–613
17. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Res Logist Q 2(1–2):83–97
18. Liu W et al (2017. SphereFace: deep hypersphere embedding for face recognition. IEEE conference on computer vision and pattern recognition (CVPR), pp 6738–6746
19. Murray S (2017) Real-time multiple object tracking-a study on the importance of speed. arXiv abs:1709.03572

20. Parveen P, Thuraisingham B (2016) Face recognition using multiple classifiers. 18th IEEE international conference on tools with artificial intelligence, pp 179–186
21. Qi X, Liu C, Schuckers S (2016) Key-frame analysis for face related video on GPU-Accelerated embedded platform. International conference on computational science and computational intelligence (CSCI), pp 682–687
22. Saez-Trigueros D, Meng L, Hartnett M (2018) Face recognition: from traditional to deep learning methods. arXiv, abs/1811.00116
23. Sajjad M et al (2020) Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities. Futur Gener Comput Syst 108:995–1007
24. Saypadith S, Aramvith S (2018) Real-time multiple face recognition using deep learning on embedded GPU System. Asia-Pacific signal and information processing association annual summit and conference (APSIPA ASC), pp 1318–1324
25. Schroff F, Kalenichenko D, Philbin J (2015) FaceNet: a unified embedding for face recognition and clustering. IEEE conference on computer vision and pattern recognition (CVPR), pp 815–823
26. Shi X et al (2018) Real-time rotation-invariant face detection with progressive calibration networks. IEEE conference on computer vision and pattern recognition (CVPR), pp 2295–2303
27. Sinha D, El-Sharkawy M (2019) Thin MobileNet: an enhanced MobileNet architecture. IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON), pp 280–285
28. Stekas N, van den Heuvel D (2016) Face recognition using local binary patterns histograms (LBPH) on an FPGA-Based system on chip (SoC). IEEE international parallel and distributed processing symposium workshops (IPDPSW), pp 300–304
29. Sujay SN, Reddy HM, Ravi J (2017) Face recognition using extended LBP features and multilevel SVM classifier. International conference on electrical, electronics, communication, computer, and optimization techniques, pp 1–4
30. Sun Y, Wang X, Tang X (2014) Deep learning face representation from predicting 10,000 classes. IEEE conference on computer vision and pattern recognition, pp 1891–1898
31. Taigman Y et al (2014) DeepFace: closing the gap to human-level performance in face verification. IEEE conference on computer vision and pattern recognition, pp 1701–1708
32. Tran D et al (2015) Learning spatiotemporal features with 3D convolutional networks. Proceedings of the IEEE international conference on computer vision
33. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vision 57(2):137–154
34. Wang H et al (2018) CosFace: large margin cosine loss for deep face recognition. IEEE conference on computer vision and pattern recognition (CVPR), pp 5265–5274
35. Yang F, Paindavoine M (2003) Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification. IEEE Trans Neural Netw 14(5):1162–1175
36. Zhang K et al (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
37. Zhang M et al (2019) Embedded face recognition system based on multi-task convolutional neural network and LBP features. IEEE international conference of intelligent applied systems on engineering (ICIASE), pp 132–135
38. Zhang S et al (2019) Faceboxes: a CPU real-time and accurate unconstrained face detector. Neurocomputing 364:297–309